Universitat de Girona
**Escola Politècnica Superior**

Master's Thesis

Study: Master in Data Science and Machine Learning

Title: CNV Detection using Machine Learning in Targeted Sequencing

Document: Memory

Student: Oriol Canal Pujol

Tutor: Maria Beatriz Lopez Ibañez
Department: Grup de Recerca en Enginyeria de Control i Sistemes Intel·ligents (EXIT)
Àrea: Enginyeria de sistemes i automàtica
Cotutor: Bernat del Olmo Cabestré
Institution: Universitat de Girona (UdG)

Call: September 2024

Universitat de Girona
**Escola Politècnica Superior**

MASTER THESIS

# CNV Detection using Machine Learning in Targeted Sequencing

*Author:*
Oriol CANAL PUJOL

September 2024

Master in Data Science and Machine Learning

*Tutors:*
Beatriz LOPEZ IBAÑEZ
Bernat DEL OLMO CABESTRÉ

# SUMMARY

This thesis introduces the development and validation of Targeted-CNV-Learner, an advanced machine learning framework designed to enhance the detection of Copy Number Variations (CNVs) in targeted gene panels. CNVs play a critical role in various genetic disorders, and their precise identification is vital for clinical diagnostics. However, current CNV detection methods, especially those used in targeted sequencing, often suffer from high false positive rates and limited precision, underscoring the need for more robust solutions.

The primary objective of this work was to address these challenges by proposing a novel methodology that integrates the outputs of multiple CNV detection algorithms. Targeted-CNV-Learner leverages the strengths of three widely used algorithms—GATK gCNV, GRAPES, and DECoN—alongside genomic features to accurately differentiate between true CNVs and artifacts. The model was trained on in silico CNVs introduced into clinical samples and validated using real CNV data obtained from the SUDD147 gene panel, which is employed in the study of sudden cardiac death. By combining multiple algorithms, this approach significantly enhances precision while reducing false positive rates.

A comprehensive analytical pipeline was developed to automate the processes of sample analysis, data labeling, and model training. This pipeline facilitated the systematic evaluation and benchmarking of Targeted-CNV-Learner against individual CNV detection algorithms. The results showed that Targeted-CNV-Learner outperformed the standalone methods, achieving the highest accuracy (95.5%) on the test set and a substantial reduction in false positives compared to DECoN, GRAPES, and GATK gCNV. Moreover, the model maintained high sensitivity, effectively detecting CNVs while minimizing the need for unnecessary orthogonal validations when validated. These findings were further confirmed through real CNV data, where Targeted-CNV-Learner matched the high sensitivity of GRAPES.

Beyond its application to the SUDD147 panel, the methodology developed in this work is adaptable to other gene panels, offering a versatile and scalable solution for CNV detection across a range of targeted sequencing applications. By improving detection accuracy and reducing the burden of extensive validation, Targeted-CNV-Learner presents a cost-effective and reliable tool for clinical diagnostics.

In conclusion, this thesis demonstrates the powerful potential of machine learning to enhance CNV detection in targeted sequencing, providing a more accurate and efficient alternative to conventional algorithms. The successful development and validation of Targeted-CNV-Learner represent a significant advancement in genetic diagnostics, with the potential to improve patient outcomes through more precise genomic analyses.

# Acknowledgments

I would like to express my deepest gratitude to Dra. Beatriz López, the tutor of my master's thesis, whose unwavering support and guidance were essential throughout this journey. Her profound knowledge of machine learning models was invaluable to my work, and her commitment to helping me, even when the subject matter diverged from her background, was truly remarkable. She made a tremendous effort to understand the complex biological and genetic aspects of this project, always showing genuine curiosity and dedication. For this, I am immensely grateful.

I would also like to extend my heartfelt thanks to Dr. Bernat del Olmo, an experienced bioinformatician at the UDMMP and the author of the GRAPES software used in this thesis. His expertise in genetic information, bioinformatics tools, and CNV detection was indispensable to this project. His profound knowledge of CNVs, demonstrated through the development of GRAPES, guided my work from start to finish. His thoughtful advice and generous sharing of knowledge were critical to the success of this thesis.

My sincere thanks also go to the entire team at UDMMP and IDIBGI for granting me access to their facilities, including servers and genetic files, which were vital to my research. Their support made it possible for me to carry out this project in a conducive and resource-rich environment.

Finally, I would like to thank the Universitat de Girona for giving me the opportunity to enroll in this master's program. The education and resources provided by the university have equipped me with the knowledge and skills necessary to successfully complete this thesis. I am deeply appreciative of this opportunity and the foundation it has given me to advance in this field.

# Acronyms

**aCGH**: Array comparative genomic hybridization
**CNV**: Copy number variant
**DECoN**: Detection of Exon Copy Number
**DNA**: Deoxyribonucleic acid
**FDR**: False discovery rate
**GATK**: Genome Analysis Toolkit
**GRAPES**: Germline Rearrangement Analysis from Panel Enrichment Sequencing
**HMM**: Hidden Markov Model
**ML**: Machine learning
**NGS**: Next-generation sequencing
**PCA**: Principal component analysis
**PCR**: Polymerase chain reaction
**RNA**: Ribonucleic acid
**SNP**: Single nucleotide polymorphism
**SUD**: Sudden unexplained death
**SUDD147**: Sudden Cardiac Death genetic panel
**SV**: Structural variant
**UDMMP**: Unit of Molecular Diagnostic and Personalized Medicine
**WES**: Whole exome sequencing
**WGS**: Whole genome sequencing

# Contents

# List of Figures

# List of Tables

CHAPTER 1

# Introduction

In this chapter the main motivations for this work are described, alongside with it's main contributions.

## 1.1 Motivation

The rapid advancements in genomic technologies have opened up new frontiers in the study of genetic diseases and personalized medicine [12]. The ability to decode the human genome with unprecedented speed and accuracy has transformed our understanding of genetic variations and their role in health and disease. This thesis aims to explore these advancements, focusing on the powerful capabilities of NGS and its application in detecting genetic variations that can influence disease outcomes.

Understanding genetic variations is crucial for diagnosing and treating various medical conditions. Among these variations, CNVs play a significant role due to their potential impact on gene dosage and expression [28]. Despite their importance, CNVs present unique challenges in detection and interpretation [19], particularly in targeted gene panels used for studying specific conditions such as sudden cardiac death. Addressing these challenges requires sophisticated analytical approaches and robust datasets.

CNVs are a major cause of several genetic disorders, making their detection an essential component of genetic analysis pipelines. Current methods for detecting CNVs from targeted-sequencing data are limited by high false positive rates and low concordance because of the inherent biases of individual algorithms.

## 1.2 Objectives

The primary objective of this project is to introduce and develop Targeted-CNV-Learner, a machine learning-driven software designed to significantly enhance the detection of CNVs in clinical diagnostics. Targeted-CNV-Learner aims to integrate the outputs of multiple CNV detection algorithms, learning to identify true CNVs with higher accuracy by leveraging both algorithm-specific and genomic features. This innovative model seeks to surpass the limitations of existing standalone algorithms by improving precision and selectivity, ultimately providing a more reliable and trustworthy solution for CNV detection.

In clinical settings, the accuracy of CNV detection is paramount to ensuring reliable diagnostic outcomes. While computational algorithms are powerful tools for identifying potential CNVs, their predictions can sometimes result in false positives, particularly in complex genomic regions. Therefore, confirming these CNV calls using orthogonal validation methods is crucial to achieving high confidence in the results (see Figure1.1) [13].

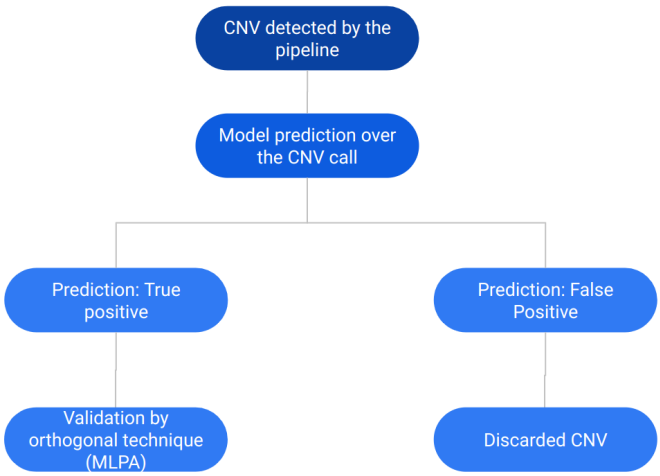



Figure 1.1: Validation process over the calls made by the CNV detection pipeline.

One of the most widely adopted techniques for this validation is MLPA. MLPA quantitatively measures changes in DNA copy number, allowing for independent verification of computational predictions. When a CNV is detected by the pipeline, the model's prediction over the CNV call plays a pivotal role in determining whether it is a true positive or a false positive. If deemed a true positive, the CNV is subjected to validation via MLPA, which provides a robust wet-lab confirmation.

A key goal of Targeted-CNV-Learner is to minimize the false positive rates that frequently arise from current CNV detection methods. Reducing false positives is critical because it directly decreases the need for costly and time-consuming orthogonal validations, such as MLPA, thus alleviating both the financial and logistical burdens on clinical diagnostic workflows.

This process significantly minimizes the likelihood of false positives, ensuring that only genuine CNVs are considered in the final diagnosis. By integrating computational predictions with an orthogonal technique like MLPA, the diagnostic workflow achieves a higher level of precision, reducing the risk of errors and bolstering the confidence in clinical decision-making.

However, while minimizing false positives is important, it is equally essential to maximize the recall of genuine CNVs. A false negative—failing to detect a CNV—could result in a missed genetic variant that might significantly impact a patient's health. Therefore, Targeted-CNV-Learner aims not only to reduce false positives but also to ensure that true CNVs are accurately identified. This balance between precision and recall will optimize the efficiency of subsequent validation efforts, making the diagnostic process more reliable and cost-effective, while still maintaining the high standards required for clinical applications.

To rigorously validate the performance of Targeted-CNV-Learner, we conduct extensive testing on a dataset comprising 400 samples from the SUDD147 genetic panel, developed by the UDMMP unit at Hospital Josep Trueta. This dataset is utilized for training, testing, and benchmarking the model, allowing us to demonstrate the enhanced accuracy and reliability

of CNV detection, as well as the reduction in the need for orthogonal validation.

In summary, Targeted-CNV-Learner is poised to offer a breakthrough in CNV detection by integrating multi-algorithm outputs, improving diagnostic precision, reducing costs, and addressing the pressing challenges of false positives and extensive validation in clinical genomics.

## 1.3 Thesis structure

This master's thesis is organized into several key sections to guide the reader through the research process and findings.

- **Preliminaries**: This section begins by introducing the necessary terminology and foundational concepts in both NGS and machine learning. A comprehensive background on NGS is provided to assist readers who may not have prior knowledge in these fields.
- **State of the art**: Following the preliminaries, this section reviews and explains the current leading algorithms for Copy Number Variation (CNV) detection. It provides an overview of the main approaches available today, setting the context for the research presented in this thesis.
- **Planning**: In this section, the step-by-step process undertaken throughout the project is detailed. This includes the planning and execution stages that led to the development and evaluation of the proposed solution.
- **Methodology**: Here, the thesis delves into the methodology behind Targeted-CNV-Learner, the central algorithm of this research. The technical details of how Targeted-CNV-Learner functions are thoroughly explained.
- **Results and discussion**: This section presents and analyzes the results obtained by applying Targeted-CNV-Learner to the SUDD147 panel samples. The performance of Targeted-CNV-Learner is compared with existing state-of-the-art algorithms, highlighting its strengths and areas for improvement.
- **Limitations and conclusion**: Finally, the thesis discusses the limitations of Targeted-CNV-Learner and offers a conclusive summary of the research findings, including suggestions for future work.

CHAPTER 2

# Preliminaries

The aim of this chapter is to present the theoretical background required for understanding the project contributions.

## 2.1 Terminology

This section introduces essential NGS terminology and some basics about ML to facilitate understanding of the project.

### 2.1.1 Terminology on NGS

- **Exon**: An exon is a segment of a gene that codes for a portion of the final mature mRNA, produced after the removal of introns (non-coding regions) through RNA splicing (Figure 2.1). Exons contain the information that directs protein synthesis and play a critical role in gene expression and regulation.

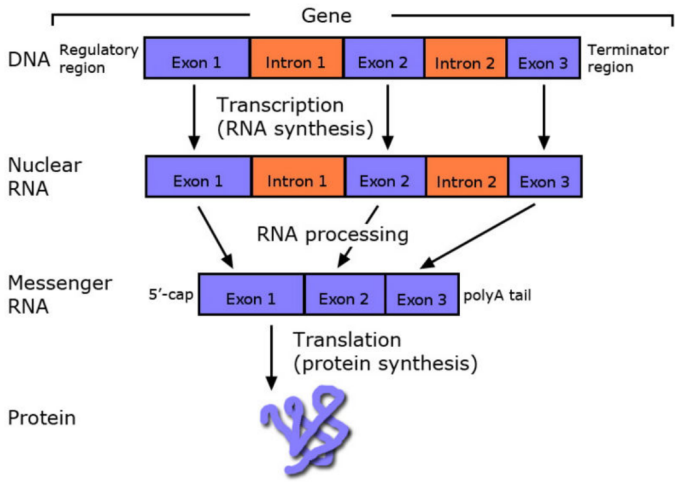



Figure 2.1: Schematic representation of gene expression from DNA to protein. The process begins with transcription, where a gene's DNA sequence (comprising exons and introns) is transcribed into nuclear RNA. During RNA processing, introns are removed, and exons are spliced together to form mature messenger RNA (mRNA). The mRNA then undergoes translation, where the encoded genetic information is used to synthesize a protein.

- **Gene panel**: Refers to a specific set of genes or genomic regions selected for sequencing and analysis. Panels are designed to focus on regions of interest that are relevant to particular research or clinical questions. In this project, we will be using a panel called SUDD 147, which contains 147 genes related to Sudden Cardiac Death.

- **Mappability**: Mappability in targeted next-generation sequencing refers to the ability to accurately align sequencing reads to a reference genome, ensuring that each read can be uniquely and correctly mapped to a specific genomic location. High mappability regions are essential for accurate variant detection and minimizing false positives. It is influenced by factors such as the sequence complexity, repeat regions, and the quality of the reference genome.

- **GC content**: GC content refers to the percentage of guanine (G) and cytosine (C) bases in a DNA sequence. In targeted NGS, regions with high or low GC content can affect sequencing efficiency and accuracy, potentially leading to biases in read coverage and challenges in variant detection.

- **Autosomal chromosome**: An autosomal chromosome is any chromosome that is not a sex chromosome. In humans, there are 22 pairs of autosomal chromosomes, which carry the bulk of genetic information influencing most inherited traits and conditions. These chromosomes are inherited equally from both parents and are present in both males and females.

- **CNV breakpoint**: A CNV breakpoint is the precise genomic location where a segment of DNA has been duplicated or deleted, leading to variations in the number of copies of that segment within the genome. These breakpoints mark the boundaries of the CNV event.

- **Read Depth**: Read depth refers to the number of times a specific nucleotide in a genome is sequenced during a sequencing experiment (Figure 2.2). Higher read depth increases the accuracy of detecting genetic variants and ensures reliable coverage of the genome. It is a critical factor in various genomic analyses, including variant calling and CNV detection.

- **CNV detection algorithm**: CNV detection algorithms are computational tools designed to identify copy number variations within genomic data. These algorithms analyze sequencing reads to detect regions where the number of DNA copies differs from the normal two copies per autosomal locus. Accurate CNV detection is crucial for identifying genetic variations that may contribute to diseases or phenotypic traits.

- **CNV detection algorithm call**: The call of a CNV detection algorithm refers to the output generated by the algorithm, indicating the presence or absence of a CNV at a specific genomic location. This call typically includes information such as the coordinates of the CNV, the type of variation (e.g., deletion, duplication), and the predicted

number of copies at that locus. The accuracy and reliability of these calls are crucial, as they directly impact downstream analyses and interpretations. In practice, the call of a CNV detection algorithm can be used to identify genetic alterations associated with diseases, traits, or responses to treatments, making it an essential component of genomic research and clinical diagnostics.

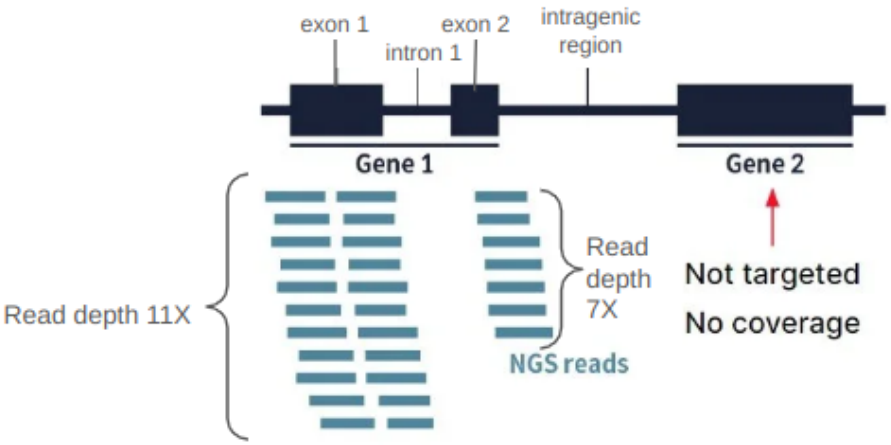



Figure 2.2: Illustration of the concepts of exons, introns, read depth, and targeted sequencing in the context of Next-Generation Sequencing (NGS) (From [1]). Gene 1 consists of two exons (exon 1 and exon 2) and one intron (intron 1), along with an intragenic region between Gene 1 and Gene 2. The read depth is indicated by the amount of stacked horizontal lines. For Gene 1, exon 1 has a read depth of 11X, and exon 2 has a read depth of 7X, demonstrating varying coverage across the gene. In contrast, Gene 2 is not targeted by the panel, resulting in no coverage for this gene.

### 2.1.2 Terminology on ML

Some essential ML terminology to facilitate understanding of the project are the following:

- **Model**: A model in machine learning is a mathematical representation of a real-world process, built using algorithms to find patterns in data. It makes predictions or decisions based on input data, learning from training data to generalize to new, unseen data.

- **Decision Tree**: A decision tree is a machine learning model that splits data into branches based on feature values to make predictions. Each internal node represents a decision on a feature, each branch represents an outcome of the decision, and each leaf node represents a final prediction or class.

- **Random Forest**: A random forest is an ensemble machine learning algorithm that combines multiple decision trees to improve prediction accuracy and robustness. Each

tree in the forest is built on a random subset of the data and features, and the final output is determined by averaging the predictions (regression) or taking a majority vote (classification). This method reduces overfitting and enhances model generalization.

- **Overfitting**: Overfitting occurs when a machine learning model learns the training data too well, capturing noise and specific patterns that do not generalize to new data. This results in high accuracy on the training set but poor performance on unseen data.

## 2.2 Background

This section begins by delving into the fundamentals and transformative impact of NGS technology. It then examines the challenges associated with CNV detection and the innovative solutions developed to overcome these hurdles.

