De novo assembly of the rooibos genome

Author: Allison Anne Stander

A thesis submitted in partial fulfilment of the requirements for the degree of Magister Scientiae in the Department of Biotechnology, University of the Western Cape.

Supervisor: Dr. Uljana Hesse

July 2020

De novo assembly of the rooibos genome

Allison Anne Stander

KEYWORDS
South Africa
Medicinal plants
Rooibos
Next-generation sequencing
De novo genome assembly
Assembly evaluation
Genome characteristics
K-mer analysis

ABSTRACT

***De novo* assembly of the rooibos genome**

A.A. Stander

MSc Thesis Department of Biotechnology, University of the Western Cape

Rooibos (*Aspalathus linearis*) is endemic to the Cederberg region of South Africa, and one of the few indigenous medicinal plants commercially cultivated in the country. International interest in rooibos is growing, and currently most of the rooibos harvest is exported overseas to more than 30 countries. Various problems hamper the growth of the rooibos industry, including insect pests, diseases, drought and a decreasing lifespan of the plants. The availability of whole-genome data for rooibos can contribute to the selection of genetically superior plants, facilitating not only the identification of important genes and metabolic pathways in rooibos, but also the establishment of breeding programs. In previous studies, the rooibos genome size had been estimated using flow cytometry, and the genome had been sequenced using Illumina sequencing technologies. In total, 331 billion reads were generated from one paired-end, small insert library (300 bp) and two mate pair libraries (insert sizes 3 kb and 8 kb). This thesis focused on the local establishment of biocomputational pipeline for plant genome analysis, including estimation of genome characteristics and plant genome assembly. Genome characteristics were investigated using five methods: GenomeScope (v1 and v2), FindGSE, BBNorm, KAT, and a standard formula. The results indicated a rooibos genome size of 1,03 ± 0,05 Gb, a high heterozygosity rate (2,09 ± 0,33) and a high repeat content (56,04 ± 8,51%). The computationally predicted genome size was comparable to the flow cytometry estimate of 1.24 ± 0.01 Gb, a result discussed and published in Mgwatyu & Stander et al. (2020). For genome assembly of the Illumina sequencing data, the following programs were evaluated in-depth: 1) FastQC, MultiQC and KAT for data quality assessment; 2) Trimmomatic, NextClip, FLASH for data quality processing; 3) ABySS 2.0, Platanus, SOAPdenovo2 for data assembly; and 4) KAT and QUAST-LG for evaluation of assembly quality. For the rooibos sequencing data, the assembly program Platanus yielded the best assembly. The N50 and N75 statistics amounted to 10 kb and 5.5 kb, respectively, and nine scaffolds were larger than 100 kbp. Moreover, BUSCO analyses indicated that 84% of conserved plant genes were present in the assembly (70% covered to completeness). Future steps to improve the rooibos genome assembly are discussed. This study, therefore, contributes to the establishment of plant genome research in South Africa.

DECLARATION

I declare that *De novo assembly of the rooibos genome* is my own work, that it has not been submitted for any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged by complete references.

Allison Anne Stander Date: 20 July 2020

Signed: ______________________

ACKNOWLEDGEMENTS

I would like to acknowledge the following people who all played integral parts towards the completion of this degree:

Dr Uljana Hesse, thank you for the constant support and guidance throughout my project, for always making sure I have funding, and for mentoring me to a better researcher.

Mr Eugene de Beste and Mr Peter van Heusden for technical support, installation of many, many tools, and answering any IT-related questions.

The CHPC and SANBI, for providing me with the needed resources and technical support. Charles Crosby for his time and effort to explain CPU efficiency to me.

Dr Gareth Linsmith and Dr Stephen Schlebusch for always being available, and keen, to answer any NGS-related questions.

Dr Emily Stander for introducing me to Dr Uljana Hesse, supporting me throughout my academic career, and inspiring me to be a better researcher.

Mr Morné Ziemkendorf for your support, patience, understanding, many laughs and love. For believing in me and encouraging me. And for the days that I could not help you cook or clean- you did it happily without any complaints.

My research team (Wesley, Yamkela, and Fanechka) for their constant support, laughs, and always being keen to help with any questions.

Jenny Webb and UKHC Genomics Core Laboratory for constructing and sequencing the genomic libraries.

Jamie Southgate and Michelle Livesey, for the chats, laughs, coffee breaks, and support. Julia Kotze, your support and humour got me through many days.

My mother, Ursula Nel, thank you for the time and effort you have put in me, for igniting and fuelling my passion for the natural sciences, for supporting and understanding my decisions, and for being the greatest example of perseverance.

Toby and Marmite, for providing me with many laughs and a lot of love.

The NRF and the South African Rooibos Council for funding this research.

For assistance in plant collection I would like to sincerely thank Rooibos LTD, specifically Johan Brand and Jan Perang; the Fair Wupperthal Rooibos Cooperative, specifically Edgar Valentyn; Darren Engelbrecht and Ruth Cuttings from the Department of Agriculture, Land Reform, and Rural Development (Northern Cape); as well as the following rooibos farmers: Adam and Johan Christiaans, Annes and Johan Brand, Carina Marais, Chrisjan Fortuin, Collin Fredericks, Danie Carstens, Davie Kritzinger, Deon Lambrechts, Edgar Valentyn, Frances Nel, Frans Du Plessis, Frans Kotze, Fritz Hanekam, Gerrie Olivier, Ian Brand, Johan and Rita Bruwer, Niklaas Kupido, Willie Nel.

Table of Contents

Chapter 1: Literature review

1.1. Background

Rooibos (*Aspalathus linearis*) is an important medicinal plant endemic to the Cederberg region of South Africa. Scientific research has proven beneficial effects of rooibos in the treatment of diverse ailments and diseases, including stomach pains, inflammation, diabetes, heart disease, HIV, and cancer (Millar et al., 2020). It is one of the few medicinal plants commercially cultivated in the country. Currently, 350 to 550 rooibos farms and eight large processing plants provide work for more than 5,000 people. These two industries represent the biggest employers in the Cederberg mountain region (Department of Agriculture and Forestry, 2015). Rooibos is best known as a herbal tea and is naturally caffeine-free and antioxidant-rich. It is widely used in the food, beverage, pharmaceutical and cosmetics industries. International interest in rooibos is growing, and currently, most of the rooibos harvest is exported overseas to more than 30 countries ("Industry Statistics | South African Rooibos Council," n.d.). Various problems hamper the growth of the rooibos industry, such as insect pests, diseases, a decreasing lifespan and drought stress. No selective breeding program has been applied in the rooibos industry, although a genetic improvement program for rooibos has been initiated (Bester et al., 2016). The availability of rooibos genome data could provide valuable information on genes and biosynthetic pathways associated with desirable phenotypic traits, which in turn could serve as biomarkers for targeted plant selection and rooibos breeding. In previous studies of the research team, the genome size of rooibos was predicted to be 1,2 Gb (estimated using flow cytometry; Mgwatyu et al., 2020). Based on this estimate, the genome had been sequenced using Illumina sequencing technologies to an approximate 259x genome coverage.

The aims of the project were to establish all essential biocomputational methods for the assembly of plant genomes at the University of the Western Cape and to generate a first assembly of the rooibos genome. To achieve this aim, the following objectives were set: 1) Compare different programs to evaluate the quality of short-read Illumina sequencing data. 2) compare different programs with various parameter settings for quality trimming and error correction of short-read Illumina sequencing data; 3) determine genome characteristics and verify rooibos genome size using k-mer analysis of Illumina sequencing data (essential to assess completeness of rooibos genome assembly); 4) identify assemblers suitable for analysing the rooibos genome Illumina sequencing data; and 5) evaluate assembly performance of three programs (ABySS 2.0, SOAPdenovo2, Platanus) using the short-read Illumina sequencing data.

1.2. Plant genome sequencing

The sequencing of plant genomes offers a wide range of applications, providing invaluable data resources for various research areas. Assembled plant genomes can be used to characterize genes involved in metabolic pathways, which produce valuable products such as secondary metabolites (Wang et al., 2019). These secondary metabolites, most often associated with plant adaptation, can be used to create pharmaceuticals, dyes, food additives, insecticides and other industrially relevant compounds. Knowing the biosynthetic pathway of specific metabolites allows us to recreate these metabolites synthetically (Hussain et al., 2012; Wang et al., 2019). It also allows us to genetically modify biosynthetic pathways in the plant to produce novel compounds. A famous example is the golden rice that produces vitamin A (Ye et al., 2000).

Genome sequence information is also useful to study crop traits for targeted plant breeding. Genomes provide information on molecular markers, which can be used to map agronomically important traits and to identify candidate genes associated with these traits. This permits marker-assisted plant breeding and facilitates the management of genetic resources (M. E. Bolger et al., 2014). Molecular markers are also used in phylogenetic and population-level studies (Li and Harkess, 2018) to investigate genomic variations such as changes in copy numbers, insertions/deletions, single nucleotide polymorphisms (SNPs) and repeats (Ekblom and Wolf, 2014). In comparative and evolutionary genomics, haplotype information and estimates of linkage disequilibrium on a genome-wide scale can reveal population histories and timing of admixture events (Ekblom and Wolf, 2014). In conservation biology, genetic markers provide a useful resource to assess biodiversity, demography, disease resistance and outbreaks, and taxonomy ("Conservation Genomics," n.d.; Ekblom and Wolf, 2014).

1.3. Genome sequencing technologies

Sequencing technologies have changed over time to the degree that we often speak of first-, second-, and now third-generation sequencing technologies. A summary is provided below.

First-generation sequencing methods were developed in the 1970s. The Maxam-Gilbert sequencing method (discontinued; Figure 1) involved chemical modification of the DNA (radiolabeling of the 5' end) and subsequent molecule cleavage into fragments of different lengths (Kchouk et al., 2017). Fragment cleavage was performed in four tubes, each with its specific enzyme that cut the DNA at a particular base (tube 1: A or G, tube 2: G, tube 3: C, tube 4: T or C). The cleaved fragments from each tube were subsequently run side by side on a polyacrylamide electrophoresis gel, the gel was exposed to an X-ray film to visualize the radiolabeled fragments, and the sequence was determined base by base based on the staggered fragments. The Sanger sequencing method (Figure 2) was much preferred over the Maxam-Gilbert method as it did not involve radioactive or toxic chemicals. It involves *de novo* synthesis of a second DNA strand and random termination of reactions using labelled dideoxynucleotide triphosphates (ddNTPs, which lack a 3'-OH group required to form another phosphodiester bond). Originally, the reactions were also performed in four tubes, each containing a primer, dNTPs, DNA polymerase, and one of the four ddNTPs at very low concentrations. Random incorporation of the ddNTP would terminate the reaction, creating a pool of fragments that vary in length. As before, the sequence of the fragments were determined using polyacrylamide gel electrophoresis and autoradiography. In modern Sanger sequencing, the ddNTPs are labelled with a fluorescent dye, and the reaction is completed in a single tube. The polyacrylamide gel electrophoresis is now replaced by capillary electrophoresis, where a Charged Coupled Device (CCD) reads the fluorescent signals. The Sanger sequencing method was used to sequence the first complete genome of an organism (bacteriophage phi X174;(Sanger et al., 1977)), and to generate the first draft of the human genome (International Human Genome Sequencing Consortium, 2001).

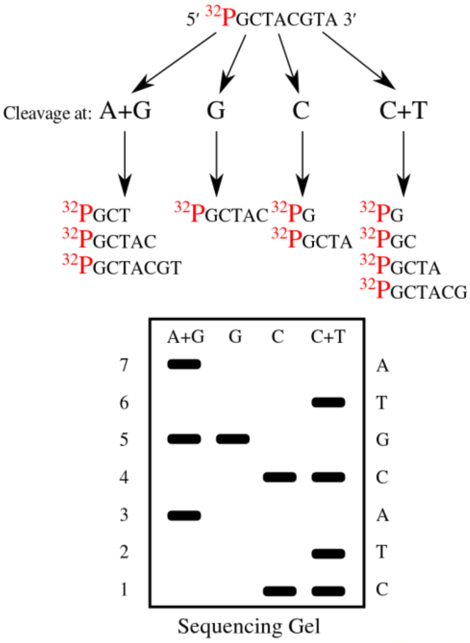


Figure 1: ***Maxam-Gilbert sequencing method*** *(Wikimedia Commons, 2013).*

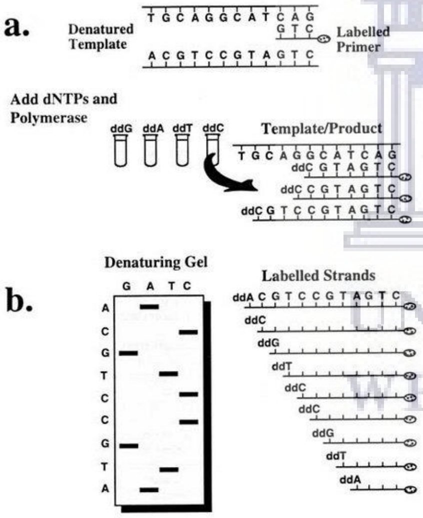


Figure 2: ***Sanger sequencing method*** *(Kchouk et al., 2017).*

The term Next Generation Sequencing (NGS) refers to massively parallel DNA sequencing technologies, which permit high-throughput analyses of samples at drastically reduced costs and shorter processing times. Second-generation sequencing generally refers to NGS methods that produce short length reads (50 bp–600 bp) and typically involve the preparation of PCR-amplified sequencing libraries before the actual sequencing of the DNA (Ambardar et al., 2016). Third-generation sequencing technologies include methods that permit real-time single-molecule sequencing (SMS) and produce very long reads (1 kbp–100 kbp), The first commercially available NGS technology, 454 sequencing, was launched in 2005. The method was initially licensed by 454 Life Sciences and later purchased by Roche. It successfully employed pyrosequencing – a sequencing-by-synthesis approach that permits determination of each specific nucleotide as it is incorporated into the complementary DNA strand during *de novo* synthesis of a dsDNA molecule. Release of the pyrophosphate would trigger a light signal that was captured for data analysis. Thousands of amplification reactions were performed simultaneously using emulsion PCR (ePCR); each reaction in a separate PCR droplet, surrounded by a hydrophobic organic phase (Kchouk et al., 2017). A high-quality sequencing run would produce up to 1 Million reads of approximately 700 bp (0.7 Gb) within just 24 hours (Liu et al., 2012). A major disadvantage of this technology was its inability to correctly interpret the length of homopolymer sequence runs that were longer than 8 bp. High equipment costs rendered the technology noncompetitive, and the 454 sequencing platform was discontinued in 2013 (Slatko et al., 2018).

The Ion Torrent is another second-generation sequencing platform, which was developed by Jonathan Rothberg (Slatko et al., 2018). Just like the 454 sequencing technologies, it makes use of pyrosequencing and ePCR. However, the method for nucleotide sequence determination is very different: using complementary metal-oxide-semiconductor technology (also employed in the manufacturing of microprocessor chips), the Ion Torrent measures differences in pH caused by the release of protons during DNA polymerization ("Ion Torrent Next-Generation Sequencing Technology," n.d.; Slatko et al., 2018). The Ion Torrent sequencer can produce reads of 200 bp to 600 bp in length, and has a maximum throughput of 10 GB within 2 to 8 hours (Kchouk et al., 2017)**.** Similar to the 454 DNA sequencers, this technology has problems with the interpretation of homopolymer sequence lengths (Slatko et al., 2018).

The currently most widely used method for DNA sequencing is Illumina sequencing, which is based on the Solexa method introduced in 2006. Illumina Inc. has become the market leader, the company holding 75% of the genetic sequencing market (Truong, 2020). The Solexa method involves sequencing by synthesis (Kchouk et al., 2017), which uses bridge amplification to generate clusters of DNA molecules on a flow cell (Figure 3A). Fluorescently labelled nucleotides are incorporated one-by-one, and the signal is detected by a CCD (Figure 3B). A number of sequencing machines are available, including the MiSeq and diverse HiSeq sequencers. The MiSeq sequencer is considered low-throughput, producing 15 GB within 4 to 55 hours, and achieving read lengths of 50 to 300 bp ("Illumina sequencing platforms," 2020).

The HiSeq platforms are high throughput, producing up to 1 TB of sequence (read lengths vary between 36 and 250 bp), which can take up to 11 days (“Performance specifications for the HiSeq 2500 System,” 2020). Since this method was used in this study to sequence the rooibos genome, all associated laboratorial and biocomputational procedures are discussed in detail in the chapters below.

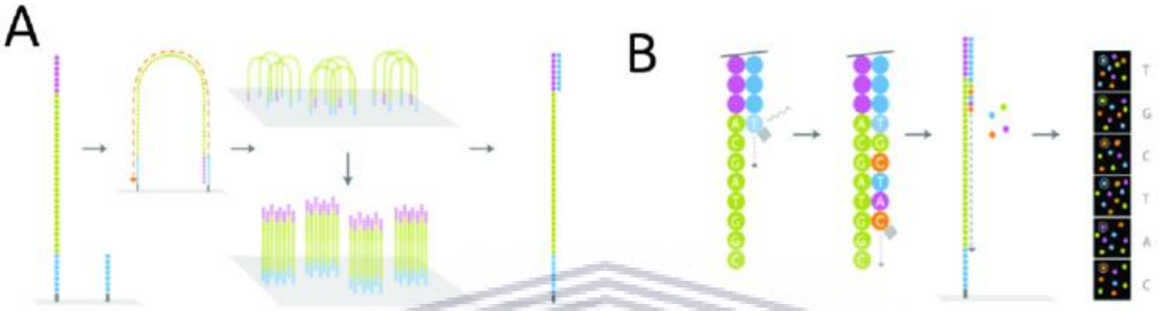


***Figure 3: A) Bridge amplification and B) Sequencing by synthesis** (Westbury, 2018).*

The above NGS methods were cost-efficient, permitting large-scale sequencing of genomes from all domains of life. The following basic workflow applies to all second-generation sequencing platforms: 1) DNA extraction and purification; 2) fragmentation of long DNA molecules (either chemically, enzymatically, or physically); 3) size-selection of the DNA fragments; 4) two-sided fragment tagging with sequencing adapters, which contain sequences that permit hybridization of the fragments to the sequencing platform (454 and Ion Torrent: beads, Illumina: flow cell); 5) clonal amplification of the fragments, completed using either ePCR (454, Ion Torrent) or by cluster generation (Illumina); and finally 6) Platform-dependent sequencing. The resulting reads are then reassembled into genomic sequences using biocomputational methods. A ballpark number for genome sequencing states that for accurate reassembly a genome coverage of approximately 100x must be achieved (Dominguez Del Angel et al., 2018; Ekblom and Wolf, 2014; Schatz et al., 2012). Considering read lengths of 150-300 bp, larger genomes (human: ~3,1 Gb, Pine: ~22 Gb), require the sequencing of billions of reads. Moreover, short read lengths have become a major limiting factor in correct genome reassembly: eukaryotic genomes (specifically those of plants) are often riddled with repeats that are longer than the maximum read lengths, which severely hampers correct reassembly of contiguous genomic sequences (Schatz et al., 2012). Diverse library preparation approaches have been developed to address long distance mapping of reassembled genomic sequences (e.g. increasing sequencing fragment length; preparation of cosmid, fosmid or BAC libraries with large insert sizes and subsequent clone by clone sequencing). However, these procedures are very laborious and severely error-prone. Third-generation sequencing technologies address the read length problem.

PacBio (Figure 4) is a third-generation sequencing platform that makes use of the single-molecule, real-time (SMRT) sequencing technology, which was commercialized in 2011 (Nakano et al., 2017). First, single-stranded hairpin adapters are ligated to both ends of linearized high molecular weight dsDNA molecules, circularizing these molecules and making them look like dumbbells. These sequencing templates are referred to as SMRTbells (Rhoads and Au, 2015; Slatko et al., 2018). The SMRTbell sample is loaded onto a SMRTcell, which contains 150,000 sequencing wells, called zero-mode waveguides (ZMW). The SMRTbell adaptor binds to a polymerase that is immobilized at the bottom of the ZMW (Rhoads and Au, 2015). The nucleotides, added for *de novo* synthesis of the DNA strands, are each labelled with a different fluorescent dye. As the nucleotides are incorporated, the dyes are released, and the signal is recorded as a movie of light pulses through the glass bottom of each ZMW (Rhoads and Au, 2015). PacBio offers longer read lengths (average read lengths over 10 kb), but its drawbacks are a higher error rate, lower throughput (0.5–1 billion bases per SMRT cell), and higher cost per base (Rhoads and Au, 2015; Slatko et al., 2018).

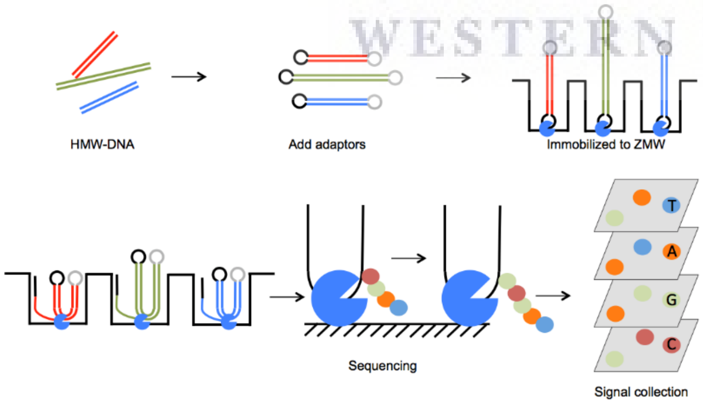


Figure 4: Outline of PacBio SMRT sequencing process *("NEXT GENERATION SEQUENCING," n.d.).*

Another sequencing method, introduced to the market as recently as 2015, was developed by Oxford Nanopore Technologies (ONT) (Slatko et al., 2018). As illustrated in Figure 5, sequencing is achieved while threading one strand of a double-stranded high molecular weight DNA molecule through a protein nanopore that perforates an electrically resistant polymer membrane (Slatko et al., 2018). Application of voltage induces a steady flow of ions through the nanopore. As the DNA strand passes through the nanopore, the nucleotides block the flow of ions, resulting in tiny changes in the electrical current. Since these changes are specific for each base, the signal can be translated back into the nucleotide sequence. Advantages of this method are the generation of long reads (up to 2 Mb), the ability to sequence RNA molecules directly, and the detection of base modifications which are important when studying gene expression and function (Heather and Chain, 2016). Moreover, the sequencing device is very small – as big as a jump-drive, weighing only 9g. However, the ONT platform shares the same drawback as PacBio: higher sequencing error rates and a lower throughput compared to SGS (Slatko et al., 2018).

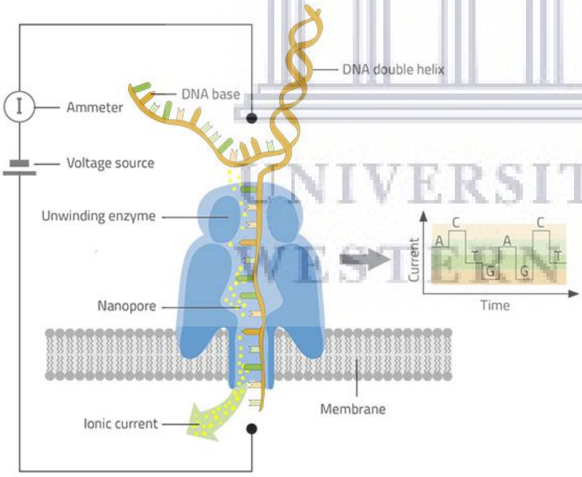


Figure 5: An example nanopore sequencer *(Göpfrich and Judge, 2018).*

1.4. Illumina sequencing

The Illumina sequencing method was used in this study to sequence the rooibos genome. Therefore, library preparation, the sequencing technology and all essential subsequent biocomputational data analysis methods require a more detailed review.

1.4.1. Library preparation

All second-generation sequencing platforms are limited in their ability to accommodate larger DNA fragments in their respective sequencing reactions. The maximum molecule size varies between the platforms (454: 400-1000 bp; Ion Torrent: 100-600 bp; Illumina: 100-1000 bp) ("454 (Roche)," n.d.; "Next Generation Sequencing / Whole Genome Sequencing," n.d.; Bronner et al., 2009). Depending on the size of the fragment, the essential steps for library preparation differ, resulting in "paired-end" and "mate pair" libraries. The Nextera library preparation protocols were used in this study, and will here serve as an example to explain all essential Illumina paired-end and mate pair library preparation steps.

The paired-end library preparation kits for Illumina (e.g., Nextera XT DNA Library Preparation Kit, TruSeq DNA Nano) serve to sequence the 5' and 3' ends from DNA fragments that are up to 1000 bp long. In the Nextera protocol, the first step is tagmentation. An engineered transposome fragments the DNA molecules and simultaneously tags both ends of each fragment with strand-specific adapters (P7 at the 5' end, P5 at the 3' end) (Figure 6A and B). The fragment size can be adjusted as it depends on the reaction parameters (e.g. temperature) and the ratio of transposase to sample DNA. Subsequently, the fragments are amplified using limited-cycle PCR, which permits end repair and addition of sequencing adapters that contain short (8 bp) index sequences (Figure 6C). Those sequencing adapters are needed to bind the PCR fragment to the oligonucleotides on the flow cell and to tag each fragment with a sample-specific index, which permits pooling of multiple samples on a flow cell (later, reads from any given sample can be recognized by their index). Subsequent steps include library purification, size selection, verification of the size distribution and template quantification. Different libraries can now be normalized and pooled together for sequencing ("Nextera DNA Library Prep Reference Guide," 2016).

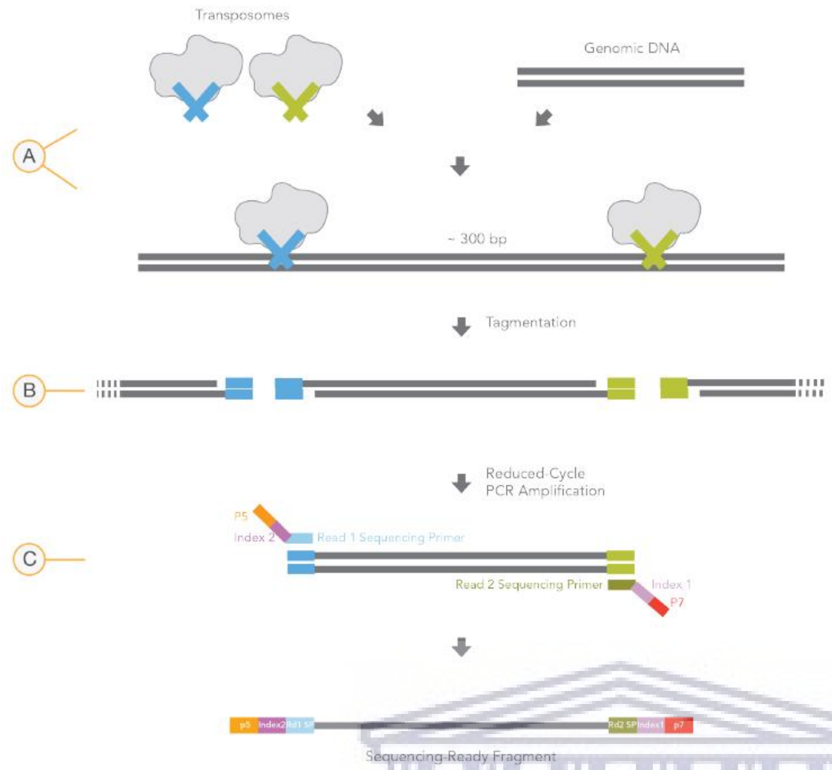


Figure 6: Steps in the Nextera paired-end library preparation protocol. *A) Template DNA bound with Nextera transposome and adapters,* ***B)*** *Tagmented DNA fragment with adapters added, and C) Limited cycle PCR to add index adapter sequences (“Nextera DNA Library Prep Reference Guide,” 2016).*

Mate pair libraries permit sequencing of the 3’ and 5’ ends from DNA fragments that are 2-10 kbp long (“Nextera® Mate Pair Library Preparation Kit: Datasheet,” 2014). Such long fragments cannot be sequenced directly. Therefore, the fragment ends are extracted by including the following additional steps into the library preparation protocols: 1) circularization of the large DNA fragments, 2) re-fragmentation of the circular DNA and 3) selection of sub-fragments that contain the ends of the original DNA fragments (Figure 7). The Nextera Mate Pair Library Preparation kit proceeds as follows. As before, long DNA molecules are tagmented into smaller pieces by an engineered transposome, except that this time the side-specific adapters are biotinylated. After size-selection, the fragments are circularized by joining the 3’ and 5’ ends of each fragment using biotin junction adapters. The circularized DNA is fragmented again, and the biotin tags are used to select those sub-fragments that include the 3’ and 5’ ends of the original DNA fragments (“Nextera® Mate Pair Library Prep Reference Guide,” 2016). Subsequent library preparation steps follow those described above.

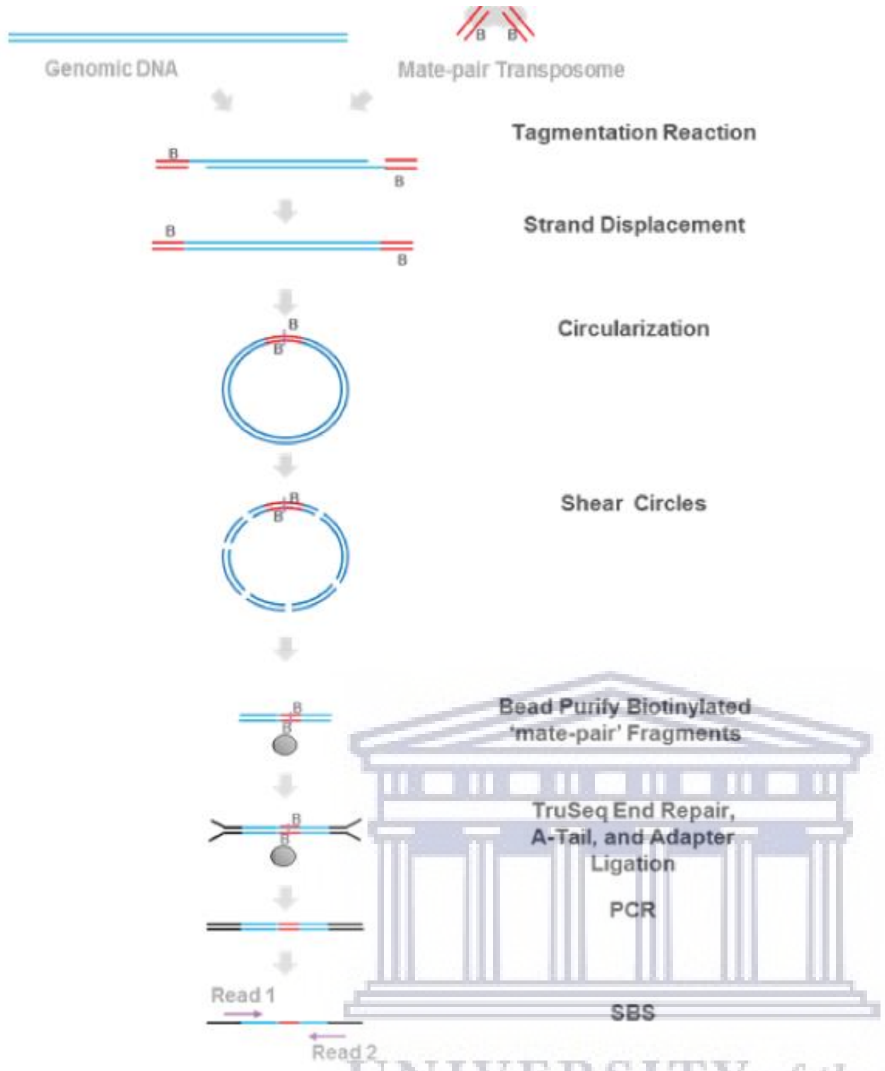


Figure 7: Nextera mate pair preparation kit procedure *("Nextera® Mate Pair Library Prep Reference Guide," 2016).*

In all Illumina sequencing runs, a PhiX spike-in is added to the reaction. A PhiX control is a concentrated Illumina library derived from the bacteriophage genome PhiX. It is added to Illumina sequencing runs for two main reasons: 1) To monitor the quality of the sequencing run and 2) to balance colour in low diversity libraries. It has an average size of 500 bp, and the percentage spike-in used per run depends on the sequencing platform (minimum 5% for MiSeq and HiSeq 2500) ("How much PhiX spike-in is recommended when sequencing low diversity libraries on Illumina platforms?," 2020; "What is the PhiX Control v3 Library and what is its function in Illumina Next Generation Sequencing?," 2020).

1.4.2. *Illumina cluster generation and sequencing*

Once the library is prepared, it is transferred onto an Illumina flow cell. A flow cell is a glass slide, about as small as a microscope slide (Figure 8). The flow cells are covered with two distinct, short, single-stranded (ss) oligonucleotides (referred to as P5 and P7 grafting adapters), which are complementary to the sequencing adapters used during library preparation (“Indexed Sequencing Overview Guide,” 2020; Launen, 2017). These oligonucleotides serve to bind the library fragments to the flow cell. The low-throughput MiSeq flow cells have uniform coverage; the HiSeq flow cells are divided into eight lanes, and the grafting adapters are attached to the bottom of those lanes. When using HiSeq, multiple different samples can be accommodated on one flow cell and even in one lane, as long as the libraries contain unique index sequences (molecular barcodes) to distinguish between samples (“Indexed Sequencing Overview Guide,” 2020; “Multiplexed Sequencing with the Illumina Genome Analyzer System,” 2008). During clonal amplification and sequencing, dNTPs and buffers are pumped through small channels in the flow cell (“Illumina Sequencing Technology,” 2010).

