IDENTIFYING STRUCTURAL VARIATIONS USING OPTIMIZED SV PIPELINE

Major Research Profile Project

Leon van Mierlo
l.vanmierlo@students.uu.nl

Prinses Maxima Centrum
Examiners: Freerk van Dijk, Jayne Hehir-Kwa, Roland Kuiper

Contents

Abstract

Structural variants (SVs) are genomic alterations of at least 50 base pairs in the DNA. The PrediCT study utilizes two gene panels to investigate tumour development linked to germline mutations in cancer predisposition genes. The aim of this project is to optimize an SV pipeline and identify if there are clinically significant SVs, focussing on deletions, in genes from the gene panels in PrediCT patients.

SV callers identify genomic alterations in the DNA and produce a VCF file. However, currently there is not a single caller good enough for accurate and comprehensive detection of SVs (Koboldt, 2020) (Kuzniar et al, 2020) (Kosugi et al, 2019). Hence, SV callers Manta and Dysgu are combined for a more accurate and comprehensive detection of SVs. Both callers contain a property in their VCF file useful for validating the accuracy of the event and can process CRAM files, therefore significantly reducing the pipeline's runtime.

The pipeline is optimized by setting thresholds for Dysgu's Probability Score- and Manta's Quality Score property, based on event verification in IGV (Robinson et al, 2011), which serve as filter. SV length and high-confidence-calls from a single caller are also filtered in. Events of interest are annotated using AnnotSV, annotation software specialized for SVs, and exon-region filtering is performed. Many identified SVs lack clinical significance due to being population-common.

Combining Dysgu and Manta showed improved results over the use of them as a single caller. With an equivalent number of correctly identified events, fewer false positives are called when combing the callers.

In the analysis, the pipeline was unable to detect any novel clinically significant SVs beyond those that were already established.

Plain language summary

In this study, the focus was to detect structural variants (SVs) in DNA, which are significant changes in the DNA of 50 or more base pairs. Specifically, it is looked at SVs in genes which are known to increase the risk of cancer development. The goal was to improve the process of identifying these SVs, with a particular emphasis on deletions, in patients from the PrediCT study by creating an SV calling pipeline. The SV calling pipeline has as input files with information about DNA and as output which mutations in the DNA are interesting and might be the cause of the cancer.

To reach this goal, specialized software known as “SV callers” are used. These SV callers find changes in the DNA and generate a VCF file, which is essentially a detailed record of DNA sequence variations, also known as mutations. However, no single SV caller is currently perfect at identifying all SVs accurately. To overcome this, two different SV callers are combined, Manta and Dysgu, to get a more comprehensive and precise detection of SVs. These tools do not only give the mutations, but they also have features that confirm the accuracy of the detections. Plus, they can process data in a way that speeds up the pipeline.

The detection process is improved by setting specific criteria for the SVs to meet, which helps filter out less likely SV candidates. Different criteria are tested to find out what good filters are for SV calls. To add more information to the interesting findings and focus on specific areas within the genes, AnnotSV is used. Many of the found SVs are common in the general population and not necessarily linked to cancer.

Three interesting pipeline runs are done: the first run used no filters, the second run used to filter which were determined in this project, the third run looked at more genes compared to the first two executions.

By combining Manta and Dysgu, better results are achieved compared to using either one alone. This approach led to fewer incorrect identifications while maintaining the same level of correct detections. However, it is important to note that the improved pipeline did not uncover any new SVs linked to cancer that were not already known. There are several reasons no clinically significant SVs are found. The first reason is there is only looked only at deletions, which is only one of the SV types. There can be other SV types in the data. Another reason might be the deletions are not detected during SV calling. It is also possible there are no more clinically significant SVs in the data.

Introduction

The quantity of a specific gene present in an individual's genome can differ. When there is a variation in the amount of a particular gene, it is known as Copy Number Variation (CNV). CNVs are caused by duplications, deletions or other genomic rearrangements that result in an imbalanced gene dosage (Shaikh, 2017). Structural variants (SVs) are genomic alterations of at least 50 base pairs in the DNA (Mahmoud, et al, 2019). There are several different SVs we discern: deletions (DEL), duplications (DUP), insertions (INS), translocations (TRA) and inversions (INV). Not all SVs result in a variation of copy number. To illustrate, when part of the DNA is inverted, the gene dosage does not change, and therefore there is no variation of copy number. SVs can have an impact on the phenotype (Mahmoud, et al, 2019). Such change in the genome can result in up- or downregulation of a gene, or even fully disrupting the gene. Not all SVs have the same impact on the phenotype. When an SV is located in an exon, the effect will likely be larger than if the SV is located in an intron. When an SV results in a frameshift, it is more probable that it will significantly affect the phenotype. During translation, every three bases are translated into a single amino acid. If this triplet sequence is altered by a deletion or an insertion, a frameshift occurs. This frameshift affects the entire sequence until a stop codon is encountered. The position of this stop codon may be shifted earlier or later in the sequence as a consequence of the frameshift.

Identifying SVs in germline data is different compared to identifying SVs in tumour data. Identifying tumour-specific SVs poses a significant challenge. This challenge arises from variability in pinpointing exact breakpoints, the diverse types of variants that can be derived, and the biological nature of certain rearrangements (van Belzen, Schönhuth, Kemmeren, & Hehir-Kwa, 2021). Cancer genomes exhibit higher levels of genomic instability. This instability leads to more, and more complex, SVs compared to germline variations. This complexity is highlighted by complex SVs, which are identified through clustering of numerous breakpoints. A complex SV means there are multiple types of SVs in the same region. These complex SVs are believed to result from a singular catastrophic process, followed by ongoing rearrangements or repair processes. The clustering of breakpoints complicates deducing the genomic rearrangements. This results in complexity of identifying the events that caused the tumour (van Belzen et al, 2021). There have been different types of SVs identified in cancer which cause dysfunction of the gene. The first type is where a SV causes an upregulation in gene dosage, resulting in a large overexpression of the gene. The second type of SV is a gene fusion (Mahmoud et al, 2019). SVs can combine multiple genes across chromosomes. The third type of SV is a change in gene expression, caused by a change in location of gene regulatory elements (Mahmoud et al, 2019).

The potential effect of a mutation is determined by the significance of the gene impacted by the SV. To illustrate, if an SV disrupts a tumour-suppressor gene, there is an increased change to the development of a tumour. However, not all genes function as tumour suppressors, and therefore, not all SVs necessarily result in tumour development. Cancer predisposition genes are the genes in which a germline mutation increases the risk of getting cancer (Rahman, 2014). Up to 10% of the cancers are caused by mutations in cancer predisposition genes (Wang, 2016). However, not all of these mutations are SVs, the majority of this percentage is caused by single nucleotide polymorphisms (SNPs).

In the PrediCT study (PREDIsposition to Childhood Tumors) a "genotype first approach" is used instead of the most commonly used "phenotype first approach" in children with cancer. To

illustrate how a genotype first approach can be effective: In 43.3% of children with low-hypodiploid acute lymphoblastic leukaemia mutations in TP53, a cancer predisposition gene, were found which were also found in non-tumour cells (Holmfeldt et al, 2013). This suggests that this is a germline mutation which could be found before the tumour was developed. This example illustrates why using a genotype first approach is interesting when we think about tumour development. In the PrediCT study, children between age 0 to 19 are used. All children in this study have been diagnosed with a type of cancer. In PrediCT, the focus is on determining whether the cancer originates from a germline mutation in one or more cancer predisposition genes. The PrediCT study uses two gene panels, one gene panel contains 139 genes and one gene panel contains 400 genes. Both gene panels are a collection of genes which are known to be linked with child cancer predisposition syndromes in children with cancer. There is whole genome sequencing (WGS) data of 355 children available. The PrediCT study is done with whole exome sequencing (WXS) data. WXS data is used for trying to understand what causes a disease or what is causing symptoms. Given that WXS only sequences exomes, there is a likelihood that segments of SVs may not be sequenced and consequently remain undetected. Therefore, WGS data is more appropriate for identifying SVs.

SVs are found through DNA sequencing. The sequencer produces DNA reads of a certain length. These reads are compared to a reference genome, the difference between the sequenced reads and the reference gives information about SVs and other genetic alterations. SVs are often harder to detect than SNPs. This is because the length of SVs can be larger than the read length, making mapping the read to the reference genome a difficult task. Therefore, the length of the reads is important for finding SVs. Short read sequencing is currently the most used form of sequencing (van Campen, 2022). The length of the reads with this form of sequencing is usually between 50 and 300 bases but can go up to 600 bases (Amarasinghe et al, 2020). A more recent form of sequencing is called long read sequencing. Long read sequencing can produce reads with a length over 10 kilobases (Amarasinghe et al, 2020). Long read sequencing is therefore more suitable for finding SVs. This method of sequencing is, however, much more expensive and consequently not as much used as short read sequencing.

The sequenced reads are mapped against a reference genome. The Human Genome Project started deciphering the entire human genetic code (Collins & Fink, 1995). This project has undergone consistent enhancement over the past two decades (Nurk et al, 2022). When the reads are mapped against a reference genome, a Sequence Alignment Map (SAM) file is produced. A SAM file can be compressed to a binary representation of the information, this is called a Binary Alignment Map (BAM) file. Compressed files require less time for transferring and require significantly less storage capacity than uncompressed files. There is an alternative to SAM- and BAM files, called a Compressed Reference-orientated Alignment Map (CRAM) file. A CRAM file reduces storage costs by describing the differences between reference sequence and the aligned sequence reads.

The identification of SVs in an alignment file is done by SV callers. SV callers produce a Variant Call Format (VCF) file where the gene sequence variations are described. SVs in the genome can be found using tools made for germline- and somatic detection of SVs. There are many different of these callers developed over the last few years. However, there is currently still not a single caller good enough for accurate and comprehensive detection of SVs. For that reason, it is recommended to combine different callers for a more accurate and comprehensive detection of SVs (Koboldt, 2020) (Kuzniar et al, 2020) (Kosugi et al, 2019). Combining different callers means

merging the SVs in VCF files produced by the callers. When doing this, some SVs are found by multiple callers while other SVs are found by less callers. There are many different callers and this number keeps increasing with the rapid growing next generation sequencing (NGS) methods. A big problem is the lack of comparability between these tools (Wittler, Marschall, Schönhuth & Mäkinen, 2015). This lack of comparability comes from errors during sequencing, an uncertainty in breakpoints, and in repetitive regions are multiple possible representations of SVs. These problems result in a challenge for comparing and merging SVs (Sedlazeck et al, 2017). In some cases, different callers describe more than one SV type in the same region. Or even a single caller describes more than one SV type in the same region. This could be the result of an error in the call made or there can be a complex SV in that region. A tandem duplication of a large region could be described by some callers as a novel insertion, while other callers describe this event as a duplication. These examples illustrate how different SV callers might disagree over the same data and the complexity of merging calls from different callers (Sedlazeck et al, 2017).

The research question is: “Can we identify constitutional small structural variants of clinical significance in cancer predisposition genes in children with cancer?”. This question is answered by creating an optimised SV pipeline.

Material & Methods

This chapter gives a description of the used cohorts, how choices are made to establish the pipeline (Fig. 1). Two SV callers are combined for a more accurate and comprehensive detection of SVs. Several filter steps are applied, followed by annotation and filtering on exon regions. Each subchapter in this chapter is dedicated to a detailed explanation of the steps illustrated in Fig. 1.

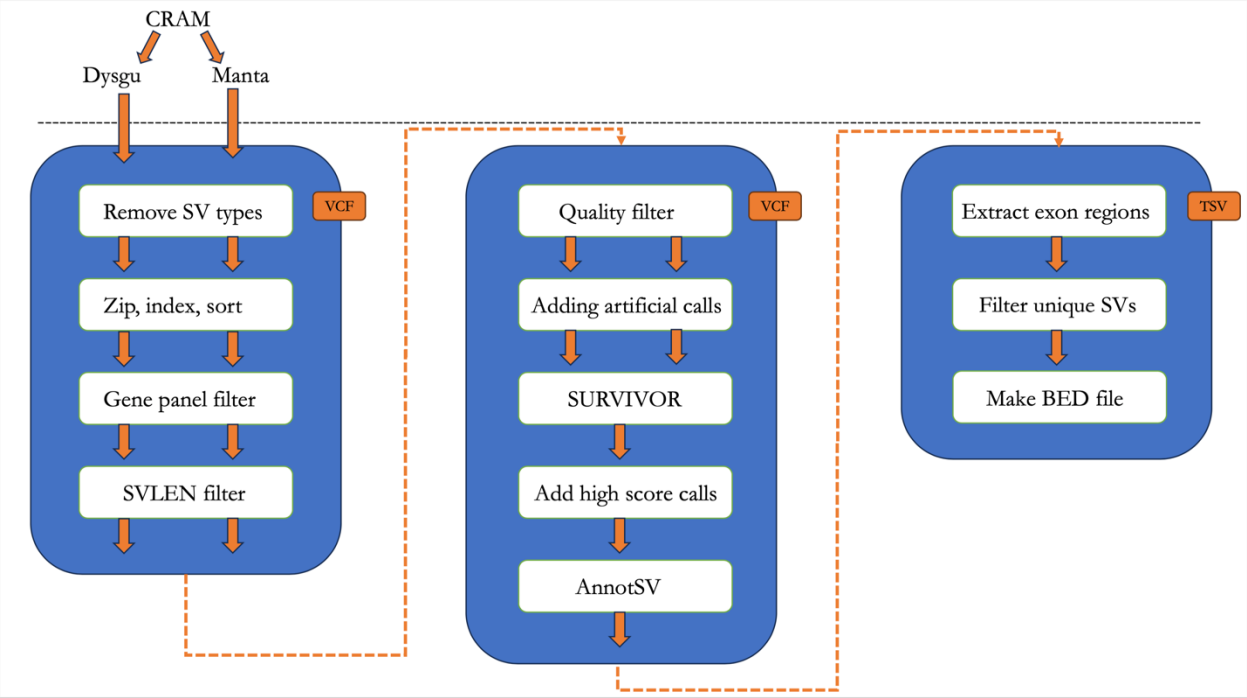


Figure 1. Schematic overview of the pipeline used to process the PrediCT samples. The pipeline has as input VCFs produced by Dysgu and Manta

Cohort

Two different cohorts are used for this project. One of the cohorts of our interest, the PrediCT samples with WGS data and proper consent, contains 318 samples. For PrediCT's samples, there is only proper consent to look in genes from the PrediCT gene panels, making it not suited for testing. For all the 318 samples, there is proper consent to identify mutations in the first gene panel, consisting of 139 genes. For the second gene panel, consisting of 400 genes, is only proper consent for 37 samples.

