The impact of third generation sequencing on haplotype assembly

Shirali Hossein Zade, R.

DOI
10.4233/uuid:16494021-9bd2-4089-8808-5ad9dffadc5d

Publication date
2024

Document Version
Final published version

Citation (APA)
Shirali Hossein Zade, R. (2024). *The impact of third generation sequencing on haplotype assembly*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:16494021-9bd2-4089-8808-5ad9dffadc5d

THE IMPACT OF THIRD GENERATION SEQUENCING ON HAPLOTYPE ASSEMBLY

THE IMPACT OF THIRD GENERATION SEQUENCING ON HAPLOTYPE ASSEMBLY

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates,
to be defended publicly on Monday 26 February 2024 at 15:00 hour

by

Ramin SHIRALI HOSSEIN ZADE

Master of Science in Information Technology Engineering,
Sharif University of Technology, Iran,
born in Ahvaz, Iran.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. M.J.T Reinders,	Delft University of Technology, promotor
Dr. T.E.P.M.F. Abeel,	Delft University of Technology, promotor
Independent members:	
Prof. dr. ir. J.G. Daran,	Delft University of Technology
Prof. dr. G.W. Klau,	Heinrich Heine University Düsseldorf
Dr. ir. L.J.J. van Iersel,	Delft University of Technology
Dr. S. Smit,	Wageningen University & Research
Dr. D. Tamarit,	Utrecht University
Prof. dr. M.M. de Weerdt,	Delft University of Technology, reserve member

Keywords: Haplotype assembly, Genome assembly, Third generation sequencing, Genome repeats

Printed by: Ridderprint (www.ridderprint.nl)

Front & Back: DNA assembly.

ISBN 978-94-6384-539-7

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Sharing knowledge is the most
fundamental act of friendship.
Because it is a way you can give
something without loosing something.

Richard Stallman

CONTENTS

SUMMARY

The genome encompasses an organism's full DNA, organized into chromosomes within the cell nucleus. Humans have 46 paired chromosomes, and within these pairs, genetic information is grouped as haplotypes—genetic packages passed from one generation to the next, ensuring genetic diversity. While DNA sequencing produces short fragments or reads, assembling these back into a complete genome can be complex. The presence of multiple, similar haplotypes in some organisms amplifies this complexity, emphasizing the need for specialized techniques to accurately capture these subtle genetic variations.

In this thesis, we dive into the de novo and haplotype assembly challenges. We aim to tackle haplotype assembly challenges and find better ways to accurately assemble the genetic puzzle pieces. Along the way, we introduce a new tool for haplotype assembly designed to make the process more interpretable.

First, in Chapter 2, we offer an in-depth exploration of essential data types, approaches, and methods for accurate haplotype assembly, especially in the context of polyploid genomes. It delves into the latest advancements in next-generation sequencing (NGS) and third-generation sequencing (TGS), highlighting their increasing importance in haplotype assembly.

Then, in Chapter 3, we examine long-read de novo assemblers designed for eukaryotic genomes. By evaluating multiple popular long-read assemblers in various settings, we provide an in-depth comparison between them.

Next, in Chapter 4, we investigate the effect of repetitive sequences in de novo assembly and how they challenge the assembly problem. This chapter investigates strategies to tackle repeat-induced overlaps and illustrates how their successful mitigation can significantly enhance the assembly outcome.

Later, in Chapter 5, we introduce HAT as a cutting-edge tool in haplotype assembly. This chapter presents the features and capabilities of HAT, emphasizing its ability to seamlessly integrate short and long reads with reference genomes. Detailed evaluations underscore its stellar performance metrics, establishing HAT as a reliable tool for the haplotype assembly problem.

Finally, in Chapter 6, we focus on the genomes of four distinct Streptomyces strains. The intriguing discovery of potential new species, particularly Streptomyces sp. JH14 and Streptomyces sp. JH34, is elaborated upon. Through phylogenetic analysis, the chapter brings unique genetic attributes of these strains, offering novel insights into the world of Streptomyces and its potential implications in pathogenicity.

This thesis aims to assist other scientists with their assembly challenges, hoping that the final chapter will offer insights for future research in the field by emphasizing utilizing various technologies to address the haplotype assembly problem.

SAMENVATTING

Het genoom omvat het volledige DNA van een organisme, georganiseerd in chromosomen in de celkern. Mensen hebben 46 gepaarde chromosomen, en binnen deze paren is genetische informatie gegroepeerd als haplotypes: genetische pakketten die van de ene generatie op de andere worden doorgegeven, waardoor genetische diversiteit wordt gegarandeerd. Hoewel DNA-sequencing korte fragmenten of reads produceert, kan het weer samenstellen ervan tot een compleet genoom complex zijn. De aanwezigheid van meerdere, vergelijkbare haplotypes in sommige organismen versterkt deze complexiteit, wat de noodzaak benadrukt van gespecialiseerde technieken om deze subtiele genetische variaties nauwkeurig vast te leggen.

In dit proefschrift duiken we in de uitdagingen op het gebied van de novo en haplotype-assemblage. We streven ernaar uitdagingen op het gebied van de assemblage van haplotypes aan te pakken en betere manieren te vinden om de genetische puzzelstukjes nauwkeurig in elkaar te zetten. Gaandeweg introduceren we een nieuwe tool voor haplotype-assemblage, ontworpen om het proces beter interpreteerbaar te maken.

Ten eerste bieden we in hoofdstuk 2 een diepgaande verkenning van essentiële datatypen, benaderingen en methoden voor nauwkeurige haplotype-assemblage, vooral in de context van polyploïde genomen. Het duikt in de nieuwste ontwikkelingen op het gebied van sequencing van de volgende generatie (NGS) en sequencing van de derde generatie (TGS), waarbij het toenemende belang ervan bij de assemblage van haplotype wordt benadrukt.

Vervolgens onderzoeken we in Hoofdstuk 3 lang gelezen de novo assemblers die zijn ontworpen voor eukaryotische genomen. Door meerdere populaire langgelezen assemblers in verschillende omgevingen te evalueren, bieden we een diepgaande vergelijking daartussen.

Vervolgens onderzoeken we in Hoofdstuk 4 het effect van repetitieve reeksen bij de novo-assemblage en hoe deze het assemblageprobleem uitdagen. Dit hoofdstuk onderzoekt strategieën om door herhaling veroorzaakte overlappingen aan te pakken en illustreert hoe de succesvolle mitigatie ervan de uitkomst van de assemblage aanzienlijk kan verbeteren.

Later, introduceren we in hoofdstuk 5 HAT als een geavanceerd hulpmiddel bij de assemblage van haplotypes. Dit hoofdstuk presenteert de kenmerken en mogelijkheden van HAT, waarbij de nadruk wordt gelegd op het vermogen ervan om korte en lange metingen naadloos te integreren met referentiegenomen. Gedetailleerde evaluaties onderstrepen de geweldige prestatiestatistieken, waardoor HAT een betrouwbaar hulpmiddel is voor het haplotype-assemblageprobleem.

Ten slotte in Hoofdstuk 6, concentreren we ons op de genomen van vier verschillende Streptomyces-stammen. De intrigerende ontdekking van potentiële nieuwe soorten, met name Streptomyces sp. JH14 en Streptomyces sp. JH34, wordt uitgewerkt. Door middel van fylogenetische analyse brengt het hoofdstuk unieke genetische kenmerken van deze stammen naar voren, waardoor nieuwe inzichten worden geboden in de wereld van Streptomyces en de mogelijke implicaties ervan op het gebied van pathogeniteit.

Dit proefschrift heeft tot doel andere wetenschappers te helpen met hun assemblage-uitdagingen, in de hoop dat het laatste hoofdstuk inzichten zal bieden voor toekomstig onderzoek in het veld door de nadruk te leggen op het gebruik van verschillende technologieën om het haplotype-assemblageprobleem aan te pakken.

1

INTRODUCTION

Haplotype assembly holds the key to unlocking complex biological mysteries such as evolution, genetic diseases, and genomic diversity within species. Unfortunately, the majority of organisms are currently represented using what's known as a "consensus" reference genome - a condensed representation of an organism's complete genetic sequence that overlooks variations between homologous chromosomes. This simplified approach often fails to capture the full spectrum of genomic diversity because it misses which alleles are grouped together on a chromosome and omits alleles on other copies of the chromosome. This leads to gaps in our understanding of how physical traits are linked to their genetic origins. Therefore, developing methods that reconstruct complete haplotypes from DNA sequencing data is essential. However, these efforts often yield incomplete reconstructions, particularly when homologous chromosomes are highly similar. This dissertation delves into the complexities and challenges of haplotype assembly, and addresses some of these challenges.

This chapter aims to introduce the concept of haplotype assembly and related background, highlighting the difficulties of this problem. Additionally, we present an overview of the contributions made by this thesis to the advancement of this field.

1.1. WHAT IS A GENOME?

The collection of all the DNA molecules in each cell of an organism, which are identical throughout the cells, is known as the genome. DNA, or deoxyribonucleic acid, is a molecule that is composed of two long, twisted chains made of nucleotides. Nucleotides consist of a nitrogenous base, a sugar molecule, and a phosphate group. Nucleotides that compose DNA contain one of four nitrogenous bases: Adenine (A), cytosine (C), guanine (G), thymine (T), with each base corresponding to a different nucleotide [1]. The DNA molecules are mostly characterized by the sequence of nucleotides composing them. All self-reproducing forms of life that we know of have genetic information stored in the form of DNA molecules [2].

The genome of more complex organisms is composed of multiple copies of genetic information, with each copy originating from an ancestor. The number of copies of the genome in the somatic cells of an organism is referred to as ploidy, with haploid denoting a single copy, diploid relating to two copies, and polyploid referring to more than two copies [3] (See Figure 1.1).

1

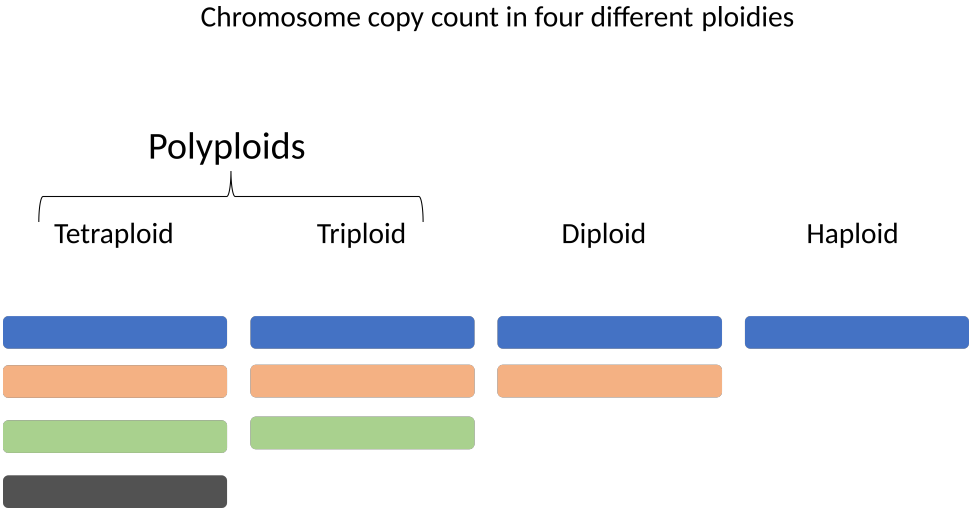


Figure 1.1: Illustration of Haplotypes of a chromosomes in a haploid, a diploid, a triploid, and tetraploid cell. Organisms that have more than two copies of homologous chromosomes are considered polyploids.

1.2. WHAT IS DNA SEQUENCING?

DNA sequencing is a process that determines the exact order of the nucleotides (A, C, G, and T) in a DNA molecule. The sequencing of DNA is an essential tool for researchers and scientists to study and understand the genetic code of organisms. Chapter 2, Table 2.3 gives a full overview of these sequencing technologies, and Chapter 2 Figure 2.3 shows how these technologies work. The earliest methods of DNA sequencing were a slow, labor-intensive, and expensive process that can only sequence one DNA fragment at a time. Next-generation sequencing (NGS) and third-generation sequencing (TGS) technologies have revolutionized the field of DNA sequencing, making it faster, more affordable, and allowing the sequencing of entire genomes or transcriptomes in a single run [4].

Next-generation sequencing (NGS) technologies use a variety of methods to sequence DNA, but they all involve the breaking of DNA into smaller fragments, attaching adapters to the ends of the fragments, and then amplifying and sequencing the fragments in parallel. Some of the commonly used NGS platforms include [2]:

- Illumina Sequencing: Illumina sequencing is a highly accurate and widely used NGS technology that uses reversible terminator chemistry to sequence millions of DNA fragments in parallel. In Illumina sequencing, the DNA is fragmented, and adapters are added to the ends of the fragments. The fragments are then attached to a flow cell, and the complementary strand is synthesized using fluorescently labeled nucleotides. The fluorescent signal is detected, and the base is identified, allowing the sequence to be determined.

- Ion Torrent Sequencing: Ion Torrent sequencing is a relatively new NGS technology that uses semiconductor sequencing to determine the sequence of DNA. In Ion Torrent sequencing, the DNA is fragmented, and adapters are added to the ends of the

fragments. The fragments are then attached to a semiconductor chip, and the complementary strand is synthesized using unlabeled nucleotides. The addition of each nucleotide releases a hydrogen ion, which is detected by a pH sensor, allowing the sequence to be determined.

Third-generation sequencing (TGS) technologies use a variety of methods to sequence DNA, but they all involve the direct reading of the nucleotides in a DNA molecule without the need for PCR amplification or sequencing-by-synthesis. Some of the commonly used TGS platforms include [2]:

- PacBio Sequencing: PacBio sequencing is a TGS technology that uses single-molecule, real-time (SMRT) sequencing to determine the sequence of DNA. PacBio SMRT sequencing employs a real-time, single-molecule approach to sequence DNA. PacBio SMRT sequencing uses a polymerase molecule to synthesize the complementary strand in real-time while the DNA strand is passed through a pore. Fluorescently labeled nucleotides are added to the DNA strand, emitting light signals that are detected to determine the sequence.
- Oxford Nanopore Sequencing: Oxford Nanopore sequencing is a TGS technology that uses nanopores to sequence DNA. In Oxford Nanopore sequencing, the DNA is passed through a nanopore, and the changes in electrical current caused by the passage of each nucleotide are detected, allowing the sequence to be determined.

Third-generation sequencing offers several advantages over next-generation sequencing, including longer read lengths, comparable accuracy, and the ability to sequence native DNA without amplification. For instance, PacBio sequencing can generate read lengths of up to 100 kb with an average accuracy of 99%, while Oxford Nanopore sequencing can produce read lengths of up to 2 Mb with an average accuracy of 90-95%. However, third-generation sequencing also has some limitations, such as lower throughput [5]. Even the latest third-generation sequencing machine, i.e., PacBio REVIO, has significantly lower throughput than the latest NGS machine, i.e., Illumina NovaSeq X.

In summary, DNA sequencing is a powerful tool that allows scientists to study the genetic code of organisms. NGS and TGS technologies have revolutionized DNA sequencing, making it faster, more affordable, and enabling the sequencing of entire genomes or transcriptomes in a single run. The choice of the sequencing platform depends on the specific needs of the experiment, including read length, accuracy, and cost.

1.3. WHAT IS GENOME ASSEMBLY?

Having the complete genome sequence of a species is important for a clear understanding of its biology. When we just look for differences by comparing to a reference genome, we might miss unique features that do not present in the reference. Big changes in the DNA, like sections being duplicated or moving around, can also be overlooked. Sequences that are very different from anything we've seen before might be ignored entirely. For species with more than two copies of each gene or a lot of variation, just comparing to a reference may not give us the whole picture. So, to truly understand the genetics of a species, it's important to have its complete genome sequence.

1

Current sequencing technologies are incapable of sequencing a chromosome from start to end, resulting in the production of smaller DNA fragments. Genome assembly is the process of piecing together the DNA sequence of an organism's genome from fragments of DNA generated by sequencing technologies. The goal of genome assembly is to reconstruct the original DNA sequence of an organism's genome, which typically contains billions of base pairs, in order to understand its genetic makeup and biological functions. There are two common approaches for genome assembly (see Figure 1.2), each with its own advantages and limitations [6]:

- De novo assembly: This approach involves reconstructing the genome from scratch, without the use of a reference genome. De novo assembly is particularly useful for organisms without a closely related reference genome or for those with significant genomic differences from the reference genome. De novo assembly typically involves sequencing the genome and using specialized algorithms to assemble overlapping reads into contiguous stretches of DNA called contigs. This is accomplished via two primary approaches: the Overlap-Layout-Consensus (OLC) method and the De Bruijn Graphs (DBG) method [7]. The OLC method involves finding overlaps between all read pairs, organizing them in a layout, and determining the consensus sequence. On the other hand, the DBG method transforms reads into a k-mer based graph called De Bruijn Graph, thereby simplifying the problem of assembly into finding a path through this graph.

- Reference-based assembly: In this approach, a reference genome is used as a guide to assemble the reads generated from sequencing the new genome. In reference-based assembly, two common approaches exist [7]. The first approach involves mapping reads to the reference genome to determine their genomic coordinates and create a consensus sequence. The second approach entails conducting a de novo assembly of the reads and subsequently aligning them to the reference genome to identify any misassembled regions. Reference-based assembly is typically faster and more accurate than de novo assembly, but it is limited by the availability and quality of the reference genome. Reference-based assembly is particularly useful for organisms with well-annotated reference genomes or for comparative genomics studies. However, reference-based assemblies are biased towards the reference genome and are less likely to preserve significant differences between the newly sequenced individual and the reference genome.

Genome assembly can be challenging due to the sheer size and complexity of the genome, as well as the presence of repetitive regions that can make it difficult to accurately assemble the genome. The presence of repeat regions in a genome can make genome assembly challenging because these regions contain sequences that are identical or very similar to each other (see Figure 1.3). This can cause sequencing reads to be misaligned or assigned to the wrong location, leading to errors in the assembly. Additionally, repetitive regions can cause gaps or breaks in the assembly, as it can be difficult to determine which sequences belong to which copy of the repeat. As a result, specialized algorithms and techniques, such as long-read sequencing and optical mapping, are often required to accurately assemble repetitive regions.

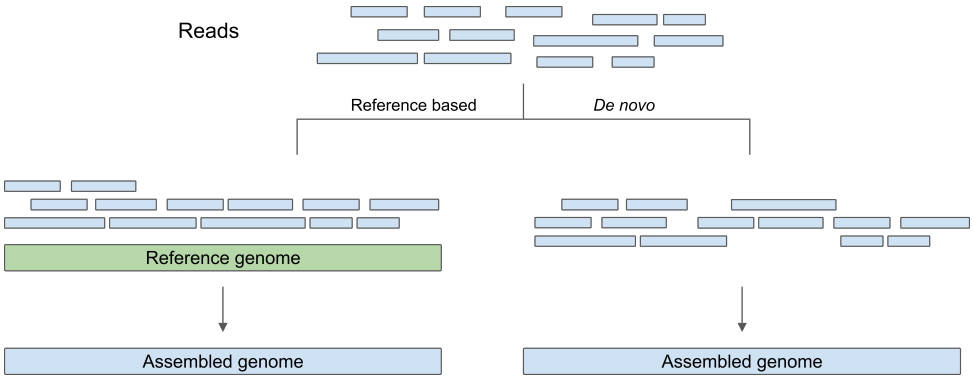


Figure 1.2: Illustration of the two main approaches for genome assembly.

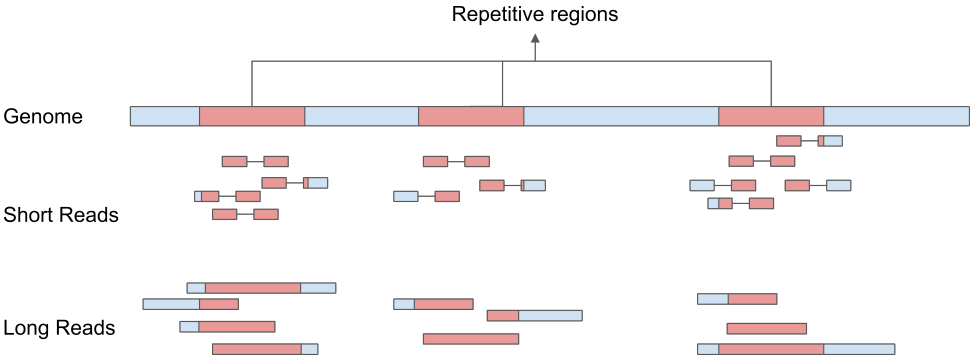


Figure 1.3: The genome has 3 instances of a repeat, and the reads originating from these regions largely have the same sequence.

1.4. WHAT IS HAPLOTYPE ASSEMBLY?

A haplotype is defined as a set of genetic information that are located together on the same chromosome. It can include as few as two genetic markers or as many as the entire chromosome's content. In this thesis, 'haplotype' refers to the whole set of genes or alleles on a single chromosome. For organisms that have more than one copy of genetic information (diploids and polyploids), standard assembly merges multiple copies of the genetic information as if they are identical, disregarding potential differences that hold biological significance. Ignoring these differences limits our understanding of how variations in different copies affect the organism's traits. To overcome this limitation, a specialized approach is required for polyploid genomes to distinguish and compare the distinct copies accurately.

Haplotype assembly is the process of reconstructing the DNA sequences of all haplotypes that comprise a genome. Haplotype assembly is a challenging problem in genomics, but it is essential for understanding the genetic basis of complex traits and diseases. There are several approaches for haplotype assembly, which can be broadly classified into three categories: (1) statistical methods, (2) sequencing-based methods, and (3) hybrid methods [8].

Statistical methods use population-level data to infer haplotypes. These methods assume that the genetic variants in a population are in linkage disequilibrium, meaning that they are inherited together more often than expected by chance. These methods rely on genotype data, which provide information about the specific genetic variants present at each genomic position in the individuals of the population. By analyzing patterns of genetic variation within a dataset, statistical algorithms can infer and reconstruct the underlying haplotypes or phased genetic sequences. However, a key limitation of these methods is the assumption that the genotype data represents all individuals in the population, when in reality, not every individual has been observed. This could hinder the accurate determination of more complex haplotypes.

Sequencing-based haplotype assembly utilizes high-throughput sequencing technologies to directly sequence DNA fragments, enabling the reconstruction of individual haplotypes. This approach is distinct from genome assembly as it specifically aims to reconstruct the sequence of every haplotype in genome, rather than providing a consensus representation of the chromosomes. By leveraging sequencing data, haplotype assembly methods aim to identify and link genetic variants that are present on the same chromosome. These methods benefit from advancements in long-read sequencing technologies, such as PacBio and Oxford Nanopore, which generate longer reads capable of spanning several genetic variants present on a haplotype. This is particularly advantageous in capturing complex haplotypes that would be challenging to reconstruct using shorter reads.

Hybrid methods combine statistical and sequencing-based approaches to improve haplotype assembly accuracy. These methods typically use DNA sequencing to generate an initial haplotype assembly, which is then refined using statistical methods based on population-level data.

Overall, haplotype assembly is a critical step in understanding the genetic basis of complex traits and diseases, and the development of new and improved methods for haplotype assembly is an active area of research in genomics.

1.5. APPLICATION OF HAPLOTYPE ASSEMBLY

Computationally assembled haplotypes have been applied in numerous analyses and studies, including those focusing on human diseases. For instance, a unique diagnostic test for a Miller syndrome patient used haplotype assembly, securing an accurate diagnosis. This would have been missed by haplotype unaware assembly methods due to the inability of such methods to demonstrate whether disease-associated SNPs are located in a single homologous chromosome or different ones [9, 10].