### 2.2.1 Next generation sequencing

In recent years, advancements in genetics technology have significantly enhanced our ability to study the genetic material of living organisms. One of the most groundbreaking developments in this field is NGS. NGS is a transformative genomic sequencing technology that enables researchers and clinicians to rapidly and cost-effectively analyze vast amounts of genetic data. This advanced method facilitates the identification of disease-causing pathogens [6] [31] and cancer-associated mutations [11] in patient genomes with enhanced speed and precision. By significantly reducing the time and cost associated with genetic analysis, NGS has revolutionized the fields of genomics and personalized medicine, providing critical insights into the genetic underpinnings of various diseases.

NGS encompasses several sequencing approaches, each tailored to specific research and clinical needs (Figure 2.3):

- **WGS**: This approach provides a comprehensive analysis of an organism's entire genome, sequencing all DNA present, including both coding and non-coding regions. While WGS offers an extensive view of genetic variations across the whole genome, it can be resource-intensive due to the large volume of data generated.

- **WES**: In contrast, WES focuses specifically on the exonic regions of the genome—the segments that encode proteins. Covering only about 1-2% of the genome, WES is more targeted and cost-effective than WGS while still providing valuable insights into mutations that affect protein function.

- **Targeted Sequencing**: This method concentrates on specific genomic regions of interest, such as particular genes or genomic hotspots. By focusing on predetermined areas, targeted sequencing allows for high-resolution analysis of selected regions, which can be especially useful for studying specific diseases or genetic conditions.


Figure 2.3: Comparative of NGS strategies: WGS, WES and targeted sequencing

NGS represents a substantial advancement over traditional sequencing methods, such as Sanger sequencing, by enabling high-throughput analysis. This capability is particularly beneficial for comprehensive genetic studies, allowing for the simultaneous examination of multiple samples and the detection of a wide range of genetic variations. NGS efficiently identifies SNPs, which are single nucleotide differences contributing to genetic diversity, as well as insertions and deletions (indels) that can impact gene function depending on their genomic context.

Another critical type of genetic variation detectable by NGS is CNVs. CNVs involve duplications or deletions of large genomic segments, which can range from thousands to millions of base pairs. These structural variations influence gene dosage and expression levels [30] and are implicated in various genetic disorders, including developmental diseases [32] [34] and cancers [36] [5].

The high-throughput nature of NGS allows for the detailed examination of genetic variation across populations, enhancing our ability to identify disease-causing mutations with unparalleled speed and accuracy. As NGS technology continues to advance, its impact on

genomics and personalized medicine is expected to grow, further deepening our understanding of the genetic mechanisms that underpin health and disease.

### 2.2.2 Targeted sequencing

Targeted sequencing is a specialized approach in NGS that enables researchers to selectively sequence specific regions of interest within the genome. This method allows for high read depth of selection regions, enhancing the detection of low-frequency variants and minimizimng data processing costs. It is particularly useful for studying specific diseases or conditions, such as hereditary cancer or inherited heart diseases [22], where certain genes or mutations are known to be relevant, making it a valuable tool for both research and clinical diagnostics. Unlike whole-genome sequencing, which sequences the entire genome indiscriminately, targeted sequencing focuses on predefined genomic regions (Figure 2.3). This method is particularly advantageous in studies where a comprehensive analysis of the entire genome is unnecessary or cost-prohibitive.

Targeted sequencing involves several key steps (Figure 2.4):

- **Genomic DNA Library Preparation**: The initial step in targeted sequencing is the preparation of a genomic DNA library. This involves extracting genomic DNA from a sample, such as blood or tissue, and fragmenting it into smaller pieces using mechanical or enzymatic methods. The fragmented DNA ends are then modified to ensure they are ready for sequencing, and short synthetic sequences called adapters are attached to both ends of each fragment. Typically, the library is amplified using PCR to increase the quantity of DNA fragments.

- **Targeted DNA Enrichment**: Once the DNA library is prepared, the next step is to enrich for the regions of interest. This is achieved through hybridization capture, where probes—short, single-stranded DNA or RNA molecules that are complementary to the target regions—are used. These probes bind to their complementary sequences within the DNA library. The probe-bound DNA fragments are then captured using magnetic beads or another separation method, while non-target sequences are washed away. Finally, the target DNA fragments are released from the beads, resulting in an enriched library that predominantly contains the regions of interest.

- **Paired-End Sequencing**: The enriched DNA library is then subjected to paired-end sequencing, which involves sequencing both ends of each DNA fragment using a sequencing platform, such as Illumina. This platform reads the nucleotide sequence from each end of the DNA fragment, creating two reads per fragment. This paired-end approach improves the accuracy of alignment and variant detection.

- **Alignment to the Human Reference Genome**: The sequencing reads are mapped to a reference genome, such as GRCh38, to determine their origin within the genome. Bioinformatics tools like BWA or Bowtie align the paired-end reads to the reference

genome, and the alignment information is stored in SAM (Sequence Alignment/Map) or BAM (Binary Alignment/Map) files, indicating the position of each read in the reference genome.

- **Variant Detection and Annotation**: Finally, the aligned reads are analyzed to identify genetic variants, which are differences between the sequenced sample and the reference genome. Bioinformatics tools such as GATK or FreeBayes detect SNPs, insertions, deletions and CNVs. Quality filters are applied to remove low-confidence variants, and annotation tools like ANNOVAR, SnpEff or Variant Effect Predictor provide information on the genomic context and potential significance of the variants.

By following these steps, targeted sequencing provides a focused and efficient method for analyzing specific genomic regions, facilitating the identification of genetic variations associated with diseases and advancing our understanding of genetic underpinnings in various biological processes.

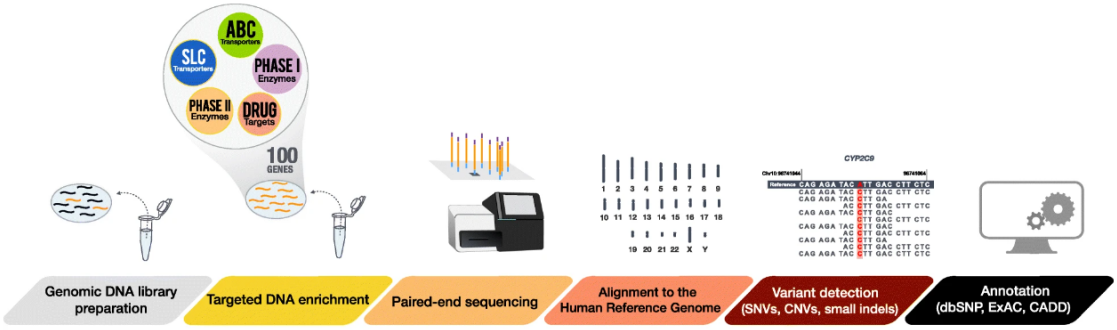



Figure 2.4: Illustrates the comprehensive workflow of genomic DNA targeted sequencing.

### 2.2.3 Copy number variants

CNVs are a form of SV characterized by the duplication or deletion of large segments of DNA, ranging from thousands to millions of base pairs (Figure 2.5). These variations can significantly impact gene dosage and expression levels, influencing phenotypic diversity and contributing to various disease processes. CNVs are particularly notable for their role in complex diseases, such as neurodevelopmental disorders, psychiatric conditions, radiological disorders and various cancers [15] [20].

CNVs can affect the patients in several ways:

- **Gene dosage**: Changes in the number of gene copies can lead to overexpression or underexpression of genes, affecting cellular function and potentially leading to disease [25]. For instance, duplications of oncogenes can drive cancer progression [10], while deletions of tumor suppressor genes can remove critical regulatory mechanisms that prevent uncontrolled cell growth [35].

- **Gene disruption**: CNVs can interrupt the coding sequence of genes, potentially leading to truncated or nonfunctional proteins [27]. This can result in loss-of-function effects that disrupt normal biological processes.

- **Regulatory elements**: CNVs can encompass regulatory regions such as promoters or enhancers, altering gene expression patterns and contributing to phenotypic variation and disease susceptibility [7].

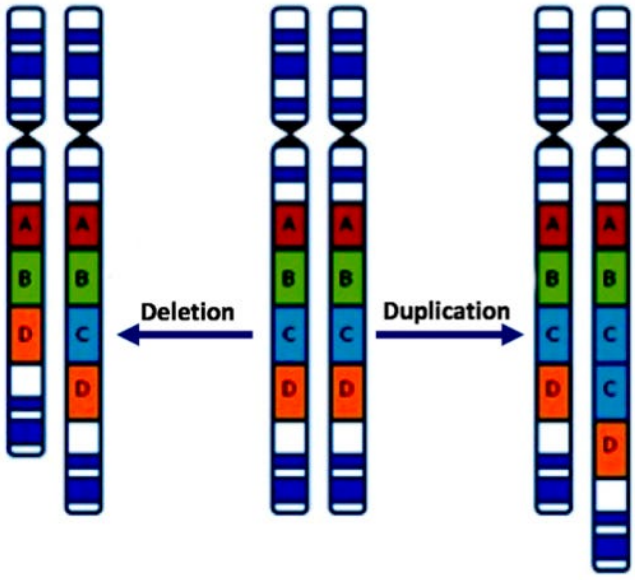



Figure 2.5: Illustration of a deletion and a duplication of the chromosomal region C.

### 2.2.4 Detection of CNVs

NGS has revolutionized genetic testing for Mendelian conditions, particularly in the detection of SNPs and small insertions and deletions. However, the detection of larger structural rearrangements, such as CNVs, presents unique challenges due to the inherent limitations of NGS technology, including short read lengths, GC-content bias, and uneven coverage [29]. Despite these obstacles, the identification of germline CNVs remains crucial, as these variants are implicated in numerous hereditary diseases, making their accurate detection a vital component of comprehensive genetic diagnostics.

Traditionally, the detection of CNVs in clinical settings has relied on methods such as MLPA and aCGH. [33] . While these techniques are highly reliable, they are also time-consuming, expensive, and often limited in scope, typically focusing on a subset of genes. This can lead to the exclusion of significant genomic regions, especially in single-gene testing approaches. As a result, leveraging NGS data for initial CNV screening offers a substantial advantage, potentially reducing the reliance on more resource-intensive methods like MLPA and aCGH, and streamlining the diagnostic process.

With the advancement of NGS technologies, diagnostics laboratories now commonly use NGS data to detect CNVs through specialized algorithms that analyze read-depth information [14]. These algorithms identify regions of the genome where deviations in coverage suggest the presence of CNVs (Figure 2.6. The high coverage depth characteristic of targeted NGS further enhances the sensitivity and accuracy of CNV detection, making it an increasingly valuable tool in genetic diagnostics. Despite these advancements, MLPA and aCGH are still employed as orthogonal techniques to validate CNV calls generated by NGS-based algorithms. This dual approach ensures that the CNVs identified are accurate and reliable, thereby integrating the strengths of both traditional and modern methods. By combining NGS with these validation techniques, laboratories can achieve more comprehensive and precise genetic diagnostics, further advancing the field of personalized medicine.

### 2.2.5 Challanges in detecting CNVs in targeted sequencing

Numerous tools have been developed for CNV detection from NGS data. Most of these tools were originally designed for whole-genome or whole-exome sequencing and often struggle with the sparser data generated from targeted NGS panels used in routine genetic testing. Despite these challenges, several algorithms have been specifically adapted or developed to detect CNVs in targeted sequencing. Notable among these are DECON [2], GATK gCNV [3], and GRAPES. Each of these algorithms employs distinct strategies to analyze read depth, normalize data, and identify CNVs, contributing to their effectiveness in various research and clinical applications.

Detecting CNVs in targeted sequencing remains a difficult task due to several intrinsic challenges. The first major challenge is the inherent variability in read depth, which can be influenced by factors such as target capture efficiency, sequencing depth, and GC content. Variations in read depth can lead to both false positives and false negatives in CNV detection,

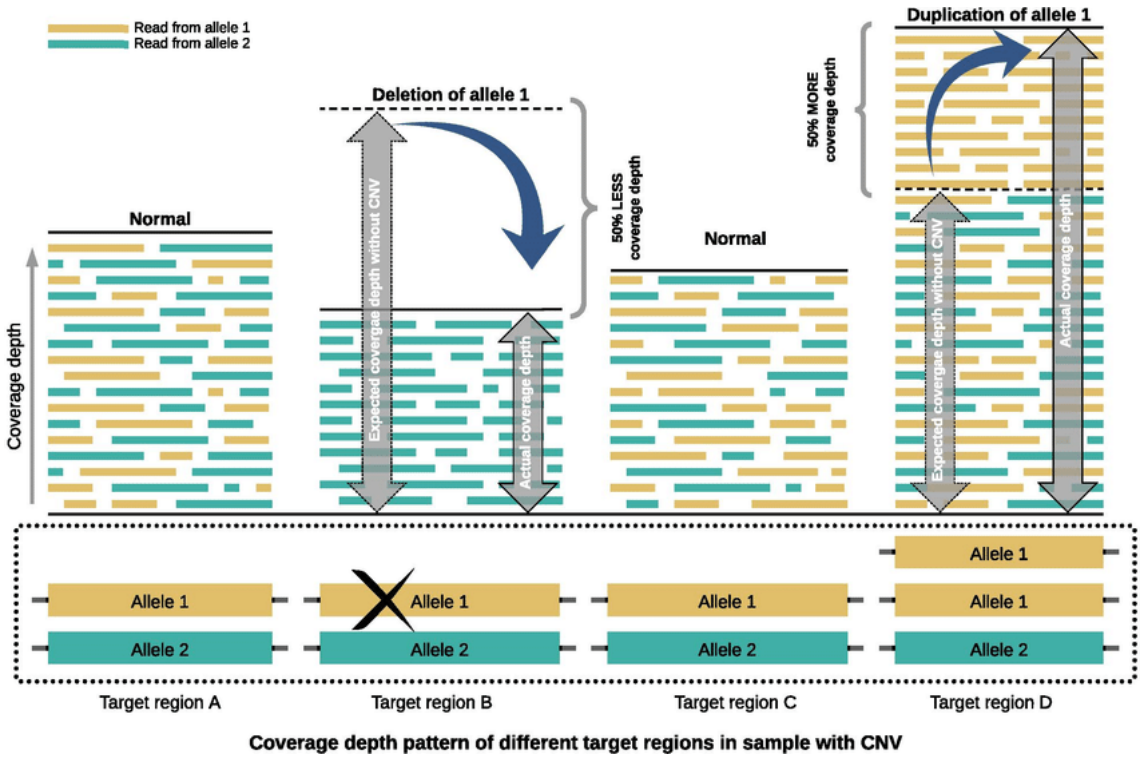



Figure 2.6: CNV detection using coverage depth information. This figure from [29], depicts the change in coverage depth of different target regions in a sample in the case of CNV events. Deletion of allele 1 in target region B reduces the coverage in that region by 50% compared to the expected coverage depth for the region. Duplication of allele 1 in target region D increases the coverage in that region by 50%, agains compared to the normal coverage depth for the region.

making it crucial to apply robust normalization techniques to correct for these biases.

Another challenge is the short read lengths characteristic of NGS data, which can complicate the accurate alignment of reads to the reference genome, particularly in repetitive regions or regions with complex structural variations. Misalignment can lead to erroneous CNV calls, necessitating the use of sophisticated alignment algorithms and quality control measures to ensure accurate mapping.

Additionally, the targeted nature of sequencing panels means that only specific regions of the genome are sequenced, resulting in sparser data compared to whole-genome or whole-exome sequencing. This sparsity can make it difficult to accurately detect CNVs, especially those that span large genomic regions or occur in low-complexity regions. The reliance on targeted panels also means that any biases in probe design or capture efficiency can disproportionately affect the accuracy of CNV detection in certain regions.

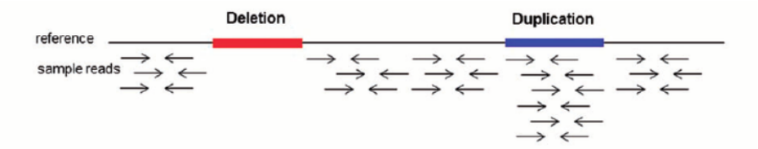
CHAPTER 3

# State of the art

The detection of CNVs in targeted sequencing has seen substantial advancements due to improvements in sequencing technologies and bioinformatics tools.

Algorithms for CNV detection in targeted sequencing typically leverage read depth information, which reflects the number of sequencing reads mapping to specific genomic regions. Variations in read depth can indicate the presence of CNVs, but accurately detecting these variations requires overcoming technical biases, such as GC-content bias, sequencing depth variability, and capture efficiency inconsistencies. A common strategy used by different CNV callers is to apply various statistical distributions to model the aggregate read depth of the exons and use read-depth fluctuations between adjacent targeted regions to identify duplication or deletion events (Figure 3.1)

Figure 3.1: Figure illustrating how deletions and duplications are respectively decreasing and increasing the read depth in the affected genomic location.




Three prominent algorithms used for CNV detection in targeted sequencing are DECON, GATK gCNV, and GRAPES. These tools are designed to handle the nuances of targeted sequencing data, employing advanced statistical models and normalization techniques to distinguish true CNVs from background noise. Each algorithm has its unique approach to read depth analysis, normalization, and segmentation, contributing to their effectiveness in both research and clinical applications.

## 3.1 GATK gCNV

The Genome Analysis Toolkit (GATK) is a comprehensive suite of tools designed to facilitate the discovery of genetic variants from high-throughput sequencing data. Developed by the Broad Institute, GATK is widely recognized for its robustness, accuracy, and efficiency in processing and analyzing genomic data. GATK gCNV (germline Copy Number Variants) is a specialized component within the GATK toolkit designed to detect CNVs in germline genomes. GATK-gCNV pipeline (Figure 3.2) begins by collecting coverage information from genome-aligned reads over a set of predefined genomic intervals (a). Next, the original interval list is filtered to remove coverage outliers, unmappable genomic sequence, and regions of segmental duplications (b). Then, samples are clustered into batches based on

read-depth profile similarity using PCA and each batch is processed separately (c). Chromosomal ploidies are inferred using total read-depth of each chromosome (d). Finally, The GATK-gCNV model learns read-depth bias and noise and iteratively updates copy number state posterior probabilities until a selfconsistent state is obtained; after convergence, constant copy number segments are found using the Viterbi algorithm along with segmentation quality scores.

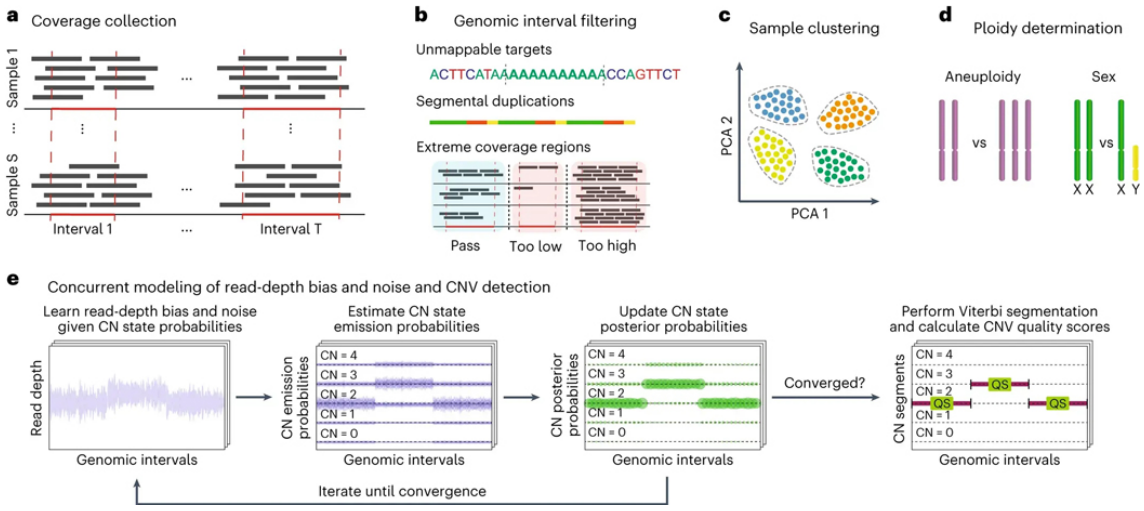



Figure 3.2: GATK-gCNV pipeline steps.