For the testing of the pipeline, three samples from the ROHHAD study (Manuscript In Preparation) are used. The ROHHAD study is a study done by Nienke van Engelen. The sample used from ROHHAD is suited here because there is proper consent to look into all sequencing data. For some ROHHAD samples, optical mapping has been used and the genome of the parents is also sequenced.

For the first PrediCT gene panel, two SVs are already identified. There is a 172679 base pair deletion in the FANCA gene and an exon deletion in SDHB of 7904 base pairs.

Combining SV callers

Not a single caller is good enough for accurate and comprehensive detection of SVs. Hence, it is recommended to combine different callers (Koboldt, 2020) (Kuzniar et al, 2020) (Kosugi et al, 2019). For this project, five SV callers are tested and compared. Manta (Chen et al, 2016) is a tool developed in 2016 and is widely used for the detection of indels and structural variants in cancer- and germline sequencing analysis. Dysgu (Cleal & Baird, 2022) is a recently developed SV caller from 2022 and uses machine learning to classify which makes it unique compared to the other SV callers. GRIDSS2 (Cameron et al, 2021) is the newer version from GRIDSS and is released in 2021. Both GRIDSS and GRIDSS2 use positional de Bruijn graph assembly to assemble the reads that potentially support a structural variant (Cameron et al, 2017) (Cameron et al, 2021). Lumpy (Layer, Chiang, Quinlan & Hall, 2014) and Delly (Rausch et al, 2012) are both also older, and therefore more tested, SV callers released in 2014 and 2016. These five SV callers differ in their approach to detect SVs, their release duration and the extent of how much they have been used in previous research. Different types of SV calls per caller can be observed in Table 1.

	DEL	DUP	INS	TRA	INV	BND	SGL	BAM	CRAM
Manta	■	■	■		■	■		■	■
GRIDSS2	■	■	■		■	■	■	■	
Delly	■	■	■		■	■		■	
Lumpy	■	■			■	■		■	
Dysgu	■	■	■	■	■			■	■

Table 1. For five SV callers (Manta, GRIDSS2, Delly, Lumpy, Dysgu) shown which SVs they identify (DEL=deletion, DUP=duplication, INS=insertion, TRA=translocation, INV=inversion, BND=breakend, SGL = single breakend SV support). CRAM indicates if the caller can directly work with CRAM files.

SURVIVOR (Jeffares et al, 2017) can merge the VCFs produced by different callers effectively. It ensures each SV found by multiple callers appears only once in the VCF with an annotation of detecting caller(s). The output by SURVIVOR is a new VCF with all merged events.

Property filtering

On average there are around 4400 SVs per individual in germline data, predominantly deletions (Abel et al, 2020). Considering that the callers called ~2x till ~19x the average amount of SVs (Table 3) in germline data, many SVs are likely incorrectly called. When there is an interest in clinically significant SVs, the VCF file should have a high percentage of correct calls. Each VCF produced by the caller gives some properties with the called event. Examples of such properties are a quality score, number of paired-reads supporting the variant, number of pieces of evidence supporting the variant, a probability score and the mean map quality for primary reads supporting the variant. To filter the VCF files on a higher percentage correctly called SVs, these properties given with the calls might be interesting. When there are more total pieces of evidence supporting a variant and the quality of the mapping is higher, it would be logical if an event is then more likely to be correct. To

test this, called events are checked in Integrative Genomics Viewer (IGV), a visualisation tool that facilitates the analysis of extensive datasets on regular desktop computers (Robinson et al, 2011). When an event is verified in IGV, it is labelled as true or false in the VCF file. By doing this, two groups are created: the correctly-called-group and the incorrectly-called-group. If enough events are checked, a difference might be shown in values of certain properties between the two groups. If there is a clear difference in value of a certain property between these groups, this property might be a good indication of an event being correct and a threshold can be set. For example, if all correctly called events have a quality score of 700 or larger and all incorrectly called events have a quality score lower than 700, a threshold can be made which filters a VCF file on keeping only the SVs which have a quality score higher than 700 for that property. In this way, the percentage of correct SVs in the VCF will be higher. This filter is applied using a Python script (Appendix 1).

For each caller are called events checked to determine useful properties for setting a threshold. These events are checked in a sample from the ROHHAD study. Each event exhibits unique properties for each identifying caller. The resulting "SURVIVOR VCF" contains a reduced set of properties. To enable a single SV verification in IGV to account for multiple callers, when an event is called by multiple callers, and retain the original properties, the true- and false labels are recorded within the SURVIVOR VCF file. A Python script is made to annotate "true" or "false" for each specific event in the original VCF (Appendix 1). Each event is assigned a unique ID within its original VCF, which is also located in the SURVIVOR VCF, facilitating linkage of true- and false labels between the two. The properties of interest are those that distinguish between SVs labelled as true or false.

Multiple properties show correlation with being correctly called but almost all show no clear threshold value. An example is shown in Figure 2. This property might give an indication of a call being correctly called but it is not convincing enough. If a threshold would be set for this property, too many correct calls would be falsely removed from the VCF. Better properties to use when trying to determine if an SV is correctly called are the probability score from Dysgu and the quality score from Manta (Fig. 3, Fig. 4). In these figures there is a clearer difference between correctly- and incorrectly called SVs. For these properties, a threshold can be determined which can be used to filter the VCF files.

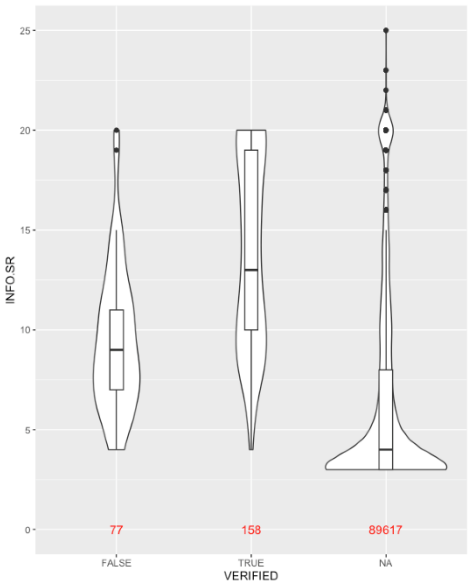


Figure 2. Number of split-reads supporting a call made by Delly. This property suggests potential accuracy for an event. However, this measure is not sufficiently reliable to establish a definitive threshold.

Determining SV callers

Dysgu and Manta are chosen as callers for the pipeline. Based on Figure 3 and Figure 4, Manta and Dysgu both have a property which is useful for indicating if an event is likely correct. VCFs from Manta and Dysgu can therefore be filtered on these properties. Only calls made by Dysgu with a probability score >= x and calls made by Manta with a quality score >= x are used. In this way, the percentage of correct calls in the VCF is higher.

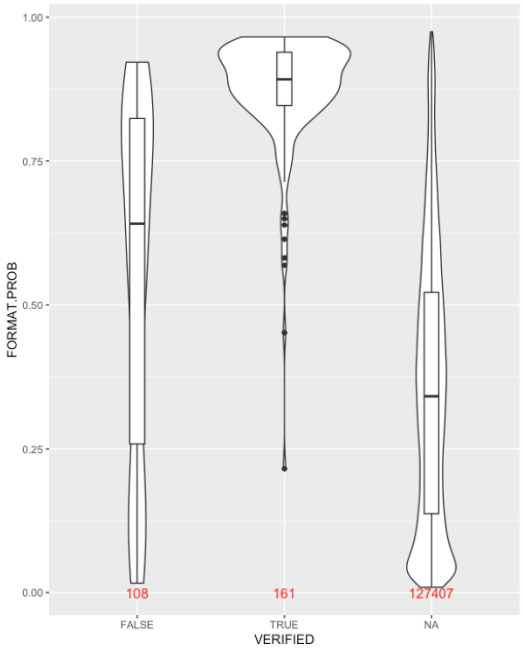


Figure 3. The Probability Score assigned by Dysgu for both verified correct- and incorrect calls shows a notable distinction in the probability values between calls made accurately and those made inaccurately.

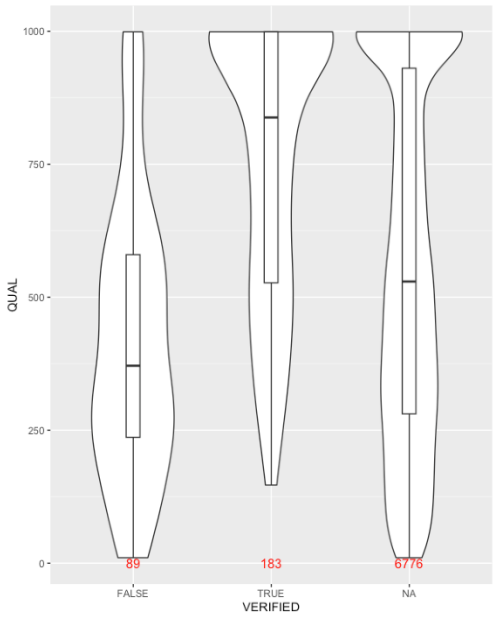


Figure 4. The Quality Score assigned by Manta for both verified correct- and incorrect calls shows a notable distinction in the probability values between calls made accurately and those made inaccurately.

Runtime plays an important role in the pipeline's efficiency. Both Dysgu and Manta are capable of processing CRAM files (Table 1), resulting in a significantly reduced runtime.

Removing SV types

Diverse SVs require distinct approaches for detection, resulting to not all SV types are identified by every SV caller (Table 1). This differentiation is due to the unique operational mechanisms of each SV type, which demand specific thresholds for each property and SV type. For instance, for a deletion, a Dysgu probability score of 0.65 might be sufficient to confirm that the call is likely correct. However, for accurately predicting translocations, a higher probability score of 0.90 may be required. Consequently, it's advisable to set a threshold for only SV type. Given that deletions are the most frequently occurring SV type, the focus is solely on these in the pipeline.

Gene panel filtering

The PrediCT study aims to identify clinically significant SVs in specific gene panels, in line with the consent obtained for the patients. Manta and Dysgu both generate a VCF file. The next step uses bcftools (Danecek et al, 2021) to confirm if each SV aligns with a gene in the gene panel. This tool filters the SVs by comparing the VCF file with a BED file or regions-of-interest file, ensuring that only SVs which are relevant are included.

SV length filtering

The pipeline also filters on SV length using a Python script (Appendix 1). In the VCF files are SVs from several tens of base pairs up to over a million base pairs. Using PrediCT's germline data, it is improbable to find undetected SVs spanning hundreds of thousands of base pairs, and if entire genes or numerous exons were deleted, they would most likely have already been identified since Copy Number Variation (CNV) analysis are already performed. But these very long SVs cause a lot of noise in the data. After applying the threshold filter for the properties, it is still possible that a long SV had a score above this threshold since the threshold will never remove all incorrect calls from the data. The threshold values are adjusted to a moderate level to ensure that correct events are not excessively excluded. For that reason, there is a possibility a long SV ends up in the filtered data. If this event gets annotated, the output is filled with noise. Each gene that falls within this event will be annotated, making it harder to find more likely correct events. To avoid this problem, the assumption is made that very large SVs are not in the data. Hence, only events are kept that fall below a certain length. This length is set at 200000 bp. This length of 200000 bp is so that the extremely large SVs are not in the data but the already known SVs are found.

Artificial VCF merging

Dysgu and SURVIVOR do not work together properly. When trying to merge a VCF with a relatively low SV count (~50) or less, with a VCF produced by Dysgu, a segmentation fault error will be produced (Fig. 5). "Segmentation fault: 11" is an error code for SURVIVOR trying to access memory, where it does not have the access for (Finn, 2013). The error is unpredictable because it does not show up every time you run the same command. The same command is run five times and only four times the error occurs (Fig. 5). The issue shown in Figure 5 seems to have no solution. Trying to find a solution, Fritz Sedlazeck, the developer of SURVIVOR, was contacted to help solve this issue. Unfortunately, Fritz Sedlazeck was unable to provide a solution for this error. He suggested using an alternative program of his called Truvari (Sadlazeck, 2023) (English, Menon, Gibbs, Metcalf, & Sedlazeck, 2022), but due to the stage of the internship, transitioning to a new program was not

```
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$ ~/SURVIVOR/Debug/SURVIVOR merge fileList 1000 1 0 0 0 0 sampleMergedWithSURVIVOR.vcf
merging entries: 12
Segmentation fault: 11
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$ ~/SURVIVOR/Debug/SURVIVOR merge fileList 1000 1 0 0 0 0 sampleMergedWithSURVIVOR.vcf
merging entries: 12
Segmentation fault: 11
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$ ~/SURVIVOR/Debug/SURVIVOR merge fileList 1000 1 0 0 0 0 sampleMergedWithSURVIVOR.vcf
merging entries: 12
merging entries: 342
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$ ~/SURVIVOR/Debug/SURVIVOR merge fileList 1000 1 0 0 0 0 sampleMergedWithSURVIVOR.vcf
merging entries: 12
Segmentation fault: 11
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$ ~/SURVIVOR/Debug/SURVIVOR merge fileList 1000 1 0 0 0 0 sampleMergedWithSURVIVOR.vcf
merging entries: 12
Segmentation fault: 11
Admins-MacBook-Pro-2:/Users/leonvanmierlo/Desktop/SURVIVOR
$
```

Figure 5. When attempting to merge a VCF file with relatively low SV count from Dysgu using SURVIVOR, it occasionally results in a segmentation fault error. Interestingly, this error is inconsistent, as repeating the same command multiple times does not always produce the error.

feasible. Instead of transitioning to a new program, a workaround was identified by merging the VCF from Manta with an artificial VCF containing non-existent mutations (Fig. 6). This new VCF includes 2000 fabricated mutations combined with the genuine mutations found by Manta. Merging the Dysgu VCF with this new VCF, which contains artificial- and Manta's mutations, successfully circumvents the segmentation error. The artificial mutations do not persist in the final SURVIVOR merged VCF because these mutations are not found by both callers. This results in a file exclusively containing calls from Manta and Dysgu.

The non-existing mutations are made using a python script (Appendix 1) and are all deletions with a length of 100bp with a start location 1000bp from each other. Each non-existing mutation has an ID which contains "mantaArtDEL". The incorporation of "Art" in the original Manta ID is not a possibility. Therefore, in the event that Dysgu found an SV at a same location as an non-existing SV and these get merged, this can readily be identified since the original IDs are included in the SURVIVOR VCF.

