BSc Bio-informatica

Course: BAFSTU
59 ECTS
August 2019 – May 2020

06-05-2020

Research Project

Developing and validating bioinformatic pipelines that enable assessment of VNTR expansions across haploid human genomes using noisy long reads

Thesis

Author
Nick Radunovic
1101024

Supervisors
Prof. Marcel Reinders (TU Delft) &
Dr. Henne Holstege (AUMC)

Study Coordinator
Hanneke Laan (HSL)

Delft University of Technology – Delft Bioinformatics Lab (Electrical Engineering, Computer Science and Mathematics Faculty)

Alzheimer Centre and Department of Clinical Genetics – Amsterdam University Medical Center

version 2

Preface

This report was written as part of my graduate internship at TU Delft and the Amsterdam UMC. During this internship, I supported the 100-plus Study with their research on how cognitively healthy centenarians (CHCs) escape Alzheimer's Disease (AD). The overall aim is to investigate the impact of VNTR-length on cognitive outcome. For this, 500 genomes are going to be sequenced in total, half of which are from CHCs representing extreme controls[1], and half of which are from (sporadic) AD patients representing cases. The goal is to identify repeat expansions and to compare them between the two groups. My contribution to this project was to facilitate the foundation of the project on the bioinformatic side; to facilitate data management for the sequence data and to design and develop bioinformatic pipelines that could predict VNTR expansions and reconstruct haplotypes from the sequence data of controls and cases. A pipeline that calls structural variations on the reconstructed haplotypes is designed and developed as well. However, this pipeline wasn't tested or validated and therefore not included in this work. An illustration of this pipeline can be found in the Appendix nonetheless.

TABLE OF CONTENTS

ABSTRACT

Background: Recently, studies have shown that *'variable number of tandem repeats'* (VNTRs) are associated with neurological diseases, suggesting that it is beneficial to study large structural variations in order to identify novel genetic factors that modulate AD risk. Thanks to long read sequencing techniques, more of these structural variations are now assessable, as long reads contain information across longer sequence spans. The project goal is to develop bioinformatic pipelines that enable assessment of VNTR expansions across a diploid genome from long read data. Additionally, as structural variations can be different for each haplotype, a pipeline is developed that can reconstruct haplotypes from long-read sequence data of diploid genomes, so that thereafter, studies can be done on each haplotype specifically.

Methods: Two pipelines have been developed and validated: one that generates consensus haplotypes (haplotype-reconstruction pipeline) and one that predicts VNTR expansions (VNTR-detection pipeline). The haplotype-reconstruction pipeline can be used both with or without a reference genome specified. Both pipelines were tested on the well-characterized and publicly available human HG002 genome, which gave an indication on how well the developed pipelines perform. The performance of both options was measured through the assembly quality they obtained. The VNTR-detection pipeline was validated by checking whether or not repeat expansions were present in the output.

Results: The quality assessment of the haplotype-based assemblies showed that supplying a reference genome obtained a N50 of at most 7,1 Mbp, whereas omitting a reference genome obtained a N50 of at most 6,2 Mbp. However, omitting a reference genome resulted in generating a haplotype-based assembly containing more indels with length >5 bp. Both designs had a similar phasing performance, with each approach phasing approximately 80% of the mapped reads. The VNTR-detection pipeline predicted VNTR expansions across the diploid genome, while roughly sensing distinct alleles.

Discussion and conclusion: The results showed that the haplotype-reconstruction pipeline generates better consensus haplotypes for VNTR-detection when no reference genome is supplied. However, more samples have to be sequenced to confirm whether this is always the case. The performance could be improved by including short reads for polishing. The obtained haplotypes could subsequently be used to detect structural variations on the haplotype-level. The VNTR-detection pipeline was shown to predict VNTR expansions in a diploid human genome and hence, could be used to assess disease-associated repeats. In order to simplify comparisons of VNTR expansions between groups, the pipelines could be extended by allowing multiple samples to be processed at once in the future.

1. INTRODUCTION

Dementia strikes half of all people over 85 years and its incidence increases exponentially with age.[2] According to the World Health Association, more than 150 million people will suffer from dementia in 2050.[3] Despite of great efforts, there is currently no cure that prevents or delays the onset of dementia. Nevertheless, studies have successfully tried to identify markers in DNA that lead to an increased risk of Alzheimer's disease showing that variations in the *ABCA7* gene can lead to AD related outcomes.[4–9] Other genes named *SORL1* and *CD33* are possibly also related in the onset of AD.[10–14] Recently, the length of a repeat expansion in an intronic region of the *ABCA7* gene was shown to have impact on cognitive outcome.[8] Of the 772 AD cases and 757 controls tested, repeat lengths >5.7 kilobases (kb) were detected in 7.3% of the cases and 1.7% of the controls (odds ratio 4.5, p=0.0008). This finding, combined with the association of other repeat sequences with neurological diseases[4], suggests that it is beneficial to study repeat expansions in order to identify novel genetic factors that modulate AD risk.

Of all classes of repeat based variation, the *'variable number of tandem repeats'* (VNTRs) are the most interesting ones to investigate in order to learn about AD, as these repeats contain the strongest gene regulatory properties.[15,16] Tandem repeats are repeat sequences in DNA that are directly adjacent to each other. Such a repeating sequence is called a repeating unit. Variable number of tandem repeats are then a subset of tandem repeats, with the property that they are polymorphic, which means that the number of repeating units of a sequence is highly variable within a population. For simplicity, the number of repeating units of a sequence of a VNTR shall be referred to as "VNTR length" from here on. Each repeating unit found in VNTRs, comprises a sequence of 16-64 base pairs.[17] Figure 1a shows an example of how the form of a VNTR i.e. the VNTR length can vary within a population. Since the length of VNTRs can vary on each allele within a genome, it is interesting to distinguish between VNTR length on the haplotype level. When the length of a VNTR on one haplotype is longer compared to that of a reference, it is named a repeat expansion. Doing haplotype specific analysis can give insights in how repeat expansions are distributed over a lineage.

1.1. Long read sequencing

Until now, highly repetitive regions in the human genome such as VNTRs, have been mostly neglected due to the technical limitations of sequencing approaches. Using long read (>10 kb) sequencing techniques, more of these regions of repeat sequences are now assessable. After all, long read sequencing techniques enable reads to be longer and hence, anchor to flanking sequences while spanning the long repeat region during assembling and mapping (See Figure 1b). For this, long read sequencing methods, currently used by Pacific Biosciences[18] (PacBio) and Oxford Nanopore technologies[19], are now taking a big leap in the sequencing market despite the higher overall error-rate (75–90% accuracy)[18,19]. Due to higher error-rates than short read sequencing, long read sequencing is not often used for finding single nucleotide variants (SNVs) and indels, but is particularly useful for *de novo* assembly, haplotype phasing, structural variant (SV) detection and finding long repeats, all of which require information across longer sequence spans[20].

Considering PacBio sequencing (also known as 'SMRT sequencing'), two different sorts of reads can be distinguished, namely: continuous long reads (CLR) and high-fidelity (HiFi) reads. The former consists of longer reads (>50 kb) while being less accurate (~10% error rate)[21], and the latter comprises shorter sequences (~13.5 kb) while maintaining a higher accuracy (<1% error rate)[20]. For finding repeats, among which VNTRs, CLR is the more appropriate choice when choosing between PacBio's CLR and HiFi data types. After all, reads containing information across longer sequence spans aid the analysis of highly repetitive regions. For that reason, we limit our discussion to CLR.

CLRs are the longest possible reads a PacBio system can produce, as its subreads are generated from a single continuous template. CLR reads have a subread length approximately equivalent to the polymerase read length, indicating that the sequence is generated from a single continuous template from start to finish. A *polymerase read* is the sequence that a polymerase is producing on a single run during sequencing and a *subread* is the actual sequence without ligate adapters, that represents the DNA of interests. A polymerase read contains, therefore, at least one subread and possibly more. CLR reads are more suitable for projects requiring very long reads as is the case in *de novo* assembly projects.[20,22]

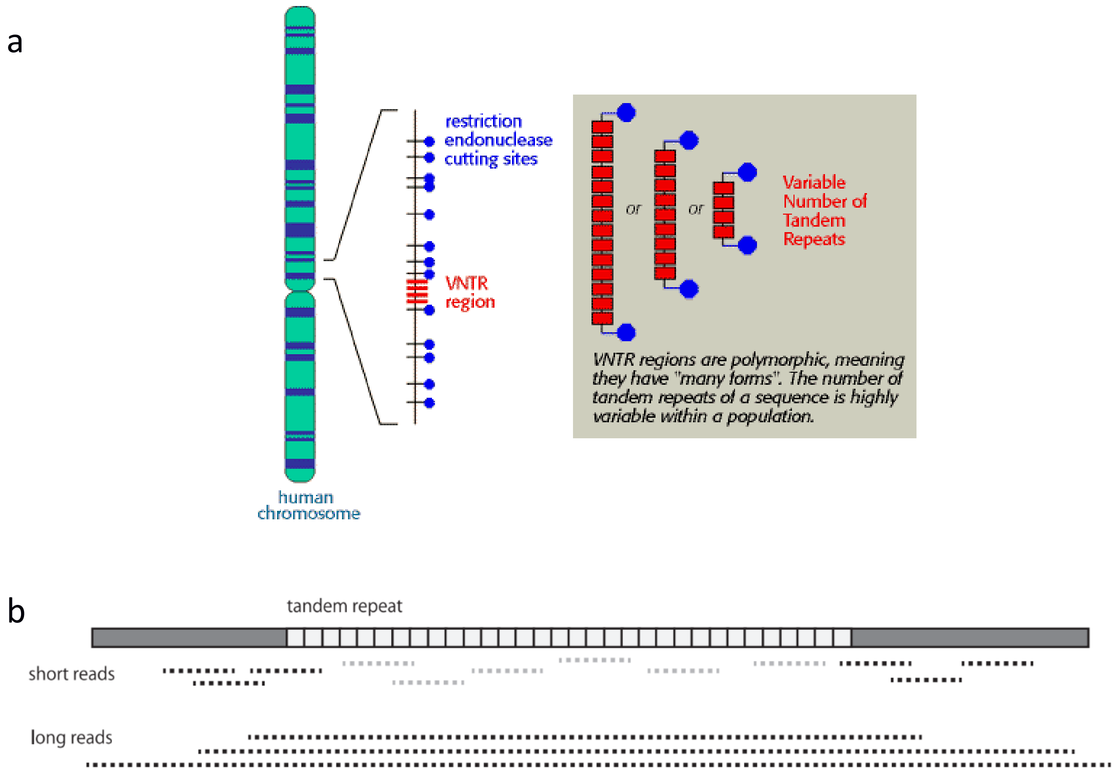

a
restriction endonuclease cutting sites
VNTR region
human chromosome
or
or
Variable Number of Tandem Repeats
VNTR regions are polymorphic, meaning they have "many forms". The number of tandem repeats of a sequence is highly variable within a population.
b
tandem repeat
short reads
long reads

Figure 1: Tandem repeats. a Schematic view of a VNTR and its polymorphic behavior within a population.[23] Note, that the red blocks represent the repeating unit, i.e. the sequence pattern that is repeated multiple times. The number of these repeating units can vary for each haplotype of each human genome. **b** Tandem repeats are difficult to detect by short reads, but long reads that encompass whole repeats can use unique flanking sequences to align whole repeats.[24]

1.2. Haplotype phasing

Regarding the detection of VNTRs across the genome, it is important to note that a human genome consists of two sets of alleles, each inherited by one parent. The alleles of one set comprise together one haplotype (haploid genotype) and the VNTR-length can thus be different for each haplotype. Therefore, more information regarding VNTRs, repeat expansions and their lineage can be obtained when both haplotypes, associated with one person, are reconstructed separately. To obtain both haplotypes from a sequenced human genome, haplotype phasing is necessary.[25,26] Haplotype phasing is the process of *phasing* the diploid genome into two distinct haplotypes. There are two distinct classes of haplotype phasing strategies: reference-based and assembly-based haplotype phasing. Since assembly-based haplotype phasing usually requires high coverage (>50x) sequence data, and for some workflows even Hi-C reads or short reads[27], we limit our discussion to reference-based haplotype phasing. After all, no short read data or high coverage sequence data shall be available for this project.

Reference-based haplotype phasing (hereafter called 'haplotype phasing') is the process of *phasing* the diploid genome into two distinct haplotypes, for example by identifying which reads belong to which haplotype. Figure 2 shows the general workflow of haplotype phasing. The process of assigning reads to one of the haplotypes usually relies on a process named variant calling, which is performed prior to the actual phasing. During variant calling, single-nucleotide polymorphism (SNPs) variants are *called* i.e. identified by comparing the reads with each other or with a reference, and when done as part of the haplotype phasing process, alleles can be recognized based on the SNPs being called.

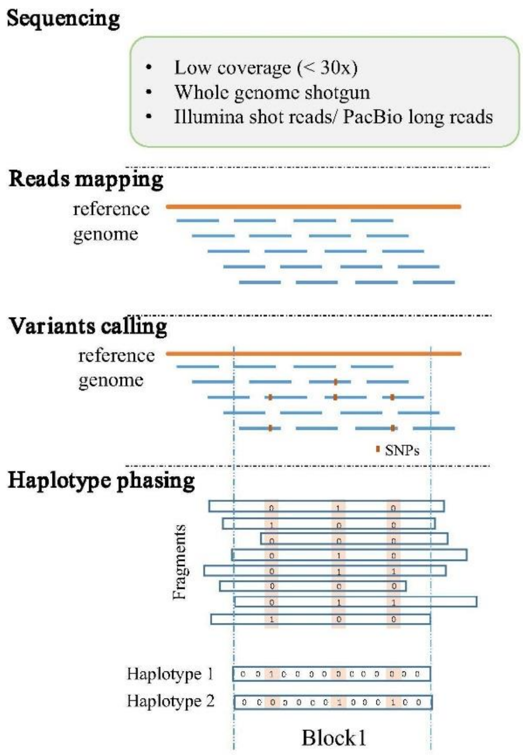


Figure 2: Alignment-based haplotype phasing workflow[27]**.** First, a diploid genome is sequenced with a relatively low coverage (<30x), generating either short or long reads. The reads are subsequently mapped to a reference genome for variant calling. Afterwards, linked variants are extended into phase blocks, or haplotype-specific contigs, each representing one of both haplotypes. Alternatively, instead of generating phase blocks, the initial reads could be assigned to one of the haplotypes. Reads containing few SNPs might stay unassigned, i.e. unphased, as these reads won't contain enough variations on which could be properly phased.

1.3. Project objective

The objective of this internship project is to design, develop and validate bioinformatic pipelines that can assess VNTR expansions in PacBio CLR sequencing data from a diploid human genome. The straight-forward way to do so is by detecting VNTRs directly from the unassembled reads, that is, without disentangling the two haplotype beforehand. The secondary aim is to design, develop and validate bioinformatic pipelines that can reconstruct haplotypes from PacBio CLR sequencing data of a diploid genome, so that thereafter, VNTRs and other structural variations can be called on those haplotypes specifically. Because of a current lack of haplotype phasing tools that also phase structural variations[27], tools that exclusively phase on SNPs have to be considered during development of the pipelines. Considering long read data, tools should be used that are especially designed for long reads analysis which differ from tools normally used in short-read analysis. This is especially the case when assembling, mapping and haplotype phasing. Moreover, it is important to take speed into account when developing the pipelines, to prevent having pipelines that require weeks, or even months, to complete for a single sample.