## 3.2 DECoN

Detection of Exon Copy Number Variants (DECoN) is a specialized algorithm designed to detect CNVs in exon regions using targeted sequencing data. To distinguish true CNVs from random fluctuations in read counts, DECoN employs a Bayesian statistical framework. This approach models the expected distribution of read counts and compares it to the observed data, estimating the likelihood that a given deviation is due to a CNV rather than noise. The Bayesian framework provides a rigorous method for assessing the confidence in each detected CNV. After completing the statistical analysis, DECoN identifies exons with significant deviations in copy number, flags them as potential CNVs, and associates a Bayesian factor with each, which helps differentiate true positive calls from noise.

## 3.3 GRAPES

Germline Rearrangement Analysis from Panel Enrichment Sequencing (GRAPES) is a sophisticated computational algorithm designed to detect CNVs and structural variations in genomic data using paired-end sequencing. Developed by B. del Olmo, the co-supervisor

of this thesis, GRAPES leverages both read depth and breakpoint information to accurately identify genomic rearrangements.(Figure 3.3)

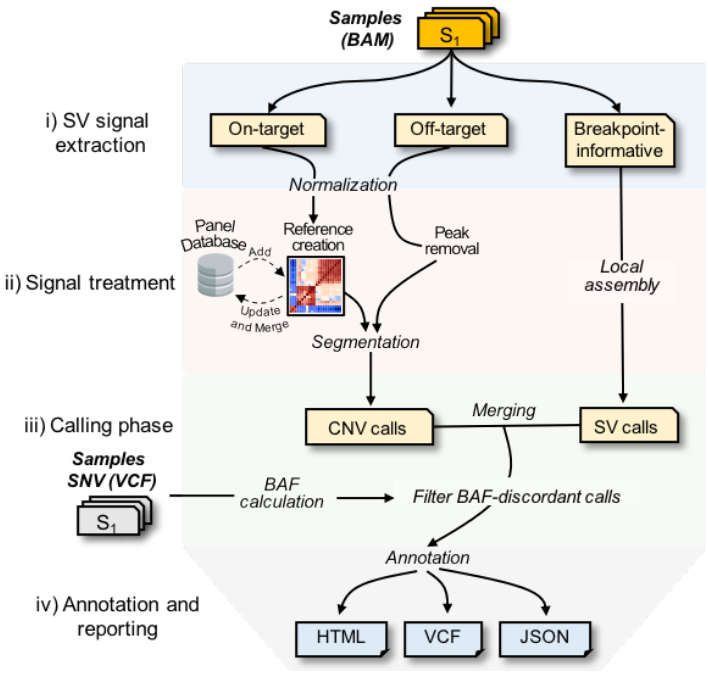



Figure 3.3: GRAPES workflow.

GRAPES extracts read signatures indicative of structural variations from both on-target and off-target reads. By analyzing the read depth across the genome, the algorithm detects regions with abnormal copy numbers, such as deletions or duplications. Additionally, GRAPES identifies breakpoints by extracting breakpoint-informative reads, including discordant read-pairs, soft-clipped reads, and split-reads.

The algorithm normalizes read depth by GC-content and DNA-capturing specificity. Highly correlated samples are clustered together to create baseline illustrating how deleerences, which are stored in a database. Finally, a 4-state Hidden Markov Model, based on a Gaussian model with positional emission probabilities, is used to call preliminary CNVs. This process incorporates threshold and quality filters such as log2 ratio, z-scores, and phred score to ensure accurate variant detection.

## 3.4 CN Learn

One notable advancement in the field of CNV detection is the development of CN Learn [24], a machine-learning-based approach designed to accurately detect CNVs from exome sequencing data . Traditional methods for CNV detection have faced challenges in balancing sensitivity and specificity, often leading to false positives or missed CNVs, particularly in the context of exome sequencing where data complexity and variability are significant.

CN Learn addresses these challenges by leveraging a supervised machine-learning framework that integrates various features derived from exome sequencing data. The model is trained on a curated dataset of known CNVs, enabling it to learn patterns associated with true CNVs and distinguish them from noise or artifacts. The key innovation of CN Learn lies in its ability to adaptively learn from data, improving detection accuracy by refining its predictions based on the underlying genomic context.

The article detailing CN Learn demonstrates its superior performance compared to traditional CNV detection methods. The approach not only enhances detection accuracy but also significantly reduces the rate of false positives, making it a valuable tool in the analysis of exome sequencing data for CNV detection. By integrating machine learning into the CNV detection pipeline, CN Learn represents a critical step forward in the field, offering a more reliable and efficient solution for researchers and clinicians working with exome sequencing data.

# CHAPTER 4
# Planning

To achieve the objectives, the following steps have been undertaken:

- **Identify samples analyzed with the panel SUDD147**: We began by identifying the 400 most recent samples analyzed using the SUDD147 panel from the UDMMP database. For each of these samples, we obtained the corresponding BAM files, which serve as the primary data source for our analyses.

- **Identify samples with bad quality**: To ensure the reliability of our model, we assessed the quality of the identified samples. Samples with poor quality, characterized by insufficient coverage or other technical deficiencies that impede accurate CNV detection, were excluded from further analysis. This step mitigates the risk of introducing bias and inaccuracies in downstream processes.

- **Insert in silico CNVs**: Given the low frequency of CNVs detected in our dataset (only 74 events across over 2000 samples since 2017), we augmented our data by inserting CNVs in silico. This augmentation creates a sufficiently large and diverse dataset necessary for training robust machine learning models.

- **Run CNV detection algorithm**: A computational pipeline was developed to automate the execution of multiple CNV detection algorithms across all samples. The pipeline processes the aligned BAM files, runs several CNV callers, and parses the resulting outputs, ensuring comprehensive detection across different methodologies.

- **Determine CNVs overlap between algorithms calls**: To avoid double counting CNV events, we merged concordant CNV predictions from different callers that overlap in the same genomic region, treating them as single events for downstream analyses.

- **Label CNVs calls**: Each CNV call was automatically labeled as either a true CNV (whether detected in silico or experimentally validated) or an artifact. This labeling step is critical to establish a reliable training dataset for the machine learning models.

- **Model construction**: With a curated and labeled dataset, we extracted relevant genomic features that could enhance the predictive power of the model. These features were selected based on their potential to provide meaningful distinctions between true CNVs and artifacts.

- **ML model selection**: We evaluated two machine learning algorithms—Random Forest and XGBoost—to identify the most suitable model for our objectives. The selection was based on a comparative analysis of their performance metrics, such as precision, recall, and F1-score.
- **Feature selection**: To enhance model efficiency and performance, a feature selection process was conducted. Features that did not contribute significantly to the model's predictive power were removed, minimizing overfitting and computational complexity.
- **Decision threshold adjustment**: Given the critical importance of recall in this project, we optimized the model by adjusting the decision threshold. This adjustment aims to maximize recall while maintaining acceptable levels of precision, ensuring the model meets user requirements.
- **Model evaluation**: We conducted a detailed evaluation of the selected model to understand how the different variables impact its performance.
- **Model validation**: The constructed model, Targeted-CNV-Learner, was rigorously validated using two datasets: a testing set containing in silico CNVs and an experimentally validated set of real samples with known CNVs. This comprehensive validation ensures that the model performs robustly in both simulated and real-world scenarios. Furthermore, we benchmarked Targeted-CNV-Learner against several state-of-the-art CNV detection algorithms, using the same datasets. The benchmarking demonstrated that Targeted-CNV-Learner offers competitive, if not superior, performance, particularly in terms of recall, which is critical for this study's objectives.

CHAPTER 5

# Methodology

To achieve the defined objectives, a comprehensive methodology has been established, encompassing all necessary steps from data acquisition to model validation. This methodology is illustrated in Figure 5.1. Central to this approach is the development of the Targeted-CNV-Learner framework, which is accessible via this GitHub link.The repository includes a detailed user manual that outlines the steps for running the framework. This framework has been meticulously designed to streamline the creation of CNV detection models for targeted sequencing. By accommodating the unique characteristics of each panel, Targeted-CNV-Learner enhances the precision and reliability of CNV detection, ensuring that the models generated are optimized for the specific demands of different panels.

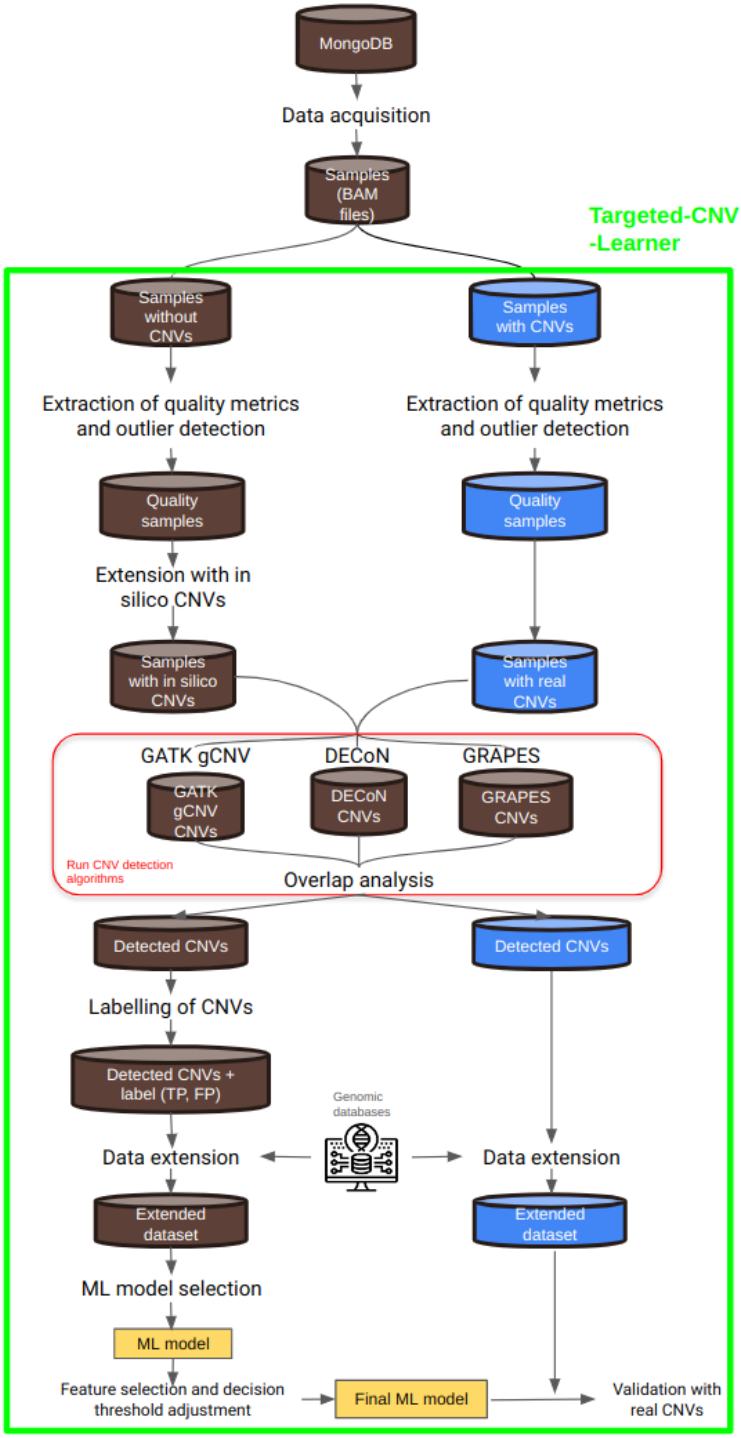



Figure 5.1: Overview of the methodology applied in this study, detailing the steps described in the methodology, from data acquisition to the final machine learning model generation and validation within the Targeted-CNV-Learner framework.

## 5.1 Data acquisition

For optimal performance of the Targeted-CNV-Learner, a dataset of at least 200 samples is recommended, as this aligns with the computational requirements of CNV detection algorithms. For instance, the GATK gCNV algorithm requires a minimum of 100 samples to establish a reliable in its initial analysis phase (baseline).

The primary input files required by Targeted-CNV-Learner include BAM files, which provide the aligned sequencing data, and a BED file that defines the genomic coordinates corresponding to the selected genetic panel. In addition, Targeted-CNV-Learner requires the identification of samples with CNVs that have been previously validated through orthogonal methods (see Section 2.2.4.

Samples with confirmed CNVs (Table 5.1) are used as a validation set to evaluate the model's performance, while samples without known CNVs are utilized to simulate in silico CNVs, which are subsequently employed to train the machine learning model.

Table 5.1: Some of the CNVs that have been detected and validated using MLPA by our group alongside information about the pathology of the patient and the gene.

| LAB ID | PATHOLOGY | GENE | EXON | TYPE |
|---|---|---|---|---|
| RB19783 | SIDS | LDB3 | ALL | DUP |
| RB20322 | Rasopathy | MYH6-MYH7 | 25-28 | DUP |
| RB20616 | SUD (no moscat) | CTNNA3 | 11 | DEL |
| RB20934 | HCM | MYPN | 1-2 | DUP |
| RB21220 | ARVC/D | MYH11 | 1-41 | DUP |
| RB21296 | TAAD | ANK2 | ALL | DUP |
| RB21276 | TAAD / AF | RAF1 | ALL | DUP |
| RB21824 | DCM | RBM20 | 2-5 | DEL |
| RB21874 | DCM | MYH11 | 1-41 | DUP |
| RB22220 | DCM | DSP | 9-24 | DEL |

## 5.2 Quality assessment and outlier detection

The aim of this step is to filter out samples that do not fulfill a given quality.

Read depth profiles are a crucial tool for assessing the quality of sequencing data [9]. They provide a snapshot of the coverage across different regions of the genome, offering insights into the uniformity and adequacy of the sequencing process. In the context of targeted sequencing panels, read depth profiles are particularly useful for identifying samples with poor quality as:

- **Uniform coverage**: High-quality sequencing data should exhibit uniform coverage across the targeted regions. Significant deviations in read depth can indicate issues

such as poor sample quality, sequencing errors, or library preparation problems. Uniform read depths ensures that each region of interest is adequately covered, which is essential for reliable CNV detection.

- **Detection of anomalies**: Samples with poor quality often show anomalies in their read depth profiles, such as regions with excessively high or low coverage. These anomalies can lead to false-positive of false-negative variant calls, thereby compromising the accuracy of downstream analyses.

To systematically assess the quality of samples using read depth profiles, the following methodology is employed

### 5.2.1 Calculation of Mean Read Depth

For each sample, the mean read depth is calculated for each exon in the targeted panel using the software Mosdepth [21]. This involves summing the read depths for each position within an exon and then averaging these values (Figure 5.2). The resulting mean read depths for all exons constitute the read depth profile of the sample.

To ensure comparability between samples, the read depth profiles are normalized and standardized by the mean read depth of the sample. This step adjusts for variations in overall sequencing depth and other systematic biases, enabling a fair comparison across different samples.

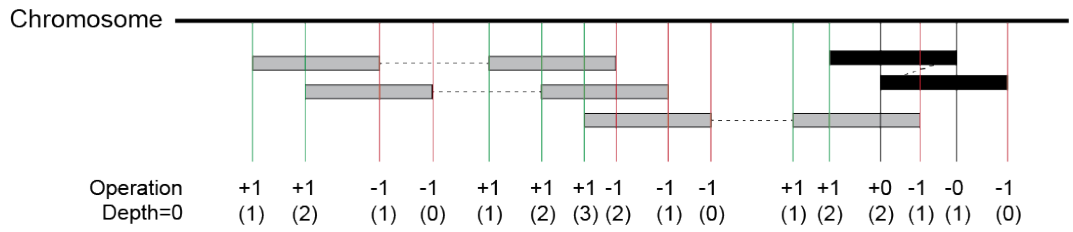



Figure 5.2: Illustration of Mosdepth's method for calculating read depth at each position. For every read start position, the value in the corresponding position of the array is incremented. Conversely, for every read stop position, the value in that position is decremented. This process accurately reflects the read depth distribution across the targeted region.

### 5.2.2 Principal component analysis

PCA is a statistical technique used to reduce the dimensionality of large datasets while preserving most of the variance [16]. By applying PCA to read depth profiles, we can transform the high-dimensional data into smaller set of principal components. These components capture the most significant patterns of variation in the data.

The first two principal components obtained from PCA can be visualized in scatter plots, where each point represents a sample. Samples with similar read depth profiles cluster together, while outliers, indicative of poor-quality samples, appear separated from the main

cluster. Based on the PCA results, we use the z-score method to identify the outliers. The z-score for the first and second principal components (PC1 and PC2) is calculated to standardize these values, enabling comparison on a common scale. The z-score for PC1 is computed by subtracting the mean of PC1 values from each PC1 value and then dividing the result by the standard deviation of the PC1 values. Similarly, the z-score for PC2 is obtained by subtracting the mean of PC2 values from each PC2 value and then dividing the result by the standard deviation of the PC2 values. Mathematically, these are expressed as:

$$\text{PC1\_zscore} = \frac{\text{PC1} - \overline{\text{PC1}}}{\sigma_{\text{PC1}}}$$

$$\text{PC2\_zscore} = \frac{\text{PC2} - \overline{\text{PC2}}}{\sigma_{\text{PC2}}}$$

where PC1 and PC2 represent the principal component values, $\overline{\text{PC1}}$ and $\overline{\text{PC2}}$ denote the mean values, and $\sigma_{\text{PC1}}$ and $\sigma_{\text{PC2}}$ signify the standard deviations of the principal component values.

Outliers are identified based on a predefined threshold, denoted as the z_score_threshold. A sample is considered an outlier if the absolute value of its z-score for either PC1 or PC2 exceeds this threshold. Specifically, a data point is flagged as an outlier if:

$$|\text{PC1\_zscore}| > \text{z\_score\_threshold} \quad \text{or} \quad |\text{PC2\_zscore}| > \text{z\_score\_threshold}$$

The z_score_threshold can be determined by the user using the z_score_threshold flag. Samples flagged as outliers are considered to fall outside the acceptable range of variation and are flagged as poor quality. These samples are then excluded from further analysis to prevent bias and ensure the integrity of the dataset.

## 5.3 Insertion of in silico CNVs

In order to develop a robust machine learning model capable of distinguishing between true positive CNV calls and false negative calls from different algorithms, it is essential to have a sufficiently large sample size. This ensures the model can learn from a diverse and representative set of examples, which is critical for accurate classification and generalization to new data.