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  XXXXXXXXXX
chr1    101     mantaArtDEL:00001:0:0:0:0:0     C       <DEL>   101     SampleFT        END=1101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    201     mantaArtDEL:00002:0:0:0:0:0     C       <DEL>   101     SampleFT        END=2101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    301     mantaArtDEL:00003:0:0:0:0:0     C       <DEL>   101     SampleFT        END=3101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    401     mantaArtDEL:00004:0:0:0:0:0     C       <DEL>   101     SampleFT        END=4101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    501     mantaArtDEL:00005:0:0:0:0:0     C       <DEL>   101     SampleFT        END=5101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    601     mantaArtDEL:00006:0:0:0:0:0     C       <DEL>   101     SampleFT        END=6101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr1    701     mantaArtDEL:00007:0:0:0:0:0     C       <DEL>   101     SampleFT        END=7101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
```

Figure 6. Artificial VCF with fake deletions.

SURVIVOR merging

The VCFs are merged using SURVIVOR. SURVIVOR takes 8 arguments. The first argument contains a file with the paths to the VCFs to be merged. The second argument is the maximum distance between breakpoints. The recommended distance for this by SURVIVOR is 1000 base pairs. When the same event is found by multiple callers, the distance between breakpoints is suspected to be much less than 1000 base pairs. Slight variances are expected due to some breakpoint uncertainties. To test this hypothesis, for the same single ROHHAD sample that was used before, the merging distance is identified when the merging distance is set at 1000 base pairs (Fig. 7). As anticipated, the vast majority of merged events occur within a narrow range of base pairs. There are merging distances greater than 1000 base pairs, even though the merging distance is set at a maximum 1000 base pairs. This occurs when one breakpoint is located within a 1000 base pairs range, leading to merging the events, while the distance of the other breakpoint exceeds 1000 base pairs. Thus, when merging events with a maximum distance of 1000 base pairs, the vast majority of events get merged within a narrow range of base pairs, hence for the pipeline the recommended merging distance of 1000 base pairs is kept. The third argument is the minimum number of supporting callers, this argument is set on two. Combining callers show improved results over one caller (Fig. 11). The fourth, fifth, sixth and seventh argument are set at 0. The eight argument is the output.

When using SURVIVOR, the command requires a file specifying the paths of the VCFs intended for merging. To ensure that SURVIVOR works properly with Dysgu, it is crucial to position the path of Dysgu's VCF at the bottom, or close to the bottom. The order matters when using Dysgu's VCF, the paths of the VCFs from other callers should precede Dysgu's VCF path. If Dysgu's file path is at the top of the file, there is an increased risk of encountering a "Segmentation fault: 11" error (Fig. 5). A workaround involves merging an additional artificial VCF for merging (Fig. 8). Consequently, the file read by SURVIVOR for merging contains three VCFs: Manta's VCF merged with 2000 artificial SVs, the original Dysgu VCF and an artificial VCF containing 5000 fabricated mutations. It is essential to note that the 2000 mutations merged with Manta's VCF and the artificial VCF containing 5000 SVs should not share the same artificial mutations. This differentiation is crucial because SURVIVOR identifies SVs that occur at least twice in total: therefore, unique artificial mutations will not be included in the merged VCF.

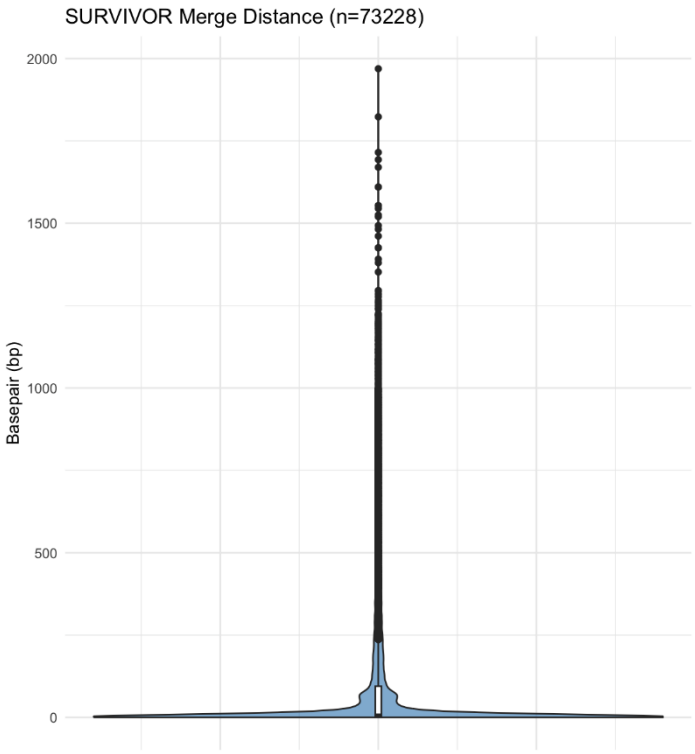


Figure 7. Amount of base pairs between merged events from different SV callers. When setting the merging distance at 1 kbp for the combined calls, it is observed that certain merging distances exceed 1 kbp. This occurs because one breakpoint is within the 1 kbp range, leading to the merging of events, while the other breakpoint lies at a greater distance.