2. MATERIALS AND METHODS

This section covers the workings of the pipelines together with the resources and data that were being used for developing and validating the pipelines. Also, the reasoning behind the design of the pipelines and justification for each relevant tool shall be discussed. Each section discusses a fraction of the pipeline and ends by explaining how the validation was done in order to interpret the results shown in Section 3.

2.1. Data and file management

The dataset that was being used for developing and validating the pipelines was a dataset from the well-characterized human genome HG002. This dataset was acquired from the PacBio open-source dataset repository[28] and was sequenced on the PacBio Sequel II System using continuous long reads (CLR). Additional information on the HG002 dataset and how sequencing and library prep was done can be found in Appendix A. With regard to the computational resources: the pipelines were developed and validated on the HPC cluster of TU-Delft[29].

The bioinformatic pipelines were implemented in such a way that they manage and maintain a specific folder structure as is shown in Figure 3. Each sample was transferred from the Sequel II System to a network storage, called 'root' in this example, after they had been sequenced. Table 1 displays an example of the contents of a sample folder output by the Sequel II System, which the pipelines takes in as input. In this example, m54008 is the instrument ID number and 160116_003634 is the run date, in YYMMDD format, and time, in UTC format. From here on, this code identifier is called *<sample_id>*, for it is different for each sequenced sample.

Table 1: Overview of contents that were present within a sample data folder output by PacBio Sequel II System. These files comprised together the input for the pipelines.

Contents of a sample data folder	Explanation
m54008_160116_003634.scraps.bam	Contains rejected subreads and excised adapters
m54008_160116_003634.scraps.bam.bai	Bam index file
m54008_160116_003634.subreads.bam	Unaligned base calls from high-quality regions
m54008_160116_003634.subreads.bam.pbi	Bam index file
m54008_160116_003634.subreadset.xml	This file is needed to import data into SMRT Link
m54008_160116_003634.sts.xml	Contains summary statistics about the collection
m54008_160116_003634.transferdone	Contains a list of files successfully transferred
m54008_160116_003634.adapters.fasta	Contains adapter sequences

The subreads.bam file contained the actual sequencing data i.e. all the unaligned base calls from high-quality regions of the concerned sample, and was already quality filtered and trimmed by the PacBio Sequel II System. When each sample was being transferred to the root folder in which all data was stored and collected, after they had been sequenced, the subreads.bam file was selected and moved to the Work_dir/Reads folder. Once this selection and organizing step was done, the bioinformatic pipelines were applied on the data in Work_dir/Reads.

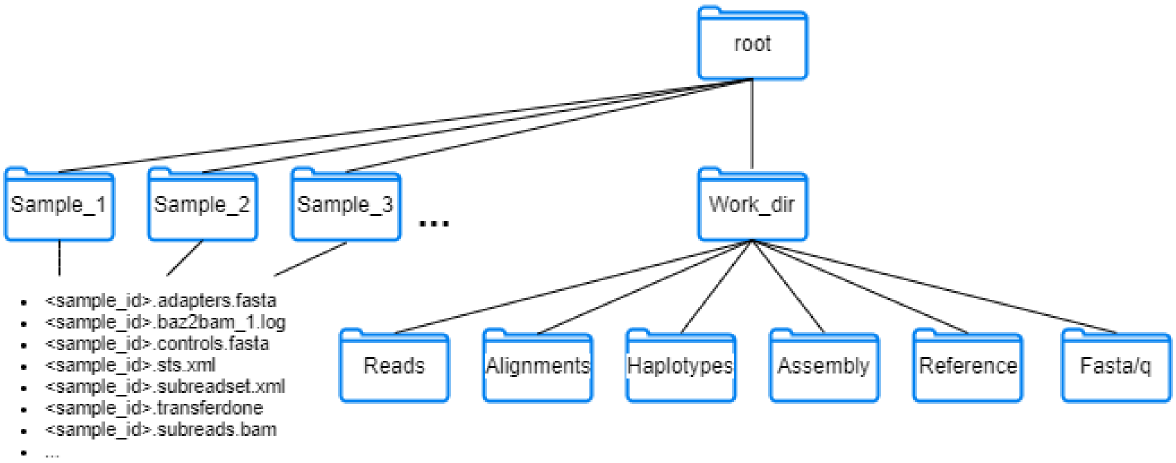


Figure 3: File and folder structure used by the bioinformatic pipelines. The Work_dir folder is created by the bioinformatic pipelines and includes all the intermediate and output files of the pipelines. The folders Sample_1, Sample_2, etc, are folders output by the PacBio sequel II System.

2.2. Software dependencies

The bioinformatic pipelines for this project were developed using Snakemake[30]. Snakemake is a Python based bioinformatics workflow engine that allows the user to create very general rules with wildcards, delivering a workflow that was optimal and well-suited for the needs of this project. With the help of applying general rules, various tools could be called and used in a specific arrangement, in order to get to the end goal. Table 2 outlines all of the dependencies of the pipelines. The source code of the pipelines is accessible on GitHub[31].

Table 2: The following software is needed on your system for all programs to work.

Software	Explanation	Version (or higher)
Python	Prog. Language	3.7.3
GNU Bash	Prog. Language	
GNU Make	Program tool	
GCC	Compiler	4.8.5
Minimap2	Aligner	2.17-r941
LAST	Aligner	last-1060
Longshot	Variant caller	0.3.5
Flye	Assembler	2.7-b1585
Miniasm	Assembler	0.3-179
Wtdbg2	Assembler	2.5
QUAST	Quality assessment tool	5.0.2
Tandem-genotypes	Detects changes in VNTR-length	1.5.0
Samtools	Suite of bioinf. Utils	1.9
Bcftools	Suite of bioinf. utils.	1.9
Bgzip	Compression utility	1.9
Gzip	Compression utility	1.5
Snakemake	Workflow engine	5.7.0

2.3. Bioinformatic pipelines

The general flowchart that outlines the bioinformatic pipelines is depicted in Figure 4. As Figure 4 shows, the analysis was divided into two pipelines: the Structural Variant Calling (SVC) pipeline and the VNTR-detection pipeline. The SVC pipeline was developed such that it can reconstruct the haplotypes of the sequence data measured on a diploid genome, so that thereafter, VNTRs and other structural variations

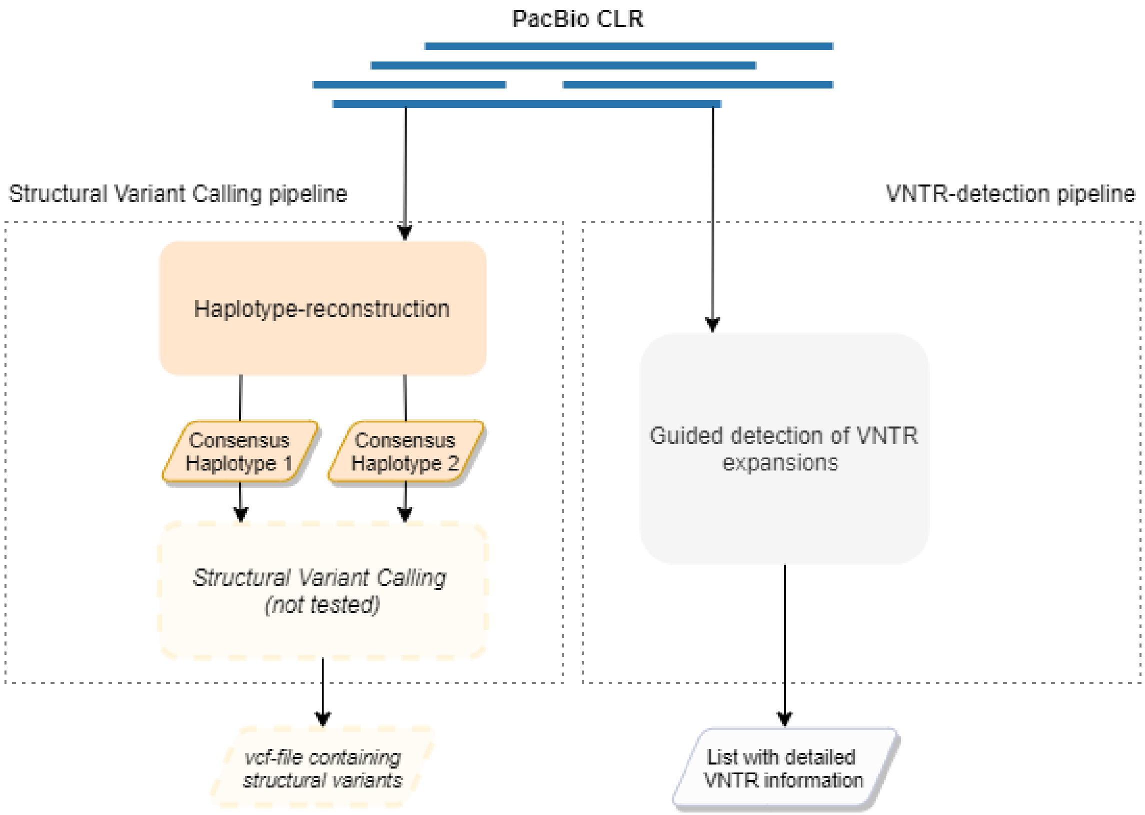

PacBio CLR
Structural Variant Calling pipeline
VNTR-detection pipeline
Haplotype-reconstruction
Consensus Haplotype 1
Consensus Haplotype 2
Structural Variant Calling (not tested)
Guided detection of VNTR expansions
vcf-file containing structural variants
List with detailed VNTR information

Figure 4: General overview of the bioinformatic pipelines, developed for downstream analysis of PacBio noisy long read data. There are two main pipelines, that is, the Structural Variant Calling (SVC) pipeline and the VNTR-detection pipeline. Both pipelines take in as input PacBio (CLR) long reads. The SVC pipeline consisted originally of two stages: the haplotype-reconstruction stage and structural variant calling stage. However, the structural variant calling stage wasn't fully tested and hence, not included in this work. Therefore, it is outlined by a dashed line. The haplotype-reconstruction stage generates consensus haplotypes. The VNTR-detection pipeline detects repeat copy number changes, which include VNTR expansions and contractions as well. The parallelograms represent the output of the bioinformatic pipelines. The rectangles, with rounded corners, represent the different pipeline stages.

can be called on those haplotypes. Originally, the SVC pipeline comprised two stages: haplotype-reconstruction and structural variant calling. However, the structural variant calling stage wasn't fully tested and hence, no results were generated. The structural variant calling stage is therefore not included in this work. However, Figure 4 does show the structural variant calling stage, as this gives a more comprehensive overview of the pipeline stages and their relation to each other. The methods of the structural variant calling stage can be found in Appendix B, as the stage was designed and developed nonetheless. For convenience, the haplotype-reconstruction stage shall be called haplotype-reconstruction pipeline from now on. The haplotype-reconstruction pipeline outputs haplotype-specific assemblies that were meant to act as an input for the structural variant calling stage. The files that the SVC pipeline obtains can, subsequently, be used in further analysis regarding both repeat oriented and non-repeat oriented research. The VNTR-detection pipeline, on the other hand, was explicitly developed to detect VNTRs directly from the unassembled subreads, that is, prior to any haplotype disentangling. Therefore, the VNTR-detection pipeline doesn't distinguish between haplotypes and hence, detects VNTRs without any link to the haplotype they came from. The VNTR-detection pipeline outputs a text-file containing detailed VNTR information on each line e.g. the start-stop position on the chromosome and the repeated pattern of the VNTR, the gene and gene part on which the VNTR is found, and the change in VNTR-length (repeat copy number changes) with respect to a reference. The next sections outline each of these pipelines in detail together with how each step was validated.

2.3.1. Haplotype-reconstruction pipeline

The workflow of the haplotype-reconstruction pipeline is depicted in Figure 5. Note, that the haplotype-reconstruction pipeline consists of two approaches, namely: the reference-based approach and the de-novo-based approach. Although each approach follows a different process, both approaches take in PacBio CLR long read data and generate consensus haplotypes. The reference-based approach follows a guided approach, as reads are initially mapped to the reference genome. The de-novo-based approach, on the other hand, generates haplotype-specific assemblies without any need for a reference. The two approaches have been developed with the viewpoint of finding out which approach eventually generates better assemblies. What the better assembly is, will depend on the subsequent task to be performed on these haplotype, which in this work is VNTR detection.

The reference-based approach starts with (a) aligning the PacBio CLR reads to a reference genome using minimap2[32]. Next, the alignment obtained by minimap2 is (b) phased using Longshot[33], assigning the mapped reads to one of the haplotypes. (c) To get a consensus for each haplotype, wtdbg2[34] assembles each haplotype-assigned set of reads, together with those reads that Longshot couldn't assign, to generate a haploid assembly. Unassigned reads, i.e. those reads for which Longshot couldn't decide to what haplotype they belong, are also used to construct the assemblies with, as the obtained assemblies would otherwise be too fragmented to be useful in further analysis. The contigs from both assemblies are then merged into one file so that thereafter, all PacBio CLR reads used in step (a), can be mapped back to both haplotypes at once, letting the aligner decide to what haplotype each read belongs in step (d). After aligning the reads to the merged haplotypes, the obtained alignment is split into two separate alignments, each representing the alignment with one of the haplotypes. Lastly, the reads that mapped to their corresponding haplotype were assembled in step (e) generating haploid assemblies.

The de-novo-based approach starts with a de novo assembly performed by Flye in step (f), constructing an assembly out of all PacBio subreads from one diploid sample. Then, an alignment was performed in step (g), aligning the PacBio subreads to the assembly. Using the alignment obtained in step (g), haplotype phasing was performed by Longshot in step (h), assigning the mapped reads to either haplotype 1 or haplotype 2. The reads that got assigned to one of the haplotypes, were then merged with the reads that Longshot couldn't assign, and subsequently were assembled by Flye to obtain consensus haplotypes in step (i).

Steps a-i are explained in more detail below, with for each step a concise description of the relevant tool. Each description ends by an outline on how validation was performed. The parameters that were used in each step are outlined in the documentation which can be found on Github[31].