Likewise, computationally assembled haplotypes have been utilized in phylogenetic studies. Because haplotypes are inherited as a unit, analyzing the alleles located on the same haplotype within a population can provide insight on the evolution of genetic variations and help to track specific historical events. For example, by forming a phylogenetic tree from phased sequences of hexaploid sweet potato Ipomoea batatas, it was possible to trace the evolution of the six haplotypes and propose two whole-genome duplications in the sweet potato's history; the first occurring around 0.8 million years ago, the second around 0.5 million years ago [11].

In addition, computationally assembled haplotypes have been implemented in crop research and breeding programs. Many commercially significant agricultural crops, such as

wheat, potato, and banana, have polyploid genomes [7]. Haplotype reconstruction for various varieties and strains exhibiting a specific phenotype enables an understanding of the genotype associated with a certain trait present in (wild) crop relatives. This can then be introduced into the crop through breeding programs [12]. A notable example is the development of enhanced cultivars where assembled haplotypes led researchers to identify genotypes that could increase resistance to bacterial blight in rice and defend wheat against rust (13). Furthermore, recent studies have shown that single nucleotide polymorphism (SNP) markers fall short in determining genotypes related to crop yield in highly conserved genomic regions of wheat. However, haplotype resolution studies of the region have allowed researchers to identify the genotype for use in crop improvement programs [13, 14].

1.6. OUTLINE AND CONTRIBUTIONS

Recent breakthroughs in sequencing technology, particularly with long-read technologies, have significantly improved genome and haplotype assembly performance. However, methods utilizing long-read data still face challenges with repetitive sections in the genome, due to the complexity they introduce to the genome assembly problem. While long-read technologies are cutting-edge, it's crucial not to overlook the value of short-read sequencing. Short-read technologies, known for their high precision and high throughput, are particularly effective at detecting small genetic variations with great accuracy, which is valuable for haplotype assembly. A method that can effectively combine both long-read and short-read technologies could achieve superior performance in haplotype assembly. Looking into these challenges and providing methods to resolve repetitive regions, as well as utilizing short-read technologies alongside the cutting-edge long-read technologies for haplotype assembly, are the main objectives of this dissertation.

In this dissertation, we begin with a detailed review of haplotype assembly, followed by an in-depth examination of genome assembly, with a focus on long-read assembly tools for eukaryotic genomes. We discuss the importance of selecting the appropriate assembly algorithm and address the challenges posed by error-prone reads. Next, we explore the challenges of repetitive sequences on the genome assembly process and propose a method to overcome these challenges. We then proceed to assemble and analyze the genomes of novel Streptomyces strains. Lastly, we introduce a new haplotype assembly tool, named HAT, that leverages both long-read and short-read data to accurately assemble haplotypes of polyploid genomes.

1.6.1. A REVIEW ON COMPUTATIONAL METHODS FOR SEQUENCING-BASED HAPLOTYPE RECONSTRUCTION

In Chapter 2 we review the essential data types and computational methods for reconstructing haplotypes with a focus on polyploid genomes (More than 2 copies of the genetic information) and commonly available data types. We discuss data types, algorithms, tools and benchmarking requirements for haplotype assembly.

In contrast to similar reviews, we focus on frequently used methods that use the most common and accessible sequencing data: next-generation sequencing (NGS) and third-

generation sequencing (TGS), to reconstruct polyploid genomes.

Finally, we address the issue that benchmarking is often done ad hoc on particular datasets and the results are not generalizable. Therefore, we suggest a roadmap for benchmarking haplotype reconstruction methods and explore the requirements for the roadmap.

1.6.2. WHEN DO LONGER READS MATTER? A BENCHMARK OF LONG READ DE NOVO ASSEMBLY TOOLS FOR EUKARYOTIC GENOMES

Selecting the right assembly algorithm is crucial when generating genome assemblies for eukaryotic organisms from third-generation sequencing technologies. Although these technologies, such as ONT and PacBio, have improved over the limitations of NGS, their error-prone reads pose new challenges for assembly algorithms. With the plethora of tools available, it is essential to choose the appropriate assembler for a project.

To aid researchers in making informed decisions, in Chapter 3, we present a benchmark study of five commonly used long-read assemblers (Canu, Flye, Miniasm, Raven, and Redbean) using real and simulated datasets from various eukaryotic genomes with different read length distributions. We evaluated the assemblers using reference-based metrics, assembly statistics, misassembly count, BUSCO completeness, runtime, and RAM usage. Our results show that Flye is the best-performing assembler overall, but there is no single assembler that performs the best in all categories. Moreover, we found that longer read lengths generally improve assembly quality, but the extent of the improvement depends on the size and complexity of the reference genome.

1.6.3. GRAPHCLEAN: IMPROVING DE NOVO ASSEMBLY BY REMOVING REPEAT-INDUCED OVERLAPS

Accurate genotyping, vital for connecting phenotypes with genotypes, often requires de novo genome assembly. Despite advancements in sequencing, repetitive sequences still complicate assembly by creating misleading overlaps in the assembly graph. Chapter 4 aims to enhance de novo assembly algorithms by removing repeat-induced overlaps and analyzing their effect on assembly performance. We demonstrate the potential improvements in assembly by removing repeat-induced overlaps and propose various methods for detecting and eliminating them. We evaluate the performance of these methods using multiple simulated datasets.

1.6.4. HAT: HAPLOTYPE ASSEMBLY TOOL USING SHORT AND LONG READS

Advancements in TGS technologies have led to the development of various methods for reconstructing complete haplotypes from DNA sequencing data. However, even with these advancements, the resulting reconstructions often remain incomplete, especially when homologous chromosomes have few differences. This is because current haplotype assembly techniques face difficulties in accurately distinguishing between regions that are identical across multiple haplotypes and regions that are distinct.

To overcome these challenges, in Chapter 5, we introduce HAT, a haplotype assembly tool that combines short and long reads with a reference genome. HAT utilizes the accuracy of short reads and the longer span of long reads to reconstruct haplotypes. A critical

aspect of HAT is the identification of multiplicity blocks, which represent regions where haplotypes differ, enabling more precise and interpretable results. We evaluated HAT using Saccharomyces pastorianus CBS1483, an aneuploid yeast strain, as well as multiple simulated polyploid datasets of the same strain. Our findings demonstrate that HAT surpasses existing tools for haplotype assembly in terms of performance and accuracy.

1.6.5. GENOMES OF FOUR STREPTOMYCES STRAINS REVEAL INSIGHTS INTO PUTATIVE NEW SPECIES AND PATHOGENICITY OF SCAB-CAUSING ORGANISMS

Chapter 6 focuses on the genomes of four Streptomyces isolates from potato crops in Colombia. Two of the isolates, Streptomyces sp. JH14 and Streptomyces sp. JH34, are potential new species, while the other two, Streptomyces sp. JH002 and Streptomyces sp. JH10, are non thaxtomin-producing pathogens. Our collaborators in Colombia isolated and sequenced the samples using PacBio SMRT, which we then used to assemble the four strains and create a phylogenetic tree to classify them. The results of the phylogenetic analysis, based on single-copy core genes, confirmed that the two pathogenic isolates belong to different lineages, Streptomyces pratensis and Streptomyces xiamenensis, respectively, and do not share a common ancestor with known pathogenic species. We also discovered the presence of unknown gene clusters and clusters associated with the synthesis of medicinal compounds and potentially linked to pathogenicity in the pathogenic isolates. Interestingly, we did not find genes similar to the protein-coding genes characteristic of scab-causing streptomycetes shared by known pathogenic species. Most genes involved in biosynthesis of known virulence factors were not present in the scab-causing isolates (S. sp. JH002 and S. sp. JH010), but we identified Tat-system substrates likely to be involved in pathogenicity. Finally, we confirmed the presence of a putative mono-ADP-ribosyl transferase homolog to scabin in S. sp. JH002. Overall, these isolates may produce novel secondary metabolites and virulence factors uncommon in Streptomyces spp.

1.6.6. DISCUSSION

In our discussion, we first connect the ideas from the earlier chapters, showing how they fit together. Next, we talk about the benefits of using multiple DNA-reading tools at once. This combination gives a clearer view of DNA details and strengthens our research. We also explore the similarities of metagenomics assembly and haplotype assembly, and how different strains in a microbial community can be looked as different haplotypes. Wrapping up, we turn our focus to future trends in genome assembly. Two main developments are of particular interest. Firstly, we're witnessing a growing preference for custom computational methods tailored for long-read sequencing data. Secondly, the emergence of high-accuracy long-read technologies offers the promise of more in-depth and accurate sequencing results. However, these innovative techniques come with their set of challenges. One notable hurdle is the integration of data from the newer long-read technologies into systems that were initially developed for short-reads. This poses a challenge that requires careful consideration for those aiming for a comprehensive understanding of genome assembly.

BIBLIOGRAPHY

[1] B. Alberts et al. “The structure and function of DNA”. In: *Molecular Biology of the Cell. 4th edition*. Garland Science, 2002.

[2] A. Hiyoshi et al. “Does a DNA-less cellular organism exist on Earth?” In: *Genes to Cells* 16.12 (2011), pp. 1146–1158.

[3] A. C. Gerstein and S. P. Otto. “Ploidy and the causes of genomic evolution”. In: *Journal of Heredity* 100.5 (2009), pp. 571–581.

[4] J. M. Heather and B. Chain. “The sequence of sequencers: The history of sequencing DNA”. In: *Genomics* 107.1 (2016), pp. 1–8.

[5] D. Lang et al. “Comparison of the two up-to-date sequencing technologies for genome assembly: HiFi reads of Pacific Biosciences Sequel II system and ultra-long reads of Oxford Nanopore”. In: *Gigascience* 9.12 (2020), giaa123.

[6] J. R. Miller, S. Koren, and G. Sutton. “Assembly algorithms for next-generation sequencing data”. In: *Genomics* 95.6 (2010), pp. 315–327.

[7] M. Kyriakidou et al. “Current Strategies of Polyploid Plant Genome Sequence Assembly”. In: *Frontiers in Plant Science* 9 (Nov. 2018), p. 1660. DOI: 10.3389/fpls.2018.01660.

[8] S. Garg et al. “Chromosome-scale, haplotype-resolved assembly of human genomes”. In: *Nature biotechnology* 39.3 (2021), pp. 309–312.

[9] S. B. Ng et al. “Exome sequencing identifies the cause of a mendelian disorder”. In: *Nature genetics* 42.1 (2010), pp. 30–35.

[10] S. B. Ng et al. “Targeted capture and massively parallel sequencing of 12 human exomes”. In: *Nature* 461.7261 (2009), pp. 272–276.

[11] J. Yang et al. “Haplotype-resolved sweet potato genome traces back its hexaploidization history”. In: *Nature plants* 3.9 (2017), pp. 696–703.

[12] J. A. Bhat et al. “Features and applications of haplotypes in crop breeding”. In: *Communications biology* 4.1 (2021), p. 1266.

[13] R. K. Varshney et al. “Designing future crops: genomics-assisted breeding comes of age”. In: *Trends in Plant Science* 26.6 (2021), pp. 631–649.

[14] J. Brinton et al. “A haplotype-led approach to increase the precision of wheat breeding”. In: *Communications Biology* 3.1 (2020), p. 712.

2

A REVIEW OF COMPUTATIONAL METHODS TO RECONSTRUCT POLYPLOID HAPLOTYPES BASED ON DNA SEQUENCING DATA

Haplotypes are the sets of alleles lying together on individual chromosomes. Accurate and efficient haplotype reconstruction is critical to characterize chromosome-level genetic variation in diploid or polyploid organisms. Because of technical limitations and computational restrictions, however, this remains a challenging problem.

In this review, we aim to elucidate the current problems in haplotype reconstruction and how existing algorithms address these obstacles. First, we introduce haplotype assembly, then dive into the challenges of haplotype assembly and how a high ploidy affects the polyploidy haplotype assembly problem. Next, we discuss approaches for resolving haplotypes and the limitations of the existing methods. We explain the impact of distinct molecular data types on the haplotype reconstruction problem, as well as the information they provide for solving this complex puzzle. In the third section we cover the available tools for reconstructing haplotypes in diploid, polyploid organisms, and microbial communities. Finally, we address the lack of systematic benchmarking and the resulting lack of generalizability of results by proposing a synthetic benchmarking scheme for haplotype assembly tools.

2.1. INTRODUCTION

Eukaryotic genomes commonly contain multiple homologous copies of each chromosome [1]. Haploid organisms (e.g. most bacteria) have only one copy of each chromosome, diploid organisms have two homologous pairs (e.g. humans), and polyploid organisms have more than two (e.g. tetraploid potato). Polyploidy is common in plants [2] and although there is no known upper limit to ploidy, values exceeding six are rare (Supplementary Figure A.1). Some organisms, like the yeast strain Saccharomyces pastorianus, have a different number of copies for different chromosomes, and are called aneuploids [3]. The set of alleles co-located in a single chromosome copy is called a haplotype.

Haplotype resolution has been desirable since the beginning of the genome sequencing era because it yields essential information about whole genomes, such as the haplotype colocation, allelic identity, homologous allele copy number [4–6]. This information might be crucial for understanding the crops, phylogeny, and alleles causing some diseases [7]. Without haplotype resolution, researchers have often approximated whole genomes into a consensus sequence, which is a flattened genome model of all the haplotypes. However, this model cannot approximate all the necessary information for a researcher to understand the genomic model.

Determining the haplotype is performed by using the reads produced by DNA sequencing technologies. Regardless of the sequencing platform, computational methods are required to reconstruct haplotypes from sequencing reads, because the read lengths are inherently shorter than chromosomal lengths. While whole chromosome sequencing is the theoretical maximum upper limit to fragment length, to date no sequencing platform can offer chromosomal read length sequencing.

Computational haplotype reconstruction approaches are often classified into two classes: (i) haplotype assembly and (ii) haplotype phasing. Haplotype assembly aims to reconstruct every haplotype from sequencing data of the individual. In contrast, haplotype phasing is typically understood to reconstruct haplotypes from genotype information of samples in a population [8, 9]. In this manuscript, we focus on haplotype assembly and use the terms haplotype assembly and haplotype reconstruction interchangeably. Figure 2.1.A and Figure 2.1.B illustrate the process of haplotype reconstruction in diploid and polyploid genomes respectively. Computational haplotype reconstruction is not straightforward and has several challenges such as repeats, which are difficult to resolve in genome assembly, or structural variants and variation deserts, which are specifically challenging for haplotype reconstruction.

Furthermore, when reconstructing haplotypes, ploidy needs to be accurately estimated to determine the correct number of haplotypes to reconstruct. Historically, karyotyping has been required to determine chromosome number, however this is not possible for every kind of organism and tissue type [10]. Therefore, researchers rely on computational methods to estimate the ploidy number. However, computational methods for ploidy estimation have limitations, such as assuming the entire chromosome has continuous ploidy [11] and only distinguishing diploid, triploid, and tetraploid genomes, thus aneuploidy or higher ploidies are not distinguished [12].

Computationally reconstructed haplotypes have been used in many downstream analysis and studies such as human diseases. For example, in a first-of-its-kind diagnostic assay for a patient with Miller syndrome [12, 13], haplotype assembly was employed, resulting

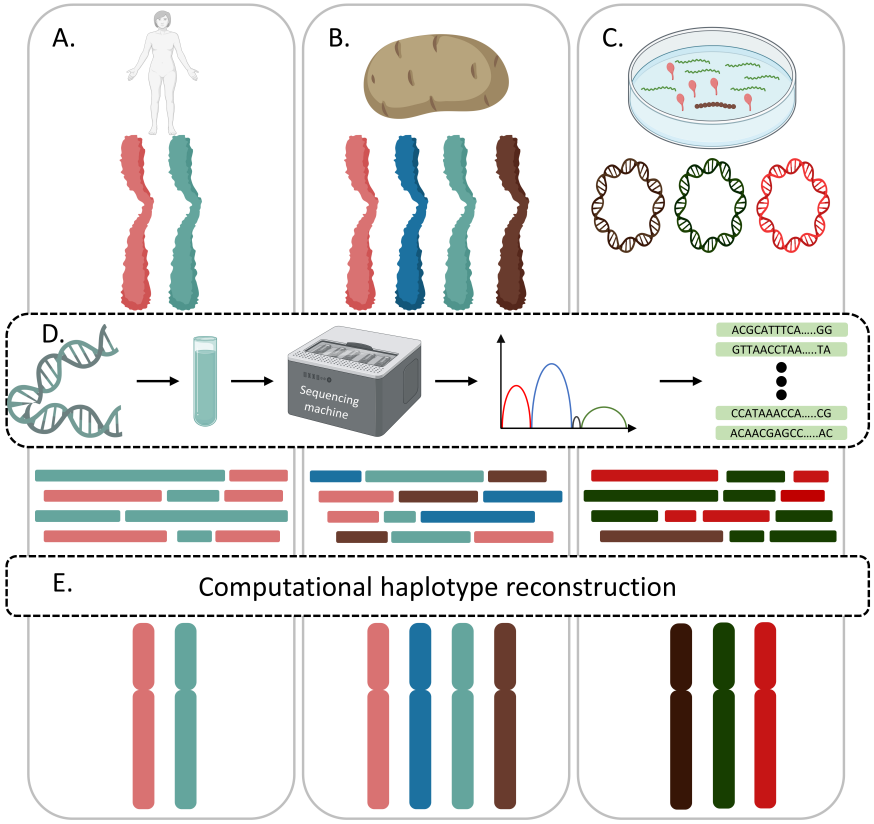


Figure 2.1: A. Human's genome is diploid (red and light green homologues chromosome copies). The chromosomes are sequenced, and the goal is to reconstruct each of the chromosome copies from the reads. The homologous chromosomes are similar, and it is not obvious to capture the differences between the chromosomes from the reads. B. The potato genome is a tetraploid, and the goal is to reconstruct all of the haplotypes and capture their differences. C. The haplotype assembly problem is similar to metagenomic assembly. A bacterial community has multiple species and often, the whole community is sequenced together. Reconstructing each species' genome is essential to study the community. D. Extracting DNA from the cell and sequencing it with the DNA sequencing machines to produce reads. E. Using a computational method to reconstruct the haplotypes from the reads.

in a correct diagnosis where haplotype unaware assembly methods would have otherwise misdiagnosed the patient, due to the inability to show that disease-associated SNP's are co-locating in a single homologous chromosome or different ones.

Moreover, computationally reconstructed haplotypes have been used in phylogenetic studies. Since haplotypes are inherited together, investigating the alleles co-locating in the population's haplotypes provides insights into how the genetic variations are shaped and help us trace a specific event in history. For instance, creating a phylogenetic tree from phased sequences of hexaploid sweet potato Ipomoea batatas helped trace the evolution of the six haplotypes and suggest two whole-genome duplications in the history of sweet

potato Ipomoea batatas which the first one happened around 0.8 million years ago and the second one happened around 0.5 million years ago [14].

Furthermore, computationally reconstructed haplotypes have been used in crop studies and breeding programs. Many agricultural commercially important crops have polyploid genomes, including cotton [15], wheat [16], potato [16], banana [16], oilseed rape [17], sugar cane [18], and many others. Haplotype reconstruction of several varieties and strains with a specific phenotype allows understanding of the genotype linked to specific trait present in (wild) relatives of crops [19] which can be used to inject the trait into the crop with breeding programs. One notable example is improved cultivars, which reconstructed haplotypes allowed researchers to find the genotypes that could increase resistance to bacterial blight in rice and protect wheat against rust [20]. In addition, recent studies revealed that single nucleotide polymorphism (SNP) markers are insufficient to determine genotypes associated with crop yield in the highly conserved genomic regions of wheat. Meanwhile, the haplotype resolution study of the region allowed researchers to determine the genotype and use it in crop improvement programs [20, 21].

While haplotype reconstruction is generally discussed in the context of eukaryote genomes such as plants, metagenomic assembly can be considered a related computational problem to haplotype assembly. In microbial communities, thousands of diverse species live, and the metagenomic assembly aims to reconstruct their genome from sequencing data. For instance, the human gut microbiome is estimated to have thousands of diverse species of archaea, bacteria, fungi, protozoa, and viruses [22]. As shown in Figure 1.C, each haplotype in metagenomics assembly represents the genome of an individual microorganism in the sequenced community. The reconstruction of microbial haplotypes is particularly useful because 99% of the environmental microorganisms are not culturable [23, 24] and the genomes of these microorganisms are usually sequenced and studied altogether [24–26].

Metagenomic assembly problem can be performed at two resolutions: (i) species or (ii) strain level. At the species level, one examines different species in the community by reconstructing a single representative genome for each of species. On the other hand, the strain level metagenomics assembly focuses on characterizing and reconstructing the genomes of individual strains of species [27]. Strain level metagenomics assembly is one of the important methods in studying microbial infections to identify which strain infects a patient for using a suitable medicine [28]. Some strains express antibiotic resistance genes which need to be considered before prescribing a medicine. Moreover, sometimes there is a "mixed infection", and multiple strains simultaneously infect a patient. For example, it has been shown that in Mycobacterium tuberculosis mixed infections the co-infecting strains can have complementary antibiotic resistance mutations, and strain level metagenomics assembly is essential to discover the best treatment for the patient [27, 28].

The virus quasispecies problem also falls under the category of haplotype reconstruction. Viral quasispecies aims to reconstruct the genomes of virus strains that infect hosts and determine their genetic variations. Virus replications are error-prone and the substitutions rate is estimated to be as high as one error every thousand nucleotides per replication cycle [29]. Reconstructing the haplotypes of the closely related viruses is essential for studying a viral borne disease, pandemics, or detection of drug resistance in a patient [30]. It is essential to reconstruct the genome of all individual viruses and capture even the slightest

differences between them to study and identify the drug resistance and virulence factors in the virus's genome [31, 32]. For example, because of the high mutation rate of HIV, patients are usually infected with many strains with different drug-resistant mutations and haplotype reconstruction of these strains can be a solution to detect the mutations and efficient drug administration [33].

This manuscript reviews the essential data types and computation methods for reconstructing haplotypes in diploid, polyploid genomes, and metagenomic communities. In contrast to the latest reviews on haplotype reconstruction [32] that spotlight the newer sequencing technologies like Hi-C reads that are expensive and less accessible, in this manuscript we focus on newly developed methods that use common and reachable sequencing data: next-generation sequencing (NGS) and third-generation sequencing (TGS), to reconstruct polyploid haplotypes. Moreover, we investigate the requirements for benchmarking a haplotype assembly tool.

2.2. HAPLOTYPE RECONSTRUCTION APPROACHES

Haplotype reconstructions methods can be categorized into four groups based on the data required: 1) de novo approaches only use sequencing data of the individual to reconstruct haplotypes, 2) reference-based approaches, use sequencing data of the individual in addition to an existing reference genome, 3) trio binning approaches use sequencing data from both the parents and the individual itself to more accurately infer haplotypes, 4) experimental chromosome separation methods rely on lab work to separate or tag the haplotypes before sequencing and reduce the computational challenge (Figure 2.2).

2.2.1. DE NOVO HAPLOTYPE RECONSTRUCTION

De novo haplotype assembly (see Figure 2.2.A) uses the least amount of data compared to the other methods, and is applicable for all use cases. In de novo haplotype reconstruction, only DNA sequencing data is used for the reconstruction. These methods distinguish themselves from referenced-based methods by not using a reference sequence, and only rely on pair-wise alignment of the reads. Regardless, de novo haplotype is computationally challenging because both de novo assembly and haplotype reconstruction are tackled in a single algorithmic step. Some algorithms in this class, like FALCON-UNZIP [34], try to split the assembly and haplotyping problem by first creating a haploid assembly from the sequencing data and then using a reference-based haplotype reconstruction method. For example, the diploid *Vitis vinifera* haplotypes have been reconstructed by taking a de novo assembly approach and subsequently unzipping the haplotypes in the genome assembly process [34].