```
chr0    101     mantaArtDEL:00001:0:0:0:0:0     C       <DEL>   101     SampleFT        END=1101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    201     mantaArtDEL:00002:0:0:0:0:0     C       <DEL>   101     SampleFT        END=2101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    301     mantaArtDEL:00003:0:0:0:0:0     C       <DEL>   101     SampleFT        END=3101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    401     mantaArtDEL:00004:0:0:0:0:0     C       <DEL>   101     SampleFT        END=4101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    501     mantaArtDEL:00005:0:0:0:0:0     C       <DEL>   101     SampleFT        END=5101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    601     mantaArtDEL:00006:0:0:0:0:0     C       <DEL>   101     SampleFT        END=6101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
151;CIEND=-175,175      GT:FT:GQ:PL:PR  1/1:MinGQ:1:101,1,0:1,11
chr0    701     mantaArtDEL:00007:0:0:0:0:0     C       <DEL>   101     SampleFT        END=7101;SVTYPE=DEL;SVLEN=-100;IMPRECISE;CIPOS=-150,
```

Figure 8. Artificial VCF with non-existing deletions.

Adding high score calls made by single caller

Manta's quality score and Dysgu's probability score have proven to be a good indication of an event likely to be correct (Fig. 3, Fig. 4). Hence, events with a high quality- or probability score are likely to be correct events. Events only found by Dysgu with a probability score over 0.90 and events found by Manta with a quality score over 950 are seen as high confidence calls. These are filtered using a Python script (Appendix 1). If one caller misses an event but the other caller found this event with a high confidence score, it can still be an interesting event. For that reason, events which are found by one caller with a high confidence score will also be annotated. These events are placed in its

own folder, separated from the SURVIVOR merged calls. For those who may not find interest in events exclusively found by a single caller, it is straightforward to avoid them by solely focussing on the events in the map with events from the SURVIVOR.

AnnotSV

Events of interest are annotated using AnnotSV (Geoffroy et al, 2023) (Geoffroy et al, 2021) (Geoffroy et al, 2018). AnnotSV is a command-line annotation tool written in the Tcl programming language, specialised for annotating SVs. It can be executed on different operating systems and be integrated in NGS analysis pipelines (Geoffroy et al, 2018). The required input for AnnotSV is a VCF or BED file. The annotation is performed by identifying the overlap between the annotation features and the input. Annotation can be carried out utilising either the GRCh37 or GRCh38 version of the human genome build. For the PrediCT samples, GRCh38 is used. Annotations linked to the gene name will also be reported. AnnotSV generates for each SV two types of annotation, the first type is the annotation based on the full length SV. The second type is annotation for each gene that falls within the SV. The output is a TSV file. This TSV file can be opened in different spreadsheet programs, for example Excel.

Events found by Dysgu and Manta and events by one caller with a high confidence score are annotated and placed separately in two different folders.

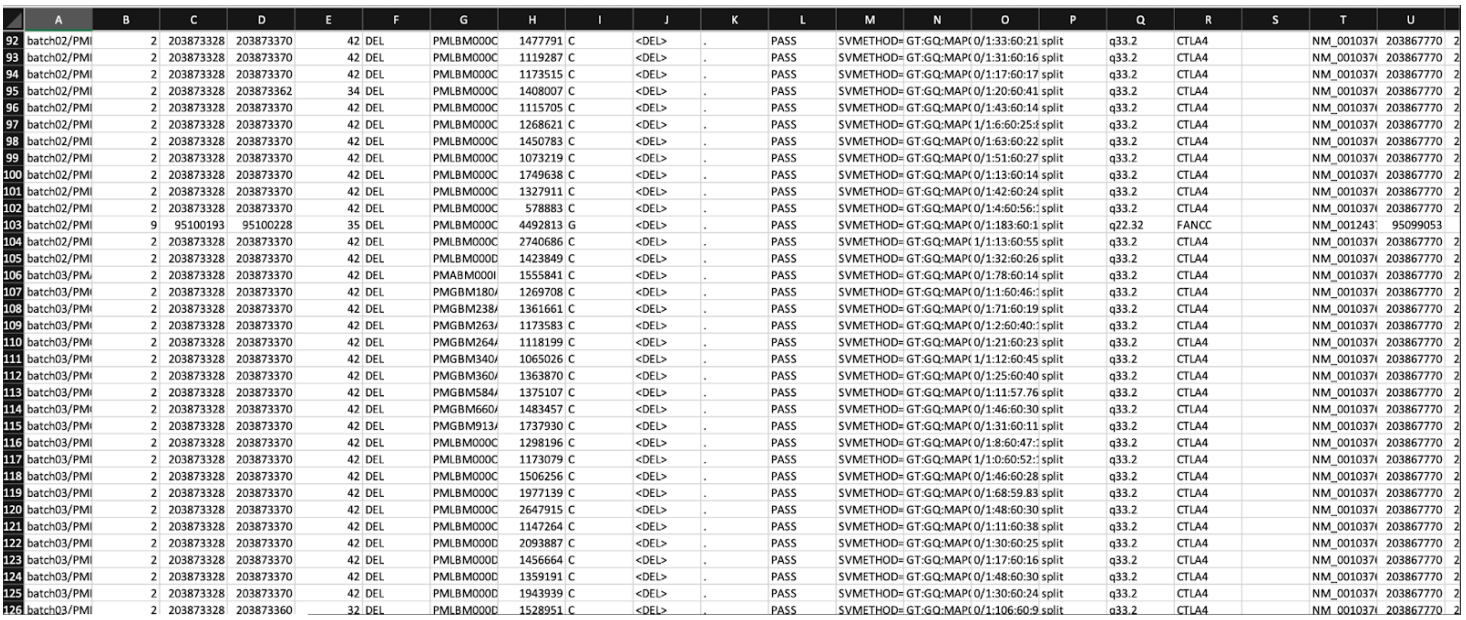
Extracting exon information

Most found SVs are in introns and are therefore not of clinical significance. The events of interest are SVs that remove bases within an exon. AnnotSV gives in its annotation the location of the event. If the deletion removes bases from an exon, it is annotated here in different ways. The first type of annotation are annotations containing “exon”, two examples to illustrate: “exon4-exon4” and “intron2-exon3”. The second type of annotation which can contain an exon is “intronX-intronX+”, thus when the deletion removes multiple introns, exons will be removed as well. An example of how it could look in the annotation file is “intron4-intron5”. The remaining two annotations do not always cover an exon, but since it is a possibility, these events should be verified. The annotations are as follows: “txStart-intronX” and “intronX-txEnd”. It is a possibility that an exon falls between the transcription start and an intron or between an intron and the transcription end, however it is not necessarily.

Filtering for uncommon mutations

Many of the found mutations are mutations that are common for a certain population and are therefore not of interest when the interest is in SVs of clinical significance (Fig. 9). Since these SVs are not of interest, they should not be in the output TSV. Removing these SVs is done with a Python script (Appendix 1) that has as input the TSV file with common mutations and an integer, N, which represents the maximum number of times an SV is allowed and as output a new TSV. The script

checks which chromosome the SV is on and the start location and if the same chromosome and start location is seen more than N times, they are not written in the new TSV.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
92	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1477791	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:33:60:21	split	q33.2	CTLA4		NM_001037(	203867770
93	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1119287	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:31:60:16	split	q33.2	CTLA4		NM_001037(	203867770
94	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1173515	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:17:60:17	split	q33.2	CTLA4		NM_001037(	203867770
95	batch02/PMI	2	203873328	203873362	34	DEL	PMLBM000C	1408007	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:20:60:41	split	q33.2	CTLA4		NM_001037(	203867770
96	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1115705	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:43:60:14	split	q33.2	CTLA4		NM_001037(	203867770
97	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1268621	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	1/1:6:60:25:6	split	q33.2	CTLA4		NM_001037(	203867770
98	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1450783	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:63:60:22	split	q33.2	CTLA4		NM_001037(	203867770
99	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1073219	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:51:60:27	split	q33.2	CTLA4		NM_001037(	203867770
100	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1749638	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:13:60:14	split	q33.2	CTLA4		NM_001037(	203867770
101	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1327911	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:42:60:24	split	q33.2	CTLA4		NM_001037(	203867770
102	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	578883	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:4:60:56:1	split	q33.2	CTLA4		NM_001037(	203867770
103	batch02/PMI	9	95100193	95100228	35	DEL	PMLBM000C	4492813	G	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:183:60:1	split	q22.32	FANCC		NM_001243	95099053
104	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000C	2740686	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	1/1:13:60:55	split	q33.2	CTLA4		NM_001037(	203867770
105	batch02/PMI	2	203873328	203873370	42	DEL	PMLBM000D	1423849	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:32:60:26	split	q33.2	CTLA4		NM_001037(	203867770
106	batch03/PM	2	203873328	203873370	42	DEL	PMABM000I	1555841	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:78:60:14	split	q33.2	CTLA4		NM_001037(	203867770
107	batch03/PM	2	203873328	203873370	42	DEL	PMGBM180/	1269708	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:1:60:46:1	split	q33.2	CTLA4		NM_001037(	203867770
108	batch03/PM	2	203873328	203873370	42	DEL	PMGBM238/	1361661	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:71:60:19	split	q33.2	CTLA4		NM_001037(	203867770
109	batch03/PM	2	203873328	203873370	42	DEL	PMGBM263/	1173583	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:2:60:40:1	split	q33.2	CTLA4		NM_001037(	203867770
110	batch03/PM	2	203873328	203873370	42	DEL	PMGBM264/	1118199	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:21:60:23	split	q33.2	CTLA4		NM_001037(	203867770
111	batch03/PM	2	203873328	203873370	42	DEL	PMGBM340/	1065026	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	1/1:12:60:45	split	q33.2	CTLA4		NM_001037(	203867770
112	batch03/PM	2	203873328	203873370	42	DEL	PMGBM360/	1363870	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:25:60:40	split	q33.2	CTLA4		NM_001037(	203867770
113	batch03/PM	2	203873328	203873370	42	DEL	PMGBM584/	1375107	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:11:57.76	split	q33.2	CTLA4		NM_001037(	203867770
114	batch03/PM	2	203873328	203873370	42	DEL	PMGBM660/	1483457	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:46:60:30	split	q33.2	CTLA4		NM_001037(	203867770
115	batch03/PM	2	203873328	203873370	42	DEL	PMGBM913/	1737930	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:31:60:11	split	q33.2	CTLA4		NM_001037(	203867770
116	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1298196	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:8:60:47:1	split	q33.2	CTLA4		NM_001037(	203867770
117	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1173079	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	1/1:0:60:52:1	split	q33.2	CTLA4		NM_001037(	203867770
118	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1506256	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:46:60:28	split	q33.2	CTLA4		NM_001037(	203867770
119	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1977139	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:68:59.83	split	q33.2	CTLA4		NM_001037(	203867770
120	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	2647915	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:48:60:30	split	q33.2	CTLA4		NM_001037(	203867770
121	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000C	1147264	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:11:60:38	split	q33.2	CTLA4		NM_001037(	203867770
122	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000D	2093887	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:30:60:25	split	q33.2	CTLA4		NM_001037(	203867770
123	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000D	1456664	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:17:60:16	split	q33.2	CTLA4		NM_001037(	203867770
124	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000D	1359191	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:48:60:30	split	q33.2	CTLA4		NM_001037(	203867770
125	batch03/PMI	2	203873328	203873370	42	DEL	PMLBM000D	1943939	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:30:60:24	split	q33.2	CTLA4		NM_001037(	203867770
126	batch03/PMI	2	203873328	203873360	32	DEL	PMLBM000D	1528951	C	<DEL>	.	PASS	SVMETHOD=	GT:GQ:MAP(	0/1:106:60:9	split	q33.2	CTLA4		NM_001037(	203867770

Figure 9. There is a 42 base pair deletion on chromosome common for a population and is therefore found in many different samples, making it not clinically significant.