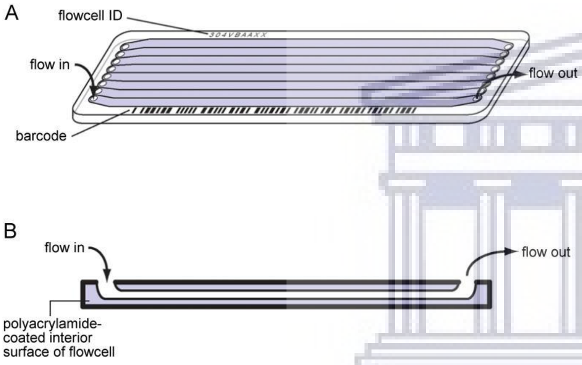
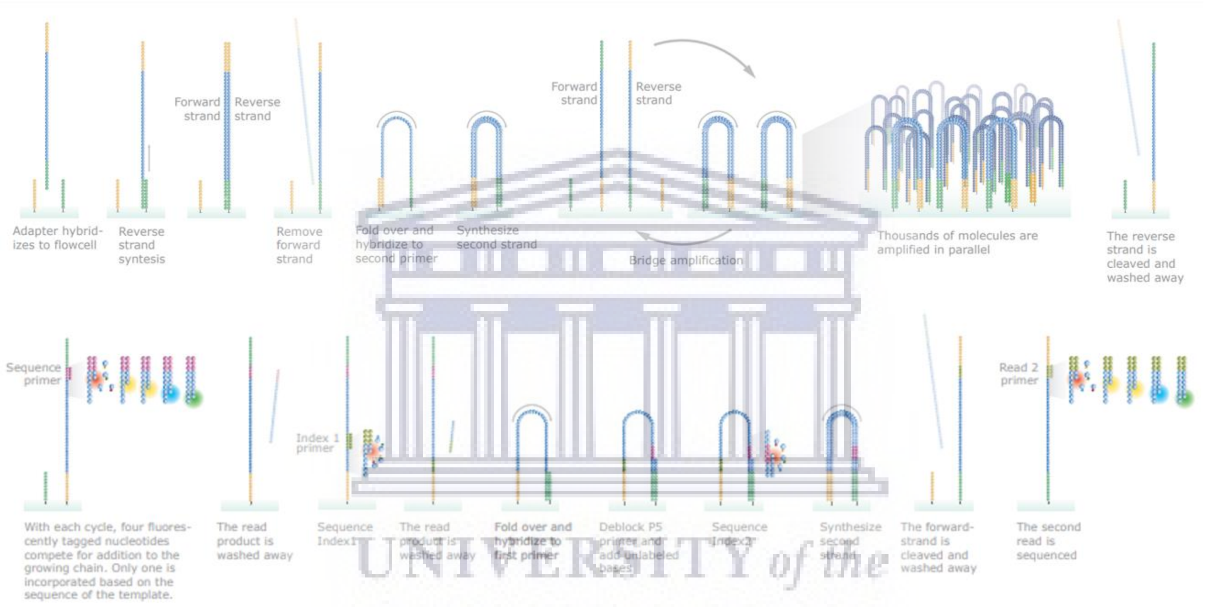


Figure 8: Illumina flow cell. A) *The Illumina HiSeq flow cell is a glass slide with eight separate lanes, through which reagents and template DNA flow.* ***B)*** *Cross-section view of a flow cell and single lane indicating direction of flow (Bronner et al., 2009).*

The Illumina cluster formation and sequencing process is described in Figure 9. For cluster generation, the single-stranded library fragments must first be hybridized to the grafting adapters on the flow cell (“Indexed Sequencing Overview Guide,” 2020; Launen, 2017). At this stage, the distancing of fragments is essential, and libraries are diluted before hybridization. The ssDNA fragments attach to their complementary oligomers via the P5 or P7 primer sequences of the sequencing adapters (Launen, 2017; “Multiplexed Sequencing with the Illumina Genome Analyzer System,” 2008). Then, polymerases attach to the grafting adapters and extend the 3’ end, creating copies of the original fragments. The resulting double-stranded DNA fragments are denatured, and the initial fragment is washed away, leaving the newly synthesized template covalently bound to the grafting adapter on the flow cell surface. The template then folds over, and the other end hybridizes to an adjacent complementary grafting adapter. Polymerases then extend the 3’ end of the hybridized grafting adapter, forming a double-stranded ‘bridge’ (Launen, 2017). The double-stranded bridge is denatured, resulting in two ssDNA molecules attached to the flow cell. These two molecules represent both strands of the original DNA fragment. Bridge amplification (bending of the ssDNA molecule, hybridization of the free end to adjacent complementary grafting adapter, dsDNA synthesis) is repeated until a cluster of ssDNA molecules that represent both strands of the original DNA fragment is formed. After the last round of bridge amplification and denaturation of dsDNA molecules, the strands attached to the P5 grafting adapters are cleaved and washed off the flow cell (Launen, 2017).



Forward strand
Reverse strand
Forward strand
Reverse strand
Adapter hybrid-izes to flowcell
Reverse strand syntesis
Remove forward strand
Fold over and hybridize to second primer
Synthesize second strand
Bridge amplification
Thousands of molecules are amplified in parallel
The reverse strand is cleaved and washed away
Sequence primer
Index 1 primer
Read 2 primer
With each cycle, four fluores-cently tagged nucleotides compete for addition to the growing chain. Only one is incorporated based on the sequence of the template.
The read product is washed away
Sequence Index1
The read product is washed away
Fold over and hybridize to first primer
Deblock P5 primer and add unlabeled bases
Sequence Index2
Synthesize second strand
The forward-strand is cleaved and washed away
The second read is sequenced

Figure 9: Sequencing by Synthesis *(Illumina Scientific Affairs, 2016).*

Prior to sequencing, the loose 3’ ends of the attached fragments are blocked to inhibit unwanted priming. Then, the first sequencing primer (i5, which matches the P5 primer sequences at the 3’ ends of the DNA molecules), as well as DNA polymerase and the four dNTPs (labelled with different fluorescent dyes that terminate polymerization) are introduced into the flow cell (“Illumina Sequencing Technology,” 2010; “Indexed Sequencing Overview Guide,” 2020). *De novo* DNA synthesis is directed towards the plate, regenerating the forward strand of the DNA fragment. After incorporation of the first dNTP the reaction is terminated by the dye. Nonincorporated dNTPs are washed away, the clusters are excited by a light source, and the colours for each cluster (representing the incorporated dNTP) are determined by an optical detector. Base calls for each cluster are made depending on the intensity of the signal from the excited fluorescent dyes. Thereafter, the dyes are enzymatically removed, permitting the next round of DNA synthesis (“Illumina Sequencing Technology,” 2010). Terminating the reaction after each single nucleotide incorporation, a method specific to Illumina, significantly improved precision when sequencing stretches of homopolymers. After a set number of sequencing rounds (e.g. 100) the dsDNA is denatured, and the unbound ssDNA molecules are washed off. The library-specific index is added to the recorded sequence as follows: the sequencing primer i7 is allowed to anneal to the 5’ end of the DNA molecules (in the P7 adapter region), and sequencing is resumed for up to 20 cycles, until the index and the remaining parts of the P7 adapter are covered (“Indexed Sequencing Overview Guide,” 2020). As a result, the recorded sequence (forward read) will have both, index and adapter sequences at the 3’ end only (Figure 10).

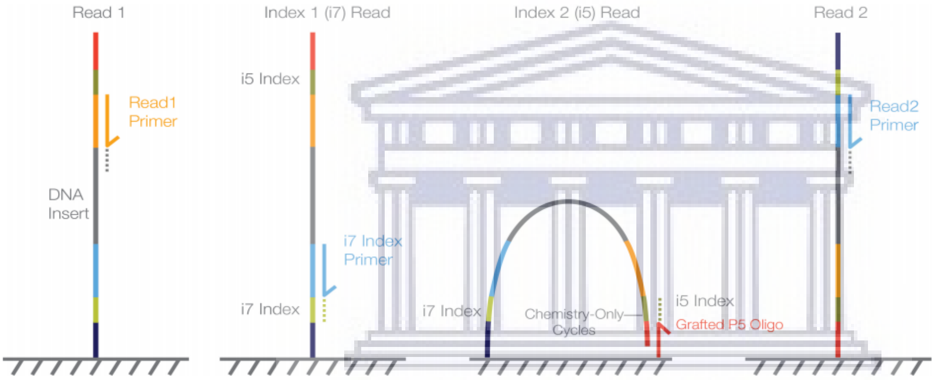


Figure 10: Dual-indexing of the forward strand on a paired-end flow cell *(“Indexed Sequencing Overview Guide,” 2020).*

After sequencing the forward strand, the template molecules fold over, and their 3’ ends hybridize to adjacent P5 grafting adapters. First, the index is sequenced using the i5 primer. Thereafter, the i5 primer and the product are removed, the complete strand is resynthesized (bridge amplification), the bridge is cleaved at the P7 adapter, the dsDNA is denatured, and the free ssDNA molecules are washed off (“Indexed Sequencing Overview Guide,” 2020). As a result, a ssDNA fragment representing the complementary molecule to the one just sequenced, is covalently bound to the P5 grafting adapters on the flow cell. The reverse strand can now be sequenced using the i7 primer, and the resulting reverse read will have the index sequence at the 3’ end (Figure 11).

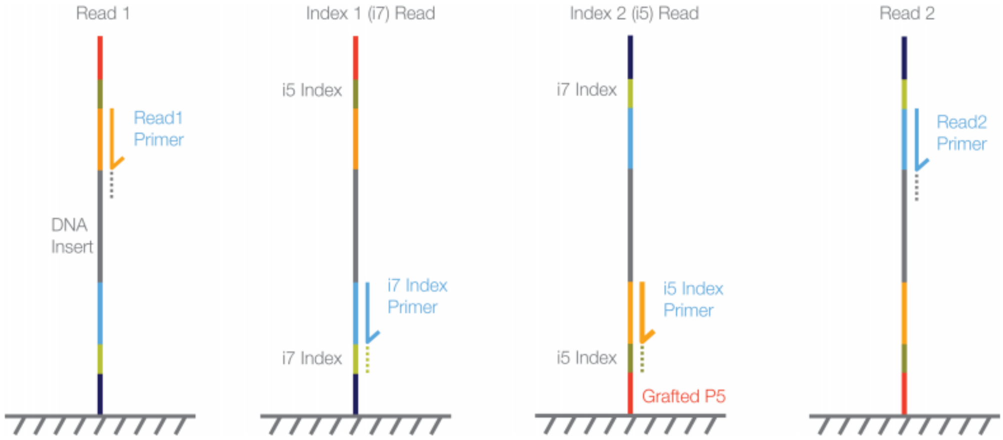


Figure 11: Dual-indexing of the forward strand on a paired-end flow cell *("Indexed Sequencing Overview Guide," 2020)***.**

1.4.3. Illumina sequencing data

This section will focus on Illumina sequencing platform-specific output data, describing data structure and format, including Illumina quality scores.

1.4.3.1. Flow cell tiles

To facilitate downstream data analyses (specifically, troubleshooting of sequencing runs), flow cell datasets are divided into smaller subsets based on flow cell tiles. A tile is defined as a small imaging area on the flow cell, which can be analyzed separately (Andrews, 2016). Each tile contains multiple clusters of reads (generated during clonal bridge amplification). All read names include the tile number as well as the x:y coordinates within the tile ("File Format," n.d.). For read pairs and mate pairs, these numbers are identical. MiSeq standard flow cells have 14 tiles ("MiSeq System Guide," 2018). For HiSeq datasets, each lane of the flow cell is divided into two columns, each containing 50 (GAII) or 60 (GAIIX) tiles ("NGS data formats and analyses," 2016).

1.4.3.2. Data format

Data derived from the sequence provider is converted from blc format to FASTQ format by the software program BlcToFastq (Illumina, USA). The sequence provider decides and programs before the run whether paired reads need to be in separate or in the same output file. FASTQ files include four lines. The first line contains a sequence identifier with information about the sequencing run, typically including the instrument name, run ID, flow cell ID, flow cell lane, tile number within flow cell lane, x:y tile coordinates, forward or reverse direction, information on read quality, control specification on or off (0 means off, and an even number means on), and the index sequence (“File Format,” n.d.). The reads of a pair (read pair or mate pair) will have the identical tile numbers and x:y tile coordinates. The second line contains the actual nucleotide sequence. The third line is a plus sign (+), which is used to separate the nucleotide sequences from line four that includes the quality values for each base in line two (“File Format,” n.d.). Consequently, lines two and four should contain the same number of symbols.

1.4.3.3. Illumina quality scores

One of the most common metrics used to assess sequencing data quality is base calling accuracy. Base-calling accuracy is measured by the Phred quality score (Q-score), which indicates the probability that a given base is called incorrectly by the sequencer (“Quality Scores for Next-Generation Sequencing,” 2011). Q-scores are defined as a property that is logarithmically related to the base calling error probability (P). The lower the error probability, the higher the quality score, and the less likely base calling was incorrect. As an example, a Phred score of 10 would imply a 90% accurate base call. A Q-score of 20 is often used as a threshold, since the probability of an incorrect base call is only 1% (“Quality Scores for Next-Generation Sequencing,” 2011). During an Illumina sequencing run, a quality score is assigned at every sequencing cycle to each base for every cluster. A quality lookup table is used to calculate quality scores. This quality lookup table uses a set of quality predictor values and relates them to corresponding quality scores (“Quality Scores for Next-Generation Sequencing,” 2011). Quality predictor values are observable properties of the light signal captured for each cluster (e.g. intensity profile and signal-to-noise ratio), which have been empirically determined to correlate with the quality of the base call. To estimate a quality score, the computed quality predictor values for a base call are compared to values in the pre-calibrated quality table (“Quality Scores for Next-Generation Sequencing,” 2011). Quality scores are recorded in base call files (*.bcl) that contain the base call and quality score for each cycle. When generating FASTQ files (*.fastq), the quality scores are converted to an encoded compact form which uses only 1 byte per quality value (“Understanding Illumina Quality Scores,” 2014). In this encoding, the quality score is represented as the ASCII code character equal to its value and adding 33. It is essential that the Q-tables are updated when characteristics of the sequencing platform, such as new hardware, chemistry, or software versions, change

("Understanding Illumina Quality Scores," 2014). This is necessary to accurately score data generated from the sequencer.

1.4.3.4. Biocomputational quality analysis of Illumina data

Quality analysis of Illumina sequencing data usually involves several rounds of quality assessment and quality processing steps using dedicated tools.

Quality assessment of Illumina sequencing data includes analysis of the overall GC content of the reads, as well as evaluation of the base quality scores, adapter contamination, read lengths, the proportion of duplicate reads, and quality loss due to position on the flow cell. Various software tools are available that provide summary statistics on the quality of NGS data, e.g. the NGS QC Toolkit (Patel and Jain, 2012) and pycoQC (Leger and Leonardi, 2019). In this study, I used FastQC (Andrews, 2010) and K-mer Analysis Toolkit (KAT;(Mapleson et al., 2016)), two widely applied, user-friendly, open-source software available online.

FastQC supports all NGS technologies, is continuously maintained and regularly updated by bioinformatics experts. It can analyze multiple input files at once, supports BAM, SAM and FASTQ input formats, and can be run from both the command line and through an interactive graphical user interface (GUI). FastQC completes a series of analysis modules which apply statistical tests to analyze the data. It generates an interactive HTML file which can be opened in a web browser. The left-hand side panel gives a quick overview of whether the results from each module seem normal (green tick), slightly abnormal (orange triangle) or very unusual (red cross). FastQC considers a sample as 'normal' when the statistical tests indicate randomness and diversity of the dataset. Biased results can be associated with genome characteristics, but also due to diverse laboratorial and computational steps introduced during sequencing (Ross et al., 2013). Coverage bias is a deviation from the uniform distribution of reads across the genome. For example, GC-rich and GC-poor regions are prone to be underrepresented in the Illumina sequencing data due to low coverage, which can be introduced during PCR amplification steps during library construction and cluster amplification. Error bias is a deviation from the expected uniform insertion, deletion, and mismatch rates in reads across the genome (Ross et al., 2013). FastQC facilitates assessment of these biases, permitting better understanding on the quality of the sequencing data. Because FastQC gives a single report per file, it is harder to analyze data from multiple input files. In this case, MultiQC (Ewels et al., 2016) is a useful tool. MultiQC scans directories for results from other bioinformatics tools to compile summary statistics into a single report. It is compatible with 89 tools and gives results in the form of graphs and tables. A list of compatible tools is available at https://multiqc.info/#supported-tools.

Another approach to assessing sequencing quality is k-mer analysis. A k-mer is a DNA sequence with a fixed-length – a 'word' of length k, made up of nucleotides. Figure 12 illustrates how a 17 bp sequence can be broken up into 11 k-mers of length 7 bp (7mers), where each k-mer differs from the previous one by one nucleotide. Breaking a sequence into k-mers reduces the complexity of the dataset: while the total number of sequences is larger, the number of unique sequences is much reduced. Saving only the unique k-mer sequences and the corresponding number of occurrences permits a more efficient approach to analyzing sequencing data (in terms of data volume and data structure). Consequently, k-mer counting is applied in many bioinformatics tools to analyze the sequencing data (evaluating sequencing biases, completeness of sequencing coverage, and contaminations in the sequencing datasets), but also to perform error correction and to assemble the sequencing data (Wright, Jon, 2016).

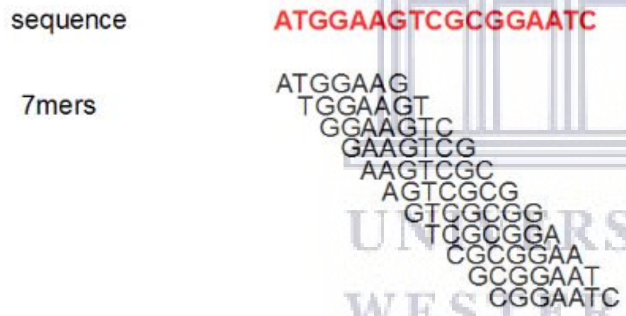


Figure 12: An illustration of how a 17 bp DNA sequence can be broken into 7 bp k-mers (7mers) *("An Intuitive Explanation for Running Velvet with Varying K-mer Sizes," 2012).*

The K-mer Analysis Toolkit (KAT) is a k-mer counting program that permits quality assessment of next-generation sequencing data, prediction of genome characteristics (including genome size) and evaluation of the quality of genome assemblies. A list of tools available from KAT, as well as their applications, is available in the KAT documentation (Wright, Jon, 2016). The following tools from KAT can be used on raw whole genome sequencing (WGS) data: hist, GCP, comp, and the two filtering tools. The hist tool takes input in the form of one or more FASTQ or FASTA files. The output is a histogram file with the number of distinct k-mers and their frequencies as well as a spectra histogram plot. The tool is used to assess data quality parameters (including sequencing bias and error levels) and to identify genomic properties of the sequenced genome. The GCP tool uses the same input as hist, and its application is similar, except that it is better at detecting contaminations. Sequence contamination is often associated with unusual GC and coverage levels. The GCP tool, therefore, counts the GC nucleotides for each distinct k-mer and creates a matrix relating the number of distinct k-mers per GC count to the k-mer coverage. The results are visualized in a density plot. The comp tool compares the k-mers from two or three datasets, which has a wide range of applications, including benchmarking of sequencing runs on a given machine, comparing error forward and reverse reads and in different libraries (PE vs PE or PE vs MP). Based on the matrix of k-mer spectra frequencies in each dataset, a number of different plots can be created. The density plot is a visual representation of the shared errors in the datasets. The spectra-mx plot visualizes shared and exclusive content from the two datasets. The spectra-cn plot compares k-mers from input reads and assemblies, effectively revealing missing content (unassembled data) and multi-copy sequences in the assemblies. The two filtering tools from KAT (kmer and seq) allow to filter k-mers by coverage or GC count and to remove reads based on these filtered k-mers. KAT also provides various additional plotting tools that can be used on the outputs generated by the above tools to visualize the results.

Essential quality processing of Illumina sequencing data includes filtering the datasets for PhiX sequences and trimming of the reads to remove library-specific sequences (e.g. adapters, spacers). Raw Illumina reads from paired-end libraries typically have adapter sequences at the 3’ end (Figure 13-A). Mate pair reads can include the sequencing adapters as well as the junction adapters (Figure 13-B). Additional quality processing steps may include trimming of low-quality bases and the removal of low-quality, short and/or duplicated reads. Again, a number of tools are available that serve this purpose, but here I will focus on the three quality processing tools that were used in this study: Trimmomatic (Bolger et al., 2014), NextClip (Leggett et al., 2014), and FLASH (Magoc and Salzberg, 2011). Trimmomatic can process Illumina mate pair as well as single- and paired-end read datasets to remove adapters, trim bases from the ends of each read (either a set number or based on the respective quality scores), remove reads below a particular quality score and filter reads below a certain length. A detailed description of the program is available at http://www.usadellab.org/cms/?page=trimmomatic. NextClip is specifically designed to process Nextera mate pair libraries. The tool comprises two parts: The NextClip tool and the NextClip pipeline. As shown in Figure 14, sequencing of a mate pair results in two reads (R1 and R2) in mate-pair direction that should contain the junction adapter sequence at the 3’ end. However, the laboratorial step that focuses on the selection of biotinylated molecules is imperfect, and nonbiotinylated DNA molecules that do not contain the ends of the original DNA fragment can also be sequenced. These sequences represent a dangerous contamination, since they severely distort scaffolding (the reads of a mate pair are expected to be 2-10kb apart, depending on the intended fragment size of the library; while the distance between these two reads would be much smaller). Therefore, any mate pair which is missing the adapter sequence in at least one of the two reads or where the adapter sequence orientation is incorrect should be filtered out. Accordingly, the NextClip tool investigates a pair of corresponding FASTQ files (containing the R1 and R2 reads of a Nextera mate pair library, respectively) for presence and relative orientation of the junction adapter sequences. It then assigns each mate pair to one of four categories: category A – the adapter is present in both reads; categories B and C – the adapter is present only in R2 or in R1, respectively; and category D – the adapter is missing in both reads. Subsequently, only the reads from category A are processed: junction adapters are removed and reads shorter than a user-configurable minimum length are discarded. Invoking the NextClip pipeline permits mapping of the mate pair reads to preassembled genome contigs of the dataset, which allows estimation of the insert size, finalizing the selection process of suitable mate pair sequences. Application of the NextClip ensures that only true mate pair reads are used in downstream analysis.

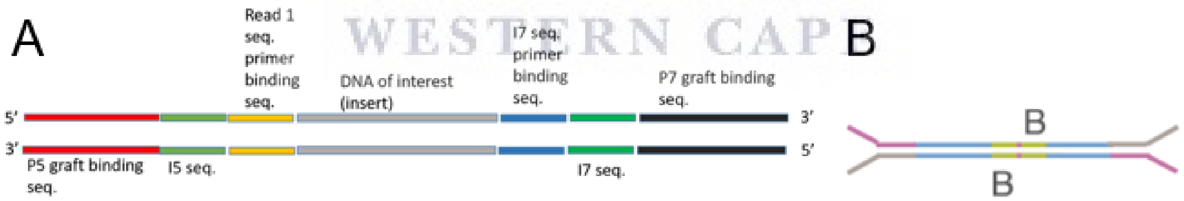


*Figure 13: **Structure of paired-end and mate pair reads. A)** paired-end reads contain adapter sequences at the ends of the reads. **B)** Mate pair sequences contain adapter sequences at the ends of the reads, as well as the middle of the reads (junction adapters) (Launen, 2017; "Nextera® Mate Pair Library Preparation Kit: Data sheet," 2014).*

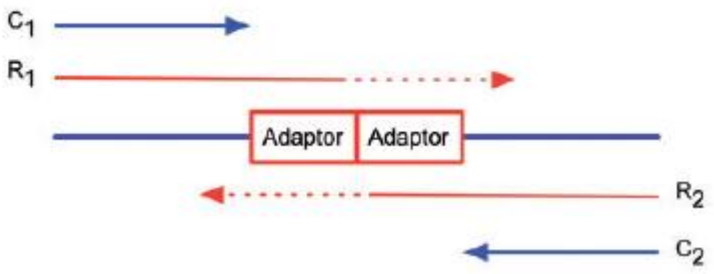


***Figure 14: Two junction adapters join Nextera mate pair fragments**. Sequencing of the fragment generates R1 and R2. Use of NextClip to find and remove junction adapters generates C1 and C2 reads (Leggett et al., 2014).*

FLASH is a tool that is used to create longer sequences out of short, overlapping, paired-end reads from the small-insert Illumina sequencing libraries (Figure 15). FLASH requires FASTQ paired-end reads as input. It processes each read pair separately and searches for an overlap between the read pairs. When a correct overlap is found, the read pairs are merged into a single, longer, read which matches the DNA fragment from which the reads were sequenced (Magoc and Salzberg, 2011). Other NGS processing tools include AfterQC (Chen et al., 2017) and FastP (Chen et al., 2018).

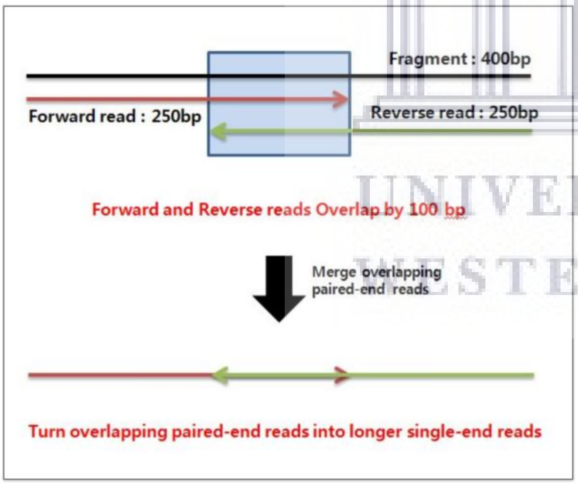

Fragment : 400bp
Forward read : 250bp
Reverse read : 250bp
Forward and Reverse reads Overlap by 100 bp
Merge overlapping paired-end reads
Turn overlapping paired-end reads into longer single-end reads

***Figure 15: Overlapping paired-end reads merged to form a longer, single read** (Lee, 2015).*

Another quality processing step often included in Illumina sequence data analysis for *de novo* genome assemblies is error correction. Error correction aims to decrease sequencing errors, thus improving downstream analysis while maintaining the data heterogeneity (Mitchell et al., 2020). Error correction tools can be categorized into one of four classes: k-mer spectrum-based, suffix tree/array-based, multiple sequence alignment (MSA)-based, and hidden Markov model (HMM)-based (Akogwu et al., 2016). K-mer spectrum-based methods are the fastest growing class of error correction tools. The idea behind k-mer spectrum-based methods is that erroneous bases in a DNA sequence occur infrequently and independently and can be corrected using the majority of reads that have the correct bases. Examples of k-mer spectrum-based error correction tools include Lighter (Song et al., 2014), BLESS (Heo et al., 2014) and Quake (Kelley et al., 2010).

1.5. Genome characteristics

Genome characteristics such as genome size, ploidy levels, rate of heterozygosity and repeat content are essential factors that need to be taken into consideration when planning a whole genome sequencing project. These factors will not only determine the type and amount of sequencing data required to reassemble a genome of desired quality, but also the approach and choice of programs for data preprocessing and assembly, as well as computational time and resource requirements (Liu et al., 2013). According to Dominguez Del Angel (2018), these characteristics may have a higher impact on the assembly results than the assembly software used. In this section, I will discuss each characteristic and its effects on genome assembly, and introduce k-mer analysis as a computational approach to estimate these genome characteristics.

Genome size refers to the total amount of DNA present in one copy of a single (haploid) genome (Fridovich-Keil, 2019). It can be measured as a mass in picograms (pg) or in nucleotide base pairs (bp) (DeSalle et al., 2005). The genome size directly affects the amount of data that needs to be sequenced. The larger the genome, the more data is required, and the more computationally intensive it will be. Plant genomes are often on the larger side due to the presence of duplications and repeats (Michael, 2014).

Zygosity refers to the degree to which copies of a chromosome (or locations on them) differ in their genomic sequence (“Zygosity,” 2020). Corresponding genome locations can be homozygous (have the same nucleotide sequence) or heterozygous (differ in their nucleotide sequence). Plant genomes generally have higher rates of heterozygosity than organisms from other kingdoms (Schatz et al., 2012). Genome assembly programs typically construct the final assembly as a haploid (single copy) genome, choosing the nucleotide in polymorphic positions arbitrarily (Chin et al., 2016). Reads from heterozygous regions are handled in one of two ways: either these allelic differences are collapsed into one consensus sequence, possibly leading to a more fragmented assembly; or these heterozygous regions are assembled separately, leading to duplicated content in the final assembly. High levels of heterozygosity can, therefore, increase not only fragmentation but also loss of content in the final genome assembly, which in turn can lead to false conclusions on the biology of the studied organism (Schatz et al., 2012).

Ploidy refers to the number of chromosome copies in the nucleus of a cell. Haploid and diploid imply one and two sets of chromosomes, respectively. Polyploidy refers to the state of an organism where more than two sets of chromosomes are present in each cell. It is estimated that up to 80% of all plant species are polyploid (Schatz et al., 2012). Genome reassembly of heterozygous polyploid organisms is particularly challenging since gene loci with nucleotide polymorphisms are present as multiple versions (depending on the ploidy level). Most genome assembly programs ignore ploidy when assembling sequences into contigs. Diploid-aware assembly programs have been created only recently. These programs use phasing and haplotyping to construct each haplotype separately. Examples of such pipelines include FALCON (Chin et al., 2016), Ranbow (Yang et al., 2017), and SDhaP (Das and Vikalo, 2015).

Repetitive sequences, or repeats, are regions in a genome which are present multiple times, whether in one or (many) different locations. Plant genomes are known to have a high proportion of repeats (Schatz et al., 2012). Repeats can be mono or polynucleotide in their sequence, spanning between 1 bp (e.g. microsatellites) and thousands of base pairs (e.g. Long Interspersed Nuclear Elements (LINEs)) (De Roeck et al., 2019). Their presence severely complicates genome assembly. Since reads from these regions are very similar or even identical, it is hard for the assembly program to determine their location in the genome. High proportions of repeats in a genome may, therefore, lead to misassembled and/or highly fragmented genome assemblies. Repeats are best resolved using reads that span them completely. For long repeats, third-generation sequencing technologies, although currently still expensive and of lower quality, represent a solution (M. E. Bolger et al., 2014; Dominguez Del Angel et al., 2018).

As discussed above, k-mer counting is used by many bioinformatics tools to analyze and error correct sequencing data. K-mer frequency analysis performed on second-generation sequencing reads can also be used to estimate genome characteristics. All that is needed are short-read reads that roughly amount to a 30x coverage of the investigated genome (Vurture et al., 2017a). Several k-mer counting programs are available, including Jellyfish (Marçais and Kingsford, 2011), DSK (Rizk et al., 2013), BBNorm (Bushnell, 2018) and KAT. In this study, KAT and BBNorm were used to count k-mers and to produce k-mer histogram files for estimation of genome characteristics. Figure 16 shows a k-mer spectrum generated by KAT.

The first peak corresponds to low-frequency k-mers derived from erroneous sequences. The second peak (here around ~100x genome coverage) corresponds to heterozygous genome content. The homozygous peak is located at double the coverage of the heterozygous peak (~200x). The tail of the graph (from ~350x coverage and onwards) mostly represents k-mers derived from repetitive regions of the genome.

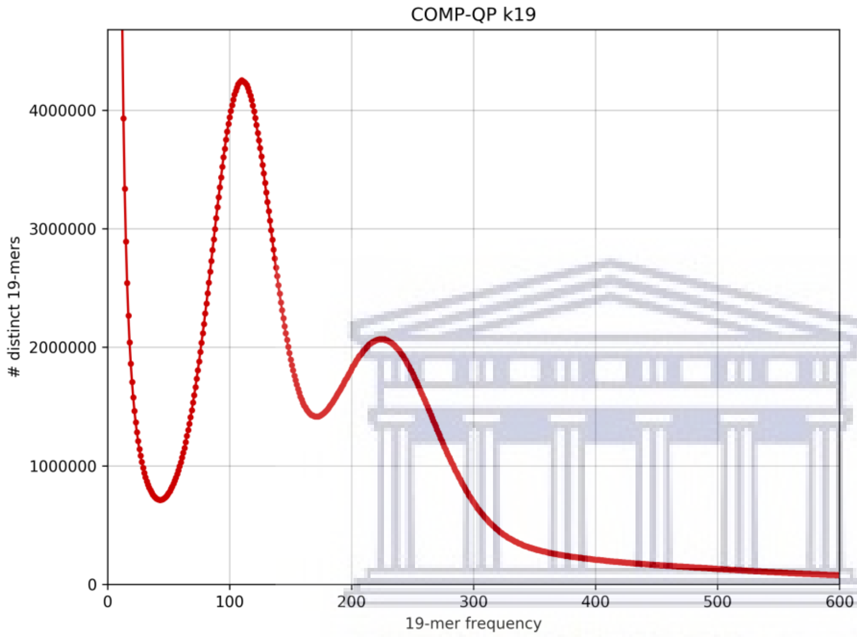


Figure 16: K-mer spectra graph generated by KAT at k=19 on quality-processed data.

1.6. Genome assembly

Genome assembly refers to the computational process of using nucleotide sequences (reads obtained through NGS sequencing) to reconstruct the genome of the organism under study. The ultimate aim is to correctly arrange the sequenced reads and determine the most likely consensus sequence – the sequence consisting of the most frequently observed nucleotide at each given position. This process is often compared to building a gigantic jigsaw puzzle made up of millions, or even billions of pieces. Genome assembly is divided into two categories: *de novo* (assembly without a reference genome) or reference-based (assembly using a reference genome). Reference genomes are currently lacking for most non-model organisms, including rooibos.