2.2.2. REFERENCE-BASED HAPLOTYPE RECONSTRUCTION

Reference-based haplotype reconstruction methods use sequencing data of the homologous chromosomes, just like de novo approaches, and a (previously) constructed reference genome of the chromosome to reconstruct the haplotypes. First, homologous chromosomes are sequenced in bulk, and the reads are mapped to the reference genome of the chromosome to find the variant loci and the approximate location in the chromosome they originate from (see Figure 2.2.B). The reads covering the same regions are aligned together to capture the allelic differences between them. Later, these approaches try to connect and link

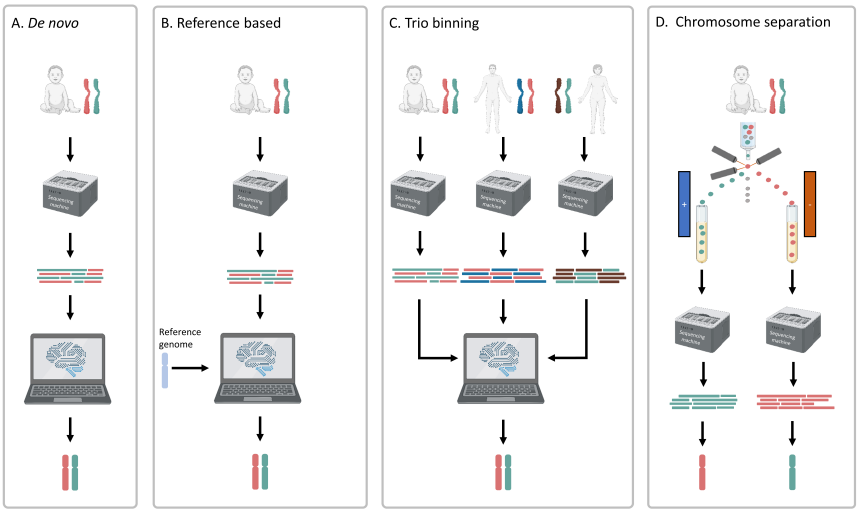


Figure 2.2: A. The genome is sequenced, and the reads might originate from either of the haplotypes. The goal is to reconstruct the haplotypes only based on the reads. B. Additionally to reads, a reference genome is used to reconstruct the haplotypes. C. DNA sequencing data from the parents' genomes are used with the target's sequencing data to reconstruct the haplotypes. D. Chromosomes are separated first and sequenced individually. This approach sidesteps all computational challenges in haplotype reconstruction and turns it into an experimental challenge to separate chromosomal DNA molecules.

the alleles together to reconstruct the haplotypes. Though reference-based haplotype reconstruction is usually less challenging than de novo haplotype reconstruction, the method suffers from a reference bias because it depends on the reads' alignment to the reference genome and downstream analysis to detect variations between haplotypes. This reference bias precludes capturing large variations between the haplotypes, such as structural variants. To overcome reference bias, some of the reference-based haplotype reconstruction methods, like HAT [35] and nPhase [36], assign reads to haplotypes based on the alleles determined for the heterozygous loci and assemble the haplotype-separated reads afterwards.

2.2.3. TRIO BINNING HAPLOTYPE RECONSTRUCTION

In this approach, both the target genome as well as the genomes of its two parents are sequenced. Since haplotypes are inherited from the parents, they are also available in the parents' genome. Trio binning methods take advantage of this data to find unique k-mers which are different between maternal and parental reads (see Figure 2.2.C). Then, these unique k-mers are used to bin the reads of the F1 offspring into a maternal and a parental group that can be assembled separately to construct the haplotypes. However, this technique is only available for species with sexual reproduction. Furthermore, it also increases the required data threefold to include parental data, which makes trio-binning unpractical for many applications. These approaches have been successfully applied to reconstruct haplotypes of human [37] and the wood tiger moth [38] genomes.

2.2.4. SINGLE CHROMOSOME SEQUENCING

There are various methods to separate the homologous chromosomes before sequencing to get per-chromosome information. As the data from each of the haplotypes are already separated, these methods avoid computational challenges in haplotype reconstruction. Regardless, these methods are often expensive and have complicated library preparation, and they also require intact cells, which makes them primarily suitable for niche applications [39].

For instance, chromosome sorting is a method to isolate an individual chromosome before sequencing, which allows separating the reads based on their origin before the assembly process (see Figure 2.2.D). There are several chromosome sorting methods, like microscopy-based chromosome isolation, fluorescence-activated sorting, and microfluidics-based sorting individual chromosomes [39]. Regardless of the sorting procedure, single-chromosome sequencing requires DNA amplification [39]. Therefore it suffers from amplification bias, and sequencing coverage across the chromosome is sparse, which leads to a lack of coverage in heterozygous sites [39]. Alternatively, it is feasible to use the natural way to separate homologous chromosomes from each other and sequence the DNA of the gametes. However, access to the required tissue is not straightforward for many organisms [39].

Single-cell DNA template strand sequencing (Strand-seq) is a single-cell sequencing method that can be leveraged to reconstruct accurate chromosome-scale haplotypes. In Strand-seq, the synthesized strand is ligated with bromodeoxyuridine (BrdU) molecules during mitosis. The daughter cell is sequenced, and the strands with BrdU molecules are destroyed, leaving only the original template strand of the parental chromatids. The direction of the template strands is identified by comparing them to the reference. If a cell has a different direction for the homolog template strands, the reads generated from this cell can get clustered based on their direction. The reads with the same direction are from the same chromatid in the parent cell. Recently, the strand-seq technology was used to reconstruct human haplotypes [40].

2.3. DNA SEQUENCING AND LONG-RANGE TECHNOLOGIES FOR HAPLOTYPE ASSEMBLY

Researchers have utilized a variety of technologies in haplotype reconstruction research. An overview of these technologies is provided in this section, along with an explanation of how they can be used for haplotype assembly (see Table 2.1).

The introduction of NGS and TGS technologies, such as various products of Illumina, PacBio and Oxford Nanopore Technologies, have revolutionized genome assembly due to the high throughput data they are able to provide, while remarkably reducing the time and cost of sequencing large genomes [45]. Regardless, the reads produced by most of the NGS technologies range from tens of base pairs to 600 base pairs [46], so they do not span most repetitive regions. That leads to ambiguous placement of reads when aligning to the reference and difficulties for genome assembly. Even when paired end reads are employed, the inner distance of the linked pairs is usually not longer than the read lengths of NGS reads [47]. Therefore, NGS technologies can only be leveraged to link variants that fall within the insertion size (see Figure 2.3.A).

Table 2.1: A summary of commonly used DNA sequencing technologies. Sequencing technologies Read length Read Accuracy paired linking range.

Sequencing technologies	Read length	Read Accuracy	Paired	Linking range
Illumina miseq	250-300bp [41]	99% [41]	Yes	Pairs insertion size.
Illumina Hiseq	90-151bp [41]	99% [41]	Yes	Pairs insertion size.
PacBio SMRT	Average 10kbp [42]	87% [42]	No	Read length
PacBio HIFI	Average 10kbp [42]	99% [42]	No	Read length
Oxford nanopore technology	Average 10kbp [42]	92% [42]	No	Read length
10x genomics [43]	Depends on the sequencing platform			∼50kb
Hi-C [44]	Depends on the sequencing platform			Mega base pairs

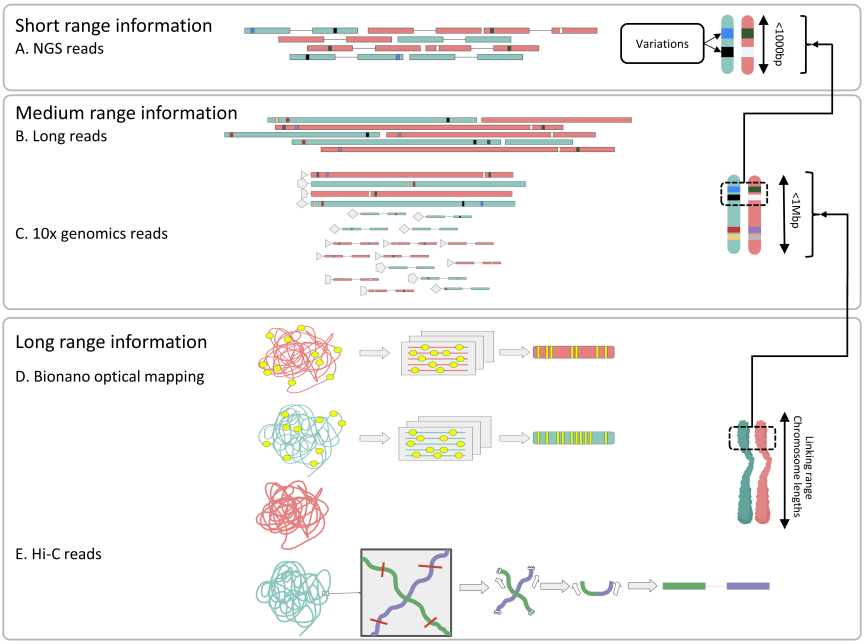


Figure 2.3: A. Paired-end reads are helpful to connect variations that are close together. However, they cannot pass most of the variation deserts due to the small read length and insertion size. B. Long reads might pass small variation deserts and help connect variations closer than the read length. They still cannot pass large variation deserts. C. In 10x Genomics data, molecules are separated and sliced into smaller pieces, and barcodes are attached to the small sequences. The small sequences originated from the same molecule have the same barcode. This barcode attached small sequences are read by Illumina sequencing technology (gray shapes attached to reads). D. With the Bionano optical mapping technique, fluorescence molecules are attached to the chromosome at specific motifs (yellow ovals). Then the Bionano device reads the chromosome through imaging and creates a mapping profile to find the location of motifs. E. Hi-C data are Illumina reads, but each pair (green/purple) is from one side of the folded region of the chromatin. This area is close in the 3D space but might be far away in the linear genome sequence.

Third-generation sequencing methods, including PacBio SMRT and Oxford Nanopore technology, produce longer reads compared to NGS technologies but are often considered

more error-prone, although PacBio HiFi has reported error rate comparable to NGS [48]. In contrast, Oxford Nanopore Technology has introduced ultra-long reads, which produce reads with much higher N50 than the other TGS platforms. Regardless, the longer read length of TGS technologies allow the reads to span some of the repetitive sequences in the genome, yielding more accurate alignments to a reference genome. Thus, third-generation sequencing technologies have been widely used to create more reliable reference genomes, essential for reference-based haplotype reconstruction methods. On top of that, their read length enables connecting variations that NGS technologies could not (see Figure 2.3.B). This allows them to connect genomic regions further apart and create longer haplotype blocks.

The 10x Genomics technology was introduced to generate synthetic long reads by labeling the reads that are generated from a single DNA molecule. It first partitions high molecular weight DNA molecules into droplets, then shears the DNA and attaches droplet-specific 16 letter barcodes. The droplets are then pooled and sequenced using a classic DNA sequencing platform. Interestingly, the same technology can also be used for haplotype reconstruction. Since the reads produced from a droplet, are originated from the same DNA molecule and belong to same haplotype [49]. When reads with the same barcode are covering variation loci, the alleles of these reads at the variation loci can be linked together (see Figure 2.3.C) [50].

Hi-C technology captures the three-dimensional folding of chromosomes to provide long-range information from the genome. The folded areas of chromosomes are close in the 3D space but still can be far away in the chromosome sequence. First, the cell is fixed with formaldehyde which cause the interacting loci bind to one another. Then, restriction enzymes fragment the DNA molecule, but the interacting loci bound remains. Next, a chimeric sequence is created from the two parts of the chromosome interacting. After that, in a typical use case, this chimeric fragment is sequenced with Illumina sequencing machines. The two pairs of the reads are now each from a side of the folded region which might be megabases away in nucleotide-distance. If the two pairs of these reads cover variation loci, they can link the alleles of these two far regions to each other (see Figure 2.3.D) [51, 52].

BioNano optical mapping provides long-range information to complement the DNA sequencing data. It modifies the DNA sequence at specific six bases length motifs and attaches a fluorescence molecule to them. After that, the fluorescence attached DNA sequence is scanned by the BioNano device, which takes image snapshots. Later, these images are matched together, and the motifs' locations in an individual chromosome are detected (see Figure 2.3.E) [53]. These motifs' locations can cover heterozygous loci and can link genomic regions that are far from each other [53]. This long-range linking can complement the sequencing data to make longer haplotype blocks [54–56].

2.4. *De novo* AND REFERENCE-BASED HAPLOTYPE RECONSTRUCTION CHALLENGES

Despite all the improvements in DNA sequencing technologies, haplotype reconstruction remains challenging. In this section, we review five main challenges and how they affect haplotype reconstruction.

2.4.1. POLYPLOIDY

Reconstructing haplotypes of a polyploid genome is more complicated than a diploid genome. In a diploid genome, knowing the sequence of a haplotype allows inference of the other one because the alleles that are not in the already resolved haplotype must be present in the other one. In polyploid genomes, however, knowing the sequence of one haplotype provides little information about the others because they might still have different or identical alleles as the already resolved haplotype [57]. Thus, depending on the ploidy and the level of heterozygosity between the haplotypes, determining alleles is harder.

Furthermore, in a diploid genome, when two reads have identical alleles in a variation locus, they belong to the same haplotype and can be clustered together. However, this is not true for polyploid genomes because they might belong to two different haplotypes that have the same allele at that specific locus. Thus, to determine which haplotype the reads belong to, they should cover sufficient variation loci that differentiate all haplotypes. Therefore, in polyploid genomes fewer reads have sufficient information to get clustered compared to diploid genomes.

2.4.2. VARIATION DESERTS

The variations between the homologous chromosomes are the primary source of information used for reconstructing haplotypes. However, even in a highly heterozygous genome, there are areas where the haplotypes are identical, which we call variation deserts [51]. Reconstructing chromosome-scale haplotypes requires connecting the alleles co-locating at the sides of these variation deserts in a haplotype. This is challenging and requires long-range data that spans the variation deserts. Moreover, even if a variation desert is comparable in size to the read length, only a small subset of the reads would span them. For instance, in the human genome, SNPs occur every 1000 bp on average, and as a result just 1% of short reads will cover two variation sites, which is not enough for chromosome-scale haplotype reconstruction [57]. Third-generation sequencing data might be sufficient to connect the sides of small variation deserts, but chromosome spanning data such as Hi-C is required for the large ones. If the data cannot span the variation desert, the two sides of it remains unlinked, leading into separated and relatively short haplotype blocks.

2.4.3. STRUCTURAL VARIANTS

Structural variants are responsible for the most nucleotide-level diversity between human genomes [58]. These variations are complex and make an immense difference between the chromosomes. Most of the recent haplotype reconstruction tools take a reference-based approach which makes them highly dependent on the alignment of the sequencing data to the reference genome. A structural variant between the target chromosomes and the reference can lead to misalignment of the reads and low-quality haplotype reconstruction [59].

On top of that, most of the current methods only consider the SNVs between the haplotypes for haplotype reconstruction. This restrains them in capturing the structural variations between the haplotypes.

2.4.4. REPETITIVE REGIONS

Repeats are highly similar sequences repeated several times in the genome. The degree of repetitiveness differs in organisms but can go up to 50% in humans [60], and 83% in some plants [61]. The repeats can be either next to or far away from each other, which is termed tandem or interspersed, respectively. Tandem repeats can be as few as two nucleotides repeated many thousands of times [62]. Interspersed repeats can be as far as several million bases away, and constitute up to 34% of the human genome [63]. Repeats create several difficulties for haplotype reconstruction: (i) Some of the sequencing technologies have limitations for handling the complex regions in the genomes, including repeats. For example, ONT has difficulties sequencing homopolymers and determining the exact length [64]. (ii) Alignment methods cannot accurately align the reads originating from the repeat regions to the reference genome which is needed for reference-based haplotype reconstruction. (iii) In order for a read to have useful information for resolving repetitive regions, it has to span the region [65]. This means only a small portion of reads are informative about these regions. (iv) Moreover, repeats between the homologous chromosomes can have slight variations. In the case of the polyploid genome, it is impossible to understand which haplotype the copy of the repeat with the small difference is located.

2.5. SOFTWARE FOR HAPLOTYPE ASSEMBLY

Various computer algorithms facilitate haplotype reconstruction with the data types described in the previous sections (see Table 2.2). Because next-generation sequencing and third-generation sequencing are the most accessible sequencing technologies and are widely used in the literature, the state-of-the-art tools that use these technologies are explained in the main text. Interestingly, all the recent methods for polyploid haplotype reconstruction take a reference-based approach, starting with aligning the read set to the reference and finding the variation loci based on the alignment to begin the algorithm.

Table 2.2: A summary of state-of-the-art haplotype assembly tools and the typical use case of the tools.

Tool name	**Read type**	**SNP based**	**Ploidy**	**Method**	**Evaluation**
HAT [35]	Third-generation sequencing + NGS	Yes	Polyploid	Reference-based	*Saccharomyces pastorianus*, *Brettanomyces bruxellensis*
Whatshap polyphase [66]	Third-generation sequencing	Yes	Polyploid	Reference-based	*Solanum tuberosum* genes
Nphase [36]	Third-generation sequencing + NGS	Yes	Polyploid	Reference-based	*Brettanomyces bruxellensis*, *Solanum tuberosum* chromosome 2
Ranbow [67]	NGS	Yes	Polyploid	Reference-based	*Ipomoea batatas*, *Capsella bursa-pastoris*
HapCut2 [68]	NGS, Third-generation sequencing, 10x Genomics, Hi-C	Yes	Diploid	Reference-based	*Homo sapiens*
ALLHiC [69]	Hi-C	No	Polyploid	Reference-based	*Saccharum spontaneum*, *Saccharum robustum*, *Arachis hypogaea L.*
HAP10 [70]	10x genomics	Yes	Polyploid	Reference-based	*Ipomoea batatas*
Haptree [71]	NGS	Yes	Polyploid	Reference-based	Simulated triploid and tetraploid genomes
SDhaP [72]	NGS, Third-generation sequencing	Yes	Polyploid	Reference-based	HURef [73] and Fosmid [74] dataset
HapCompass [8]	NGS	Yes	Diploid	Reference-based	1000 genomes data [75]
Falcon-unzip [7]	PacBio SMRT	No	Diploid	De novo assembly	Vitis vinifera cv. Cabernet Sauvignon, Clavicorona pyxidata

As explained in the previous sections, haplotype reconstruction in diploid genomes is less challenging than in polyploid genomes, which has enabled diploid haplotype reconstruction tools to have more success. HapCut2, Falcon-unzip, and HapCompass are examples of successful diploid reconstruction tools. HapCut2 aims to optimize the minimum error correction (MEC) to reconstruct haplotypes of diploid genomes. HapCUT2 finds a solution that requires the minimum number of alleles in the reads that need to be changed to be perfectly consistent with one of the two haplotypes. In contrast, HAPCompass takes an entirely different approach and creates a graph called a compass graph. In the compass graph, nodes are SNPs, and two SNPs are connected if there is at least one read that covers both. Each edge is assigned a weight, which is the absolute number of reads supporting the two possible combinations for the two SNPs subtracted from each other. HAPCompass then solves a Minimum weighted edge removal problem and removes the minimum weight of edges in total until the resulting graph has a unique phasing.

In contrast, all polyploid haplotype reconstruction tools take a reference-based approach, which simplifies the polyploid haplotype reconstruction problem, but limit them against structural variants and haplotypes significantly different from the reference. Most recent reference-based polyploid haplotype reconstruction tools take either a read clustering or a seed-and-extend approach. The read clustering approach tries to group the reads based on similarities and dissimilarities at the variation loci. On the other hand, the seed-and-extend approach starts by determining the alleles of variation loci that are close together and then extends these regions based on the read alignments.

SdhaP [72], WhatsHap Polyphase [76], and nPhase [36] are the most recent tools that take a read clustering approach. SDhaP and WhatsHap Polyphase are based on correlation clustering on a graph where reads and if two reads cover the same SNP loci, then there is an edge between them. They are trying to separate the reads into different sets where each set belongs to a haplotype, but they have different approaches for weighting the edges and different heuristics to solve the correlation clustering problem. However, WhatsHap Polyphase has an additional step, "threading," to connect the clusters covering different parts of the chromosome and belonging to the same true haplotype based on three objectives: (i) genotype concordance, (ii) read coverage and (iii) haplotype contiguity. nPhase also aims to separate reads into clusters for each haplotype but takes a different approach. Instead, it assumes that each read is a separate cluster and tries to merge them as much as possible based on the similarities at the SNP loci.

on the other hand, Haptree [71], Ranbow [67], and HAT [35] are taking a seed-and-extend approach. They link the alleles together and try to reconstruct all haplotypes' alleles by linking them together. Haptree starts with phasing the first two SNP loci and finds the most likely haplotypes for the two positions based on the reads. Then it tries to phase the following SNP loci by finding the most probable haplotypes based on the maximum likelihood of the reads and all possible haplotypes. Ranbow, on the other hand, starts by creating seeds, the set of consecutive SNPs covered by reads. Later, it creates a graph between the seeds and connects them based on the reads covering multiple seeds. In the end, it reconstructs the haplotypes by finding paths based on the read coverage and consistency of reads covering multiple seeds. HAT also takes the same approach but takes advantage of both short reads to create the seeds to create a valid starting point for the phasing and expands and merges them with the long reads. One of the main advantages of HAT is

detecting the areas where haplotypes are identical and defining multiplicity blocks before haplotype reconstruction.

2.6. BENCHMARKING SEQUENCING-BASED HAPLOTYPE RECONSTRUCTION METHODS

Recently many tools have been developed to solve the haplotype reconstruction problem [51]. Benchmarking new methods is often done ad hoc on particular datasets, making their results incomparable with the other tools. Even when the methods are compared with each other directly, it is on a particular dataset, which does not allow the generalization of the conclusions on the performance of the methods. Previous a systematic approach was taken to benchmark de novo assembly tools which use NGS and third-generation sequencing data [77, 78]. The same idea should also be applied to haplotype assembly and finding a unified way to benchmark the haplotype assembly methods is crucial.

Here, we propose an experimental design for benchmarking haplotype reconstruction methods. Comparing the reconstructed haplotypes and the actual haplotypes would give much insight into how well the methods perform, but the sequence of actual haplotypes is not available in real datasets. Therefore, we believe benchmarking a haplotype reconstruction method against a simulated dataset with simulated haplotypes and reads is necessary. Moreover, it is critical to test these methods on multiple genomes with different characteristics because some of the methods might only perform well on specific ploidies and heterozygosity levels. Ultimately, the right metrics should be used to evaluate a haplotype reconstruction method and correctly compare it to the simulated haplotypes. Figure 2.4 illustrates the proposed approach to evaluate haplotype reconstruction methods.

2.6.1. SIMULATING SYNTHETIC DATA SETS

We propose four categories of simulated datasets to benchmark a newly developed haplotype assembly method: (i) a highly heterozygous simple genome, (ii) a highly heterozygous complex genome, (iii) a largely homozygous simple genome, and (iv) a largely homozygous complex genome, in ascending order of difficulty. Each of these data sets is needed at multiple ploidies: triploid, tetraploid, and hexaploid which are the most common ploidies (Supplementary Figure A.1). Following suggested protocol leads to 12 total datasets.

The first step of creating each of the mentioned datasets required is simulating the haplotypes. However, to simulate high-quality haplotypes, it is required to analyze several haplotypes resolved genomes and model. Some high-quality diploid genome assemblies are available in the literature, but this is not the case for polyploid genomes, which makes simulating polyploid haplotypes a challenge. Regardless, there are some tools for simulating set of haplotypes based on a reference genome (see Table 2.3), with different ploidies and levels of heterozygosity, which can be used for benchmarking haplotype reconstruction methods. Next, to increase the complexity of the simulated haplotypes structural variants should be added to the haplotypes.