BED file

The TSV produced after the previous step contains only SVs which are worth checking in IGV. To make the process of verifying the events in IGV faster, a BED file is produced based on the events of interest. This script is made in Python (Appendix 1) and makes a BED file based on an AnnotSV TSV file. The number of base pairs, N, left- and right of the breakpoints can be specified. N is set at 10 for the pipeline. A few base pairs left- and right of the indicated breakpoints can give a better overview of the event when checking in IGV.

Bitbucket

Repository with all required scripts for the pipeline is located on Bitbucket:
https://bitbucket.org/princessmaximacenter/pmc_kuiper_projects/src/master/svpipeline/

Results

This section of the report presents the outcomes generated throughout the project, detailing the selection process for SV callers and thresholds. Additionally, it provides an analysis of the results obtained from three pipeline executions: one with property thresholds implemented, another without property thresholds, and a third using the Hemato gene panel.

Assessing SV caller's and SURVIVOR's performance

To assess the performance of each SV caller in comparison to the other, all five callers identify SVs in the same sample. The overlap between these calls gives information how the caller performs. To determine overlapping calls, the produced VCFs are merged. Next, ensure each SV found by multiple callers appears only once in the VCF with an annotation of detecting caller(s). This is

Argument		**Value used for assessing Caller's performance**
1	File with VCF names and paths	
2	Maximum distance between breakpoints	1000
3	Minimum number of supporting callers	1
4	Take type of SV into account (1==yes, else no)	0
5	Take strand of SV into account (1==yes, else no)	0
6	Disabled.	0
7	Minimum size of SVs to be considered	0
8	Output VCF name and path	

Table 2. As the first argument, SURVIVOR needs a file with the VCF names and paths to them. The second command is the maximum distance between the breakpoints when it merges SVs from different VCFs. Not every tool will give the exact same breakpoint locations, so this number sets the maximum distance when it calls a mutation as the same or if you want to see them as separate SVs. This value is set at 1000 bp, the recommended length by the developers. This length works well for different datasets (Sadlazeck, 2021) The third argument is the minimum number of supporting callers. If we set the minimum number of supporting callers at 1, all found SVs by all the callers will be written in the new VCF file. If this argument is set at 5, only the SVs which are found by all five callers will be written in the new VCF file. The fourth argument is if the type of SV should be considered. If within the given maximum distance between the breakpoints different callers call an SV but it is a different SV type, should this be merged into one SV in the newly produced VCF. This value is set at 0 since the type of SV is not that important and distinguishing differences between insertions and duplications for example can be hard. Hence, it is set at 0. The fifth argument is if the strand should be considered when merging SVs is set at 0. This is set at 0 because this flag can be hard to parse, according to the developer (Sadlazeck, 2021). The sixth argument is disabled in SURVIVOR 1.0.7. The seventh argument is the minimum size of the SVs to be considered. If SVs are smaller than this given length, they will not be written in the new VCF. Minimum length is set at 0 since all SVs are interesting at this point. And the final, eighth, argument is the output VCF name and path.

facilitated with SURVIVOR. SURVIVOR can merge VCFs from different SV callers efficiently. SURVIVOR has multiple options when merging SVs, which need to be specified in the arguments (Table 2).
The testing of SURVIVOR and assessing the performance of the callers (Table 2) is done on a single WGS germline sample from the ROHHAD study.

Figure 10 describes the overlap between the calls, 1820 SVs are found by all five SV callers. What is also immediately noticeable is the difference in the number of SVs called per caller. Dysgu calls more than 16x the number of SVs compared to Manta (Table 3).

	DEL	DUP	INS	TRA	INV	BND	SGL	TOTAL
Manta	4293	589	1755		394	766		7797
GRIDSS2	9160	1472	1002		466	2336	4294	18730
Delly	11642	8887	82		54110	15131		89852
Lumpy	5239	1685			408	16482		23814
Dysgu	19203	2442	26508	30692	48831			127676

Table 3. Presentation of the number of SVs identified by five different callers in a single ROHHAD germline WGS sample. It is important to note that calls classified as breakends (BND) or single breakend SV support (SGL) are not of interest in the context of identifying clinically significant SVs.

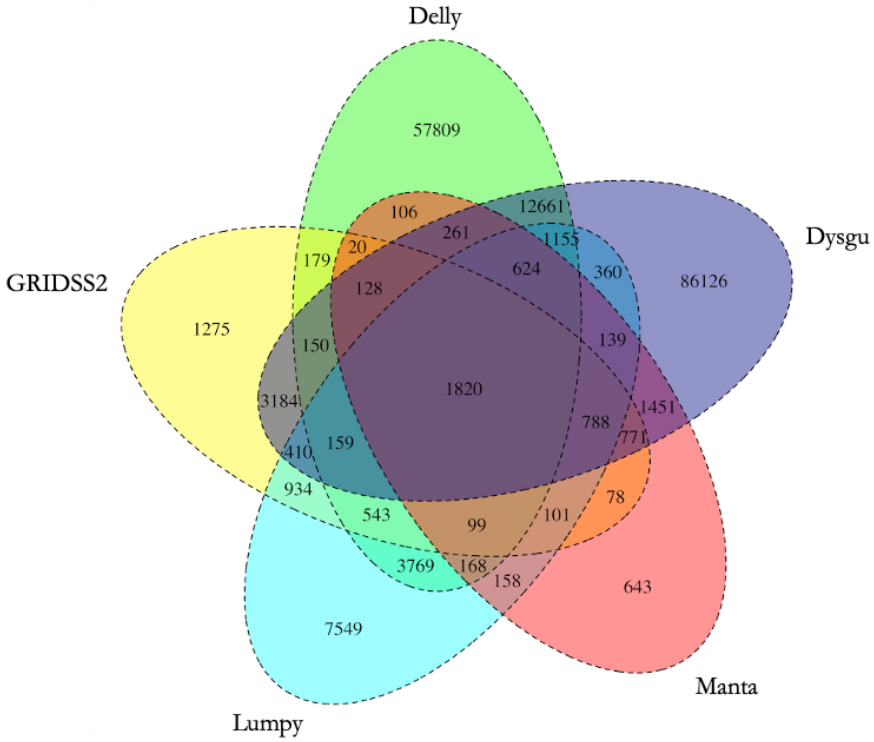


Figure 10. This Venn Diagram illustrates the quantity and overlap of calls made five SV callers. Noticeable is the variation in the number of calls each caller makes.

Determining property thresholds

To accurately determine a threshold for Manta's quality score and Dysgu's probability score for deletions in the VCF files, more verified events are required. The same ROHHAD sample as used for Figure 3 & 4 is used. Dysgu- and Manta calls get merged and only SVs which are found by both callers are verified. Verified calls found by a single caller are not used for the merged set. When thresholds are applied, correctly- and incorrectly called SVs are lost (Table 4 & 5). Optimal determination for the thresholds of the SV caller's properties rely on the specified percentage of True Positives (TP) to be identified. To illustrate, if the interest is to find at least 90% of the TP SVs in the data,

258 * 0.90 = 232.2

Finding ~90% of the true SVs in this dataset means finding ~232 true SVs.

Using Dysgu, to find 232 true SVs a probability threshold of 0.65 is suited (Table 4). This results in finding 233 TP calls and 163 FP calls. Consequently, losing ~10% of the correct SVs and losing ~45% of the incorrectly called SVs.

Dysgu Prob.	**0**	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**	**0.55**	**0.60**	**0.65**	**0.70**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
True	258	256	255	254	253	253	252	252	251	249	246	242	239	233	217	189	138	117	74	23
False	298	294	281	266	260	247	237	228	221	214	205	191	176	163	130	81	31	13	3	0

Table 4. Shown how many correctly called- (True) and incorrectly called (False) SVs are found for different thresholds for Dysgu's Probability Score.

Manta Quality	**0**	**50**	**100**	**150**	**200**	**250**	**300**	**350**	**400**	**450**	**500**	**550**	**600**	**650**	**700**	**750**	**800**	**850**	**900**	**950**
True	258	258	258	253	249	242	234	220	214	199	184	165	151	139	131	118	106	86	78	73
False	298	274	246	224	206	186	158	136	109	99	86	72	51	45	32	27	20	19	12	8

Table 5. Shown how many correctly called- (True) and incorrectly called (False) SVs are found for different thresholds for Manta's Quality Score.

To find at least 232 true SVs using Manta, a quality score threshold could be set at 300. This results in finding 234 TP and 158 FP (Table 4). To put in percentages, losing ~10% of the correct SVs and ~47% of the incorrectly called SVs.

If Dysgu and Manta are merged, around the same percentage of correctly called SVs can be found but with less incorrectly called SVs. When combining calls filtered on Dysgu probability 0.55 and Manta quality 200, 231 TP- and 139 FP calls are found (Appendix 1). To put in percentages, ~10% of the TP calls and ~53% of the FP are lost in this dataset. Hence, combining Dysgu and Manta can lead to finding the same percentage of correctly called SVs with less incorrectly called SVs.

If there are a lot of SVs in the data, the percentage FP found can be too high when trying to find 90% of the TP calls. If the interest is therefore to find 70% of the TP, the following thresholds can be set,

258 * 0.70 = 180.6

To find around 181 correct SVs for Dysgu, a threshold of 0.75 is appropriate. With this threshold, 189 correctly called SVs and 81 incorrectly called SVs are found (Table 4). To put this in percentages: losing ~27% of the correctly called SVs and losing ~73% of incorrectly called SVs.

To find around 181 correct SVs for Manta, a threshold of around 500 can be used. With this threshold, 184 correctly called SVs and 86 incorrectly called SVs are found (Table 5). To put this in percentages: losing ~29% of the correctly called SVs and losing ~71% of the incorrectly called SVs.

Combining Manta and Dysgu leads again to finding less incorrectly called SVs while finding around the same percentage of correct calls. Combining Manta's quality threshold of 400 with Dysgu's probability score threshold of 0.70 leads to a dataset with 182 correctly called SVs and 50 incorrectly called SVs. To put in percentages, losing ~29% of correctly called SVs and losing 83% of incorrectly called SVs. Consequently, combining SV callers lead to the less falsely called SVs for the same amount of correctly called SVs (Fig. 6).

In the PrediCT dataset, it is crucial to identify nearly all SVs. For that reason, a Dysgu probability threshold of 0.55 and Manta quality threshold of 200 is suited, resulting in finding ~90% of SVs. For the pipeline, Dysgu's probability threshold is set at 0.54 instead of 0.55. This is due to one of the already verified SVs in the PrediCT data having a Dysgu probability score of 0.54 and this SV should be in the final output.

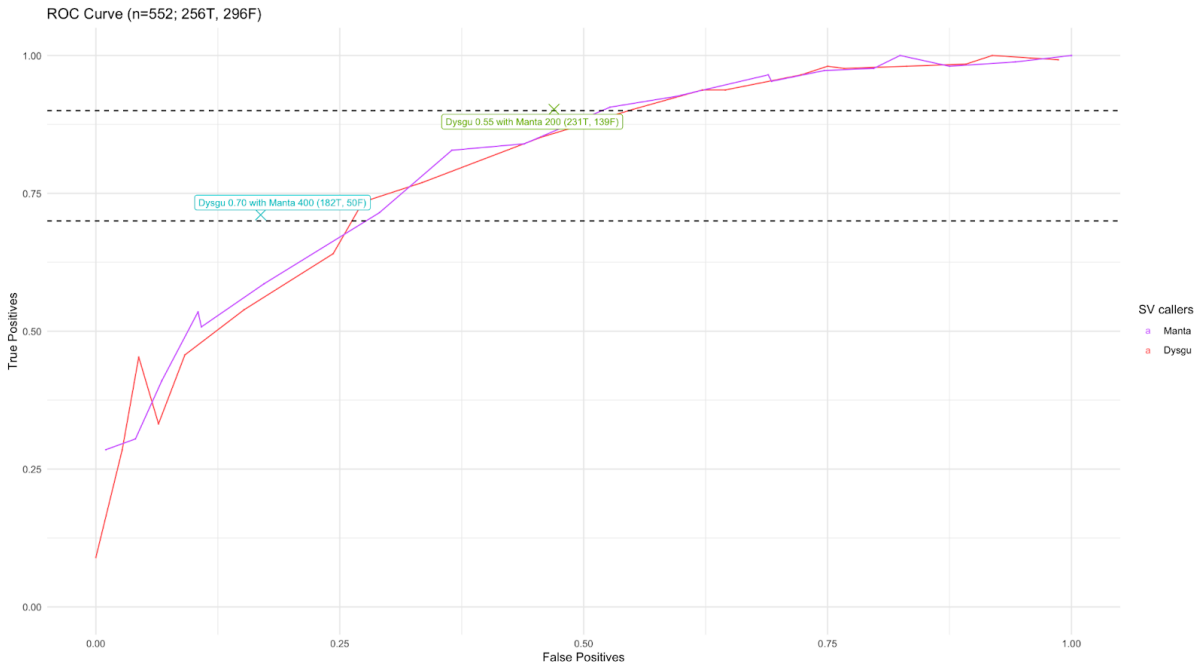


Figure 11. Manta's- and Dysgu's precision are shown in purple and red. The dotted lines are respectively found 70% and 90% of the True Positives (TP). In the figure can be seen that combining Manta and Dysgu results in a better dataset. For the same percentage TP found, fewer False Positives (FP) are found when combining callers compared to using a single caller.

Results with thresholds

The pipeline (Fig. 1) is executed using thresholds of 0.54 for Dysgu's probability and 200 for Manta's quality, aiming for identifying 90% of the SVs in the data (Appendix 1).

The evaluation in IGV, along with explanations for why SVs are inaccurately or accurately identified, is included in the Appendix. For cases where the assessment was more complex, the details of these events are discussed in this section of the chapter.

In total, 4606 SVs are called in all PrediCT patients combined, 247 are in exons. The events which are not common in the population are shown in Table 6.

An SDHB exon 3 deletion in PrediCT196 is found by Dysgu and Manta. However, this event was already previously identified.

A FANCA deletion in PrediCT878 is found by Dysgu and Manta. However, this event was already identified.