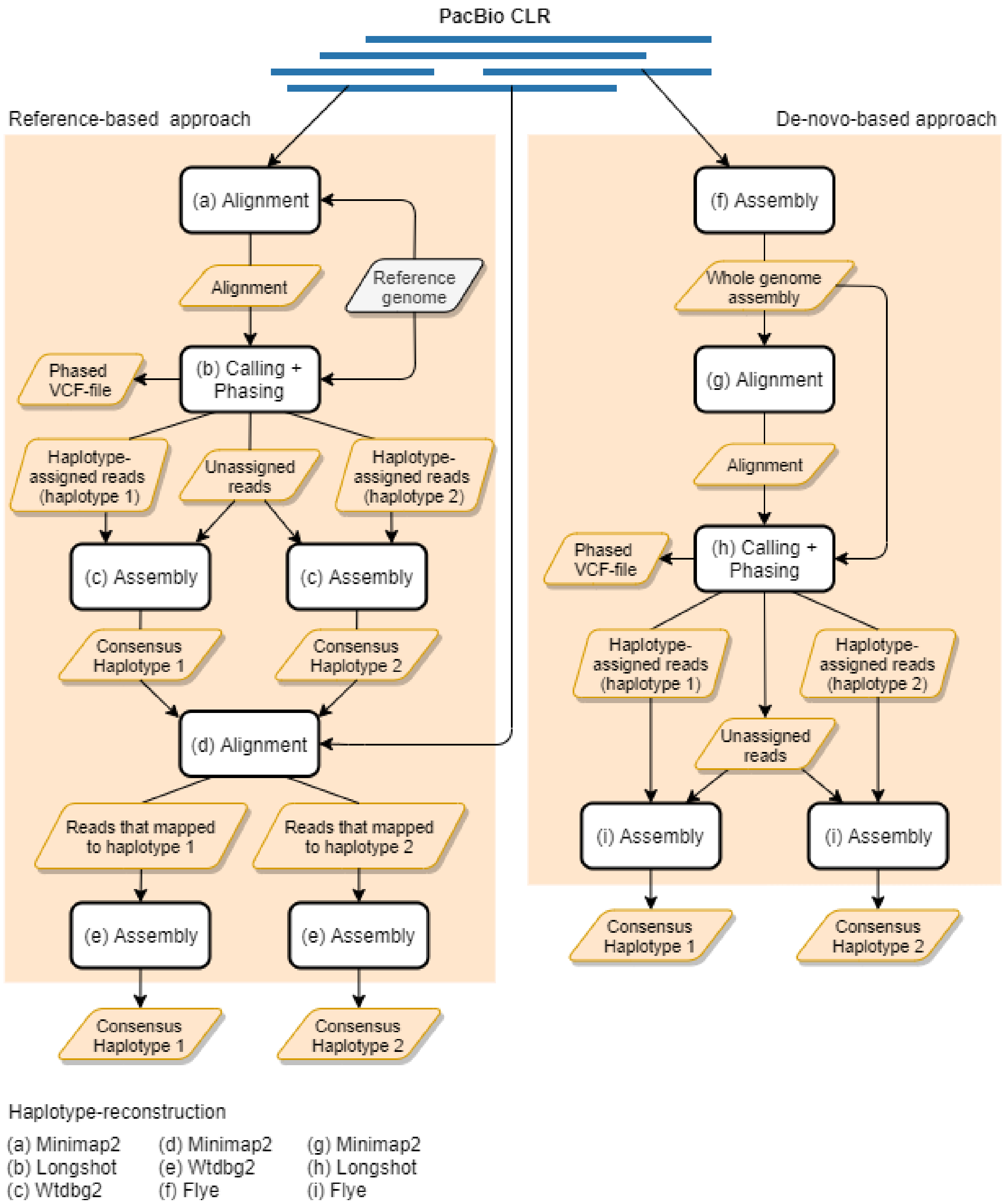


Figure 5: Flowchart of the haplotype-reconstruction pipeline. Two approaches to the haplotype-reconstruction pipeline have been developed: a reference-based approach (left panel) and a de-novo-based approach (right panel). Both approaches take in as input PacBio CLR long read data of one diploid sample and generate consensus haplotypes for that sample. The reference-based approach generates consensus haplotypes guided by a reference genome. The de-novo-based approach constructs consensus haplotypes by means of a *de novo* assembly, thus without any need of a reference genome.

Steps (a) and (g) – Aligning PacBio subreads to a reference

In step (a) and (g), an alignment is performed to a reference, either a human reference genome (a), or the intermediate reconstructed diploid genome (g). Using minimap2, the subreads from a sample are mapped to these references. Minimap2 is a sequence alignment tool typically used for aligning noisy long reads, containing an error rate up to ~10%, against a large reference.[32] Minimap2 is many times faster and more accurate in mapping long reads to a large reference genome.[32] Therefore, Minimap2 performed the alignment in step (a), instead of other long read aligners like BWA-MEM[35] and BLASR[36]. The obtained alignment-file is subsequently sorted and indexed using samtools sort and samtools index, respectively, so that an index file (.bai file) is created. This index file was required to view the alignment in the Integrative Genomics Viewer (IGV)[37] during testing and error checking.

Validation of steps (a) and (g). Part of the validation of steps (a) and (g) was done by assessing the global statistics of the alignment which was obtained by samtools[38] using the command stats. PacBio CLR sequencing data from the human genome HG002 was aligned against the human genome reference GRCh38/hg38[39]. Next to looking at the global statistics, the alignment from step (a) was evaluated in IGV in order to see whether or not there were any reads covering regions of the reference containing VNTRs, such as the Alzheimer's disease-associated *ABCA7* VNTR[8]. Note, that no screenshot of IGV was included in the results, as IGV was solely used for error checking during development. Checking whether or not reads spanned disease-associated VNTRs was only possible for the alignment in step (a), since GRCh38 was annotated.

Steps (b) and (h) – Variant calling and haplotype phasing

In steps (b) and (h), variants are called on the mapped reads which, thereafter, are phased into sets of reads each representing one of two haplotypes. Longshot both calls SNPs and subsequently phases the mapped reads from the alignment obtained in step (a) as well as in step (g). Longshot is a variant calling tool for diploid genomes that takes in long, error prone reads and output phased bam-files containing reads that belong to one haplotype.[33] Longshot has a high accuracy for SNP detection and enables variant detection in duplicated regions of the genome that cannot be mapped using short reads. Each chromosome is called and phased separately so that Longshot could be parallelized and hence would run faster. The obtained vcf-files and haplotype-separated bam-files for each chromosome are merged together afterwards. If Longshot can't decide to what haplotype a read should be assigned due to too little variations in the read with respect to other reads, than the read stays unassigned. The same method and parameters are used in both steps (h) and (b).

Validation of steps (b) and (h). Steps (b) and (h) were validated partly by looking at the percentage of reads that got assigned to each haplotype. A better phasing was indicated by a lower percentage of unassigned reads and when the percentage of reads assigned to haplotype 1 was similar to the percentage of reads assigned to haplotype 2. Also, the phasing result of Longshot was checked by looking for reads that were present in more than one of the separated bam-files it outputted, notifying that something went wrong in the process if some duplicates were found.

Steps (c) and (e) – Haplotype-specific assembly performed with wtdbg2

In steps (c) and (e), haplotype-specific assemblies are generated. Wtdbg2, a *de novo* assembler for long reads[34], creates the assemblies for each haplotype in steps (c) and (e), using both sets of haplotype-assigned and unassigned reads together in step (c), and using reads that mapped to their corresponding haplotype in step (e). After the reads are assembled, wtdbg2 polishes the resulting raw contigs using three iterations, so that it generates polished contigs that represent the consensus haplotypes. The methods of step (e) are identical to those of step (c), though the input and hence the quality of the output, might be different to some extent. The input may differ, since the set of reads that is used to assemble one consensus haplotype is obtained in different ways, either by merging the haplotype-assigned reads with the unassigned reads obtained in step (b), or by selecting the reads that mapped to one of both haplotypes in step (d).

Validation of steps (c) and (e). Steps (c) and (e) were validated by assessing the quality of the obtained assemblies with the quality assessment tool QUAST[40]. When given a genome assembly and a reference, QUAST generates statistics on the overall assembly quality such as N50, overall genome coverage, contig length and number of mismatches and indels with respect to a reference. Of all of these metrics, N50 is the most commonly used to assess the quality of genome assemblies, as it represents the contiguity of the assembly. N50 stands for the maximum length *x* such that contigs with a length at least *x* account for at least 50% of the total assembly length.[40] Generally, higher N50 values indicates better overall quality of the assemblies. Accordingly, others such as N75 or N90 stands for the same, except now *x* accounts for at least 75% or 90% of the genome, respectively. However, in order to use the assembly for finding haplotype-specific VNTR expansions, the assembly must cover VNTRs. Evaluating the number of large indels contained in the assembly is therefore more useful.

Step (d) – Aligning subreads to intermediate reconstructed haplotypes

In step (d), the PacBio subreads obtained after sequencing, are mapped to the intermediate haploid assemblies by merging the contigs of both haploid assemblies and mapping all subreads to this merged set of contigs. In order to preserve the two haplotypes during the merging step, the contig headers are modified such that they contain tags which link their contig to their corresponding haplotype. Minimap2 maps the PacBio subreads to the merged set of contigs, figuring out to what contig, and hence to which of the haplotypes the reads belongs. This alignment-step is performed to find out whether or not more reads will map to the merged haploid assemblies than there were reads mapped to the reference genome GRCh38. After aligning the reads, the alignment is split into two alignments, each representing an alignment to one of both haplotypes.

Validation of step (d). The validation of step (d) was the same as the validation performed for step (b), as both steps covered an alignment performed by minimap2.

Steps (f) and (i) – De novo assembly performed with Flye

In step (f), the pipeline performs a *de novo* assembly on the initial PacBio CLR data using Flye, generating an assembly without preserving information about the haplotypes originally contained in the dataset. In step (i), Flye is being used to generate the assembly for each haplotype. Although the assembly performed in steps (f) and (i) uses a different input (i.e. all subreads in step (f) and a subset of subreads for step (i)), the method and parameters being used are identical. In the assembly process of both steps, Flye polishes the obtained raw contigs with three iterations, deriving polished consensus assemblies in the end. Additional polishing using short reads isn't carried out, for there was no short read data available.

Validation of steps (f) and (i). Since step (f) and (i) produced assemblies, the validation of step (f) and (i) was the same as the validation performed for step (c) and (e), thus by assessing the assembly quality obtained by QUAST.

Comparison of assemblers for the haplotype-reconstruction approaches

In the developmental stage, various assemblers were tried in order to see which assembler suited the concerning step most. Steps (c) and (f) were used to try the assemblers miniasm[41], wtdbg2 and Flye in, and a report was obtained by QUAST showing the performance of each assembler in terms of assembly quality. In the end, a comparison between the assembly quality obtained by each assembler accounted for the choice regarding what assembler to use in steps (c) and (f). Accordingly, the assemblers that suited the steps (c) and (f) most, were used for steps (e) and (i), respectively, as these assembly-steps were part of the corresponding approaches.

Overall speed and resource efficiency of the haplotype-reconstruction pipeline

Since it was important that the pipeline doesn't take too long to process one sample, the amount of time and resources each step required was taken into account while designing the pipelines. Therefore, some general statistics on the overall performance of each considered pipeline step or approach, in terms of required resources, were collected in the development phase. In the end, a table was created, containing the resources that were used while testing and the time for each step to finish, so to give an indication on how long the pipelines may take to process one sample.

2.3.2. VNTR-detection pipeline

The pipeline that detects VNTR expansions is visualized by the workflow depicted in Figure 6. First, the PacBio long reads enter the pipeline where they, subsequently, get aligned to a reference genome with the LAST[42] aligner. Using a repeat annotation file, the repeat copy number changes can then be determined in the aligned reads by comparing the repeat length with that of the reference, using tandem-genotypes[43]. Tandem-genotypes is a tool that predicts the copy number change in noisy long reads. Together with the repeat copy number changes for each VNTR, tandem-genotypes obtains the start-stop position on the chromosome, the repeated pattern and both the gene and gene part on which the VNTR is found. The output of the pipeline is a text file, with on each line the obtained information for each VNTR obtained by tandem-genotypes.

Validation of the VNTR-detection pipeline. The pipeline was tested on the HG002 dataset, using GRCh38 as the reference genome and a repeat annotation file from the UCSC genome database[44], containing repeats with unit length from 1 to >1000 made with Tandem Repeats Finder[45]. For one sample, i.e. HG002, the output of the pipeline was checked in order to see whether or not repeat expansions could be found based on the repeat annotation that was used. We also checked whether or not two clear alleles could be distinguished from the repeat copy number changes, by plotting histograms for the repeat copy number changes using tandem-genotypes-plot[43].

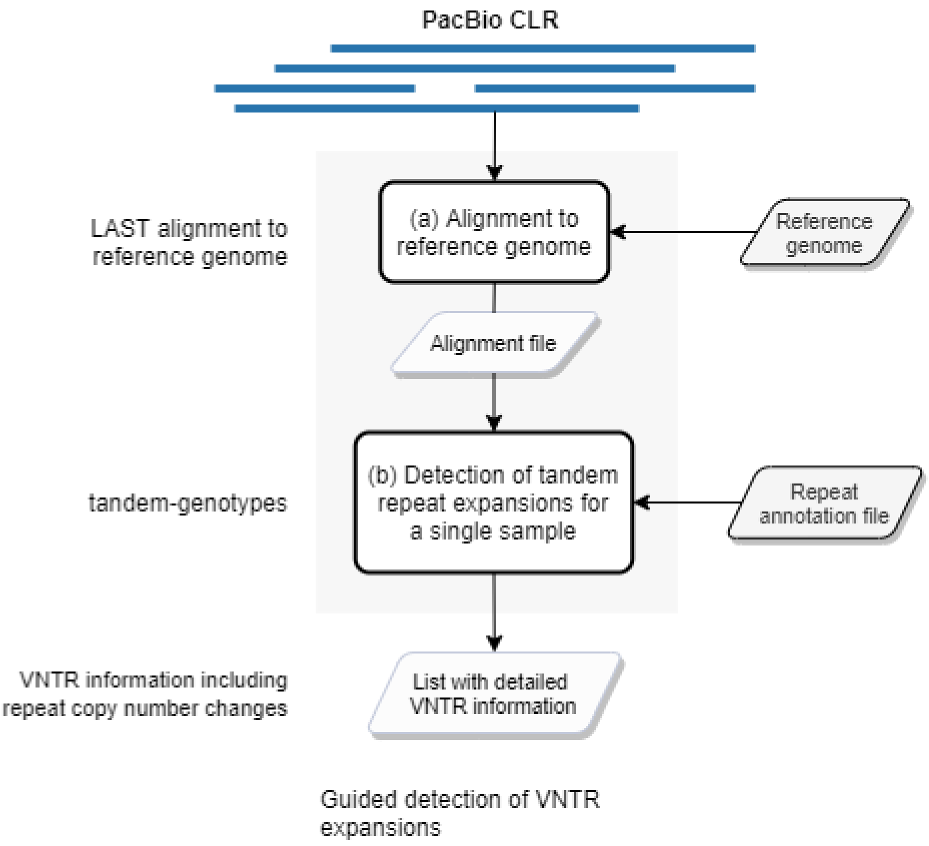

PacBio CLR
LAST alignment to reference genome
(a) Alignment to reference genome
Reference genome
Alignment file
tandem-genotypes
(b) Detection of tandem repeat expansions for a single sample
Repeat annotation file
VNTR information including repeat copy number changes
List with detailed VNTR information
Guided detection of VNTR expansions

Figure 6: Flowchart of the VNTR-detection pipeline. The pipeline takes PacBio CLR data as input and generates a file containing VNTR information, among which repeat copy number changes i.e. VNTR expansions with respect to a reference genome. The pipeline requires that a reference genome and a repeat annotation file is supplied.