Current assembly programs (assemblers) use one of two assembly algorithms, either the overlap-layout-consensus (OLC) method or the de Bruijn graph (DBG) method (reviewed in Rizzi et al., 2019). The OLC method includes three main steps: 1) computing the overlaps between the reads, 2) building an overlap graph (Figure 17A), where the reads represent the nodes and the overlap relationships between the reads represent the arcs; and 3) inferring the most likely consensus sequence for each contiguous sequence stretch (contig). This algorithm is used in the Celera Assembler (Miller et al., 2008) and in the Maryland Super-Read Celera Assembler (MaSuRCA; Zimin et al., 2013)). Assemblers that use the DBG method do not use the complete reads. Instead, each read is converted into a set of k-mers. In the de Bruijn graph (Figure 17B), the nodes are the k-mers, and the arches are the last characters of the following node. Although this step appears to increase the dataset at first glance, the complexity of the dataset is actually lower: shorter sequences are less variable than longer sequences, and therefore the number of sequences that have to be stored is reduced. Consequently, DBG algorithms are computationally less demanding and less time consuming than the OLC algorithm. A consequential drawback, however, remains – reassembly of repeat regions requires additional analysis steps. A vast number of genome assemblers use the de Bruijn graph algorithm, and some have been successfully employed in plant genome reassembly. These include Platanus, ABySS, and ALLPTHS-LG (Basantani et al., 2017). However, to date not a single assembler dedicated to address all plant-specific genome assembly issues (such as large genome sizes, polyploidy and high levels of heterozygosity) has been developed.

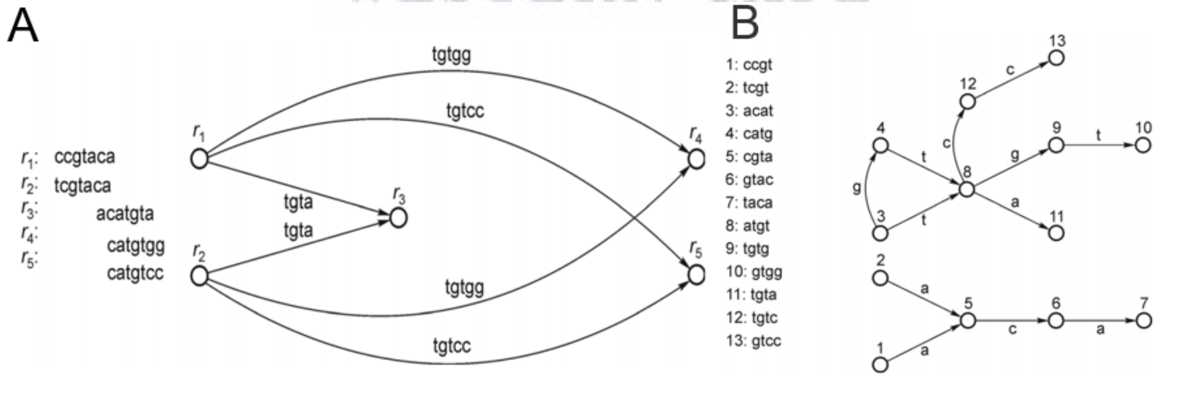


Figure 17: Example of an A) *overlap graph for five reads (r_1-r_5), and a* **B)** *de Bruijn graph of two reads* ccgtac *and* catgtg *for k=3. Each one of the 13 k-mers represents a node (Rizzi et al., 2019).*

Considering the complexity of genome reassembly, an obvious question arises: how does one know if an assembly is of good quality? Which program and which parameter settings work best? This can be investigated by comparing assembler performance using the datasets at hand.

Essential evaluation criteria are completeness, correctness and contiguity of the assembly (Bradnam et al., 2013). Several ballpark numbers inform on the contiguity of a genome assembly: N50, NG50, and L50. N50 is the sequence length of the shortest contig at 50% of the total assembly length, i.e. 50% of the entire assembly is contained in contigs or scaffolds equal to or larger than this value (Bradnam et al., 2013). NG50 is similar to the N50 statistic, but considers the actual estimated genome size, i.e. 50% of the entire genome is contained in contigs or scaffolds equal to or larger than this value (Bradnam et al., 2013). The L50 statistics is the smallest number of contigs whose length sum makes up half of the assembly size ("N50, L50, and related statistics," 2020). These can be calculated manually using assembly statistics provided by the assemblers. However, by now, a number of tools dedicated to genome assembly evaluation have been developed.

In this study, QUAST-LG (Mikheenko et al., 2018) and KAT were employed. QUAST-LG (QUality Assessment Tool), published in 2018, is an upgraded version of QUAST, specifically created to evaluate large *de novo* genome assemblies (Mikheenko et al., 2018). QUAST-LG can assess the completeness and correctness of the assembled genome by calculating the portion of the genome that is assembled and estimating the amount of errors the assembly contains, respectively (Mikheenko et al., 2018). Still, without a reference genome, it is hard to determine how accurate these estimates are. A relatively recent approach to genome assembly assessment includes the prediction of genes and subsequent analysis of that dataset for core genes (genes that are universally present between species). QUAST-LG not only permits gene prediction using the programs GeneMark-ES (Ter-Hovhannisyan et al., 2008) and GlimmerHMM (Majoros et al., 2004), but also employs BUSCO (Benchmarking Universal Single-Copy Orthologs) (Simão et al., 2015) to assess core gene detection statistics for the assembled genome. BUSCO utilizes the OrthoDB database (www.orthodb.org) to measure genome completeness according to an expected gene content (single-copy orthologs). Genes which are present in at least 90% of the species in the OrthoDB database are included (Simão et al., 2015).

The program KAT employs a very different approach to genome assembly evaluation. A spectra-cn graph (where k-mers from the reads are compared to the k-mers from the assembled genome contigs) can be generated using the comp tool. This graph permits analyses of the content of an assembly, including errors, homozygous content, heterozygous content and repeats. K-mer occurrence in the assembly dataset are shown as different colours (black implies that these k-mers are missing in the assembly dataset, red indicates one appearance of the k-mer, orange indicates two appearances (duplicated assembly content), light-green corresponds to three appearances (triplicated assembly content), etc.) (Wright, Jon, 2016). Figure 18A shows what a perfect spectra-cn plot would look like for a diploid heterozygous genome. The first peak is entirely black, indicating that no error k-mers are present in the assembly. The heterozygous content (second peak) is present exactly once (red), with some content missing (black). This is logical: the heterozygous peak of a diploid organism includes both copies of a genome location, one of which is present in the assembly. The homozygous content (third peak) is present exactly once (red) in the assembly. In practice, it is more common to see assemblies with some degree of errors included in the assembly, missing homozygous content, and content that is present more than once in the assembly (indicated as colours other than red or black; Figure 18B) (Wright, Jon, 2016).

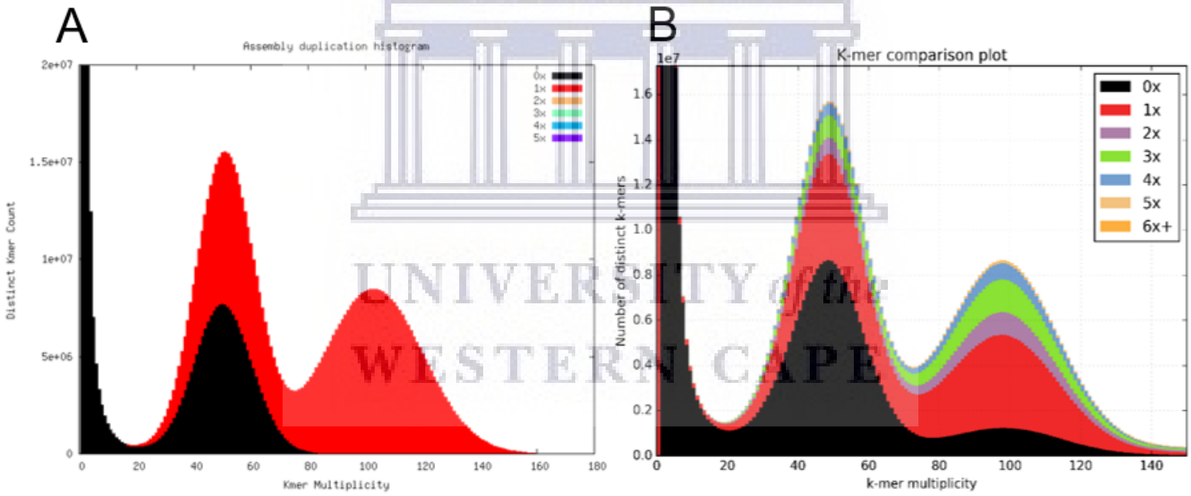


Figure 18: Spectra-cn plots generated by KAT. A) *An example of a "perfect" spectra-cn plot generated for a diploid, heterozygous genome.* ***B)*** *An example of a spectra-cn plot which is most often observed for a diploid, heterozygous organism (Wright, Jon, 2016).*

Third-generation sequencing technologies are currently revolutionizing the field of genome sequencing, opening doors to analysing even the largest and most complex genomes on earth. Biocomputational procedures must be adapted. The revolution of this research field can be expected in the near future. However, this topic is beyond the scope of this study.

Chapter 2: Materials and Methods

2.1. Illumina sequencing data

This study focuses on the analysis of Illumina sequencing data from rooibos DNA that had been generated in a previous study. Leaf material was obtained from a commercial rooibos plant in Nieuwoudtville (S031˚23'0" E019˚06'0"), Northern Cape province, South Africa. It was flash frozen in the field using liquid nitrogen, transported on dry ice to the laboratory and maintained at -80˚C. DNA extraction was performed as described previously (Mgwatyu, 2019). Thereafter, the DNA sample (~ 5 µg) was packaged on ice and sent by DHL to the sequencing service provider.

DNA library preparation and sequencing was performed at UKHC Genomics Core Laboratory (UK Chandler Hospital, Lexington, KY, 40536, USA) using reagents and equipment from Illumina (Illumina, San Diego, CA, USA) unless stated otherwise. Three Illumina libraries with average insert sizes of ~300 bp, ~3 kbp and ~8 kbp were prepared. The 300bp paired-end library was constructed using the Nextera DNA Library Preparation kit following the manufacturer's instructions. First, this library was sequenced on an Illumina MiSeq platform using two single-lane MiSeq flow cells and the MiSeq reagent kit (2 x 125 bp). Thereafter, the library was sequenced on a HiSeq 2500 platform in high output mode using six lanes of one HiSeq flow cell. The remaining two lanes were used to sequence the rooibos mate pair libraries i.e. only the rooibos sample was investigated on this flow cell. HiSeq sequencing was conducted using the HiSeq PE Cluster Kit v4 cBot and the HiSeq SBS Kit v4 (125 cycles). For the two mate-pair libraries, the Nextera Mate Pair Library Prep Kit was used for library construction following the gel-plus protocol (using 4 µg of input DNA, CloneWell agarose gels (Invitrogen, Thermo Fisher Scientific) for fragment size selection and Dynabeads M-280 (Invitrogen, Thermo Fisher Scientific) for purification of sheared DNA. The mate pair libraries were sequenced on an Illumina HiSeq 2500 platform in rapid run mode using one flow cell (one lane per library) and the sequencing reagents from the HiSeq PE Rapid Cluster Kit v2, HiSeq Rapid SBS Kit v2 (125 cycles) and HiSeq Rapid Duo cBot Sample Loading Kit. In all sequencing runs, Illumina PhiX v3 was used as a spike-in.

The service provider also conducted data format conversion and basic data processing. The Illumina bcl2FASTQ2 Conversion Software v2.20 (Illumina, San Diego, CA, USA) was used for demultiplexing, conversion of base call (BCL) files into FASTQ files, trimming of adapter sequences, as well as removing PhiX reads and separating undetermined reads (reads which did not contain the expected index) from determined reads (reads which contained the expected index). The command line used for this was: sudo bcl2FASTQ -r 8 -d 8 -p 8 -w 8 -l TRACE --no-bgzf-compression -R <input_folder_path -o <input_folder_path">. In this thesis, the data obtained from the service provider will be referred to as “raw”.

2.2. Computational data analysis

The computational data analyses in this study included: 1) quality assessment and quality processing of the Illumina sequencing data, 2) analysis of rooibos genome characteristics using raw and quality processed Illumina sequencing data, and 3) assessment of assemblers for rooibos genome assembly. Figure 19 shows the primary outlay.

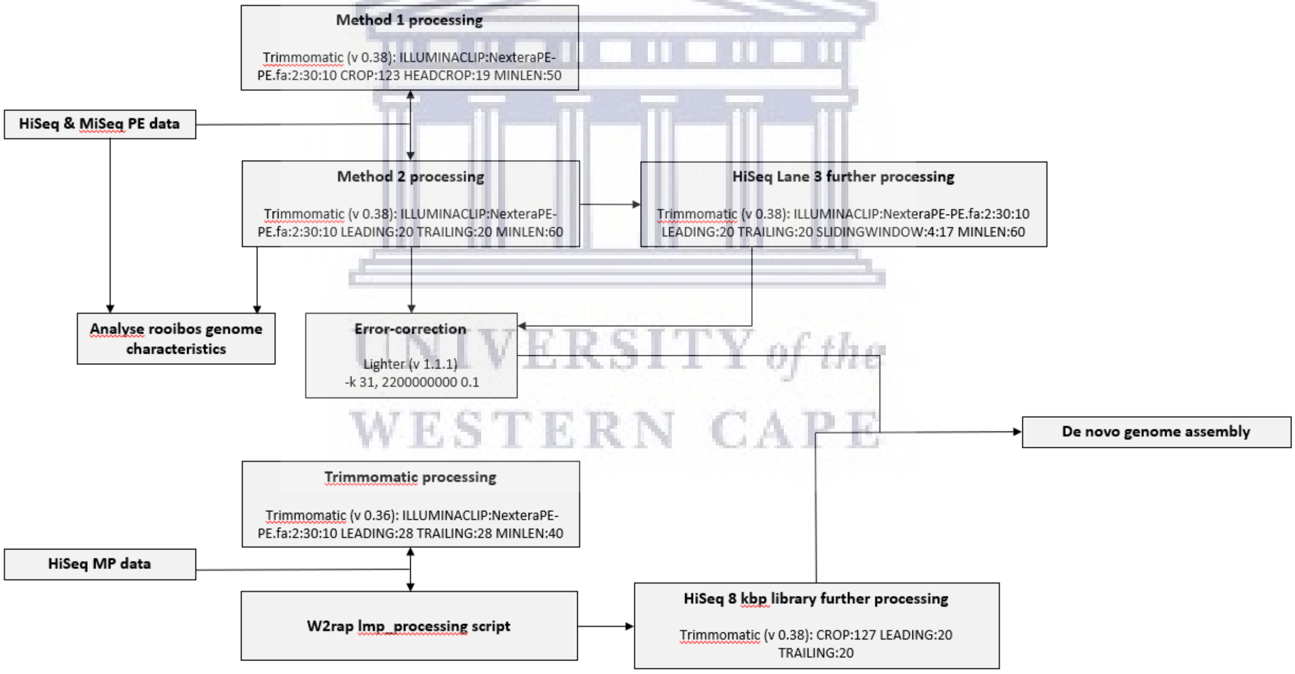

Method 1 processing
Trimmomatic (v 0.38): ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 CROP:123 HEADCROP:19 MINLEN:50
HiSeq & MiSeq PE data
Method 2 processing
Trimmomatic (v 0.38): ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:20 TRAILING:20 MINLEN:60
HiSeq Lane 3 further processing
Trimmomatic (v 0.38): ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:20 TRAILING:20 SLIDINGWINDOW:4:17 MINLEN:60
Analyse rooibos genome characteristics
Error-correction
Lighter (v 1.1.1)
-k 31, 2200000000 0.1
De novo genome assembly
Trimmomatic processing
Trimmomatic (v 0.36): ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:28 TRAILING:28 MINLEN:40
HiSeq MP data
W2rap lmp_processing script
HiSeq 8 kbp library further processing
Trimmomatic (v 0.38): CROP:127 LEADING:20 TRAILING:20

Figure 19: Schematic of the analysis pipeline.

2.2.1. Illumina data quality assessment and quality processing

Read quality was assessed using FastQC (v 0.11.5 and v 0.11.7; Andrews, 2010) and MultiQC (v 01.7, Ewels et al., 2016). FastQC was applied to determine read numbers, read lengths and %GC, and to visualize sequence quality for the forward and reverse reads in each data set.

MultiQC was used to calculate average read length, percent duplicates per lane, the percentage of reads with a mean quality score ≥ 30 (using the data from the per sequence quality scores plot generated by MultiQC), as well as to compile summary statistics from FastQC reports.

Two quality processing methods were tested on the paired-end data (300bp library) on a lane per lane basis. First, the effect of hard cropping (removal of a fixed number of bases from the sequence ends) was investigated (PE-Method 1). To achieve this, Trimmomatic (v 0.38; Bolger et al., 2014) was run with the following parameters: ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 CROP:123 HEADCROP:19 MINLEN:50. This served to remove remaining adapter sequences, trim the bases 124 and 125 bp in reads that had them, hard-crop the first 19 bases in all reads and, thereafter, discard reads that were less than 50bp long. This method did not include trimming based on sequence quality. In a second approach (PE-Method 2), paired-end reads from each lane were first trimmed based on sequence quality using Trimmomatic (v 0.38) and subsequently error corrected using Lighter (v1.1.1; Song et al., 2014). The following parameters in Trimmomatic were applied to each dataset: ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:20 TRAILING:20 MINLEN:60. These parameters served to remove adapter sequences, trim bases from the 5' and 3' ends of reads with a Phred score below 20, and to discard reads shorter than 60 bp. The HiSeq Lane 3 datasets (forward and reverse) were processed further to improve the per-base quality of the reads: Trimmomatic was run using ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:20 TRAILING:20 SLIDINGWINDOW:4:17 MINLEN:60. Thereafter, all datasets were error corrected using Lighter (v 1.1.1) with the following parameters: -k 31, 2200000000 0.1. Following the advice of the program author (personal communications), the genome size used in the Lighter analysis was twice the actual rooibos genome size estimate to account for high levels of heterozygosity predicted in this genome.

Mate-pair libraries were also quality processed using two different approaches. As a first method (MP-Method 1), Trimmomatic (v 0.36) was run using the following parameters: ILLUMINACLIP:NexteraPE-PE.fa:2:30:10 LEADING:28 TRAILING:28 MINLEN:40. This resulted in adapter removal, trimming of bases with quality scores below 28 at the 5' and 3' ends and removal of all reads shorter than 40 bp. As an alternative approach (MP-Method 2), the Python (v2.7) lmp_processing script from w2rap: the WGS (Wheat) Robust Assembly Pipeline (Clavijo et al., 2017) was investigated. This script uses FLASH and Nextclip to 1) merge overlapping reads, 2) perform deduplication of reads, 3) identify reads containing the

Nextera adapter (i.e. identifying true mate-pairs), and 4) discarding reads not containing the adapter. The script was run on both mate-pair libraries (MP3 and MP8) using default settings on a lane-by-lane basis. To address low quality read ends of the MP8 library, it was further processed using Trimmomatic (v 0.36) with the following parameters: CROP:127 LEADING:20 TRAILING:20.

Due to better quality parameter statistics, only the data obtained using PE-Method 2 and MP-Method 2 was used in subsequent analyses and is referred to as quality-processed (QP) data.

2.2.2. Investigation of genome characteristics

K-mer histogram analysis permits estimation of genome characteristics, such as genome size, level of heterozygosity and repeat content. For the rooibos data, different tools for generating k-mer histogram files and subsequent analysis of genome characteristics were investigated.

First, 16 k-mer histogram files were generated using the khist.sh script from BBNorm. Histogram files were generated for four datasets (MiSeq-raw, MiSeq-QP, COMP-raw, and COMP-QP) and four k-mer values (19, 23, 27, and 47) per dataset, using default parameters with the specific k-mer value and setting the histogram length (histlen) to 900000. COMP represents the combined MiSeq and HiSeq datasets of the 300 bp Illumina library and QP implies quality filtering. Mate-pair reads were excluded from this analysis, because these datasets contained a high proportion of duplicated reads. Furthermore, eight k-mer histogram files were generated using the hist tool from KAT. Histogram files were generated for the two COMP datasets, testing k-mers 19, 23, 27 and 47. This tool was run using default parameters, specifying the k-mer value (-m) and setting the hash size (-H) to 1000000. The large hash size allowed KAT to analyse the data without growing the hash while running, saving time and reducing memory usage. Max coverage (900,000x) was set using -h 900000.

The 24 k-mer histogram files were subsequently analysed using the programs GenomeScope v1 (Vurture et al., 2017b), GenomeScope v2 (Ranallo-Benavidez et al., 2019), FindGSE (Sun et al., 2018), BBNorm, and KAT to estimate rooibos genome characteristics.
The two GenomeScope versions have different adjustable parameters. With both versions, the effect of three maximum k-mer coverage thresholds (1k, 10k, and 900k) was investigated. For GenomeScope v1, read length was set to 119 nt for raw data, and 120 nt for the quality processed data (based on FastQC results). For GenomeScope v2, ploidy level was set to 2, and

the “average k-mer coverage for polyploid genome” was left in the default setting (-1). FindGSE was run specifying the respective k-mer value and exp_hom (the expected k-mer coverage for the homozygous region). The value for exp_hom was selected following authors instructions (https://github.com/schneebergerlab/findGSE/blob/master/R/findGSE_v1.94.R): it was the k-mer coverage value, two counts, before the k-mer coverage value at the maximum height of the homozygous peak (satisfying the requirement that fp < VALUE < 2*fp, where fp is the maximum frequency of the homozygous peak). The khist.sh script from BBNorm not only produces the k-mer histogram files but also calculates the genome size, as well as heterozygosity and repeat content. The callpeaks.sh from BBNorm was used to generate BBNorm estimates for the KAT histogram files, specifying the k-mer size and setting ploidy=2 (the rooibos genome is diploid). The hist tool from KAT produces k-mer histogram files and k-mer spectra graphs, and estimates the genome size and heterozygosity rates. Since it does not accept histogram files as an input, KAT was not used to analyse the BBNorm histogram files.

In addition, the histogram files were used to estimate the rooibos genome size using the following popular, simple formula derived from equations introduced by the M.S. Waterman group (Lander and Waterman, 1988; Li and Waterman, 2003): $G = \frac{N}{C}$, where G is genome size, N is the total number of k-mers, and C is the k-mer frequency at the homozygous peak. Low-frequency k-mers (corresponding to the first peak up to the lowest point of the first valley) likely represented sequencing errors and were excluded from the calculation of the total k-mer numbers.

2.2.3. De novo genome assembly

De novo genome assembly analysis was performed on the high-performance computing clusters at the South African National Bioinformatics Institute (SANBI-UWC: 32 cores, 500 GB RAM) and the Centre for High Performance Computing (CHPC: 56 cores, 990 GB RAM) in Cape Town, South Africa.

Initially, analyses were performed on data subsets, using either paired-end datasets from the two MiSeq lanes, or paired-end datasets from two HiSeq lanes plus the 3 kbp mate-pair library dataset (Table 7). Six genome assembly programs were tested for their performance, namely ALLPATHS-LG (Gnerre et al., 2011), MaSuRCA (Zimin et al., 2013), IDBA (Peng et al., 2010), SOAPdenovo2 (Luo et al., 2012), ABySS 2.0 (Jackman et al., 2017), and Platanus

(Kajitani et al., 2014). Scripts for these assemblies, specifying parameter settings, can be found in the Appendix. In most cases, default settings were used, changing only the k-mer value (k41, k53 and k71) and setting the genome size to 1,1 Gb.

After successfully completing analyses of the subsets, three *de novo* genome assembly programs (SOAPdenovo 2, ABySS 2.0 and Platanus) were used to assemble the entire quality processed rooibos genome dataset. This dataset comprised the MiSeq and HiSeq data from the 300 bp library (COMP-QP) and the HiSeq data from the 3 kbp mate pair library (MP3-QP). The 8 kbp mate pair library dataset was excluded from these assemblies as it was found to interfere with successful completion of the analyses: SOAPdenovo2 could not estimate the insert size for the 8 kbp library, stating "too few PE links" and Platanus was unable to complete the scaffolding step, stating "Error(6): Kmer mapping exception!! no read mapped in the same contig!!". All assemblies were run with the k-mer values 41 and 71, using default settings, except where specified otherwise. SOAPdenovo2 was used with the -R option to resolve repeats by reads. The asm_flags option in the config file was set to 3 for the COMP-QP dataset (to use this library during contig and scaffold assembly), and 2 for the MP3-QP dataset (using the libraries only during scaffolding step). ABySS 2.0 was run using default settings. Platanus was run in three steps: First, the assembly script was run using only COMP-QP. Then, the scaffolding script was run using COMP-QP and MP3-QP, specifying minimum and average insert size for each library (-n1 200 -n2 2000 -a1 300 -a2 3000). Lastly, the Platanus gap-closing script was run using the COMP-QP and MP3-QP datasets as well as the scaffolds generated in the previous analysis step.

2.2.4. Assembly quality assessment

For assemblies of data subsets, only completion of the assembly and computational statistics (number of cores, amount of RAM, and running time) were recorded. The assemblies of the entire dataset were further investigated for contiguity (fragmentation of assembly and fragment length), completeness (assembled vs predicted genome size) and correctness (how many errors the assembly contains).

First, assemblies were investigated using QUAST-LG (QUAST version 5.0.0), which generates scaffold and contig statistics and permits a first screen on gene content, using GlimmerHMM and BUSCO. GlimmerHMM provides information on the total number of predicted genes (whether they are single copies or multiple copies), and BUSCO reports on near-universal single-copy orthologs. Parameters used for QUAST-LG analysis were: -s -e -b --large --glimmer --est-ref-size 1100000000 --no-snps -m 1000 --contig-thresholds 0,1000,5000,10000,25000,50000,100000 -t 36. QUAST-LG contig statistics were created using the --split-scaffolds flag, which breaks scaffolds that contain continuous fragments of N's of length ≥ 10.

In addition, the comp tool and the spectra-cn plotting tool from KAT were used to visualise missing sequences, as well as expanded and collapsed regions within the assemblies. The comp tool creates a matrix of k-mers shared between paired-end sequencing reads and the corresponding assembly file, and the spectra-cn plotting tool permits visualization the output. comp was run using the COMP-QP dataset with a k-mer length of 27.

Chapter 3: Results

3.1. Illumina data

In total, three libraries had been constructed by the service provider: one paired-end 300 bp insert library, which was subsequently sequenced using MiSeq and HiSeq, and two mate pair libraries with insert sizes of 3 kbp and 8 kbp, sequenced using HiSeq. All datasets, including the MiSeq paired-end, the HiSeq paired-end, and the HiSeq mate pair data, had been preprocessed by the sequencing provider, i.e., adapter sequences and the PhiX spike sequences had been removed. These datasets are referred to as "raw". The first task was to assess read quality and to investigate approaches for improving it where appropriate.

3.1.1. MiSeq paired-end data

Table 1 shows the results for the two MiSeq flow cells. For the raw data, a total of 608,576,970 reads were obtained, with a target insert size of 300 nt. The average read length was 119 bp (ranging between 35 bp and 125 bp), and the average proportion of duplicated reads was 32%. Nearly 95% of the reads had a mean quality score at or above 30, and the average per-base quality scores were never below 30 (Figure 20A).

Table 1: Results for the MiSeq paired-end sequencing reads.

Dataset	Lane	Number of reads	Read length range (bp)	Average read length (bp)	Duplicates (%)	GC (%)	Duplicates F/R (%)	Reads with mean quality score of 30 and above (%)
Raw	MiSeq 1	455 045 310	35 - 125	119	37,4	36	37,7/37,0	93,5
	MiSeq 2	153 531 660	35 - 125	119	27,0	36	27,0/26,9	95,4
PE-Method 1	MiSeq 1	442 430 780	50 - 104	101	36,7	36	36,9/36,5	92,9
	MiSeq 2	149 485 286	50 - 105	101	26,5	36	26,6/26,4	95,0
PE-Method 2	MiSeq 1	446 988 504	60 - 125	120	38,2	36	38,5/37,8	93,7
	MiSeq 2	150 862 318	60 - 125	120	27,5	36	27,6/27,4	95,5

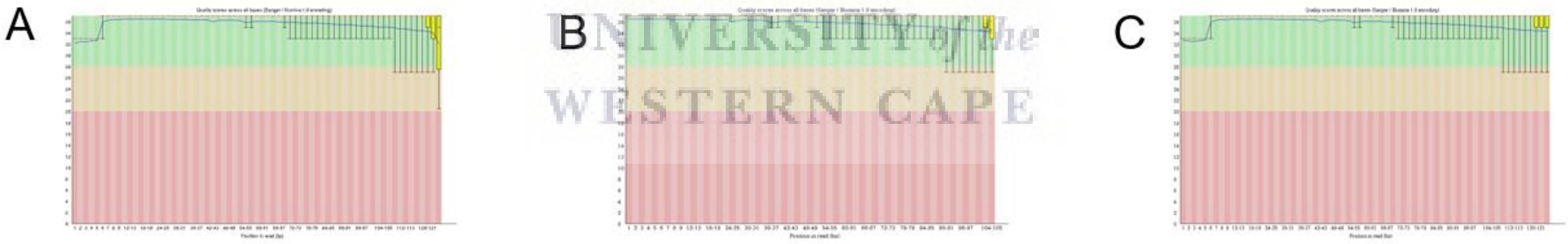


Figure 20: An example per base quality plot generated by FastQC. A) *MiSeq 2, forward raw reads,* ***B)*** *PE-Method 1 quality processed reads, and* ***C)*** *PE-Method 2 quality processed reads*

Per tile analyses of the data showed that all but very few tiles were of high quality, as indicated by the dark blue colour (Figure 21A). The average GC% content for the reads from the MiSeq datasets was 36,0%, as indicated by the large peak in Figure 22A. A smaller peak around 66% was noted.

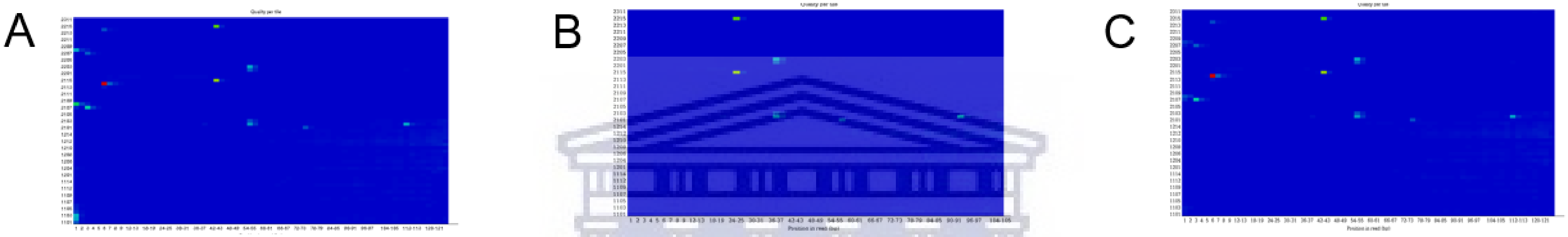


Figure 21: An example per tile sequence quality plot generated by FastQC. A) *MiSeq 2, forward raw reads,* ***B)*** *PE-Method 1 quality processed reads, and* ***C)*** *PE-Method 2 quality processed reads.*

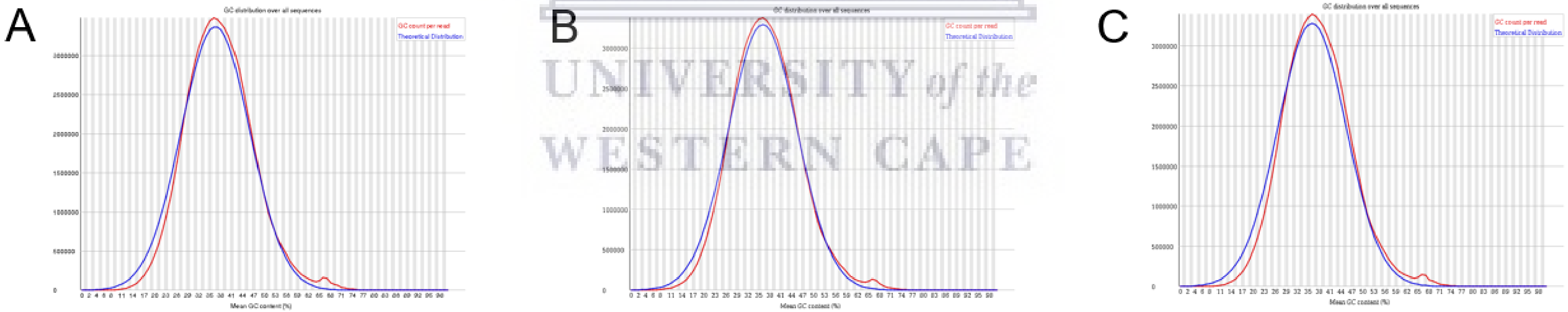


Figure 22: An example per sequence GC content plot generated by FastQC. A) *MiSeq 2, forward raw reads,* ***B)*** *PE-Method 1 quality processed reads, and* ***C)*** *PE-Method 2 quality processed reads.*

The average per base sequence content module failed for both flow cells as large differences in the A and T, and the G and C contents were determined for the first 17 bases, and again for the last two bases of reads (Figure 23A).