After haplotype simulation, the sequencing data needs to be simulated from the obtained haplotypes. Various tools are available for simulating every kind of sequencing data (see Table 2.3). SimLord [79] and PBSIM [80] are widely used for simulating PacBio reads, whilst NanoSim [81], and DeepSimulator [82, 83] are used to simulate ONT reads, while

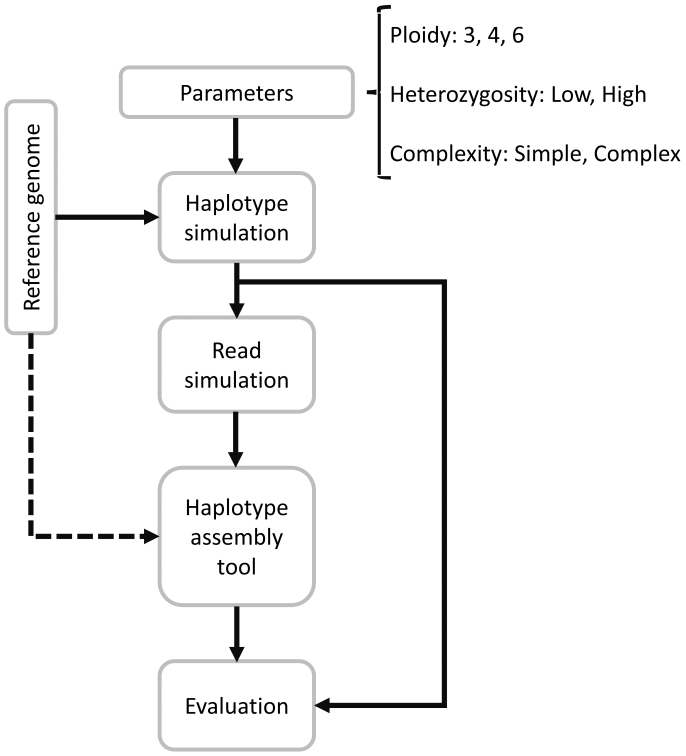


Figure 2.4: Illustrates systematic benchmarking design for haplotype assembly algorithms. Any haplotype assembly algorithm needs to be tested on at least 12 different test cases, based on ploidy, heterozygosity level and complexity of the genome. Here, we define genomes with large repeats and structural variants as 'complex' genomes. First, the haplotypes need to be simulated based on the reference genome and the parameters. Then reads should be simulated based on the simulated haplotypes. Next, the haplotype assembly tool should be used on the sequencing data to reconstruct the haplotypes. The dashed line is indicating the need of reference-based methods to use the reference next to the sequencing data for haplotype reconstruction. Finally, the output of the tool is compared with the simulated haplotypes in the evaluation step to assess the quality of the haplotype reconstruction. The result of this process is a valid assessment of the tool.

LongISLND [86], PBSIM2 [85] and Badread [84] can simulate both types of data. Recently, Badread [84] was introduced, which aims to simulate long error-prone reads. The advantage of Badread over the other tools is that it can produce other kinds of errors like chimeras, adapters, glitches, and "junk DNA". Moreover, Badread does not simulate the long reads based on real datasets; instead, it uses gamma distribution for sampling read lengths, making it less realistic but highly tunable. There are many tools available for simulating NGS reads as well. Merly et al. [98] have compared most of these tools, and the results of the comparison can be used to choose the suitable tool for the simulation. There are fewer tools available for simulating Hi-C and 10x genomics reads, and the most recent ones are listed in Table2.3.

Table 2.3: A summary of simulation tools required for benchmarking a haplotype assembly tool.

Simulation	Tool name	Authors	Release year
Pacbio reads	SimLord [79]	Stöcker et al.	2016
	PBSIM [80]	Ono et al.	2013
Nanopore reads	NanoSim [81]	Yang et al.	2017
	DeepSimulator [82, 83]	Li et al.	2018, 2020
Simulating both Pacbio and ONT reads	Badread [84]	Ryan R Wick	2019
	PBSIM2 [85]	Ono et al.	2021
	LongISLND [86]	Lau et al.	2016
NGS reads	Art [87]	Huang et al.	2012
	MetaSim [88]	Richter et al.	2008
	CuReSim [89]	Caboche et al.	2014
Hi-C Reads	Sim3C [90]	DeMaere et al.	2017
	FreeHi-C [91]	Zheng et al.	2019
10x genomics reads	LRSim [92]	Luo et al.	2017
	Tenxsim [93]	Dong	2018
Haplotype simulation	Aneusim [94]	Van Dijk et al.	2018
	Haplogenerator [95]	Motazadi et al.	2018
	PolyHapSim [96]	Moeinzadeh et al.	2019
Structural variant	VISOR [97]	Bolognini et al.	2019

2.6.2. EVALUATING HAPLOTYPE RECONSTRUCTION METHODS

A non-biased systematic evaluation needs to be performed on the output haplotypes, and several aspects of metrics such as statistical, reference-based, and gene-based should be tested. First, statistical metrics can be used to assess the reconstructed haplotypes independently from the reference genome. In the field genome assembly, the following metrics are often used: N50, NG50, and GC content. These metrics can be used to assess the continuity of the reconstructed haplotypes. Existing software packages such as QUAST [99] can be used to calculate these metrics, for genome assembly.

Second, as the ground truth of the haplotypes is available in the proposed benchmark, reference-based metrics that calculate the difference between the produced haplotype and the ground truth can be used. Common metrics include hamming rate and switch-error rate, which both take advantage of the reference haplotypes and provide valuable information. The hamming rate is the distance of the assembled and reference haplotype at every variation locus. The switch error rate is the rate of required switches between the reconstructed haplotypes to produce the reference haplotypes, where a switch is swapping a set of consecutive alleles between the haplotypes. The hamming rate is a more sensitive metric as it takes every single misscalling of the alleles into account, while a series of misscalled SNVs might be possible to fix by only one switch. In the current state of haplotype reconstruction, the switch error rate has more importance as it led to cutting the haplotypes into blocks and a high hamming rate.

Third, it is important to ensure that all haplotypes contain the critical single-copy orthologs. The Busco [100] tool can check the presence of single-copy orthologs in assemblies. When using a de novo assembly approach, it is crucial to evaluate haplotypes with Busco to make sure all essential genes are present, since de novo methods only use reads

and they are more likely to miss some of the vital genes.

2.7. CONCLUSION

There has been much advancement in the technologies and tools for reconstructing haplotypes. However, some challenges remain. The successful approaches and technologies have their limitations and are not applicable to all use cases.

The chromosome separation approaches have been used to reconstruct human haplotypes successfully, but they are challenging to employ as they require intact cells and are often expensive. Therefore, the reconstruction of haplotypes with other approaches is valuable. The trio-binning approaches have also shown success in reconstructing human and wood tiger moth haplotypes but are inapplicable for most use cases since they require parental information.

As shown in the software for haplotype reconstruction section, most state-of-the-art haplotype reconstruction methods take a reference-based approach. They align the reads to a reference genome to find the alleles, which means a reliable alignment and variant calling have extra importance for these types of methods. Regardless, aligning the reads to complex regions in some genomes is challenging, and the methods which rely upon the alignment will struggle for downstream analysis and haplotype reconstruction. This becomes even more important when variant calling with the long reads is still challenging due to the error rates, and the short reads cannot be appropriately aligned to the complex regions because of their small size. Furthermore, as is indicated in the previous sections, most haplotype reconstruction tools rely on single-nucleotide variations to resolve the haplotypes, as such, they ignore the large indels and structural variants.

Next to limitations of the approaches, every sequencing technology also has limitations, and for telomere-to-telomere haplotype reconstruction it is likely required to combine multiple data types. For example, using Hi-C data, around 20% of the variations that are not close to the restriction enzyme cut site are not detected which limits the usefulness for haplotype reconstruction. Meanwhile, TGS and NGS reads are able to detect most of the variations, however, the small linking range makes them inapplicable for organisms with low heterozygosity or long repeats resulting in haplotypes that divided into several unlinked haplotype blocks. As shown in Table 2.2, many methods have been developed that only use a single data type to reconstruct the haplotypes, yet we do not have any chromosome-scale haplotypes in polyploids. One possible method to overcome these difficulties is using a mixture of data, as they can provide complementary information. Every type of data can fill a piece in this complex puzzle and provides insight to reconstruct the haplotypes.

In terms of existing software tools, all the available state-of-the-art tools for reconstructing haplotypes from sequencing data in polyploid genomes are reference-based. The major risk is that this could lead to a reference bias, and structural variants between the haplotypes will likely be missed.

We believe that future research for solving the haplotype reconstruction problem should focus on de novo approaches, because they do not suffer from reference bias, although they are the most computationally challenging. A future direction could be detecting and linking the variation between the haplotypes without an alignment to the reference and in the assembly graph level.

Finally, we suggest developing several gold standard polyploid datasets with different

ploidy levels, heterozygosity, and structural variants rate, next to a standard set of simulated reads. It is expected for the future that all the tools should use the gold standards and follow the suggested benchmarking methodology shown in the previous sections for evaluation.

2.8. FUNDING

Erin Noel Jordan was supported by the Federal Ministry of Education and Research (BMBF) Germany in the VIPplus programme (03VP06 370).

Lucas R. van Dijk was funded in part with Federal funds from the National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under Grant Number U19AI110818 to the Broad Institute.

2.9. ACKNOWLEDGEMENTS

Figure 2.1, Figure 2.2, and Figure 2.3 were created with BioRender.com.

BIBLIOGRAPHY

[1] L. W. Parfrey, D. J. Lahr, and L. A. Katz. “The dynamic nature of eukaryotic genomes”. In: *Molecular biology and evolution* 25.4 (2008), pp. 787–794.

[2] J. Ramsey and D. W. Schemske. “Pathways, mechanisms, and rates of polyploid formation in flowering plants”. In: *Annual review of ecology and systematics* 29.1 (1998), pp. 467–501.

[3] N. K. Chunduri and Z. Storchová. “The diverse consequences of aneuploidy”. In: *Nature Cell Biology* 21.1 (2019), pp. 54–62.

[4] R.-S. Wang et al. “Haplotype reconstruction from SNP fragments by minimum error correction”. In: *Bioinformatics* 21.10 (2005), pp. 2456–2462.

[5] T. Niu. “Algorithms for inferring haplotypes”. In: *Genetic Epidemiology: The Official Publication of the International Genetic Epidemiology Society* 27.4 (2004), pp. 334–347.

[6] L. Eronen, F. Geerts, and H. Toivonen. “A Markov chain approach to reconstruction of long haplotypes”. In: *Biocomputing 2004*. World Scientific, 2003, pp. 104–115.

[7] X. Zhang et al. “Unzipping haplotypes in diploid and polyploid genomes”. In: *Computational and structural biotechnology journal* 18 (2020), pp. 66–72.

[8] D. Aguiar and S. Istrail. “HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data”. In: *Journal of Computational Biology* 19.6 (2012), pp. 577–590.

[9] S. Majidian, M. H. Kahaei, and D. de Ridder. “Minimum error correction-based haplotype assembly: Considerations for long read data”. In: *Plos one* 15.6 (2020), e0234470.

[10] J. Viruel et al. “A target capture-based method to estimate ploidy from herbarium specimens”. In: *Frontiers in Plant Science* 10 (2019), p. 937.

[11] C. L. Weiß et al. “nQuire: a statistical framework for ploidy estimation using next generation sequencing”. In: *BMC bioinformatics* 19.1 (2018), pp. 1–8.

[12] S. B. Ng et al. “Targeted capture and massively parallel sequencing of 12 human exomes”. In: *Nature* 461.7261 (2009), pp. 272–276.

[13] S. B. Ng et al. “Exome sequencing identifies the cause of a mendelian disorder”. In: *Nature genetics* 42.1 (2010), pp. 30–35.

[14] J. Yang et al. “Haplotype-resolved sweet potato genome traces back its hexaploidization history”. In: *Nature plants* 3.9 (2017), pp. 696–703.

[15] C. A. Saski et al. “Sub genome anchored physical frameworks of the allotetraploid Upland cotton (Gossypium hirsutum L.) genome, and an approach toward reference-grade assemblies of polyploids”. In: *Scientific Reports* 7.1 (2017), p. 15274.

[16] M. Kyriakidou et al. “Current Strategies of Polyploid Plant Genome Sequence Assembly”. In: *Frontiers in Plant Science* 9 (Nov. 2018), p. 1660. DOI: 10.3389/fpls.2018.01660.

[17] I. Bancroft et al. “Dissecting the genome of the polyploid crop oilseed rape by transcriptome sequencing”. In: *Nature biotechnology* 29.8 (2011), pp. 762–766.

[18] N. Jannoo et al. “Orthologous comparison in a gene-rich region among grasses reveals stability in the sugarcane polyploid genome”. In: *The Plant Journal* 50.4 (2007), pp. 574–585.

[19] J. A. Bhat et al. “Features and applications of haplotypes in crop breeding”. In: *Communications biology* 4.1 (2021), p. 1266.

[20] R. K. Varshney et al. “Designing future crops: genomics-assisted breeding comes of age”. In: *Trends in Plant Science* 26.6 (2021), pp. 631–649.

[21] J. Brinton et al. “A haplotype-led approach to increase the precision of wheat breeding”. In: *Communications Biology* 3.1 (2020), p. 712.

[22] L. A. Frame, E. Costa, and S. A. Jackson. “Current explorations of nutrition and the gut microbiome: a comprehensive evaluation of the review literature”. In: *Nutrition Reviews* 78.10 (2020), pp. 798–812.

[23] R. Daniel. “The metagenomics of soil”. In: *Nature reviews microbiology* 3.6 (2005), pp. 470–478.

[24] C. Rinke et al. “Insights into the phylogeny and coding potential of microbial dark matter”. In: *Nature* 499.7459 (2013), pp. 431–437.

[25] C. Simon and R. Daniel. “Metagenomic analyses: past and future trends”. In: *Applied and environmental microbiology* 77.4 (2011), pp. 1153–1161.

[26] J. Vollmers, S. Wiegand, and A.-K. Kaster. “Comparing and evaluating metagenome assembly tools from a microbiologist’s perspective-not only size matters!” In: *PloS one* 12.1 (2017), e0169662.

[27] C. Anyansi et al. “Computational methods for strain-level microbial detection in colony and metagenome sequencing data”. In: *Frontiers in Microbiology* 11 (2020), p. 1925.

[28] C. Anyansi et al. “QuantTB–a method to classify mixed Mycobacterium tuberculosis infections within whole genome sequencing data”. In: *BMC genomics* 21 (2020), pp. 1–16.

[29] S. Knyazev et al. “Epidemiological data analysis of viral quasispecies in the next-generation sequencing era”. In: *Briefings in bioinformatics* 22.1 (2021), pp. 96–108.

[30] E. Domingo et al. “Viral quasispecies: dynamics, interactions, and pathogenesis”. In: *Origin and evolution of viruses* (2008), pp. 87–118.

[31] S. Dávila-Ramos et al. “A review on viral metagenomics in extreme environments”. In: *Frontiers in microbiology* 10 (2019), p. 2403.

[32] S. Garg et al. “Chromosome-scale, haplotype-resolved assembly of human genomes”. In: *Nature biotechnology* 39.3 (2021), pp. 309–312.

[33] S. Prabhakaran et al. “HIV haplotype inference using a propagating dirichlet process mixture model”. In: *IEEE/ACM transactions on computational biology and bioinformatics* 11.1 (2013), pp. 182–191.

[34] C.-S. Chin et al. “Phased diploid genome assembly with single-molecule real-time sequencing”. In: *Nature methods* 13.12 (2016), pp. 1050–1054.

[35] R. Shirali Hossein Zade et al. “HAT: haplotype assembly tool using short and error-prone long reads”. In: *Bioinformatics* 38.24 (2022), pp. 5352–5359.

[36] O. Abou Saada et al. “nPhase: an accurate and contiguous phasing method for polyploids”. In: *Genome Biology* 22.1 (2021), pp. 1–27.

[37] S. Garg et al. *A haplotype-aware de novo assembly of related individuals using pedigree graph*. Tech. rep. 2019, p. 580159.

[38] E. C. Yen et al. “A haplotype-resolved, de novo genome assembly for the wood tiger moth (Arctia plantaginis) through trio binning”. In: *GigaScience* 9.8 (2020), giaa088.

[39] M. W. Snyder et al. “Haplotype-resolved genome sequencing: experimental methods and applications”. In: *Nature Reviews Genetics* 16.6 (2015), pp. 344–358.

[40] D. Porubsky et al. “Fully phased human genome assembly without parental data using single-cell strand sequencing and long reads”. In: *Nature biotechnology* 39.3 (2021), pp. 302–308.

[41] E. R. Mardis. “DNA sequencing technologies: 2006–2016”. In: *Nature protocols* 12.2 (2017), pp. 213–218.

[42] T. Hu et al. “Next-generation sequencing technologies: An overview”. In: *Human Immunology* 82.11 (2021), pp. 801–811.

[43] G. Zheng et al. “Massively parallel digital transcriptional profiling of single cells. Nat Commun 8: 14049”. In: *Data Set5. Putative transcription factors binding motifs identified for genes in trans-eQTL (expression quantitative trait loci) hotspots Data Set6. Putative master regulators in the trans-eQTL (expression quantitative trait loci) hotspots Figure S* 1 (2017).

[44] N. L. Van Berkum et al. “Hi-C: a method to study the three-dimensional architecture of genomes.” In: *JoVE (Journal of Visualized Experiments)* 39 (2010), e1869.

[45] J. Henson, G. Tischler, and Z. Ning. “Next-generation sequencing and large genome assemblies”. In: *Pharmacogenomics* 13.8 (2012), pp. 901–915.

[46] D. Porubskỳ et al. “Direct chromosome-length haplotyping by single-cell sequencing”. In: *Genome research* 26.11 (2016), pp. 1565–1574.

[47] F. S. Turner. “Assessment of insert sizes and adapter content in fastq data from NexteraXT libraries”. In: *Frontiers in genetics* 5 (2014), p. 77572.

[48] A. Abbasi and L. B. Alexandrov. “Significance and limitations of the use of next-generation sequencing technologies for detecting mutational signatures”. In: *DNA repair* 107 (2021), p. 103200.

[49] G. X. Zheng et al. “Haplotyping germline and cancer genomes with high-throughput linked-read sequencing”. In: *Nature biotechnology* 34.3 (2016), pp. 303–311.

[50] D. Redin et al. “High throughput barcoding method for genome-scale phasing”. In: *Scientific reports* 9.1 (2019), p. 18116.

[51] S. Garg. “Computational methods for chromosome-scale haplotype reconstruction”. In: *Genome biology* 22.1 (2021), pp. 1–24.

[52] Z. N. Kronenberg et al. “Extended haplotype-phasing of long-read de novo genome assemblies using Hi-C”. In: *Nature communications* 12.1 (2021), p. 1935.

[53] Y. Yuan, C. Y.-L. Chung, and T.-F. Chan. “Advances in optical mapping for genomic research”. In: *Computational and Structural Biotechnology Journal* 18 (2020), pp. 2051–2062.

[54] Y. Yang et al. *Chromosome-scale de novo assembly and phasing of a Chinese indigenous pig genome*. Tech. rep. 2019, p. 770958.

[55] W. Y. Low et al. “Haplotype-resolved cattle genomes provide insights into structural variation and adaptation”. In: *BioRxiv* (2019), p. 720797.

[56] S. Ananthasayanam et al. *First near complete haplotype phased genome assembly of River buffalo (Bubalus bubalis)*. Tech. rep. 2019, p. 618785.

[57] M. Moeinzadeh. “De novo and haplotype assembly of polyploid genomes”. PhD thesis. 2019.

[58] P. H. Sudmant et al. “An integrated map of structural variation in 2,504 human genomes”. In: *Nature* 526.7571 (2015), pp. 75–81.

[59] M. Mahmoud et al. “Structural variant calling: the long and the short of it”. In: *Genome biology* 20.1 (2019), pp. 1–14.

[60] B. Haubold and T. Wiehe. “How repetitive are genomes?” In: *BMC bioinformatics* 7.1 (2006), pp. 1–10.

[61] J. Macas et al. “In depth characterization of repetitive DNA in 23 plant genomes reveals sources of genome size variation in the legume tribe Fabeae”. In: *PloS one* 10.11 (2015), e0143424.

[62] T. J. Treangen and S. L. Salzberg. “Repetitive DNA and next-generation sequencing: computational challenges and solutions”. In: *Nature Reviews Genetics* 13.1 (2012), pp. 36–46.

[63] I. H. G. S. Consortium. “Initial sequencing and analysis of the human genome”. In: *nature* 409.6822 (2001), pp. 860–921.

[64] C. Delahaye and J. Nicolas. “Sequencing DNA with nanopores: Troubles and biases”. In: *PloS one* 16.10 (2021), e0257521.

[65] T. P. Michael and R. VanBuren. “Building near-complete plant genomes”. In: *Current Opinion in Plant Biology* 54 (2020), pp. 26–33.

[66] S. D. Schrinner et al. “Haplotype threading: accurate polyploid phasing from long reads”. In: *Genome biology* 21.1 (2020), pp. 1–22.

[67] M.-H. Moeinzadeh et al. “Ranbow: a fast and accurate method for polyploid haplotype reconstruction”. In: *PLOS Computational Biology* 16.5 (2020), e1007843.

[68] P. Edge, V. Bafna, and V. Bansal. “HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies”. In: *Genome research* 27.5 (2017), pp. 801–812.

[69] X. Zhang et al. “Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data”. In: *Nature plants* 5.8 (2019), pp. 833–845.

[70] S. Majidian, M. H. Kahaei, and D. De Ridder. “Hap10: reconstructing accurate and long polyploid haplotypes using linked reads”. In: *BMC bioinformatics* 21.1 (2020), pp. 1–18.

[71] E. Berger et al. “HapTree: a novel Bayesian framework for single individual polyplotyping using NGS data”. In: *PLoS computational biology* 10.3 (2014), e1003502.

[72] S. Das and H. Vikalo. “SDhaP: haplotype assembly for diploids and polyploids via semi-definite programming”. In: *BMC genomics* 16 (2015), pp. 1–16.

[73] S. Levy et al. “The diploid genome sequence of an individual human”. In: *PLoS biology* 5.10 (2007), e254.

[74] J. Duitama et al. “Fosmid-based whole genome haplotyping of a HapMap trio child: evaluation of Single Individual Haplotyping techniques”. In: *Nucleic acids research* 40.5 (2012), pp. 2041–2053.

[75] 1. G. P. Consortium et al. “A map of human genome variation from population scale sequencing”. In: *Nature* 467.7319 (2010), p. 1061.

[76] M. Martin et al. *WhatsHap: fast and accurate read-based phasing*. Tech. rep. 2016, p. 085050.

[77] D. Earl et al. “Assemblathon 1: a competitive assessment of de novo short read assembly methods”. In: *Genome research* 21.12 (2011), pp. 2224–2241.

[78] K. R. Bradnam et al. “Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species”. In: *Gigascience* 2.1 (2013), pp. 2047–217X.

[79] B. K. Stöcker, J. Köster, and S. Rahmann. “SimLoRD: simulation of long read data”. In: *Bioinformatics* 32.17 (2016), pp. 2704–2706.

[80] Y. Ono, K. Asai, and M. Hamada. “PBSIM: PacBio reads simulator—toward accurate genome assembly”. In: *Bioinformatics* 29.1 (2013), pp. 119–121.

[81] C. Yang et al. “NanoSim: nanopore sequence read simulator based on statistical characterization”. In: *GigaScience* 6.4 (2017), gix010.

[82] Y. Li et al. “DeepSimulator: a deep simulator for Nanopore sequencing”. In: *Bioinformatics* 34.17 (2018), pp. 2899–2908.

[83] Y. Li et al. “DeepSimulator1. 5: a more powerful, quicker and lighter simulator for Nanopore sequencing”. In: *Bioinformatics* 36.8 (2020), pp. 2578–2580.

[84] R. R. Wick. “Badread: simulation of error-prone long reads”. In: *Journal of Open Source Software* 4.36 (2019), p. 1316.

[85] Y. Ono, K. Asai, and M. Hamada. “PBSIM2: a simulator for long-read sequencers with a novel generative model of quality scores”. In: *Bioinformatics* 37.5 (2021), pp. 589–595.