However, obtaining a sufficiently large sample size for training the model is challenging due to the low recurrence of CNVs in targeted sequencing, where only a subset of genes is analyzed. CNVs are relatively rare events, especially in focused gene panels, where the likelihood of encountering a single CNV in a sample is low. This rarity limits the availability of naturally occurring CNVs in our dataset, thereby constraining the number of positive examples available for model training.

To address this limitation, we employ in silico insertion of CNVs into the BAM files. In silico CNVs are artificially generated variants that are computationally inserted into existing BAM files. This technique allows us to augment our dataset with additional CNV examples without the need for new sequencing efforts or the reprocessing of raw data. By creating these synthetic CNVs, we can significantly increase the number of positive samples in our dataset.

The inclusion of in silico CNVs is crucial for several reasons:

- **Enhanced training data**: By artificially increasing the number of CNV examples, we provide the machine learning model with more instances to learn from, improving its ability to recognize true CNVs and differentiate them from false positives
- **Balanced dataset**: Machine learning models perform better when trained on balanced datasets. The low natural occurrence of CNVs results in an imbalanced dataset with a high proportion of negative examples. In silico CNVs help balance the dataset, leading to more effective training.
- **Comprehensive learning**: In silico CNVs can be designed to cover a wide range of scenarios, including different types of CNVs, varying sized, and different genomic contexts. This diversity ensures that the model learns to detect CNVs under various conditions, enhancing its robustness and accuracy.

It is wothy to observe that both, in-silico CNVs and real CNVs that have been previously confirmed in the samples are required. The combination of real and in silico CNVs provides a comprehensive dataset for training and validation, ultimately leading to a more reliable and effective machine learning model for CNV detection.

To implement the in silico insertion of CNVs, we utilized a specialized tool that allows for precise and efficient modification of BAM files. This tool, known as Spike in BAM, facilitates the seamless integration of synthetic CNVs, ensuring that our dataset is both comprehensive and reflective of various CNV scenarios.

### 5.3.1 Spike in BAM

The insertion of in silico CNVs is a common technique in CNV analysis studies [4] [23]. This approach is widely adopted to augment datasets with synthetic CNVs, enabling the development and validation of CNV detection algorithms. Various algorithms have been developed to facilitate the insertion of in silico CNVs into NGS data [26] [37]. For this project, we have chosen to use Spike in BAM. Spike in BAM allows the insertion of CNVs directly into BAM files with the use of a simple configuration file (Figure 5.3). This configuration file must include the BAM file path, the start and end positions of the CNV, the type of CNV (duplication or deletion), and the exons encompassed by the CNV.

One of the key advantages of Spike in BAM is its ability to utilize multiple cores, significantly improving execution time. Additionally, Spike in BAM modifies the variant allele

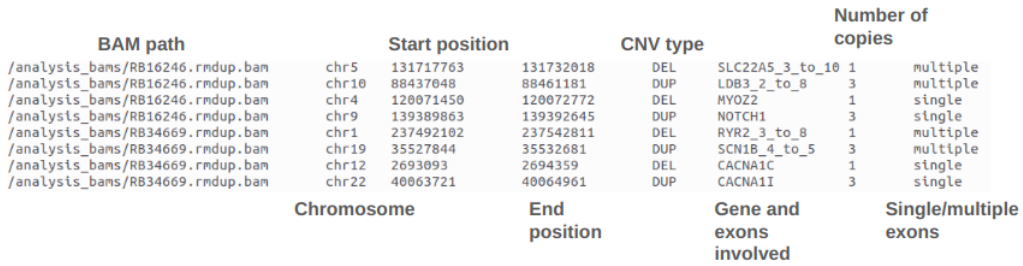



Figure 5.3: Example of the configuration file needed to insert CNVs using Spike in BAM. The configuration file should include the BAM file path, the start and end of the CNV, the type of CNV (duplication or deletion) and the exons that englobes the CNV.

frequencies of any SNPs that overlap with the simulated CNV, providing a more realistic representation of CNV insertion. This feature ensures that the synthetic CNVs closely resemble naturally occurring variants, enhancing the robustness of the dataset for training and validating CNV detection models.

To automate the insertion of CNVs into BAM files, the Targeted CNV Learner generates a configuration file from a BED file, which specifies the genomic regions of interest. This autogenerated configuration file enables the subsequent execution of Spike in BAM to create in silico CNVs within the BAM files, while maintaining a detailed record of all inserted CNVs.

Targeted-CNV-Learner automatizes the insertion of four CNVs in each sample:

- 1 single exon duplication
- 1 multiple exon duplication
- 1 single exon deletion
- 1 multiple exon deletion

## 5.4 Run CNV detection algorithms

The subsequent phase of the project involves the automated execution of three CNV detection algorithms across the samples to identify CNVs. This step is essential not only for comparing the performance of different CNV detection algorithms but also for ensuring comprehensive and reliable CNV detection across the dataset.

To achieve this, we have developed a pipeline module in Targeted-CNV-Learner that automatically runs the following CNV detection algorithms:

1. GRAPES
2. GATK g-CNV
3. DECON

The Targeted-CNV-Learner pipeline integrates these algorithms into a seamless workflow, ensuring efficient and reproducible CNV detection. The key features of the pipeline include:

- **Automated execution**: The pipeline automates the execution of GRAPES, GATK g-CNV, and DECON, minimizing manual intervention and reducing the potential for human error.
- **Data management**: It efficiently manages the input data, ensuring that the corresponding samples are processed together.
- **Result parsing**: The pipeline automatically parses the CNV calls from each algorithm, facilitating the integration of the results for downstream analyses.

By automating the entire process, the Targeted-CNV-Learner pipeline not only saves time but also ensures consistency and accuracy in CNV detection across different algorithms. The generation of VCF files by each algorithm, followed by automated parsing, allows for an efficient and reliable compilation of CNV data, streamlining downstream analyses and enhancing the overall robustness of the detection process.

### 5.4.1 GRAPES and DECON

Both GRAPES and DECON are designed to operate on each sequencing run, meaning they analyze all samples that were prepared and sequenced together as a batch. This batch-wise analysis is critical for mitigating technical biases introduced during sample preparation and sequencing. By processing the samples collectively, these algorithms ensure more consistent and reliable CNV detection across samples prepared under similar conditions.

### 5.4.2 GATK g-CNV

In contrast to the batch-wise approach of GRAPES and DECON, GATK g-CNV employs a cohort-based strategy. This method involves several key steps:

1. **Cohort Mode**: GATK g-CNV uses a cohort of 100 samples to establish a baseline read depth profile. These samples are selected to represent the typical read depth distribution across the targeted regions.
2. **Baseline Creation**: The read depth profile generated from the cohort serves as a reference, capturing expected read depth variations under normal conditions and accounting for systematic biases and variability inherent to the sequencing process.
3. **Comparison and Detection**: Each sample to be analyzed is then compared against this baseline read depth profile. By evaluating deviations from the baseline, GATK g-CNV can detect CNVs with greater precision. This approach enhances the accuracy

of CNV detection by reducing false positives and ensuring that true CNVs are more reliably identified.

This cohort-based approach allows GATK g-CNV to enhance the accuracy of CNV detection, providing a powerful complement to the batch-wise methods used by GRAPES and DECON. By leveraging both approaches, our pipeline ensures a robust and comprehensive detection of CNVs across a wide range of conditions.

## 5.5 Detected CNV overlap analysis

Once we have obtained the CNV calls from all the algorithms, along with the record of the inserted in silico CNVs, it is expected that the detected CNVs does not have identical coordinates across different algorithms, neither they perfectly match the inserted in silico CNVs. This variation arises from differences in the sensitivity and specificity of the algorithms, as well as the inherent variability in the CNV calling process. Therefore, it is essential to have a robust method to determine when CNVs identified by different algorithms—or when compared to *in silico* CNVs—should be considered equivalent based on their overlapping regions.

In this study, we implemented a method within the Targeted-CNV-Learner pipeline that deems two CNVs equivalent if they share a minimum percentage of overlap.

To illustrate this, consider the following CNVs:

| | Chromosome | Start | End | CNV Type | Algorithm |
|---|---|---|---|---|---|
| CNV call 1 | 2 | 1233 | 2000 | Duplication | DECoN |
| CNV call 2 | 2 | 1200 | 1633 | Duplication | GRAPES |
| In silico CNV | 2 | 1320 | 2100 | Duplication | Real CNV |

Table 5.2: CNVs for overlap demonstration. CNV call 1 and CNV call 2 are identified by different algorithms, and the *in silico* CNV is an artificially inserted CNV for validation.

These CNVs are considered equivalent if there is sufficient overlap between them, as determined by a predefined percentage threshold. This threshold is adjustable by the user through a specific flag in the Targeted-CNV-Learner software.

In our case study, we selected a 20% overlap threshold. The rationale behind this choice is that, while precise positioning of CNVs is important, our primary focus is on the detection of CNVs rather than their exact boundaries. Given that CNVs are typically validated using orthogonal techniques, we prioritize the presence or absence of a CNV over its precise coordinates.

### 5.5.1 Overlap Calculation Methodology

To determine whether CNV call 1 and CNV call 2 (Table 5.2) can be considered the same, we perform the following calculations:

**Overlap region calculation**:

$$\text{Overlap} = \min(\text{end}_1, \text{end}_2) - \max(\text{start}_1, \text{start}_2) = 1633 - 1233 = 400$$

**Total length of CNV1:**

$$\text{Length}_{\text{CNV1}} = 2000 - 1233 = 767$$

**Total length of CNV2:**

$$\text{Length}_{\text{CNV2}} = 1633 - 1200 = 433$$

**Percentage overlap for CNV1:**

$$\text{Overlap}_{\text{CNV1}} = \left(\frac{400}{767}\right) \times 100 = 52.14\%$$

**Percentage overlap for CNV2:**

$$\text{Overlap}_{\text{CNV2}} = \left(\frac{400}{433}\right) \times 100 = 92.38\%$$

Since both percentages exceed the 20% threshold, CNV1 and CNV2 are considered the same CNV. The coordinates for the resulting CNV are determined by taking the furthest boundaries, resulting in:

| | Chromosome | Start | End | CNV Type | Algorithm |
|---|---|---|---|---|---|
| Overlapped CNV | 2 | 1200 | 2000 | Duplication | DECoN and GRAPES |

Table 5.3: CNV resulting from the overlapping analysis.

This approach allows us to consider CNVs that overlap in the same region as the same CNV, while excluding those that have significantly different lengths.

## 5.6 Labelling CNVs

After determining the overlapping CNV calls from various algorithms, the next step is to label these overlaps based on their overlapping with the *in silico* CNVs. This labeling process enables the precise categorization of CNVs as either true positives or false negatives.

Using the methodology described in Section 5.5 and the example provided in Table 5.4, we demonstrate how CNVs are labeled:

To determine whether the overlapped CNV corresponds to the *in silico* CNV, we calculate the overlap between the two:

**Overlap region calculation**:

$$\text{Overlap} = \min(\text{end}_1, \text{end}_2) - \max(\text{start}_1, \text{start}_2) = 2000 - 1320 = 680$$

| | Chromosome | Start | End | CNV Type | Algorithm |
|---|---|---|---|---|---|
| Overlapped CNV | 2 | 1200 | 2000 | Duplication | DECoN and GRAPES |
| In silico CNV | 2 | 1320 | 2100 | Duplication | Real CNV |

Table 5.4: Comparision of final CNV against in silico CNV.

**Total length of Final CNV:**

$$\text{Length}_{\text{Overlapped CNV}} = 2000 - 1200 = 800$$

**Total length of in silico CNV:**

$$\text{Length}_{\text{in silico CNV}} = 2100 - 1320 = 780$$

**Percentage overlap for Overlapped CNV:**

$$\text{Overlap}_{\text{Overlapped CNV}} = \left(\frac{680}{800}\right) \times 100 = 85\%$$

**Percentage overlap for in silico CNV:**

$$\text{Overlap}_{\text{in silico CNV}} = \left(\frac{680}{780}\right) \times 100 = 87,18\%$$

Since both percentages exceed the 20% threshold, the in silico CNV is considered to be detected by the overlapped CNV.

This approach allows us to classify each overlapped algorithm call as a true positive (if an in silico CNV overlaps with the call) or a false positive (if an in silico CNV does not overlap with the call). In this example, the overlapped CNV was detected by both DECoN and GRAPES, but not by GATK gCNV. Since it sufficiently overlaps with the *in silico* CNV, it is classified as a true positive, obtaining a CNV call with the following labels:

| | Chromosome | Start | End | Grapes | DECoN | GATK gCNV | CNV Type | Overlap in silico |
|---|---|---|---|---|---|---|---|---|
| Overlapped CNV | 2 | 1200 | 2000 | True | True | False | Duplication | True |

Table 5.5: Labels assigned to the Overlapped CNV containing if the CNV was detected by the different algorithm and if it overlaps a inserted in silico CNV.

## 5.7 Data model construction

The development of a robust predictive model within the Targeted-CNV-Learner framework necessitates the careful selection, extraction, and integration of features from a variety of sources. This process is essential for ensuring that the model can effectively differentiate between true CNVs and artifacts, thereby enhancing its predictive performance and reliability.

| Category | Feature | Description | Origin |
|---|---|---|---|
| CNV quality | decon | DECoN call overlap with CNV | algorithm-derived |
| | gatk | GATK gCNV call overlap with CNV | algorithm-derived |
| | grapes | Grapes call overlap with CNV | algorithm-derived |
| | decon_qual | Quality score assigned by DECoN | algorithm-derived |
| | gatk_qual | Quality score assigned by GATK gCNV | algorithm-derived |
| | grapes_qual | Quality score assigned by Grapes | algorithm-derived |
| Genomic features | chr | Chromosome where CNV is located | algorithm-derived |
| | type | Type of CNV (duplication or deletion) | algorithm-derived |
| | numb_exons | Number of exons involved in the CNV | internally computed |
| | cnv_length | Length of the CNV | internally computed |
| | gene | Gene(s) involved in the CNV | internally computed |
| | gc_content | GC content in the CNV region | genomic repositories |
| | mappability | Mappability score of the CNV region | genomic repositories |
| Quality Samples | sample_correlation | Correlation of sample with cohort | internally computed |
| Target | true_positive | Indicates if the CNV is a true positive | internally computed |

Table 5.6: Detailed categorization of features used in the predictive model for CNV detection.

To achieve this, the data model incorporates features from three primary sources. First, externally sourced features are gathered from **genomic databases and repositories**, providing crucial genomic context to help the model understand the broader genomic landscape surrounding each CNV.

Second, **algorithm-derived** features come directly from the CNV detection algorithms. These features include key quality metrics that indicate the confidence level of each CNV call. They also provide crucial information about the genomic position (such as the chromosome) and the type of the CNV. Furthermore, these features specify whether each CNV was detected by the respective algorithms, helping to identify areas of agreement or disagreement between different methods.

Lastly, **internally computed** features are generated within the Targeted-CNV-Learner software itself. These include metrics that are calculated based on the existing CNV data, such as sample correlation with the cohort, and specific measurements like CNV length. These internally derived features capture additional patterns that may not be immediately apparent from external data or algorithm outputs.

By integrating features from these three distinct sources, Targeted-CNV-Learner constructs a comprehensive and diverse dataset. This enriched data model enables the algorithm to more effectively distinguish between true positive and false positive CNV calls, thereby improving its overall reliability and generalizability across different datasets.

The selected features are categorized into three groups: CNV Quality Variables, CNV genomic features, and Quality Sample Variables. These features collectively form the final data model, which is summarized in Table 5.6.

### 5.7.1 CNV quality variables

Each CNV detection algorithm provides a quality score for its calls, based on its internal calculations. These quality scores are informative about the reliability of the CNV calls and are included as features in the model.

- **DECoN Call (decon)**: Indicates if a DECoN call overlaps the CNV.
- **GATK gCNV Call (gatk)**: Indicates if a GATK gCNV call overlaps the CNV.
- **Grapes Call (grapes)**: Indicates if a Grapes call overlaps the CNV.
- **DECoN Quality (decon_qual)**: DECoN calculates the quality score of the CNV call using a Bayesian framework, where it compares the likelihood of observed read depth data under the hypothesise of a CNV's presence versus its absence. The resulting Bayes factor, indicates the confidence level in the CNV detection and it is only available if DECoN made a call overlapping the CNV.
- **GATK gCNV Quality (gatk_qual)**: GATK gCNV calculates the quality score of a CNV by using a Hidden Markov Model to estimate the likelihood of different copy number states based on observed read depth data. It then computes a log-likelihood ratio comparing the most likely CNV state to the normal state, with higher log-likelihood ratios indicating a more confident CNV call. This variable is only available if GATK gCNV made a call overlapping the CNV.
- **Grapes Quality (grapes_qual)**: Grapes uses a HMM with position-specific emission probabilities based on a Gaussian distribution. It employs the Viterbi algorithm to estimate the most likely sequence of states (e.g., homozygous loss, heterozygous loss, diploid, one gain, two gains). The quality score for each CNV is derived from the mean probability of the exons constituting the CNV region. This quality score is available only if Grapes has made a call overlapping the CNV.

### 5.7.2 Context variables

Contextual variables provide critical information regarding the genomic environment in which CNVs are located. These features help in identifying and mitigating the biases and technical artifacts often associated with CNV detection.

- **Chromosome (chr)**: The chromosome on which the CNV is located.
- **CNV type(type)**: The type of CNV (duplication or deletion) is provided to the model as an essential feature.
- **Exon number in gene (num_exons)**: False-positive CNV calls tend to accumulate in the first exons of genes due to their properties (e.g., higher GC content) and technical

factors related to sequencing (e.g., mapping artifacts, capture efficiency, read start position bias). Including the exon number in the gene as a feature helps the model account for these tendencies.

- **GC content (gc_content)**: GC content is a well documented bias that significantly affects NGS data (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3378858/). Regions with high or low GC content can exhibit differential sequencing efficiency and read depth, leading to false-positive CNV calls. By calculating the GC content for each CNV, we can help the model account for these biases, potentially increasing the reliability of CNV detection.

- **Mappability (mappability)**: Low mappability regions can cause ambiguous read alignment and reduced accuracy in CNV detection. By incorporating mappability scores (https://pubmed.ncbi.nlm.nih.gov/22276185/), the model can better identify areas of the genome where sequencing data may be less reliable, thus improving its ability to distinguish between genuine CNVs and artifacts.

- **CNV length (cnv_length)**: CNV detection algorithms often struggle with short CNVs (those spanning a single exon or a small number of exons). Including the length of the CNV as a feature can help the model improve its accuracy.

- **Gene (gene)**: Certain genes are more prone to CNV calls due to their inherent properties. Providing the gene information as a parameter can help the model improve its predictive metrics by accounting for gene-specific biases.

The start and end positions of CNVs, typically included in CNV annotation data (see Table 5.5), are not considered in the final data model. This is because these positions are implicitly accounted for through other variables, such as the CNV genomic context variables features (e.g., chromosome, CNV length and gene). Additionally, the chromosome information is separately provided as a context variable, ensuring that positional data contributes meaningfully to the model without redundancy.