3. RESULTS

The results presented in this section are outlined in the same order as was done in the discussion of the pipelines in Section 2.3., presenting the results of each step from the haplotype-reconstruction pipeline first. Each result refers to the flowchart of the pipelines depicted in Figures 5-6, using the same step names e.g. (a), (b), (c), etc. as was used in Section 2.3. As for the results from the haplotype-reconstruction pipeline, the results are outlined such that the results of each step concerning a specific operation e.g. alignment, assembly, haplotype phasing, are clustered together.

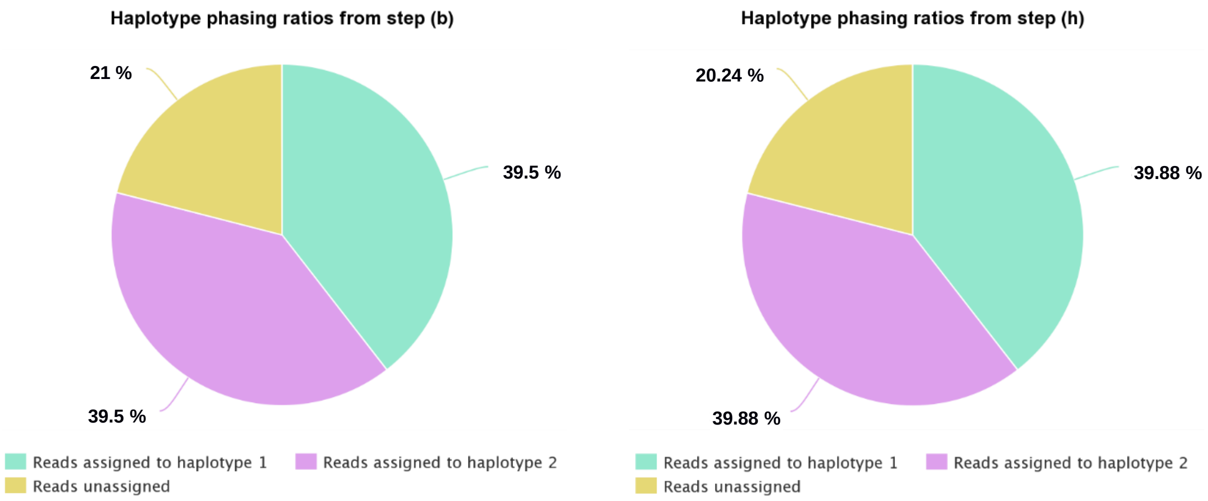
3.1. Haplotype-reconstruction pipeline

Results for steps (a), (d) and (g). After the alignment was finished in step (a) presented by Figure 5, 87,95% of all subreads from the open-source PacBio (CLR) HG002 human dataset, were mapped to GRCh38 as depicted in Table 3. Note that as a consequence, 12.05% of the subreads were discarded at this point, having no contribution in step (b), since these reads didn't align against the reference. With regards to the alignment performed in step (d), the amount of mapped reads was lower compared to that of step (a), having 83,72% of the subreads mapped to either one of the two haplotypes. As for the alignment result of step (g), more reads got aligned compared to the alignment of step (a), discarding 11,60% of the subreads at this point. An evaluation in IGV showed that the alignment-file obtained in step (a) contained mapped reads that encompassed the VNTR in the intronic region of *ABCA7* (not visualized).

Table 3: Global statistics of the alignment performed by minimap2, aligning all PacBio CLR subreads from the open-source PacBio HG002 dataset[28] against GRCh38 and its own assembly in step (a) and (g) in Figure 5, respectively.

PacBio human dataset HG002	Global statistics of alignment		
	Step (a)	Step (d)	Step (g)
Raw total reads	5.683.490	5.683.490	5.683.490
Average raw read length	15.712	15.712	15.712
Reference genome	GRCh38	Haplotype 1 & 2	HG002 assembly
% mapped reads	87,95%	83,72%	88,40%
% unmapped reads	12,05%	16,28%	11,60%

Results for steps (b) and (h). Regarding the reads that did align against GRCh38 in step (a), these reads were assigned to either haplotype 1 or haplotype 2 during haplotype phasing by Longshot in step (b). The ratio of reads that were assigned to haplotype 1, haplotype 2 or stayed unassigned, are visualized in Figure 7a. As for the haplotype phasing that was performed in step (h), less reads stayed unassigned after haplotype phasing as compared to step (b).



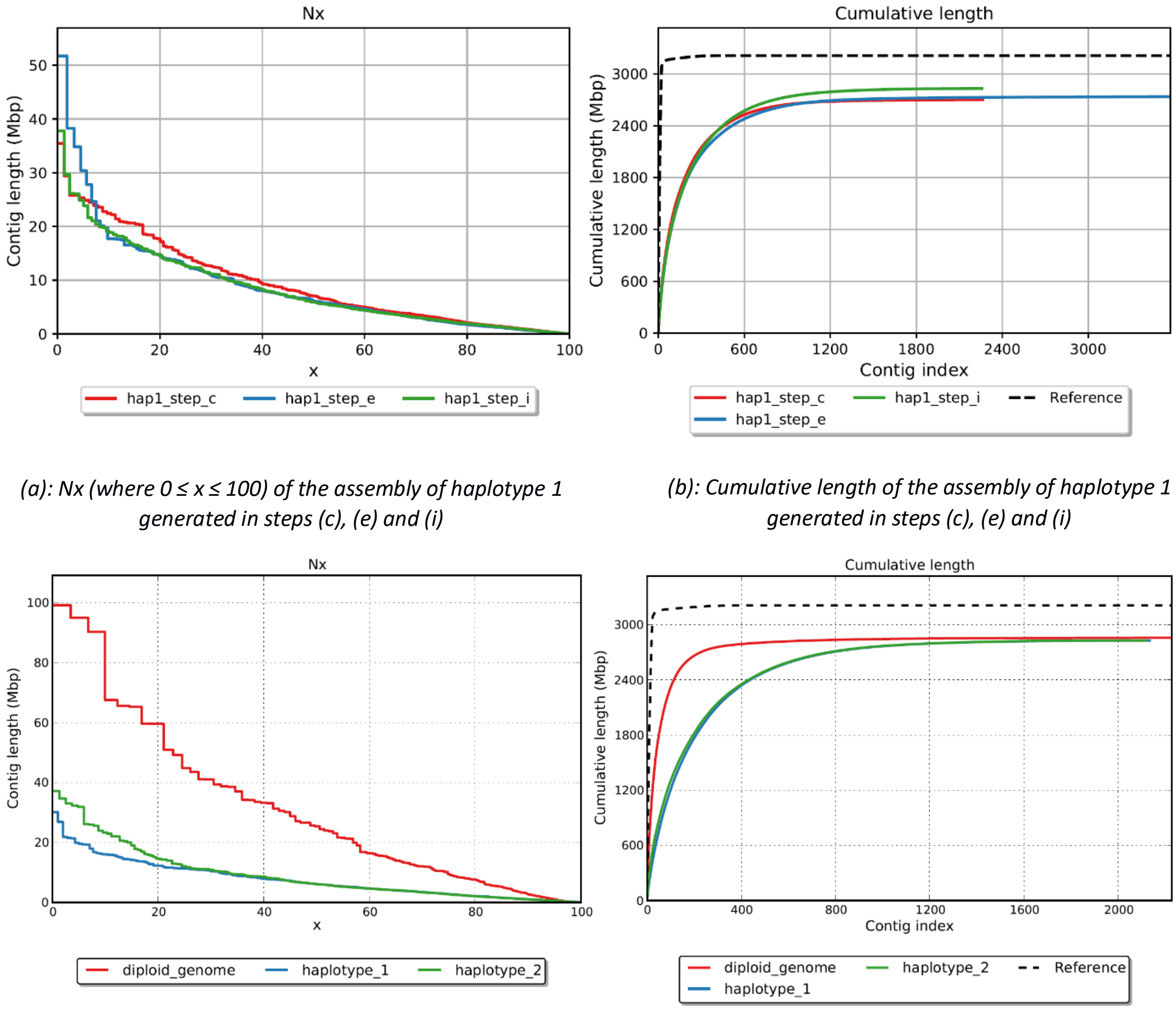
(a): Ratios of haplotype phasing in step (b)

(b): Ratios of haplotype phasing in step (h)

Figure 7: Haplotype-phasing ratios obtained with Longshot. The ratios represent the haplotype-phasing performance on the PacBio CLR subreads from HG002 in steps (b) (left panel) and (h) (right panel) in Figure 5. The results were interpreted as being better as the amount of reads that stayed unassigned was lower and the reads that got assigned were distributed evenly over the two haplotypes.

Results of steps (c) and (e). The quality metrics of haplotype 1 generated by steps (c) and (e) are depicted in Figures 8a and 8b showing the N*x* (where 0 ≤ *x* ≤ 100) and cumulative length, respectively. To improve the clarity of the plots, and since the genome quality of haplotype 1 and 2 was similar, only one haplotype is shown. The N*x* of an assembly represents the quality of it, with N50 being the most used metric for assessing assemblies. Generally, a higher N50 indicates a better overall quality of the assembly. As Figure 8a shows, the assembly of haplotype 1 from step (c) had a N50 of 7,1 Mbp, whereas haplotype 1 from step (e) had a N50 of 6,1 Mbp. The N50 of haplotype 2 was the same as that of haplotype 1, for both steps (c) and (e). Figure 8b shows that the assembly of haplotype 1 for step (c) had a genome fraction of 87,82%, whereas the assembly for step (e) had a genome fraction of 88,45%. The genome fraction is the total number of aligned bases in the reference, divided by the genome size of the reference genome. The genome fraction for haplotype 2 was similar to that of haplotype 1, for both steps (c) and (e). For reference, the cumulative length of the assemblies were outlined against the reference genome GRCh38 presented by the dotted line, which used only 23 contigs to represent the whole genome, one contig for each chromosome. Note, that the assemblies of haplotype 1 and haplotype 2 from step (c) were generated using only a subset of all subreads from the PacBio HG002 human dataset. This subset of reads that was used to generate the assembly of haplotype 1 and 2, consisted of both unassigned reads and reads that were assigned to one of both haplotypes. Hence, 39,5 + 21 = 60,5% (see Figure 7a) of all subreads that mapped to the reference, were used to build the assembly of haplotype 1 and haplotype 2 in step (c). In Figures 8a-b, the assembly of haplotype 1 for step (i) is shown as well, for comparison.

Results of steps (f) and (i). Figures 8c and 8d show the performance of the assembly-step in steps (f) and (i), depicting the N*x* (where 0 ≤ *x* ≤ 100) and cumulative length of the assemblies they generated, respectively. Concerning the performance of Flye in step (f), the diploid assembly had a N50 of 25,4 Mbp and genome fraction of 91,90%. As for the performance of step (i), haplotype 1 and haplotype 2 had a N50 of 5,9 and 6,2 Mbp, respectively. The genome fraction was 91,27% for the assembly of haplotype 1 and 91,36% for that of haplotype 2. The number of mismatches and indels found in the assembly of haplotype 1 obtained in steps (c), (e) and (i) is collected in Table 4, showing that step (i) generated the haploid assembly with the least amount of mismatches and indels with respect to GRCh38.



(a): Nx (where 0 ≤ x ≤ 100) of the assembly of haplotype 1 generated in steps (c), (e) and (i)

(b): Cumulative length of the assembly of haplotype 1 generated in steps (c), (e) and (i)

(c): Nx (where 0 ≤ x ≤ 100) of the obtained diploid genome, haplotype 1 and haplotype 2 from step (f) and (i)

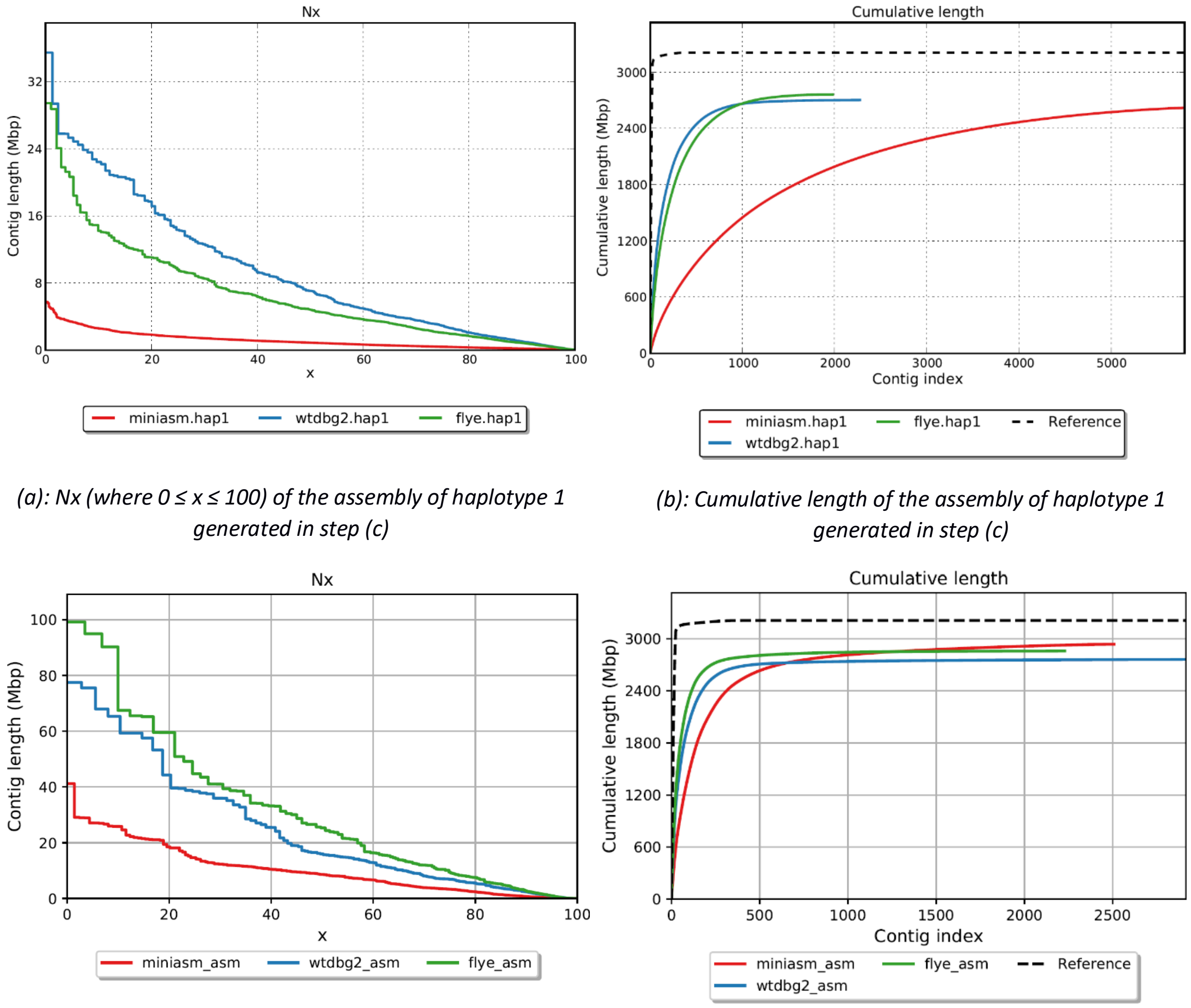
(d): Cumulative length of the obtained diploid genome, haplotype 1 and haplotype 2 from step (f) and (i)

Figure 8: Graphs visualizing the Nx (where 0 ≤ *x* ≤ 100) and cumulative length for the assemblies obtained in steps (c), (e), (f) and (i) of the haplotype-reconstruction pipeline in Figure 5. The genome assembly quality was assessed using QUAST. An overall higher Nx (where 0 ≤ *x* ≤ 100) indicates better chromosome contiguity and, consequently, better assembly quality. The N50 value is the standard metric for evaluating the quality of assemblies. Regarding Figures 8b and 8d, the x-axis is the index of all contigs sorted from the largest to the smallest, from left to right. Generally, the assembly that has the higher cumulative length comprises most of the genome. For reference, the cumulative length of the assemblies were outlined against the reference genome GRCh38 presented by the dotted line. The PacBio CLR data used for creating the assemblies came from HG002. **Figures 8a-b** depict the assembly quality for haplotype 1 obtained in steps (c) and (e) using wtdbg2, and the assembly quality of haplotype 1 obtained in step (i) using Flye. **Figures 8c-d** depict the assembly quality for the diploid genome as well as for both haplotypes obtained in steps (f) and (i), respectively, using Flye. Note, that the blue line that represents haplotype 1 in Figure 8d is overshadowed by the line that represents haplotype 2.