[86] B. Lau et al. “LongISLND: in silico sequencing of lengthy and noisy datatypes”. In: *Bioinformatics* 32.24 (2016), pp. 3829–3832.

[87] W. Huang et al. “ART: a next-generation sequencing read simulator”. In: *Bioinformatics* 28.4 (2012), pp. 593–594.

[88] D. C. Richter et al. “MetaSim—a sequencing simulator for genomics and metagenomics”. In: *PloS one* 3.10 (2008), e3373.

[89] S. Caboche et al. “Comparison of mapping algorithms used in high-throughput sequencing: application to Ion Torrent data”. In: *BMC genomics* 15.1 (2014), pp. 1–16.

[90] M. Z. DeMaere and A. E. Darling. “Sim3C: simulation of Hi-C and Meta3C proximity ligation sequencing technologies”. In: *GigaScience* 7.2 (2018), gix103.

[91] Y. Zheng and S. Keleş. “FreeHi-C: high fidelity Hi-C data simulation for benchmarking and data augmentation”. In: *bioRxiv* (2019), p. 629923.

[92] R. Luo et al. “LRSim: a linked-reads simulator generating insights for better genome partitioning”. In: *Computational and structural biotechnology journal* 15 (2017), pp. 478–484.

[93] G. Dong. “Tenxsim: Simulator for Pure and Heterogeneous Genomic Sequence with 10X Genomics”. In: (2018).

[94] L. van Dijk. *aneusim - A tool to generate synthetic aneuploid/polyploid genomes*. original-date: 2017-07-13. Oct. 2018. URL: `https://github.com/AbeelLab/aneusim` (visited on 11/09/2021).

[95] E. Motazedi et al. “Exploiting next-generation sequencing to solve the haplotyping puzzle in polyploids: a simulation study”. In: *Briefings in bioinformatics* 19.3 (2018), pp. 387–403.

[96] moeinzadeh. *PolyHapSim: A haplotype simulator for polyploid genomes*. original-date: 2020-01-29. Jan. 2020. URL: `https://github.com/moeinzadeh/PolyHapSim` (visited on 11/09/2021).

[97] D. Bolognini et al. “VISOR: a versatile haplotype-aware structural variant simulator for short-and long-read sequencing”. In: *Bioinformatics* 36.4 (2020), pp. 1267–1269.

[98] M. Escalona, S. Rocha, and D. Posada. “A comparison of tools for the simulation of genomic next-generation sequencing data”. In: *Nature Reviews Genetics* 17.8 (2016), pp. 459–469.

[99] A. Gurevich et al. “QUAST: quality assessment tool for genome assemblies”. In: *Bioinformatics* 29.8 (2013), pp. 1072–1075.

[100] F. A. Simão et al. “BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs”. In: *Bioinformatics* 31.19 (2015), pp. 3210–3212.

3

EVALUATING LONG READ DE NOVO ASSEMBLY TOOLS FOR EUKARYOTIC GENOMES: INSIGHTS AND CONSIDERATIONS

Assembly algorithm choice should be a deliberate, well-justified decision when researchers create genome assemblies for eukaryotic organisms from third-generation sequencing technologies. While third-generation sequencing by Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PacBio) have overcome the disadvantages of short read lengths specific to next-generation sequencing (NGS), third-generation sequencers are known to produce more error-prone reads, thereby generating a new set of challenges for assembly algorithms and pipelines. However, the introduction of HiFi reads, which offer substantially reduced error rates, has provided a promising solution for more accurate assembly outcomes. Since the introduction of third-generation sequencing technologies, many tools have been developed that aim to take advantage of the longer reads, and researchers need to choose the correct assembler for their projects.

We benchmarked state-of-the-art long-read de novo assemblers, to help readers make a balanced choice for the assembly of eukaryotes. To this end, we used 12 real and 64 simulated datasets from different eukaryotic genomes, with different read length distributions, imitating PacBio CLR, PacBio HiFi, and ONT sequencing to evaluate the assemblers. We include five commonly used long read assemblers in our benchmark: Canu, Flye, Miniasm, Raven, and wtdbg2 for ONT and PacBio CLR reads. For PacBio HiFi reads LJA, we include five state-of-the-art HiFi assemblers: HiCanu, Flye, Hifiasm, LJA and MBG. Evaluation categories address the following metrics: reference-based metrics, assembly statistics, misassembly count, BUSCO completeness, runtime, and RAM usage. Additionally, we investigated the effect of increased read length on the quality of the assemblies, and report that read length can, but does not always, positively impact assembly quality

This chapter has been published in GigaSciences,
`https://doi.org/10.1093/gigascience/giad100`

Our benchmark concludes that there is no assembler that performs the best in all the evaluation categories. However, our results show that overall Flye is the best-performing assembler for PacBio CLR and ONT reads, both on real and simulated data. Meanwhile, best performing PacBio HiFi assemblers are Hifiasm, and LJA. Next, the benchmarking using longer reads shows that the increased read length improves assembly quality, but the extent to which that can be achieved depends on the size and complexity of the reference genome

.

3.1. INTRODUCTION

De novo genome assembly is essential in several leading fields of research, including disease identification, gene identification, and evolutionary biology [1–4]. Unlike reference-based assembly, which relies on the use of a reference genome, de novo assembly only uses the genomic information contained within the sequenced reads. Since it is not constrained to the use of a reference, high quality de novo assembly is essential for studying novel organisms, as well as for the discovery of overlooked genomic features, such as gene duplication [5], in previously assembled genomes.

The introduction of Third Generation Sequencing (TGS) led to massive improvements in de novo assembly. The advent of TGS has addressed the main drawback of Next Generation Sequencing (NGS) platforms, namely the short read length, but has introduced new challenges in genome assembly, because of the higher error rates of long reads. The leading platforms in long-read sequencing are Pacific Biosciences Single Molecule, Real-Time sequencing (often abbreviated as "PacBio") and Oxford Nanopore (ONT) sequencing [6].

Since the introduction of TGS platforms, many methods have been developed that aim to take the most benefits from the longer read length and overcome the new challenges due to sequencing error. Recent studies have been conducted to compare long-read de novo assemblers. One such study was conducted by Wick and Holt [7], who focused on long-read de novo assembly of prokaryotic genomes. Eight assemblers were tested on real and simulated reads from PacBio and ONT sequencing, and evaluation metrics included sequence identities, circularisation of contigs, computational resources, as well as accuracy. Murigneux et al. [8] performed similar experiments on the genome of M. jansenii, although in this case, the focus was on comparatively benchmarking Illumina sequencing and three long-read sequencing technologies, in addition to the comparison of long-read assembly tools. Studies narrowed down to just one type of sequencing technology include those of Jung et al. [9], who evaluated assemblers on real PacBio reads from five plant genomes, and Chen et al. [10], who used Oxford Nanopore real and simulated reads from bacterial pathogens in their comparison. Except for the Wick and Holt study, which provides a compressive comparison on de novo assembly of prokaryotic genomes, other studies are either comparing the assemblers on single genome or using data from a single sequencing platform. Here, we provide a comprehensive comparison on de novo assembly tools on the most used TGS technologies and 7 different eukaryotic genomes, to complement the study of Wick and Holt.

In this study, we are benchmarking these methods using 12 real and 64 simulated datasets (see Figure 3.1) from PacBio CLR, PacBio HiFi and ONT platforms to guide researchers to choose the proper assembler for their studies. Benchmarking using simulated reads allows us to accurately compare the final assembly with the ground truth, and benchmarking using the real reads can validate the results based on simulated reads. The assembler comparison presented in this manuscript complements the literature that has already been published, by introducing an analysis of not just assembler performance, but also of the effect of read length on assembly quality. Although increased read length is considered an advantage, we investigate if it is always a necessary advantage to have for assembly performance. To that end, the scope of the study extends to six model eukaryotes that provide a performance indication for genomes of variable complexity, covering a wide range of taxa on the eukaryotic branch of the Tree of Life [11]. Complexity in genome

assembly is determined by multiple variables, the most notable of which is the proportion of repetitive sequences within the genome of a particular organism. Complexity in eukaryotic genomes is further exacerbated by size and organization of chromosomal architecture, including telomeres and centromeres, and the presence of circular elements such as mitochondrial and chloroplast DNA.

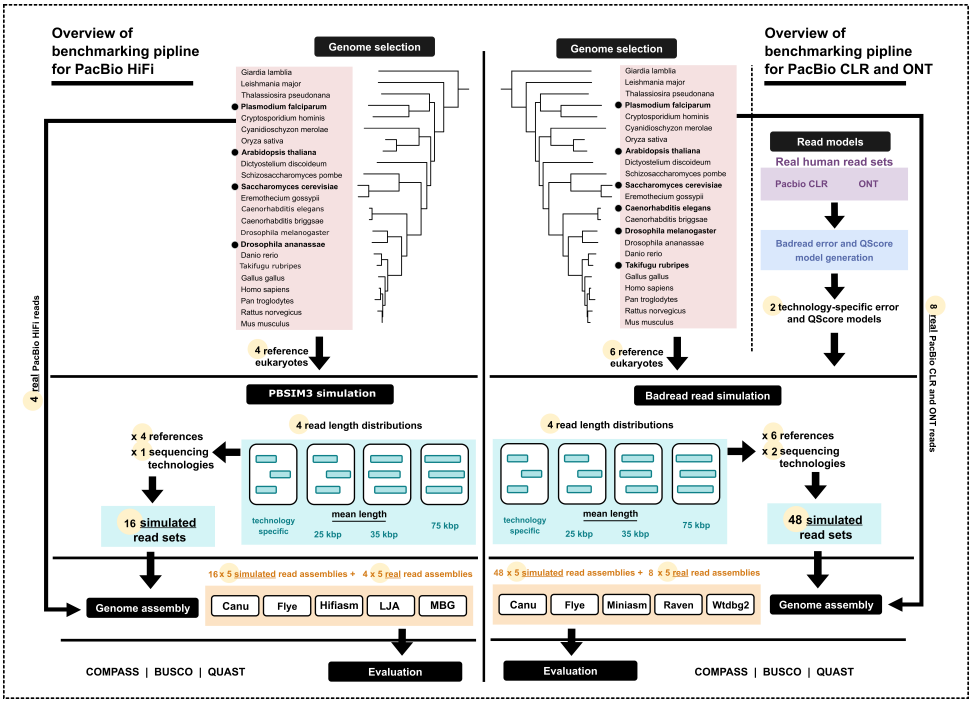


Figure 3.1: The benchmarking pipeline. For PacBio CLR and ONT (right panel), first we select 6 representative eukaryotes from the Tree of Life [11] and use Badread's [12] error and Qscore model generation feature to create 2 models of PacBio CLR and ONT long sequencing technologies. This is input to the read simulation stage, where we simulate reads from all genomes, with four different read length distributions. We then perform assembly of simulated and real reads, using five long-read assemblers. For PacBio HiFi (left panel), first we select 4 representative eukaryotes and use PBSIM3 to simulate HiFi reads. These reads are then assembled using five state-of-the-art HiFi assemblers. Lastly, we evaluate all PacBio HiFi, PacBio CLR and ONT assemblies based on several criteria.

De novo genome assembly evaluation remains challenging, as it represents a process that must account for variables such as the goal of an assembly and the existence of a ground-truth reference. A standard evaluation procedure was introduced in the literature by the two Assemblathon competitions [13, 14], which outlined a selection of metrics that encompasses the most relevant aspects of genome assembly, however, these metrics require a reference sequence. Most of these metrics are adopted in our benchmark.

Consequently, this study addresses two main objectives. First, we provide a systematic comparison of state-of-the-art long-read assembly tools, documenting their performance in assembling real and simulated PacBio Continuous Long (CLRs), PacBio High fidelity (HiFi), and Oxford Nanopore (ONT) reads on a diverse set of eukaryotic organisms. The

PacBio CLR and ONT reads generated from the genomes of *S. cerevisiae*, *P. falciparum*, *C. elegans*, *A. thaliana*, *D. melanogaster*, and *T. rubripes* and the PacBio HiFi reads are generated from the genomes of *S. cerevisiae*, *P. falciparum*, *A. thaliana* and *D. ananassae*. Our second objective is to investigate whether increased read length has a positive effect on overall assembly quality, given that increasing the length of reads is an on-going effort in the development of Third Generation Sequencing platforms [15].

It is important to note that our objective is to evaluate the performance of these tools in generating a consensus assembly without taking haplotypes into account. Moreover, it is crucial to highlight that the results and conclusions drawn from this comparison may not be directly applicable to metagenome assembly. The unique characteristics and complexities associated with metagenomic data warrant a separate and distinct analysis, which is beyond the scope of this study.

3.2. MATERIALS AND METHODS

3.2.1. DATA

In this study, we are using real and simulated data from various organisms to benchmark long read de novo assembly tools.

REFERENCE GENOMES

We selected six reference genomes from eukaryotic organisms represented in the Interactive Tree Of Life (iTOL) v6 [11] for evaluating PacBio CLR and ONT assemblers: *S. cerevisiae* (strain S288C), *P. falciparum* (isolate 3D7), C. elegans (strain VC2010), *A. thaliana* (ecotype Col-0), *D. melanogaster* (strain ISO-1), and *T. rubripes*. Moreover, we selected the four eukaryotic organisms to evaluate PacBio HiFi assemblers: *S. cerevisiae* (strain S288C), *P. falciparum* (isolate 3D7), *A. thaliana* (ecotype Col-0), and *D. ananassae* (strain 14024-0371.13). Assembly accessions are included in Supplementary Table S1 in [16].

The reference assemblies for *C. elegans*, *D. melanogaster*, and *T. rubripes* included uncalled bases. In these cases, before read simulation, each base N was replaced with base A, as done by Wick and Holt [7]. This avoids ambiguity in the read simulation process and consequently simplifies the evaluation of the simulated-read assemblies. As such, we used this modified version as a reference when evaluating all assemblies of simulated reads from these four genomes. In the evaluation of real-read assemblies, the original assemblies were used as references.

SIMULATED READS

The PacBio CLR and ONT simulated read sets were generated using Badread v0.2.0 [12]. To create read error and Qscore (quality score) models in addition to the simulator's own default models, Badread requires the following three parameters: a set of real reads, a high-quality reference genome, and an alignment file, obtained by aligning the reads to the reference genome. We used real read sets from the human genome to create error and Qscore models that reflect the state-of-the-art for PacBio Continuous Long Reads (CLRs), and Oxford Nanopore reads. The simulated PacBio HiFi reads were generated using PBSIM3. To generate reads similar to HiFi, we used –num-pass 10 parameter, and then applied ccs version 6.4.0 to generate the consensus reads.

To create the models for PacBio CLR and Oxford Nanopore reads, we used the real read sets sequenced from the human genome and aligned to the latest high-quality human genome reference assembled by [17]: assembly T2T-CHM13v2.0, with RefSeq accession GCF_009914755.1. The alignment was performed using Minimap2 v2.24 [18] with default parameters. The sources for these sequencing data are outlined in Supplementary Table S2 in [16], as well as the read identities for each technology, which are later passed as parameters for the simulation stage.

To study the effect of read length on genome assembly, we simulated reads that imitate PacBio CLR, PacBio HiFi, and Oxford Nanopore sequencing, with four different read length distributions, using Badread for PacBio CLR and Oxford Nanopore sequencing while using PBSIM3 for PacBio HiFi. The first read simulation represents the current state of the three long-read technologies. The other three simulations reflect data points in-between technology-specific values and ultra-long reads, data points of a similar length as ultra-long-reads, and longer than ultra-long reads. We need to define the mean and standard deviation of the read length distributions for these simulations. The values for the mean and standard deviation of these distributions were selected as follows. First, we calculated the read length distributions of the real read sets in Supplementary Table S2 in [16] and simulated an initial iteration of reads using these technology-specific values. For choosing these values for the other three iterations, we analysed a set of Oxford Nanopore Ultra-Long reads used in the latest assembly of the human genome [17]. We selected GridION run SRR12564452, available as sequence data in BioProject PRJNA559484, with a mean read length of approximately 35.7 kbp, and a standard deviation of 42.5 kbp. A summary of the Badread and PBSIM3 commands used in our simulation can be found in Supplementary Table S3 and S4 in [16].

A full overview of the mean and standard deviation of all four read length distributions is given in Table 3.1. Note that, for each of the technologies, the standard deviation for the last three distributions was derived from the mean, using the ratio between the mean and standard deviation reflected by the technology-specific values. Hence, for the last three iterations, the mean read length is consistent across sequencing technologies, but the standard deviation varies.

Table 3.1: The mean and standard deviation describing the read length distributions used in our simulations. Note that read length increases with each iteration, and the distribution parameters are different for each technology.

	Read length distribution parameters (kbp), per technology					
	PacBio CLR		PacBio HiFi		Oxford Nanopore	
	Mean	Stdev	Mean	Stdev	Mean	Stdev
Iteration 1 (technology-specific values)	15.7	14.4	20.7	2.5	12.1	17.1
Iteration 2	25	22.5	25	3	25	35
Iteration 3 (imitate ultra-long reads)	35	31.5	35	4.2	35	49
Iteration 4	75	67.5	75	9	75	105

Consequently, we ran the simulations for each reference genome. As described above,

we used our own models for each technology, and passed them to the simulator as the –error_model and –qscore_model. The read identities per technology were set to the values included in Supplementary Table S3 in [16]. Across all simulations, we chose a coverage depth of 30x. Canu's documentation [19] specifies a minimum coverage of 20 – 25x for HiFi data, and 20x for other types of data, while Flye's guidelines [20] indicate a minimum coverage of 30x. As there is no minimum recommended coverage indicated for the other assemblers we used in our benchmark, we simulated reads following the stricter guideline among these two, that is, 30x coverage.

REAL READS

In support of our evaluation on simulated reads, we also performed a benchmark on real-read assemblies from Oxford Nanopore and PacBio reads sequenced from the reference genomes. These reads were sampled to approximately 30x coverage, to avoid introducing potentially confounding variables when comparing assemblies of real and simulated datasets. The data sources for all real sets are included in Supplementary Table S5 in [16]. Please note that the PacBio CLR data from C. elegans was generated using the older RSII technology. These reads inherent characteristics of the RSII system, such as shorter average reads and a higher error rate, which might have influenced the assembly results.

3.2.2. ASSEMBLIES

For the PacBio CLR and ONT reads, we included the following five long-read de novo assemblers: Canu v2.2 [19], Flye v2.9 [20], Wtdbg2 (also known as Redbean) v2.5 [21], Raven v1.7.0 [22], and Miniasm v0.3_r179 [23]. For the PacBio HiFi reads, we included HiCanu v2.2 [24], Flye v2.9, Hifiasm 0.19.5-r587 [25], LJA v 0.2 [26], and MBG v 1.0.14 [27]. We used the most recent releases of the assemblers at the time we started this study.

The assemblies were performed with default values for most parameters. Canu and Wtdbg2 require the estimated genome size as a parameter, and we set the following values: S. cerevisiae = 12 Mbp, P. falciparum = 23 Mbp, A. thaliana = 135 Mbp, D. melanogaster = 139 Mbp, C. elegans = 103 Mbp, and T. rubripes = 384 Mbp, D. ananassae = 217 Mbp. All commands used in the assembly pipelines are available in Supplementary Table S6 in [16]. We note that further polishing of assemblies using high-fidelity short reads, although common in practice [28–30], is omitted in this study, as the focus is exclusively on assembler performance on long-read data and not polishing tools. We added a long-read polishing step for Miniasm and Wtdbg2, as their assembly pipelines do not include long-read based polishing. Following Raven's default pipeline, which performs two rounds of Racon polishing [31], we used two rounds of Racon polishing on Wtdbg2 and Miniasm. We note that for Miniasm, we used Minipolish [7], which simplifies Racon polishing by applying it in two iterations on the GFA (Graphical Fragment Assembly) files produced by the assembler. For both Miniasm and Wtdbg2, the alignments required for polishing were generated with Minimap v2.24.

3.2.3. EVALUATION

We evaluated the assemblies in three different categories of metrics. The COMPASS analysis compares the assemblies with their corresponding reference genome and provides insight into their similarities. The assembly statistics provide some basic knowledge about

the contiguity and misassemblies. Finally, the BUSCO assessment investigates the presence of essential genes in the assemblies. These three categories of metrics, next to each other, can provide a complete overview of the assembly's quality.

CORRECTNESS ANALYSIS

For each assembly, we ran the COMPASS script to measure the coverage, validity, multiplicity, and parsimony, to assess the quality of the assemblies, as defined in Assemblathon 2 [14]. These metrics describe several characteristics that were deemed important for comparing de novo assembly tools, and they were computed using three types of data: (1) the reference sequence, (2) the assembled scaffolds, and (3) the alignments (sequences from the assembled scaffolds that were aligned to the reference sequences). Definitions and formulas for the metrics are reported in Supplementary Table S7 in [16].

Moreover, we use QUAST v5.2.0 [32] to calculate the number of misassemblies. QUAST identifies misassemblies based on the definition outlined by [33]. The total number of misassemblies is the sum of all relocations, inversions, and translocations. Considering two adjacent flanking sequences, if they both align to the same chromosome, but 1 kbp away from each other, or overlapping for more than 1 kbp, this is counted as a relocation. If these flanking sequences, aligned to the same chromosome, are on opposite strands, the misassembly is considered an inversion. Lastly, translocations describe events in which two flanking sequences align to different chromosomes.

CONTIGUITY ASSESSMENT

We use QUAST v5.2.0 [32] to measure the auNGA of an assembly. The auNGA metric, standing for the area under the NGAx curve [13], is a measure of assembly contiguity. By calculating the area beneath this profile, which integrates the aligned sequence fragment or contig lengths at various percentage thresholds, it provides a more thorough understanding of the contiguity of the assembly compared to single-value metrics. A larger auNGA value indicates better contiguity in the genome assembly.

COMPLETENESS ASSESSMENT

BUSCO v5.4.2 assessment [34, 35] is performed to evaluate the completeness of the essential genes in the assemblies. This quantifies the number of single-copy, duplicated, fragmented and missing orthologs in an assembled genome. From the number of orthologs specific to each dataset, BUSCO identifies how many orthologs are present in the assembly (either as single-copy or duplicated), how many are fragmented, and how many are missing. We ran these evaluations with a different OrthoDB lineage dataset for each genome: *S. cerevisiae* – saccharomycetes, *P. falciparum* – plasmodium, *A. thaliana* – brassicales, *D. melanogaster* – diptera, *C. elegans* – nematoda, *T. rubripes* – ctinopterygii, and . *ananassae* – diptera.

3.3. RESULTS AND DISCUSSION

3.3.1. OVERVIEW OF THE BENCHMARKING PIPELINE

Figure 3.1 shows an overview of the benchmarking pipeline. For the PacBio CLR and Oxford Nanopore reads we begin with the selection of six representative eukaryotes from the interactive Tree of Life [11]: *Saccharomyces cerevisiae*, *Plasmodium falciparum*,

Arabidopsis thaliana, *Drosophila melanogaster*, *Caenorhabditis elegans*, and *Takifugu rubripes*. We also use three read sets from the latest human assembly project [17] to generate Badread error and Qscore models [12] for PacBio Continuous Long Reads (CLRs), and Oxford Nanopore reads (see Supplementary Table S2 in [16]). The reference sequences and models become input to the Badread simulation stage. For each genome, we simulate reads with four different read length distributions and two sequencing technologies (see Table 3.1), amounting to a total of 8 simulated read sets per reference genome. These reads, as well as real read sets, are assembled with five assembly tools: Canu, Flye, Miniasm, Raven, and Wtdbg2.

For the PacBio HiFi reads we begin with the reference genome of the 4 selected eukaryote species: *Saccharomyces cerevisiae*, *Plasmodium falciparum*, *Arabidopsis thaliana*, and *Drosophila ananassae*. Then we use PBSIM3 and CCS to generate PacBio HiFi reads. Similar to the previous setup, for each reference genome we simulate reads with four different read length distributions. The simulated reads along with real reads for each of the four reference genomes are assembled with five assembly tools: HiCanu, Flye, Hifiasm, LJA, and MBG.