### 5.7.3 Quality samples variables

There is a single quality sample variable: Sample correlation over cohort samples (sample_correlation). In our analysis, it is crucial to assess the quality of the sample and the sequencing process to reliably detect CNVs. One way to achieve this is by calculating the correlation of the normalized read depth profiles of the analysis sample with those of a cohort of samples. This correlation helps identify the degree of similarity between the analysis sample and the cohort, providing insights into the reliability of detected CNVs. The steps followed to obtain this calculation are the following ones:

The following steps outline the process of calculating the sample correlation:

- **Normalization of Read Depth**: For each sample $i$ in both the cohort and the analysis sample, the read depth for each exon $j$ is normalized. This is done by dividing the read depth of the exon $\text{RD}_{ij}$ by the mean read depth of the corresponding sample $\overline{\text{RD}_i}$:

$$\text{NRD}_{ij} = \frac{\text{RD}_{ij}}{\overline{\text{RD}_i}}$$

  where $\text{NRD}_{ij}$ is the normalized read depth for exon $j$ in sample $i$.

- **Computation of Correlations**:The Spearman correlation coefficient $\rho$ is calculated between the normalized exon coverage profile of the analysis sample $\mathbf{X}$ and each sample in the cohort $\mathbf{Y}_k$. The Spearman correlation is chosen for its robustness in handling non-linear relationships and varying distributions:

$$\rho(\mathbf{X}, \mathbf{Y}_k) = \frac{\text{cov}(\text{rank}(\mathbf{X}), \text{rank}(\mathbf{Y}_k))}{\sigma_{\text{rank}(\mathbf{X})}\sigma_{\text{rank}(\mathbf{Y}_k)}}$$

  where rank($\mathbf{X}$) and rank($\mathbf{Y}_k$) are the ranks of the normalized read depth values of the analysis sample and cohort sample $k$ respectively, cov is the covariance, and $\sigma$ is the standard deviation.

- **Mean Correlation Value**: The mean of the correlation values is computed:

$$\overline{\rho} = \frac{1}{N}\sum_{k=1}^{N}\rho(\mathbf{X}, \mathbf{Y}_k)$$

  where $N$ is the number of cohort samples. This mean correlation $\overline{\rho}$ provides a single metric that reflects the overall similarity between the analysis sample and the cohort. A higher mean correlation indicates a higher degree of similarity and thus greater reliability in CNV detection.

### 5.7.4 Target variable

The target variable for prediction is: **True Positive CNV Call (true_positive)**: Indicates whether the CNV overlaps an in-silico CNV, thereby determining if it is a True Positive call (True) or not (False).

## 5.8 ML model selection

The selection of an optimal model with the best parameters is crucial to ensure that we are leveraging the most effective approach for achieving superior results. In our dataset, various variables may contain null values, particularly when the quality of a CNV detection

algorithm is not applicable if the CNV was not detected by that specific algorithm. Consequently, it is imperative to select a model capable of effectively handling such missing data.

Two leading machine learning approaches that excel in managing null values are Random Forest and XGBoost. Both methods are based on decision trees, which naturally offer several advantages in dealing with incomplete data.

Random Forest constructs an ensemble of decision trees during training, where each tree contributes either the mode of the classes (for classification tasks) or the mean prediction (for regression tasks) as the final output. This method gracefully handles null values by building trees using only the available data and making decisions based on subsets of features without the need for imputation.

Extreme Gradient Boosting (XGBoost), on the other hand, iteratively builds a series of decision trees, with each new tree focusing on correcting the errors made by its predecessors. During training, XGBoost learns how to navigate missing values and strategically uses them to split nodes, ensuring robust model performance even with incomplete data.

In the Targeted-CNV-Learner framework, both Random Forest and XGBoost models are constructed using the available dataset. Model metrics are generated for both approaches, allowing the user to compare their performance. The user can then select the preferred model by setting a specific flag, ensuring flexibility and adaptability in choosing the most suitable algorithm for their specific needs.

## 5.9 Feature selection

Once we have established that which is the best model for our dataset, we employ feature selection to enhance the performance and interpretability of our model. It is crucial to identify and retain only the most relevant features to improve model efficiency and accuracy **??**.

Tree-based models, such as XGBoost and random forest, inherently provide a measure of feature importance. This measure reflects the contribution of each feature to the reduction of impurity in the model's decision trees. By leveraging these feature importance scores, we can rank all features and eliminate the less significant ones, thereby reducing dimensionality and potentially improving model performance. We use feature selection based on feature importance to reduce the number of features included in the model without compromising its feasibility, following the process represented in Figure 5.4.

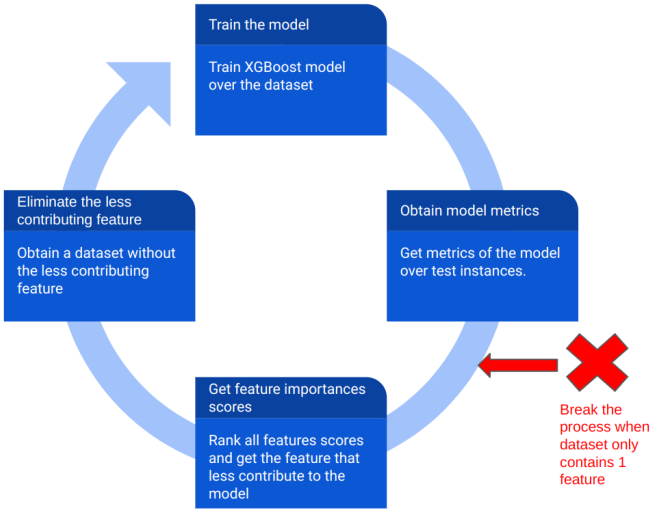



Figure 5.4: Feature selection process illustrating the impact of removing the least contributing feature on the model.

By implementing this approach, we aim to achieve a more streamlined and interpretable model without compromising its predictive power. This process help us strike a balance between model simplicity and performance, ensuring that the final model remains both effective and manageable.

## 5.10 Decision threshold adjustment

In clinical genomics, particularly in the detection of CNVs using NGS, the identification of true positives is critical. CNVs often require further validation using orthogonal techniques, making it imperative that our model emphasizes high recall.

Prioritizing recall is essential in this context, as missed CNVs—false negatives—can have significant implications for patient care. The clinical consequences of failing to identify a true CNV underscore the necessity of fine-tuning the model's decision threshold to optimize sensitivity.

To optimize the balance between recall and precision, we adjust the decision threshold of our predictive model. Typically, models classify outputs based on a default threshold, such as 0.5 in binary classification tasks. However, by varying this threshold, we can influence the trade-off between different performance metrics, particularly recall and precision.

We systematically evaluate the model's performance at various thresholds and select the threshold that best satisfies our requirements. Specifically, we focus on thresholds that maximize recall while maintaining acceptable levels of precision. This approach ensures that we capture the majority of true CNVs, minimizing the risk of overlooking clinically significant variants.

Final model metrics be computed for each threshold, and the threshold yielding the optimal balance between recall and other performance metrics is chosen. This threshold

adjustment is crucial for tailoring the model to the specific needs of clinical CNV detection, where sensitivity is often prioritized.

## 5.11 Validation with real CNV samples

While in silico data is invaluable for training and initial testing, the true measure of a predictive model's effectiveness lies in its performance on real-world data. To ensure that our model accurately detects clinically relevant CNVs, it is essential to validate it using real CNV samples. This step is critical for confirming that the model's predictions translate effectively into practical, clinical scenarios.

For this reason, these CNVs that have been rigorously validated through orthogonal methods, are an ideal benchmark for assessing the model's accuracy. The samples containing these CNVs serve as our validation set, providing a robust means to evaluate the model's true positive rate in a real-world context.

By testing the model on this validation set, we gain insights into its performance on actual data, ensuring that it can reliably detect CNVs that have already been confirmed through experimental validation. This validation process is crucial not only for assessing the model's predictive power but also for establishing confidence in its clinical applicability.

## 5.12 Experimental set-up

This section presents the rationale behind the design of our experimental setup, which is tailored to develop a predictive model for CNV detection with high generalization ability and clinical relevance. The setup involves distinct phases: model selection, feature reduction, decision threshold optimization, and final validation with real CNV samples. The overall workflow is illustrated in Figure 5.5.

The decision to reserve 80% of the in silico CNV data for model selection is driven by the need to thoroughly explore the performance of different model architectures (e.g., XGBoost and Random Forest) across a substantial portion of the data. To ensure robust performance, a 10-fold cross-validation (CV) approach is utilized. The dataset is divided into 10 subsets, where in each iteration, the model is trained on 9 subsets and validated on the remaining one. This process is repeated across all subsets, ensuring that each data point is used for both training and validation. This approach minimizes the risk of overfitting to any specific subset of the data, leading to a model that is more likely to generalize well to new, unseen data.

This rigorous cross-validation allows for an unbiased estimation of model performance and aids in selecting the most suitable model architecture—whether XGBoost or Random Forest—based on the mean following metrics:

- **Accuracy**: The ratio of correctly classified instances to the total number of instances.

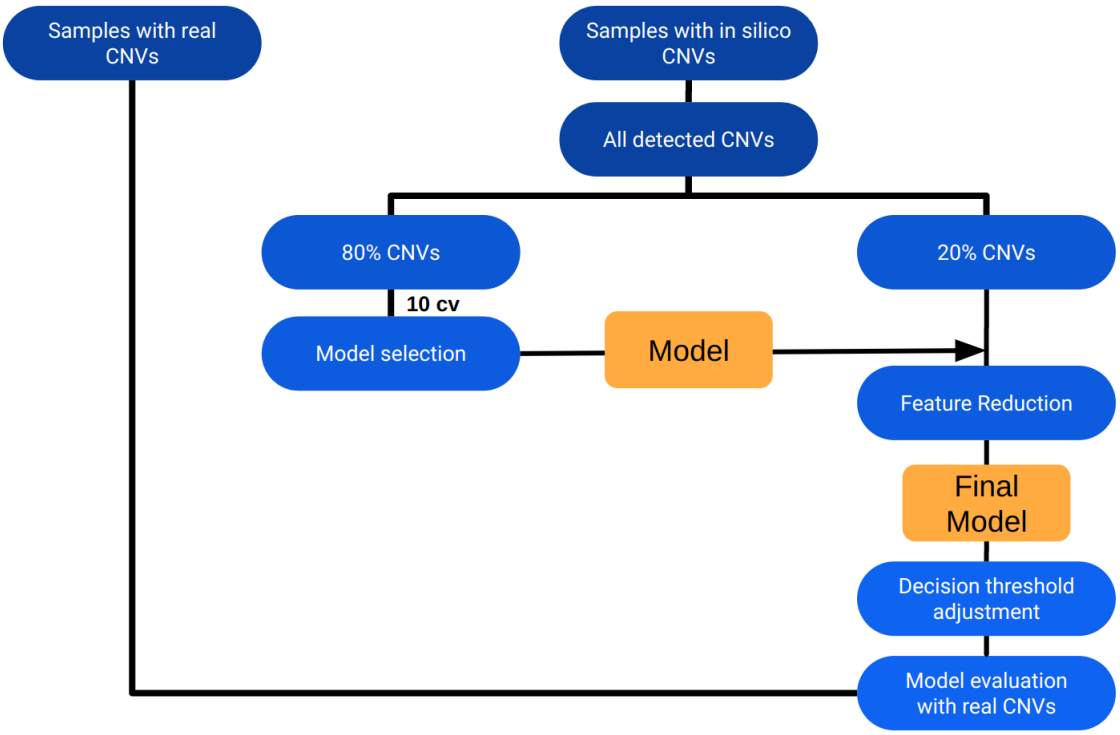



Figure 5.5: Experimental design for model construction and model evaluation.

- **Selectivity (Specificity)**: The proportion of true negatives correctly identified by the model.
- **Sensitivity (Recall)**: The proportion of true positives correctly identified by the model.
- **F1 Score**:The harmonic mean of precision and recall, balancing these two critical aspects of model performance.

These mean metrics, illustrated in Figure 5.6, are key to assessing the generalization ability of the models across different data splits.

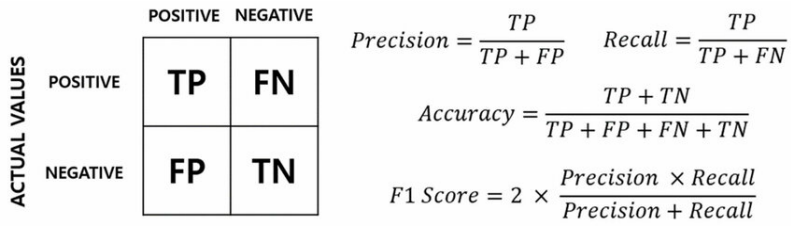



Figure 5.6: Model metrics used to evaluate the model.

The remaining 20% of the in silico data is purposefully reserved for two critical tasks: feature impact analysis and decision threshold optimization. This split allows us to isolate the processes of feature selection and threshold setting from the initial model training,

ensuring that these optimizations are not biased by the same data used for model selection. By using this separate subset, we can fine-tune the model's complexity and decision-making criteria, which is essential for maximizing both its interpretability and predictive power.

Finally, the performance of the model evaluated using real CNV samples. This validation step with real data is essential to confirm the model's clinical applicability and to assess its effectiveness in detecting CNVs in a real-world setting.

CHAPTER 6

# RESULTS AND DISCUSSION

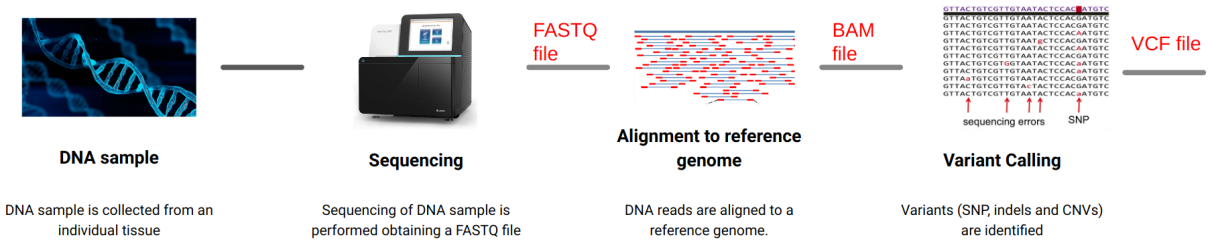
This chapter presents the results of our methodology applied over the SUDD147 samples. First we provide the results regarding the outlier analysis. Next, the results on the existing CNVs detecting models are detailed. Afterwards, an exploration data analysis of all the varibles gathered in the methodology is provided. Finally, the results of the model selection, feature selection, final model and model evaluation with in silico and real CNVs are provided.

## 6.1 Dataset

Since the UDMMP group began sequencing, approximately 30,000 variants have been identified and stored in a MongoDB database. This database serves as a central repository where all samples and their associated variants analyzed over the years are systematically cataloged. Specifically, the samples analyzed using the SUDD147 panel have also been indexed in this MongoDB, providing a comprehensive record of the sequencing data generated by UDMMP.

The overall process for variant detection, from sequencing to analysis, is illustrated in Figure 6.1. Due to the high memory consumption associated with storing intermediate files, UDMMP follows a policy of deleting BAM files and other temporary analysis outputs one year after their creation. To move forward with this project, we opted to retrieve BAM files from the MongoDB database for samples analyzed within the past year using the SUDD147 panel. This allowed us to access an adequate dataset, comprising approximately 400 samples, without the need to reprocess the raw FASTQ files. By leveraging the pre-processed BAM files, we avoided the significant computational costs associated with re-aligning the sequencing data and regenerating the BAM files.

Figure 6.1: Simplified workflow used for variant calling bioinformatic analysis with all the intermediate files created shown in red.



## 6.2 Identification of samples with bad quality

Samples of poor quality, whether due to the quality of the extracted tissue or issues arising from the NGS protocol, exhibit anomalies in their read depth profiles that can hinder the detection of CNVs. Such anomalies bias the results, making it crucial to identify and remove these samples from the analysis.

To address this, we employed PCA to reduce the dimensionality of the read depth data for each exon, as detailed in Section 5.2. PCA allows us to capture the most significant patterns of variation in the data, thereby facilitating the identification of outliers.

Figure 6.2 shows the outlier detection performed on SUDD147 samples using a z_score_threshold of 2. In this PCA plot, each point represents a sample, with colors indicating whether a sample is classified as an outlier or not. The majority of samples cluster tightly together, indicating uniform and high-quality sequencing data. However, a number of samples are dispersed across the plot, representing the poor-quality samples.

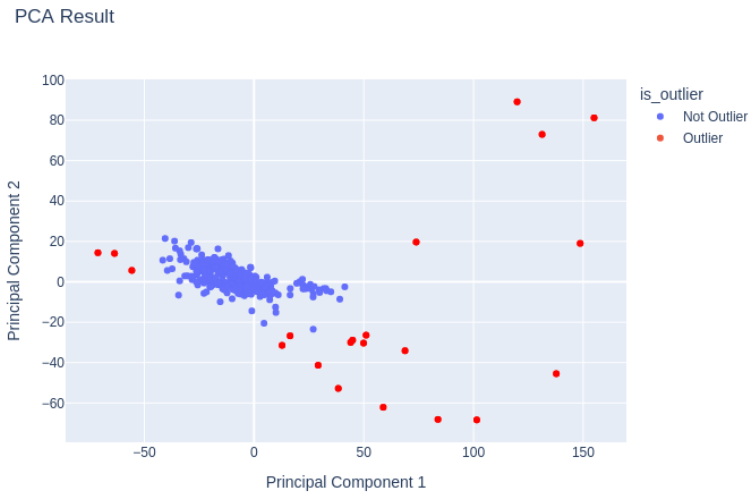



Figure 6.2: PCA plot illustrating the clustering of high-quality samples and the dispersion of poor-quality samples.

By cross-referencing these outlier samples with the quality metrics previously obtained by the group, we confirmed that the identified poor-quality samples were correctly detected. This validation step is crucial, as it ensures that the removal of these outliers is justified, thereby preserving the integrity and accuracy of the CNV detection process.

The clustering of high-quality samples and the dispersion of outliers highlight the effectiveness of the PCA and z-score method in distinguishing between samples of varying quality. Removing these outliers from further analysis minimizes bias and enhances the reliability of the results.

## 6.3 Evaluation of stand-alone CNV detection methods

This section presents an in-depth evaluation of the three CNV detection algorithms utilized in this project: GATK, DECoN, and Grapes. Our analysis emphasizes the performance of these algorithms in detecting CNVs across varying exon counts, with particular focus on their effectiveness and accuracy in identifying CNVs with smaller exon sizes.

A total of 1161 in silico CNVs were introduced for evaluation. Figure 6.3 summarizes the performance metrics for the stand-alone CNV detection algorithms:

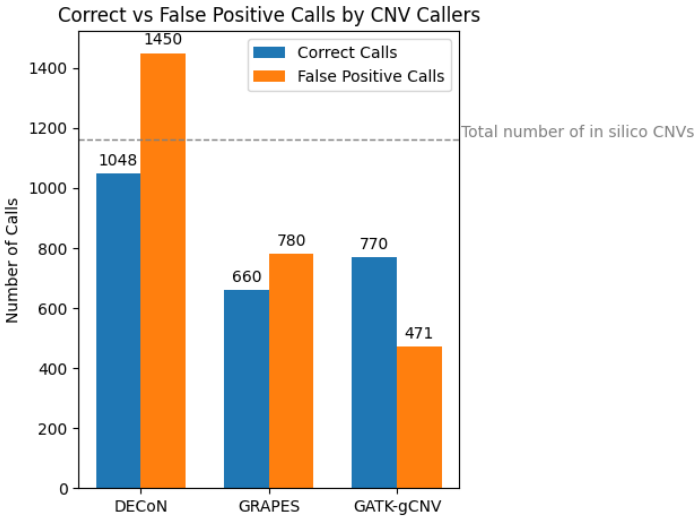



Figure 6.3: Illustrates the comprehensive workflow of genomic DNA targeted sequencing.