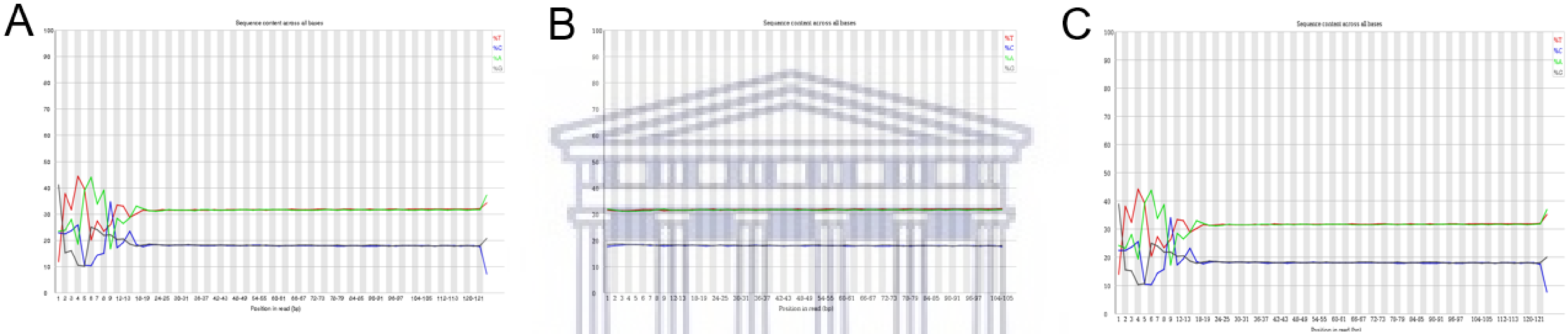


Figure 23: An example per base sequence content plot generated by FastQC. A) *MiSeq 2, forward raw reads,* ***B)*** *PE-Method 1 quality processed reads, and* ***C)*** *PE-Method 2 quality processed reads.*

After performing quality processing using PE-Method 1 (hard crop of a set number of bases from all reads), the number of read-pairs decreased by 16.7 million, and the average read length decreased to 101 bp, now ranging from 50 bp to 105 bp (Table 1). The proportion of duplicated reads was reduced (on average by 0,8%). Neither the percentage of reads with a Phred score at or above 30 nor the average per-base quality scores (Figure 20B) were markedly affected. The per tile quality heatmaps of the MiSeq datasets changed only minimally (Figure 21B). As an example: the heatmap for the flow cell 2 forward read dataset did not show the red tile observed before quality processing (Figure 21A). The average per base sequence content module passed (Figure 23B). The average GC content remained 36,0%, and the small peak at 66% was still present (Figure 22B).

After performing quality processing using PE-Method 2 (quality-based trimming and subsequent error correction), the number of reads decreased by only 10,7 million, and the average read length increased to 120 bp, ranging from 60 bp to 125 bp (Table 1). Here, the proportion of duplicates was somewhat higher (0,7%) than in the raw datasets. The percentage of reads with a Phred score at or above 30 increased by 0,2% for MiSeq 1, and 0,1% for MiSeq 2 (Table 1), and the per tile quality heatmaps (Figure 21C) of the MiSeq sequences changed minimally. The per base sequence content plots failed for all reads, and minimal differences can be seen between the plots of raw and PE-Method 2 quality processed reads (Figure 23C). The average GC% content remained 36,0%, and the small peak at 66% was still present after PE-Method 2 quality processing (Figure 22C).

3.1.2. HiSeq paired-end data

The sequencing of the small insert library using one HiSeq flow cell (6 lanes) in paired-end mode generated nearly 2,3 billion read pairs (12 datasets in total). However, about 1/6th of this data (approximately 0,4 billion reads) did not contain the correct sequencing index, and were filtered out by the service provider. The statistics for the raw determined (with correct index) and the raw undetermined read pairs (without index) are provided in Table 2.

For the determined raw reads, read lengths varied between 35 bp and 125 bp, and the average read length was 119 bp. The proportion of duplicated reads varied substantially between the lanes (ranging from 0% to 22%) and even between the forward and reverse read datasets. Across all lanes, forward read datasets had a five times higher proportion of duplicated reads (22,5%) than the reverse read datasets (4,0%) (Table 2).

Table 2: Results for the HiSeq paired-end sequencing reads.

Dataset	Lane	Number of reads	Read length range (bp)	Average read length (bp)	Duplicates (%)	GC (%)	Duplicates F/R (%)	Reads with mean quality score of 30 and above (%)
Raw Determined								
	Lane 1	284 728 634	35 - 125	119	0,0	36,5	0,0/0,0	64,1
	Lane 2	299 563 658	36 - 125	119	20,2	36,5	28,9/11,4	70,6
	Lane 3	311 852 326	37 - 125	119	15,1	36	30,1/0,0	76,4
	Lane 4	355 362 948	38 - 125	119	13,1	36	26,0/0,2	78,3
	Lane 5	374 476 482	39 - 125	119	22,5	36	32,7/12,4	77,9
	Lane 6	321 173 550	40 - 125	119	8,6	36	17,1/0,2	80
PE-Method 1 Determined								
	Lane 1	276 495 580	50 - 104	101	0,0	36	0,0/0,0	66,4
	Lane 2	290 623 870	50 - 104	101	24,4	35,5	28,7/20,2	72,4
	Lane 3	302 626 294	50 - 104	101	14,4	35,5	28,9/0,0	77,5
	Lane 4	344 709 884	50 - 104	101	15,9	35,5	30,4/1,3	78,9
	Lane 5	363 270 028	50 - 104	101	27,5	35,5	31,9/23,1	78,7
	Lane 6	311 299 952	50 - 104	101	15,2	35,5	23,6/6,8	80,5
PE-Method 2 Determined								
	Lane 1	278 778 534	60 - 125	120	1,2	36,5	2,4/0,0	64,8
	Lane 2	292 564 436	61 - 125	120	24,3	36	31,1/17,4	71,3
	Lane 3	304 280 282	62 - 125	120	14,7	36	29,4/0,0	77,1
	Lane 4	347 367 568	63 - 125	120	18,9	36	30,3/7,5	79,0
	Lane 5	366 378 088	64 - 125	120	27,7	36	34,1/21,2	78,6
	Lane 6	314 166 814	65 - 125	120	15,6	36	24,5/6,7	80,7
PE-Method 2 + Slidingwindow								
	Lane 3	227 020 170	60 - 125	117	14,4	36	28,7/0,2	91,3
Raw Undetermined								
	Lane 1	41 552 130	35 - 125	123	0,1	40	0,2/0,0	53,7
	Lane 2	45 072 562	36 - 125	122	17,0	39,5	24,5/9,4	60
	Lane 3	48 471 116	37 - 125	122	12,6	38,5	25,2/0,0	65,7
	Lane 4	53 480 014	38 - 125	122	11,0	38,5	21,9/0,1	67,5
	Lane 5	57 858 318	39 - 125	122	19,0	38,5	27,8/10,2	67
	Lane 6	128 159 942	40 - 125	121	7,3	37,5	14,5/0,1	72,3

In total, 75% of the reads had a mean quality score at or above 30. Yet, the per tile sequence quality modules of all datasets failed, and the heat maps indicated that the reverse reads were of worse quality than the forward reads (Figure 24A-B). The average per base sequence quality modules also failed, showing substantial differences in the A and T, and the G and C contents for the first 18 bases, and again for the last two bases of the reads (Figure 25A). The GC content plot for the determined reads showed a significant peak at 36,2%, and a smaller peak around 66% (Figure 26A).

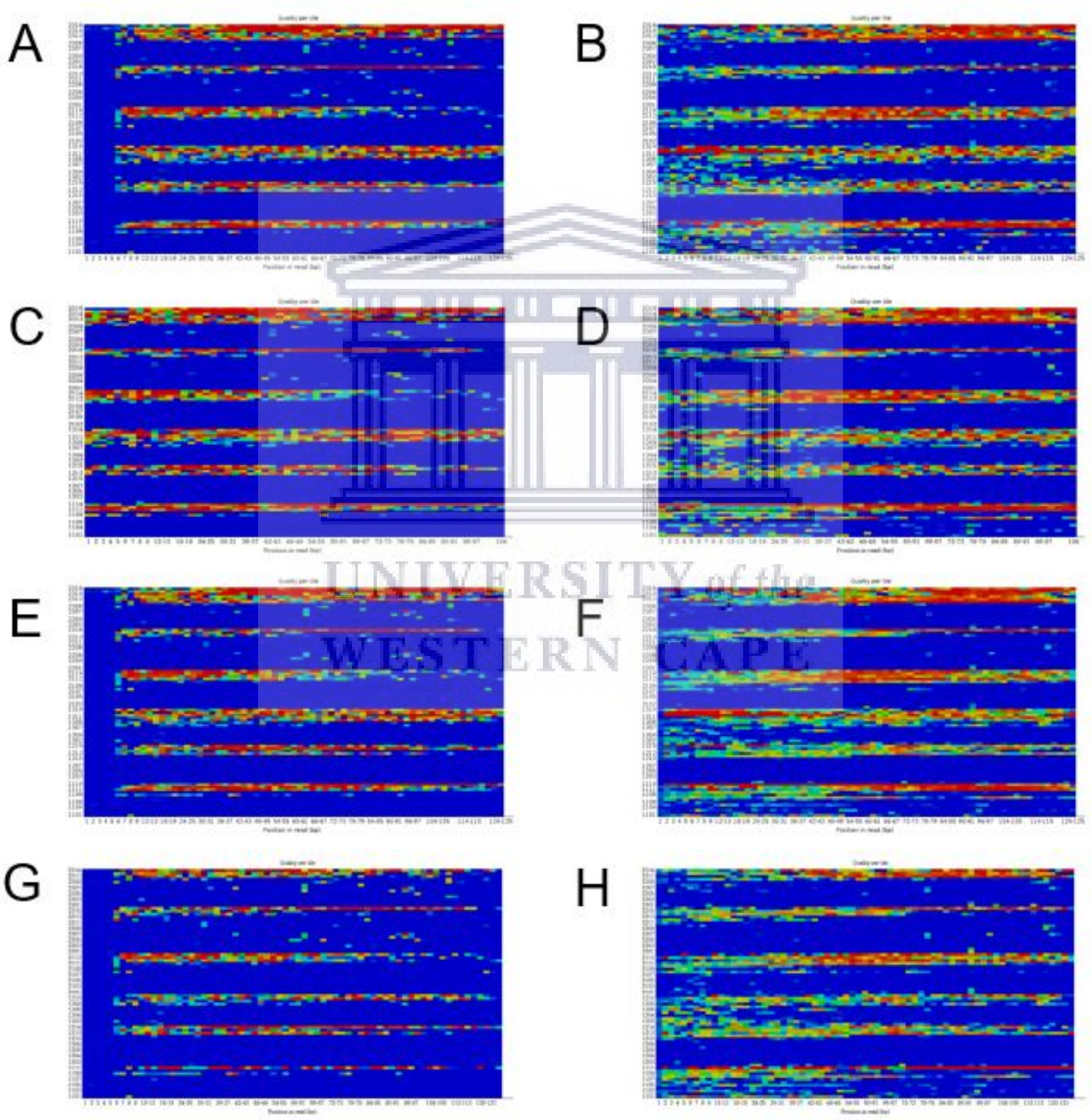


***Figure 24: Per tile sequence quality plots generated by FastQC for Lane 3**: **A)** forward raw reads, **B)** reverse raw reads, **C)** forward PE-Method 1 processed reads, **D)** reverse PE-Method 1 processed, **E)** forward PE-Method 2 processed reads, **F)** reverse PE-Method 2 processed reads, **G)** forward SLIDINGWINDOW processed reads, and **H)** reverse SLIDINGWINDOW processed reads*

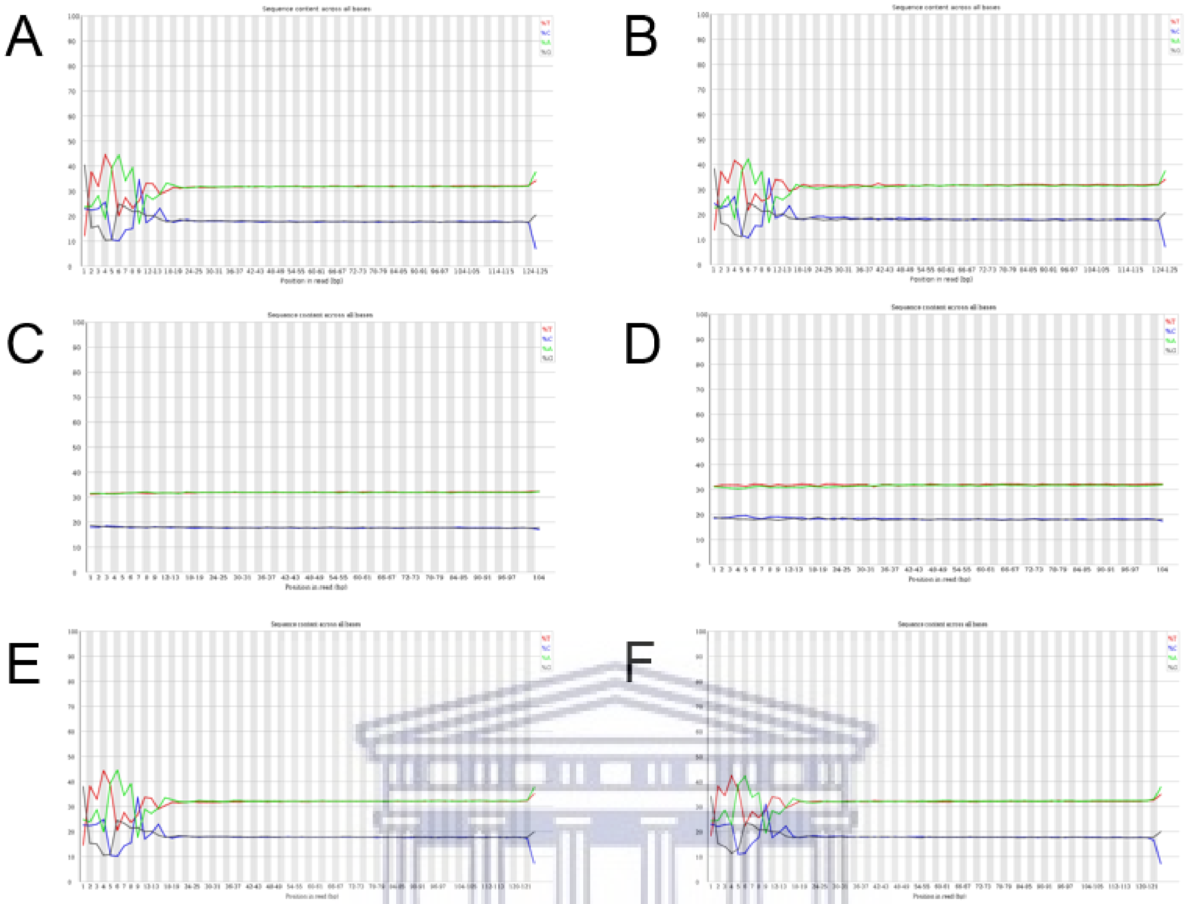


***Figure 25: Per base sequence content plots generated by FastQC for Lane 3: A)** forward raw reads, **B)** reverse raw reads, **C)** forward PE-Method 1 processed reads, **D)** reverse PE-Method 1 processed, **E)** forward PE-Method 2 processed reads, and **F)** reverse PE-Method 2 processed reads.*

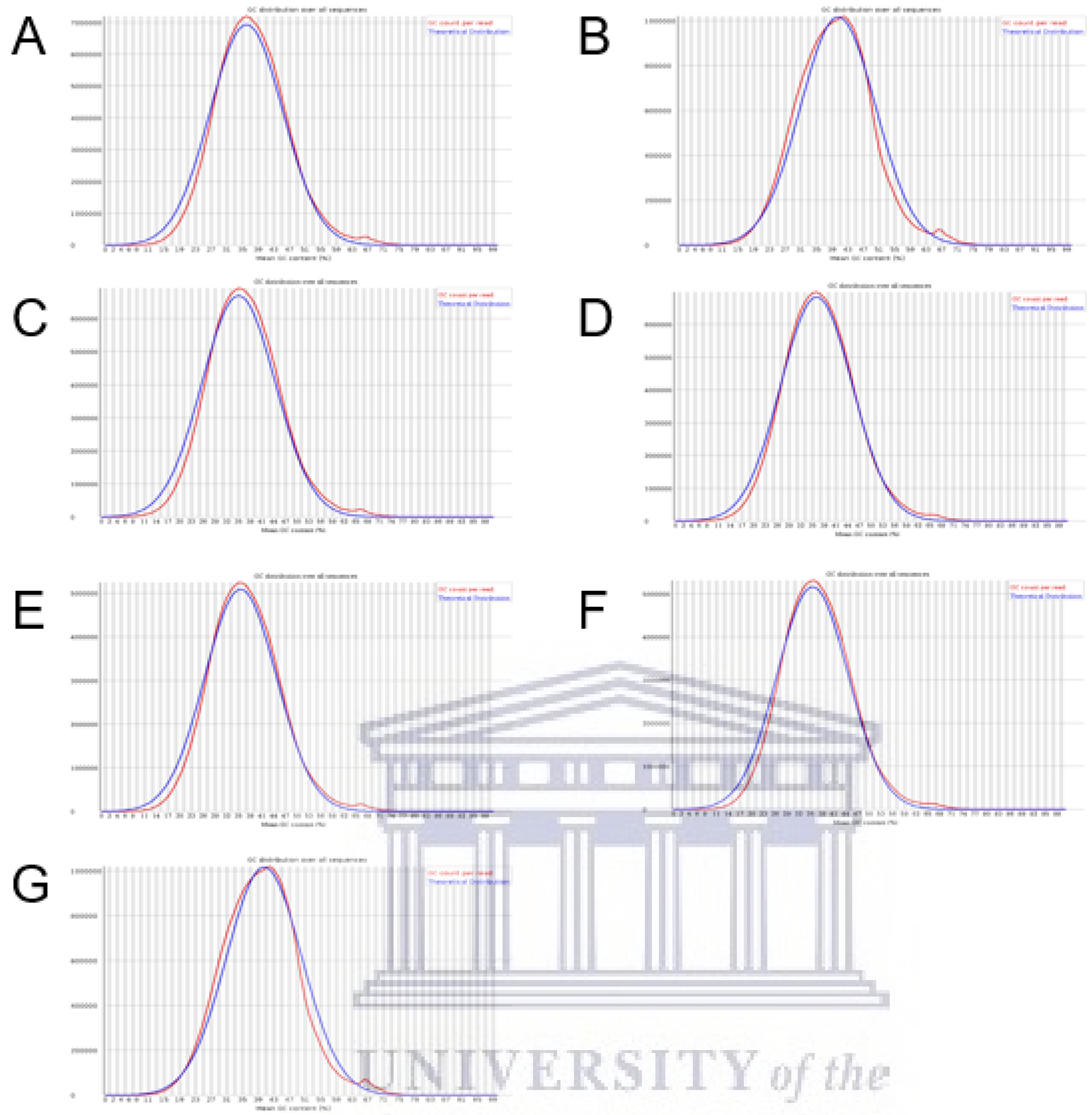


***Figure 26: Per sequence GC content plots generated by FastQC for Lane 3*: A)** *forward raw reads,* **B)** *reverse raw reads,* **C)** *forward PE-Method 1 processed reads,* **D)** *reverse PE-Method 1 processed,* **E)** *forward PE-Method 2 processed reads,* **F)** *reverse PE-Method 2 processed reads, and* **G)** *forward raw undetermined reads.*

The undetermined reads were somewhat longer (on average 122 bp), but also of lower quality than the determined reads: only 64,4% of the undetermined reads had a quality score at or above 30. Interestingly, the average GC value was somewhat higher in the undetermined datasets (38,8%; Table 2) The undetermined datasets also had a small peak at 66% GC (Figure 26G). Because of the missing index, the origin of the undetermined reads could not be established. Considering that 1) the HiSeq flow cell was used to sequence all three Nextera libraries (the paired-end small insert and both mate pair large insert libraries), and that 2) a substantial number of the undetermined reads may represent the PhiX spike, the undetermined read datasets were excluded from all subsequent analyses.

Quality processing of the determined reads using PE-Method 1 resulted in a loss of 58 million read pairs and reduced the average read length from 119 bp to 101 bp, read lengths now ranging from 50 bp to 104 bp (Table 2). Across all lanes, the proportion of duplicated reads increased from 13,3% to 16,3%, the forward read datasets still containing approximately three times the number of duplicated reads as compared to the reverse read datasets. The proportion of reads with a mean quality score at or above 30 increased by 1,1% (Table 2). The processing had no beneficial visual effect on the per tile sequence quality, and this module still failed for all lanes (Figure 24C-D). The per base sequence content of all lanes passed the module, as the trimming of the 19 first bases reduced the differences between the A and T contents and the G and C contents to less than 10% throughout the length of sequences (Figure 25C-D). The average GC content decreased to 35,5%, and the minor peak was observed at 66% (Figure 26C-D).

After quality processing of the determined reads with PE-Method 2, the total number of read pairs decreased by 44 million (Table 2). Lane 3 was further processed using the SLIDINGWINDOW method, which reduced the number of reads by another 77 Million. However, this analysis step also substantially improved read quality, as visualized in Figure 24 and Figure 27. The final dataset (including the additional processing step for Lane 3), comprised 1.8 billion read pairs with an average read length of 120 bp, ranging from 60 bp to 125 bp. The average duplication rate amounted to 17,5% across all lanes, and forward read datasets still had a higher percentage of duplicated reads than the reverse read datasets (25,2% vs 8,8%) (Table 2). The proportion of reads with a quality score at or above 30 increased by 3% to 77,6%. The per tile quality heat maps generated by FastQC showed some improvement, but this module still failed for all lanes (Figure 24E-F). The effect of the PE-Method 2 quality processing on the per-base sequence content plot was minimal (Figure 25E-F). The module failed for 8 of the 12 datasets (Figure 25E); for the reverse reads of Lanes 1, 2, 3 and 5 the message changed from "failed" to "warning" (implying that the difference between the A and T, and the G and C contents were now between 10% and 20% throughout the length of sequences for these reads) (Figure 25F). The average GC content was 36,1%, and the smaller peak was again observed at 66% (Figure 26E-F).

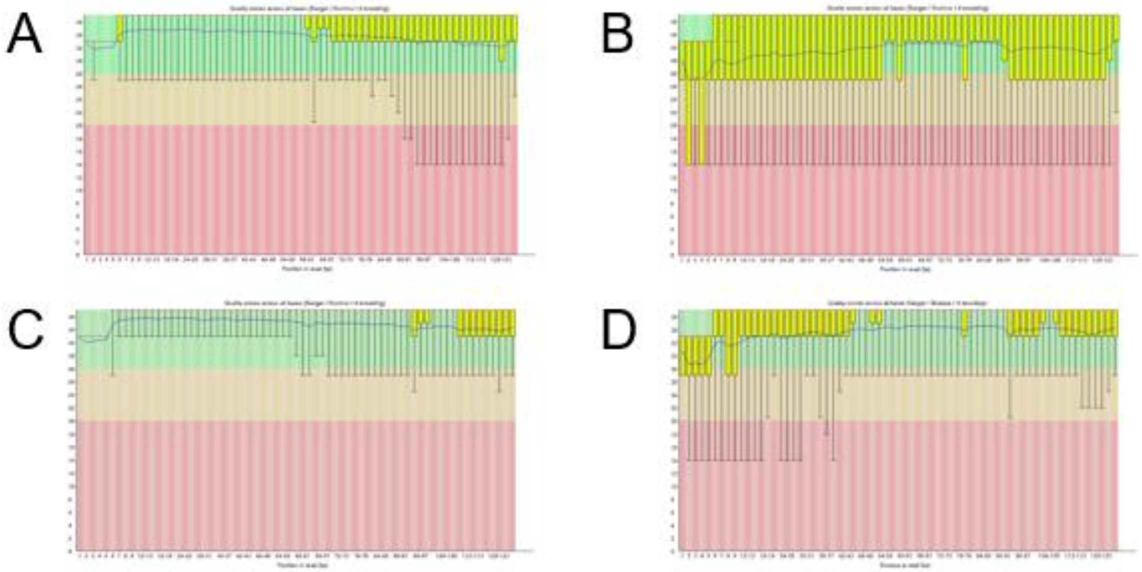


Figure 27: Comparison of the per base sequence quality plots generated by FastQC for Lane 3 before, and after using SLIDINGWINDOW: ***A)*** *forward PE-Method 2 processed reads,* ***B)*** *reverse PE-Method 2 processed reads,* ***C)*** *forward reads processed using PE-Method 2 plus SLIDINGWINDOW:4:17,* ***D)*** *reverse reads processed using PE-Method 2 plus SLIDINGWINDOW:4:17. Reads represented by* ***C)*** *and* ***D)*** *were used for downstream analysis.*

3.1.3. HiSeq mate-pair data

Two mate pair libraries, a 3 kb library (MP3; target insert size of 3,000 bp), and an 8 kb library (MP8; target insert size of 8,000 bp), had been constructed by the service provider and sequenced using HiSeq using one lane per library. All quality evaluation analysis results are provided in Table 3. In total, nearly 300 million read pairs with an average read length of 92 bp (35 bp - 101 bp) were generated. However, 73% of the reads were duplicates. Less than 4% of the reads in the 8 kb insert library (MP8) were unique (Figure 28). Read quality for the mate pair library was quite high: the proportion of reads with a quality score at or above 30 were 91% (MP3) and 81% (MP8) (Table 3), and the per-base sequence quality ranged between a Phred score of 30 and 40 (Figure 29A-B). The per base sequence content module failed for all four datasets, although the differences in the A and T, and the G and C contents at the beginnings and ends of the reads were not very high (Figure 30A-B and Figure 31A-B). Only a small number of tiles were flagged by the per tile sequence quality module of FastQC in any of the datasets (Figure 32). The average GC% of both libraries was 36,0%. The GC content plots indicated that both libraries contained a high percentage of reads with <1% GC; and that the average GC content of the reads was 36,0% (Figure 33A-B). In both libraries, many reads still contained the Nextera transposase adapter sequence (Figure 34A-B), and had high proportions of undetermined bases (Ns) within the first 34 bases (7,8% for MP3 and 16% for MP8; Figure 35A-B).

Table 3: General statistics generated by MultiQC, indicating percent duplicates, percent GC, average sequence length, percentage fails and total sequences for forward and reverse reads for both the MP libraries.

Dataset	Lane	Number of reads	Read length range (bp)	Average read length (bp)	Duplicates (%)	GC (%)	Reads with mean quality score of 30 and above (%)
Raw	3 kbp	93 213 534	35 - 101	94,5	49,8	36,0	90,5
	8 kbp	201 478 550	36 - 101	89,2	96,1	36,0	80,9
MP-Method 1	3 kbp	60 154 150	40 – 101	91,3	43,9	36,0	98,2
	8 kbp	112 968 692	40 – 101	91,2	93,9	35,0	96,4
MP-Method 2	3 kbp	30 388 300	25 - 151	67,3	10,0	36,0	99,1
	8 kbp	51 964 340	25 - 166	67,5	17,1	35,0	97,9
MP-Method 2 and trimmed	8 kbp	51 963 434	2 - 127	67,4	16,7	35,0	98,0

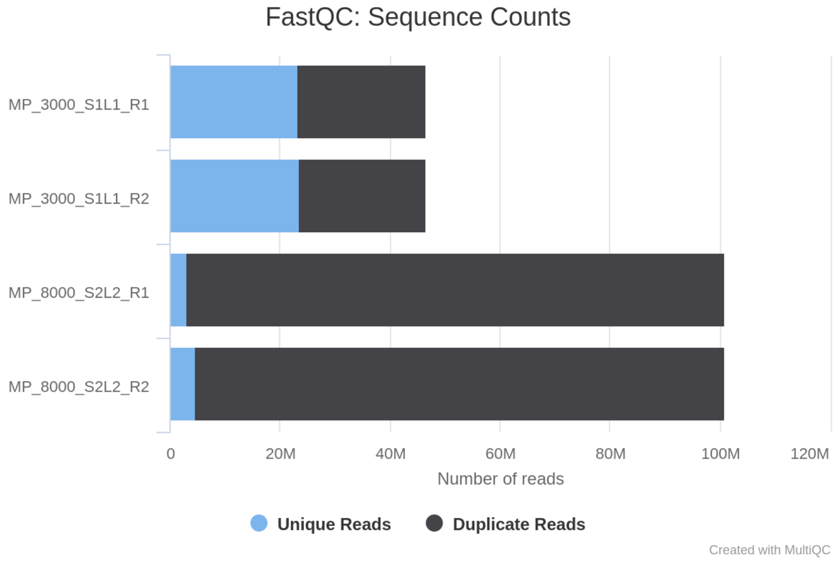


Figure 28: Graph generated by MultiQC shows the comparison of the number of reads from libraries MP3 and MP8. *The unique reads are shown in blue, and the duplicate reads are shown in black.*

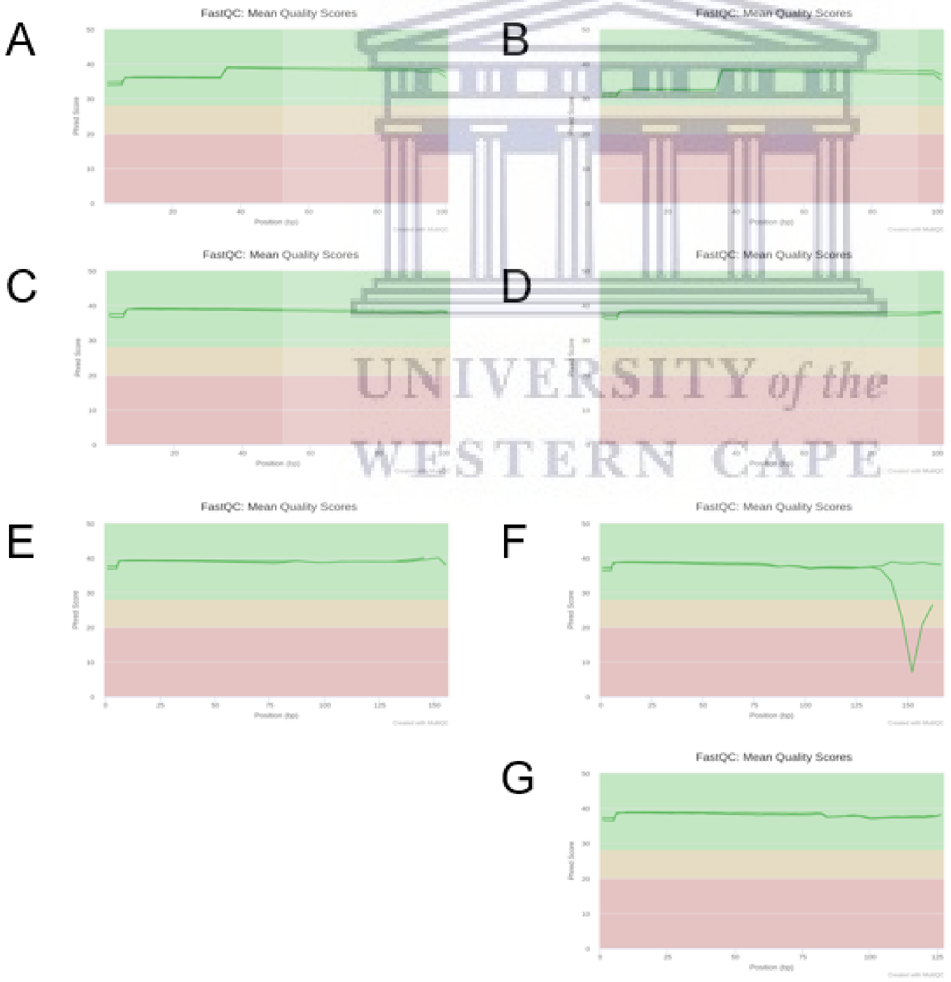


Figure 29: Results compiled using MultiQC of the mean quality value across each base position in the forward and reverse reads for the: A) *MP3 raw reads,* ***B)*** *MP8 raw reads,* ***C)*** *MP3 MP-Method 1 reads****,*** ***D)*** *MP8 MP-Method 1 reads,* ***E)*** *MP3 MP-Method 2 reads, and* ***F)*** *MP8 MP-Method 2 reads.* ***G)*** *MP8 MP-Method 2 + Trimmed reads.*

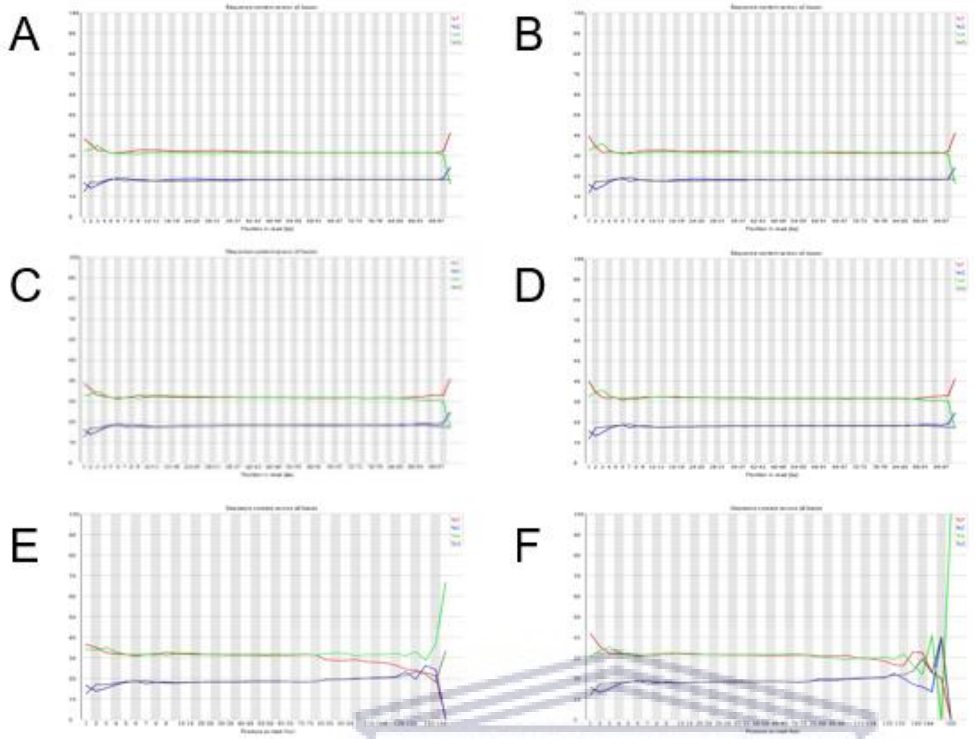


***Figure 30: Comparison of the per base sequence content plots of MP3 generated by FastQC : A)** raw forward reads, **B)** raw reverse reads, **C)** MP-Method 1 forward reads, **D)** MP-Method 1 reverse reads, **E)** MP-Method 2 forward reads, and **F)** MP-Method 2 reverse reads.*