PrediCT745 contains a deletion, eliminating parts of the genes CEP57 and MTMR2 (Appendix 6). This event is found by Dysgu and Manta. CEP57 is a gene that is in the PrediCT gene panel, and therefore interesting. The patient has Non-Hodgkin lymphoma (NHL). The PrediCT study used in their research Whole Exome Sequencing (WXS) data (Appendix 7). This mutation is not reported in the VCF that was earlier produced. The deletion does not cause a frameshift since the end part is removed. A mutation frequently observed in the population occurs in close proximity to this

specific structural variant (UCSC, n.d.). Considering that there is a deletion more common at this location, it is not likely this is a causation for NHL.

Sample	DEL length	Gene	Tumour type	Assessment	Appendix
PrediCT878	172679 bp	FANCA	ALL	Correct	
PrediCT196	7904 bp	SDHB	Feochromocytoma	Correct	
PrediCT177	91 bp	RTEL1	Wilms	Incorrect	5
PrediCT745	1300 bp	CEP57	NHL	Correct but likely not clinically significant	6
PrediCT403	36 bp	ETV6	NBL	Correct but likely not clinically significant	8

Table 6. SVs found by the pipeline in all PrediCT samples using a threshold of 0.54 for Dysgu's Probability and 200 for Manta's Quality which are in an exon and not common in a population.

PrediCT403 has a deletion in ETV6, one of the cancer predisposition genes (Appendix 8) and is found by Dysgu. There are no reads at this location in the WXS, and thus not found in the VCF. There are no reads at this location in the WXS because the bed file that is used, "KAPA_HyperExome_hg38_capture_targets.bed", does not have this location in the file. Hence, it is not found in the WXS data and the VCF. PrediCT403 is diagnosed with a Neuroblastoma (NBL). The deletion has a length of 36 base pairs, meaning there is no frameshift. In literature, there are no mentions about ETV6 segment deletion and NBL. Considering that there is no frameshift and no information in literature about NBL and ETV6 segment deletion, this deletion is likely not the cause for NBL.

The threshold combination using Dysgu's probability of 0.54 and Manta's Quality of 200 resulted in 80% correctly called events and no correct SVs missed when these results are compared with the pipeline execution without thresholds.

Results without thresholds

For the PrediCT cohort, it is to identify all the interesting SVs. Hence, the pipeline is run with no thresholds for Dysgu's probability- and Manta's quality score. The SV length filtered is applied with a maximum length of 200000 base pairs. The SVs found using thresholds for Dysgu's Probability of 0.54 and Manta's Quality Score of 200 are identified again, however more uniquely called SVs are in the output (Table 7).

The evaluation in IGV, along with explanations for why SVs are inaccurately or accurately identified, is included in the Appendix. For cases where the assessment was more complex, the details of these events are discussed in this section of the chapter.

In total 5167 SVs are called, 429 of those are in an exon and 13 are unique events. The unique events are in Table 7.

Sample	DEL length	Gene	Tumour type	Assessment	Appendix
PrediCT878	172679 bp	FANCA	ALL	Correct	
PrediCT196	7904 bp	SDHB	Feochromocytoma	Correct	
PrediCT177	91 bp	RTEL1	Wilms	Incorrect	5
PrediCT745	1300 bp	CEP57	NHL	Correct but likely not clinically significant	6
PrediCT403	36 bp	ETV6	NBL	Correct but likely not clinically significant	8
PrediCT242	1045	RPL11	ALL	Incorrect	9
PrediCT242	433	RPL11	ALL	Incorrect	9
PrediCT242	2691	RPL35A	ALL	Incorrect	9
PrediCT242	3304	RPS24	ALL	Incorrect	9
PrediCT242	125	EIF4G2	ALL	Incorrect	9
PrediCT738	26673	PAX5	ALL	Incorrect	10
PrediCT138	154090	CREB3L3	ALL	Incorrect	11
PrediCT917	89363	DDB2	NHL	Incorrect	12

Table 7. SVs which are in an exon and not common in a population found by the pipeline using only a filter on SV length in all PrediCT samples.

In PrediCT138 is a 154090 base pairs deletion called (Appendix 11). The two identified breakpoints are correct. However, it is not likely that there is a deletion between these breakpoints. If the genes within these breakpoints are deleted, this could be seen in the CNV plot (Appendix 3). In this plot, there is no clear CNV seen at chromosome 19 around 4MB. Therefore, it is not likely this deletion is correctly identified.

Using no threshold for Dysgu's Probability and Manta's Quality score resulted in 31% correctly called events.

Hemato panel

The "Hemato gene panel" is PrediCT's second gene panel. While this panel contains a large number of genes, there are only 37 samples for which proper consent and WGS data is available. The pipeline was executed using no thresholds for Dysgu's Probability and Manta's quality. There was a SV length filter used with a maximum length of 200000 base pairs. In total, 103 SVs are called, 52 of those in an exon and 3 are unique called events.

Three uniquely called SVs were identified using no thresholds for Dysgu's Probability and Manta's Quality Score, two of the events were identified with the threshold filter (Table 8).

The called event in PrediCT738was already found with PrediCT's other gene panel (Appendix 10).

Sample	DEL length	Gene	Tumour type	Assessment	Appendix	Identified with Dysgu 0.54 and Manta 200 filter
PrediCT738	26673	PAX5	ALL	Incorrect	10	No
PrediCT160	83342		ALL	Incorrect	13	Yes
PrediCT347	78	GP1BA	ALL	Correct but likely not clinically significant	14	Yes

Table 8. SVs which are in an exon and not common in a population for PrediCT samples with proper consent for the Hemato gene panel.

The second called event is found in PrediCT160 (Appendix. 13). This event is an 83342 base pairs deletion on chromosome 11. The event is called between 64195495 and 64278841 base pairs. Chromosome 11 spans around 135 million base pairs (MedlinePlus, 2016). This means the deletion should occur approximately in the central region of the chromosome. The CNV data (Appendix 4) shows no indication of a deletion at this location. Hence, the called deletion is likely incorrect.

The third called event is in PrediCT347 (Appendix 14). This deletion in the GP1BA gene has a length of 78 base pairs. Hence, the deletion does not cause a frameshift. The patient is diagnosed with Acute Lymphocytic Leukaemia (ALL). In literature, there are no mentions about GP1BA segment deletion and ALL. Considering that there is no frameshift and no information in literature about ALL and GP1BA segment deletion, this deletion is likely not the cause for ALL.

Conclusion

There are many different SV callers developed over the last years. None of these callers performs with well enough for accurate and comprehensive detection of SVs. For that reason, it is recommended to combine different callers for a more accurate and comprehensive detection of SVs (Koboldt, 2020) (Kuzniar et al, 2020) (Kosugi et al, 2019). Manta and Dysgu are chosen as callers for the reasons they both contain a property useful for indicating if an event is correctly called. Manta has the property Quality Score and Dysgu has the property Probability Score. The second reason these SV callers are chosen is for the reason they can process CRAM files. Combining the two SV callers show improved results over using them as a single caller (Fig. 11). For the same amount of correctly found SVs, less incorrectly called SVs were found when combining Dysgu and Manta. This is tested for finding 70%- and 90% of the correct SVs in the dataset.

When the interest is finding at least 70% of the correct SVs in the data, a Dysgu probability threshold of 0.70 and Manta quality threshold of 400 is recommended (Appendix 1). When the interest is finding at least 90% of the correct SVs in the data, a Dysgu probability threshold of 0.55 and Manta quality threshold of 200 is recommended (Appendix 1). Hence, the determination of the combination thresholds relies on identifying the minimum number of SVs.

The established thresholds for Dysgu's Probability at 0.54 and Manta's Quality Score at 200 are functioning effectively. The set of SVs identified using these thresholds show improved accuracy compared to the subset identified without any thresholds for the two properties. The correctly called SVs are in both sets, however, the set with the thresholds applied contained less incorrectly called SVs.

The research question is: "Can we identify constitutional small structural variants of clinical significance in cancer predisposition genes in children with cancer?". In the analysis, the pipeline was unable to detect any clinically significant SVs beyond those two that were already established.

Discussion

During this project, no clinically significant SVs that had not been previously identified are discovered. It was expected to identify three clinically significant SVs since around 1% of the tumours are the result of a germline SV in a cancer predisposition gene. This suggests that either there are no tumours in this cohort caused by SVs in cancer predisposition genes, or SVs are not being detected by the current pipeline. Alternatively, it is possible that samples potentially containing an SV in a cancer predisposition gene were of insufficient quality, leading to not getting detected during sequencing. Short read sequencing is used as sequencing method, while long read sequencing might reveal more SVs in the data since it is better for SV detection.

The analysis solely focused on identifying deletions, which represents merely one category of SV. However, it is possible that another type of SV in a cancer predisposition gene might be present in the data and could be the underlying cause of the tumour.

Manta uses mapped paired-end sequencing reads to call SVs (Saunders, 2018), Dysgu is programmed to work with paired-end reads as well (Cleal, 2018). GRIDSS2 uses positional de Bruijn graph assembly, perhaps this technique could identify SVs which are not found now.

Dysgu incorporates a range of artificial intelligence modules and has called more SVs which are common in the population compared to Manta. This detection might be the result of the AI elements that have been trained to recognize these specific population-common SVs.

The SVs identified in PrediCT403 and PrediCT347 are labelled as incorrect. However, more research must be done to verify that the deletion, even though there is no frameshift, is not a reason to lead to tumour development.

When testing how well SV callers perform, it is very difficult to know how many true SVs are in the data but are missed by the callers. Therefore calculating the real specificity is not possible. Only SVs that are found by at least one tool, and checked, can be used for determining how well callers perform. When at least one caller found an SV and this SV is checked and labelled as true, it can be used to see if other tools found this SV as well. This problem can be resolved by using data where every SV is known. If you know how many -and which SVs are in the data, you can measure how well the caller performs and also identify which SVs are missed by all callers. The problem with this method is however that finding real data for which a “true” vcf exists is very difficult. There are “artificial SV generators” but the problem with these generators is that this data might be very different from real data, and therefore the performance of the caller might be different on fake data compared to real data.