Table 4: Mismatches found in the assembly of haplotype 1 obtained in steps (c), (e) and (i) in Figure 5. # Mismatches is the number of mismatches in all aligned bases. # Indels is the number of indels in all aligned bases. These results were obtained by aligning the contigs of each assembly to GRCh38. The colored cells represent the relative best and worst scores, given by the colors blue and red, respectively. The white cells represent the median.

Mismatches	Hap1_step_c	Hap1_step_e	Hap1_step_i
# mismatches	3.965.348	3.938.092	3.641.818
# indels	4.563.164	3.955.910	2.888.308
# indels (<= 5 bp)	4.434.286	3.829.041	2.751.689
# indels (> 5bp)	128.878	126.869	136.619

Result regarding the comparison of assemblers for steps (c) and (f). In order to determine what assembler to use for step (c) and step (f), a modest comparison has been made between miniasm, wtdbg2 and Flye regarding overall assembly quality. The performance of each of the tested assembler is depicted in Figure 9. Figures 9a-b show the comparison that was made in step (c), representing the performance of miniasm, wtdbg2 and Flye in the reference-based approach. Figures 9c-d show the comparison that was made in step (f), representing the performance of miniasm, wtdbg2 and Flye in the de-novo-based approach. In Figure 9a, wtdbg2 created the assembly for haplotype 1 with the overall higher N*x* compared to the assembly generated by miniasm and Flye, with a N50 of 7,1 Mbp. In Figure 9b, Flye obtained the greatest assembly size, with a genome fraction 89,85%. In Figure 9c, the assembly that Flye created had the overall higher N*x* compared to wtdbg2 and miniasm, with having a N50 of 25,4 Mbp. Miniasm and wtdbg2 had a N50 of 8,6 and 16,0 Mbp, respectively. In Figure 9d, the assembly generated by miniasm was the greatest in assembly size, with a genome fraction of 92,74%. A report containing supplementary quality metrics regarding differences in performance between the tested assemblers can be found in Appendix C.



(a): Nx (where 0 ≤ x ≤ 100) of the assembly of haplotype 1 generated in step (c)

(b): Cumulative length of the assembly of haplotype 1 generated in step (c)

(c): Nx (where 0 ≤ x ≤ 100) of the diploid genome assembly generated in step (f)

(d): Cumulative length of the diploid genome assembly generated in step (f)

Figure 9: Graphs visualizing the Nx (where 0 ≤ *x* ≤ 100) and cumulative length of the assemblers miniasm, wtdbg2 and Flye in steps (c) and (f) in Figure 5, comparing the performance of the assemblers with each other. The genome assembly quality was assessed using QUAST. An overall higher Nx (where 0 ≤ *x* ≤ 100) indicates better chromosome contiguity and, consequently, better assembly quality. The N50 value is the standard metric for evaluating the quality of assemblies. Regarding Figures 9b and 9d, the x-axis is the index of all contigs sorted from the largest to the smallest, from left to right. Generally, the assembly that has the higher cumulative length comprises most of the genome. For reference, the cumulative length of the assemblies were outlined against the reference genome GRCh38 presented by the dotted line. The PacBio CLR data used for creating the assemblies came from HG002.

Result regarding overall speed and resource efficiency. Table 5 shows the resource usage for the haplotype-reconstruction pipeline in Figure 5. What stands out is that the de-novo-based approach ran slightly faster compared to the reference-based approach, given the amount of threads and memory allocated for each step. The assembly steps, i.e. steps (c), (e), (f) and (i), required the greatest amount of memory, with up to 500Gb RAM for Flye in step (f). The wall-clock time measures the actual amount of time taken to perform a job. The assembly steps were the most time-consuming steps in the haplotype-reconstruction pipeline.

Table 5: Resource usage of the haplotype-reconstruction pipeline. The resources used by each step within the haplotype-reconstruction pipeline are presented with the corresponding dataset and the amount of threads that were being used.

Step	Description	Dataset	GSize	Cov	threads	Wall-clock time	RAM
			Reference-based approach				
(a)	Alignment to ref	Human HG002	3Gb	PB x30	20	13h 30m	20Gb
(b)	Haplotype phasing	Human HG002	3Gb	PB x30	70	14h	20Gb
(c)	Assembly 1st (hap)	Human HG002	3Gb	PB x30	24	34h 30m	130Gb
(d)	Alignment to haploid	Human HG002	3Gb	PB x30	50	13h	20Gb
(e)	Assembly 2nd (hap)	Human HG002	3Gb	PB x30	24	88h	130Gb
			De-novo-based approach				
(f)	Assembly (diploid)	Human HG002	3Gb	PB x30	36	75h	500Gb
(g)	Alignment to asm	Human HG002	3Gb	PB x30	20	5h	20Gb
(h)	Haplotype phasing	Human HG002	3Gb	PB x30	70	14h	20Gb
(i)	Assembly (haploids)	Human HG002	3Gb	PB x30	50	42h	350Gb

3.2. VNTR-detection pipeline

The output of the VNTR-detection pipeline i.e. the repeat copy number changes of HG002 with respect to GRCh38, is visualized in Figure 10. Of all tandem repeats that were specified in the supplied repeat annotation file from the UCSC genome database, 16 with relatively large expansions were randomly selected and plotted as example. No VNTRs associated with AD were selected, as these were not present in the repeat annotation file. Each plot represents a tandem repeat from the supplied annotation file, with each histogram bar representing the amount of subreads that contained a specific repeat copy number change with respect to GRCh38. The validation on how representative the predictions for repeat copy number changes were, was based on how clear two alleles could be distinguished when examining the repeat copy number changes.

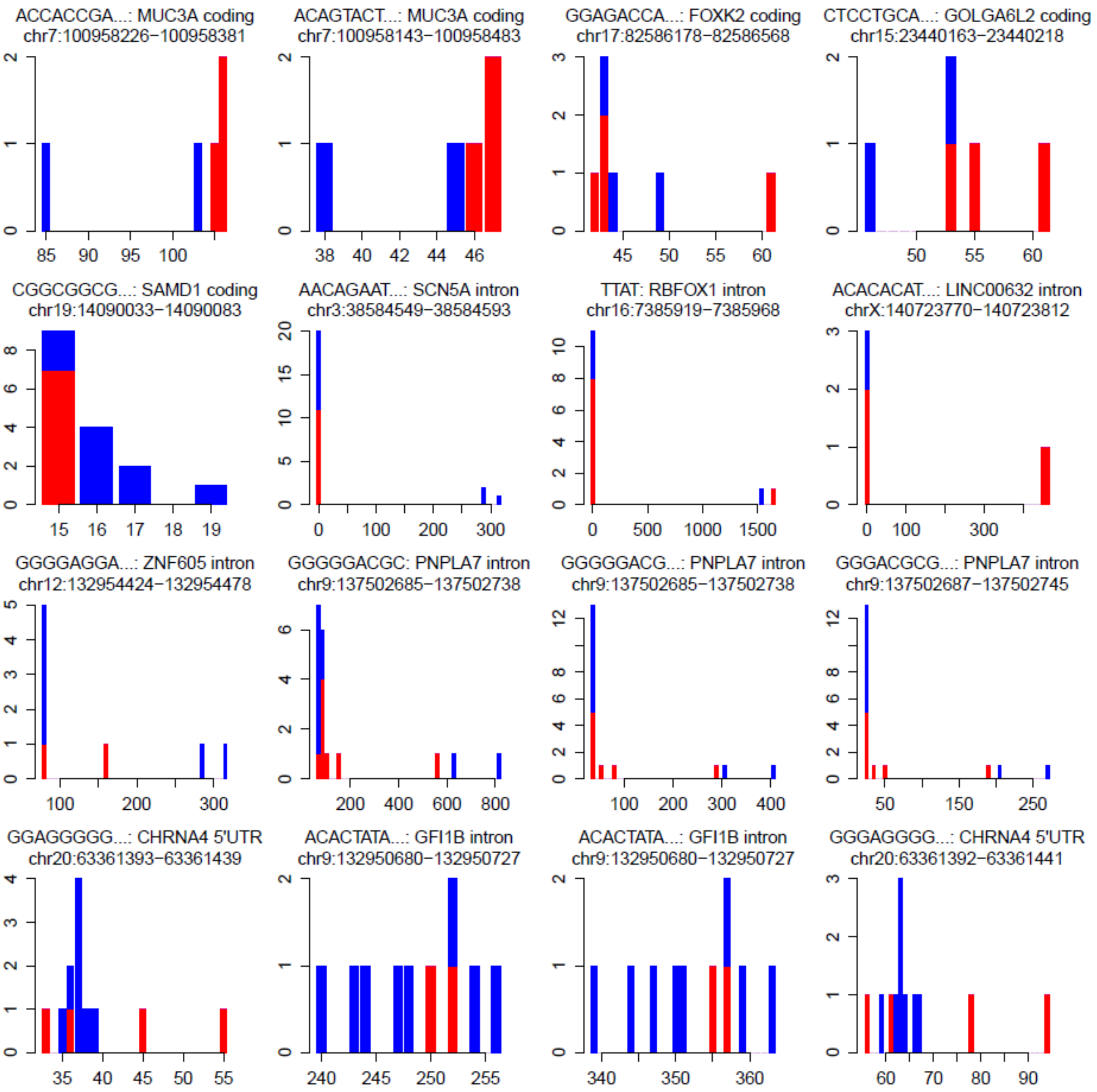


Figure 10: Repeat copy number changes in HG002 with respect to GRCh38. 16 randomly selected tandem repeats from the supplied repeat annotation file from the UCSC genome database are plotted. Each plot represents one tandem repeat that is expanded with respect to GRCh38. The x-axis shows the amount of repeat copy number changes with respect to GRCh38 and the y-axis shows the number of subreads on which the concerning repeat was found. Each histogram bar is divided in two colored parts, with red indicating the number of forward-strand reads, and blue indicating the number of reverse-strand reads.

4. DISCUSSION

4.1. Haplotype-reconstruction pipeline

One part of the project aim was to design, develop and validate bioinformatic pipelines that could reconstruct haplotypes from PacBio CLR sequence data of diploid genomes. Additionally, the pipelines had to be relatively fast in analyzing a human genome. For this, two pipeline approaches were designed and developed in order to see which of the two generates the overall better haploid assemblies for VNTR detection: a reference-based and a de-novo-based approach. The reference-based and the de-novo-based approach differ mainly in two aspects, that is, they use a different assembler and they incorporate different ways in which they assign reads to one or the other haplotype. Note, that the assembly and haplotype phasing steps influence the accuracy of the resulting haplotype representation the most as well.

4.1.1. Assembly steps

The number of large indels contained in an assembly could give an approximation of the amount of VNTRs that assembly contains. As Table 4 showed, the de-novo-based approach generated the assembly with the most indels with length >5bp, indicating the de-novo-based approach generates the haplotype-specific assemblies that are most useful for VNTR detection. However, not all large structural variations are enclosed by large indels. Therefore, the haploid assemblies generated by the reference-based approach may be more useful for detecting other large structural variations, as these haploid assemblies had the highest N50 and hence, were the most contiguous. Other studies obtained a N50 of at most 6.3 Mbp for haplotype-specific assemblies[46], suggesting that the performance of the haplotype-reconstruction pipeline is valid. More samples have to be processed in order to determine whether or not these results are consistent for each sample.

The diploid assembly obtained by the de-novo-based approach had a N50 comparable to that obtained by other studies[46], which suggests that the diploid genome assembly step is valid. The assemblies generated by the diploid genome assembly step can therefore not only act as an intermediate product, but be used in follow-up analysis considering both VNTR-related and non VNTR-related research as well, though it doesn’t contain information on the haplotype-level. A N50 of 29.3 Mbp could be achieved with CLR data from the human CHM13 genome (accession GCA_002884485.1) using the Falcon[47] assembler[21], suggesting that even higher assembly quality can be obtained. However, Falcon and other assemblers like Canu[48] are 4-5 times slower compared to wtdbg2 and Flye[21], and therefore no good alternatives, as the pipelines are meant to be used on hundreds of genomes.

Better assemblies can be obtained if short read data would be available, so that additional polishing with accurate short reads could be carried out. Alternatively, the genomes that have to be analyzed could be sequenced such that PacBio high-fidelity (HiFi) reads are generated instead of CLR, as HiFi reads were shown to enable assemblers to both be faster and obtain generally superior or at least comparable assemblies.[21] Additionally, it is likely that HiFi assemblers will improve in the future, as PacBio HiFi reads become more used.[21] However, not many studies have compared the difference between the CLR and HiFi data types in detecting VNTRs. It is shown that using HiFi reads with an average length of 10.9 kbp enables better or at least comparable VNTR-resolution in assemblies compared to CLR reads with an average length of 17.5 kbp.[21] But since CLR reads can have a length up to >50kb, it is not clear whether or not HiFi reads are as beneficial for VNTR-detection as CLR reads with longer lengths.

4.1.2. Haplotype phasing

Both approaches had a similar phasing performance. However, approximately 20% of the mapped reads stayed unassigned. Normally, this is caused by reads that doesn’t contain enough SNPs. However, it could be that these reads stayed unassigned due to the relatively high error rates that interfere during SNP calling. Currently, there aren’t any haplotype phasing tools that circumvent this problem by phasing on both SNPs and structural variations.[27] If either short read data was available or a rapid pipeline was not necessary, then assembly-based haplotype phasing (e.g. Falcon-unzip[47], Canu-trio binning[49]) could be considered, as this type of phasing is generally more accurate.[27]

4.1.3. Alignment steps

The performance of the alignment steps from both approaches were quite similar, as the same aligner (Minimap2) was being used in each alignment step. As mentioned before, Minimap2 is many times faster and more accurate in mapping long reads to a large reference genome.[32] Minimap2 was therefore being used in each alignment step, instead of other long read aligners like BWA-MEM[35] and BLASR[36]. The alignment steps are considered valid, as IGV showed that Minimap2 succeeded to map reads to disease-associated regions that generally are difficult to identify.