- **DECoN**: DECoN detected the highest number of CNVs, identifying 1048 out of 1161. However, it also generated a significant number of false positives, totaling 1450. Consequently, DECoN achieved a precision of 0.4195 and a FDR of 0.5804.

- **GRAPES**: GRAPES correctly identified 660 CNVs and produced 780 false positives, resulting in a precision of 0.4583 and an FDR of false discovery rate (0.5417.

- **GATK gCNV**: GATK gCNV identified 770 CNVs with 471 false positives, achieving the highest precision of 0.6205 and the lowest FDR of 0.3795 among the evaluated algorithms.

Collectively, these algorithms detected a total of 1128 CNVs, leaving 33 CNVs undetected. These undetected CNVs highlight the inherent challenges in current CNV detection methodologies.

A common challenge among all CNV detection algorithms is their reduced effectiveness in detecting CNVs with a small number of exons [18]. Figure 6.4 illustrates the precision and recall of each algorithm across varying exon counts, highlighting their performance in detecting CNVs with smaller exon sizes. All three algorithms show diminished precision and recall at lower exon counts, indicating that CNVs with fewer exons are more challenging to detect accurately. This leads to either missed detections (lower recall) or incorrect predictions (lower precision).

DECoN stands out with the highest recall, indicating its strong capability to detect a wide range of CNVs. However, this high recall comes at the expense of precision, resulting in a higher rate of false positives. This trade-off between recall and precision is critical, especially in applications where the cost of false positives must be carefully balanced against

the necessity to detect as many true CNVs as possible. GATK and Grapes maintain high precision but with a slight compromise in recall, particularly at extreme exon sizes. DECoN, conversely, offers higher recall but at a notable drop in precision.

By combining the outputs of these three algorithms into a machine learning model, incorporating genomic characteristics of the CNV call region, we anticipate improving the overall ability to accurately detect CNVs. This approach aims to enhance the FDR without compromising the detection capability, ultimately providing a more robust and reliable CNV detection framework.

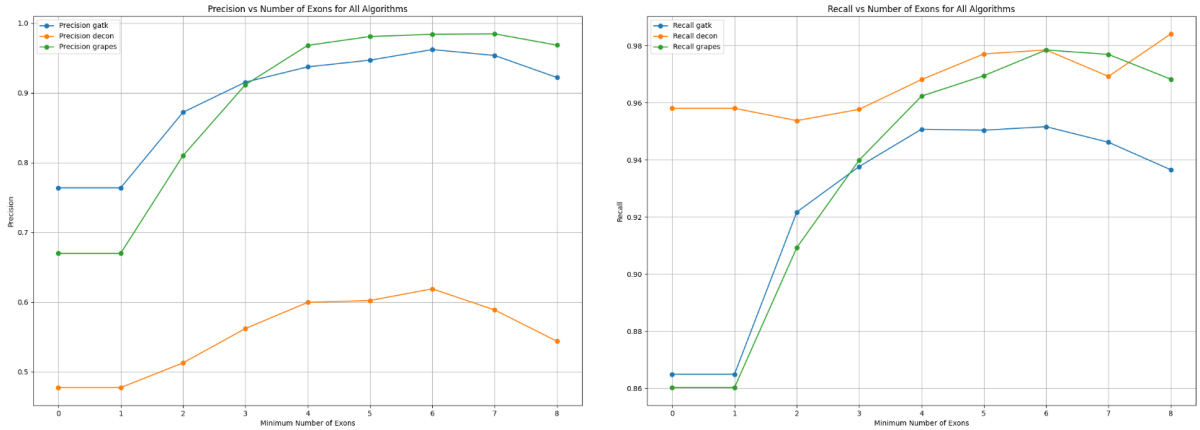



Figure 6.4: Precision and recall of the different algorithms plotted against the minimum number of exons a CNV has. For example, data points for *Minimum Number of Exons* = 1 include all CNV calls that have 1 or more exons. The figures are truncated at *Minimum Number of Exons* = 8 as, out of the 2992 total CNV calls made by the algorithms, 2888 calls involve CNVs with 8 or fewer exons. This truncation ensures a focused and clear analysis of the majority of CNV calls.

## 6.4 Model variables analysis

The feature variables play a vital role in machine learning algorithms. They provide the necessary information for the algorithm to learn and make predictions. The quality and relevance of the feature variables greatly impact the accuracy and performance of the model. By choosing the right feature variables, we can improve the predictive power of our machine learning models.

To enhance the efficiency of our model for detecting CNVs, we analyze how these different variables influence model performance to achieve a model that optimally balances accuracy and simplicity.

Variables fall into three main categories: sample quality, CNV quality, and CNV genomic featu. Each provides crucial information to the model, aiding in the accurate prediction of whether a CNV call is a True Positive or a False Positive.

In the following analysis, we explore the relationships between these variables and their impact on the predictive accuracy of the model.

### 6.4.1 CNV quality variables

The CNV quality variables encompass all metrics related to CNV calls and their respective accuracies, including DECoN call, GATK gCNV call, Grapes call, DECoN quality, GATK gCNV quality, and Grapes quality.

The performance of individual CNV detection algorithms has been assessed in Section 6.3. Here, we evaluate whether overlapping calls from different algorithms enhance the confidence in CNV detection.

Figure 6.5 presents a Venn diagram showing the overlap of CNV calls made by DECoN, GATK, and Grapes. The central region, where all three algorithms concur, exhibits the highest number of true positive calls (837) and a relatively low number of false positive calls (54). This indicates a high level of confidence in CNV calls identified by all three algorithms.

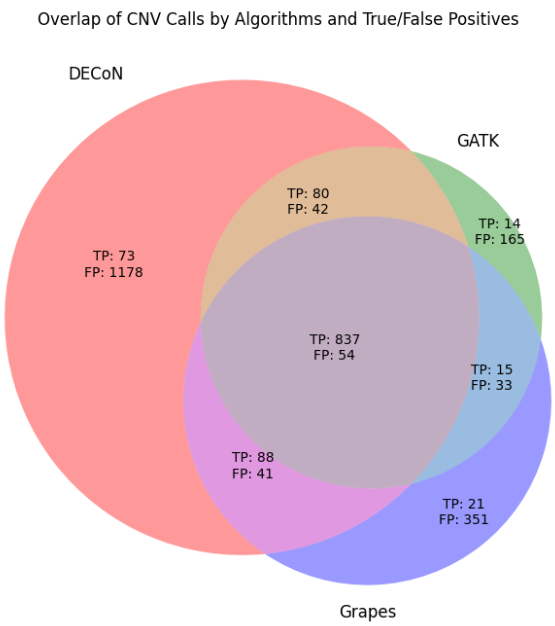



Figure 6.5: Venn diagram illustrating the overlap between calls made by the CNV detection algorithms: DECoN, GATK, and Grapes. Each section of the diagram is annotated with the number of true positive (TP) and false positive (FP) CNV calls. The diagram visualizes the intersections of the algorithms' calls and highlights both unique and overlapping CNV identifications.

Examining the regions where two algorithms agree, there is a noticeable increase in the false discovery rate compared to the region of complete overlap. Notably, combinations involving DECoN with either GATK or Grapes demonstrate better performance compared to the combination of GATK and Grapes. Calls made by single algorithms, despite identifying some true positives, generally exhibit a high false positive rate, suggesting lower reliability in these cases.

The results suggest that the overlap of algorithm calls is a critical factor in determining the reliability of CNV detection. Consequently, these overlapping calls are expected to significantly influence the machine learning model that we are going to develop, given their demonstrated high reliability when all three algorithms agree.

Next, we examine how the quality scores of the algorithm calls relate to the likelihood of a CNV being a true positive or a false positive. Figure 6.6 shows the distribution of CNV call quality scores for each algorithm against the predicted variable (whether the CNV is a real CNV or an artifact). It is evident that for each algorithm, false positive CNVs tend to accumulate at lower quality scores, whereas true positive CNVs are more prevalent at higher quality scores. This observation suggests that the quality score is a valuable feature for the model to distinguish between true positive and false positive CNVs.

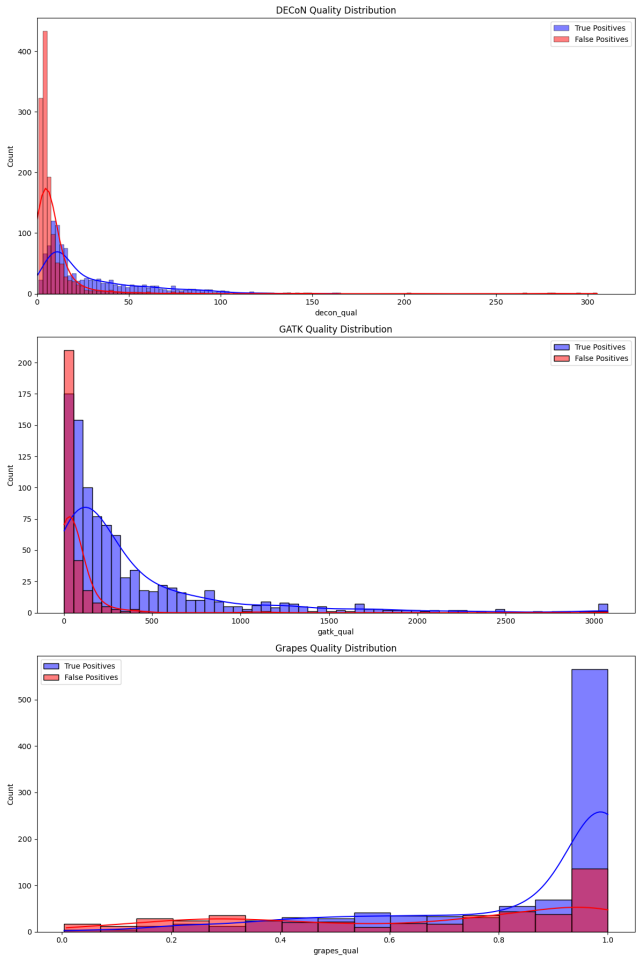



Figure 6.6: Distribution of CNV quality scores for DECoN, GATK, and Grapes algorithms against the whether if the CNV is a True Positive or an artifact. True positive CNVs are shown in blue, while false positive CNVs are shown in red. The quality scores are indicative of the confidence in the CNV call, with higher scores correlating with true positive CNVs.

These findings underscore the importance of quality scores in improving the accuracy of CNV detection. By incorporating these quality variables into our machine learning model, we can enhance its ability to correctly classify CNVs, thereby reducing the false discovery rate.

### 6.4.2 CNV genomic feature variables

The genomic featu of CNVs is described by several variables, including chromosome, CNV type, number of exons included in the CNV, GC content, mappability, CNV length, and gene association. We now examine the distribution of true positive and false positive CNVs across these variables.

#### 6.4.2.1 Genes

Figure 6.7 illustrates the top 10 genes with the highest number of false positive CNVs alongside the top 10 genes with the highest number of true positive CNVs. The random insertion of in silico CNVs often leads to a higher accumulation of true positive CNVs in larger genes. However, some genes also show a tendency to accumulate false positive calls, making them significant variables to consider in our model.

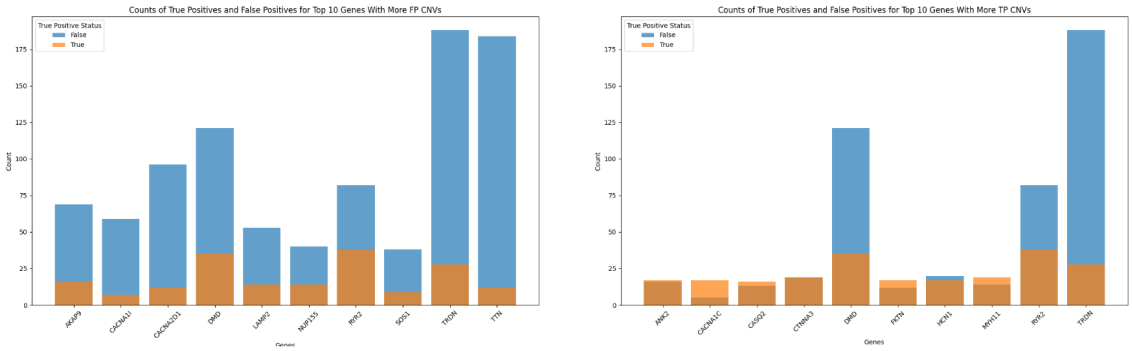



Figure 6.7: The figure on the left displays the top 10 genes with the highest number of false positive CNVs, along with the true positive CNVs found in these genes. The figure on the right shows the top 10 genes with the highest number of true positive CNVs, along with the false positive CNVs identified in these genes.

#### 6.4.2.2 Chromosome

Figure 6.8 illustrates the chromosomes with higher accumulations of false positive CNVs. Notably, chromosomes 2, 6, 7, and X exhibit elevated frequencies of false CNV calls. This pattern is partly attributed to the presence of genes on these chromosomes that are prone to generating false positives. For instance, chromosome 2 contains genes such as TTN and SOS1, chromosome 6 includes TRDN, and chromosome 7 features CACNA2D1 and AKAP9, all of which are associated with higher false positive rates.

Chromosome X, a sex chromosome with differing copy numbers between males (one copy) and females (two copies), introduces additional challenges in CNV detection. The variable copy number can complicate accurate CNV assessment, leading to a higher incidence of false positive calls.

To better address these issues, we propose incorporating a new categorical variable to distinguish between autosomal and sex chromosomes (chrX and chrY). This addition helps in understanding and mitigating the specific challenges associated with CNV detection on sex chromosomes, thereby improving the overall accuracy of CNV identification.

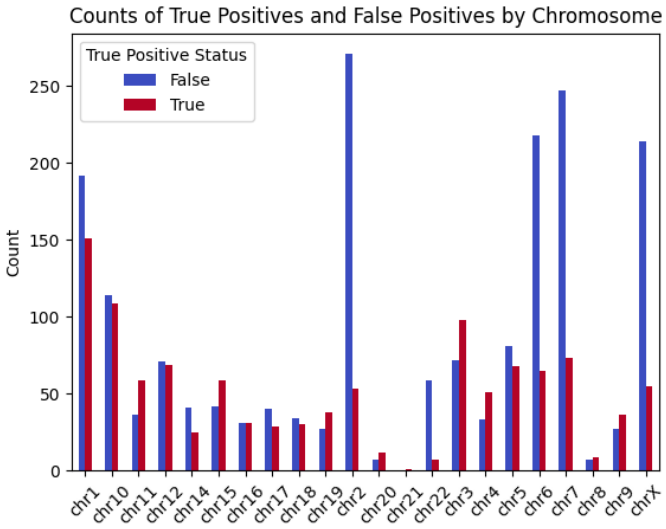



Figure 6.8: Distribution of CNVs across chromosomes, highlighting the chromosomes with higher accumulations of false positives.

#### 6.4.2.3 CNV type

Previous studies have suggested that duplications are generally more challenging to detect compared to deletions. For instance, some references highlight the difficulties associated with detecting duplications in CNV detection algorithms due to subtler signal changes and increased background noise [8] [18].

However, as illustrated in Figure 6.9, our analysis reveals that the number of false positive CNV calls is approximately equivalent for both duplications and deletions. In addition, the number of True positive CNV calls is consistent with our experimental design, where we introduced two duplications and two deletions per sample in silico. Consequently, the algorithms detected these CNVs in roughly equal proportions.

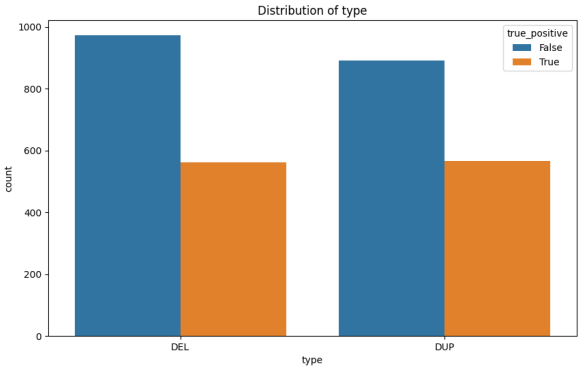



Figure 6.9: Distribution of CNV types in the dataset, illustrating the prevalence of deletions versus duplications among false positive and true positive calls.

Given these findings, it appears that CNV type (duplication versus deletion) does not significantly influence the accuracy of detection in the context of our dataset. Therefore, for the model constructed using the SUDD147 panel, it may be reasonable to exclude the CNV type variable from consideration, as its impact on the detection accuracy does not appear to be substantial.

#### 6.4.2.4 Number of exons and CNV length

The number of exons and CNV length are two key variables that measure the size of the CNV, albeit in different ways. These variables are highly correlated, with a Pearson correlation coefficient of 0.738. This high correlation suggests that they may convey similar information.

However, as shown in Figure 6.10, the distribution of true positive and false positive calls against these two variables appears to be quite similar. This similarity indicates that these variables might not provide significant discriminatory power for the model, as they do not distinguish well between true positive and false positive CNV calls.

Given this observation, we conduct a further analysis to determine the impact of these variables on the model's performance. If it is found that these variables do not enhance the predictive accuracy of the model, we may consider removing them to simplify the final model.

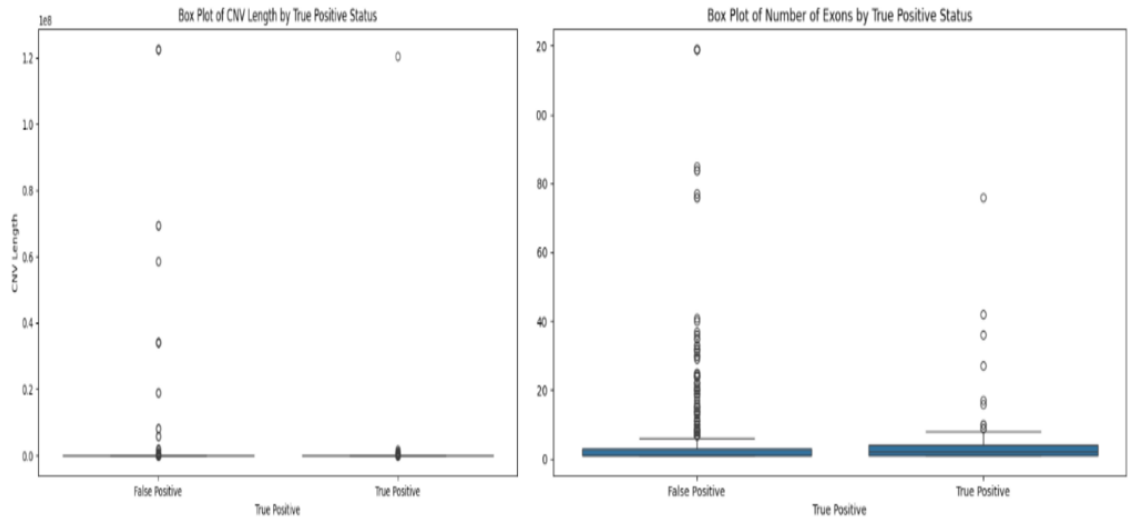



Figure 6.10: Boxplots representing the distribution of true positive calls and false positive calls against the variables: number of exons and cnv length.