Next, the resulting assemblies are evaluated using COMPASS, QUAST, and BUSCO, and based on the reported metrics we distinguish six main evaluation categories: sequence identity, repeat collapse, rate of valid sequences, contiguity, misassembly count, and gene identification. The selected COMPASS metrics are the coverage, multiplicity, and validity of an assembly, which provide insight on sequence identity, repeat collapse, and the rate of valid sequences, respectively. In this regard, an ideal assembly has coverage, multiplicity and validity close to 1. This suggests that a large fraction of the reference genome is assembled, repeats are generally collapsed Instead of replicated, and most sequences in the assembly are validated by the reference. Among others, QUAST reports the number of misassemblies and the auNG of an assembly. A high auNG value indicates high contiguity. In order to assess contiguity across genomes of different sizes, we report the ratio between the assembly's auNG and the N50 of the references. Lastly, gene identification is quantified in terms of the percentage of complete BUSCOs in an assembly.

The search for an optimal assembler for PacBio CLR and ONT reads is influenced by read sequencing technology, genome complexity, and research goal To select an assembler that is most versatile across eukaryotic taxa, we simulate PacBio Continuous Long Reads (CLRs), and Oxford Nanopore reads from the genomes of six eukaryotes, assemble these reads, and evaluate the assemblers in the six main categories mentioned in the previous section. The results for each evaluation category are normalized in the range given by the worst and best values encountered in the evaluation of all assemblies of reads with default length. This highlights differences between assemblers, as well as between genomes and sequencing technologies.

The results of the benchmark on the PacBio CLR and ONT read sets with default lengths, namely those belonging to the first iteration (see Table 3.1), are illustrated in Figure 3.2. A full report of the evaluation metrics in this figure is included in the Supplementary Tables S8 – S24 in [16], under "Iteration 1". We note that no assembler unanimously ranks first in all categories, across different sequencing technologies and eukaryotic genomes, although our findings highlight some of their strengths and thus their potential for various research aims. The runtime and memory usage of the assembly tools on all of the simulated

datasets are reported in Supplementary Tables S25 – S30 in [16], since this can also be a deciding factor next to the quality of the assembly for the researchers to choose the suitable assembler for their purpose. We note that all assemblies were run on our local High Performance Computing Cluster, and the runtime and RAM usage may have been affected by the heterogeneity of the shared computing environment in which the assembly jobs executed.

While working with PacBio CLR and ONT reads, Miniasm, Raven, and Wtdbg2 are all well-rounded choices for the simpler S. cerevisiae, P. falciparum and C. elegans genomes, with a balanced trade-off between assembly quality and computational resources. For PacBio HiFi reads, Raven is generally qualitatively outperformed by other assemblers like Canu, Flye, and Miniasm, likely as a consequence of the fact that its pipeline is not customized for all long-read sequencing technology. Nonetheless, if computational resources are a concern, Raven is a more suitable choice, since Miniasm and Wtdbg2 do not scale well for larger genomes.

We can single out Flye as the most robust assembler for PacBio CLR and ONT reads across all six organisms, although for larger genomes such as T. rubripes, Canu is a better tool. Both produce assemblies with high sequence identity and validity, as well as good gene prediction, but Flye assemblies generally rank first when we compute the average score across all six metrics. For Canu, we notice more variation in assembly quality across different genomes, particularly for P. falciparum and A. thaliana, while Flye maintains more consistent results. Nonetheless, on the T. rubripes genome, Canu assemblies have higher sequence identity and contiguity, as well as more accurate gene identification.

To determine assembler performance on real PacBio CLR and ONT reads and validate the rankings of the simulated-read assemblies, we assemble several real read sets from the six reference eukaryotes (Supplementary Table S5 in [16]). Supplementary Figures S1-S12 in [16] provide a visual representation of the read length distribution for all of the real read sets. The evaluation results on the real-read assemblies, summarized in Figure 3.3, indicate that assemblers which perform well on simulated reads perform similarly well in assembling the sets of real reads. The full report of metrics on the real read assemblies is included in Supplementary Table S31 in [16]. We conclude that, overall, the assembler rankings remain consistent. This illustrates that benchmarking using simulated data is similar to real read sets. For reference-based metrics, we used the reference genomes given in Supplementary Table S1 in [16].

Notably, reference-based metrics in the evaluation of real-read assemblies rely on comparisons with an assembly, and not the genome from which the reads were initially sequenced. In contrast to the evaluation of simulated-read assemblies, the existence of a ground truth reference is not available in this case, but reference-based metrics are included for the sake of consistency with the simulated-read evaluation.

In the evaluation of real-read assemblies of PacBio CLR and ONT reads, Flye ranks first for nearly all datasets, with the exception of the T. rubripes and C. elegans PacBio reads, for which Raven performs better overall. However, even in *C. elegans*, Flye performance is close to the best values in all metrics other than contiguity. As expected, overall assembler performance decreases for reference-based metrics like sequence identity, repeat collapse and validity, but surprisingly the misassembly count is considerably lower.

Similarly, in order to identify the best performing HiFi assembler for diverse eukaryotic taxa, we first generate simulated PacBio HiFi reads from the genomes of four different

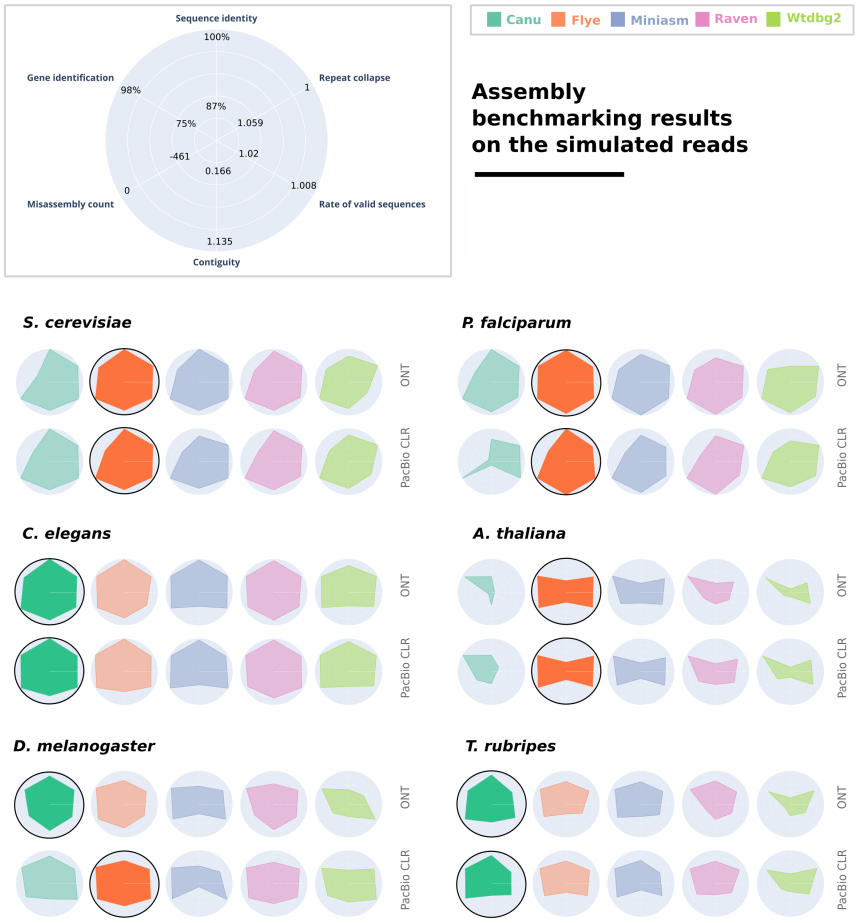


Figure 3.2: The performance of the five assemblers on the read sets with default read lengths, from iteration 1 (see Table 3.1), generated from six eukaryotic genomes. Six evaluation categories are reported for each assembler, and the results are normalized among all assemblies included in the figure. Ranges for each metric are reported as the best and worst values computed for these assemblies. The best performing assembler is highlighted and has a black outline. Evaluation of PacBio CLR and ONT real-read assemblies supports our rankings on simulated-read assemblies.

eukaryotes. These simulated reads are then assembled, and the performance of each assembler is evaluated based on the six primary categories outlined in the previous section. For comparative clarity, the results for each evaluation category are normalized within the range established by the lowest and highest values observed across all assembly evaluations of reads of default length. This method emphasizes both the variations among different assemblers, as well as the discrepancies across genomes and sequencing technologies.

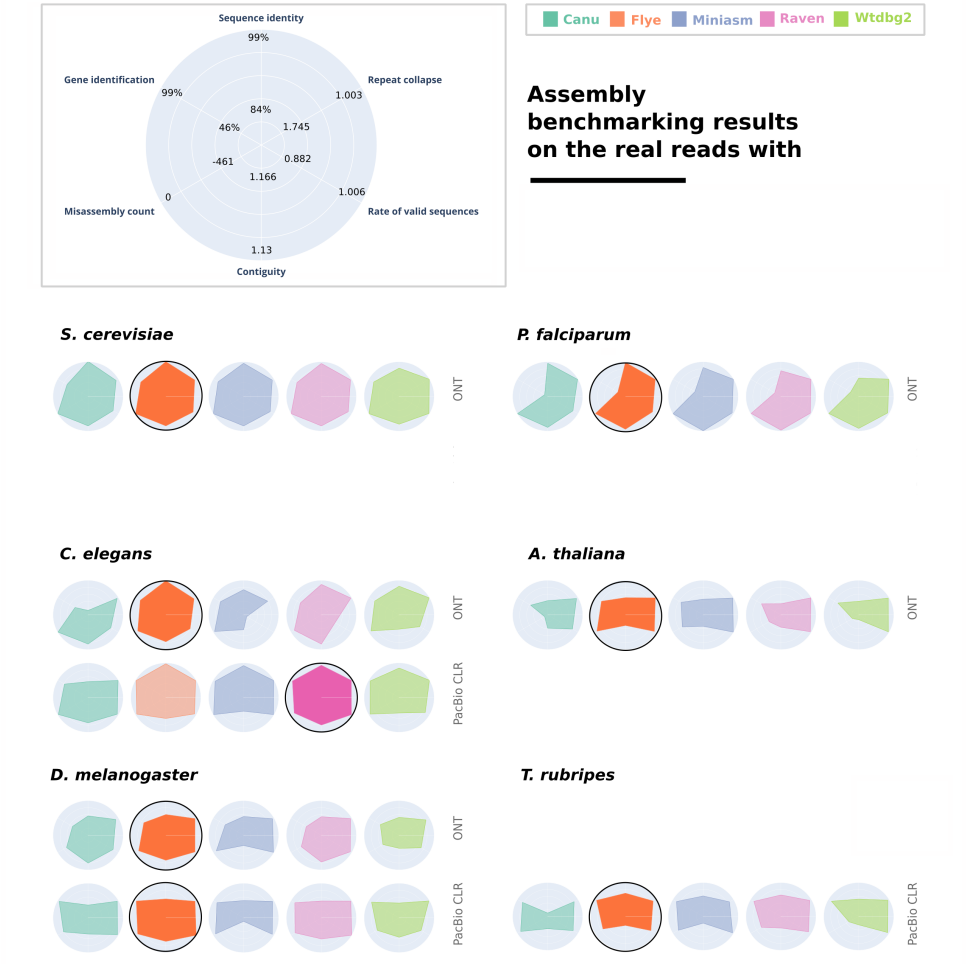


Figure 3.3: The performance of the five assemblers on the real PacBio CLR and ONT reads (see Supplementary Table S5 in [16]), sequenced from six eukaryotic genomes. As in Figure 3.2, six evaluation categories are reported for each assembler, and the results are normalized among all assemblies included in the figure. Ranges for each metric are reported as the best and worst values computed for these assemblies. The best performing assembler is highlighted and has a black outline. Searching for the best HiFi assembler based on simulated and real datasets

The results from simulated PacBio HiFi read sets with default lengths, namely those belonging to the first iteration (see Table 3.1), are illustrated in Figure 3.4. Next to that, the results of real HiFi reads of the same species are presented in Figure 3.4. We note that Hifiasm, and LJA are outperforming other assemblers and perform well in all datasets. The assembly results generated by the MBG assembler demonstrated notably low sequence identity when compared to the reference genome.

Figure 3.4: Theperformance of the five assemblers on the real PacBio HiFi read sets and simulated PacBio HiFi read sets with default read lengths, from iteration 1 (see Table 3.1), generated from four eukaryotic genomes. Six evaluation categories are reported for each assembler, and the results are normalized among all assemblies included in the figure. Ranges for each metric are reported as the best and worst values computed for these assemblies. The best performing assembler is highlighted and has a black outline.

3.3.2. LONGER READS LEAD TO MORE CONTIGUOUS ASSEMBLIES OF LARGE GENOMES, BUT DO NOT ALWAYS IMPROVE ASSEMBLY QUALITY

To investigate the effect of increased read length on assembly quality, we simulate Oxford Nanopore, as well as PacBio CLR and HiFi reads with different read length distributions (see Table 3.1). These reads are simulated from the genomes of *S. cerevisiae*, *P. falciparum*, *C. elegans*, *A. thaliana*, *D. melanogaster*, and *T. rubripes* for PacBio CLR and ONT reads, as well *S. cerevisiae*, *P. falciparum*, *A. thaliana*, and *D. ananassae* for PacBio HiFi reads. We assemble PacBio CLR and ONT reads with Canu, Flye, wtdbg2, Raven, and miniasm and assemble PacBio HiFi reads with HiCanu, Flye, Hifiasm, LJA, and MBG. We evaluate assembly quality based on six evaluation categories (see section 3.3.1). It is worth mentioning that Canu's PacBio CLR and ONT reads iteration 4 (the longest reads) assemblies of *A. thaliana* and *T. rubripes* did not finish within reasonable time and are excluded from the evaluation.

Figure 3.5 shows a summary of the assemblers' performance on all simulated read sets, highlighting changes in performance for each read length distribution. All six evaluation metrics are normalized given the maximum and minimum metric values per genome, per sequencing technology, and combined to obtain an average score. For PacBio CLR and ONT read sets, we then average the two resulted scores. Finally we report a rate between 1 and 10 for each assembler, per read length distribution for PacBio CLR and ONT read sets, and a separate score for PacBio HiFi read sets. The results on all computed metrics are fully described in Supplementary Tables S8 – S24 in [16].

The results imply that there is a correlation between the size and complexity of the reference genome and the extent of the improvement in assembly quality that can be achieved by increasing the length of the reads. While we observe no trend in assembly quality improvement on the assemblies of smaller genomes, the results on the *T. rubripes* assemblies are more conclusively in favour of the longer reads. For instance, on the shorter and simpler *S. cerevisiae* and *P. falciparum* genomes, identification of repetitive and complex regions is not aided by increased read length, likely as these regions are already spanned by the reads with default lengths. However, the benchmark results suggest that more complex and repetitive regions within the *A. thaliana*, *D. melanogaster* and, most notably, *T. rubripes* genomes are better captured by longer reads.

As recorded in Supplementary Tables S22 and S23 in [16], for larger genomes, longer reads generally lead to significantly higher assembly contiguity and a lower misassembly count. The latter implies that the resulting assemblies are more faithful to the references, although this is not necessarily supported by other metrics. We cannot report any compelling improvements in sequence identity, multiplicity, validity, and gene identification.

3.4. CONCLUSION

In fulfilment of the first objective of this study, we conclude that Flye is the highest performing assembler when considering the overview of all evaluation categories in this benchmark, which include the sequence identity, repeat collapse, rate of valid sequences, contiguity, misassembly count, and gene identification. Rankings are mostly consistent for all three sequencing platforms included in the study: PacBio CLR, PacBio HiFi, and ONT. However, no assembler ranks first in all evaluation categories, suggesting that the choice of assembler is often a trade-off between certain advantages and disadvantages. Therefore, we have corroborated the conclusion of Wick and Holt [7], who benchmarked long-read assemblers on prokaryotes, for eukaryotic organisms, and recommend that these benchmarking parameters are considered in relation to the desired outcome of an assembly experiment.

Additionally, the tests performed on real reads validate our rankings of simulated-read assemblies. Flye, the assembler that scored consistently well in most evaluation categories for assemblies of simulated reads in PacBio CLR and ONT datasets, also ranks first when evaluated on several sets of real reads sequenced on long-read platforms.

In our analysis, we found that when processing HiFi reads, both LJA and Hifiasm assemblers showed better performance than other options. While LJA and Hifiasm may not always have been the absolute best, their high performance was a constant, irrespective of the dataset. This was not dataset-specific but was consistently observed in both simulated and real datasets. This underscores their efficiency and accuracy in assembling genomic sequences using HiFi reads. Regarding our second objective, which is addressing the effect

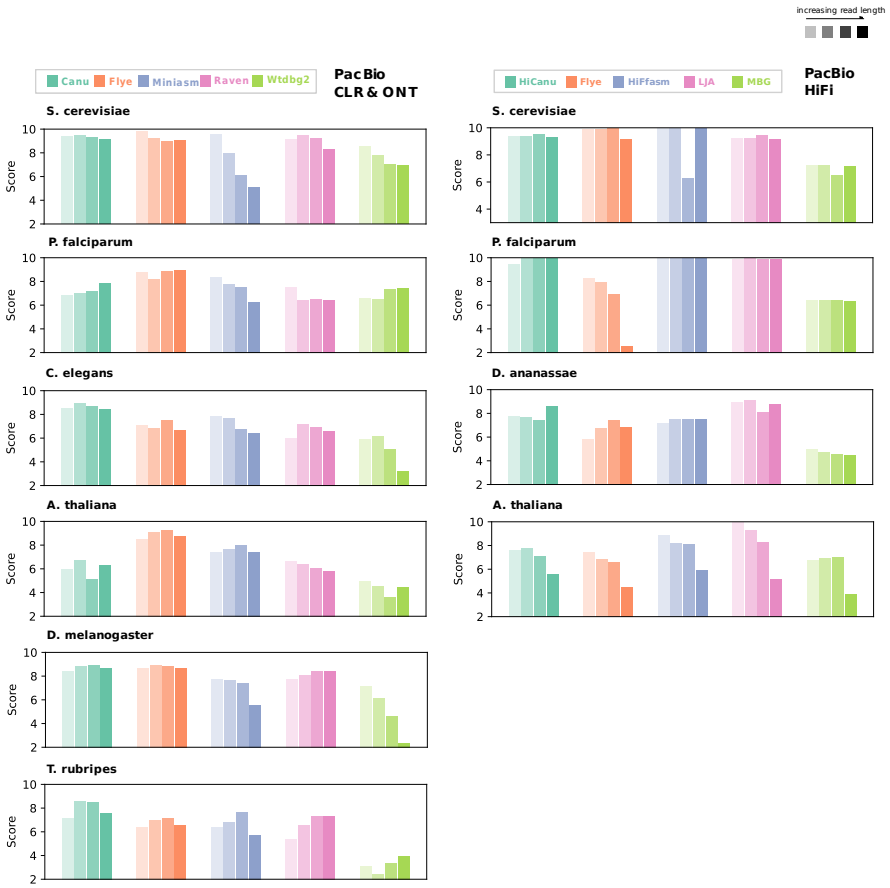


Figure 3.5: The left panel shows the performance of the five assemblers on all simulated PacBio CLR and ONT read sets, with four different read length distributions (as previously described in Table 3.1). A score of 1 - 10 is reported for each assembler. We did not divide auNGA with the n50 of the reference genomes for this figure. The results are normalized for each genome, per sequencing technology. For PacBio CLR and ONT, an average score for each read length distribution is first computed and then these two scores are averaged to obtain an overall score per read length distribution. For the A. thaliana and T. rubripes ONT iteration 4, the Canu assembly was not completed. Therefore, the iteration 4 bar in the plot represents only the PacBio CLR assemblies. Similarly, the right panel shows the performance of the five HiFi assemblers on all simulated PacBio HiFi read sets with four different read length distributions.

of increased read length on assembly quality, the benchmarking of assemblers on read sets with different read length distributions suggests that longer reads have the potential to improve assembly quality. However, this depends on the size and complexity of the genome that is being reconstructed. We found that improvements in contiguity were most significant

among all metrics, as also supported by the conclusion of [8], who showed that using third generation sequencing considerably improves contiguity in assembling a plant genome (*M. jansenii*). However, we did not find significant improvements in other aspects of assembly quality, such as sequence identity or gene identification.

This study focused on comparison of different sequencing technologies and assemblers on a specific coverage level of 30x, which provided insights into the performance of different assemblers. However, it's important to recognize that assemblers may behave differently at lower or higher coverage levels, and project planners need guidance in selecting the right coverage for their goals and budget. Unfortunately, studying the effect of different coverages on assembly performance is not part of this study.

The field of genomics is continuously evolving, and advancements in sequencing technologies can significantly influence assembly outcomes. While our study focuses on benchmarking long read de novo assembly tools for eukaryotic genomes, the rapid progress in sequencing technologies introduces complexities and challenges in comparing different data types, chemistries, and versions of the tools. In an ideal situation, it would be important to consider all the various factors, including different chemistries, sequencing devices, and base callers when evaluating assemblies. However, due to the limitations of available data and resources, we focused primarily on analyzing the impact of specific chemistry and related factors in this study. We recognize that this represents one of the limitations of our research.

The generations of HiFi reads have witnessed substantial advancements in both read length and accuracy. In earlier versions, HiFi reads typically had read lengths ranging from around 10 to 15 kilobases (kb) with high accuracy rates of 99.9% or greater. However, with subsequent generations, there has been a significant increase in read lengths. The latest versions of HiFi reads now offer read lengths exceeding 20 kilobases, with some reaching up to 30 kilobases or more, while still maintaining high accuracy rates above 99.9%. These longer and highly accurate HiFi reads provide researchers with more contiguous and reliable genomic sequences, enabling improved de novo assembly and enhancing various genomic analyses. An interesting innovation worth mentioning, while not included in this study, is the introduction of Oxford Nanopore's Duplex reads. This cutting-edge technology holds the potential to enhance sequencing accuracy even further, making it a worthwhile subject for future investigations.

BIBLIOGRAPHY

[1] K. M. Boycott et al. “Rare-disease genetics in the era of next-generation sequencing: discovery to translation”. In: *Nature Reviews Genetics* 14.10 (2013), pp. 681–691.

[2] J. Bras, R. Guerreiro, and J. Hardy. “Use of next-generation sequencing and other whole-genome strategies to dissect neurological disease”. In: *Nature Reviews Neuroscience* 13.7 (2012), pp. 453–464.

[3] A. Grada and K. Weinbrecht. “Next-generation sequencing: methodology and application”. In: *Journal of Investigative Dermatology* 133.8 (2013), pp. 1–4.

[4] C. Schlötterer et al. “Combining experimental evolution with next-generation sequencing: a powerful tool to study adaptation from standing genetic variation”. In: *Heredity* 114.5 (2015), pp. 431–440.

[5] A. N. Salazar et al. “Nanopore sequencing enables near-complete de novo assembly of Saccharomyces cerevisiae reference strain CEN. PK113-7D”. In: *FEMS Yeast Research* 17.7 (2017), fox074.

[6] S. L. Amarasinghe et al. “Opportunities and challenges in long-read sequencing data analysis”. In: *Genome biology* 21.1 (2020), pp. 1–16.

[7] R. R. Wick and K. E. Holt. “Benchmarking of long-read assemblers for prokaryote whole genome sequencing”. In: *F1000Research* 8 (2019).

[8] V. Murigneux et al. “Comparison of long-read methods for sequencing and assembly of a plant genome”. In: *GigaScience* 9.12 (2020), giaa146.

[9] H. Jung et al. “Comparative evaluation of genome assemblers from long-read sequencing for plants and crops”. In: *Journal of Agricultural and Food Chemistry* 68.29 (2020), pp. 7670–7677.