4.1.4. Speed and resources

The amount of time each approach used to process the human HG002 genome was approximately 5.5 (de-novo-based) and 6.5 days (reference-based). Note, that only one of the approaches have to be used for each sample. The required time to process one sample can easily be reduced by allocating more threads and memory. However, the assembly steps demand enormous amounts of memory as the number of threads gets bigger. Alternatively, if the pipeline is used on a private cluster, it doesn't share the CPU with other users and hence, processes samples faster.

4.2. VNTR-detection pipeline

The VNTR-detection pipeline predicts VNTR expansions directly from the unassembled reads by measuring the repeat-length on reads that encompasses repeats that are contained within a supplied repeat annotation file. However, this approach doesn't detect novel VNTRs and doesn't link VNTRs to a specific haplotype. The pipeline is useful in comparing VNTR expansions between groups nonetheless. The performance of the VNTR-detection pipeline could be validated further by analyzing more samples. The pipeline could be used on HiFi data, in order to see whether or not more repeats could be detected and whether or not distinct alleles are more pronounced in the output, compared to using CLR. The pipeline could be extended by implementing ways such that VNTR expansions could be predicted for multiple samples at once. In this way, the VNTR expansions associated with one group are immediately clustered together, which helps analysis in which VNTR expansions are compared between two groups.

5. CONCLUSION

The main goal of this project was to design, develop and validate bioinformatic pipelines that can assess VNTR expansions in PacBio CLR data from diploid genomes. This goal has been achieved by developing a Snakemake pipeline that detects VNTRs and predicts VNTR expansions directly from unassembled reads. From the predicted VNTR expansions in PacBio CLR data of the well-characterized and publicly available human HG002 genome, two alleles could be distinguished approximately, indicating that the VNTR assessment was sensible to differences in VNTR-length between haplotypes. The observed sensibility suggests that the pipeline enables accurate VNTR assessment even from long read sequence data with high error rates. The pipeline could be extended by allowing more samples to be processed at once, so that the predicted VNTR expansions could represent those associated with one group. In this way, comparisons of VNTR expansions between two groups are more easily carried out.

The secondary goal was to design, develop and validate bioinformatic pipelines that are able to reconstruct haplotypes from noisy long reads of diploid genomes. For this, two pipeline designs have been developed that both generate haplotype-specific assemblies, either guided by a reference genome (reference-based approach) or without any need for a reference (de-novo-based approach). Both pipeline designs were validated on PacBio CLR data of the human HG002 genome, which showed that the de-novo-based approach generated the best haploid assemblies for VNTR-detection. However, more samples have to be processed in order to determine whether or not these results are consistent. Concerning follow-up research: the pipeline could be used on PacBio's high-fidelity reads in order to determine whether or not high-fidelity reads aid the reconstruction of haplotypes. One way to expand the pipeline is to implement ways to call large structural variations on each obtained haplotype, so that further analysis is possible concerning disease-associated structural variants.

REFERENCES

1. Holstege H, Beker N, Dijkstra T, et al. The 100-plus Study of cognitively healthy centenarians: rationale, design and cohort description. *bioRxiv*. September 2018:295287. doi:10.1101/295287

2. Corrada MM, Brookmeyer R, Paganini-Hill A, Berlau D, Kawas CH. Dementia incidence continues to increase with age in the oldest old the 90+ study. *Ann Neurol*. 2010;67(1):114-121. doi:10.1002/ana.21915

3. Livingston G, Sommerlad A, Orgeta V, et al. Dementia prevention, intervention, and care. *Lancet*. 2017;390(10113):2673-2734. doi:10.1016/S0140-6736(17)31363-6

4. Roeck A De, Coster W De, Bossaerts L, et al. Accurate characterization of expanded tandem repeat length and sequence through whole genome long-read sequencing on PromethION. *bioRxiv*. November 2018:439026. doi:10.1101/439026

5. De Roeck A, Van den Bossche T, van der Zee J, et al. Deleterious ABCA7 mutations and transcript rescue mechanisms in early onset Alzheimer's disease. *Acta Neuropathol*. 2017;134(3):475-487. doi:10.1007/s00401-017-1714-x

6. Van den Bossche T, Sleegers K, Cuyvers E, et al. Phenotypic characteristics of Alzheimer patients carrying an ABCA7 mutation. *Neurology*. 2016;86(23):2126-2133. doi:10.1212/WNL.0000000000002628

7. De Roeck A, Van Broeckhoven C, Sleegers K. The role of ABCA7 in Alzheimer's disease: evidence from genomics, transcriptomics and methylomics. *Acta Neuropathol*. 2019;138(2):201-220. doi:10.1007/s00401-019-01994-1

8. De Roeck A, Duchateau L, Van Dongen J, et al. An intronic VNTR affects splicing of ABCA7 and increases risk of Alzheimer's disease. *Acta Neuropathol*. 2018;135(6):827-837. doi:10.1007/s00401-018-1841-z

9. Cuyvers E, De Roeck A, Van den Bossche T, et al. Mutations in ABCA7 in a Belgian cohort of Alzheimer's disease patients: a targeted resequencing study. *Lancet Neurol*. 2015;14(8):814-822. doi:10.1016/S1474-4422(15)00133-7

10. Sleegers K, Van Broeckhoven C. Novel Alzheimer's disease risk genes: exhaustive investigation is paramount. *Acta Neuropathol*. 2019;138(2):171-172. doi:10.1007/s00401-019-02041-9

11. Campion D, Charbonnier C, Nicolas G. SORL1 genetic variants and Alzheimer disease risk: a literature review and meta-analysis of sequencing data. *Acta Neuropathol*. 2019;138(2):173-186. doi:10.1007/s00401-019-01991-4

12. Estus S, Shaw BC, Devanney N, Katsumata Y, Press EE, Fardo DW. Evaluation of CD33 as a genetic risk factor for Alzheimer's disease. *Acta Neuropathol*. 2019;138(2):187-199. doi:10.1007/s00401-019-02000-4

13. Rojas I de, Moreno-Grau S, Tesi N, et al. Common variants in Alzheimer's disease: Novel association of six genetic variants with AD and risk stratification by polygenic risk scores. *medRxiv*. January 2020:19012021. doi:10.1101/19012021

14. Kunkle BW, Grenier-Boley B, Sims R, et al. Genetic meta-analysis of diagnosed Alzheimer's disease identifies new risk loci and implicates Aβ, tau, immunity and lipid processing. *Nat Genet*. 2019;51(3):414-430. doi:10.1038/s41588-019-0358-2

15. Gemayel R, Vinces MD, Legendre M, Verstrepen KJ. Variable Tandem Repeats Accelerate Evolution of Coding and Regulatory Sequences. *Annu Rev Genet*. 2010;44(1):445-477.

doi:10.1146/annurev-genet-072610-155046

16. De Roeck A, Duchateau L, Van Dongen J, et al. IN-DEPTH ANALYSIS OF AN ABCA7 VNTR IN ALZHEIMER'S DISEASE. *Alzheimer's Dement*. 2018;14(7):P1400. doi:10.1016/j.jalz.2018.06.2909

17. Variable Number Tandem Repeat - an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/neuroscience/variable-number-tandem-repeat. Accessed April 30, 2020.

18. Eid J, Fehr A, Gray J, et al. Real-time DNA sequencing from single polymerase molecules. *Science*. 2009;323(5910):133-138. doi:10.1126/science.1162986

19. Mikheyev AS, Tin MMY. A first look at the Oxford Nanopore MinION sequencer. *Mol Ecol Resour*. 2014;14(6):1097-1102. doi:10.1111/1755-0998.12324

20. Wenger AM, Peluso P, Rowell WJ, et al. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol*. August 2019:1-8. doi:10.1038/s41587-019-0217-9

21. Vollger M, Logsdon G, Audano P, et al. Improved assembly and variant detection of a haploid human genome using single-molecule, high-fidelity long reads. *bioRxiv*. August 2019:635037. doi:10.1101/635037

22. HiFi Reads with CCS - SMRT Sequencing Modes - PacBio. https://www.pacb.com/smrt-science/smrt-sequencing/smrt-sequencing-modes/. Accessed February 20, 2020.

23. DNA Forensics Problem Set. http://www.biology.arizona.edu/human_bio/problem_sets/DNA_forensics_1/05t.html. Accessed February 20, 2020.

24. Mitsuhashi S, Matsumoto N. Long-read sequencing for rare human genetic diseases. *J Hum Genet*. 2020;65(1):11-19. doi:10.1038/s10038-019-0671-8

25. Tourdot RW, Zhang C-Z. Whole Chromosome Haplotype Phasing from Long-Range Sequencing. *bioRxiv*. May 2019:629337. doi:10.1101/629337

26. Roach JC, Glusman G, Hubley R, et al. Chromosomal haplotypes by genetic phasing of human families. *Am J Hum Genet*. 2011;89(3):382-397. doi:10.1016/j.ajhg.2011.07.023

27. Zhang X, Wu R, Wang Y, Yu J, Tang H. Unzipping haplotypes in diploid and polyploid genomes. *Comput Struct Biotechnol J*. 2020;18:66-72. doi:10.1016/j.csbj.2019.11.011

28. HG002 Structural Variant Analysis with CLR data · PacificBiosciences/DevNet Wiki · GitHub. https://github.com/PacificBiosciences/DevNet/wiki/HG002-Structural-Variant-Analysis-with-CLR-data. Accessed December 9, 2019.

29. HPC Cluster. https://login.hpc.tudelft.nl/. Accessed December 9, 2019.

30. Koster J, Rahmann S. Snakemake--a scalable bioinformatics workflow engine. *Bioinformatics*. 2012;28(19):2520-2522. doi:10.1093/bioinformatics/bts480

31. nickradunovic/longread-analysis. https://github.com/nickradunovic/longread-analysis. Accessed April 30, 2020.

32. Li H. Minimap2: pairwise alignment for nucleotide sequences. Birol I, ed. *Bioinformatics*. 2018;34(18):3094-3100. doi:10.1093/bioinformatics/bty191

33. Edge P, Bansal V. Longshot: accurate variant calling in diploid genomes using single-molecule

long read sequencing. *bioRxiv*. March 2019:564443. doi:10.1101/564443

34. Ruan J, Li H. Fast and accurate long-read assembly with wtdbg2. *Nat Methods*. 2019;17(2):155-158. doi:10.1038/s41592-019-0669-3

35. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. March 2013. http://arxiv.org/abs/1303.3997. Accessed April 26, 2020.

36. Chaisson MJ, Tesler G. Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): Application and theory. *BMC Bioinformatics*. 2012;13(1):1-18. doi:10.1186/1471-2105-13-238

37. Robinson JT, Thorvaldsdóttir H, Wenger AM, Zehir A, Mesirov JP. Variant review with the integrative genomics viewer. *Cancer Res*. 2017;77(21):e31-e34. doi:10.1158/0008-5472.CAN-17-0337

38. GitHub - samtools/samtools: Tools (written in C using htslib) for manipulating next-generation sequencing data. https://github.com/samtools/samtools. Accessed April 8, 2020.

39. Schneider V, Graves-Lindsay T, Howe K, et al. Evaluation of GRCh38 and de novo haploid genome assemblies demonstrates the enduring quality of the reference assembly. *Eval GRCh38 novo haploid genome Assem Demonstr Endur Qual Ref Assem*. 2016:072116. doi:10.1101/072116

40. Gurevich A, Saveliev V, Vyahhi N, Tesler G. Genome analysis QUAST: quality assessment tool for genome assemblies. 2013;29(8):1072-1075. doi:10.1093/bioinformatics/btt086

41. Vaser R, Sović I, Nagarajan N, Šikić M. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res*. 2017;27(5):737-746. doi:10.1101/gr.214270.116

42. LAST: genome-scale sequence comparison. http://last.cbrc.jp/. Accessed April 18, 2020.

43. Mitsuhashi S, Frith MC, Mizuguchi T, et al. Tandem-genotypes: robust detection of tandem repeat expansions from long DNA reads. *Genome Biol*. 2019;20(1):58. doi:10.1186/s13059-019-1667-6

44. Haeussler M, Zweig AS, Tyner C, et al. The UCSC Genome Browser database: 2019 update. *Nucleic Acids Res*. 2018;47:853-858. doi:10.1093/nar/gky1095

45. Benson G. Tandem repeats finder: A program to analyze DNA sequences. *Nucleic Acids Res*. 1999;27(2):573-580. doi:10.1093/nar/27.2.573

46. Chaisson MJP, Sanders AD, Zhao X, et al. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun*. 2019;10(1):1-16. doi:10.1038/s41467-018-08148-z

47. Chin CS, Peluso P, Sedlazeck FJ, et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat Methods*. 2016;13(12):1050-1054. doi:10.1038/nmeth.4035

48. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: Scalable and accurate long-read assembly via adaptive κ-mer weighting and repeat separation. *Genome Res*. 2017;27(5):722-736. doi:10.1101/gr.215087.116

49. Koren S, Rhie A, Walenz BP, et al. De novo assembly of haplotype-resolved genomes with trio binning. *Nat Biotechnol*. 2018;36(12):1174-1182. doi:10.1038/nbt.4277

50. GitHub - jasperlinthorst/reveal: Graph based multi genome aligner. https://github.com/jasperlinthorst/reveal. Accessed September 17, 2019.

51. A proposal of the Grapical Fragment Assembly format. https://lh3.github.io/2014/07/19/a-proposal-of-the-grapical-fragment-assembly-format. Accessed April 19, 2020.

DATA AND SOURCE CODE AVAILABILITY

Data availability. HG002 CLR reads were acquired from the PacBio repository. https://downloads.pacbcloud.com/public/dataset/SV-HG002-CLR/.

Code availability. The whole pipeline is available at https://github.com/nickradunovic/longread-analysis.

APPENDIX A: INFORMATION REGARDING THE HG002 (CLR) DATASET FROM THE PACBIO REPOSITORY

Table 6 shows statistics on the quality and methods concerning the PacBio open-repository dataset sequenced on the PacBio Sequel II System. The sample: GIAB HG002 extracted DNA.

Table 6: Information on library prep and sequencing of the HG002 PacBio open-repository dataset which was used to develop and validate the pipelines. Additional information including the actual dataset can be found in the PacBio repository.[28]

METHODS
Shearing 75 kb with Megaruptor
Library prep SMRTbell Express 2.0
Size selection >30 kb with BluePippin
Sequencing PacBio Sequel II System with "Early Access" binding kit (101-490-800) and chemistry (101-490-900)
Run time 15 hrs per SMRT Cell

APPENDIX B: STRUCTURAL VARIANT CALLING STAGE

This pipeline stage wasn't fully tested and therefore not included in this work. However, the pipeline stage was designed and developed nonetheless. In the section below, the workings of the pipeline are outlined.

Structural variant calling pipeline. The second part of the haplotype-reconstruction pipeline represents the structural variant calling stage, as was visualized by the flowchart in Figure 4. The workflow of the structural variant calling stage is depicted in Figure 11. The structural variant calling stage makes solely use of the tool REVEAL[50], which can make graph-based comparisons of multiple *de novo* assembled genomes. First, the contigs from the consensus haplotypes that came from the haplotype-reconstruction stage, enter the structural variant calling stage where they are transformed into a graph-based representation using the Graphical Fragment Alignment (GFA) format[51]. Second, multiple graph-based genomes can be compared to each other by means of an alignment in which the genomes of interest are aligned to each other. This multi-alignment step can be performed using at least two genomes, for example a haplotype-specific assembly of interest and a reference genome like GRCh38, or by specifying multiple genomes in GFA format. The resulting alignment is an anchored graph in GFA format, from which variants could be extracted in the end.