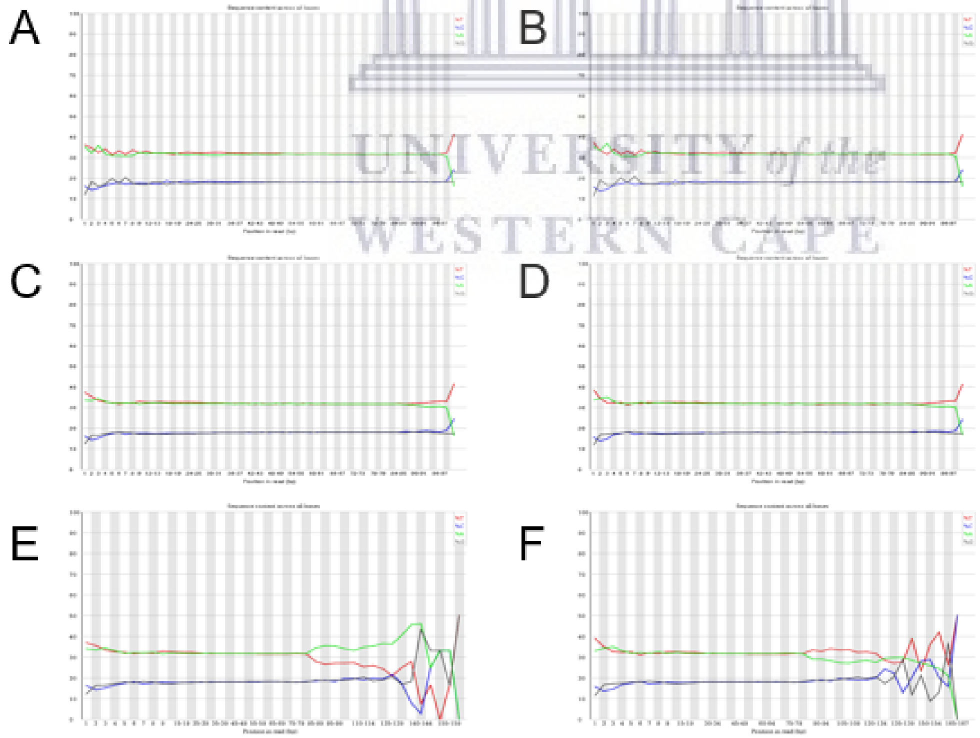


***Figure 31: Comparison of the per-base sequence content plots of MP8 generated by FastQC: A)** raw forward reads, **B) r**aw reverse reads, **C)** MP-Method 1 forward reads, **D)** MP-Method 1 reverse reads, **E)** MP-Method 2 forward reads, and **F)** MP-Method 2 reverse reads.*

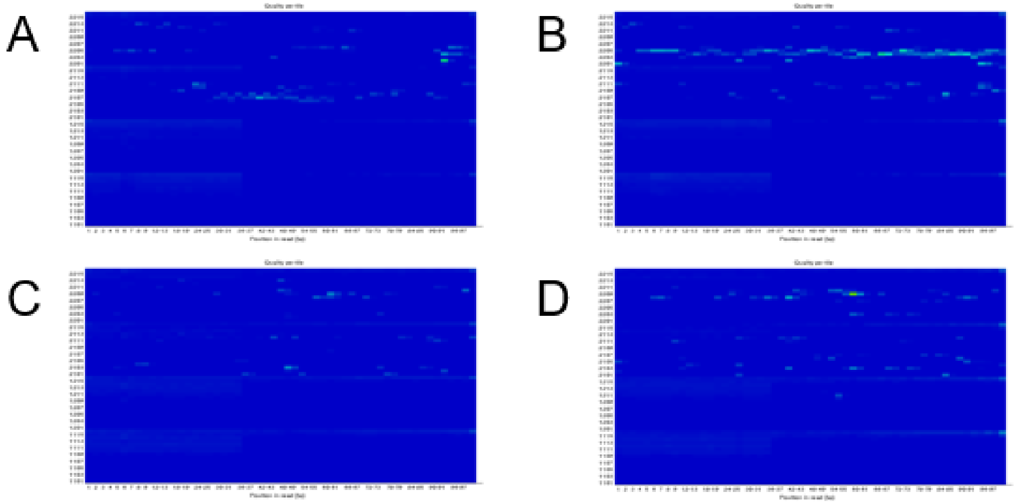


Figure 32: Per tile sequence quality plots generated by FastQC for the raw mate pair libraries. A) *MP3 forward reads,* ***B)*** *MP3 reverse reads,* ***C)*** *MP8 forward reads,* ***D)*** *MP8 reverse reads.*

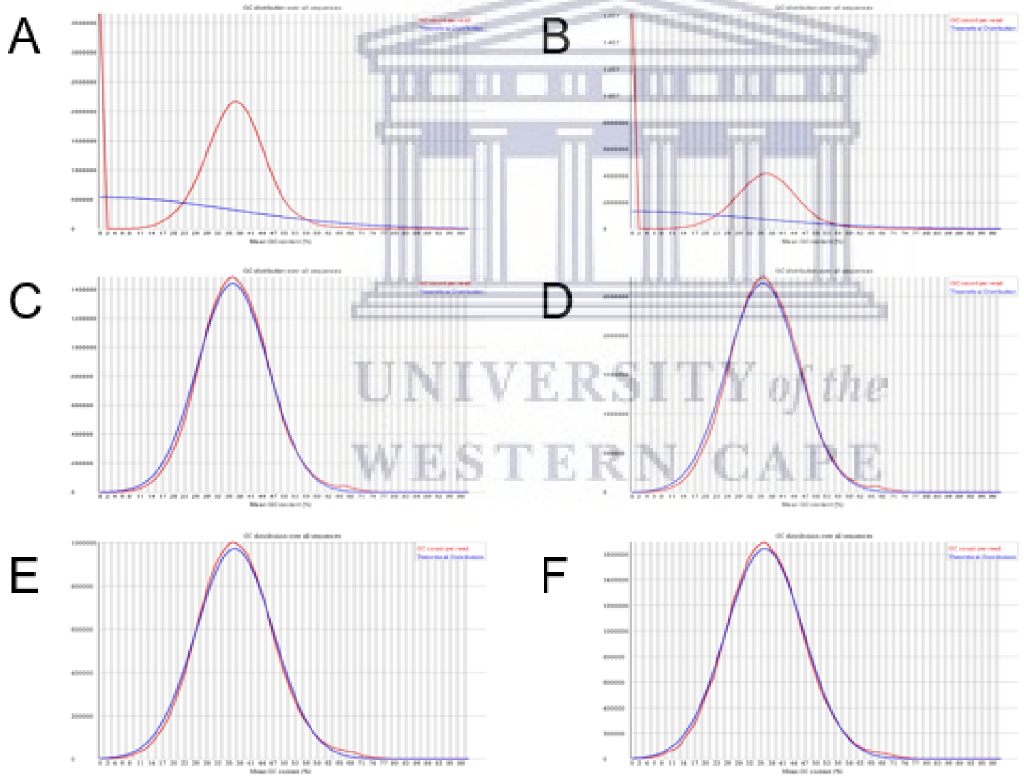


Figure 33: Comparison of the per sequence GC content plots for the forward reads of MP3 and MP8, generated by FastQC. A) *MP3 raw reads,* ***B)*** *MP8 raw reads,* ***C)*** *MP3 MP-Method 1 reads,* ***D)*** *MP8 MP-Method 1 reads,* ***E)*** *MP3 MP-Method 2 reads, and* ***F)*** *MP8 MP-Method 2 reads. The blue line indicates the theoretical distribution, and the red line shows the actual GC count per read.*

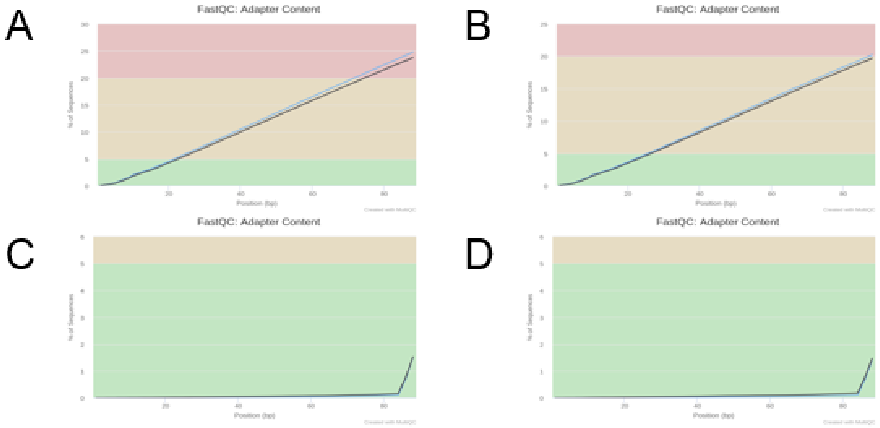


Figure 34: MultiQC results of the adapter content plots of each mate pair library, which show the presence of the Nextera transposase sequence. *In each plot, the blue lines indicate forward reads and the black lines indicate reverse reads.* ***A)*** *raw MP3 reads,* ***B)*** *raw MP8 reads,* ***C)*** *MP-Method 1 MP3 reads, and* ***D)*** *MP-Method 1 MP8 reads. MultiQC reported that no adapters were present for either MP3 or MP8 MP-Method 2 processed reads (not shown in the figure).*

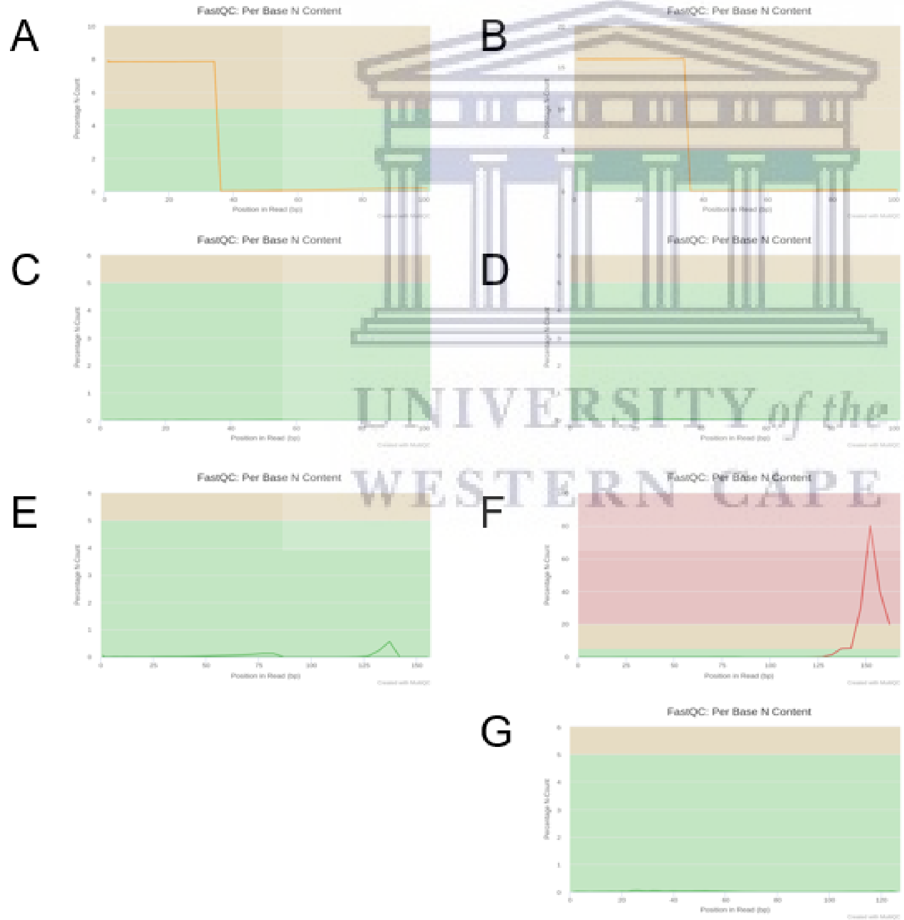


Figure 35: MultiQC results of the percentage of base calls at each position of reads for which an N was called. ***A)*** *raw MP3 reads,* ***B)*** *raw MP8 reads,* ***C)*** *MP-Method 1 MP3 reads,* ***D)*** *MP-Method 1 MP8 reads,* ***E)*** *Method 2 MP3 reads,* ***F)*** *MP-Method 2 MP8 reads, and* ***G)*** *MP8 MP-Method 2 + Trimmed reads.*

Quality processing using MP-Method 1 (only Trimmomatic) resulted in a loss of 35,5% of read pairs from MP3, and 43,9% of the read pairs from MP8 (Table 3). The range of read lengths for both mate pair libraries improved (40 - 101 bp), but the average read length did not change (91 bp). Quality trimming decreased the proportion of duplicated reads in the datasets by 6% in MP3 and by 2% in MP8. The percentage of reads with a mean quality at or above 30 increased by 8% and 16% in the MP3 and MP8 libraries, respectively. The per base sequence content plot changed minimally, and the last base pair position still had >20% deviation between the A and T content (Figure 31C-D). The per tile sequence quality remained mostly unchanged for both mate pair libraries. Quality processing effectively removed the low GC content reads without affecting the average GC content (36,0% in MP3 and 35,0% in MP8; Table 3). The small peak at 66% GC content was now visible in the mate pair data (Figure 33C-D). The Nextera transposase sequence was still detected in up to 1,5% of the reads (Figure 34C-D), but the proportion of N's decreased to 0% for all forward and reverse reads of both libraries (Figure 35C-D).

Quality processing using MP-Method 2 (lmp_processing script from w2rap) reduced the mate pair datasets to 30 million mate pairs for MP3 and 52 million mate pairs for MP8. Read lengths were drastically reduced, now averaging only 67 bp and ranging between 25 bp to 166 bp. The proportion of duplicated reads decreased to 10% in MP3, and 17% in MP8 (Table 3). Nearly all reads had a Phred score at or above 30 throughout the entire sequence; only in the forward read dataset of the MP8 library the average per-base quality substantially decreased after the 137 base (Figure 29F). The per base sequence content module failed for all four datasets: in both libraries, a substantial deviation between A% and T% after the 80th position in the reads was observed (Figure 30E-F and Figure 31E-F). The per tile sequence quality information was not present after processing; the reason remains unknown. The quality-filtered datasets did not contain low GC% reads, and the average GC content was 36,0%, with a small peak discernible in the 66% region (Figure 33E-F). Nextera transposase adapters were not found in the sequence data after processing. The proportion of N's were undetectably low at the beginning of the sequences (first 34 bases), but increased in the forward read dataset of MP8 after the 142 base, reaching a maximum of 80% at base 152 (Figure 35F). To address low quality read ends, the MP8 library datasets were further processed using Trimmomatic (v0.36) to trim all reads to a maximum length of 127 bp, which substantially improved the mean quality per base (Figure 29G) and reduced the proportion of N's at the read ends (Figure 35G). Due to the low quality, and effort required from the MP8 library, these datasets were excluded in the subsequent analyses. The MP3 library processed using MP-Method 2 was used in downstream analysis and will be referred to as MP3-QP.

3.2. Investigation of genome characteristics

The rooibos genome sequencing data was used to predict genome characteristics, including genome size, level of heterozygosity, and repeat content, computationally. BBNorm was employed to produce k-mer histogram files for four datasets (MiSeq-Raw, MiSeq-QP, COMP-Raw, and COMP-QP, where COMP represents the combined MiSeq and HiSeq datasets and QP represents the quality filtering PE-Method 2) at different k-mer values (k19, k23, k27, k47). Subsequently, three programs (GenomeScope, FindGSE, BBNorm) were compared for their ability to estimate the above genomic parameters. Since the program KAT does not permit external histogram files as input, it was used to generate the histograms for the same datasets and to estimate genome size and heterozygosity levels.

3.2.1. K-mer spectra graphs

The programs KAT, GenomeScope, and FindGSE, produce k-mer spectra graphs that visualize k-mer frequency distributions, providing first information about sequence quality and genome characteristics. Figure 36 shows the results for the COMP-Raw and the COMP-QP datasets at k19 obtained using the programs KAT, GenomeScope2, and FindGSE (the other spectra graphs are provided under Supplementary Material: Supplementary Figures 1-7). All graphs visualized three peaks, irrespective of the dataset and k-mer value.

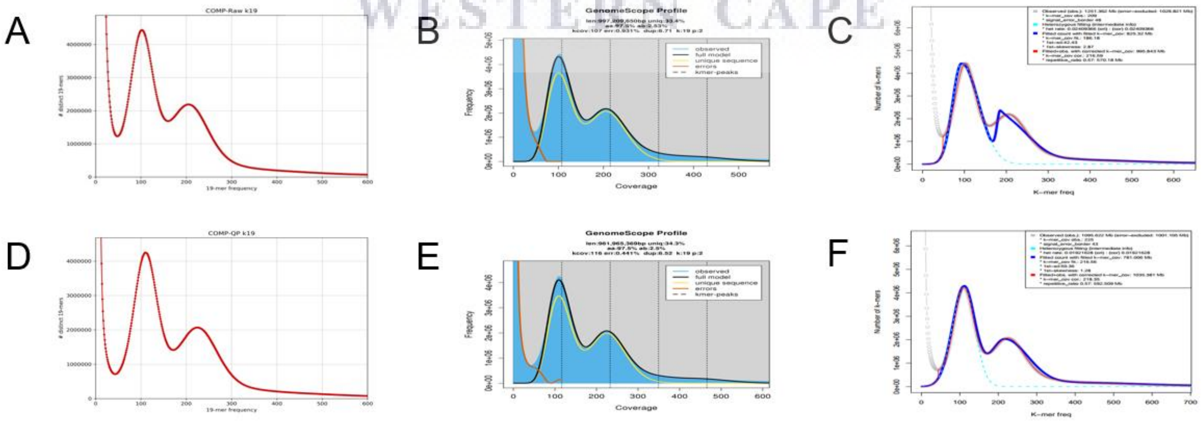


Figure 36: The effect of quality processing on k-mer spectra graphs. Results from the programs KAT, GenomeScope2, and FindGSE at k19. Histogram files were generated by KAT using the COMP-Raw and COMP-QP datasets. A) *KAT COMP-Raw,* ***B)*** *GenomeScope2 COMP-Raw,* ***C)*** *FindGSE COMP-Raw.* ***D)*** *KAT COMP-QP,* ***E)*** *GenomeScope2 COMP-QP, and* ***F)*** *FindGSE COMP-QP.*

The first peak corresponds to low-frequency k-mers, which are associated with sequencing errors. In the raw datasets, their numbers ranged between 24 billion and 43 billion. In the processed datasets, there were 14 billion to 17 billion k-mers, reflecting effective reduction of erroneous k-mers due to read processing. Depending on the dataset, the lowest point of the first valley appeared at 22x to 48x k-mer coverage. At this point, a lower coverage indicates a narrower first peak, which is associated with fewer erroneous k-mers. Increasing the k-value reduced the k-mer coverage value for the lowest point of the first valley, shifting it to the left of the graph (Figure 37). This was observed for both the raw and the processed datasets, respectively. The effect of read processing on the k-mer coverage differed depending on the k-value: at k19, the processed reads had 5x lower k-mer coverage; at k47, the processed reads had 5x higher coverage than that of the raw reads. In addition to the graphs, the program GenomeScope2 provided numeric values: at k19, the error rate was 0,93% for raw and 0,44% for processed reads, and at k47 these numbers were 0,66% for raw and 0,27% for processed reads. These values do not correspond with the above values calculated manually. The maximum error rate calculated by GenomeScope2 (0,93%) was therefore below the maximum acceptable error rate of 2% suggested in the GenomeScope1 paper as a threshold for k-mer analysis (Vurture et al., 2017b).

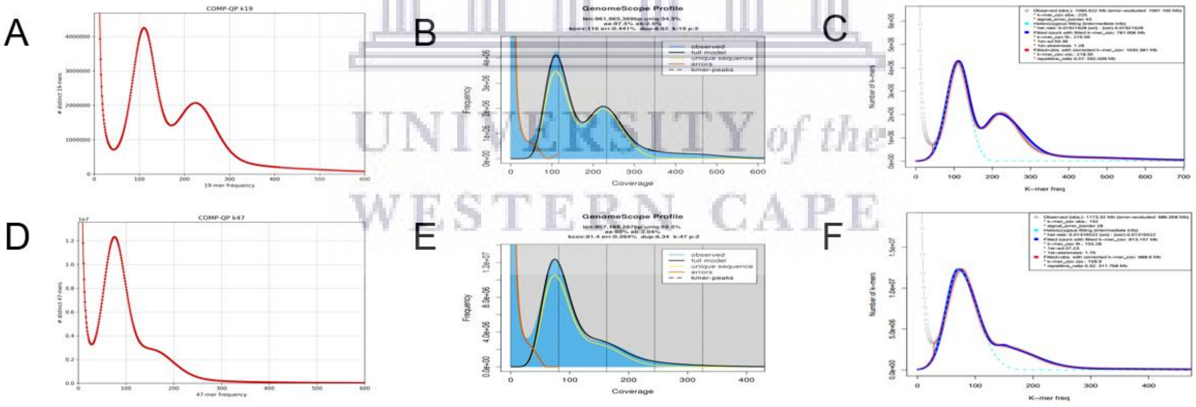


Figure 37: The effect of k-value on k-mer spectra graphs. Results from the programs KAT, GenomeScope2, and FindGSE at k19 and k47. Histogram files were generated by KAT using the COMP-QP dataset. A) *KAT k19,* ***B)*** *GenomeScope2 k19,* ***C)*** *FindGSE k19,* ***D)*** *KAT k47,* ***E)*** *GenomeScope2 k47, and* ***F)*** *FindGSE k47.*

The second and third peaks in the k-mer graphs represented k-mers from the heterozygous and homozygous genome locations, respectively. In all graphs, the second peak was notably higher than the third peak, indicating that the rooibos genome is highly heterozygous. Increasing the k-value shifted the second and third peaks to the left of the graph, as seen in the KAT and FindGSE k-mer spectra graphs (Figure 37). Moreover, choosing higher k-values also increased heterozygous content, simultaneously reducing homozygous content (causing peak three to shift towards peak two) and lowered high copy k-mer numbers in the tail of the graph. At k47, the second valley was basically non-existent due to the substantial overlap of peaks two and three. In contrast, quality processing of reads shifted the second and third peaks to the right, best seen in the KAT output (Figure 36A and D).

In the linear graphs, no further peaks were observed, although the frequency histogram included k-mers at extremely high coverage (up to 900000x). GenomeScope (v1 and v2) also generates a logarithmic k-mer graph, which visualized additional peaks in the high-k-mer coverage regions of the graphs (Figure 38). These additional peaks were observed in all investigated datasets at all k-mer values. Two small peaks (arrows in Figure 38) are present in all four graphs at k19 and k47 of processed and raw datasets. The first peak is present between 1,000x and 10,000x, and the second peak is present around 100,000x (between 1e+04 and 1e+06). With an increase in k-value, both peaks shift towards the left of the graph.

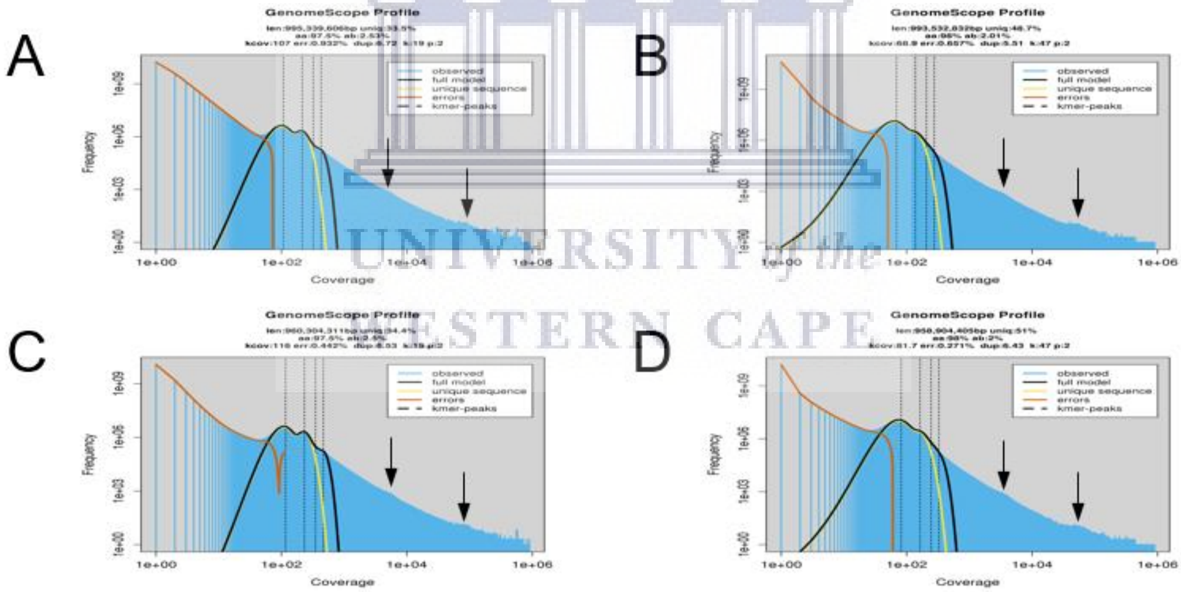


***Figure 38: Log-scale graphs generated by GenomeScope2 using the BBNorm-generated histogram files for: A)** COMP-Raw at k19, **B)** COMP-Raw at k47, **C)** COMP-QP2 at k19, and **D)** COMP-QP2 at k47.*

3.2.2. Genome size estimation

Genome size was estimated using five methods, including the four dedicated programs KAT, GenomeScope, FindGSE, and BBNorm, and a simple formula first introduced by the research group of M. S. Waterman. For each analysis method, the effects of 1) sequence subset vs complete dataset, 2) k-mer size, and 3) raw vs quality-filtered data were investigated. The results are provided in Table 4.

The performance of GenomeSope (both v1 and v2) was strongly affected by parameter settings: the rooibos genome size estimates varied from 0,51 Gb to 1,01 Gb. The most influential parameter was the cutoff threshold for maximum k-mer coverage (CovMax). Increasing the threshold from 1k (default setting of GenomeScope v1) to 10k or 900k resulted in average increases of genome size estimates by 0,14 Gb and 0,33 Gb, respectively. Genome size estimates also increased with increasing k-mer size. The differences were higher at the lower CovMax settings, ranging from 0,17 Gb at 1k to 0,01 Gb at 900k. For GenomeScope, the effects of using the MiSeq subsets vs complete datasets and raw vs quality processed data were small (<0.10 Gb). FindGSE predicted a rooibos genome size of 1,06 ± 0.03 Gb (averaged over all tested parameters). With this program, the differences between the MiSeq subset and corresponding values in the complete dataset were small (ranging from 0,01 Gb to 0,09 Gb). Increasing k-mer size only marginally increased genome size estimates (max by 0,04 Gb), and differences between raw and quality processed datasets were also small (max 0,04 Gb). BBNorm estimated a rooibos genome size of 1,08 ± 0,03 Gb. The differences between the MiSeq subset and the complete dataset were minimal (varying from 0,00 to 0,06 Gb, with higher estimates obtained for the MiSeq dataset). An increase in k-mer size increased genome size estimates by only 0,05 Gb. Differences between quality processed and raw datasets amounted to a maximum of 0,04 Gb. When using the formula, the rooibos genome size estimate amounted to 1,03 ± 0,04 Gb. The effects of dataset size, k-mer size and data quality were also small (at most 0,08 Gb, 0,05 Gb, and 0,04 Gb, respectively). The choice of histogram file (BBNorm or KAT) did not substantially affect the above results: on average, the values differed by 0,004 ± 0,016 (Supplementary Table 1). Therefore, the results obtained using KAT are comparable with those described above. When averaged across all datasets, KAT predicted the largest genome size, but also showed the highest standard deviation (1,18 Gb ± 0,22 Gb). This was mainly due to the large effect of the k-mer size: genome size estimates ranged between 0,93 at k19 and 1,49 at k47. The effects of dataset size and quality processing were minor (0,10 Gb and 0,05 Gb, respectively).

Table 4: Genome size estimates for rooibos (in Gb) using raw and quality processed Illumina data.

	K19		K23		K27		K47			
MiSeq (0,6 Billion read pairs)	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Average**	**SD**
GenomeScope v1 (CovMax 1k)	0,53	0,52	0,59	0,58	0,63	0,62	0,74	0,74	0,62	0,08
GenomeScope v2 (CovMax 1k)	0,53	0,51	0,58	0,56	0,61	0,60	0,73	0,73	0,61	0,08
GenomeScope v1 (CovMax 10k)	0,71	0,69	0,75	0,74	0,78	0,77	0,85	0,85	0,77	0,06
GenomeScope v2 (CovMax 10k)	0,70	0,69	0,74	0,72	0,76	0,75	0,84	0,83	0,76	0,06
GenomeScope v1 (CovMax 900k)	1,00	0,97	1,00	0,97	1,01	0,98	1,00	0,97	0,99	0,02
GenomeScope v2 (CovMax 900k)	1,00	0,96	0,99	0,96	0,99	0,96	0,99	0,96	0,98	0,02
FindGSE	1,01	1,04	1,10	1,04	1,03	1,05	1,04	1,06	1,05	0,03
BBNorm	1,07	1,04	1,08	1,04	1,08	1,05	1,11	1,07	1,07	0,02
Formula	1,07	1,02	1,06	1,01	1,06	1,02	1,00	0,97	1,03	0,03
KAT*	1,00	1,01	1,14	1,13	1,22	1,13	1,59	1,57	1,17	0,23
MiSeq + HiSeq (1,9 Billion read pairs)	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Average**	**SD**
GenomeScope v1 (CovMax 1k)	0,60	0,60	0,64	0,64	0,67	0,67	0,74	0,75	0,66	0,06
GenomeScope v2 (CovMax 1k)	0,59	0,59	0,63	0,63	0,66	0,66	0,74	0,74	0,66	0,06
GenomeScope v1 (CovMax 10k)	0,76	0,76	0,79	0,79	0,81	0,81	0,84	0,85	0,80	0,03
GenomeScope v2 (CovMax 10k)	0,76	0,76	0,79	0,79	0,80	0,80	0,83	0,84	0,80	0,03
GenomeScope v1 (CovMax 900k)	0,97	0,97	0,97	0,97	0,97	0,97	0,95	0,95	0,97	0,01
GenomeScope v2 (CovMax 900k)	0,97	0,96	0,97	0,96	0,96	0,96	0,94	0,94	0,96	0,01
FindGSE	1,05	1,06	1,08	1,06	1,08	1,09	1,13	1,11	1,08	0,03
BBNorm	1,05	1,06	1,08	1,06	1,08	1,09	1,14	1,13	1,09	0,03
Formula	1,07	1,03	0,98	0,97	1,08	1,06	1,01	1,02	1,03	0,04
KAT*	0,91	0,94	1,09	1,14	1,18	1,19	1,49	1,48	1,18	0,22

K19–K47: k-mer sizes; QP: Quality Processed; CovMax: cutoff threshold for maximum k-mer coverage (varied for GenomeScope from 1k to 900k, 900k for FindGSE and BBNorm); SD: Standard Deviation.

*Note: All results were generated from BBNorm files except the KAT results

3.2.3. Heterozygosity

The programs GenomeScope, FindGSE, BBNorm and KAT were used to estimate heterozygosity in the rooibos genome sequencing data (Table 5). As indicated by the k-mer spectra graphs, rooibos appears to be very heterozygous: the average heterozygosity rate was 2,28 ± 0,37 %. However, the choice of program, quality processing of the reads and k-mer size had an impact.

The most consistent results were obtained with GenomeScope, where heterozygosity rates varied between 2,02% and 2,67%. The differences were mostly associated with the choice of the version (v1 estimates were on average 0,12% higher than v2 estimates), and with the k-mer size (the higher the k-mer, the lower the estimate; the difference between k19 and k47 amounting to 0,43%). Quality filtering of the reads and increasing CovMax reduced heterozygosity rates at most by 0,03%. For FindGSE, heterozygosity rates ranged between 1,15% and 2,89%. As with GenomeScope, increasing k-mer sizes decreased the estimates (from an average of 2,41% at k19 to an average of 1,27% at k47). With this program, quality processing of the reads reduced heterozygosity rate estimates by up to 0,43%. The average heterozygosity rate obtained using BBNorm (1,68 ± 0,10 %) was lower than with the above programs. In contrast to the above programs, estimates increased with increasing k-mer size. At k47, BBNorm predicted a haploid genome, and heterozygosity rates were not calculated. Quality processing of the reads had negligible effects, increasing the estimates on average by 0,03%. The heterozygosity rates calculated by KAT were substantially lower than the rates calculated by the other programs, ranging between 0,00% and 0,05%. These KAT results were in stark contrast to the information indicated by the KAT k-mer spectra, where the height of the second peak indicated a high heterozygosity rate.

Table 5: Heterozygosity rate estimates for rooibos (in Gb) using raw and quality processed Illumina data.

	K19		**K23**		**K27**		**K47**			
MiSeq + HiSeq (1,9 Billion read pairs)	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Average**	**SD**
GenomeScope v1 (CovMax 1k)	2,67	2,58	2,64	2,58	2,55	2,51	2,25	2,34	2,51	0,15
GenomeScope v2 (CovMax 1k)	2,57	2,54	2,53	2,49	2,42	2,39	2,04	2,03	2,38	0,22
GenomeScope v1 (CovMax 10k)	2,62	2,55	2,60	2,54	2,51	2,47	2,22	2,29	2,47	0,15
GenomeScope v2 (CovMax 10k)	2,57	2,53	2,52	2,49	2,42	2,38	2,03	2,02	2,37	0,22
GenomeScope v1 (CovMax 900k)	2,60	2,54	2,58	2,53	2,50	2,46	2,20	2,27	2,46	0,15
GenomeScope v2 (CovMax 900k)	2,57	2,53	2,52	2,49	2,42	2,38	2,03	2,02	2,37	0,22
FindGSE	2,89	1,92	1,78	1,84	2,25	1,69	1,39	1,15	1,86	0,53
BBNorm	1,54	1,57	1,71	1,74	1,73	1,76	N/A (0)	N/A (0)	1,68	0,10
KAT	0,00	0,03	0,01	0,00	0,03	0,05	0,03	0,02	0,02	0,02

*Note: All results were generated from BBNorm files except the KAT results

Table 6: Estimated repeat content for rooibos (in %) using raw and quality processed Illumina data.