[10] Z. Chen, D. L. Erickson, and J. Meng. “Benchmarking long-read assemblers for genomic analyses of bacterial pathogens using oxford nanopore sequencing”. In: *International journal of molecular sciences* 21.23 (2020), p. 9161.

[11] I. Letunic and P. Bork. “Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation”. In: *Nucleic acids research* 49.W1 (2021), W293–W296.

[12] R. R. Wick. “Badread: simulation of error-prone long reads”. In: *Journal of Open Source Software* 4.36 (2019), p. 1316.

[13] D. Earl et al. “Assemblathon 1: a competitive assessment of de novo short read assembly methods”. In: *Genome research* 21.12 (2011), pp. 2224–2241.

[14] K. R. Bradnam et al. “Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species”. In: *Gigascience* 2.1 (2013), pp. 2047–217X.

[15] E. L. Van Dijk et al. “Ten years of next-generation sequencing technology”. In: *Trends in genetics* 30.9 (2014), pp. 418–426.

[16] B.-M. Cosma et al. “Evaluating long-read de novo assembly tools for eukaryotic genomes: insights and considerations”. In: *GigaScience* 12 (2023), giad100.

[17] S. Nurk et al. “The complete sequence of a human genome”. In: *Science* 376.6588 (2022), pp. 44–53.

[18] H. Li. “Minimap2: pairwise alignment for nucleotide sequences”. In: *Bioinformatics* 34.18 (2018), pp. 3094–3100.

[19] S. Koren et al. “Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation”. In: *Genome research* 27.5 (2017), pp. 722–736.

[20] M. Kolmogorov et al. “Assembly of long, error-prone reads using repeat graphs”. In: *Nature biotechnology* 37.5 (2019), pp. 540–546.

[21] J. Ruan and H. Li. *Fast and accurate long-read assembly with wtdbg2*. Tech. rep. 2. 2020, pp. 155–158.

[22] R. Vaser and M. Šikić. “Time-and memory-efficient genome assembly with Raven”. In: *Nature Computational Science* 1.5 (2021), pp. 332–336.

[23] H. Li. “Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences”. In: *Bioinformatics* 32.14 (2016), pp. 2103–2110.

[24] S. Nurk et al. “HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads”. In: *Genome research* 30.9 (2020), pp. 1291–1305.

[25] H. Cheng et al. “Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm”. In: *Nature methods* 18.2 (2021), pp. 170–175.

[26] A. Bankevich et al. “Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads”. In: *Nature biotechnology* 40.7 (2022), pp. 1075–1081.

[27] M. Rautiainen and T. Marschall. “MBG: Minimizer-based sparse de Bruijn graph construction”. In: *Bioinformatics* 37.16 (2021), pp. 2476–2478.

[28] Z. Chen, D. L. Erickson, and J. Meng. “Polishing the Oxford Nanopore long-read assemblies of bacterial pathogens with Illumina short reads to improve genomic analyses”. In: *Genomics* 113.3 (2021), pp. 1366–1377.

[29] T. Hu et al. “Next-generation sequencing technologies: An overview”. In: *Human Immunology* 82.11 (2021), pp. 801–811.

[30] R. R. Wick and K. E. Holt. “Polypolish: short-read polishing of long-read bacterial genome assemblies”. In: *PLoS computational biology* 18.1 (2022), e1009802.

[31] R. Vaser et al. “Fast and accurate de novo genome assembly from long uncorrected reads”. In: *Genome research* 27.5 (2017), pp. 737–746.

[32] A. Gurevich et al. “QUAST: quality assessment tool for genome assemblies”. In: *Bioinformatics* 29.8 (2013), pp. 1072–1075.

[33] R. Barthelson et al. “Plantagora: modeling whole genome sequencing and assembly of plant genomes”. In: *PLoS One* 6.12 (2011), e28436.

[34] F. A. Simão et al. “BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs”. In: *Bioinformatics* 31.19 (2015), pp. 3210–3212.

[35] R. M. Waterhouse et al. “BUSCO applications from quality assessments to gene prediction and phylogenomics”. In: *Molecular biology and evolution* 35.3 (2018), pp. 543–548.

4

THE EFFECT OF REMOVING REPEAT-INDUCED OVERLAPS IN *de novo* ASSEMBLY

Determining accurate genotypes is important for associating phenotypes to genotypes. de novo genome assembly is a critical step to determine the complete genotype for species for which no reference exists yet. The main challenge of de novo eukaryote genome assembly, particularly plant genomes, are repetitive DNA sequences within their genomes. The introduction of third generation sequencing and corresponding long reads has promised to resolve repeat-related problems. While there have been notable improvements, reads originating from these repeats are still creating errors because they introduce false overlaps in the assembly graph. This study focuses on analyzing the effect of repeats on de novo assembly and improving performance of existing de novo assembly algorithms by removing repeat-induced overlaps. First, we show the possible improvements in de novo assembly with removing repeat-induced overlaps. Then we propose several methods for detecting and removing repeat-induced overlaps and evaluate their performance on several simulated datasets.

This chapter has been submitted to PLOS One

4.1. INTRODUCTION

The goal of *de novo* genome assembly is to reconstruct a species' genome sequence as completely as possible using a large number of relatively short sequences referred to as "reads" that are read from the species' genome. While high-quality assemblies are already available for many species, many branches of the tree of life still need representative genome sequences. Recently, due to the popularity of long-read sequencing technologies, *de novo* assembly has once more become of interest. In this paper, we focus on improving the standard long read *de novo* assembly pipeline.

Most *de novo* assembly pipelines suitable for long reads follow the OLC paradigm: overlap-layout-consensus. First, in the overlap step, pairwise alignments between the reads are identified. The output of the overlap step is a set of pairwise read overlaps that can be represented as a graph, where nodes are the reads, and edges indicate overlaps between the reads. This graph will be referred to as the assembly graph. Second, the layout step tries to identify bundles of overlaps that belong together. This is done by pruning unwanted edges from the graph such that it becomes more linear through several graph cleaning procedures. Once all procedures are done, the graph is split up into contigs. Finally, the consensus step of the assembly pipeline identifies the most likely base for each position. The layout step is arguably the most differentiating step between the various *de novo* assembly methods that exist. This can go from extremely simple, e.g. miniasm [1] to very intricate with many manually optimized rules and corresponding specific data types, e.g. DISCOvar [2].

A problem that has plagued *de novo* assembly since the beginning is interspersed repeats in the species' genome sequence. The interspersed repeats are sufficiently similar sequences that occur in two or more distinct genomic locations. The reads originated from any of the repeat instances introduce pair-wise overlaps with all instances of the repetition across the genome, which leads to cross-connections in the assembly graph. This will confuse the 'layout' step in the OLC assembly paradigm. Reads spanning the repetitive region can resolve the confusion by connecting the two sides of the repetitive regions together. While read lengths have been increasing dramatically for Third Generation Technologies (TGS), for the vast majority of eukaryotic species, the read length is still orders of magnitude smaller than the genome size. Moreover, it is unlikely that we will experience the luxury of chromosome-spanning reads like the ones observed for some microbial genomes soon [3–5]. Finally, TGS reads are often still not (yet) long enough to span most of the repetitive regions in eukaryotic genomes.

In this paper, we analyze the effect of interspersed repeats on *de novo* assembly. Next, we show that removing repeat-induced overlaps can improve the performance of *de novo* assembly in different eukaryotic genomes, e.g. yeast, human, and potato. We demonstrate that a perfect classifier can increase the coverage of genome assembly by 0.1, 4 and 7 in yeast, potato, and human chromosome 9, respectively. Finally, we also investigate some methods to detect and remove repeat-induced overlaps and compare their performance to the standard *de novo* assembly pipeline. Initially, we tried a baseline method and removed overlaps based on their degree in the assembly graph. Second, we trained a machine-learning model to detect and remove repeat-induced overlaps based on GraphSage node embeddings [6]. While this method makes the overlaps set much smaller, it does not improve the assembly performance and the results are close to the standard *de novo* assembly pipeline.

4.2. MATERIAL AND METHODS

4.2.1. DATA

REFERENCE SEQUENCES

In this study, we use the reference sequences of three species with differing degrees of repetitive sequences: *S. cerevisiae* (yeast) and *S. tuberosum* (potato), and *H. sapiens* (human) chromosome 9, which is the most repetitive chromosome in the human genome. We use high-quality available reference sequences as the source to simulate reads. We retrieve sequences from Genbank: yeast S288C genome assembly R63 (GCA_000146045.2), potato DM_1-3_516_R44 genome assembly version 6.1 (GCA_000226075.1), and human genome assembly T2T-CHM13v2.0 (GCA_009914755.3).

The potato reference sequence contains Ns to fill the gaps and unplaced sequences, complicating analysis. The Ns make problems for the evaluation step because we need a complete genome to compare the assemblies with it. We remove the unplaced sequences and the Ns to make the experiments straightforward. After removing Ns and unplaced contigs, we have one complete sequence for each chromosome.

DETECTING INTERSPERSED REPEATS

We use Generic Repeat Finder [7] version 1.0 with the default parameters to detect interspersed repeats in these three reference sequences.

4.2.2. SIMULATING READS AND GENOMES

We use aneusim [8] version 0.4.1 with default parameters to simulate diploid sequences (ploidy=2) close to the reference sequences but with mutations and translocations. We use the simulated haplotype 1 and 2 sequences as genomes of two other individuals of these organisms for further analysis.

We use SimLoRD [9] version 1.0.2 to simulate reads similar to PacBio with 40x of coverage (-c 40) from the reference, and the simulated sequences. Using simulated reads allows us to label the alignments between the reads since we know where the reads originated from.

4.2.3. ALIGNMENTS AND LABELING

We use minimap2 [10] version 2.13-r858-dirty with the default parameters to find the pairwise alignments between the reads. We label each alignment according to the origination coordinates of the reads participating in it. If the origination coordinates of the reads participating in an alignment overlap, then we label the alignment as a normal overlap. Otherwise, we label the overlap as a repeat-induced overlap.

4.2.4. GENOME ASSEMBLY AND EVALUATION

We use the miniasm [1] version 0.3-r179 with default parameters to assemble the sets of overlaps before and after intervening and removing the candidate alignments.

We use compass [11, 12] to evaluate the *de novo* assemblies. While compass reports many metrics, we only report coverage, validity, multiplicity, the number of contigs, and the longest contig. Supplementary Table C.1 list the metrics and explain them. Coverage is the most important metric for this study because it shows what percentage of the genome is

missing in the assemblies and can show us how much extra sequence, we achieve by removing repeat-induced overlaps. Another important metric is the number of contigs representing the assembly's contiguity. It is essential to achieve higher coverage while maintaining the contiguity of the assembly.

4.2.5. FEATURE EXTRACTION AND TRAINING CLASSIFIER

We use the reference sequences and the first simulated haplotypes as the training set and the second simulated haplotypes for the test. To train the model, first, we need to extract features for each overlap based on the assembly graph.

First, we create the graph using networkx [13] version 2.8.4. Then, we train a GraphSage (6) model on the assembly graph using the StellarGraph [14] library version 1.2.1 while the only attribute we add to the nodes is their degree. To learn the embeddings, we make a model which gets two nodes as input and predicts if there is a normal edge, repeat-induced edge, or no edge between them. Our model consists of three GraphSage layers with followed by a softmax layer for the prediction. We use categorical cross entropy as the loss function and Adam optimizer to train the network (learning rate = 0.001). This model contains 3 GraphSage blocks, which each contains 50, 50, and 20 GraphSage layers, respectively. Moreover, the network iterates each GraphSage block 20 times before delivering the output to the next block. We train the network for 20 epochs and the batch size is 50. Since GraphSage models are inductive, after training the model, we can use the output of GraphSage layers to get the node embeddings in other graphs.

However, because the assembly graphs are huge, we need to subsample the graph for training and testing the model. We use the edgesampler module in the StellarGraph library to get the subgraphs. For yeast sequences, we take 20% of the nodes for training and 20% of the nodes for testing, while for human sequences, we use 2% of the nodes for training and 2% for testing.

Then, we use GraphSage embeddings to train a logistic regression classifier for separating repeat-induced and normal overlaps. We use the first simulated dataset to train this classifier. First, we create the assembly graph of the simulated dataset, and then extract the node embeddings using the previously trained GraphSage.

We use the GraphSage model to extract node embedding for every node in the assembly graph, and we concatenate embeddings of the two nodes participating in an edge, to get embedding of that edge, which represents an overlap. After creating the embedding of each overlap, we use sklearn [15] version 1.0.2 to train a logistic regression classifier with parameter C=0.001 to detect repeat-induced overlaps. We use 10-fold cross-validation to evaluate the classifier and select the model with the highest F1 score.

Finally, we use the GraphSage model to extract the embeddings of the second simulated dataset. Then we use the selected model from the previous step to remove overlaps classified as repeat-induced. Next, use miniasm [1] version 0.3-r179 to assemble the remaining overlap set and compare the results with the standard genome assembly pipeline.

4.3. RESULTS AND DISCUSSIONS

4.3.1. CHARACTERISTICS OF INTERSPERSED REPEATS IN YEAST, POTATO, AND HUMAN GENOMES.

In the first step, we used Generic Repeat Finder to detect interspersed repeats in the genome of yeast, potato, and human chromosome 9. Table 4.1 shows the statistics of the interspersed repeats available in these genomes. There are gaps in the potato reference sequence, which are indicated by Ns in the sequence. To simplify the analysis, we removed Ns from the reference sequence. Unresolved repeats are usually responsible for most Ns in the sequence. Consequently, in Table 4.1, we report fewer interspersed repeats for the potato genome than are present. The analysis is also simplified for human chromosome 9 since it is separated from the rest of the chromosomes, thereby excluding the occurrence if interspersed repeats in the other chromosomes from the analysis.

Table 4.1: The amount of interspersed repeats in yeast, potato and human chromosome 9 genomes.

Organism	Genome size	Number of repeats	Repeat content (%)
Yeast	12Mbp	4022	28Kbp (0.2%)
Potato	731Mbp	8582087	76Mbp (10.3%)
Human chromosome 9	150Mbp	625288	9Mbp (6%)

As shown in Table 4.1, the repeat content is much higher in human chromosome 9 and potato than in yeast. Around 10% of a potato genome is interspersed repeats, which shows the high repetitive content in that is a hallmark of plant genomes. Human chromosome 9 contains 6% interspersed repeats, but this number may be higher if the entire genome is considered. There are only 0.2% interspersed repeats in yeast's reference genome, indicating a simpler genome architecture.

The distribution of interspersed repeats follows a similar pattern in the three test organisms. However, human chromosome 9 has many longer repeats than the other two organisms (see Figure 4.1). As mentioned before, the count of repeats in the human genome can be even more than what is shown in Figure 4.1 because they might also be present in other chromosomes, which we did not consider in this study. Interestingly, although yeast has lower repeat content (see Table 4.1) than the other two organisms, it has some very long repeats. The longest repeats in the yeast genome are even longer than the potato's longest repeats. However, this is likely due to the fact that the potato reference sequence is incomplete and the Ns are representing unresolved repeats.

The number of times each repeat occurs varies from 2 to more than 1000 times in the three model organisms (see Figure 4.2). There are interspersed repeats in Human chromosome 9 that occur more than 40000 times, without considering other chromosomes that these repeats might be present. It is worth noting that the smaller repeats occur more often through the genome (see Supplementary Figure C.1).

4.3.2. THE EFFECT OF INTERSPERSED REPEATS IN GENOME ASSEMBLY

Next, we inspected the effect of interspersed repeats in genome assembly based on simulated reads from the reference genomes. Since the simulator reports the coordinates where a simulated read originated from, it is possible to label the pairwise alignment of reads. If

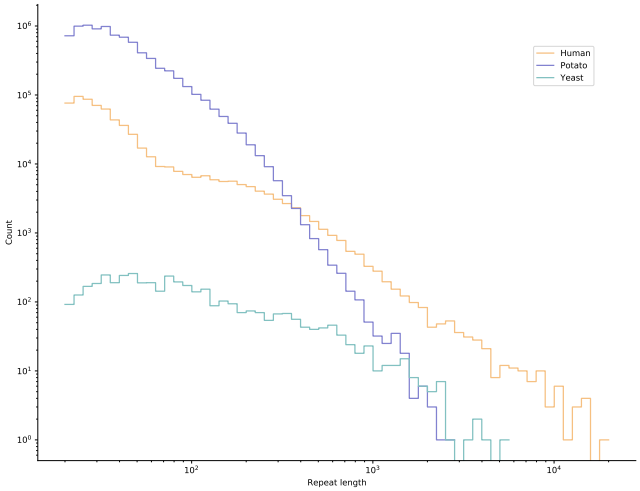


Figure 4.1: Histogram of the length distributions of interspersed repeats on chromosomes 9, potato, and yeast. In these three organisms, most interspersed repeats are smaller than 1000 bp. Despite this, all three organisms have repeats longer than 1000 bp, which complicates the *de novo* assembly process, as not all long reads will span the repeats completely.

there is an alignment between two reads but the coordinates these reads are sampled from do not overlap, we considered the alignment as repeat-induced. Otherwise, we labeled the alignment as normal. Table 4.2 shows the number of repeat-induced edges in yeast, human chromosome 9, and potato.

Table 4.2: This table shows the number of repeat-induced and normal edges in the assembly graphs. Although humans and potatoes have only 6% and 10% repetitive sequences in their genomes, they have 71% and 96% repeat-induced edges in their assembly graphs.

Organism	**Repeat-induced edges (%)**	**Normal edges (%)**
Yeast	189842 (8%)	2093297 (92%)
Potato	308658703 (96%)	12084513 (4%)
Human chromosome 9	63004592 (71%)	25221954 (29%)

Reads that originate from one of the interspersed repeats align with reads from all other instances, which creates repeat-induced edges in the assembly graph. The human and potato reference sequences have considerably high repetitive sequences. Therefore, in the human and potato assembly graphs, the majority of the edges are repeat-induced in their assembly graphs (see Table 4.2). Subsequently, the reads originating from interspersed repeat regions also have a high degree in the assembly graph. Figure 4.3 shows the degree of the normal and repeat-induced edges in the assembly graphs. We define the degree of an edge as the sum of the degree of the two nodes connected by the edge. Figure 4.3 shows that most

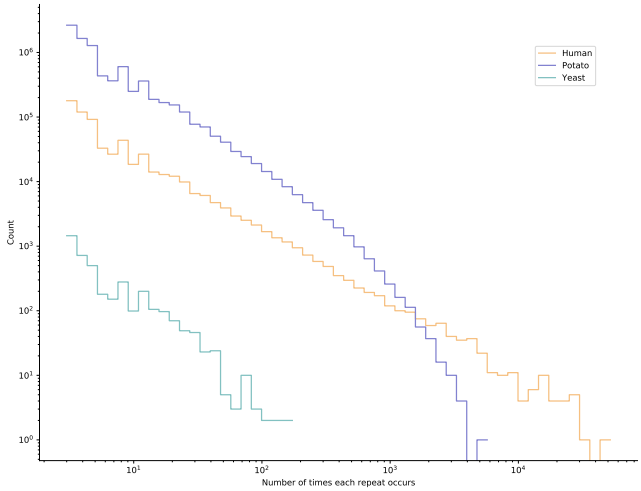
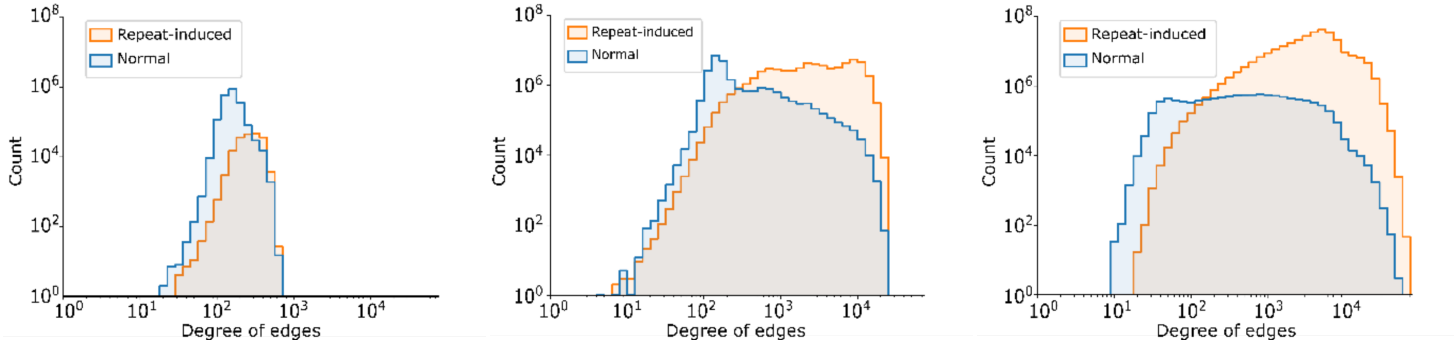


Figure 4.2: Histogram of the number of times each repeat occurs in the genome. The majority of interspersed repeats occur less than 100 times, but there are repeats in potato and human genomes that occur more than 1000 and 10,000 times, respectively.

edges with a degree greater than 1000 are repeat-induced.



(a) Yeast (b) Human chromosome 9 (c) Potato

Figure 4.3: This figure compares the histogram of the degrees of repeat-induced and normal edges in the assembly graphs of yeast (a), human chromosome 9 (b), and potato (c). The degree of an edge is defined as the sum of the degrees of the two nodes it connects. There is no significant difference between the degree of repeat-induced and normal edges in the yeast assembly graph. On the other hand, in human chromosome 9 and potato, most edges with degrees greater than 1000 are repeat-induced.

To analyze the effect of repeat-induced overlaps in the assembly, we evaluated assemblies in the three model organisms before and after removing repeat-induced overlaps. In the normal scenario, we aligned the reads with minimap2 and assembled the genome with miniasm, reads, and the overlaps from the last step. In the removing repeat-induced overlaps scenario, we intervened in the assembly process, removed all the alignments labeled as repeat-induced, and used miniasm to assemble the remaining overlaps set.

Table 4.4 shows the results of these two scenarios in the three model organisms. In all three datasets, removing repeat-induced overlaps improves genome assembly. In the yeast genome, removing repeat-induced overlaps lead to 6% more coverage. In the potato genome removing repeat-induced overlaps lead to 8% more coverage. This is expected since the potato genome is much more repetitive than yeast and suffers from more repeat-induced edges. In the human chromosome 9 dataset removing repeat-induced edges lead to 3% more coverage.

We tested whether removing a percentage of repeat-induced overlaps would still improve assembly performance in another experiment, where we removed 25%, 50%, and 75% of repeat-induced overlaps in the human chr9 genome and compared the final assemblies. It is clear from Table 4.3 that removing more repeat-induced overlaps improves coverage and validity and increases the length of the longest contig. However, the multiplicity, number of contigs and the assembly size is increasing after removing 25%, 50%, 75% repeat-induced overlaps and finally drops and get closer to one after removing all of the repeat-induced overlaps. This means by removing a portion of repeat-induced overlaps the assembler is replicating some of the repetitive regions which are valid sequences, but increases multiplicity and assembly size. Finally, with removing all of the repeat-induced overlaps, the assembler can fully resolve these repetitive regions and merge the corresponding contigs together which results in multiplicity closer to one, assembly size closer to the reference size, and reduced number of contigs. In conclusion, comparatively to the standard *de novo* assembly pipeline, removing 25%, 50%, and 75% of repeat-induced overlaps produces more contigs. This means even removing a subset of repeat-induced overlaps accurately, without false positives, can improve *de novo* assembly performance.

Table 4.3: The performance of standard *de novo* assembly pipeline compared to *de novo* assembly after removing 25%, 50%, 75% and all of the repeat-induced. These metrics are described in Supplementary Table C.1. With removing more repeat-induced overlaps, the coverage of assemblies is increasing. However, with removing 25%, 50%, and 75% of the repeat-induced overlaps, the number of contigs, the assembly size and the multiplicity is increasing. Meanwhile, with removing all of the repeat-induced overlaps, the number of contigs drops significantly which shows the importance of removing all of the repeat-induced overlaps.