In this project, numerous SVs are labelled as true or false, forming the basis for following decisions. These categorizations were determined through a careful examination of SVs using IGV. However, it is important to note potential inaccuracies, given the complexity of certain SVs. Deletions with clear breakpoints but minimal coverage drop or distinct coverage drops without identifiable breakpoints present difficulties. Therefore, when using these labels, it’s important to recognize the sensitivity to human error, and the accuracy may not be perfect.

When checking SVs in IGV, it was noticeable that many of the falsely called SVs are in bad sequenced repetitive regions. We are looking for SVs of clinical significance, these are most likely

in exons. Many repeats (~89.5%) are assumed to be non-functional because they are located in introns (Liao et al, 2023). When looking at SVs in regions of interest, a higher percentage is true since regions of interest are not in repetitive regions, where many of the falsely called SVs are.

To make the best choices for deciding a threshold for the properties of different VCF files, it seemed logical at first to take SVs from the called-by-five-callers group. This is because a single labelled SV could be used five times, one time for each tool. However, because an SV is called by five different callers, it is much more likely to be true than if it is called by only one caller. This led to the fact that many of the verified calls came from the called-by-five-callers group. Resulting in a sample group where true-labelled-SVs are “very true”. So when looking for thresholds for properties, thresholds are for the “obviously true” SVs. With obviously true deletions, deletions with close to perfect breakpoints are meant and homozygous deletions with good breakpoint indications. Even though in the data are more deletions, which are also true, but less clear. These have less total reads around the mutation or are in a less well sequenced region. This results in lower confidence properties. When in the true- and false labelled groups are only these high confidence mutations, the set is too biassed. Resulting in too many correct SVs which will be filtered out. To reduce the effect of these high-confidence-SVs in the set, SVs which were called by less callers are verified and added to labelled set.

To establish the thresholds for the pipeline, evaluation of SVs was conducted using data from a single sample (Appendix 2). The use of a single sample is a requirement due to the necessity of merging calls generated by Dysgu and Manta. The merging process requires that the SVs originate from the same sample to ensure meaningful consistency. This is due to the fact that the rationale behind merging SVs lies in identifying mutations where both SV callers independently detect the same SV. When dealing with VCF files containing SVs from multiple samples, a potential issue arises. The issue comes to life when one caller identifies an SV on chromosome X at position X for one sample, and another caller detects an SV at the same chromosome and position but for a different sample, merging these SVs would cause flawed data merging. To avoid the risk of this happening, all SVs were exclusively verified from a single sample during the threshold determination process. Consequently, the thresholds established during this process may be too tailored to the characteristics of the sample chosen. It is important to acknowledge that if a different sample were chosen, results might show some differences. This leads to cautious interpretation of the thresholds, recognizing potential sample-specific influences on the determined thresholds.

The total amount of checked SVs in Table 4 & Table 5 compared to Appendix 2 is different. When merging the VCFs, SURVIVOR did not recognize for four SVs they should be merged.

The order of the different steps in the pipeline is important. Especially the step for applying the gene panel filter before merging the different VCFs. The reason this is important is because after merging the VCFs, only one of the coordinates is kept. The merging distance is set at 1000 bp, this means that SVs are merged if they are within 1000 bp from each other. So it is possible that the location of a SV changes from within the gene panel to outside of the gene panel, or the other way around. To be sure that we only look at SVs that we are sure of are within the gene panel, this step is applied early in the pipeline.

Manta’s quality score gives an indication of a call being true or not (Fig. 4). The maximum quality score Manta gives is 999. When looking at calls made by Manta which have a quality score of 999,

there are many large SVs (>100000 bp) which have a quality score of 999 but are, however, not correct. Using the quality score of Manta to predict if an SV is correct, works better with smaller SVs. When going through the Manta VCFs there were too many large SVs with a quality score of 999. For that reason, the step of the pipeline where calls made by a single caller which have a high quality score are added uses the VCFs where there is already filtered on SV length. By doing this, the really large SVs with a quality score of 999 are not in the annotated files.

On reflection, considerable time was devoted to selecting the optimal combination of SV callers for the pipeline. The tests consistently showed the same result: more callers enhanced precision but at the cost of losing some true calls. The search for the "perfect combination" continued, even though this does not even exist. Subsequently, when implementing Manta and Dysgu, the search for the "perfect thresholds" began, taking more time than necessary. Throughout the project, there was an assumption that identifying the ideal combination of tools and thresholds would optimise the pipeline's performance. However, this assumption overlooked the variability in performance across different samples, SV types, and perhaps variables like sequence depth. In hindsight, a more efficient approach would involve spending less time on SV caller(s) and threshold(s) selection, but instead relying on results to make adjustments based on the current performance of the pipeline.

For future projects of similar nature, using AI would be in consideration. A substantial portion of the project involved manually labelling SVs as true or false, essentially generating training data for an AI system. In this process, VCF file entries were marked based on manual assessment of their properties to recognize the likelihood of accuracy. An AI, given sufficient training data, has the potential to perform this task more effectively. Exploring the development and testing of an AI for this purpose could be very promising. This approach of using AI to filter VCF files, might be a viable solution in the future for this task.

References

Abel, H. J., Larson, D. E., Regier, A. A., Chiang, C., Das, I., Kanchi, K. L., ... & Hall, I. M. (2020). Mapping and characterization of structural variation in 17,795 human genomes. *Nature*, *583*(7814), 83-89.

Amarasinghe, S. L., Su, S., Dong, X., Zappia, L., Ritchie, M. E., & Gouil, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. *Genome biology*, *21*(1), 1-16.

Cameron, D. L., Baber, J., Shale, C., Valle-Inclan, J. E., Besselink, N., van Hoeck, A., ... & Papenfuss, A. T. (2021). GRIDSS2: comprehensive characterisation of somatic structural variation using single breakend variants and structural variant phasing. *Genome biology*, *22*, 1-25.

Cameron, D. L., Schröder, J., Penington, J. S., Do, H., Molania, R., Dobrovic, A., ... & Papenfuss, A. T. (2017). GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. *Genome research*, *27*(12), 2050-2060.

Chen, X., Schulz-Trieglaff, O., Shaw, R., Barnes, B., Schlesinger, F., Källberg, M., ... & Saunders, C. T. (2016). Manta: rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics*, *32*(8), 1220-1222.

Cleal, K. (2023, October 18). Dysgu. GitHub. https://github.com/kcleal/dysgu

Cleal, K., & Baird, D. M. (2022). Dysgu: efficient structural variant calling using short or long reads. *Nucleic Acids Research*, *50*(9), e53-e53.

Collins, F. S., & Fink, L. (1995). The human genome project. *Alcohol health and research world*, *19*(3), 190.

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., ... & Li, H. (2021). Twelve years of SAMtools and BCFtools. *Gigascience*, *10*(2), giab008.

English, A. C., Menon, V. K., Gibbs, R. A., Metcalf, G. A., & Sedlazeck, F. J. (2022). Truvari: refined structural variant comparison preserves allelic diversity. *Genome Biology*, *23*(1), 271.

Finn, E. (2013, October 28). What is “Segmentation fault (core dumped)?”. Stack overflow. https://stackoverflow.com/questions/19641597/what-is-segmentation-fault-core-dumped

Geoffroy, V., Guignard, T., Kress, A., Gaillard, J. B., Solli-Nowlan, T., Schalk, A., ... & Muller, J. (2021). AnnotSV and knotAnnotSV: a web server for human structural variations annotations, ranking and analysis. *Nucleic acids research*, *49*(W1), W21-W28.

Geoffroy, V., Herenger, Y., Kress, A., Stoetzel, C., Piton, A., Dollfus, H., & Muller, J. (2018). AnnotSV: an integrated tool for structural variations annotation. *Bioinformatics*, *34*(20), 3572-3574.

Geoffroy, V., Lamouche, J. B., Guignard, T., Nicaise, S., Kress, A., Scheidecker, S., ... & Muller, J. (2023). The AnnotSV webserver in 2023: updated visualization and ranking. *Nucleic Acids Research*, gkad426.

Holmfeldt, L., Wei, L., Diaz-Flores, E., Walsh, M., Zhang, J., Ding, L., ... & Mullighan, C. G. (2013). The genomic landscape of hypodiploid acute lymphoblastic leukemia. *Nature genetics*, *45*(3), 242-252.

Jeffares, D. C., Jolly, C., Hoti, M., Speed, D., Shaw, L., Rallis, C., ... & Sedlazeck, F. J. (2017). Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nature communications*, *8*(1), 14061.

Koboldt, D. C. (2020). Best practices for variant calling in clinical sequencing. *Genome Medicine*, *12*(1), 1-13.

Kosugi, S., Momozawa, Y., Liu, X., Terao, C., Kubo, M., & Kamatani, Y. (2019). Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome biology*, *20*, 1-18.

Kuzniar, A., Maassen, J., Verhoeven, S., Santuari, L., Shneider, C., Kloosterman, W. P., & de Ridder, J. (2020). sv-callers: a highly portable parallel workflow for structural variant detection in whole-genome sequence data. *PeerJ*, *8*, e8214.

Layer, R. M., Chiang, C., Quinlan, A. R., & Hall, I. M. (2014). LUMPY: a probabilistic framework for structural variant discovery. *Genome biology*, *15*(6), 1-19.

Liao, X., Zhu, W., Zhou, J., Li, H., Xu, X., Zhang, B., & Gao, X. (2023). Repetitive DNA sequence detection and its role in the human genome. *Communications Biology*, *6*(1), 954.

Mahmoud, M., Gobet, N., Cruz-Dávalos, D. I., Mounier, N., Dessimoz, C., & Sedlazeck, F. J. (2019). Structural variant calling: the long and the short of it. *Genome biology*, *20*(1), 1-14.

MedlinePlus. (2016, May 1). Chromosome 11. MedlinePlus. https://medlineplus.gov/genetics/chromosome/11/#:~:text=Chromosome%2011%20spans%20about%20135,the%20total%20DNA%20in%20cells.