	K19		**K23**		**K27**		**K47**			
MiSeq + HiSeq (1,9 Billion read pairs)	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Average**	**SD**
GenomeScope v1 (CovMax 1k)	34,20	33,96	30,76	30,01	30,15	29,28	32,15	32,20	31,59	1,72
GenomeScope v2 (CovMax 1k)	36,44	35,03	32,23	30,83	31,23	30,00	33,52	32,74	32,75	2,04
GenomeScope v1 (CovMax 10k)	50,96	50,98	45,88	45,36	43,82	43,07	41,28	40,82	45,27	3,67
GenomeScope v2 (CovMax 10k)	52,66	51,83	47,10	46,09	44,77	43,76	42,52	41,37	46,26	3,86
GenomeScope v1 (CovMax 900k)	65,25	64,96	59,77	58,71	56,67	55,19	50,16	48,51	57,40	5,73
GenomeScope v2 (CovMax 900k)	65,67	65,60	60,72	59,33	57,47	55,80	51,26	49,04	58,11	5,66
FindGSE	57,47	57,57	52,61	50,88	47,78	46,41	36,42	33,98	47,89	8,25
BBNorm	69,67	69,08	64,39	63,38	60,85	59,60	50,78	48,24	60,75	7,30
KAT*	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

*KAT does not calculate repeat content.

3.2.4. Repeats

Repeats are defined as sequences within a genome that occur more than 2x (for a diploid organism). The repeat content of the rooibos genome was estimated using the programs GenomeScope, FindGSE, and BBNorm (Table 6).

The average repeat content predicted by GenomeScope (across both versions) was 45,23 ± 11,35%, version choice affecting the results by at most 0,68%. The most influential parameter was CovMax. Increasing the threshold from 1k to 10k and 900k increased predicted repeat contents by 13,60% and 25,59%, respectively. At CovMax 900k, the average repeat content amounted to 57,76 ± 5,54%. At CovMax 10k and 900k, k-mer size also substantially affected repeat content predictions. As the k-mer size increased from k19 to k47, predicted repeat contents decreased on average by 2,26% (1k), 10,11% (10k), and 15,63% (900k). In contrast, the effect of quality processing was negligible, the largest difference amounting to 1,34%.

The average repeat content predicted by FindGSE was 47,89 ± 8,25%. The estimates were substantially affected by k-mer length: predicted repeat contents decreased from 57,52% at k19 to 35,20% at k47. Quality processing again had little effect, reducing the values by max 1.36%. BBNorm predicted the highest repeat content for the rooibos genome (on average 60,75 ± 7,81%), when compared to the other methods. As observed with GenomeScope and FindGSE, increasing the k-mer size decreased the estimates, the highest difference amounting to 19,87% (k19 vs k47). Quality processing again had only minor effects, reducing the predicted repeat contents by at most 1,35%.

When taking into consideration only the 900k histograms, the mean predicted repeat content (averaged across the three programs, data processing, and k-mer lengths) for the rooibos genome was 56,04% ± 8,52%. The effects of quality processing of the data were negligible (1,27%), but the effect of k-mer length substantial (max 18,36% between k19 and k47).

3.3 Draft genome assembly analysis

3.3.1 Assembler tests

The first aim was to identify assemblers, that would complete analyses of a data subset (comprising one or two lanes of the HiSeq paired-end dataset and the quality processed data from MP3) and/or the entire quality processed Illumina sequencing dataset (including the MiSeq and HiSeq data from the 300 bp insert library and the HiSeq data from MP3) on the SANBI and/or the CHPC clusters within a suitable period. Table 7 provides an overview of the used programs, program requirements, and successful/unsuccessful runs. Six programs were tested using data subsets, namely ALLPATHS-LG, MaSuRCA, IDBA, SOAPdenovo 2, ABySS 2.0, and Platanus.

Two programs, ALLPATHS-LG and MaSuRCA, did not complete the analyses. ALLPATHS-LG had problems with allocating memory. It was using only 140 GB of RAM and 32 cores, although access to 500 GB of RAM and 32 cores was provided. A range of output files were either truncated or not created at all, leading to program termination. Even stepwise creation of the input files (first creating the cache libraries, then creating the cache groups, and then creating the input files for the program) did not alleviate the problem. Subsequently, it was discovered that ALLPATHS-LG only accepts overlapping read pairs, which was not immediately clear from the description in the manual ("paired reads of length ~100 bases from fragments of 180 bp" http://software.broadinstitute.org/allpaths-lg/blog/?page_id=215). With an insert size of 300 bp and an average read length of approximately 100 bp, few read pairs of the small insert library were expected to overlap. Although error and output files indicated which files were missing, determining the cause as to why these files were not created or why they were truncated was hard. Surprisingly, despite program failure, ALLPATHS-LG produced a large amount of output data (nearly 1 TB). Support for ALLPATHS-LG is located on a Google Groups page, and feedback on queries range from days to months, with some queries not being answered at all. An advantage of this program is that it can be rerun from wherever it terminated.

Table 7: Results from the assembler tests on a subset or complete datasets for six assembly programs.

Subset dataset						
Program	ALLPATHS-LG	MaSuRCA	IDBA	SOAPdenovo2	AbySS 2.0	Platanus
Server	SANBI	SANBI	CHPC	CHPC	CHPC	SANBI
Dataset	HiSeq L3 & L5, & MP3	MiSeq 1 & 2, & MP3	L6	HiSeq L3 & L5, & MP3-QP	HiSeq L3 & L5, & MP3-QP	HiSeq L3 & L5, & MP3-QP
Data type	QP-1	QP-1	QP 1	Raw, QP 1 and QP	Raw, QP 1 and QP	Raw, QP 1 and QP
Completion of analysis	No	No	Yes	Yes	Yes	Yes
Amount of data generated	N/A	N/A	N/A	74 GB	37 GB	4,5 GB
Time (hours)	N/A	357,4	30,44	14,2	8,6	N/A
Number of cores (CPU efficiency)	32	24	112 (34% efficacy)	32 (16% efficiency)	32 (71% efficiency)	32
Memory requirements	>500 GB	170,8 GB	193,8 GB	233,32 GB	252,33 GB	359,7 GB
Complete dataset						
Program	SOAPdenovo2	AbySS 2.0	Platanus			
Server	CHPC	CHPC	CHPC			
Dataset	Complete	Complete	Complete			
Data type	QP	QP	QP			
Completion of analysis	Yes	Yes	Yes			
Amount of data generated	409 GB	43 GB	15 GB			
Time (hours)	36,58 hours	22,55 hours	74 hours			
Number of cores (CPU efficiency)	56 (17% efficiency)	56 (69% efficiency)	56 cores (64% efficiency)			
Memory requirements	602 GB	908 GB	747.9 GB			

MaSuRCA was the second program that did not run to completion even with a subset of the data. This program requires a configuration file, and information on how to prepare the configuration file is available online. The input files can be submitted as multiple compressed paired-end FASTQ files. The subset analysis used 24 cores and 171 GB RAM, and ran for 357 hours, generating a large number of folders and output files. The error and output files were not helpful in determining why the program was not completing the analyses. These files pointed to directories and files that would be helpful for troubleshooting, but the content in those files was very technical and required such extensive technical knowledge on the program that problems could not be addressed. MaSuRCA is available and supported on GitHub (https://github.com/alekseyzimin/masurca), but many of the queries on the page remain unanswered. MaSuRCA is able to be restarted where it was terminated.

IDBA proved to be a very cumbersome program: it permits only one unzipped fasta-formatted input file. This implied 1) unzipping of all four files, 2) reformatting them to fasta format, 3) concatenating the complementary forward and reverse read datasets to generate one file per lane with interleaved read pairing, and 4) concatenating the resulting output files into one. Identification of this problem was difficult, because neither the manual nor the error messages were informative in this regard. IDBA successfully completed the analysis with one data subset, using 193,8G RAM and 56 cores (34% efficiency) and taking 30 hours to complete. However, due to the challenging requirements for data input, this program was not used to analyze the complete dataset.

SOAPdenovo2 completed the analyses with all investigated datasets, requiring between 14 to 30 hours for genome assembly. The program requires configuration of an input file, where certain parameters such as insert size, as well as the location and names of datasets, are provided. Multiple datasets can be provided, and the datasets can be left zipped. While memory usage was similar to that of other programs, the amount of output data was 2-30 fold higher, reaching 0,4 TB with the complete datasets. Output files included contig and scaffold sequences, as well as files which provided information on edges, read location on contigs, reads likely to be located in gaps (which can be used to close gaps), the position of contigs in scaffolds, info on contigs that form bubble structures in scaffolds, files that can be used to resolve short repeats, and statistics for the pre-graph step, as well as contigs. There was almost no support for SOAPdenovo2. Problems encountered during the use of the program were submitted to GitHub and emailed directly to the creator, but were never addressed. The program was initially indicating segmentation fault errors with the complete dataset, and neither me, nor the CHPC support team could resolve the problem. Care was taken to ensure that the configuration file had been set up correctly. Several months later, the program was successfully rerun, and it worked consistently thereafter. The cause and fix of the problem were never identified. Another setback was that it could not be restarted where it had left off previously.

ABySS 2.0 also completed assemblies with all tested datasets. This program required the most amount of RAM (up to 908 GB) and used the highest number of cores at high efficiency (56 cores, 69% efficiency). This may explain the comparatively short running time. It was easy to understand and run the program, with clear examples given on the GitHub page. The support was very good, as all queries were answered. Another positive aspect of ABySS 2.0 was that it could be rerun from where it last terminated. The amount of output data was independent of the input dataset, averaging 40 GB. Descriptions for the output files generated by AbySS can be found on the wiki page at https://github.com/bcgsc/abyss/wiki/ABySS-File-Formats.

Platanus was the third assembler that successfully completed analyses of all datasets. It requires input of decompressed, paired-end FASTQ files. The analysis is conducted in three distinct steps (assembly of contigs, scaffolding, and gap-closing) and is comparatively time-consuming: assembly of the entire assembly required 74 hours. Platanus produced the least amount of output data (~5 GB for the subsets and ~15 GB for the whole dataset). Output data included the assembled contig, scaffold and gap-closed scaffold sequences, the merged and removed bubble sequences, a table describing contig joins, and a k-mer frequency distribution file. During the analysis, a log file informed on the progress of the analysis at each step, providing information on the memory usage. Due to the step-wise analysis, the program can be restarted, given the completion of a step.

3.3.2. Comparison of genome assemblies

For the assembly of the complete rooibos genome data, Platanus, ABySS 2.0, and SOAPdenovo 2 were run using two k-mer values (k41 and k71) on the quality processed datasets (COMP-QP and MP3-QP). Assembly quality was assessed by comparing length statistics, BUSCO results, QUAST-LG gene prediction results, and "assembly spectra copy number plots" or spectra-cn plots produced by the KAT comp tool. The results are displayed in Figure 39 and Table 8A and B

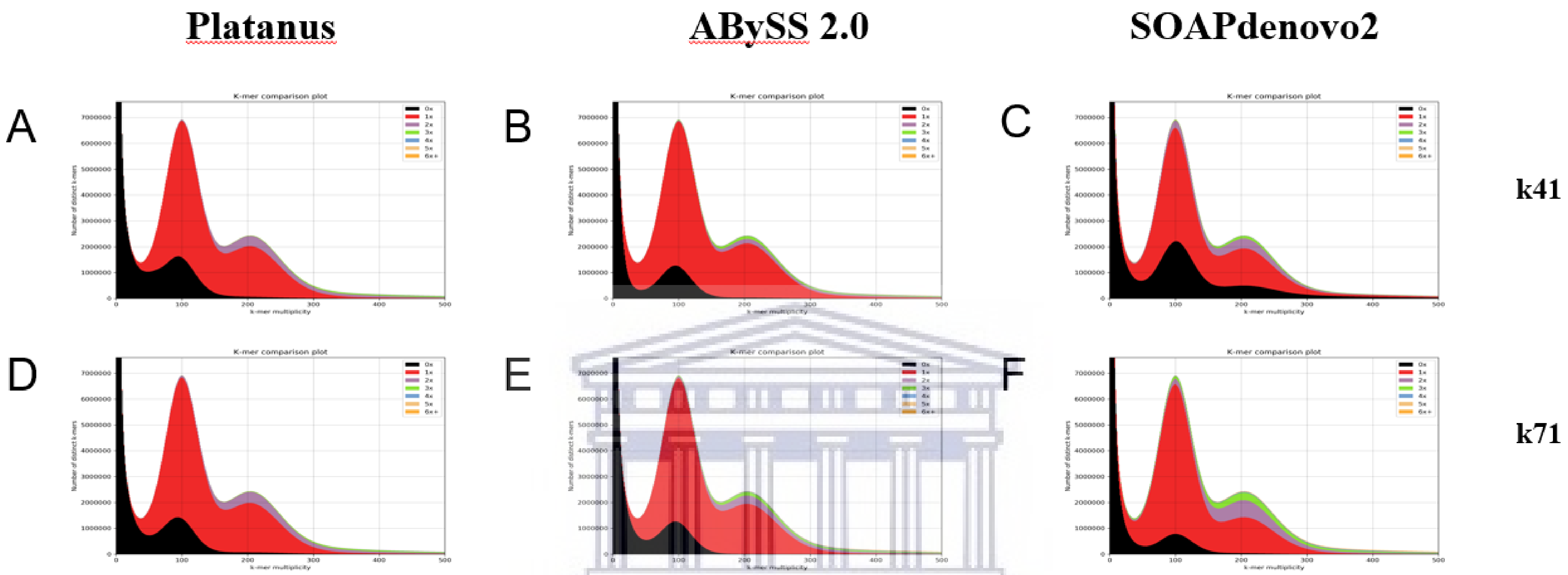


Figure 39: KAT spectra-cn plots comparing the PE-QP reads to the six assembly scaffolds*. Plots were generated using k=27. The colour of the plots indicate how many times k-mers from the reads appear in each assembly. Black indicates k-mers missing from the assembly, red indicates k-mers that are present once in the assembly, green, twice, etc..*

Table 8A: Assembly scaffold statistics generated by QUAST-LG, with the minimum scaffold length set to 1 kbp

Assembly	Abyss-71	Abyss-41	Platanus-71	Platanus-41	Soapdenovo2-71	Soapdenovo2-41
Number of scaffolds	138 266	65 664	186 411	170 125	258 027	27 349
N50	3 257	2 539	9 470	10 568	4 092	1 444
NG50	-	-	7 304	7 766	2 393	-
N75	2 032	1 722	4 327	4 748	2 086	1 172
NG75	-	-	1 819	1 632	-	-
Largest contig	35 754	23 448	177 918	144 124	74 616	7 692
Number contigs > 10k bp	1 861	237	25 287	25 593	9 316	0
Number contigs > 50k bp	0	0	221	306	4	0
Number contigs > 100k bp	0	0	9	9	0	0
# N's per 100 kbp	47	155	8 610	8 210	15 994	2
Total length	387 767 872	153 762 673	922 263 025	899 383 410	784 509 931	40 604 886
Complete BUSCO (%)	45,21	32,01	51,49	69,97	12,87	5,28
Partial BUSCO (%)	22,11	20,13	20,46	13,86	14,19	6,6
Complete + partial	67,32	52,14	71,95	83,83	27,06	11,88
Missing BUSCOs (%)	32,67	47,85	28,05	16,17	72,94	88,12
# predicted genes (unique)	114 323	51 311	229 718	217 824	126 682	18 708
# predicted genes (>= 0 bp)	115 731	51 228	230 000	217 892	133 539	18 412
# predicted genes (>= 300 bp)	64 831	31 854	145 268	140 307	45 875	10 788
# predicted genes (>= 1500 bp)	16 020	9 365	29 801	31 607	6 469	1 037
# predicted genes (>= 3000 bp)	5 300	2 836	9 048	10 837	1 484	99

Table 8B: Assembly contig statistics generated by QUAST-LG, with the minimum contig length set to 1 kbp.

Assembly	Abyss-71	Abyss-41	Platanus-71	Platanus-41	Soapdenovo2-71	Soapdenovo2-41
Number of contigs	139 682	67 148	247 381	231 773	116 230	27 346
N50	3 203	2 426	4 696	4 970	564	1 443
NG50	-	-	3 171	3 212	139	-
N75	2 017	1 668	2 391	2 505	262	1 172
NG75	-	-	1 030	-	-	-
Largest contig	35 754	21 191	69 460	65 547	17 843	7 692
Number contigs > 10k bp	1 673	166	11 308	11 978	107	0
Number contigs > 50k bp	0	0	8	8	0	0
Number contigs > 100k bp	0	0	0	0	0	0
# N's per 100 kbp	2	10	3	4	6	0
Total length	387 367 308	151 993 878	832 658 333	811 621 596	200 796 213	40 598 836
Complete BUSCO (%)	44,22	29,70	39,93	58,42	9,24	5,28
Partial BUSCO (%)	22,77	20,13	25,08	20,79	10,56	6,60
Complete + partial	66,99	49,83	65,01	79,21	19,80	11,88
Missing BUSCOs (%)	33,00	50,17	34,98	20,79	80,20	88,12
# predicted genes (unique)	114 898	52 164	253 145	241 463	144 217	18 707
# predicted genes (>= 0 bp)	116 305	52 065	254 027	241 902	144 849	18 411
# predicted genes (>= 300 bp)	64 996	32 102	149 795	144 341	43 220	10 788
# predicted genes (>= 1500 bp)	15 936	9 091	27 697	29 754	3 520	1 037
# predicted genes (>= 3000 bp)	5 177	2 564	6 486	8 816	721	99

The best assemblies were generated by the program Platanus. In comparison to ABySS 2.0 and SOAPdenovo2, it produced longer and more contiguous assemblies that yielded substantially higher BUSCO matches and numbers of predicted genes**.** K-mer length did not substantially affect contig and scaffold assembly statistics, and results were thus averaged over the two k-values. The nearly 240,000 contigs assembled into approximately 180,000 scaffolds, 14% of which were larger than 10 kbp. Depending on the k-mer size, between 221 and 306 scaffolds were larger than 50 kbp, and the longest assembled scaffold was 178 kbp. Total assembly length amounted to 910 ± 16 Mbp. The average N50 and N75 values were 10 kbp and 4,5 kbp, respectively; and the average NG50 and NG75 values amounted to 7,535 and 1,726, respectively. On average, the proportion of N's per 100 kbp was 0,8%. The assembly with k71 generated the higher number of contigs and scaffolds in total, as well as the longest contig, the longest scaffold, and the longest total assembly length.

The k41 assembly produced a higher number of large scaffolds, and had higher N50 and N75 values for both, contig and scaffold assemblies. Genome annotation statistics were also better for the k41 assembly: 70% of the BUSCO hits were covered to completeness, and 16% were missed. The assembly spectra-cn plots indicated that the Platanus assemblies might have only a small amount of errors (Figure 39). Both assemblies had a black heterozygous peak (at the 100x coverage), which indicated missing heterozygous content from the assemblies. For K41, both the valley before this peak and the peak itself were higher than at k71, implying that more heterozygous content was missing at the lower k-mer value. Purple and green peaks, which indicate duplicated and triplicated assembly contents, were present in both assemblies. Although this content did not substantially differ between the k41 and k71 assemblies, the Platanus-71 assembly contained slightly more duplicated and triplicated content than the Platanus-41 assembly.

ABySS 2.0 performed second-best. Interestingly, here scaffolding did not substantially affect assembly, although k-mer choice had a substantial effect on contig and scaffold statistics. Abyss-71 generated a higher number of scaffolds and contigs, larger N50 and N75 values, as well as the longest contig and fewer number of N's per 100 kbp. For Abyss-41, just over 67,148 contigs were assembled into 65,664 scaffolds, and for Abyss-71, 139,682 contigs were assembled into 138,266 scaffolds. However, only 0,5% of the scaffolds were larger than 10 kbp (237 at k41 and 1,861 at k71), and none were larger than 50 kbp. The longest assembled contig was 35,75 kbp (Abyss-71), which could not be extended by scaffolding. The total assembly length of Abyss-71 was 2,5 times larger than that of Abyss-41. The N50 and N75 values did not differ substantially between the contig and scaffold assemblies (scaffold values amounting to 2,539 and 1,722, respectively for Abyss-41, and 3,257 and 2,032, respectively for Abyss-71). The NG50 and NG75 values could not be generated due to the short total assembly length. Genome annotation statistics were somewhat better for the k71 assembly: the number of complete BUSCO hits and the number of predicted genes (≥ 3kb) were higher than in the k41 assembly (13% more complete BUSCOs and 2,5 k more predicted genes (≥ 3kb)), and the number of missed BUSCOs was lower than in the k41 assembly (15% fewer missing BUSCOs). However, the assembly spectra-cn plots indicated that the k71 assembly had more duplicated and triplicated assembly content, as visualized by the broader purple and green peaks for this assembly.

The assembly spectra-cn plots indicated that the AbySS 2.0 assemblies might contain a number of errors (Figure 39). At k41, a larger amount of errors was present in the assembly, as can be seen from the larger red portion in the first peak of the plot. Both assemblies had a black peak around the 100x heterozygous peak, which indicated missing heterozygous content from the assemblies. The k41 assembly also contained more missing content in the homozygous region of the plot (thicker black line at the homozygous peak). The purple and green peaks, which were highest at the homozygous peak in both plots, and continued throughout the high coverage areas, indicated duplicated and triplicated assembly content, respectively. This content was present in both assemblies, but, as can be seen from the broader purple and green peaks of the k71 assembly, this assembly contained more triplicated and duplicated content.

In comparison to Platanus and ABySS 2.0, SOAPdenovo2 performed the worst out of all assemblers. It produced the second shortest total assembly length, substantially lower BUSCO matches, and lower number of predicted genes. K-mer size substantially affected contig and scaffold statistics, as well as BUSCO content and gene predictions. The assembly that performed the worst of all assemblies and k-values (Soapdenovo-k41), only contained 5% complete BUSCOs, and had 88% of BUSCOs missing from the assembly. The k41 assembly also contained by far the least number of predicted genes (≥ 3kb) among all assemblies and k-values (only 99). Soapdenovo-41 was also minimally affected by scaffolding. Although Soapdenovo-71 generated higher values across all length statistics, had more than twice the amount of complete BUSCOs than that of Soapdenovo-41, and 15 times more predicted genes

(≥ 3kb), it still underperformed when compared to the other assemblies. Soapdenovo-41 produced a total assembly length of 784 Mbp, assembled from 116 k contigs into 258 k scaffolds, of which only 3,6% were larger than 10 kbp. The reason for the higher number of scaffolds than contigs in the SOAPdenovo2 assemblies, is because there was a large number of contigs that were shorter than 1,000 bp, and were not incorporated in the contig statistics.

The assembly spectra-cn plots indicated that the SOAPdenovo2 assemblies contained a small amount of errors (Figure 39). Both assemblies had a black peak around the 100x heterozygous peak, which indicated missing heterozygous content from the assemblies. At k41, the peak was slightly higher and broader than at k71, which implied that the k41 assembly has more missing heterozygous content than the k71 assembly. The k41 assembly also had a black peak at the homozygous content, which continued through the tail of the graph. This indicates that there is homozygous and repeat content missing from the assembly. Purple and green peaks, which indicate duplicated and triplicated assembly contents were higher in the k71 assembly. The highest amount of duplications and triplications was observed around the 200x homozygous peak; These duplications and triplications were also present throughout the higher coverage regions of the tail of the graph.

Chapter 4: Discussion

Plant genome sequences are useful in a wide range of applications, providing invaluable data resources for various research areas. These include investigation of secondary metabolites, the establishment of breeding programs, and conservation biology. Next-generation sequencing enables the sequencing of whole genomes in a comparatively time and cost-effective way. In this study, a total of 2,6 billion read pairs with an average length of ~119 bp were sequenced, resulting in a 276X genome coverage of the 1,1 Gb rooibos genome. Reads were processed, and error corrected to be used in the estimation of genome characteristics, and *de novo* genome assembly.

4.1. Quality pre-processing

Although there is some debate whether preprocessing of sequencing reads is essential, most studies do preprocess raw Illumina sequencing reads before genome assembly. Yang et al. (2019) showed that quality processing significantly reduced the computational time necessary for genome assembly without affecting the number of predicted genes. Moreover, assembly of raw reads was found to result in less accurate and less complete genome assemblies - although the unprocessed datasets had longer scaffolds, these were more often associated with misassemblies. Therefore, in this study, two methods for quality preprocessing were tested on the paired-end Illumina sequencing data. Method 1 was applied based on the "per base sequence content" plots produced by FastQC, which indicated module failure due to substantial differences between the A and T and/or G and C contents at the 5' and 3' ends of the sequences. This method significantly reduced read length and read numbers, and therefore genome coverage. Considering the arguments by Andrews (2016), who showed that hard trimming of 5’ and 3’ read end bases usually does not have a significant effect on downstream analysis, this method was not analysed further. Method 2, which focused on trimming the reads based on the quality scores of the bases, showed satisfactory retention of read pairs and higher read lengths. This method also included error correction, which has been shown to improve assembly quality and reduce computational time. Error correction reduces the number of distinct k-mers in the dataset, which in turn reduces the complexity of assembly graphs, speeding up the assembly process and lowering the RAM requirements. For error correction, choice of the program was based on the ability of Lighter to handle heterozygous datasets from polyploid organisms and technical advantage, specifically acceptance of multiple compressed input files. Subsequent assembly tests showed that for our datasets read preprocessing was essential, as assembly of the complete set of raw reads consistently failed due to RAM and time limitations.

Two methods of data size reduction not tested in the course of this study are filtering by tile and deduplication of reads. The per tile quality heatmaps generated by FastQC all failed for the HiSeq paired-end dataset. The heat map patterns (a broad loss of quality over six areas of the flow cell) were similar for all six lanes, both the forward and reverse reads. Quality processing with Trimmomatic did not have a significant effect on the heatmap. According to https://sequencing.qcfail.com/articles/position-specific-failures-of-flowcells/, this pattern suggests a very biased sequence run. The data is still usable, but considering the high genome coverage obtained in this study, its appears reasonable to discard all reads from the low quality flow cell positions, using for example the program FilterByTile, a member of the BBMap package (Bushnell, 2015). Deduplication of reads is suggested by many as a necessary preprocessing step of NGS data (Ebbert et al., 2016). It is believed to reduce computational resources and decrease potential biases on variant calling algorithms (Ebbert et al., 2016). However, Ebbert et al. (2016) shows that in deep sequenced, whole-genome data, deduplication has minimal effect on the accuracy of variant datasets. Because of the high coverage of the paired-end data in this study, deduplication of the reads should be considered in subsequent assemblies to decrease computational time and resources, and to compare results with the unduplicated assemblies. Many programs are available to perform deduplication of reads such as FastUniq (Xu et al., 2012).

To improve continuity of the genome assembly, two mate pair libraries had been sequenced. However, the sequencing results indicated that both libraries had very high duplication rates (on average 73%) and a high proportion of low quality reads. Similar problems have been reported in previous studies ("Illumina Mate Pair Read Duplication Level," 2013). In addition, trimming of the Nextera adapter sequence by the sequencing service provider significantly hampered mate pair identification. As a result, less than one-third of the nearly 300 million sequenced mate pairs were retained. Still, the MP3 data, amounting to 2x genome coverage, substantially contributed to scaffolding during genome assembly (discussed below). The MP8 library, initially excluded from genome assembly tests, has since been successfully included in subsequent analyses, and will be used in future assemblies. However, considering the price of sequencing, the high amount of duplicate reads, and the low coverage, long-read sequencing using PacBio or Oxford Nanopore sequencing technologies appear preferable.

4.2. Rooibos genome characteristics

GC content is an important feature describing genome organization. It is associated with genome size, genome organization and chromosomal structure (Šmarda et al., 2014). The GC for the rooibos sequencing data peaked at 36%, which is comparable to the average GC contents of other dicotyledonous plants (Singh et al., 2016). Interestingly, all datasets (including the two mate-pair libraries) showed a very small, but well-defined peak at 66% GC. It does not appear to represent chloroplast DNA (which has a mean GC content of 34-36%; Weihong et al., 2017), mitochondrial DNA (which has a GC content around 42-45%; Feng et al., 2019), or PhiX (which has a mean GC content of 45%; Minoche et al., 2011). Alternatively, this peak may be associated with sequencing bias, GC rich regions in the rooibos genome or contaminating plant symbiont DNA. Future analyses may include screening these reads (or assemblies of these reads) against specific organisms, using for example FastQ Screen ("Babraham Bioinformatics - FastQ Screen," n.d.) or the reformat tool from the BBMap package.

Other important genome characteristics, such as genome size, heterozygosity and repeat content were estimated based on k-mer histogram analyses using a number of programs (GenomeScope, FindGSE, BBNorm and KAT). Considering parallel studies, the combined results indicate that the rooibos genome is approximately 1,1 Gb large, very heterozygous and rich in repeats.

Across all biocomputational programs, the average predicted genome size was 1,03 ± 0,05 Gb (when using the complete k-mer histogram with a maximum k-mer coverage of 900,000x). This value is very close to the genome size predicted in a parallel study where flow cytometry analyses of rooibos radicles indicated a genome size of 1,24 ± 0,01 Gb (Mgwatyu et al., 2020). Discrepancies between flow cytometry and biocomputationally derived genome size estimates have been reported before, even for the model plant *Arabidopsis thaliana* (Sun et al., 2018). The authors argued that such differences are likely associated with the interference of chemical compounds in the stoichiometric DNA content measurements in flow cytometry analyses. Considering that in the rooibos study equipment related restrictions in the choice of the fluorescent stain (DAPI) may have also led to higher flow cytometry genome size estimates, the k-mer based rooibos genome size estimate is considered to be closer to the true value. With that, the rooibos genome size is very similar to the one of *Lupinus angustifolius*, a close relative of *A. linearis,* which has a predicted genome size of 0,924 Gb – 1,15 Gb (Hane et al., 2017;

Kasprzak et al., 2006; Yang et al., 2013). However, in the current analysis chloroplast and mitochondrial DNA reads had not been removed, which may have impacted rooibos genome size estimations. With regards to the choice of program and parameter settings, the predominant effect on genome size prediction was found to be the k-mer coverage threshold (1k vs 900k, GenomeScope, parameter CovMax). At a CovMax threshold of 1k (default in GenomeScope v1), highly covered k-mers that mostly represented repeats were excluded from the analyses, which reduced genome size estimates by half. This may explain why, in studies with vanilla (Hu et al., 2019), cane toad (Edwards et al., 2018), and pacific oyster (Hedgecock et al., 2005; Vurture et al., 2017b), k-mer based GenomeScope estimates of genome sizes were only half of those obtained by flow cytometry analyses and considerably smaller than those obtained after genome assembly. Other parameters (choice of program, k-mer size, data preprocessing, dataset size) had only small effects on the genome size estimates. In fact, when using the same histogram file, the results from the commonly used simple formula was very close to those obtained with all tested programs (including GenomeScope, FindGSE, BBNorm and KAT).

The sequenced rooibos genome is very heterozygous, as clearly visualized by the k-mer spectra graphs that depict k-mer frequency distributions. In all graphs, the heterozygous peak was notably higher than the homozygous peak. Determining the actual value, however, was complicated. The program KAT produced k-mer spectra graphs that were identical to the other programs, but predicted heterozygosity rates of 0,00% to 0,05%. For the other programs, the values ranged between 1,15% and 2,89%, depending not only on the choice of the program, but also on the k-value. For GenomeScope and FindGSE, higher k-values reduced the predicted heterozygosity rate, while the opposite was true for BBNorm. In fact, at k47, BBNorm reported a ploidy level of 1 and a heterozygosity rate of 0. This can be explained by investigating the effect of the k-value on the k-mer frequency distribution: higher k-values increased the height of the heterozygous peak, simultaneously reducing the height of the homozygous peak and shifting the two peaks together (Figure 37). At k47, the heterozygous and homozygous peaks overlapped to the degree that the valley between them was virtually non-existent. The value associated with this valley is of apparent importance when calculating heterozygosity rates and may have failed the BBNorm threshold that distinguishes between homozygous and heterozygous genomes. These results imply that current predictions of heterozygosity rate values that are based solely on k-mer frequency distributions must be considered estimates with substantial uncertainties.

For repeat content, results again largely depended on the choice of the program, but also on the k-mer coverage threshold (1k vs 900k; not surprising, as repeat content is associated with high k-mer coverage) and the k-value. The average repeat content estimated over all programs for the 900k histogram file was 56,04 ± 8,51%. This value is close to the estimated 50% repeat content of *Lupinus angustifolius* (Yang et al., 2013). The estimated repeat content of rooibos may be somewhat lower than the value estimated in this study, since reads encoding chloroplast and mitochondrial DNA had not been filtered. Removing these sequences from the datasets is expected to improve estimation accuracy. The semi-log graphs produced by GenomeScope visualized small peaks in the high-frequency regions of the k-mer graphs. These could represent high-frequency repeats in the rooibos genome, but also organelle sequences, "noise" and/or contamination. Identification of sequences that gave rise to these k-mers and subsequent blast analyses may provide insight into their origins (Vurture et al., 2017b).