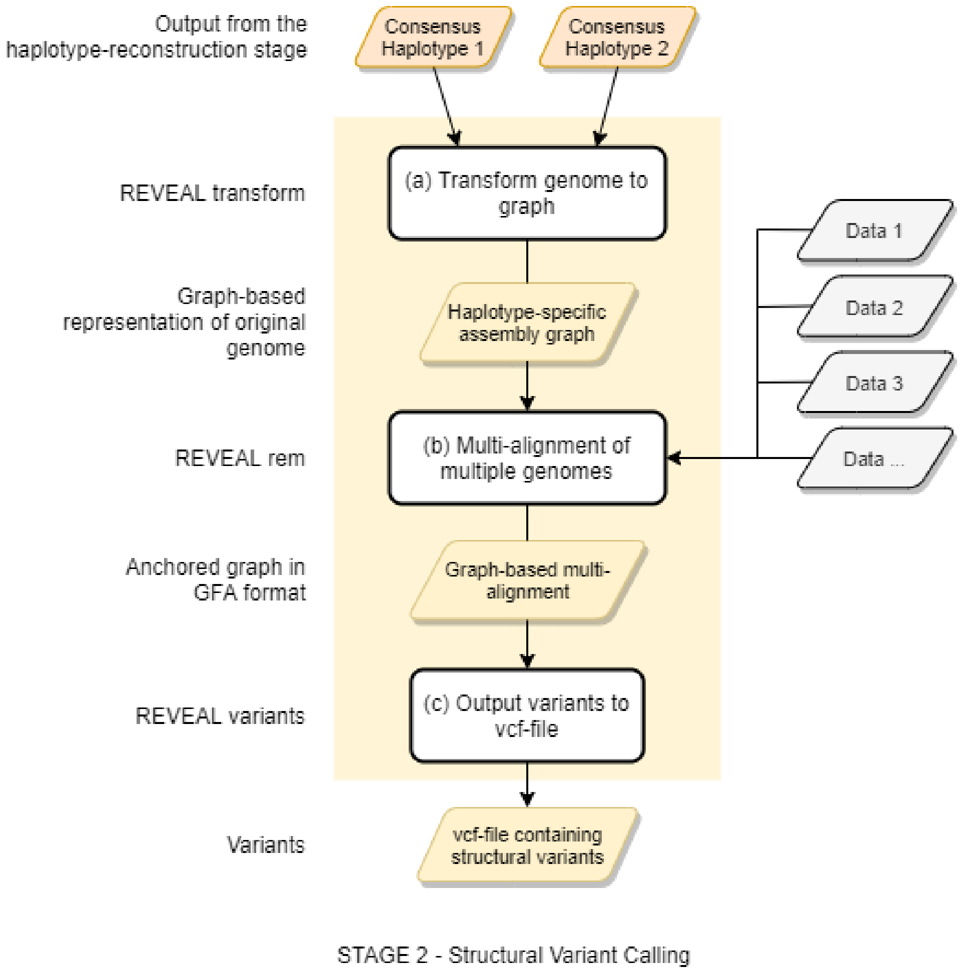
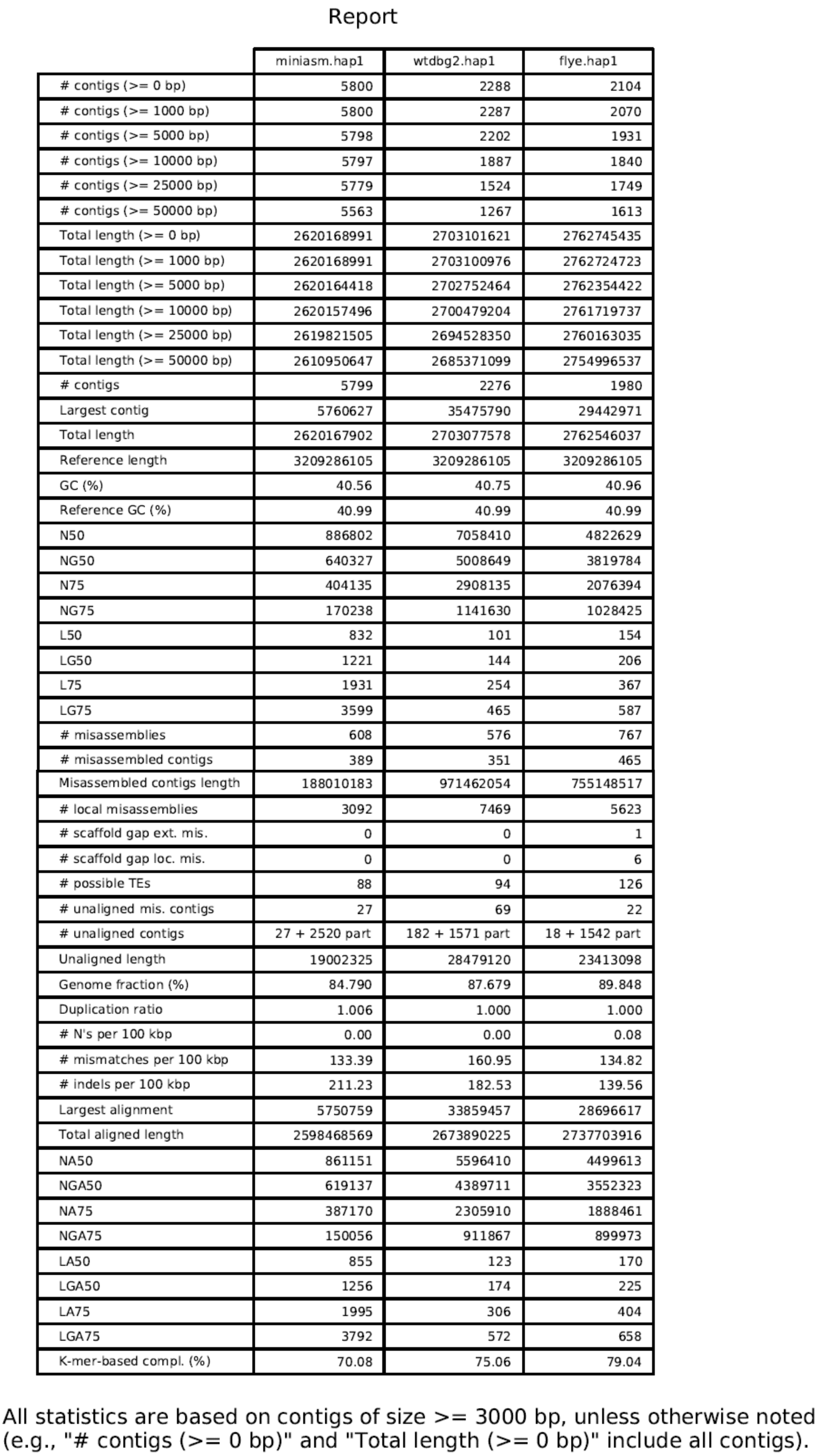

Output from the haplotype-reconstruction stage
Consensus Haplotype 1
Consensus Haplotype 2
REVEAL transform
(a) Transform genome to graph
Graph-based representation of original genome
Haplotype-specific assembly graph
Data 1
Data 2
Data 3
Data ...
REVEAL rem
(b) Multi-alignment of multiple genomes
Anchored graph in GFA format
Graph-based multi-alignment
REVEAL variants
(c) Output variants to vcf-file
Variants
vcf-file containing structural variants
STAGE 2 - Structural Variant Calling

Figure 11: Flowchart of the Structural Variant Calling pipeline. The consensus haplotypes obtained from either the reference-based or the de-novo-based approach, function as an input for the variant calling stage. The output is a vcf-file containing variants for both haplotype 1 and haplotype 2. The data that is necessary in step (b) represent graph-based genomes in GFA format to which the haplotypes align. Either one genome, such as a reference genome, or multiple genomes could be supplied. Each step a-c in the structural variant calling stage uses the tool REVEAL[50].

APPENDIX C: ASSEMBLY QUALITY REPORT FOR THE PERFORMANCE COMPARISON OF THE ASSEMBLERS

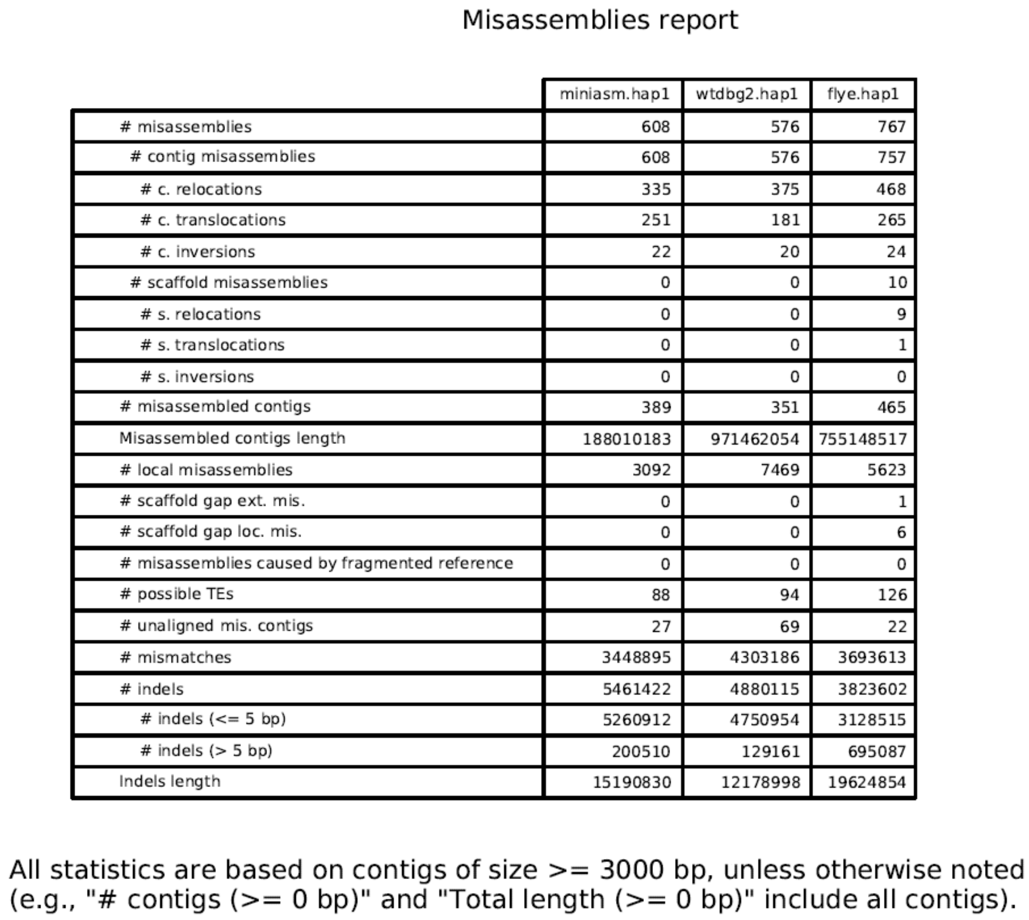
Assembly quality report of the assemblies created with miniasm, wtdbg2 and Flye in step (c). The report was generated with QUAST. Metrics on misassemblies and unaligned contigs are included.



Report

	miniasm.hap1	wtdbg2.hap1	flye.hap1
# contigs (>= 0 bp)	5800	2288	2104
# contigs (>= 1000 bp)	5800	2287	2070
# contigs (>= 5000 bp)	5798	2202	1931
# contigs (>= 10000 bp)	5797	1887	1840
# contigs (>= 25000 bp)	5779	1524	1749
# contigs (>= 50000 bp)	5563	1267	1613
Total length (>= 0 bp)	2620168991	2703101621	2762745435
Total length (>= 1000 bp)	2620168991	2703100976	2762724723
Total length (>= 5000 bp)	2620164418	2702752464	2762354422
Total length (>= 10000 bp)	2620157496	2700479204	2761719737
Total length (>= 25000 bp)	2619821505	2694528350	2760163035
Total length (>= 50000 bp)	2610950647	2685371099	2754996537
# contigs	5799	2276	1980
Largest contig	5760627	35475790	29442971
Total length	2620167902	2703077578	2762546037
Reference length	3209286105	3209286105	3209286105
GC (%)	40.56	40.75	40.96
Reference GC (%)	40.99	40.99	40.99
N50	886802	7058410	4822629
NG50	640327	5008649	3819784
N75	404135	2908135	2076394
NG75	170238	1141630	1028425
L50	832	101	154
LG50	1221	144	206
L75	1931	254	367
LG75	3599	465	587
# misassemblies	608	576	767
# misassembled contigs	389	351	465
Misassembled contigs length	188010183	971462054	755148517
# local misassemblies	3092	7469	5623
# scaffold gap ext. mis.	0	0	1
# scaffold gap loc. mis.	0	0	6
# possible TEs	88	94	126
# unaligned mis. contigs	27	69	22
# unaligned contigs	27 + 2520 part	182 + 1571 part	18 + 1542 part
Unaligned length	19002325	28479120	23413098
Genome fraction (%)	84.790	87.679	89.848
Duplication ratio	1.006	1.000	1.000
# N's per 100 kbp	0.00	0.00	0.08
# mismatches per 100 kbp	133.39	160.95	134.82
# indels per 100 kbp	211.23	182.53	139.56
Largest alignment	5750759	33859457	28696617
Total aligned length	2598468569	2673890225	2737703916
NA50	861151	5596410	4499613
NGA50	619137	4389711	3552323
NA75	387170	2305910	1888461
NGA75	150056	911867	899973
LA50	855	123	170
LGA50	1256	174	225
LA75	1995	306	404
LGA75	3792	572	658
K-mer-based compl. (%)	70.08	75.06	79.04

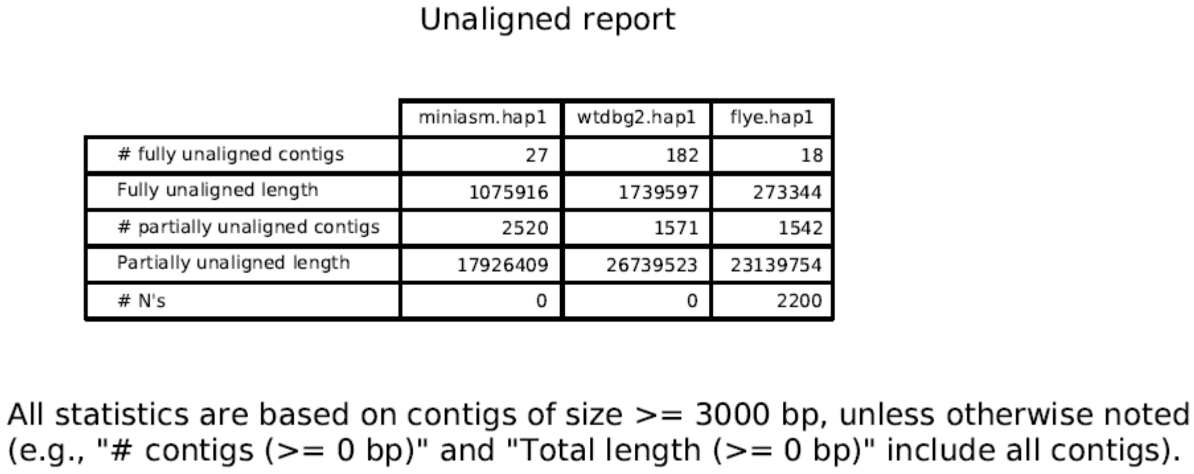
All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).