Nurk, S., Koren, S., Rhie, A., Rautiainen, M., Bzikadze, A. V., Mikheenko, A., ... & Phillippy, A. M. (2022). The complete sequence of a human genome. *Science*, *376*(6588), 44-53.

Rahman, N. (2014). Realizing the promise of cancer predisposition genes. *Nature*, *505*(7483), 302-308.

Rausch, T., Zichner, T., Schlattl, A., Stütz, A. M., Benes, V., & Korbel, J. O. (2012). DELLY: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics*, *28*(18), i333-i339.

Robinson, J. T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E. S., Getz, G., & Mesirov, J. P. (2011). Integrative genomics viewer. *Nature biotechnology*, *29*(1), 24-26.

Saunders, C. (2018, August 14). Manta Structural Variant Caller. GitHub. https://github.com/Illumina/manta/blob/master/README.md

Sadlazeck, F. (2021, July 10). Details of merge options. GitHub. https://github.com/fritzsedlazeck/SURVIVOR/issues/144

Sadlazeck, F. (2023, October 4). Segmentation fault (core dumped) during SURVIVOR merge. GitHub. https://github.com/fritzsedlazeck/SURVIVOR/issues/176#issuecomment-1748328946

Sedlazeck, F. J., Dhroso, A., Bodian, D. L., Paschall, J., Hermes, F., & Zook, J. M. (2017). Tools for annotation and comparison of structural variation. *F1000Research*, *6*.

Shaikh, T. H. (2017). Copy number variation disorders. *Current genetic medicine reports*, *5*, 183-190.

UCSC (n.d.). hg38 Database of Genomic Variants: Gold Standard Variants (gsslV20573). UCSC. https://genome.ucsc.edu/cgi-bin/hgc?hgsid=1794904506_09B4nB5DQDrflNeeYgbIXxkgfbIm&db=hg38&c=chr11&l=95831338&r=95835268&o=95832627&t=95833966&g=dgvGold&i=gssvL20573

van Belzen, I. A., Schönhuth, A., Kemmeren, P., & Hehir-Kwa, J. Y. (2021). Structural variant detection in cancer genomes: computational challenges and perspectives for precision oncology. *NPJ Precision Oncology*, *5*(1), 15.

van Campen, J. (2022, September 2). Short-read sequencing. GeNotes. https://www.genomicseducation.hee.nhs.uk/genotes/knowledge-hub/short-read-sequencing/#:~:text=Short%2Dread%20sequencing%20is%20currently,300%20bases)%20before%20being%20sequenced

Wang, Q. (2016). Cancer predisposition genes: molecular mechanisms and clinical impact on personalized cancer care: examples of Lynch and HBOC syndromes. *Acta Pharmacologica Sinica*, *37*(2), 143-149.

Wittler, R., Marschall, T., Schönhuth, A., & Mäkinen, V. (2015). Repeat-and error-aware comparison of deletions. *Bioinformatics*, *31*(18), 2947-2954.

Appendix

Appendix 1

Scripts can be found in the repository on Bitbucket:
https://bitbucket.org/princessmaximacenter/pmc_kuiper_projects/src/master/svpipeline/CODE/

Script	Function
generateSVpipelineBatchJobs.sh	SV pipeline which can run samples parallel using scheduler.
generateSVpipelineSettings.sh	Creates generateSVpipelineBatchJobs scripts with a variety of different thresholds combinations for Manta's Quality- and Dysgu's Probability Score.
dysguVCFfilterBySVLEN.py	Filter Dysgu's VCF by retaining only SVs that are smaller than the specified integer.
mantaVCFfilterBySVLEN.py	Filter Manta's VCF by retaining only SVs that are smaller than the specified integer.
dysguFilter.py	Filter Dysgu's VCF by retaining only SVs that have a Probability Score value larger than the specified value.
mantaFilter.py	Filter Manta's VCF by retaining only SVs that have a Quality Score value larger than the specified value.
annotSVexonInformationToUniques.py	Extract non-population common events from AnnotSV TSV.
makeBedfileFromAnnotSV_regularFormat.py	Create BED file based on AnnotSV TSV for original format VCFs
makeBedfileFromAnnotSV_SURVIVORformat.py	Create BED file based on AnnotSV TSV for SURVIVOR format VCFs.
extractExonInformationFromAnnotSV.sh	Extracts exon regions and filters for unique SVs from an AnnotSV file.
createMantaArtificialVCF.py	Create artificial deletions in Manta's VCF format.
SURVIVORvcfToOriginalVCF_Manta.py	Takes an original VCF and a SURVIVOR merged VCF as input and gives only the SURVIVOR merged calls in original Manta VCF format. There is also an option to add TRUE / FALSE verifications to the txt format if these are added to the SURVIVOR VCF in the last column.
SURVIVORvcfToOriginalVCF_Dysgu.py	Takes an original VCF and a SURVIVOR merged VCF as input and gives only the SURVIVOR merged calls in original Dysgu VCF format. There is also an option to add TRUE / FALSE verifications to the txt format if these are added to the SURVIVOR VCF in the last column.

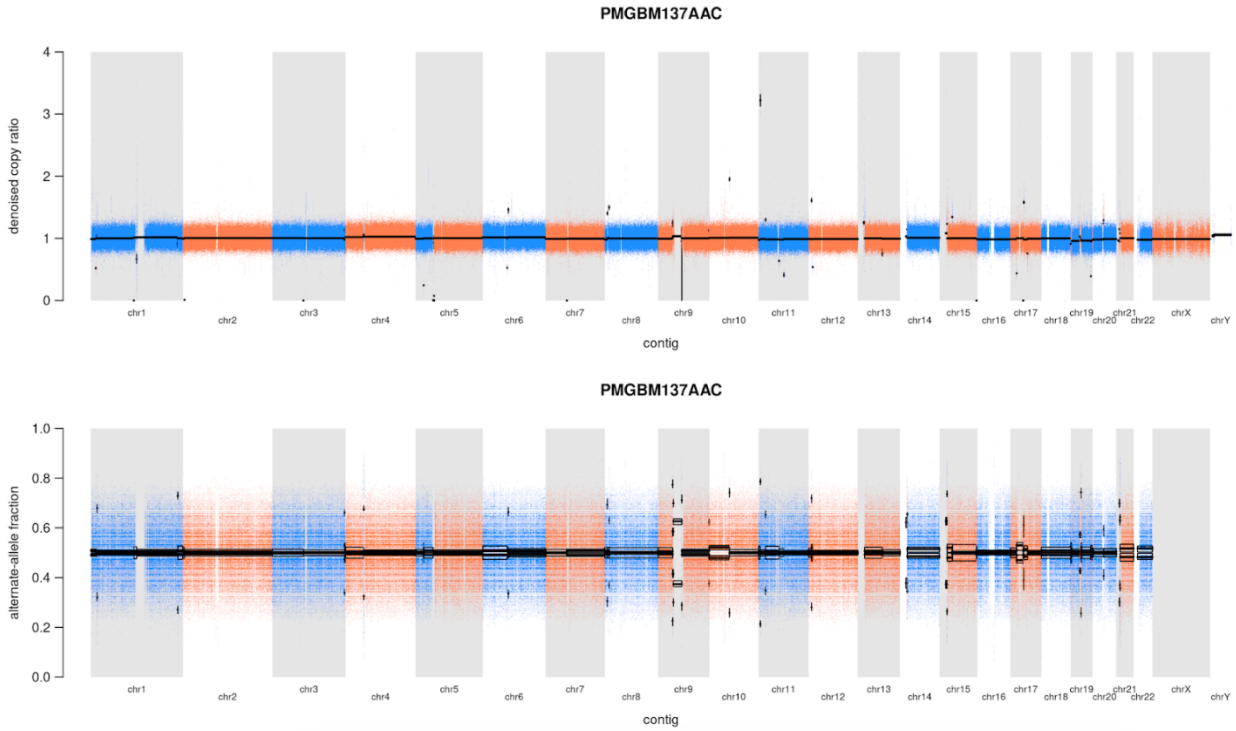
Appendix 1. Scripts used for the pipeline and their function.

Appendix 2

MQ DP	0	50	100	150	200	250	300	350	400	450	500	550	600	650	700	750	800	850	900	950
0	256T 296F	256T 272F	256T 244F	251T 222F	247T 204F	240T 184F	232T 156F	218T 135F	212T 108F	197T 99F	183T 86F	164T 72F	150T 51F	138T 45F	130T 32F	117T 27F	105T 20F	85T 19F	78T 12F	73T 8F
0.05	254T 292F	254T 270F	254T 242F	249T 220F	245T 202F	238T 182F	230T 154F	216T 133F	210T 106F	195T 97F	181T 84F	162T 70F	148T 49F	136T 43F	128T 31F	115T 26F	103T 19F	83T 18F	77T 11F	72T 7F
0.10	253T 279F	253T 260F	253T 234F	248T 214F	244T 196F	237T 177F	229T 150F	215T 131F	209T 104F	194T 95F	180T 82F	161T 68F	147T 48F	135T 42F	128T 31F	115T 26F	103T 19F	83T 18F	77T 11F	72T 7F
0.15	252T 264F	252T 249F	252T 229F	247T 209F	243T 191F	236T 173F	228T 146F	214T 128F	208T 101F	193T 92F	179T 79F	161T 66F	147T 47F	135T 41F	128T 30F	115T 25F	103T 18F	83T 17F	77T 10F	72T 6F
0.20	251T 259F	251T 245F	251T 226F	246T 207F	242T 189F	235T 171F	227T 144F	213T 126F	207T 100F	192T 91F	178T 78F	160T 65F	146T 47F	134T 41F	127T 30F	114T 25F	102T 18F	82T 17F	76T 10F	71T 6F
0.25	251T 246F	251T 234F	251T 215F	246T 197F	242T 180F	235T 162F	227T 137F	213T 119F	207T 94F	192T 86F	178T 73F	160T 60F	146T 45F	134T 39F	127T 28F	114T 23F	102T 16F	82T 15F	76T 9F	71T 6F
0.30	250T 236F	250T 224F	250T 206F	245T 188F	241T 171F	234T 154F	226T 139F	212T 112F	206T 90F	191T 82F	177T 69F	159T 57F	145T 43F	133T 38F	127T 27F	114T 22F	102T 15F	82T 14F	76T 8F	71T 5F
0.35	250T 227F	250T 217F	250T 199F	245T 182F	241T 165F	234T 149F	226T 126F	212T 109F	206T 87F	191T 79F	177T 67F	159T 55F	145T 41F	133T 36F	127T 26F	114T 22F	102T 15F	82T 14F	76T 8F	71T 5F
0.40	249T 221F	249T 212F	249T 195F	244T 178F	240T 161F	233T 145F	225T 122F	212T 105F	206T 84F	191T 76F	177T 64F	159T 52F	145T 38F	133T 33F	127T 23F	114T 20F	102T 13F	82T 12F	76T 7F	71T 4F
0.45	247T 214F	247T 206F	247T 189F	242T 173F	238T 156F	231T 140F	223T 117F	211T 101F	206T 80F	191T 73F	177T 61F	159T 49F	145T 35F	133T 30F	127T 20F	114T 17F	102T 11F	82T 10F	76T 5F	71T 3F
0.50	244T 205F	244T 197F	244T 181F	239T 165F	235T 149F	228T 134F	220T 112F	208T 96F	203T 75F	188T 68F	175T 57F	157T 46F	143T 32F	131T 27F	125T 19F	112T 16F	100T 10F	80T 9F	74T 5F	69T 3F
0.55	240T 191F	240T 183F	240T 170F	235T 154F	231T 139F	224T 125F	216T 104F	205T 89F	200T 70F	185T 63F	172T 52F	154T 41F	140T 28F	128T 23F	122T 15F	110T 12F	98T 7F	78T 6F	73T 5F	68T 3F
0.60	237T 176F	237T 169F	237T 159F	232T 145F	229T 131F	222T 118F	214T 100F	203T 85F	198T 67F	183T 60F	171T 50F	153T 39F	139T 27F	127T 22F	121T 14F	109T 11F	97T 7F	77T 6F	2T 5F	67T 3F
0.65	231T 163F	231T 157F	231T 149F	226T 135F	223T 122F	216T 109F	208T 91F	197T 77F	192T 60F	178T 54F	167T 44F	150T 33F	136T 22F	124T 18F	118T 13F	106T 10F	94T 7F	74T 6F	69T 5F	64T 3F
0.70	215T 130F	215T 124F	215T 120F	212T 108F	209T 98F	204T 91F	197T 74F	187T 62F	182T 50F	169T 44F	158T 34F	142T 24F	128T 17F	117T 14F	112T 9F	101T 6F	90T 3F	72T 3F	67T 2F	62T 2F
0.75	188T 81F	188T 80F	188T 79F	185T 74F	182T 70F	177T 65F	172T 52F	162T 43F	157T 37F	146T 33F	135T 24F	122T 16F	110T 11F	103T 9F	99T 5F	89T 3F	81T 3F	67T 3F	63T 2F	59T 2F
0.80	137T 31F	137T 31F	137T 31F	136T 30F	136T 30F	133T 27F	130T 23F	122T 20F	118T 17F	110T 14F	104T 12F	95T 8F	90T 4F	86T 4F	84T 2F	76T 1F	71T 1F	61T 1F	59T 1F	55T 1F
0.85	116T 13F	116T 13F	116T 13F	116T 13F	116T 13F	114T 13F	111T 12F	105T 10F	103T 9F	97T 7F	91T 6F	83T 5F	79T 2F	75T 2F	73T 1F	69T 1F	64T 1F	58T 1F	56T 1F	53T 1F
0.90	73T 3F	73T 3F	73T 3F	73T 3F	73T 3F	73T 3F	73T 3F	73T 3F	72T 3F	71T 3F	66T 3F	63T 3F	61T 1F	60T 1F	59T 0F	58T 0F	56T 0F	51T 0F	51T 0F	49T 0F
0.95	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	23T 0F	22T 0F	22T 0F	22T 0F	22T 0F	22T 0F	22T 0F	22T 0F	20T 0F

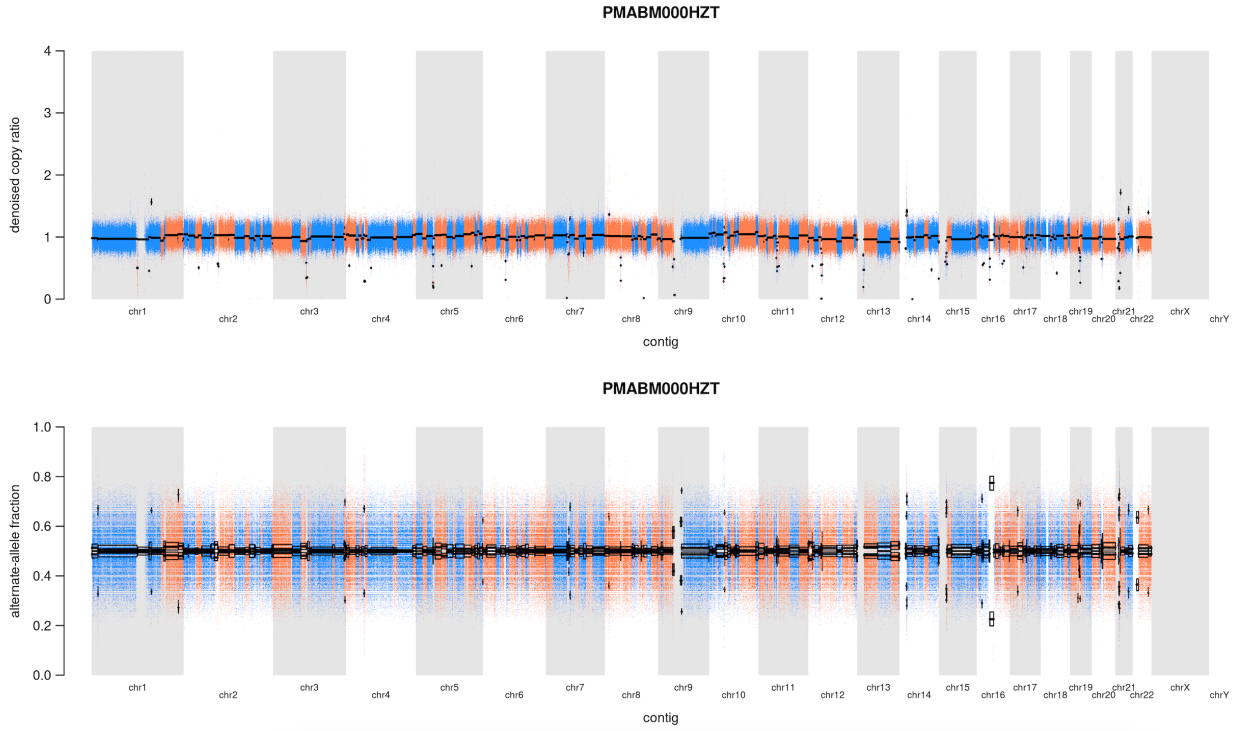
Appendix 2. Merged calls by Dysgu and Manta and their minimal Dysgu Probability (DP) and Manta Quality (MQ) Score. Each cell represents the verified events with a threshold set for the probability- and quality scores. The number before the T(rue) and F(alse) indicates how many SVs are found with the determined threshold combination. In total there are 552 checked events, 256 true and 296 false, which will all be found if the minimal probability- and quality score are set at 0 (top left cell).

Appendix 3



Appendix 3. CNV data from PrediCT138. The mutation is called on chromosome 19 around 4MB. Chromosome 19 has a length of around 59 Mbp. This plot shows no clear indication of a CNV at this location.

Appendix 4

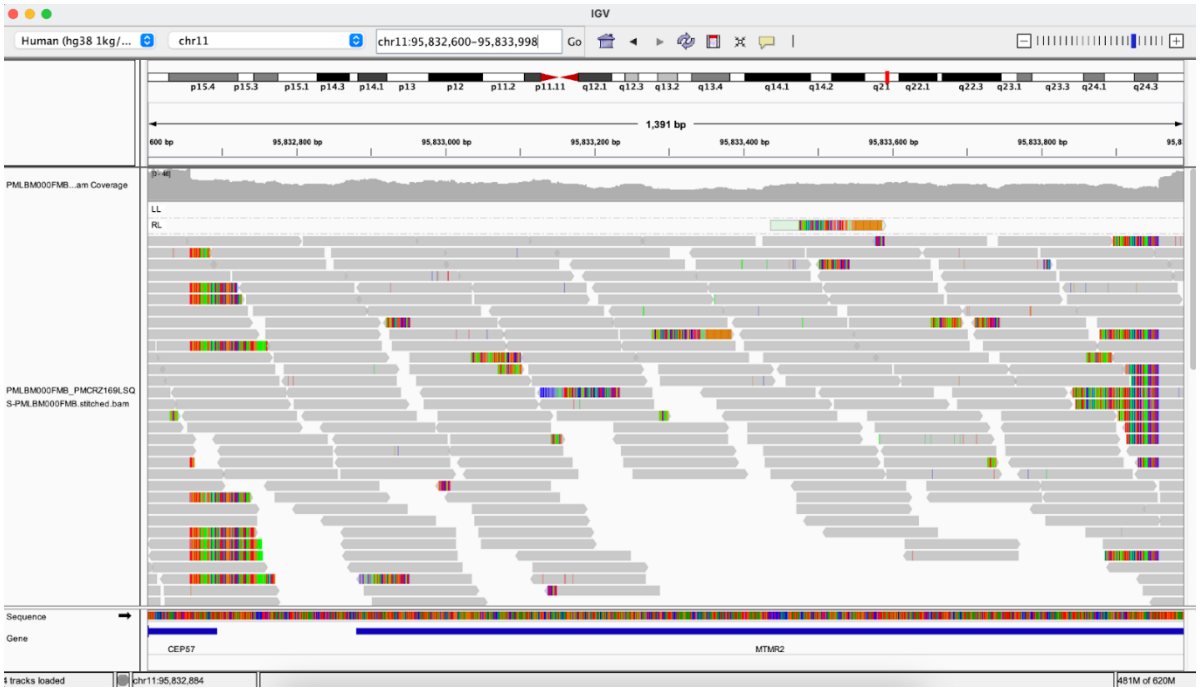


Appendix 4. 2. CNV data from PrediCT160. The mutation is called on chromosome 11 around 65 Mbp. This is around the central region of the chromosome. There is no clear indication of a mutation at this location.

Appendix 5

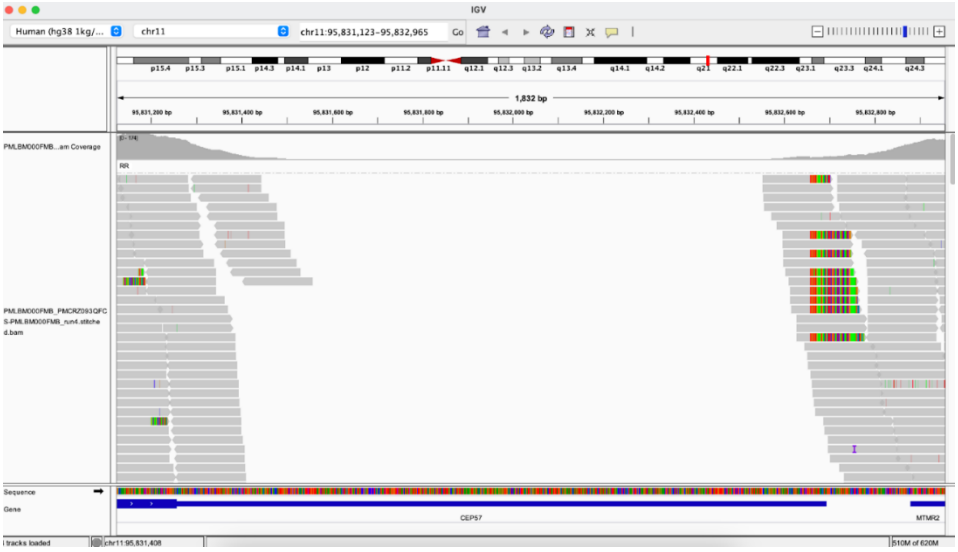
Appendix 5. This event has been identified by Dysgu and Manta but appears to be insignificant. The event was called in the exon. There are two points that might be breakpoints outside of the exon, but the presence of a deletion is not convincing.

Appendix 6



Appendix 6. This deletion eliminates the terminal part of CEP57 and the initial part of MTMR2. The cancer predisposition gene, CEP57, is interesting for this project.

Appendix 7

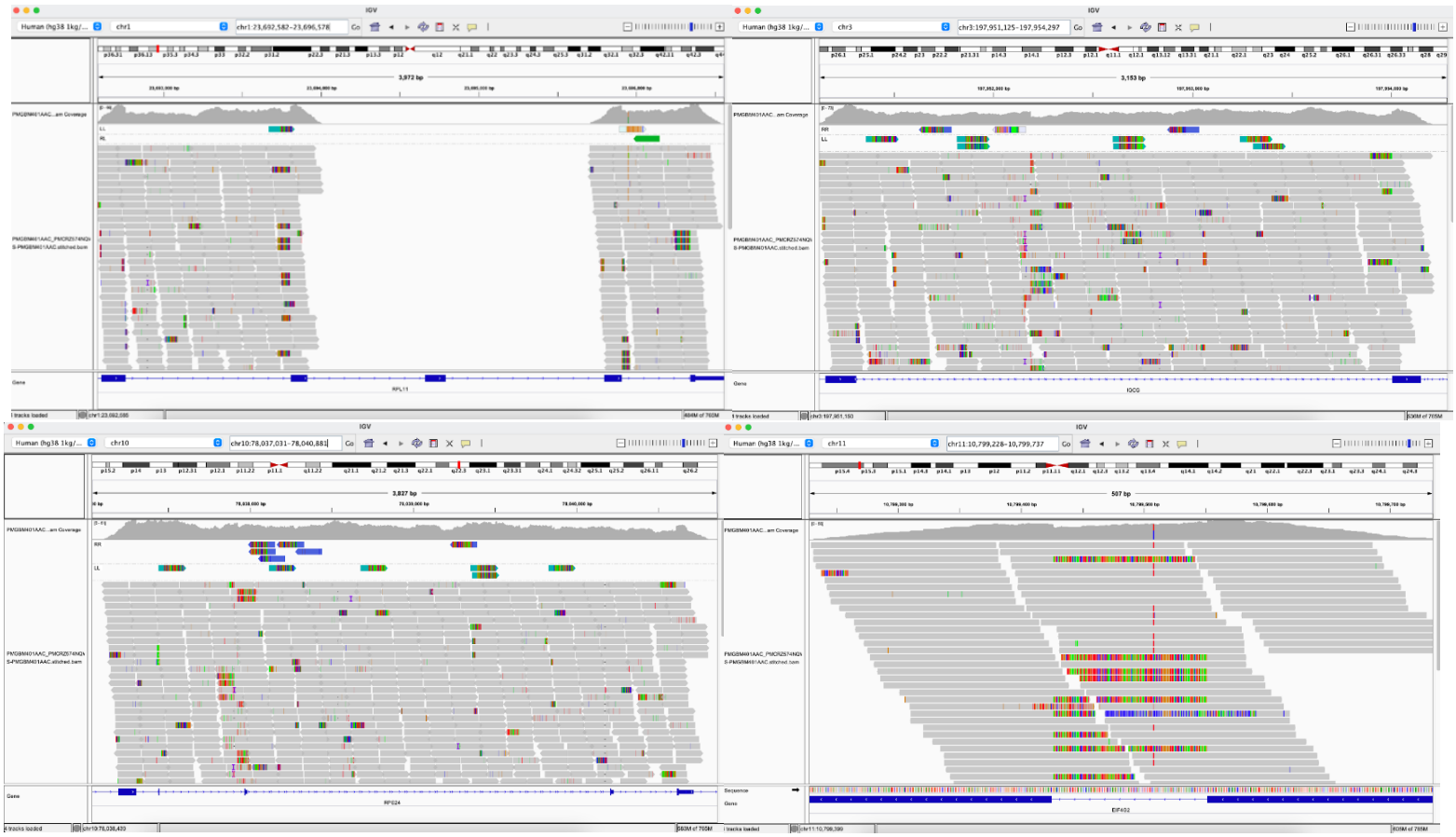


Appendix 7. Segment deletion of CEP57 as revealed by the WXS data.

Appendix 8

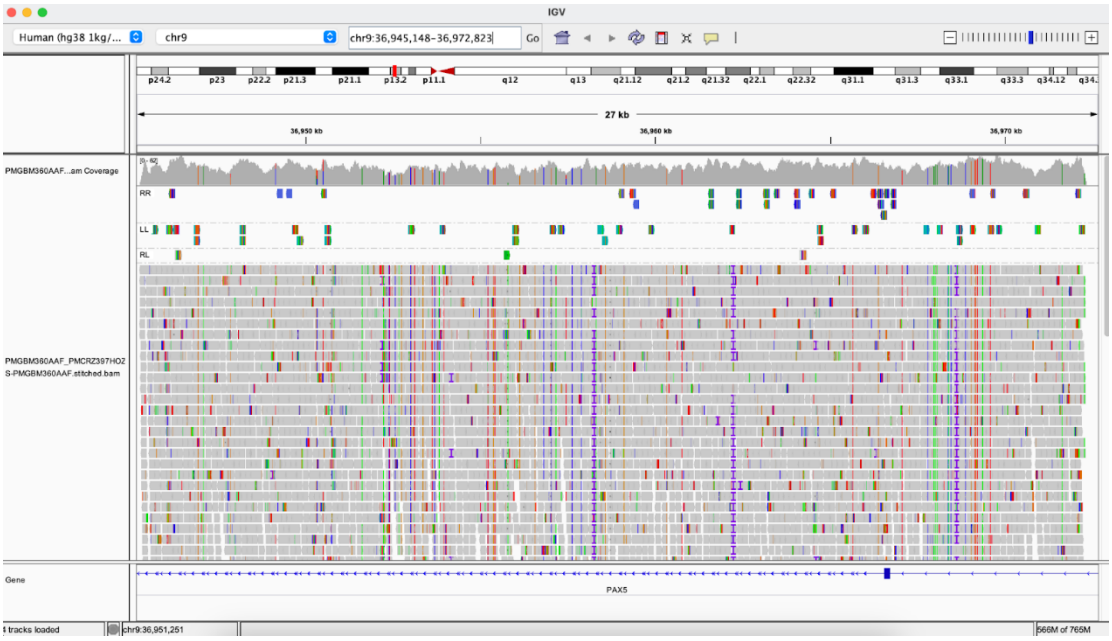
Appendix 8. Deletion in ETV6 in PrediCT403

Appendix 9



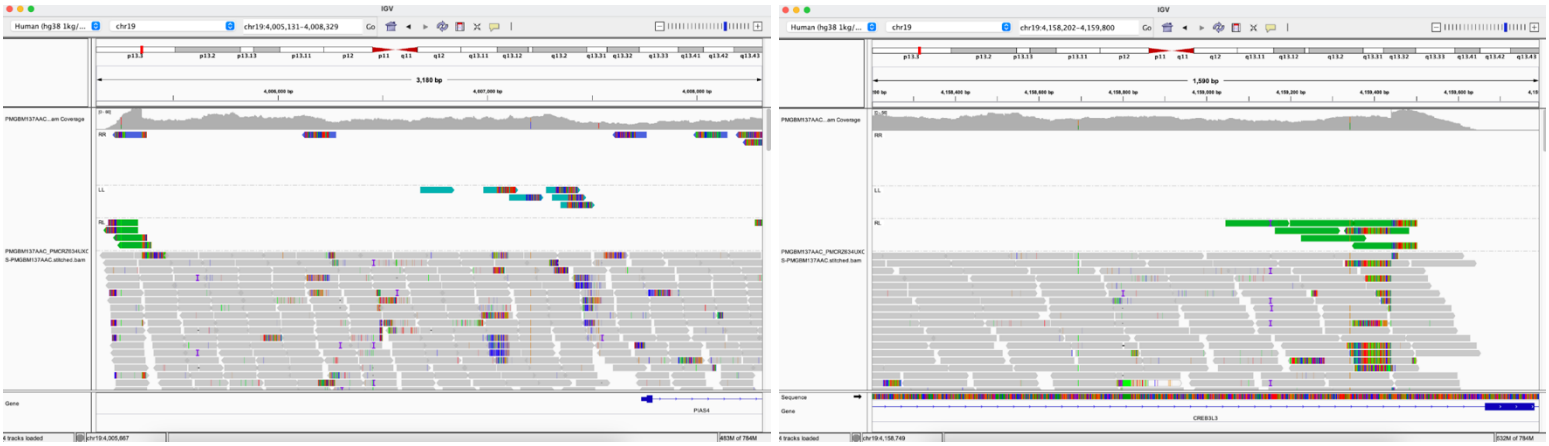
Appendix 9. Five SV called in PrediCT242. In the top left gene are two events called, in the other three genes is one event called. All called events provide insufficient evidence for being correct.

Appendix 10



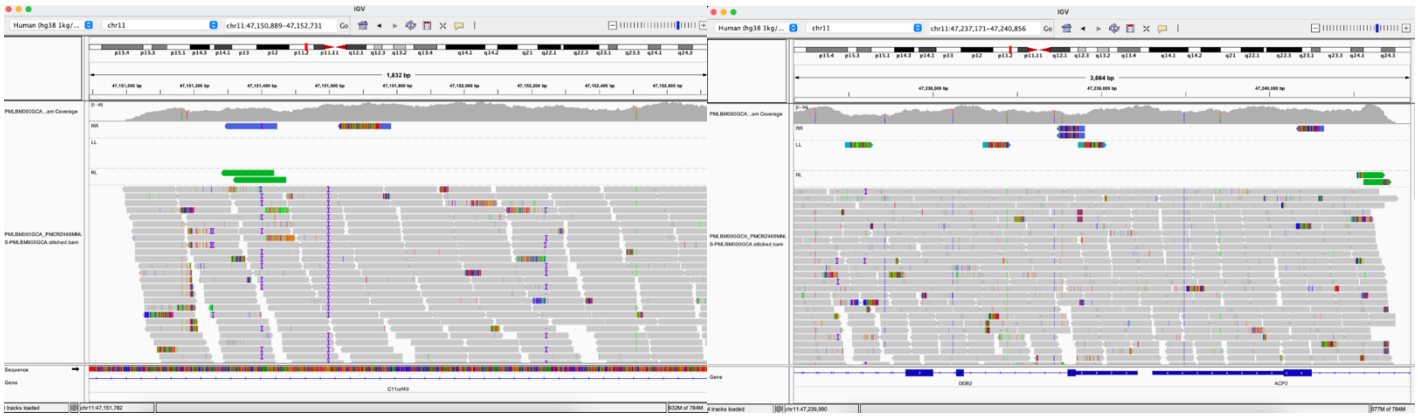
Appendix 10. 26673 base pairs deletion called in PrediCT738. No breakpoints are present, and there is no decrease in coverage observed. Therefore, the SV is most likely incorrectly called.

Appendix 11



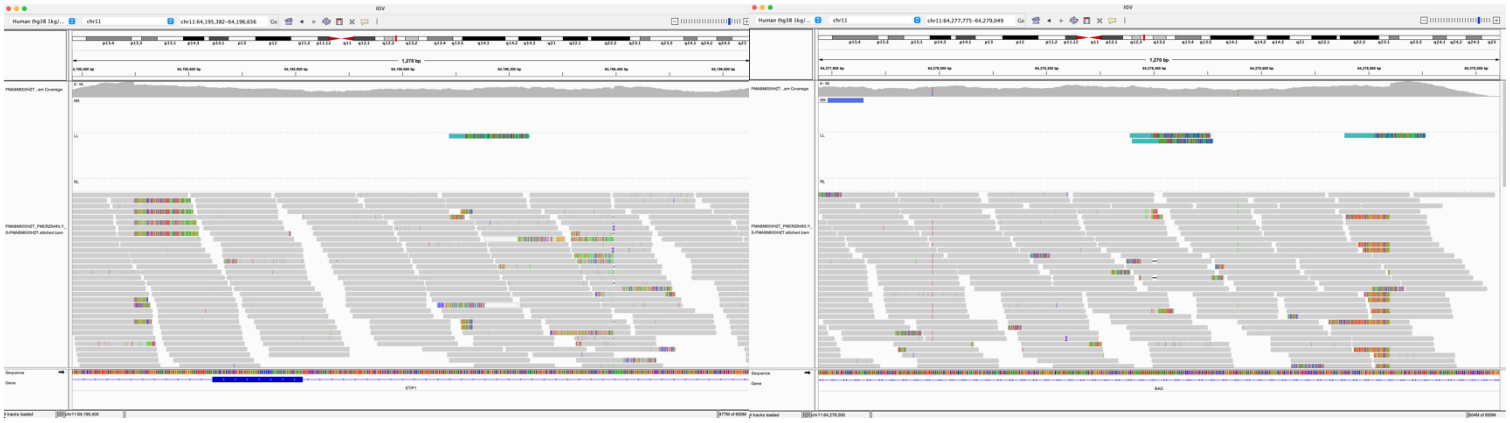
Appendix 11. Two breakpoints of the 154090 base pairs deletion in PrediCT138. There is no evidence in the CNV plot (Appendix 2) that there is a deletion between these breakpoints.

Appendix 12



Appendix 12. Two called breakpoints of an 89363 base pairs deletion in PrediCT917. No breakpoints are present and no decrease in coverage is observed. Hence, the SV is most likely incorrectly called.

Appendix 13



Appendix 13. Two breakpoints of the 83342 base pairs deletion called in PrediCT160.

Appendix 14

Appendix 14. Deletion in PrediCT347. This event does not cause a frameshift.