4.3. Genome assembly – technical aspects

Initially, six assembly programs were chosen for testing based on recently published plant genome studies and genome assembler performance comparisons. However, getting these programs to work on the computational clusters (both, at SANBI and CHPC) with the rooibos datasets proved to be challenging. For a beginner in Linux and next-generation sequencing analyses, two of the most crucial aspects are the documentation of the software and the support provided for each program. In my experience, and at the time of writing this thesis, the programs that had the most useful documentation were ABySS 2.0 and SOAPdenovo2. In contrast, support for the programs ALLPATHS-LG, MaSuRCA, IDBA and Platanus was less helpful to lacking. Computational resources, such as memory and number of cores, were major limiting factors during this study. Despite substantial resource availability at CHPC (up to 1 TB memory and 56 cores), MaSuRCA never completed any assemblies; and attempts to assemble the complete raw data using ABySS 2.0, SOAPdenovo2 or Platanus consistently failed as the programs ran out of memory. Runtime was another limiting factor for choosing an assembly program – MaSuRCA ran for 357 hours (14 days) on a subset of data without showing any signs of progress. In addition, the amount of data generated during an assembly can become very large when taking into account test runs of different programs and unsuccessful runs. Space is, therefore, another essential factor that has to be considered when conducting genome analyses.

A very helpful feature in assembly programs is the ability to restart from the point of termination. Runs can be interrupted due to various reasons: lack of memory, time restrictions (at CHPC, if a job uses more than 48 hours, time extension needs to be requested), power outages and other problems that can cause hardware failure. Therefore, the possibility to restart a run from the point of termination saves time and computational resources. This feature is available in ABySS 2.0, ALLPATHS-LG, MaSuRCA, and Platanus. Finally, input file format restrictions have to be considered. IDBA only accepts one single unzipped FASTA-formatted input file, which is cumbersome when working with several large fastq files as provided by the sequencing service provider.

CPU efficiency (in %) is often used as a parameter when comparing performance of genome assembly programs. The value is reported by the Portable Batch System (PBS – the software that performs job scheduling on a cluster) whenever a job is completed. This % CPU efficiency is the fraction of the available CPU time that was used for a job. CPU time (or core hours) is calculated by multiplying the requested number of cores with the requested amount of time. If 56-cores were requested for 41 hours, 2,296 core hours would be available. A 36% efficiency would imply that the actual computations only used 827 core hours. It is, however, more complicated than that, as CPU efficiency is affected by several factors that are not related to the program at hand. These factors include: 1) program-related processes which run outside of the PBS; 2) processes which run outside of PBS and persist after the PBS job has completed; 3) MPI implementations that use idle time, and 4) the shared storage system.

Processes which run outside of PBS will not be accounted for in the calculation of the CPU efficiency. Processes which run outside of PBS and persist after the job has completed can interfere with subsequent jobs, and result in an inaccurate CPU efficiency reported. So, a subsequent task may appear to be using 100% of a CPU core while not carrying out the presumed task. This happens because some MPI implementations use "idle" time to poll other MPI processes (which ensures that inter-process communication latency is as low as possible). Lastly, because a shared storage is used, the speed is variable and depends on how full the file system is, and how congested the Infiniband network is. Because the processes perform a lot of input-output (I/O), the shared storage system substantially affects efficiency and performance.

4.4. Genome assembly – assembler comparisons

Only three assembly programs (ABySS 2.0, Platanus, and SOAPdenovo2) successfully completed analysis on the entire quality-processed dataset within a reasonable amount of time. Assembly quality was compared using QUAST-LG and KAT. As discussed in previous studies (Salzberg et al., 2012; Yang et al., 2019), length statistics such as the number of contigs/scaffolds, N50 and N75 do not necessarily imply better assembly quality. It is essential to take other parameters into account, such as the number of unknown bases (Ns), BUSCO statistics, gene prediction results and spectra-cn plot results. Soapdenovo-71 is an example of an assembly with good length statistics but poor BUSCO and spectra-cn results. Nonetheless, in this study, higher values for N50, N75, largest contig/scaffold and total contig/scaffold length correlated positively with the proportion of complete BUSCOs and the numbers of predicted genes. The gene numbers predicted in this study by QUAST-LG for the rooibos genome were very high (over 200,000 for the Platanus assemblies). Related plant genomes encode just about one-quarter of this number (*Lupinus angustifolius*: 57,807 [Yang et al., 2013]; *Medicago truncatula*: 50,894 [Tang et al., 2014]). However, it must be kept in mind that essential data processing steps, including separation of chloroplast and mitochondrial sequences as well as repeat masking had not been performed prior to the gene prediction step. Moreover, GlimmerHMM had not been trained to specifically identify rooibos genes, as the only available training files available were from *Arabidopsis thaliana*. The predicted gene numbers must, therefore not be mistaken for the true values. The k-mer spectra-cn plots were found to be a useful tool to assess assembly quality. These plots visualized that in this study higher k-value were associated with increased duplicated and triplicated assembly contents, and that they may result in higher error rates (as observed for the Platanus and SOAPdenovo2 assemblies, although the opposite was true for the AbySS assemblies). These plots can therefore be used to determine an appropriate k-value for a specific dataset and assembly program.

Of the three assemblers, Platanus performed best: it produced the longest and most contiguous assemblies yielding substantially higher BUSCO matches and numbers of predicted genes, and produced the best spectra-cn plots with the least amount of duplicated and triplicated assembly content. Platanus was specifically developed to handle heterozygous genomes, and the poor performance of ABySS and SOAPdenovo2 may well be associated with the high heterozygosity and repeat contents predicted for the rooibos genome. Platanus has recently been used to assemble the heterozygous plant genomes of *Boehmeria nivea* (Luan et al., 2018) and *Calotropis gigantea* (Hoopes et al., 2018). Both studies used only Illumina sequencing data, achieving a total genome coverage of 256X and 193X, respectively. Although these genome coverages were comparable to the one obtained for the rooibos genome in this study (269X), their N50 values and BUSCO statistics were better.

4.5. Final note

Previous projects investigating large genomes indicated that good assembly results were obtained at 100X genome coverage (Dominguez Del Angel et al., 2018; Ekblom and Wolf, 2014; Schatz et al., 2012). Higher genome coverage, as obtained in this study, could, in fact, hamper genome assembly. During assembly, excessive coverage in a particular genome location may be seen as a sequencing error, and true sequencing errors can propagate and start to look like correct sequences (Dominguez Del Angel et al., 2018; Ekblom and Wolf, 2014). The authors suggest the use of normalization tools (e.g. BBNorm) for correction of high coverage areas. Considering the amount of data generated in this project, filtering by tile and deduplication should also be considered. However, latest technological developments open doors to novel approaches in plant genome assembly. Recent plant genome studies increasingly focus on a hybrid assembly approach combining short-read Illumina sequencing data with long-read data obtained using PacBio or Oxford Nanopore sequencing technologies (Li and Harkess, 2018). New tools are essential, that cater to this type of data analyses.

Chapter 5: Conclusion

This thesis focused on the establishment of methodologies for plant genome analysis, including estimation of genome characteristics and evaluation of methods essential for plant genome assembly using Illumina sequencing data. K-mer analysis was found to be a suitable approach to estimate the genome size. For rooibos, the computationally predicted value of 1,03 ± 0,05 Gb was affected little by choice of program and parameters (as long as the complete k-mer histogram, here with a coverage of up to 900,000x, was used), and was close to the rooibos genome size predicted using flow cytometry (1,24 ± 0,01 Gb). GenomeScope's approach to limiting the k-mer coverage threshold is not suitable for organisms that are repeat-rich and should, therefore, be used with caution when working with plants. It appears that the formula for genome size estimation may also suffice. K-mer analysis also indicated high heterozygosity (2,09 ± 0,33) and repeat content (56,04 ± 8,51%) for the rooibos genome. The actual values varied substantially depending on the choice of program and parameter settings. They should therefore be used as guidelines, rather than true values.

For genome assembly using Illumina sequencing data, the following programs were installed, evaluated and found suitable: FastQC, MultiQC and KAT for quality assessment, Trimmomatic, FLASH, and Nextclip for quality processing, Platanus for data assembly and QUAST-LG and KAT for evaluation of assembly quality. The first assembly of the rooibos genome, although reasonable, is still highly fractionated. Future work should focus on the removal of chloroplast and mitochondrial DNA. Considering the amount of data (genome coverage ~276X) additional quality filtering steps, such as filtering by tile and deduplication of the Illumina reads, may not only improve assembly quality but substantially speed up assembly and permit assembly using different programs (e.g. MaSuRCa or the ploidy conscious assemblers Ranbow and SDhaP). Considering the high repeat content, long-read data for the rooibos genome is highly desirable.

References

454 (Roche) [WWW Document], n.d. . AllSeq. URL https://allseq.com/knowledge-bank/ngs-necropolis/454-roche/ (accessed 7.17.20).

Akogwu, I., Wang, N., Zhang, C., Gong, P., 2016. A comparative study of k-spectrum-based error correction methods for next-generation sequencing data analysis. Hum. Genomics 10, 20. https://doi.org/10.1186/s40246-016-0068-0

Ambardar, S., Gupta, R., Trakroo, D., Lal, R., Vakhlu, J., 2016. High Throughput Sequencing: An Overview of Sequencing Chemistry. Indian J. Microbiol. 56, 394–404. https://doi.org/10.1007/s12088-016-0606-4

An Intuitive Explanation for Running Velvet with Varying K-mer Sizes, 2012. . Homolog.us. URL https://homolog.us/blogs/genome/2012/06/17/an-intuitive-explanation-for-running-de-bruijn-assembler-with-varying-k-mer-sizes/ (accessed 7.12.20).

Andrews, S., 2016. QC Fail Sequencing » Position specific failures of flowcells. QC Fail. URL https://sequencing.qcfail.com/articles/position-specific-failures-of-flowcells/ (accessed 7.12.20).

Andrews, S., 2010. FastQC: a quality control tool for high throughput sequence data. Babraham Bioinformatics, Babraham Institute, Cambridge, United Kingdom.

Basantani, M.K., Gupta, D., Mehrotra, R., Mehrotra, S., Vaish, S., Singh, A., 2017. An update on bioinformatics resources for plant genomics research. Curr. Plant Biol. 11–12, 33–40. https://doi.org/10.1016/j.cpb.2017.12.002

Bolger, A., Lohse, M., Usadel, B., 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30, 2114–2120. https://doi.org/10.1093/bioinformatics/btu170

Bolger, M.E., Weisshaar, B., Scholz, U., Stein, N., Usadel, B., Mayer, K.F., 2014. Plant genome sequencing — applications for crop improvement. Curr. Opin. Biotechnol. 26, 31–37. https://doi.org/10.1016/j.copbio.2013.08.019

Bradnam, K.R., Fass, J.N., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., Boisvert, S., Chapman, J.A., Chapuis, G., Chikhi, R., Chitsaz, H., Chou, W.-C., Corbeil, J., Del Fabbro, C., Docking, T.R., Durbin, R., Earl, D., Emrich, S., Fedotov, P., Fonseca, N.A., Ganapathy, G., Gibbs, R.A., Gnerre, S., Godzaridis, É., Goldstein, S., Haimel, M., Hall, G., Haussler, D., Hiatt, J.B., Ho, I.Y., Howard, J., Hunt, M., Jackman, S.D., Jaffe, D.B., Jarvis, E.D., Jiang, H., Kazakov, S., Kersey, P.J., Kitzman, J.O., Knight, J.R., Koren, S., Lam, T.-W., Lavenier, D., Laviolette, F., Li, Y., Li, Z., Liu, B., Liu, Y., Luo, R., MacCallum, I., MacManes, M.D., Maillet, N., Melnikov, S., Naquin, D., Ning, Z., Otto, T.D., Paten, B., Paulo, O.S., Phillippy, A.M., Pina-Martins, F., Place, M., Przybylski, D., Qin, X., Qu, C., Ribeiro, F.J., Richards, S., Rokhsar, D.S., Ruby, J.G., Scalabrin, S., Schatz, M.C., Schwartz, D.C., Sergushichev, A., Sharpe, T., Shaw, T.I., Shendure, J., Shi, Y., Simpson, J.T., Song, H., Tsarev, F., Vezzi, F., Vicedomini, R., Vieira, B.M., Wang, J., Worley, K.C., Yin, S., Yiu, S.-M., Yuan, J., Zhang, G., Zhang, H., Zhou, S., Korf, I.F., 2013. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. GigaScience 2. https://doi.org/10.1186/2047-217X-2-10

Bronner, I.F., Quail, M.A., Turner, D.J., Swerdlow, H., 2009. Improved Protocols for Illumina Sequencing. Curr. Protoc. Hum. Genet. Editor. Board Jonathan Haines Al 0 18. https://doi.org/10.1002/0471142905.hg1802s62

Bushnell, B., 2018. BBTools: a suite of fast, multithreaded bioinformatics tools designed for analysis of DNA and RNA sequence data. Joint Genome Institute.

Bushnell, B., 2015. BBMap: A Fast, Accurate, Splice-Aware Aligner.

Chin, C.-S., Peluso, P., Sedlazeck, F.J., Nattestad, M., Concepcion, G.T., Clum, A., Dunn, C., O'Malley, R., Figueroa-Balderas, R., Morales-Cruz, A., Cramer, G.R., Delledonne, M., Luo, C., Ecker, J.R., Cantu, D., Rank, D.R., Schatz, M.C., 2016. Phased diploid genome assembly with single-molecule real-time sequencing. Nat. Methods 13, 1050–1054. https://doi.org/10.1038/nmeth.4035

Clavijo, B.J., Garcia Accinelli, G., Wright, J., Heavens, D., Barr, K., Yanes, L., Di-Palma, F., 2017. W2RAP: a pipeline for high quality, robust assemblies of large complex genomes from short read data (preprint). Bioinformatics. https://doi.org/10.1101/110999

Conservation Genomics [WWW Document], n.d. . US Geol. Surv. URL https://www.usgs.gov/centers/fort/science/conservation-genomics?qt-science_center_objects=0#qt-science_center_objects (accessed 5.27.20).

Das, S., Vikalo, H., 2015. SDhaP: haplotype assembly for diploids and polyploids via semi-definite programming. BMC Genomics 16, 260. https://doi.org/10.1186/s12864-015-1408-5

De Roeck, A., De Coster, W., Bossaerts, L., Cacace, R., De Pooter, T., Van Dongen, J., D'Hert, S., De Rijk, P., Strazisar, M., Van Broeckhoven, C., Sleegers, K., 2019. NanoSatellite: accurate characterization of expanded tandem repeat length and sequence through whole genome long-read sequencing on PromethION. Genome Biol. 20, 239. https://doi.org/10.1186/s13059-019-1856-3

DeSalle, R., Gregory, T.R., Johnston, J.S., 2005. Preparation of Samples for Comparative Studies of Arthropod Chromosomes: Visualization, In Situ Hybridization, and Genome Size Estimation. Methods Enzymol. 395, 460–488. https://doi.org/10.1016/S0076-6879(05)95025-8

Dominguez Del Angel, V., Hjerde, E., Sterck, L., Capella-Gutierrez, S., Notredame, C., Vinnere Pettersson, O., Amselem, J., Bouri, L., Bocs, S., Klopp, C., Gibrat, J.-F., Vlasova, A., Leskosek, B.L., Soler, L., Binzer-Panchal, M., Lantz, H., 2018. Ten steps to get started in Genome Assembly and Annotation. F1000Research 7, 148. https://doi.org/10.12688/f1000research.13598.1

Ebbert, M.T.W., Wadsworth, M.E., Staley, L.A., Hoyt, K.L., Pickett, B., Miller, J., Duce, J., Kauwe, J.S.K., Ridge, P.G., 2016. Evaluating the necessity of PCR duplicate removal from next-generation sequencing data and a comparison of approaches. BMC Bioinformatics 17. https://doi.org/10.1186/s12859-016-1097-3

Edwards, R.J., Tuipulotu, D.E., Amos, T.G., O'Meally, D., Richardson, M.F., Russell, T.L., Vallinoto, M., Carneiro, M., Ferrand, N., Wilkins, M.R., Sequeira, F., Rollins, L.A., Holmes, E.C., Shine, R., White, P.A., 2018. Draft genome assembly of the invasive cane toad, Rhinella marina. GigaScience 7. https://doi.org/10.1093/gigascience/giy095

Ekblom, R., Wolf, J.B.W., 2014. A field guide to whole-genome sequencing, assembly and annotation. Evol. Appl. 7, 1026–1042. https://doi.org/10.1111/eva.12178

Ewels, P., Magnusson, M., Lundin, S., Käller, M., 2016. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics 32, 3047–3048. https://doi.org/10.1093/bioinformatics/btw354

Feng, L., Li, N., Yang, W., Li, Y., Wang, C.-M., Tong, S.-W., He, J.-X., 2019. Analyses of mitochondrial genomes of the genus Ammopiptanthus provide new insights into the evolution of legume plants. Plant Syst. Evol. 305, 385–399. https://doi.org/10.1007/s00606-019-01578-2

File Format [WWW Document], n.d. . Illumina Support. URL https://support.illumina.com/help/BaseSpace_OLH_009008/Content/Source/Informatics/BS/FileFormat_FASTQ-files_swBS.htm (accessed 7.12.20).

Fridovich-Keil, J.L., 2019. Human genome [WWW Document]. Encycl. Br. URL https://www.britannica.com/science/human-genome (accessed 5.25.20).

Gnerre, S., MacCallum, I., Przybylski, D., Ribeiro, F.J., Burton, J.N., Walker, B.J., Sharpe, T., Hall, G., Shea, T.P., Sykes, S., Berlin, A.M., Aird, D., Costello, M., Daza, R., Williams, L., Nicol, R., Gnirke, A., Nusbaum, C., Lander, E.S., Jaffe, D.B., 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. Proc. Natl. Acad. Sci. 108, 1513–1518. https://doi.org/10.1073/pnas.1017351108

Göpfrich, K., Judge, K., 2018. Decoding DNA with a pocket-sized sequencer. Sci. Sch. 4.

Hane, J.K., Ming, Y., Kamphuis, L.G., Nelson, M.N., Garg, G., Atkins, C.A., Bayer, P.E., Bravo, A., Bringans, S., Cannon, S., Edwards, D., Foley, R., Gao, L., Harrison, M.J., Huang, W., Hurgobin, B., Li, S., Liu, C.-W., McGrath, A., Morahan, G., Murray, J., Weller, J., Jian, J., Singh, K.B., 2017. A comprehensive draft genome sequence for lupin (Lupinus angustifolius), an emerging health food: insights into plant–microbe interactions and legume evolution. Plant Biotechnol. J. 15, 318–330. https://doi.org/10.1111/pbi.12615

Heather, J.M., Chain, B., 2016. The sequence of sequencers: The history of sequencing DNA. Genomics 107, 1–8. https://doi.org/10.1016/j.ygeno.2015.11.003

Hedgecock, D., Gaffney, P.M., Goulletquer, P., Guo, X., Reece, K., Warr, G.W., 2005. The case for sequencing the Pacific oyster genome. J. Shellfish Res. 24, 429–442.

Heo, Y., Wu, X.-L., Chen, D., Ma, J., Hwu, W.-M., 2014. BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads. Bioinformatics 30, 1354–1362. https://doi.org/10.1093/bioinformatics/btu030

Hoopes, G.M., Hamilton, J.P., Kim, J., Zhao, D., Wiegert-Rininger, K., Crisovan, E., Buell, C.R., 2018. Genome Assembly and Annotation of the Medicinal Plant *Calotropis gigantea* , a Producer of Anticancer and Antimalarial Cardenolides. G3amp58 GenesGenomesGenetics 8, 385–391. https://doi.org/10.1534/g3.117.300331

How much PhiX spike-in is recommended when sequencing low diversity libraries on Illumina platforms? [WWW Document], 2020. . Illumina Support. URL https://support.illumina.com/bulletins/2017/02/how-much-phix-spike-in-is-recommended-when-sequencing-low-divers.html (accessed 7.10.20).

Hu, Y., Resende, M.F.R., Bombarely, A., Brym, M., Bassil, E., Chambers, A.H., 2019. Genomics-based diversity analysis of Vanilla species using a Vanilla planifolia draft genome and Genotyping-By-Sequencing. Sci. Rep. 9. https://doi.org/10.1038/s41598-019-40144-1

Hussain, Md.S., Rahman, Md.A., Fareed, S., Ansari, S., Ahmad, I., Mohd. Saeed, 2012. Current approaches toward production of secondary plant metabolites. J. Pharm. Bioallied Sci. 4, 10. https://doi.org/10.4103/0975-7406.92725

Illumina Mate Pair Read Duplication Level [WWW Document], 2013. Biostars. URL https://www.biostars.org/p/74029/ (accessed 7.19.20).

Illumina Scientific Affairs, 2016. For all you seq... [Online poster]. [7/16/2020]. Available from: https://www.illumina.com/content/dam/illumina-marketing/documents/applications/ngs-library-prep/ForAllYouSeqMethods.pdf

Illumina sequencing platforms [WWW Document], 2020. Illumina. URL https://www.illumina.com/systems/sequencing-platforms.html (accessed 7.10.20).

Illumina Sequencing Technology [WWW Document], 2010. URL https://www.illumina.com/documents/products/techspotlights/techspotlight_sequencing.pdf (accessed 7.16.2020).

Indexed Sequencing Overview Guide (15057455) [WWW Document], 2020. URL https://emea.support.illumina.com/content/dam/illumina-support/documents/documentation/system_documentation/miseq/indexed-sequencing-overview-guide-15057455-06.pdf (accessed 7.10.2020).

Industry Statistics | South African Rooibos Council [WWW Document], n.d. . Rooibos Counc. - South Afr. URL https://sarooibos.co.za/industry-statistics/ (accessed 6.3.20).

International Human Genome Sequencing Consortium, 2001. Initial sequencing and analysis of the human genome. Nature 409, 860–921. https://doi.org/10.1038/35057062

Ion Torrent Next-Generation Sequencing Technology [WWW Document], n.d. . ThermoFisher Sci. URL https://www.thermofisher.com/za/en/home/life-science/sequencing/next-generation-sequencing/ion-torrent-next-generation-sequencing-technology.html (accessed 7.10.20).

Jackman, S.D., Vandervalk, B.P., Mohamadi, H., Chu, J., Yeo, S., Hammond, S.A., Jahesh, G., Khan, H., Coombe, L., Warren, R.L., Birol, I., 2017. ABySS 2.0: resource-efficient assembly of large genomes using a Bloom filter. Genome Res. 27, 768–777. https://doi.org/10.1101/gr.214346.116

Kajitani, R., Toshimoto, K., Noguchi, H., Toyoda, A., Ogura, Y., Okuno, M., Yabana, M., Harada, M., Nagayasu, E., Maruyama, H., Kohara, Y., Fujiyama, A., Hayashi, T., Itoh, T., 2014. Efficient de novo assembly of highly heterozygous genomes from whole-genome shotgun short reads. Genome Res. 24, 1384–1395. https://doi.org/10.1101/gr.170720.113

Kasprzak, A., Šafář, J., Janda, J., Doležel, J., Wolko, B., Naganowska, B., 2006. The bacterial artificial chromosome (BAC) library of the narrow-leafed lupin (Lupinus angustifolius L.). Cell. Mol. Biol. Lett. 11. https://doi.org/10.2478/s11658-006-0033-3

Kchouk, M., Gibrat, J.F., Elloumi, M., 2017. Generations of Sequencing Technologies: From First to Next Generation. Biol. Med. 09. https://doi.org/10.4172/0974-8369.1000395

Kelley, D.R., Schatz, M.C., Salzberg, S.L., 2010. Quake: quality-aware detection and correction of sequencing errors. Genome Biol. 11, R116. https://doi.org/10.1186/gb-2010-11-11-r116

Launen, L., 2017. Illumina Sequencing (for Dummies) -An overview on how our samples are sequenced. kscbioinformatics. URL https://kscbioinformatics.wordpress.com/2017/02/13/illumina-sequencing-for-dummies-samples-are-sequenced/ (accessed 7.16.20).

Lee, J., 2015. Overlapping paired-end reads [WWW Document]. Incodom. URL http://www.incodom.kr/Overlapping_paired-end_reads (accessed 7.12.20).

Leger, A., Leonardi, T., 2019. pycoQC, interactive quality control for Oxford Nanopore Sequencing. J. Open Source Softw. 4, 1236. https://doi.org/10.21105/joss.01236

Leggett, R.M., Clavijo, B.J., Clissold, L., Clark, M.D., Caccamo, M., 2014. NextClip: an analysis and read preparation tool for Nextera Long Mate Pair libraries. Bioinformatics 30, 566–568. https://doi.org/10.1093/bioinformatics/btt702

Li, F.-W., Harkess, A., 2018. A guide to sequence your favorite plant genomes. Appl. Plant Sci. 6, e1030. https://doi.org/10.1002/aps3.1030

Liu, B., Shi, Y., Yuan, J., Hu, X., Zhang, H., Li, N., Li, Z., Chen, Y., Mu, D., Fan, W., 2013. Estimation of genomic characteristics by analyzing k-mer frequency in de novo genome projects. ArXiv13082012 Q-Bio.

Liu, L., Li, Y., Li, S., Hu, N., He, Y., Pong, R., Lin, D., Lu, L., Law, M., 2012. Comparison of Next-Generation Sequencing Systems. J. Biomed. Biotechnol. 2012. https://doi.org/10.1155/2012/251364

Luan, M.-B., Jian, J.-B., Chen, P., Chen, Jun-Hui, Chen, Jian-Hua, Gao, Q., Gao, G., Zhou, J.-H., Chen, K.-M., Guang, X.-M., Chen, J.-K., Zhang, Q.-Q., Wang, X.-F., Fang, L., Sun, Z.-M., Bai, M.-Z., Fang, X.-D., Zhao, S.-C., Xiong, H.-P., Yu, C.-M., Zhu, A.-G., 2018. Draft genome sequence of ramie, Boehmeria nivea (L.) Gaudich. Mol. Ecol. Resour. 18, 639–645. https://doi.org/10.1111/1755-0998.12766

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., He, G., Chen, Y., Pan, Q., Liu, Yunjie, Tang, J., Wu, G., Zhang, H., Shi, Y., Liu, Yong, Yu, C., Wang, B., Lu, Y., Han, C., Cheung, D.W., Yiu, S.-M., Peng, S., Xiaoqian, Z., Liu, G., Liao, X., Li, Y., Yang, H., Wang, Jian, Lam, T.-W., Wang, Jun, 2012. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. GigaScience 1, 18. https://doi.org/10.1186/2047-217X-1-18

Magoc, T., Salzberg, S.L., 2011. FLASH: fast length adjustment of short reads to improve genome assemblies. Bioinformatics 27, 2957–2963. https://doi.org/10.1093/bioinformatics/btr507

Majoros, W.H., Pertea, M., Salzberg, S.L., 2004. TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. Bioinformatics 20, 2878–2879. https://doi.org/10.1093/bioinformatics/bth315

Mapleson, D., Garcia Accinelli, G., Kettleborough, G., Wright, J., Clavijo, B.J., 2016. KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. Bioinformatics btw663. https://doi.org/10.1093/bioinformatics/btw663

Marçais, G., Kingsford, C., 2011. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics 27, 764–770. https://doi.org/10.1093/bioinformatics/btr011

Mgwatyu, Y., 2019. Investigations into genome size and genetic diversity of distinct rooibos growth forms. University of the Western Cape, South African National Bioinformatics Institute.

Mgwatyu, Y., Stander, A.A., Ferreira, S., Williams, W., Hesse, U., 2020. Rooibos (Aspalathus linearis) Genome Size Estimation Using Flow Cytometry and K-Mer Analyses. Plants 9, 270. https://doi.org/10.3390/plants9020270

Michael, T.P., 2014. Plant genome size variation: bloating and purging DNA. Brief. Funct. Genomics 13, 308–317. https://doi.org/10.1093/bfgp/elu005

Mikheenko, A., Prjibelski, A., Saveliev, V., Antipov, D., Gurevich, A., 2018. Versatile genome assembly evaluation with QUAST-LG. Bioinformatics 34, i142–i150. https://doi.org/10.1093/bioinformatics/bty266

Millar, D.A., Bowles, S., Windvogel, S.L., Louw, J., Muller, C.J.F., 2020. Effect of Rooibos (*Aspalathus linearis*) extract on atorvastatin-induced toxicity in C3A liver cells. J. Cell. Physiol. jcp.29756. https://doi.org/10.1002/jcp.29756

Miller, J.R., Delcher, A.L., Koren, S., Venter, E., Walenz, B.P., Brownley, A., Johnson, J., Li, K., Mobarry, C., Sutton, G., 2008. Aggressive assembly of pyrosequencing reads with mates. Bioinformatics 24, 2818–2824. https://doi.org/10.1093/bioinformatics/btn548

Minoche, A.E., Dohm, J.C., Himmelbauer, H., 2011. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and Genome Analyzer systems. Genome Biol. 12, R112. https://doi.org/10.1186/gb-2011-12-11-r112

MiSeq System Guide [WWW Document], 2018. Illumina. URL https://support.illumina.com/content/dam/illumina-support/documents/documentation/system_documentation/miseq/miseq-system-guide-for-miseq-reporter-1000000061014-00.pdf (accessed 7.16.2020).

Mitchell, K., Brito, J.J., Mandric, I., Wu, Q., Knyazev, S., Chang, S., Martin, L.S., Karlsberg, A., Gerasimov, E., Littman, R., Hill, B.L., Wu, N.C., Yang, H.T., Hsieh, K., Chen, L., Littman, E., Shabani, T., Enik, G., Yao, D., Sun, R., Schroeder, J., Eskin, E., Zelikovsky, A., Skums, P., Pop, M., Mangul, S., 2020. Benchmarking of computational error-correction methods for next-generation sequencing data. Genome Biol. 21, 71. https://doi.org/10.1186/s13059-020-01988-3

Multiplexed Sequencing with the Illumina Genome Analyzer System [WWW Document], 2008. Illumina. URL https://www.illumina.com/documents/products/datasheets/datasheet_sequencing_multiplex.pdf (accessed 7.16.2020).

N50, L50, and related statistics [WWW Document], 2020. Wikipedia. URL https://en.wikipedia.org/w/index.php?title=N50,_L50,_and_related_statistics&oldid=965436346 (accessed 7.14.20).

Nakano, K., Shiroma, A., Shimoji, M., Tamotsu, H., Ashimine, N., Ohki, S., Shinzato, M., Minami, M., Nakanishi, T., Teruya, K., Satou, K., Hirano, T., 2017. Advantages of genome sequencing by long-read sequencer using SMRT technology in medical area. Hum. Cell 30, 149–161. https://doi.org/10.1007/s13577-017-0168-8

Next Generation Sequencing / Whole Genome Sequencing [WWW Document], n.d. . Biocompare. URL https://www.biocompare.com/Molecular-Biology/9187-Next-Generation-Sequencing/ (accessed 7.17.20).

NEXT GENERATION SEQUENCING [WWW Document], n.d. . 3402 Bioinforma. URL http://www.3402bioinformaticsgroup.com/service/ (accessed 7.16.20).

Nextera® Mate Pair Library Prep Reference Guide [WWW Document], 2016. Illumina. URL https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/samplepreps_nextera/nexteramatepair/nextera-mate-pair-reference-guide-15035209-02.pdf (accessed 7.12.2020).

Nextera® Mate Pair Library Preparation Kit: Datasheet [WWW Document], 2014. Illumina. URL https://www.illumina.com/content/dam/illumina-marketing/documents/products/datasheets/datasheet_nextera_mate_pair.pdf (accessed 7.10.2020).

Nextera DNA Library Prep Reference Guide [WWW Document], 2016. Illumina. URL https://support.illumina.com/content/dam/illumina-support/documents/documentation/chemistry_documentation/samplepreps_nextera/nexteradna/nextera-dna-library-prep-reference-guide-15027987-01.pdf (accessed 7.11.2020).

NGS data formats and analyses, 2016. [Online Poster]. [7.12.2020]. Available from: https://www.slideshare.net/rjorton/ngs-data-formats-and-analyses.

Patel, R.K., Jain, M., 2012. NGS QC Toolkit: A Toolkit for Quality Control of Next Generation Sequencing Data. PLoS ONE 7. https://doi.org/10.1371/journal.pone.0030619.

Peng, Y., Leung, H.C.M., Yiu, S.M., Chin, F.Y.L., 2010. IDBA – A Practical Iterative de Bruijn Graph De Novo Assembler, in: Berger, B. (Ed.), Research in Computational Molecular Biology, Lecture Notes in Computer Science. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 426–440. https://doi.org/10.1007/978-3-642-12683-3_28

Performance specifications for the HiSeq 2500 System [WWW Document], 2020. Illumina. URL https://www.illumina.com/systems/sequencing-platforms/hiseq-2500/specifications.html (accessed 7.10.20).

Quality Scores for Next-Generation Sequencing [WWW Document], 2011. Illumina. URL https://www.illumina.com/Documents/products/technotes/technote_Q-Scores.pdf (accessed 7.12.2020).

Ranallo-Benavidez, T.R., Jaron, K.S., Schatz, M.C., 2019. GenomeScope 2.0 and Smudgeplots: Reference-free profiling of polyploid genomes. bioRxiv 747568. https://doi.org/10.1101/747568

Rhoads, A., Au, K.F., 2015. PacBio Sequencing and Its Applications. Genomics Proteomics Bioinformatics 13, 278–289. https://doi.org/10.1016/j.gpb.2015.08.002

Rizk, G., Lavenier, D., Chikhi, R., 2013. DSK: k-mer counting with very low memory usage. Bioinformatics 29, 652–653. https://doi.org/10.1093/bioinformatics/btt020

Rizzi, R., Beretta, S., Patterson, M., Pirola, Y., Previtali, M., Della Vedova, G., Bonizzoni, P., 2019. Overlap graphs and de Bruijn graphs: data structures for de novo genome assembly in the big data era. Quant. Biol. 7, 278–292. https://doi.org/10.1007/s40484-019-0181-x

Ross, M.G., Russ, C., Costello, M., Hollinger, A., Lennon, N.J., Hegarty, R., Nusbaum, C., Jaffe, D.B., 2013. Characterizing and measuring bias in sequence data. Genome Biol. 14, R51. https://doi.org/10.1186/gb-2013-14-5-r51

Salzberg, S.L., Phillippy, A.M., Zimin, A., Puiu, D., Magoc, T., Koren, S., Treangen, T.J., Schatz, M.C., Delcher, A.L., Roberts, M., Marcais, G., Pop, M., Yorke, J.A., 2012. GAGE: A critical evaluation of genome assemblies and assembly algorithms. Genome Res. 22, 557–567. https://doi.org/10.1101/gr.131383.111

Sanger, F., Air, G.M., Barrell, B.G., Brown, N.L., Coulson, A.R., Fiddes, J.C., Hutchison, C.A., Slocombe, P.M., Smith, M., 1977. Nucleotide sequence of bacteriophage φX174 DNA. Nature 265, 687–695. https://doi.org/10.1038/265687a0

Schatz, M.C., Witkowski, J., McCombie, W.R., 2012. Current challenges in de novo plant genome sequencing and assembly 7.