#### 6.4.2.5 GC Content

Regions of the genome where the GC content significantly deviates from 50% tend to experience more misalignments, leading to a less uniform distribution of reads. As a result,

these regions are prone to higher rates of false positive CNV calls. This phenomenon is due to the challenges in sequencing and alignment accuracy in regions with extreme GC content, which complicates the detection of true CNVs.

In Figure6.11, we analyze the distribution of GC content for true positive and false positive CNV calls. The histogram and corresponding density plots illustrate that false positive CNV calls are more frequent in regions where the GC content is either much lower or much higher than 50%. Conversely, true positive CNV calls tend to cluster around the 50% GC content mark. Therefore, accounting for GC content in the model could help reduce the rate of false positives and improve overall detection accuracy.

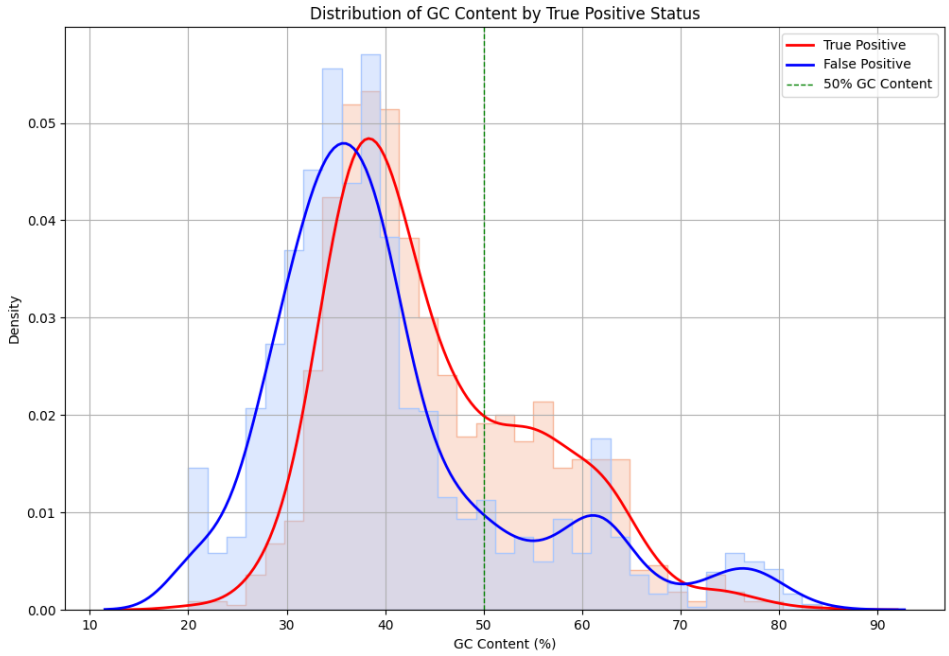



Figure 6.11: GC content distribution differences between true positive and false positive CNV calls. The vertical dashed line indicates the 50% GC content mark.

#### 6.4.2.6 Mappability

Mappability refers to the ability to uniquely map sequencing reads to a reference genome, which can impact the detection and accuracy of copy number variations (CNVs). To evaluate whether mappability plays a significant role in distinguishing between true positive and false positive CNV calls, we analyzed the distribution of these calls across varying mappability values.

As illustrated in Figure 6.12, the density distributions of true positive and false positive CNVs relative to the mappability values of their corresponding genomic regions are highly

similar. Both types of CNV calls exhibit a significant peak at a mappability value of 1.0, indicating regions with perfect mappability. However, there are no substantial differences in the distributions at lower mappability values.

This similarity suggests that mappability does not provide a distinguishing factor between true positive and false positive CNVs. Therefore, incorporating mappability into our model does not enhance its ability to accurately differentiate between genuine CNVs and artifacts. Consequently, mappability can likely be excluded as a predictive variable in our model without sacrificing accuracy.

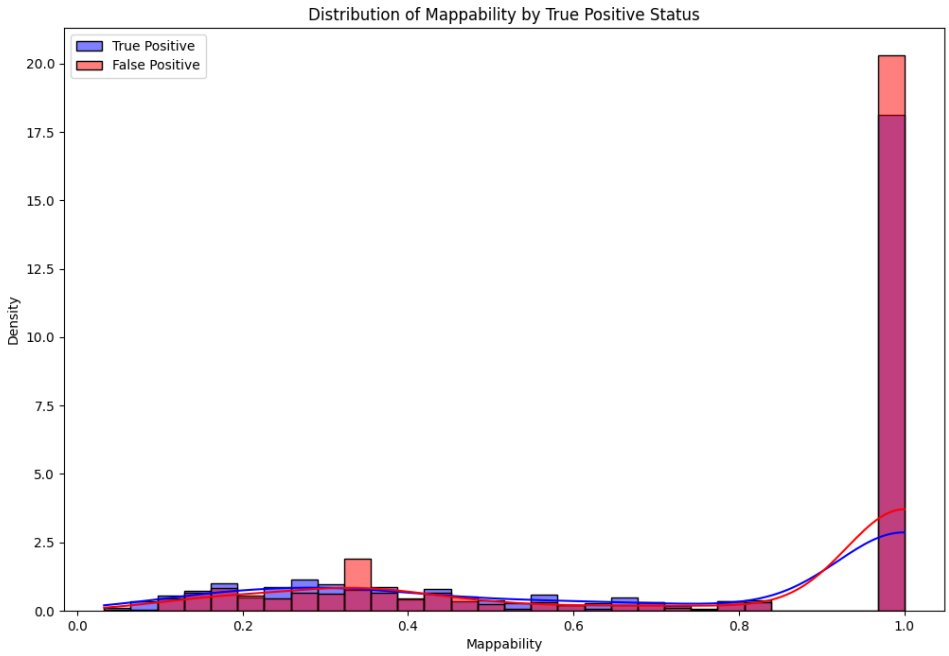



Figure 6.12: Distribution of mappability values for true positive and false positive CNVs. The density plots indicate that both true positives and false positives are similarly distributed across mappability values, with a notable peak at a mappability value of 1.

### 6.4.3 Quality samples variable

In this group, we are only using one variable that is describing the quality of the sample and it is the sample correlation.

We analyze how the correlation values affect the model's ability to differentiate between true positive and false positive CNV detections.

To visualize this, we generated a density plot, as shown in Figure 6.13. The density plot displays the distribution of correlation values for true positive and false positive CNVs. We observe that the correlation values of the samples range between 0.70 and 0.91. We should

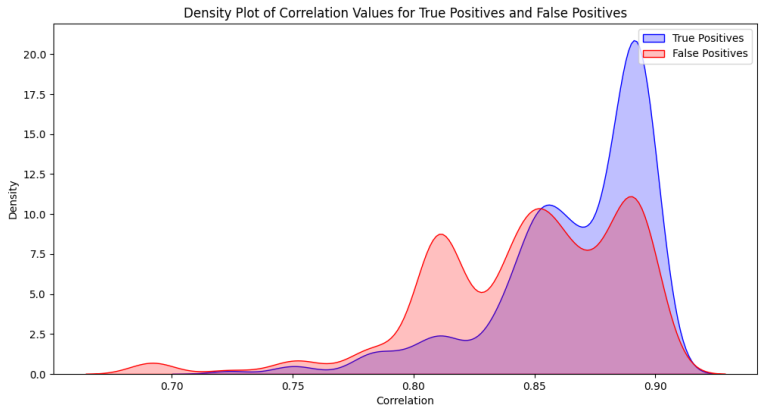



Figure 6.13: Density plot showing how the correlation of samples affects the number of true positive and false positive CNVs.

take into account that four CNVs were introduced in each sample, so a higher number of True positive CNVs in the distribution is an indicative of a higher number of samples between these correlation range. By comparing the distribution of CNVs between true positive and false positive detections, we note that at lower correlation values (<0.85), there is a tendency to obtain more false positive CNVs than true positive CNVs. However, at higher correlation values, this trend reverses, resulting in a higher number of true positive CNVs compared to false positive CNVs.

This observation demonstrates that lower quality samples tend to have a higher rate of false positive CNVs, whereas higher quality samples are associated with a lower rate of false positive CNVs.

## 6.5 Model Selection

To ensure the best model is selected, we performed a rigorous evaluation using both random forest and XGBoost by first finding the best hyperparameters for both models. Then, to mitigate the risk of overfitting and ensure that our model generalizes well to unseen data, we employed k-fold cross-validation. This technique divides the data into k subsets, in this case 10, and trains the model k times, each time using a different subset as the validation set while the remaining k-1 subsets are used for training.

As presented in Table 6.1 the evaluation metrics indicate that XGBoost consistently outperforms Random Forest across all considered metrics, including accuracy, precision, recall, and F1 score.

Based on these findings, XGBoost is selected as the primary machine learning algorithm for this project. Its ability to deliver higher predictive accuracy and its overall better performance metrics make it the most suitable choice for our objectives.

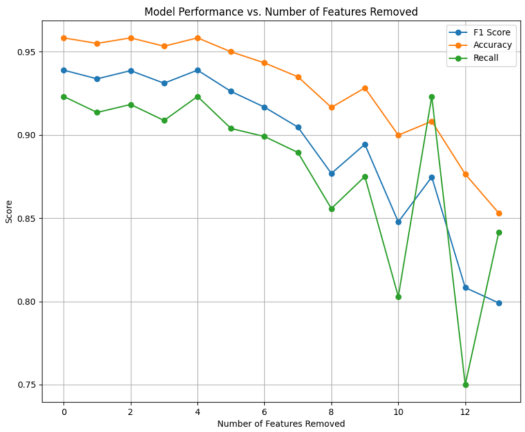
| Model | Accuracy | Specificity | Recall | F1 Score |
|---|---|---|---|---|
| Random Forest | 0.9515 | 0.9693 | 0.9182 | 0.9294 |
| XGBoost | 0.955 | 0.9744 | 0.9183 | 0.9363 |

Table 6.1: Metrics to evaluate Random Forest and XGBoost model performance.

## 6.6 Feature Selection

First, we train an XGBoost model on the entire dataset to obtain an initial set of feature importance scores and model metrics. The feature importance scores indicate the relative importance of each feature in predicting the target variable. Features are then ranked based on these importance scores, and the feature that contributes the least to the model is eliminated. This process is iterative: we repeatedly remove the least contributing feature, train a new model without this feature, and compare the model parameters to evaluate the impact of each feature removal on the model's performance.

In Figure 6.14, the model's metrics remain stable when the first four features are removed. From this point onward, the metrics start to decrease, indicating that the removal of these initial features (Chromosome, Mappability, Gene, and CNV type) does not significantly affect the model's predictive effectiveness. This suggests that these features can be excluded without compromising the model's performance.




| Iteration round | Feature eliminated | Importance score |
|---|---|---|
| 1 | Chromosome | 0.0117 |
| 2 | Mappability | 0.01 |
| 3 | Gene | 0.0118 |
| 4 | CNV type | 0.0142 |
| 5 | CNV length | 0.0219 |
| 6 | Number of exons | 0.0250 |
| 7 | GC content | 0.0238 |
| 8 | Sample correlation | 0.0354 |
| 9 | Grapes Q | 0.0244 |
| 10 | DECoN Q | 0.0244 |
| 11 | GATK Q | 0.0318 |
| 12 | DECoN | 0.1763 |
| 13 | Grapes | 0.3739 |

Figure 6.14: Model metrics with the associated features eliminated in each round to perform feature selection.

As previously observed, Mappability and CNV type do not provide substantial information to the model, as their distributions are similar between true positives and false positives. However, the features Gene and Chromosome exhibit different distributions when compared to the predicted feature, suggesting that their removal does not harm the model due to their generalized nature.

Chromosome and Gene are broad variables, and their different distributions likely stem from inherent properties that are already included in the model (GC content, CNV length,

number of exons) that do not contribute significantly to distinguishing between true positive and false positive CNVs. Consequently, their exclusion does not reduce the model's ability to make accurate predictions.

The final model includes the following variables: number of exons, CNV length, DECoN, Grapes, GATK, DECoN quality call, Grapes quality call, GATK quality call and sample correlation. The performance metrics, as depicted in Figure 6.15, demostrate the effectiveness of these features in predicting CNVs.

Confusion matrix over test set

| TARGET \ OUTPUT | TP CNV | FP CNV | SUM |
|---|---|---|---|
| TP CNV | 192<br>32.05% | 11<br>1.84% | 203<br>94.58%<br>5.42% |
| FP CNV | 16<br>2.67% | 380<br>63.44% | 396<br>95.96%<br>4.04% |
| SUM | 208<br>92.31%<br>7.69% | 391<br>97.19%<br>2.81% | 572 / 599<br>95.49%<br>4.51% |

| Recall | Precision | F1 Score | Accuracy | Specificity |
|---|---|---|---|---|
| 0.9275 | 0.9606 | 0.9489 | 0.9649 | 0.9795 |

Figure 6.15: Confussion matrix and metrics for the final model.

## 6.7 Decision Threshold Adjustment

In our specific context, recall is of paramount importance as explained in Section XXXXX.

To enhance recall, we adjusted the decision threshold for the XGBoost model. The standard classification threshold of 0.5 means that a CNV is classified as positive only if more than half of the trees in the ensemble agree. By lowering this threshold, we increase the sensitivity of the model, making it more likely to predict positive CNVs and thereby improving recall. However, this adjustment inevitably leads to an increase in false positives. These false positives are subsequently addressed through our orthogonal validation process, ensuring that the overall reliability of CNV detection remains high.

Figure 6.16 illustrates how different model metrics respond to varying decision thresholds for considering a CNV a true positive instance. As shown, the decision threshold increases the recall of the model, meaning it can identify a higher number of true CNV instances. However, this comes with a trade-off: the precision decreases as more non-CNV instances are misclassified as true CNVs.

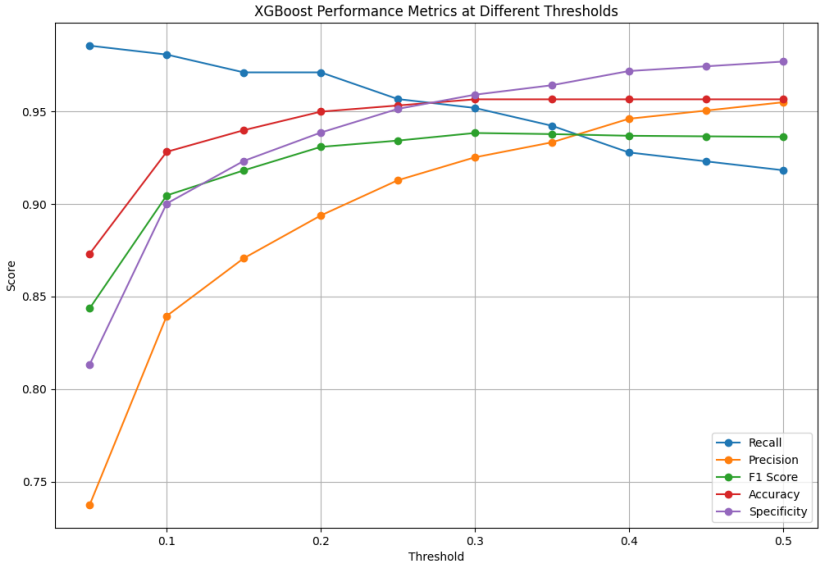



Figure 6.16: Impact of varying decision thresholds on XGBoost model metrics.

A reliable decision threshold could be between 0.2 and 0.3, where the model achieves a recall of 0.9519 to 0.9711 and a precision of 0.9252 to 0.8938, respectively. This range represents a balance between high sensitivity and acceptable precision, ensuring that almost all CNVs are detected while minimizing the economic and labor costs associated with validating CNVs through MLPA.

By selecting a threshold within this range, we prioritize the detection of true CNVs, which is crucial given the potential consequences of missing true variants. Although this approach increases the number of false positives, the subsequent orthogonal validation process filters out these inaccuracies, preserving the overall reliability and effectiveness of our CNV detection pipeline.

## 6.8 Validation of Real CNVs

The validation of the Targeted-CNV-Learner on real CNV samples represents a crucial phase in evaluating the clinical applicability and reliability of the model. As emphasized in Section 5.11, while in silico data forms the backbone of model training and initial testing, the true measure of any predictive model's effectiveness is its performance in a real-world clinical context. Real-world data, in this case, validated CNVs, serves as the gold standard for assessing the model's ability to identify clinically relevant CNVs with precision and accuracy, which is essential when these predictions have direct implications for patient management and care.

CNVs are relatively rare in patients suffering from sudden cardiac death (SCD), with

an estimated occurrence rate of approximately 1.4% [17]. Over the past eight years, the analysis of samples using the SUDD147 gene panel led to the identification of 79 samples harboring CNVs which can be observed in the following excel link. However, due to ongoing modifications in sequencing protocols during this time, not all samples met the required quality control standards. As described in Section 5.2, these quality control filters resulted in the exclusion of 38 samples that were flagged as outliers due to issues with read depth and overall sequencing quality. Consequently, the final validation dataset used for the Targeted-CNV-Learner comprised 41 experimentally validated CNVs.

Although the dataset is limited in size, it provides a representative sample of real CNVs that have been confirmed using orthogonal validation techniques, specifically MLPA. This selection, while modest, offers valuable insight into the model's performance in detecting real CNVs. The dataset allows us to critically assess the performance of the Targeted-CNV-Learner in comparison with stand-alone CNV detection algorithms such as GATK gCNV, DECoN, and GRAPEs.

Table 6.2 presents a detailed summary of the performance metrics for each CNV detection algorithm, including the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) for the Targeted-CNV-Learner and the stand-alone algorithms.

Table 6.2: Performance Metrics of Stand-Alone CNV Detection Algorithms and Targeted-CNV-Learner over the experimentally validated dataset.

| **Algorithm** | **TP** | **FP** | **TN** | **FN** |
|---|---|---|---|---|
| GATK gCNV | 40 | 47 | 205 | 1 |
| DECoN | 38 | 213 | 39 | 3 |
| GRAPEs | 40 | 31 | 221 | 1 |
| Targeted-CNV-Learner | 40 | 31 | 221 | 1 |

The performance of Targeted-CNV-Learner stands out by achieving a high level of sensitivity, correctly detecting 40 of the 41 real CNVs. This yields a sensitivity rate of approximately 97.56%, which aligns closely with that of GATK gCNV and GRAPEs, both of which also identified 40 CNVs. DECoN, on the other hand, detected only 38 CNVs, missing three true positives, which reflects a lower sensitivity.

Moreover, the false positive rates (FP) across the different algorithms reveal notable differences in specificity. DECoN exhibited the highest number of false positives. In contrast, GRAPEs and Targeted-CNV-Learner demonstrated a marked reduction in false positive calls, both generating 31 false positives. GATK gCNV, while successful in detecting 40 true CNVs, produced 47 false positives, indicating that it could be more prone to overcalling CNVs compared to GRAPEs and Targeted-CNV-Learner.