Misassemblies report

	miniasm.hap1	wtdbg2.hap1	flye.hap1
# misassemblies	608	576	767
# contig misassemblies	608	576	757
# c. relocations	335	375	468
# c. translocations	251	181	265
# c. inversions	22	20	24
# scaffold misassemblies	0	0	10
# s. relocations	0	0	9
# s. translocations	0	0	1
# s. inversions	0	0	0
# misassembled contigs	389	351	465
Misassembled contigs length	188010183	971462054	755148517
# local misassemblies	3092	7469	5623
# scaffold gap ext. mis.	0	0	1
# scaffold gap loc. mis.	0	0	6
# misassemblies caused by fragmented reference	0	0	0
# possible TEs	88	94	126
# unaligned mis. contigs	27	69	22
# mismatches	3448895	4303186	3693613
# indels	5461422	4880115	3823602
# indels (<= 5 bp)	5260912	4750954	3128515
# indels (> 5 bp)	200510	129161	695087
Indels length	15190830	12178998	19624854

All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

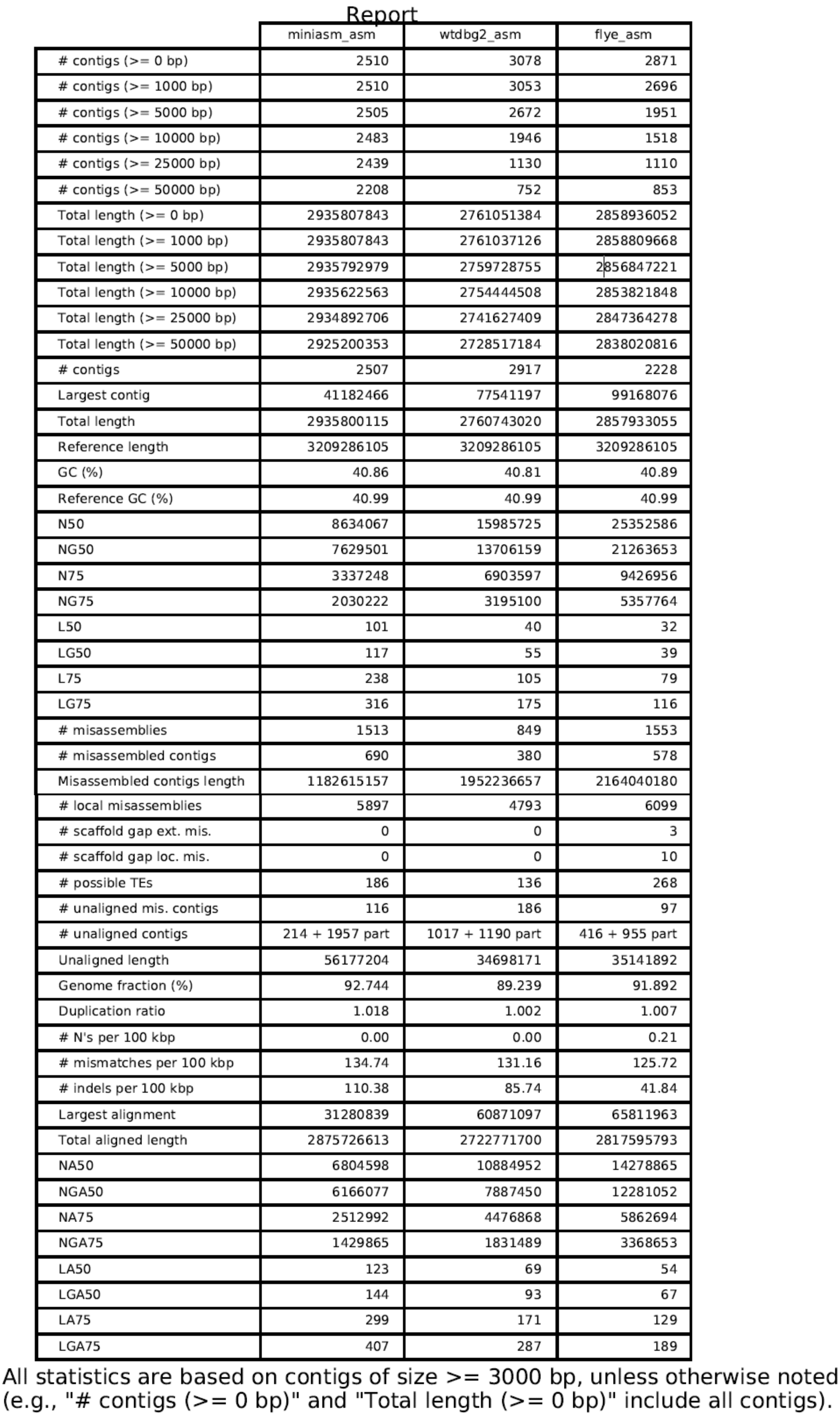


Unaligned report

	miniasm.hap1	wtdbg2.hap1	flye.hap1
# fully unaligned contigs	27	182	18
Fully unaligned length	1075916	1739597	273344
# partially unaligned contigs	2520	1571	1542
Partially unaligned length	17926409	26739523	23139754
# N's	0	0	2200

All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

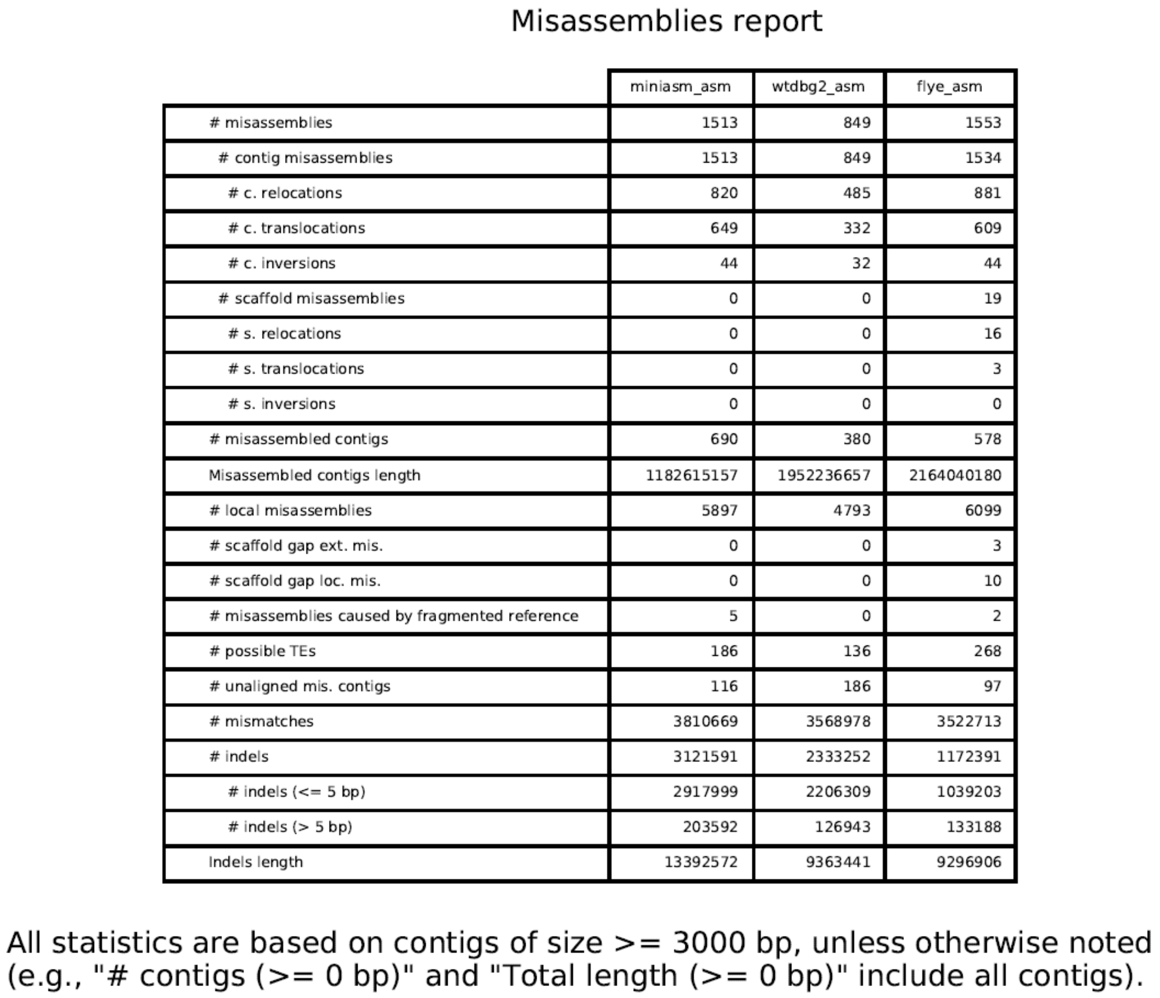
Assembly quality report of the assemblies created with miniasm, wtdbg2 and Flye in step (f). The report was generated with QUAST. Metrics on misassemblies and unaligned contigs are included.



Report

	miniasm_asm	wtdbg2_asm	flye_asm
# contigs (>= 0 bp)	2510	3078	2871
# contigs (>= 1000 bp)	2510	3053	2696
# contigs (>= 5000 bp)	2505	2672	1951
# contigs (>= 10000 bp)	2483	1946	1518
# contigs (>= 25000 bp)	2439	1130	1110
# contigs (>= 50000 bp)	2208	752	853
Total length (>= 0 bp)	2935807843	2761051384	2858936052
Total length (>= 1000 bp)	2935807843	2761037126	2858809668
Total length (>= 5000 bp)	2935792979	2759728755	2856847221
Total length (>= 10000 bp)	2935622563	2754444508	2853821848
Total length (>= 25000 bp)	2934892706	2741627409	2847364278
Total length (>= 50000 bp)	2925200353	2728517184	2838020816
# contigs	2507	2917	2228
Largest contig	41182466	77541197	99168076
Total length	2935800115	2760743020	2857933055
Reference length	3209286105	3209286105	3209286105
GC (%)	40.86	40.81	40.89
Reference GC (%)	40.99	40.99	40.99
N50	8634067	15985725	25352586
NG50	7629501	13706159	21263653
N75	3337248	6903597	9426956
NG75	2030222	3195100	5357764
L50	101	40	32
LG50	117	55	39
L75	238	105	79
LG75	316	175	116
# misassemblies	1513	849	1553
# misassembled contigs	690	380	578
Misassembled contigs length	1182615157	1952236657	2164040180
# local misassemblies	5897	4793	6099
# scaffold gap ext. mis.	0	0	3
# scaffold gap loc. mis.	0	0	10
# possible TEs	186	136	268
# unaligned mis. contigs	116	186	97
# unaligned contigs	214 + 1957 part	1017 + 1190 part	416 + 955 part
Unaligned length	56177204	34698171	35141892
Genome fraction (%)	92.744	89.239	91.892
Duplication ratio	1.018	1.002	1.007
# N's per 100 kbp	0.00	0.00	0.21
# mismatches per 100 kbp	134.74	131.16	125.72
# indels per 100 kbp	110.38	85.74	41.84
Largest alignment	31280839	60871097	65811963
Total aligned length	2875726613	2722771700	2817595793
NA50	6804598	10884952	14278865
NGA50	6166077	7887450	12281052
NA75	2512992	4476868	5862694
NGA75	1429865	1831489	3368653
LA50	123	69	54
LGA50	144	93	67
LA75	299	171	129
LGA75	407	287	189

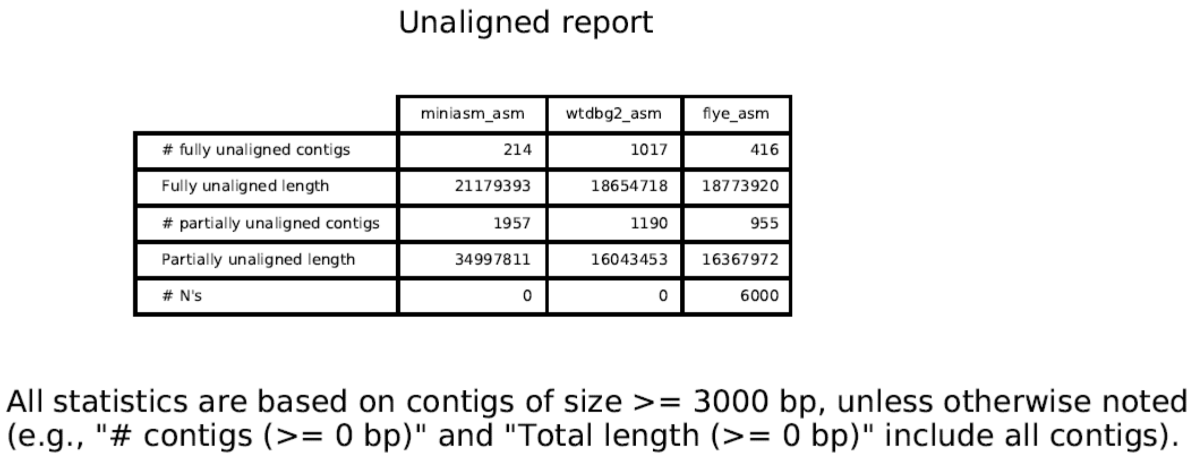
All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).



Misassemblies report

	miniasm_asm	wtdbg2_asm	flye_asm
# misassemblies	1513	849	1553
# contig misassemblies	1513	849	1534
# c. relocations	820	485	881
# c. translocations	649	332	609
# c. inversions	44	32	44
# scaffold misassemblies	0	0	19
# s. relocations	0	0	16
# s. translocations	0	0	3
# s. inversions	0	0	0
# misassembled contigs	690	380	578
Misassembled contigs length	1182615157	1952236657	2164040180
# local misassemblies	5897	4793	6099
# scaffold gap ext. mis.	0	0	3
# scaffold gap loc. mis.	0	0	10
# misassemblies caused by fragmented reference	5	0	2
# possible TEs	186	136	268
# unaligned mis. contigs	116	186	97
# mismatches	3810669	3568978	3522713
# indels	3121591	2333252	1172391
# indels (<= 5 bp)	2917999	2206309	1039203
# indels (> 5 bp)	203592	126943	133188
Indels length	13392572	9363441	9296906

All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).



Unaligned report

	miniasm_asm	wtdbg2_asm	flye_asm
# fully unaligned contigs	214	1017	416
Fully unaligned length	21179393	18654718	18773920
# partially unaligned contigs	1957	1190	955
Partially unaligned length	34997811	16043453	16367972
# N's	0	0	6000

All statistics are based on contigs of size >= 3000 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).