Simão, F.A., Waterhouse, R.M., Ioannidis, P., Kriventseva, E.V., Zdobnov, E.M., 2015. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. Bioinformatics 31, 3210–3212. https://doi.org/10.1093/bioinformatics/btv351

Singh, R., Ming, R., Yu, Q., 2016. Comparative Analysis of GC Content Variations in Plant Genomes. Trop. Plant Biol. 9, 136–149. https://doi.org/10.1007/s12042-016-9165-4

Slatko, B.E., Gardner, A.F., Ausubel, F.M., 2018. Overview of Next Generation Sequencing Technologies. Curr. Protoc. Mol. Biol. 122, e59. https://doi.org/10.1002/cpmb.59

Šmarda, P., Bureš, P., Horová, L., Leitch, I.J., Mucina, L., Pacini, E., Tichý, L., Grulich, V., Rotreklová, O., 2014. Ecological and evolutionary significance of genomic GC content diversity in monocots. Proc. Natl. Acad. Sci. U. S. A. 111, E4096–E4102. https://doi.org/10.1073/pnas.1321152111

Song, L., Florea, L., Langmead, B., 2014. Lighter: fast and memory-efficient sequencing error correction without counting 13.

Sun, H., Ding, J., Piednoël, M., Schneeberger, K., 2018. findGSE: estimating genome size variation within human and Arabidopsis using k-mer frequencies. Bioinformatics 34, 550–557. https://doi.org/10.1093/bioinformatics/btx637

Tang, H., Krishnakumar, V., Bidwell, S., Rosen, B., Chan, A., Zhou, S., Gentzbittel, L., Childs, K.L., Yandell, M., Gundlach, H., Mayer, K.F., Schwartz, D.C., Town, C.D., 2014. An improved genome release (version Mt4.0) for the model legume Medicago truncatula. BMC Genomics 15, 312. https://doi.org/10.1186/1471-2164-15-312

Ter-Hovhannisyan, V., Lomsadze, A., Chernoff, Y.O., Borodovsky, M., 2008. Gene prediction in novel fungal genomes using an ab initio algorithm with unsupervised training. Genome Res. 18, 1979–1990. https://doi.org/10.1101/gr.081612.108

Truong, K., 2020. Sequencing giant Illumina scraps $1.2 billion PacBio acquisition [WWW Document]. San Franc. Bus. Times. URL https://www.bizjournals.com/sanfrancisco/news/2020/01/03/sequencing-giant-illumina-scraps-1-2-billion.html (accessed 7.10.20).

Understanding Illumina Quality Scores [WWW Document], 2014. Illumina. URL https://www.illumina.com/content/dam/illumina-marketing/documents/products/technotes/technote_understanding_quality_scores.pdf (accessed 7.12.2020)

Vurture, G.W., Sedlazeck, F.J., Nattestad, M., Underwood, C.J., Fang, H., Gurtowski, J., Schatz, M.C., 2017a. GenomeScope: Fast reference-free genome profiling from short reads- Supplementary notes and figures 22.

Vurture, G.W., Sedlazeck, F.J., Nattestad, M., Underwood, C.J., Fang, H., Gurtowski, J., Schatz, M.C., 2017b. GenomeScope: Fast reference-free genome profiling from short reads 22.

Wang, S., Alseekh, S., Fernie, A.R., Luo, J., 2019. The Structure and Function of Major Plant Metabolite Modifications. Mol. Plant 12, 899–919. https://doi.org/10.1016/j.molp.2019.06.001

Weihong, Y., Wengui, S., Lei, L., Yubao, M., Libo, C., Zhaolan, W., Xiangyang, H., 2017. Complete sequencing of the chloroplast genomes of two *Medicago* species. Mitochondrial DNA Part B 2, 302–303. https://doi.org/10.1080/23802359.2017.1325336

Westbury, M., 2018. Unraveling evolution through Next Generation Sequencing. University of Potsdam.

What is the PhiX Control v3 Library and what is its function in Illumina Next Generation Sequencing? [WWW Document], 2020. . Illumina Support. URL https://support.illumina.com/bulletins/2017/02/what-is-the-phix-control-v3-library-and-what-is-its-function-in-.html (accessed 7.10.20).

Wright, Jon, 2016. KAT Documentation [WWW Document]. KAT Kmer Anal. Toolkit. URL https://kat.readthedocs.io/en/latest/walkthrough.html (accessed 5.16.20).

Xu, H., Luo, X., Qian, J., Pang, X., Song, J., Qian, G., Chen, J., Chen, S., 2012. FastUniq: A Fast De Novo Duplicates Removal Tool for Paired Short Reads. PLoS ONE 7, e52249. https://doi.org/10.1371/journal.pone.0052249

Yang, H., Tao, Y., Zheng, Z., Zhang, Q., Zhou, G., Sweetingham, M.W., Howieson, J.G., Li, C., 2013. Draft Genome Sequence, and a Sequence-Defined Genetic Linkage Map of the Legume Crop Species Lupinus angustifolius L. PLoS ONE 8, e64799. https://doi.org/10.1371/journal.pone.0064799

Yang, J., Moeinzadeh, M.-H., Kuhl, H., Helmuth, J., Xiao, P., Haas, S., Liu, G., Zheng, J., Sun, Z., Fan, W., Deng, G., Wang, H., Hu, F., Zhao, S., Fernie, A.R., Boerno, S., Timmermann, B., Zhang, P., Vingron, M., 2017. Haplotype-resolved sweet potato genome traces back its hexaploidization history. Nat. Plants 3, 696–703. https://doi.org/10.1038/s41477-017-0002-z

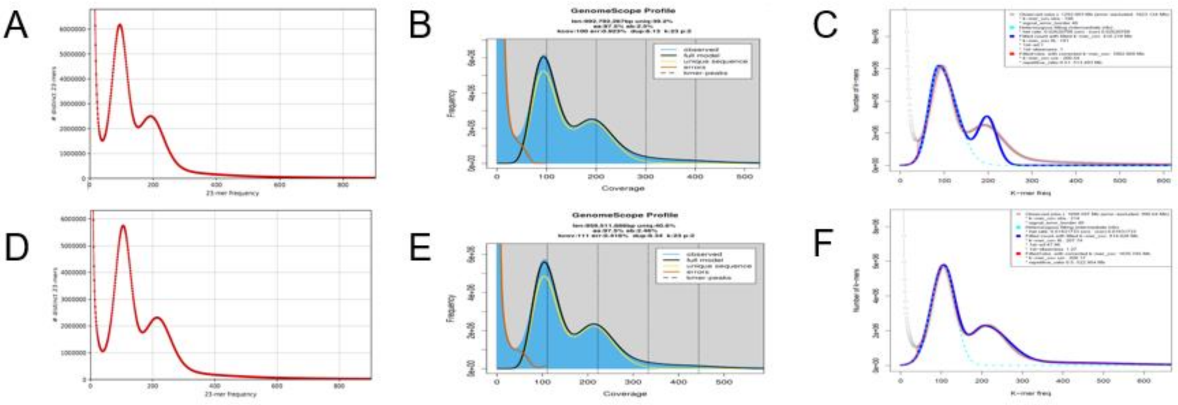
Yang, S.-F., Lu, C.-W., Yao, C.-T., Hung, C.-M., 2019. To Trim or Not to Trim: Effects of Read Trimming on the De Novo Genome Assembly of a Widespread East Asian Passerine, the Rufous-Capped Babbler (Cyanoderma ruficeps Blyth). Genes 10, 737. https://doi.org/10.3390/genes10100737

Ye, X., Al-Babili, S., Klo¨ti, A., Zhang, J., Lucca, P., Beyer, P., Potrykus, I., 2000. Engineering the Provitamin A (-Carotene) Biosynthetic Pathway into (Carotenoid-Free) Rice Endosperm. Science 287, 303–305. https://doi.org/10.1126/science.287.5451.303

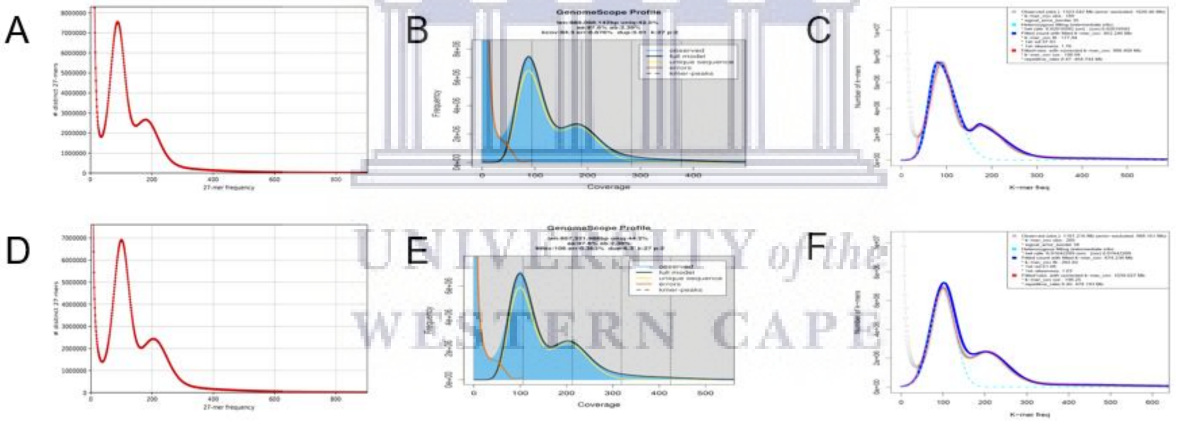
Zimin, A.V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S.L., Yorke, J.A., 2013. The MaSuRCA genome assembler. Bioinformatics 29, 2669–2677. https://doi.org/10.1093/bioinformatics/btt476

Zygosity [WWW Document], 2020. . Wikipedia. URL https://en.wikipedia.org/wiki/Zygosity (accessed 7.12.20).

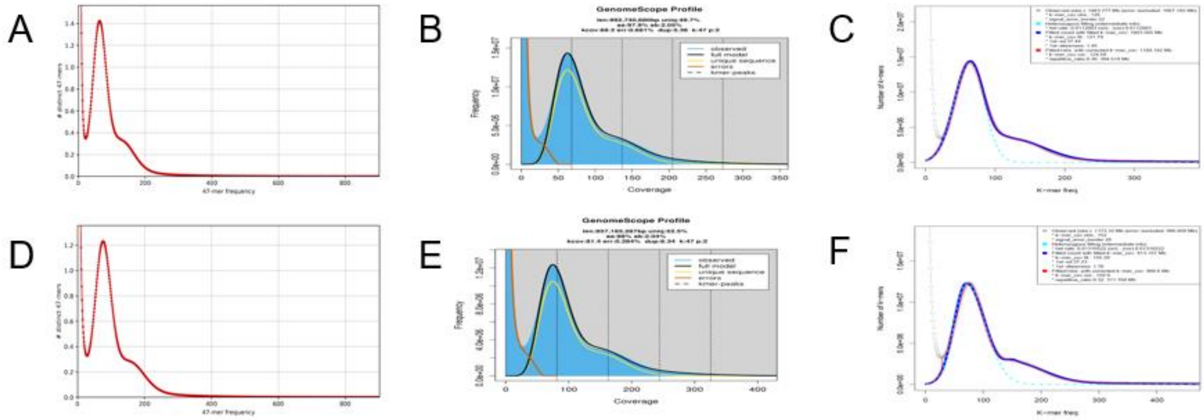
Supplementary Material



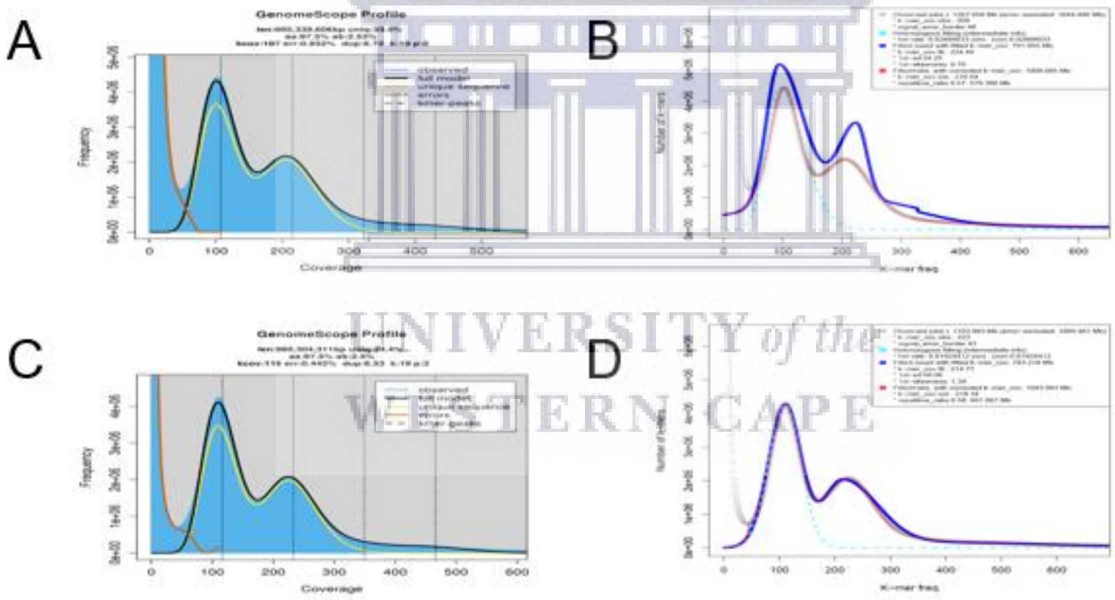
***Supplementary Figure 1: The effect of quality processing on k-mer spectra graphs. Results from the programs KAT, GenomeScope2, and FindGSE at k23. Histogram files were generated by KAT using the COMP-Raw and COMP-QP datasets. A)** KAT COMP-Raw, **B)** GenomeScope2 COMP-Raw, **C)** FindGSE COMP-Raw. **D)** KAT COMP-QP, **E)** GenomeScope2 COMP-QP, and **F)** FindGSE COMP-QP.*



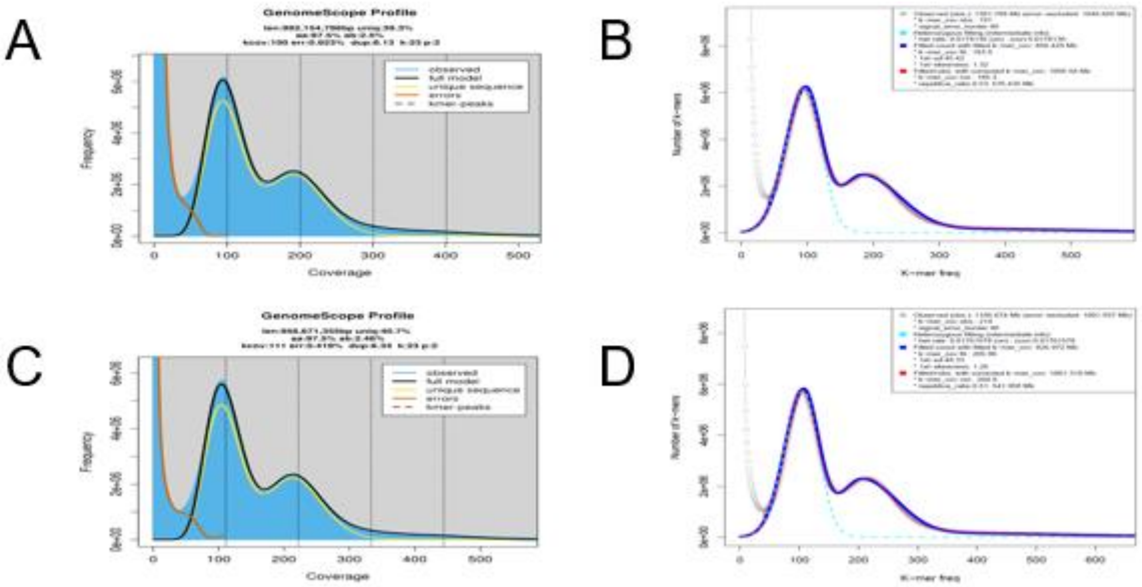
***Supplementary Figure 2: The effect of quality processing on k-mer spectra graphs. Results from the programs KAT, GenomeScope2, and FindGSE at k27. Histogram files were generated by KAT using the COMP-Raw and COMP-QP datasets. A)** KAT COMP-Raw, **B)** GenomeScope2 COMP-Raw, **C)** FindGSE COMP-Raw. **D)** KAT COMP-QP, **E)** GenomeScope2 COMP-QP, and **F)** FindGSE COMP-QP.*



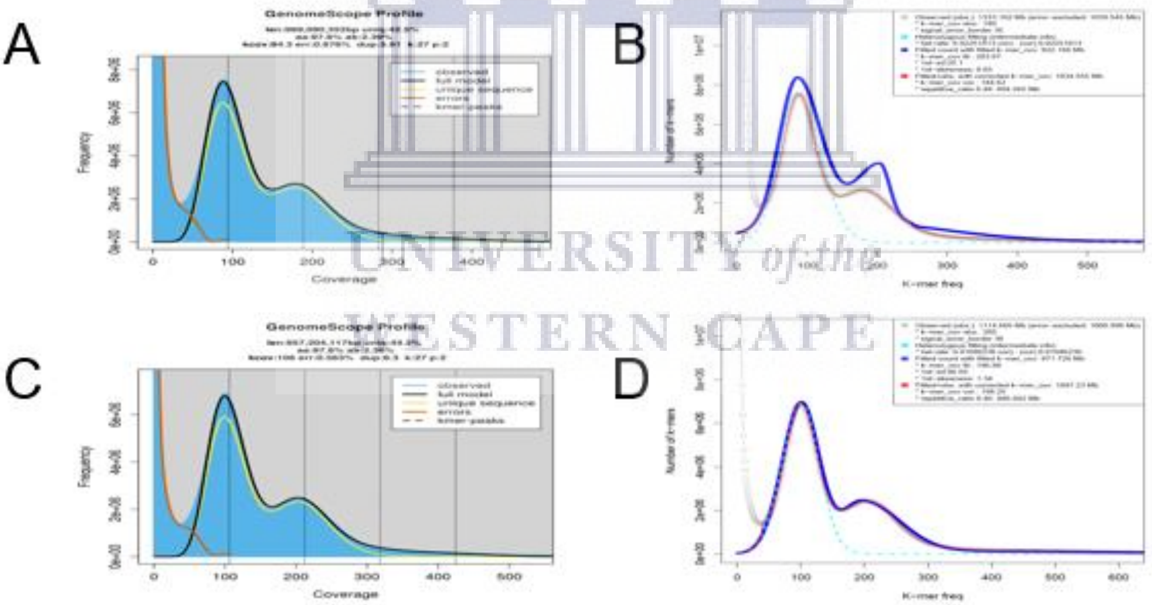
Supplementary Figure 3: The effect of quality processing on k-mer spectra graphs. Results from the programs KAT, GenomeScope2, and FindGSE at k47. Histogram files were generated by KAT using the COMP-Raw and COMP-QP datasets. A) *KAT COMP-Raw,* ***B)*** *GenomeScope2 COMP-Raw,* ***C)*** *FindGSE COMP-Raw.* ***D)*** *KAT COMP-QP,* ***E)*** *GenomeScope2 COMP-QP, and* ***F)*** *FindGSE COMP-QP.*



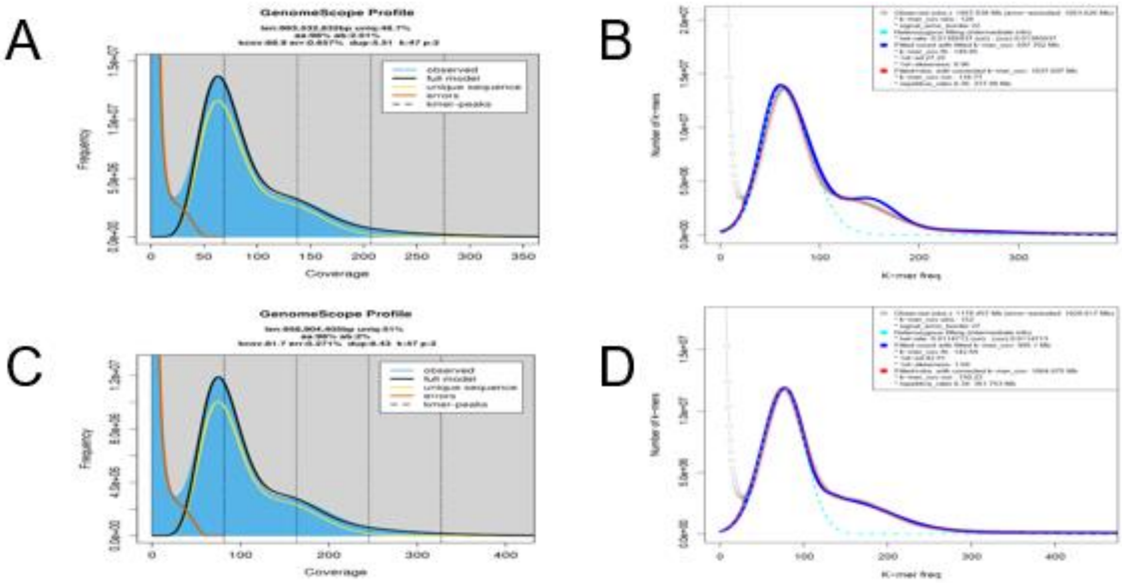
Supplementary Figure 4: The effect of quality processing on k-mer spectra graphs. Results from the programs GenomeScope2 and FindGSE at k19. Histogram files were generated by BBNorm using the COMP-Raw and COMP-QP datasets. A) *GenomeScope2 COMP-Raw,* ***B)*** *FindGSE COMP-Raw,* ***C)*** *GenomeScope2 COMP-QP, and* ***D)*** *FindGSE COMP-QP.*



Supplementary Figure 5: The effect of quality processing on k-mer spectra graphs. Results from the programs GenomeScope2 and FindGSE at k23. Histogram files were generated by BBNorm using the COMP-Raw and COMP-QP datasets. A) *GenomeScope2 COMP-Raw,* ***B)*** *FindGSE COMP-Raw,* ***C)*** *GenomeScope2 COMP-QP, and* ***D)*** *FindGSE COMP-QP.*



Supplementary Figure 6: The effect of quality processing on k-mer spectra graphs. Results from the programs GenomeScope2 and FindGSE at k27. Histogram files were generated by BBNorm using the COMP-Raw and COMP-QP datasets. A) *GenomeScope2 COMP-Raw,* ***B)*** *FindGSE COMP-Raw,* ***C)*** *GenomeScope2 COMP-QP, and* ***D)*** *FindGSE COMP-QP.*



Supplementary Figure 7: The effect of quality processing on k-mer spectra graphs. Results from the programs GenomeScope2 and FindGSE at k47. Histogram files were generated by BBNorm using the COMP-Raw and COMP-QP datasets. A) *GenomeScope2 COMP-Raw,* ***B)*** *FindGSE COMP-Raw,* ***C)*** *GenomeScope2 COMP-QP, and* ***D)*** *FindGSE COMP-QP*

Supplementary Table 1: Calculated difference of estimated genome sizes (in Gb) between KAT- and BBNorm generated histogram files. The average difference is 0,004 ± 0,016.

	K19		K23		K27		K47	
	Raw	**QP**	**Raw**	**QP**	**Raw**	**QP**	**Raw**	**QP**
GenomeScope 1 1k cutoff	-0,05	0,00	0,00	0,00	0,00	0,00	0,01	0,01
GenomeScope 2 1k cutoff	-0,05	0,00	0,00	0,00	0,00	0,00	0,01	0,01
GenomeScope 1 10k cutoff	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
GenomeScope 2 10k cutoff	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
GenomeScope 1 900k cutoff	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
GenomeScope 2 900k cutoff	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
FindGSE	-0,01	-0,01	-0,01	-0,01	-0,02	-0,01	0,07	-0,08
BBNorm	-0,01	-0,01	-0,01	-0,01	-0,01	-0,01	0,00	-0,01
Formula	-0,01	-0,01	-0,02	-0,01	-0,01	-0,01	0,01	-0,01

Appendix

Assembly scripts

Platanus K41 assembly:

ASSEMBLE K41 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=700GB
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusAssembleK41_EC.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusAssembleK41_EC.err
#PBS -N PlatK41_EC
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd /home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC

platanus assemble -o PlatanusAssembleK41_EC -f
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -k
41 -t 56 -m 700 2>AssembleFullK41_EC.log
```

SCAFFOLD K41 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56
#PBS -l walltime=48:00:00
#PBS -l place=excl
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusScaff_K71_EC_NoMP8.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusScaff_K71_EC_NoMP8.err
#PBS -N PlatK41_ScaffECNoMP8
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd /home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC

platanus scaffold -o PlatanusScaffoldK41_EC_NoMP8  -c
./PlatanusAssembleK41_EC_contig.fa -b ./PlatanusAssembleK41_EC_contigBubble.fa -IP1
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -
OP2
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R2.fastq -n1 200 -n2 2000 -a1 300 -a2 3000 -t 56 2>
ScaffoldK41_EC_NoMP8.log
```

GAPCLOSE K41 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=32:mpiprocs=32:mem=300GB
#PBS -l walltime=02:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusGapClose_K41_EC.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC/P
latanusGapClose_K41_EC.err
#PBS -N PlatK41_ScaffEC
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/01_K41_EC

platanus gap_close -o GapCloseFullStephEC_NoMP8_1 -c
ScaffoldFullK41_1_NoMP8_scaffold.fa -IP1
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -
OP2
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
```

```
S1L1_nc_ABC_R2.fastq -OP3
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R2.fastq  -t 32 2> GapCloseK41_EC.log
```

Platanus K71 assembly:

ASSEMBLE K71 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=700GB
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/PlatanusFullK
71_EC.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/PlatanusFullK
71_EC.err
#PBS -N PlatK71_EC
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd /home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC

platanus assemble -o PlatanusFullK71_EC -f
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -k
71 -t 56 -m 700 2>AssembleFullK71_EC.log
```

SCAFFOLD K71 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56
#PBS -l walltime=48:00:00
#PBS -l place=excl
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC/PlatanusScaff_K71_EC_NoMP8.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC/PlatanusScaff_K71_EC_NoMP8.err
#PBS -N PlatK71_ScaffEC
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC

platanus scaffold -o PlatanusScaffoldK71_EC  -c ./PlatanusFullK71_EC_contig.fa -b
./PlatanusFullK71_EC_contigBubble.fa -IP1
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -OP2
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_S1L1_nc_ABC_R2.fastq -n1 200 -n2 2000 -a1 300 -a2 3000 -t 56 2> ScaffoldK71_EC.log
```

GAPCLOSE K71 EC

```
#!/bin/bash
#PBS -l select=1:ncpus=32:mpiprocs=32:mem=300GB
#PBS -l walltime=04:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC/P
latanusGapClose_K71_EC_NoMP8.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC/P
latanusGapClose_K71_EC_NoMP8.err
#PBS -N PlatK71_GapCloseEC
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load Platanus/1.2.4

cd
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/03_Platanus/01_EC/00_K71_EC

platanus gap_close -o PlatanusGapClose_K71_EC_NoMP8 -c
PlatanusScaffoldK71_EC_NoMP8_scaffold.fa -IP1
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1_p.cor.fq
/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2_p.cor.fq -
OP2
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R2.fastq -OP3
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R1.fastq
```

```
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R2.fastq  -t 32 2> GapCloseK71_EC.log
```

ABySS 2.0 K41 assembly:

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=1000gb
#PBS -l place=excl
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5/Abyss41Full_EC.
out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5/Abyss41Full_EC.
err
#PBS -N Abyss41_EC
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module add ABySS/2.1.5_sh

cd /home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected

export TMPDIR=/home/astander2/lustre/temporary

abyss-pe k=41 G=1100000000 name=Abyss41Full_EC lib='L5 L6 S1L1 S1L2 S1L3 S1L4
S1L5 S1L6' mp='MP3 MP8' L5='./L5R1n_p.cor.fq ./L5R2n_p.cor.fq' L6='./L6R1n_p.cor.fq
./L6R2n_p.cor.fq' S1L1='./S1L1R1_p.cor.fq ./S1L1R2_p.cor.fq' S1L2='./S1L2R1_p.cor.fq
./S1L2R2_p.cor.fq' S1L3='./S1L3R1_p.cor.fq ./S1L3R2_p.cor.fq' S1L4='./S1L4R1_p.cor.fq
./S1L4R2_p.cor.fq' S1L5='./S1L5R1_p.cor.fq ./S1L5R2_p.cor.fq' S1L6='./S1L6R1_p.cor.fq
./S1L6R2_p.cor.fq'
MP3='/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_
3000_S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R2.fastq'
MP8='/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/next
clip/MP_8000_S2L2_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R2.fastq' -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5
```

ABySS 2.0 K71 assembly:

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=1000gb
#PBS -l place=excl
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5/Abyss71Full_EC
3.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5/Abyss71Full_EC
3.err
#PBS -N Abyss71_EC
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module add ABySS/2.1.5_sh

cd /home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected

export TMPDIR=/home/astander2/lustre/temporary

abyss-pe k=71 G=1100000000 name=Abyss71Full_EC lib='L5 L6 S1L1 S1L2 S1L3 S1L4
S1L5 S1L6' mp='MP3 MP8' L5='./L5R1n_p.cor.fq ./L5R2n_p.cor.fq' L6='./L6R1n_p.cor.fq
./L6R2n_p.cor.fq' S1L1='./S1L1R1_p.cor.fq ./S1L1R2_p.cor.fq' S1L2='./S1L2R1_p.cor.fq
./S1L2R2_p.cor.fq' S1L3='./S1L3R1_p.cor.fq ./S1L3R2_p.cor.fq' S1L4='./S1L4R1_p.cor.fq
./S1L4R2_p.cor.fq' S1L5='./S1L5R1_p.cor.fq ./S1L5R2_p.cor.fq' S1L6='./S1L6R1_p.cor.fq
./S1L6R2_p.cor.fq'
MP3='/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_
3000_S1L1_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_3000_
S1L1_nc_ABC_R2.fastq'
MP8='/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/next
clip/MP_8000_S2L2_nc_ABC_R1.fastq
/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/01_MP8/nextclip/M
P_8000_S2L2_nc_ABC_R2.fastq' -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/04_AbySS_2.1.5
```

SOAPdenovo2 configuration file used in both K41 and K71 assemblies:

```
max_rd_len=125

[LIB]

avg_ins=300

reverse_seq=0
```

```
asm_flags=3
rd_len_cutoff=125
rank=1
map_len=32
#L5
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R1n_p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L5R2n_p.cor.fq
#L6
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R1n_p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/L6R2n_p.cor.fq
#S1L1
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L1R2p.cor.fq
#S1L2
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L2R2p.cor.fq
#S1L3
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L3R2p.cor.fq
#S1L4
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L4R2p.cor.fq
#S1L5
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L5R2p.cor.fq
#S1L6
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R1p.cor.fq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/04_ErrorCorrected/S1L6R2p.cor.fq
# 3kbp insert library
[LIB]
avg_ins=3000
reverse_seq=1
asm_flags=2
rd_len_cutoff=101
```

```
rank=3
#MP3000
q1=/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_30
00_S1L1_nc_ABC_R1.fastq
q2=/home/astander2/lustre/00_GenomeRooibos/01_Data/01_lmp_Processed/nextclip/MP_30
00_S1L1_nc_ABC_R2.fastq
```

SOAPdenovo2 K41 job script

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=950gb
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
01_K41_EC_NoMP8/SoapK41_EC_NoMP8.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
01_K41_EC_NoMP8/SoapK41_EC_NoMP8.err
#PBS -N SoapK41_EC
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load SOAPdenovo2

cd
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
01_K41_EC_NoMP8
```

```
SOAPdenovo-63mer all -s ./00_ConfigFile.txt -o SOAPK41_EC_NoMP8.graph -K 41 -p 56 -
R -N 1100000000 1>SOAPK41_EC_NoMP8.log 2>SOAPK41_EC_NoMP8.err
```

SOAPdenovo2 K71 job script

```
#!/bin/bash
#PBS -l select=1:ncpus=56:mpiprocs=56:mem=950gb
#PBS -l walltime=48:00:00
#PBS -q bigmem
#PBS -W group_list=bigmemq
#PBS -P CBBI1133
#PBS -o
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
00_K71_EC_NoMP8/SoapK71_EC_NoMP8.out
#PBS -e
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
00_K71_EC_NoMP8/SoapK71_EC_NoMP8.err
#PBS -N SoapK71_ECNoMP8
#PBS -m abe
#PBS -M 3859586@myuwc.ac.za

module load chpc/BIOMODULES
module load SOAPdenovo2

cd
/home/astander2/lustre/00_GenomeRooibos/05_FullRuns/01_SOAPdenovo2/07_EC_2019/0
00_K71_EC_NoMP8

SOAPdenovo-127mer all -s ./00_ConfigFile.txt -o SoapK71_EC_NoMP8.graph -K 71 -p 56 -
R -N 1100000000 1> SoapK71_EC_NoMP8.log 2> SoapK71_EC_NoMP8.err
```