The results from the real-world validation thus demonstrate that the Targeted-CNV-Learner achieves a strong balance between sensitivity and specificity, making it a promising tool for clinical CNV detection. Its ability to minimize false positives while maintaining high sensitivity is a critical factor in ensuring that patients receive accurate diagnoses.

## 6.9 Discussion

CNVs represent a significant source of genetic diversity and are crucial for understanding the genetic basis of numerous diseases. Accurate detection of CNVs in targeted sequencing is, therefore, essential. However, existing methods often suffer from limitations in both sensitivity and specificity, which can result in false positives or missed detections. These inaccuracies ultimately impact the reliability of genetic analyses.

In this section, I present a comprehensive evaluation of the proposed method, Targeted-CNV-Learner, and benchmark its performance against three widely-used CNV detection algorithms: DECoN, GRAPES, and GATK gCNV.

To ensure a fair comparison of the different methods, the test set (20% of the entire dataset) used to previously evaluate Targeted-CNV-Learner is also employed to compare the performance of the standalone algorithms. The results, as shown in Table 6.3, highlight the differences in performance across these methods.

Table 6.3: Comparison of Performance Metrics for CNV Detection Algorithms

| Algorithm | TP | FP | TN | FN | Recall | Specificity | Precision | Accuracy |
|---|---|---|---|---|---|---|---|---|
| **DECoN** | 203 | 281 | 109 | 5 | 0.976 | 0.279 | 0.419 | 0.526 |
| **GRAPES** | 177 | 107 | 284 | 31 | 0.851 | 0.726 | 0.623 | 0.766 |
| **GATK gCNV** | 175 | 55 | 336 | 33 | 0.841 | 0.859 | 0.761 | 0.853 |
| **Targeted-CNV-Learner** | 198 | 16 | 375 | 10 | 0.952 | 0.959 | 0.925 | 0.955 |

The Targeted-CNV-Learner framework outperforms all standalone algorithms across most performance metrics. While DECoN achieves a slightly higher recall (0.976) compared to Targeted-CNV-Learner (0.952), this comes with a significantly higher number of false positive (FP) calls—281 for DECoN versus only 16 for Targeted-CNV-Learner. This large discrepancy in false positives is crucial because reducing false positives directly reduces the need for costly and logistically challenging orthogonal validation methods in clinical diagnostics.

Targeted-CNV-Learner maintains high sensitivity while significantly improving other key performance metrics:

- **Specificity**: Targeted-CNV-Learner achieves the highest specificity (0.959), demonstrating its ability to correctly identify true negatives and minimize false positives. This is substantially better than the specificity of DECoN (0.279), as well as higher than GRAPES (0.726) and GATK gCNV (0.859).

- **Precision**: With a precision of 0.925, Targeted-CNV-Learner has a high positive predictive value, meaning most of its CNV calls are true positives. This significantly reduces the number of false positives compared to all other methods. For example, GRAPES has a precision of 0.623, GATK gCNV has 0.761, and DECoN has only 0.419.

- **Accuracy**: Targeted-CNV-Learner achieves the highest overall accuracy (0.955), reflecting a balanced performance in detecting CNVs with both high sensitivity and

specificity. This accuracy is higher than that of GRAPES (0.766), GATK gCNV (0.853), and DECoN (0.526).

Overall, Targeted-CNV-Learner provides a more reliable and well-rounded approach to CNV detection, significantly reducing false positives while maintaining high sensitivity, thus outperforming the standalone algorithms in clinical diagnostics.

These results illustrate that the objectives of the project have been successfully met. The Targeted-CNV-Learner provides a more reliable approach for CNV detection by enhancing both precision and specificity compared to existing standalone algorithms. This improvement ensures that true CNVs are accurately identified while significantly reducing the number of false positives, ultimately minimizing the need for costly and time-consuming orthogonal validation methods such as MLPA. This is particularly valuable in clinical settings where the accuracy of CNV detection is crucial for patient diagnosis and treatment planning.

CHAPTER 7

# Drawbacks and Limitations

Despite the promising performance of the Targeted-CNV-Learner, several limitations need to be acknowledged.

First, the size of the validation dataset, particularly in the real-world clinical setting, was relatively small. This restricts the generalizability of the results, as the model's performance may vary with a more diverse or larger set of CNVs. A larger dataset, containing a wider variety of clinically validated CNVs, would be essential for confirming the robustness of the model

Second, the performance of the Targeted-CNV-Learner has only been tested on the SUDD147 gene panel, which may limit its applicability to other gene panels. Complex genomic regions with high variability, segmental duplications, or regions of low mappability could present significant challenges for CNV detection. Therefore, testing the model across a broader range of gene panels and genomic featus is crucial to fully evaluate its efficacy.

Another limitation is the requirement for at least 200 samples to generate a sufficiently large dataset of in silico CNVs for training the model. This need for a large sample size may present logistical challenges in certain clinical or research settings, especially in rare diseases where fewer patient samples are available. In these scenarios, the model's ability to generalize to smaller datasets remains untested, raising potential concerns about its adaptability.

Finally, the Targeted-CNV-Learner comes with an increased computational burden due to the integration of three separate CNV detection algorithms. Each algorithm must be run on every sample, which significantly increases the analysis time, particularly for larger datasets. This may pose challenges in clinical settings where time is a critical factor, and rapid turnaround times are required for patient diagnosis and treatment. Consequently, while the multi-algorithm approach improves accuracy, it may not always be feasible in situations requiring fast, high-throughput analysis.

In summary, while Targeted-CNV-Learner has demonstrated strong performance in CNV detection, addressing these limitations is key to optimize its clinical utility and expanding its applicability to a broader range of genomic contexts.

# CHAPTER 8
# CONCLUSION

The rapid evolution of genomic technologies has brought about transformative changes in our ability to understand, diagnose, and treat genetic diseases. Among these technologies, Next Generation Sequencing (NGS) has proven to be a powerful tool for decoding the human genome, leading to important advances in personalized medicine. However, despite these advancements, the detection of Copy Number Variants (CNVs) remains a significant challenge, particularly in the context of targeted sequencing panels used for specific disease studies, such as sudden cardiac death.

The work presented in this thesis was motivated by the need to improve the accuracy and reliability of CNV detection in targeted sequencing, a critical component of modern clinical diagnostics. Current standalone CNV detection algorithms are often plagued by high false positive rates, which complicate the diagnostic process by necessitating additional, costly orthogonal validation techniques, such as MLPA. This challenge provided the foundation for developing Targeted-CNV-Learner, a novel machine learning framework designed to integrate outputs from multiple CNV detection algorithms and improve overall performance.

Through the Targeted-CNV-Learner, this thesis has successfully addressed the key limitations of existing CNV detection methods. The model effectively combines outputs from multiple CNV detection algorithms—GATK gCNV, GRAPEs, and DECoN—leveraging caller-specific and genomic features to more accurately distinguish between true and false CNV calls. By training the model on a large dataset enriched with in silico CNVs and testing it against both synthetic and real-world clinically validated CNVs from the SUDD147 gene panel, Targeted-CNV-Learner has demonstrated significant improvements in both precision and specificity compared to standalone methods.

The results showed that Targeted-CNV-Learner achieved the best overall performance metrics on the test set across various performance indicators such as precision, specificity, and accuracy. Among all the models, Targeted-CNV-Learner stood out by obtaining the highest accuracy (95.5%) and significantly reducing false positives, thereby offering a substantial improvement in the reliability of CNV calls compared to DECoN, GRAPEs, and GATK gCNV. This improvement directly impacts clinical workflows, reducing the burden of unnecessary validations.

Moreover, when evaluated on real CNVs that had been experimentally validated, Targeted-CNV-Learner continued to excel. It achieved the best metrics alongside GRAPEs in terms of true positive detection, identifying 40 of the 41 validated CNVs, while also maintaining a notably low false positive rate. This ability to consistently perform well with real CNV data underscores the practical clinical utility of the model, particularly in reducing costly orthogonal validation procedures without compromising sensitivity.

In addition to its immediate clinical applications, the structured methodology of Targeted-CNV-Learner offers the flexibility to adapt to different gene panels, making it a versatile tool for various targeted sequencing studies. This adaptability, combined with its demonstrated performance in reducing validation costs and complexity, positions Targeted-CNV-Learner as a valuable contribution to the field of genetic diagnostics.

# Bibliography


[1] 3billion. Sequencing depth vs coverage. *3billion*, 2023.

[2] Fowler A. Decon: A detection and visualization tool for exonic copy number variants. *Methods in molecular biology (Clifton, N.J.)*, 2022.

[3] Fu-J. M. Lee S. K. Smirnov A. N. Gauthier L. D. Walker M. Benjamin D. I. Zhao X. Karczewski K. J. Wong I. Collins R. L. Sanchis-Juan A. Brand H. Banks E. Talkowski M. E. Babadi, M. Gatk-gcnv enables the discovery of rare copy number variants from exome sequencing data. *Nature genetics*, 2023.

[4] Zhao-L. Wang Y. et al. Chen, Y. Seqcnv: a novel method for identification of copy number variations in targeted next-generation sequencing data. *BMC Bioinformatics*, 2016.

[5] Tyrer-J.P. Walker L.C. et al. Dennis, J. Rare germline copy number variants (cnvs) and breast cancer risk. *Communication Biology*, 2022.

[6] Gouveia S. Couce M. L. Fernandez-Marmiesse, A. Ngs technologies as a turning point in rare disease research , diagnosis and treatment. *Current medicinal chemistry*, 2018.

[7] Mathieu Fusaro, Cyrille Coustal, Laura Barnabei, Quentin Riller, Marion Heller, Duong Ho Nhat, Cécile Fourrage, Sophie Rivière, Frédéric Rieux-Laucat, Alexandre Thibault Jacques Maria, and Capucine Picard. A large deletion in a non-coding regulatory region leads to nfkb1 haploinsufficiency in two adult siblings. *Clinical Immunology*, 261:110165, 2024.

[8] Rasmussen MS Andreu-Sánchez S Vieira FG Pedersen CB Kinalis S Madsen MB Kodama M Demircan GS Simonyan A Yde CW Olsen LR Marvig RL Østrup O Rossing M Nielsen FC Winther O Bagger FO Gabrielaite M, Torp MH. A comparison of tools for copy-number variation detection in germline whole exome and whole genome sequencing data. *Cancers (Basel)*, 2021.

[9] Amy S. Gargis, Lisa Kalman, and Ira M. Lubin. Assuring the quality of next-generation sequencing in clinical microbiology and public health laboratories. *Journal of Clinical Microbiology*, 54(12):2857–2865, 2016.

[10] Innan H. Glenfield, C. Gene duplication and gene fusion are important drivers of tumourigenesis during cancer evolution. *Genes*, 2021.

[11] Li G. R. Wang-R. J. Yi Y. T. Yang L. Jiang D. Zhang X. P. Peng Y. Guan, Y. F. Application of next-generation sequencing in clinical oncology to advance personalized treatment of cancer. *Chinese journal of cancer*, 2012.

[12] Upasana Mangrolia Sanober Waghoo Gulnaz Zaidi Shravani Rawool Ritesh P. Thakare Shahid Banday Alok K. Mishra Gautam Das Heena Satam, Kandarp Joshi and Sunil K. Malonia2. Next-Generation Sequencing Technology: Current Trends and Advancements. *Biology*, 12, 2023.

[13] Jennifer Kerkhof, Laila C. Schenkel, Jack Reilly, Sheri McRobbie, Erfan Aref-Eshghi, Alan Stuart, C. Anthony Rupar, Paul Adams, Robert A. Hegele, Hanxin Lin, David Rodenhiser, Joan Knoll, Peter J. Ainsworth, and Bekim Sadikovic. Clinical validation of copy number variant detection from targeted next-generation sequencing panels. *The Journal of Molecular Diagnostics*, 19(6):905–920, 2017.

[14] Schenkel L. C. Reilly J. McRobbie S. Aref-Eshghi E. Stuart A. Rupar C. A. Adams P. Hegele R. A. Lin H. Rodenhiser D. Knoll J. Ainsworth P. J. Sadikovic B. Kerkhof, J. Clinical validation of copy number variant detection from targeted next-generation sequencing panels. *The Journal of molecular diagnostics : JMD*, 2017.

[15] George Kirov. CNVs in neuropsychiatric disorders. *Human Molecular Genetics*, 24(R1):R45–R49, 06 2015.

[16] Takio Kurita. *Principal Component Analysis (PCA)*, pages 1–4. Springer International Publishing, Cham, 2019.

[17] Mademont-Soler I. del Olmo B. et al Mates, J. Role of copy number variants in sudden cardiac death and related diseases: genetic analysis and translation into clinical practice. *Eur J Hum Genet*, 2018.

[18] del Valle J. Castellanos E. et al. Moreno-Cabrera, J.M. Evaluation of cnv detection tools for ngs panel data in genetic diagnostics. *Eur J Hum Genet*, 2020.

[19] Zare F. Nabavi, S. Identification of copy number alterations from next-generation sequencing data. *Advances in experimental medicine and biology*, 2022.

[20] Anna C. Need, Dongliang Ge, Michael E. Weale, Jessica Maia, Sheng Feng, Erin L. Heinzen, Kevin V. Shianna, Woohyun Yoon, Dalia Kasperavičiūtė, Massimo Gennarelli, Warren J. Strittmatter, Cristian Bonvicini, Giuseppe Rossi, Karu Jayathilake, Philip A. Cola, Joseph P. McEvoy, Richard S. E. Keefe, Elizabeth M. C. Fisher, Pamela L. St. Jean, Ina Giegling, Annette M. Hartmann, Hans-Jürgen Möller, Andreas Ruppert, Gillian Fraser, Caroline Crombie, Lefkos T. Middleton, David St. Clair, Allen D. Roses, Pierandrea Muglia, Clyde Francks, Dan Rujescu, Herbert Y. Meltzer, and David B. Goldstein. A genome-wide investigation of snps and cnvs in schizophrenia. *PLOS Genetics*, 5(2):1–19, 02 2009.

[21] Quinlan AR Pedersen BS. Mosdepth: quick coverage calculation for genomes and exomes. *Bioinformatics*, 2018.

[22] Yeung M. H. Y. Wong A. N. N. Tsang H. F. Yu A. C. S. Yim A. K. Y. Wong S. C. C. Pei, X. M. Targeted sequencing approach and its clinical applications for the molecular diagnosis of human diseases. *Cells*, 2023.

[23] Hao Peng, Lan Lu, Zisong Zhou, Jian Liu, Dadong Zhang, Kejun Nan, Xiaochen Zhao, Fugen Li, Lei Tian, Hua Dong, and Yu Yao. Cnv detection from circulating tumor dna in late stage non-small cell lung cancer patients. *Genes*, 10(11), 2019.

[24] Jayakar G. Jensen M. Kelkar N. Girirajan S. Pounraja, V. K. A machine-learning approach for accurate detection of copy number variants from exome sequencing. *Genome research*, 2019.

[25] McLysaght A Rice, A. Dosage sensitivity is a major determinant of human copy number variant pathogenicity. *Nature Communication*, 2017.

[26] Soroush Samadian, Jeff P. Bruce, and Trevor J. Pugh. Bamgineer: Introduction of simulated allele-specific copy number variants into exome and targeted sequence data sets. *PLOS Computational Biology*, 14(3):1–18, 03 2018.

[27] Haldeman-Englert C. Geiger E. A. Ponting C. P. Webber C. Shaikh, T. H. Genes and biological processes commonly disrupted in rare and heterogeneous developmental delay syndromes. *Human molecular genetics*, 2011.

[28] Tamim H. Shaikh. Copy number variation disorders. *Genetic Medical Report*, 2017.

[29] Olsen M.F.-Lavik L.A.S. et al Singh, A.K. Detecting copy number variation in next generation sequencing data from diagnostic gene panels. *BMC Med Genomics*, 2021.

[30] de Santana-C. De T. Gresham D. Spealman, P. Multilevel gene expression changes in lineages containing adaptive copy number variants. *bioRxiv : the preprint server for biology*, 2024.

[31] Man J.-Wan Y. et al. Sun, Y. Targeted next-generation sequencing as a comprehensive test for mendelian diseases: a cohort diagnostic study. *Science Reports*, 2018.

[32] Toru Takumi and Kota Tamada. Cnv biology in neurodevelopmental disorders. *Current Opinion in Neurobiology*, 48:183–192, 2018. Neurobiology of Disease.

[33] Eric Talevich, A. Hunter Shain, Thomas Botton, and Boris C. Bastian. Cnvkit: Genome-wide copy number detection and visualization from targeted dna sequencing. *PLOS Computational Biology*, 12(4):1–18, 04 2016.

[34] Wolfe-K. McQuillin A. Viñas-Jornet M. Baena N. Brison N. D'Haenens G. Esteba-Castillo S. Gabau E. Ribas-Vidal N. Ruiz A. Vermeesch J. Weyts E. Novell R. Buggenhout G. V. Strydom A. Bass N. Guitart M. Vogels A. Thygesen, J. H. Neurodevelopmental risk copy number variants in adults with intellectual disabilities and comorbid psychiatric disorders. *The British journal of psychiatry : the journal of mental science*, 2018.

[35] Wu-C. F. Rajasekaran N.-Shin Y. K. Wang, L. H. Loss of tumor suppressor gene function in human cancer: An overview. *Cellular physiology and biochemistry : international journal of experimental cellular physiology, biochemistry, and pharmacology*, 2018.

[36] Niu L.-Wu B. He C.-Deng L. Chen C. Lan Z. Lin C. Kuang-W. Lin H. Zou J.-Zhang W. Luo Z. Yan, K. Copy number variants landscape of multiple cancers and clinical applications based on ngs gene panel. *Annals of medicine*, 2023.

[37] Xiguo Yuan, Junying Zhang, and Liying Yang. Intsim: An integrated simulator of next-generation sequencing data. *IEEE Transactions on Biomedical Engineering*, 64(2):441–451, 2017.

Appendix A

# Appendix

| Chr | Length | Type | Nº exons | DECoN | GATK | GRAPES | GC cont | Mapp | Gene | DECoN Q | GATK Q | GRAPES Q | Corr | TP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| chr7 | 439 | DUP | 1 | 1 | 1 | 1 | 58,41 | 1 | DPP6 | 18,7 | 83,12 | 0,91 | 0,8996 | True |
| chr10 | 18986 | DEL | 6 | 1 | 1 | 1 | 40,49 | 0,367 | MYPN | 74.5 | 1226.05 | 0.957 | 0,8996 | TRUE |
| chr1 | 219 | DEL | 1 | 1 | 0 | 0 | 28,64 | 1 | NEXN | 2,99 | - | - | 0,814 | FALSE |
| chr6 | 4759 | DUP | 3 | 1 | 0 | 0 | 35,17 | 0,375 | TRDN | 3,97 | - | - | 0,802 | FALSE |

Table A.1: Features provided as input to the model. Detection by DECoN, GATK, and GRAPES is indicated by 0 (not detected) or 1 (detected). Quality scores for these calls are indicated by 'Q' after the algorithm name. 'TP' (True Positive) indicates whether the variant was inserted in-silico (True) or is a false positive call (False).