Genome	Method	Coverage	Validity	Multiplicity	Assembly size	# contigs	Longest contig
Human chr 9 (size = 150464616 bp)	Baseline	0.850	9.17	1.075	150023015	1961	7250746
	Repeat-induced removal 25%	0.858	0.913	1.092	154715454	2405	7254952
	Repeat-induced removal 50%	0.868	9.16	1.117	159261685	2673	8686274
	Repeat-induced removal 75%	0.881	9.19	1.134	163756583	2806	8686376
	Perfect repeat removal	0.907	9.23	1.031	152588360	924	27151259

Finally, we examined the sequence differences we got from removing the repeat-induced edges compared to following the normal genome assembly pipeline. The assembly with all repeat induced edges removed is covering additional 9476429 bp of the reference genome that is not covered in the baseline assembly. Of this additional sequence, 92% turns out to be interspersed repeat sequences. Conversely, the assembly with all repeat induced

edges removed is also missing 3293397 bp with respect to the baseline assembly. Again, 93% of these are from the interspersed repeat regions. In conclusion, the majority of the newly discovered regions as well as those lost when repeat-induced overlaps were removed come from repetitive regions of human chromosome 9. It appears that repeat-induced overlaps are occasionally helpful in assembling repetitive regions, but that removing repeat-induced overlaps will result in the assembly of more repetitive regions overall.

4.3.3. TRAINING A CLASSIFIER TO REMOVE REPEAT-INDUCED OVERLAPS

Since the sequence of the interspersed repeats is almost identical, we relied only on graph-based features to find and remove them. One of graph based features that can be informative to detect repeat-induced overlaps is degree. We expect the edges in the assembly graph representing repeat-induced overlaps to have a high degree since they connect two reads from the repetitive regions and those reads also align to reads originating from all other instances of the repeat. Figure 4.3 compares the degree of repeat-induced and normal edges in the assembly graphs. Based on Figure 4.3, the number of repeat-induced edges with a degree greater than 1000 is more than normal edges. However, considering edges with a degree greater than 10000, the difference is much higher, and the number of repeat-induced edges is significantly more.

Therefore, we intervened in the *de novo* assembly process and removed the nodes representing overlaps with a degree greater than 10000 to see if removing them can improve the final assembly result. Table 4.4 shows the result of removing repeat-induced overlaps based on degree. No improvements are observed using this method over standard assembly pipelines. Since the yeast assembly graph does not have any edge with degree greater than 10000, we did not apply this method on it.

Table 4.4: The standard *de novo* assembly pipeline performance compared to perfect repeat-induced overlap removal and various repeat-induced overlap detection methods. The metrics are described in Supplementary Table C.1. In all of the three test organisms, removing all of the repeat-induced overlaps improve the performance significantly, compared to the baseline scenario. In the degree method, edges with degree greater than 10000 are removed from the assembly graphs. Since the yeast assembly graph has no edge with a degree greater than 10000, we cannot apply the degree method to the yeast dataset. On the other hand, training and testing the machine-learning models require huge memory and is not achievable on the potato dataset. Our results show that, unlike the perfect repeat-induced removal scenario, these methods cannot improve the standard *de novo* assembly pipeline. The machine learning method results in fewer contigs compared to the standard *de novo* assembly pipeline, while it is losing some coverage.

Organism	Model	Coverage	Validity	Multiplicity	Assembly size	# contigs	Longest contig
Yeast	Baseline	0.973	0.943	1.014	12726687	33	958030
	Machine-learning	0.933	0.934	1.004	12174134	29	1297877
(size = 12144833 bp)	Perfect repeat removal	0.961	0.934	1.003	12531324	25	1162078
Human chr 9	Baseline	0.811	0.878	1.082	150646955	2143	6179208
	Degree-based removal	0.811	0.879	1.084	150834800	2173	5430347
(size = 150617247 bp)	Machine-learning removal	0.691	0.939	1.006	111503649	722	2659552
	Perfect repeat removal	0.907	9.23	1.031	152588360	924	27151259
Potato	Baseline	0.631	0.945	1.068	522035794	12794	315461
	Degree-based removal	0.629	0.945	1.069	520480215	12796	315461
(size = 731207187 bp)	Perfect repeat removal	0.701	0.941	1.008	549511126	11805	315508

Another way to detect repeat-induced overlaps is to train a machine learning-based classifier based on graph-based embedding. First, we generated separate train and test datasets

to evaluate this method fairly. We simulated two reference sequences based on the reference genome of the three organisms we analyze. After that, we simulated reads from these simulated reference sequences and performed a pairwise alignment between the reads. We used the reference genome and the first simulated read set to train and test the GraphSage embedding model. To train the GraphSage embedding, we select subgraphs using StellarGraph's edgesplitter method. Then we labeled each pair of nodes in the subgraph as 0, 1, 2 where 0 represents normal edge, 1 repeat-induced edge, and 2 no edge. Table 4.5 shows the performance of the GraphSage embedding model on train and validation data. Interestingly, the model is not efficient in separating the three classes of edges in the yeast dataset, while it is performing well on human chromosome 9 dataset.

Table 4.5: This table shows the performance of the GraphSage embedding model and the logistic regression classifier. We use the edgesplitter module in the StellarGraph library to sample subgraphs for the train and test datasets. The size of subgraphs is 20% and 2% of the actual yeast's and human's assembly graphs, respectively. To test the performance of the logistic regression classifier, we use a 10-fold cross-validation. Interestingly, the human GraphSage and logistic regression models perform better than the yeast ones, showing more significant differences between the repeat-induced and normal edges in the human assembly graph.

GraphSage model				
Metric	**Train accuracy**		**Validation accuracy**	
Yeast	0.5356		0.5387	
Human chromosome 9	0.7653		0.7646	
Logistic regression classifier				
Metric	**F1 score (SD)**	**Accuracy (SD)**	**Precision (SD)**	**Recall (SD)**
Yeast	0.761 (0.007)	0.936 (0.002)	0.788 (0.008)	0.740 (0.007)
Human chromosome 9	0.887 (0.001)	0.911 (0.001)	0.915 (0.001)	0.868 (0.001)

Next, we used the extracted embeddings of overlaps in the second simulated dataset to train a classifier for separating normal and repeat-induced overlaps. Since the dataset is imbalanced, and the graphs have more normal edges in yeast genome and more repeat-induced edges in human, we up-sampled and down-sampled repeat-induced edges in yeast and human datasets, respectively. Following that, we trained a logistic regression classifier and evaluated it with 10-fold cross-validation (see Table 4.5). While the GraphSage embedding model failed to separate the three classes of edges in the yeast dataset, the logistic regression classifier achieved impressive results in separating repeat-induced and normal edges using the same embedding model on the second simulated dataset. Interestingly, the GraphSage model performed much better on the human chromosome 9 assembly graph and achieved 76% validation accuracy.

Last, we extracted the embeddings of overlaps in the last dataset and used the classifier trained in the previous step that achieved the highest F1 score to predict the repeat-induced overlaps. After removing the overlaps predicted as repeat-induced, we assembled the remaining overlaps and evaluated the results (see Table 4.4). The performance of yeast assembly drops after removing the overlaps predicted as repeat-induced. That means that the disadvantage of losing some of the normal edges in the yeast assembly graph because of prediction errors is more than the advantage of removing repeat-induced overlaps. Since the yeast genome does not have many interspersed repeats and repeat-induced edges (see Tables 3.1 and 4.2), this is not surprising. On top of that, the only feature we assigned to the

nodes before training the GraphSage model is the degree of nodes, while in the yeast assembly graph, the degree of repeat-induced and normal edges is not significantly different (see Figure 4.3a). However, the length of the longest contig is increased, and the number of contigs is reduced, which shows that the method solved the previously challenging repetitive regions.

Similar to yeast, human chromosome 9 assembly performance is lower than baseline after removing overlaps predicted to be repeat-induced (see Table 4.4). The coverage is 12% lower and the assembly size is 40Mbp smaller than the actual chromosome 9 size. The number of contigs is smaller than all the other cases, and the multiplicity and validity are close to one, which means the assembly and reference map are nearly one-to-one. As a result, the machine learning method is successful in removing some essential repeat-induced overlaps, which enables the assembler to merge the contigs that were split apart before. However, the model also incorrectly predicts some critical normal overlaps as repeat-induced, resulting in decreased coverage and assembly size when they are removed. Despite our best efforts, we were unable to apply the machine-learning method to the potato dataset due to its large size and memory requirement.

4.4. CONCLUSION

In this study, we study the effect of interspersed repeats on *de novo* genome assemblies of three organisms, i.e., yeast, human chromosome 9, and potato. The reads originating from interspersed repeat regions align with those from all instances. Therefore, it is possible to label the alignments with not overlapping originating coordinates as repeat-induced overlaps. Here, we analyze the effect of repeat-induced overlaps in the assembly graph and *de novo* assembly. At last, we investigate some strategies to detect and remove repeat-induced overlaps.

Interspersed repeats make up approximately 1, 6, and 10% of the yeast, human chromosome 9, and potato genomes, respectively. Although the repeats are causing only 1% of the overlaps in the yeast dataset, they correspond to 76% and 96% of overlaps in human and potato datasets. Since most of the overlaps in the assembly graph of these two genomes are repeat-induced, this is the most challenging problem to solve in genome assembly.

To investigate the effect of repeat-induced edges in the assembly graph on the final assembly result, we removed all of the repeat-induced overlaps and compared the results to the normal *de novo* assembly pipeline. We observed that removing repeat-induced overlaps improved coverage and continuity of the assembly, even in yeast with much lower repetitive content. In potato, which has the most repetitive contents among the test organisms, removing repeat-induced edges leads to a 9% improvement in coverage.

We investigate if it is possible to detect repeat-induced overlaps based on the degree of their corresponding edges in the assembly graph. We define the degree of an edge as the sum of the degree of two nodes connecting the edge. As shown in Figure 4.3, most of the repeat-induced overlaps in human chromosome 9 and potato assembly graphs have more than degree 10000. Therefore, we remove edges with more than degree 10000 and see the effect of it on the final assemblies. As shown in Table 4.4, there is no improvement in the assemblies after removing edges with degrees greater than 10000, and the final assemblies are very close to the standard assembly pipeline.

We also attempt to train a classifier to detect repeat-induced edges based on graph-

based features. Although we achieved some improvement after removing repeat-induced edges with the classifier, the results are far from the results when all of the repeat-induced edges are removed. This shows great potential for a follow-up project to detect and remove repeat-induced overlaps accurately.

We suggest that detecting and removing repeat-induced overlaps can be one a smart edge filtering method during assembly. Our attempt to train a classifier that accurately detects and removes repeat-induced overlaps did not achieve significant results. However, our results show that a perfect classifier that removes all the repeat-induced overlaps can make impressive improvements in the genome assembly process.

BIBLIOGRAPHY

[1] H. Li. “Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences”. In: *Bioinformatics* 32.14 (2016), pp. 2103–2110.

[2] N. I. Weisenfeld et al. “Comprehensive variation discovery in single human genomes”. In: *Nature genetics* 46.12 (2014), pp. 1350–1355.

[3] A. N. Salazar et al. “Nanopore sequencing enables near-complete de novo assembly of Saccharomyces cerevisiae reference strain CEN. PK113-7D”. In: *FEMS Yeast Research* 17.7 (2017), fox074.

[4] J. R. Tyson et al. “MinION-based long-read sequencing and assembly extends the Caenorhabditis elegans reference genome”. In: *Genome research* 28.2 (2018), pp. 266–274.

[5] D. Eccles et al. “De novo assembly of the complex genome of Nippostrongylus brasiliensis using MinION long reads”. In: *BMC biology* 16.1 (2018), pp. 1–18.

[6] W. Hamilton, Z. Ying, and J. Leskovec. “Inductive representation learning on large graphs”. In: *Advances in neural information processing systems* 30 (2017).

[7] J. Shi and C. Liang. “Generic repeat finder: a high-sensitivity tool for genome-wide de novo repeat detection”. In: *Plant physiology* 180.4 (2019), pp. 1803–1815.

[8] L. van Dijk. *aneusim - A tool to generate synthetic aneuploid/polyploid genomes*. original-date: 2017-07-13. Oct. 2018. URL: `https://github.com/AbeelLab/aneusim` (visited on 11/09/2021).

[9] B. K. Stöcker, J. Köster, and S. Rahmann. “SimLoRD: simulation of long read data”. In: *Bioinformatics* 32.17 (2016), pp. 2704–2706.

[10] H. Li. “Minimap2: pairwise alignment for nucleotide sequences”. In: *Bioinformatics* 34.18 (2018), pp. 3094–3100.

[11] D. Earl et al. “Assemblathon 1: a competitive assessment of de novo short read assembly methods”. In: *Genome research* 21.12 (2011), pp. 2224–2241.

[12] K. R. Bradnam et al. “Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species”. In: *Gigascience* 2.1 (2013), pp. 2047–217X.

[13] A. Hagberg, P. Swart, and D. S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[14] C. Data61. *StellarGraph Machine Learning Library*. `https://github.com/stellargraph/stellargraph`. 2018.

[15] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

5

HAT: HAPLOTYPE ASSEMBLY TOOL USING SHORT AND ERROR-PRONE LONG READS

Haplotypes are the set of alleles co-occurring on a single chromosome and inherited together to the next generation. Because a monoploid reference genome loses this co-occurrence information, it has limited use in associating phenotypes with allelic combinations of genotypes. Therefore, methods to reconstruct the complete haplotypes from DNA sequencing data are crucial.
Recently, several attempts have been made at haplotype reconstructions, but significant limitations remain. High-quality continuous haplotypes cannot be created reliably, particularly when there are few differences between the homologous chromosomes.

Here, we introduce HAT, a haplotype assembly tool that exploits short and long reads along with a reference genome to reconstruct haplotypes. HAT tries to take advantage of the accuracy of short reads and the length of the long reads to reconstruct haplotypes. We tested HAT on the aneuploid yeast strain Saccharomyces pastorianus CBS1483 and multiple simulated polyploid data sets of the same strain, showing that it outperforms existing tools.

This chapter has been published in Bioinformatics,
`https://doi.org/10.1093/bioinformatics/btac702`

5.1. INTRODUCTION

Most eukaryotes have more than one copy of each chromosome, and some species have more than two homologous copies of each chromosome (i.e., polyploids), which is common in plants [1]. In genetics, a haplotype is the combination of individual alleles (one allele of each gene) located on the same chromosome. Because these alleles are located on the same chromosome, they are passed on together to the next generation [2]. Haplotype assembly, or reconstruction refers to the task of reassembling each individual haplotype. The need to reconstruct haplotypes arises from the inability of current DNA sequencing technologies, such as next-generation (NGS) and third-generation (TGS) sequencing, to read a chromosome's sequence from beginning to end. These technologies instead sequence shorter fragments called reads. In addition, chromosome separation before sequencing requires complicated and expensive lab work that is not feasible for most studies. Therefore, it is more common to sequence chromosomes together, and then use computational methods to separate the reads and reconstruct the haplotypes. A monoploid reference genome consists of a mosaic structure of haplotypes with allelic combinations that do not co-occur within any haplotype. Additionally, some of the alleles found in the haplotypes are absent in the monoploid reference. In contrast, with a haplotype-resolved reference, we can understand genetic variation and link phenotypic traits with the associated alleles in the haplotype better.

It is significantly more challenging to reconstruct haplotypes for polyploid genomes than for diploid genomes. If one of the haplotypes of a diploid genome has been phased (i.e., the said haplotype has been inferred), it is trivial to determine the alleles of the other haplotype based on this. On the other hand, in polyploid genomes, other haplotypes may have the same or different alleles [3]. Hence, the phasing of one haplotype does not clearly indicate what alleles are present in other haplotypes.

Recognizing the wide-spread use of NGS and TGS, it is imperative to develop algorithms for polyploid haplotype reconstruction from sequencing reads to facilitate various research applications such as finding compound mutations that cause a disease [4], or studying yield related markers that are located in a haplotype that can be used in plant breeding programs [5]. In the past years, few tools have tackled this problem. nPhase [6] and Whatshap [7] are among some examples of recently developed tools.

This study presents HAT, a haplotype assembly tool that combines short reads and error-prone long reads along with a reference genome to reconstruct haplotypes. Similar to Ranbow [8], HAT first creates seeds from short reads, but then it expands the seeds with long reads. We benchmark HAT against Whatshap and nPhase because both use long reads to phase haplotypes. Using simulated and real yeast genome data, we demonstrate that HAT outperforms both Whatshap polyphase and nPhase in terms of contiguity and the accuracy of phased alleles.

5.2. METHODS

5.2.1. DATA

Using both simulated and real data is essential to test HAT properly. Simulated data provides the ground truth of haplotypes to evaluate phasing accuracy and the real data validates HAT's performance.

SIMULATED DATA

We use Haplogenerator [9] to simulate haplotypes from the base genome – chromosome ScII of *Saccharomyces pastorianus* CBS1483 with accession number ASM1102231v1 (ChrSc2) [10]. The ground truth is the simulated haplotypes, and ChrSc2 base sequence is the reference. Next, we simulate reads with 20x coverage per haplotype, similar to the simulation design used in previous studies including nPhase. We use Badread [11] version 0.2.0 and and ART [12] version 2.5.8 to simulate reads similar to Oxford Nanopore Technology (ONT) and to Illumina's HiSeq 2500, respectively. Badread is used with default parameters, ART parameters are available in Supplementary Table D.2. Supplementary Table D.4 shows the simulated ONT reads' error rates and compares them to the real data. In total, 6 datasets are generated for ploidy levels 3,4, and 5, with low and high heterozygosity. For the low heterozygosity datasets, we set the parameters of Haplogenerator to produce the same number of SNPs/Insertions/Deletions as the chromosomes ScII, SeIII-ScIII, and ScVIII of CBS1483 which are triploid, tetraploid, and pentaploid respectively. Because chromosomes SeIII-ScIII and ScVIII are smaller than ScII, the chromosome we use for the simulations, we multiply the number of SNPs/Insertions/Deletions by the ratio of genome sizes. For the high heterozygosity datasets, we fit a lognormal distribution on the distances [9] between consecutive SNPs/Insertions/Deletions of chromosome ScII, SeIII-ScIII, and ScVIII and use the parameters on Haplogenerator. The parameter settings for Haplogenerator are in Supplementary Table D.1.

Then, the short reads are mapped to the monoploid reference genome using BWA-MEM (Li, 2013) with default parameters. We obtain variations using FreeBayes [13] version 0.9.21 from the short read alignments. We use vcffilter from vcflib [14] package version 1.0.2 to extract the SNPs. We map the long reads using minimap2 [15] version 2.13-r858-dirty. Parameters for all tools are in Supplementary Table D.2. The short read and long read mapping, ploidy of the chromosome, and the SNPs are the inputs for HAT.

REAL DATA

We reconstruct the haplotypes of CBS1483, which is aneuploid and has ploidy ranging from one to five. It consists of ONT and paired-end Illumina reads that are available under the BioProject PRJNA522669. There are 4 ONT runs in this BioProject, and we used all of them in this study. Short reads have coverage of 159x and are 151bp, Nanopore reads have coverage of 72x with an average read length of 7kbp and N50 of 10kbp. We use the ASM1102231v1 assembly as the reference genome of CBS1483.

Moreover, we reconstruct the haplotypes of *Brettanomyces bruxellensis* strain GB54, a triploid genome which has higher heterozygosity, and longer chromosomes than CBS1483. The longest chromosome of GB54 is 4Mbp which is 3 times larger than CBS1483 largest chromosome. The ONT and paired-end Illumina reads are available under the BioProject PRJEB40511. The Illumina short reads are 75bp and 30x coverage. The nanopore long reads have the average read length of 12kbp, 82x coverage, and 23kbp N50. We use the DEBR_UMY321v1 assembly as the reference genome of GB54.
In both real datasets, the SNPs and the alignments are obtained with the same method as in the simulated datasets.

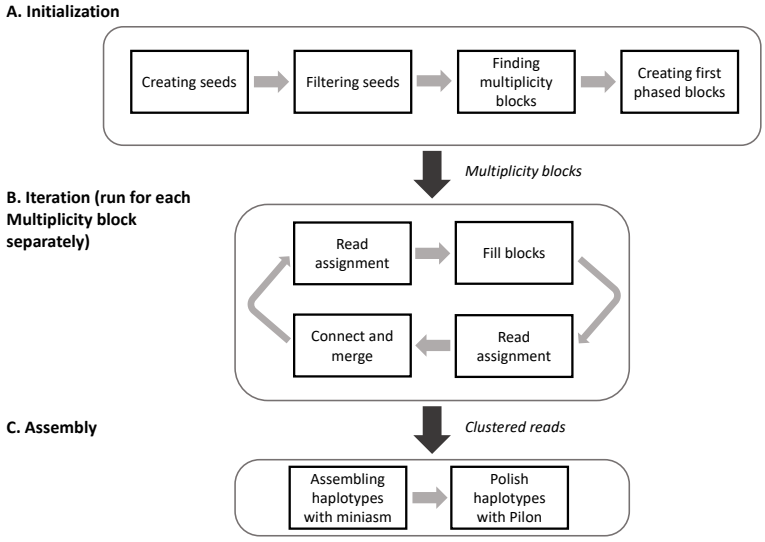


Figure 5.1: HAT overview. (A.) HAT creates seeds based on short read alignments and the location of SNPs. Then, it removes the combinations of alleles with low support as well as overlapping seeds. Next, HAT finds multiplicity blocks and creates the first phased blocks within them. (B.) HAT assigns reads to the blocks and haplotypes; based on these read assignments it fills the unphased SNPs within blocks. (C.) Finally, HAT can also use miniasm to assemble haplotype sequences for each block and polishes the assemblies using Pilon, but this step is optional

5.2.2. HAT METHOD

HAT reconstructs haplotypes by linking alleles at SNP loci together using short and long reads. HAT comprises three components - initialization, iteration and assembly. Initialization creates the first phased blocks. The iteration expands the phased blocks and finds alleles of all haplotypes. Then, HAT clusters the reads, and assembles haplotypes using these clustered reads. An overview of the HAT algorithm can be seen in Figure 5.1.

INITIALIZATION

In initialization (see Figure 5.1A), the multiplicity blocks are found, and then the first phased blocks are created. Phased blocks are a set of consecutive SNP loci in the phase matrix where the alleles are connected. First HAT creates seeds, which are a combination of consecutive SNP loci covered by the same short read; a single seed can be as small as two SNP loci. To create the seeds, we determine the SNP loci each short read is covering. If a read covers more than two SNP loci, we create all combinations of consecutive SNPs with different lengths and starting points. When we create a seed, based on the alleles present in the short reads that cover it, we obtain a set of combinations of alleles. While processing a short read, if the seed it would create already exists, no new seed is created and only the combination of alleles in the new read is added to the existing seed. In addition, we store the number of reads supporting each combination of alleles. Next, we filter the combination of alleles and the seeds. The combinations of alleles with fewer than five reads supporting are removed to remove erroneous combinations of alleles. If a seed ends up with fewer than two combinations of alleles, it is removed.

Next, HAT finds overlapping seeds and keeps only one of them because each SNP locus should be at most in one of the seeds to avoid conflicts. When HAT finds overlapping seeds, we check the number of combinations of alleles in each seed, the support of the seeds, and the first SNP locus of the seeds, then HAT picks the seed with the maximum number of combinations of alleles. If there are two overlapping seeds with the same number of combinations, we pick the longest one.
Then, HAT detects the regions that contain at least two different haplotypes, which we call multiplicity blocks. We use the sorted set of seeds as the input for Algorithm 1 to find multiplicity blocks and their corresponding multiplicity. The seeds are sorted with respect to the number of combinations of alleles, and when two seeds have the same number, the one with an earlier position of the first SNP locus will come first. Once Algorithm 1 has identified the multiplicity blocks, if the estimated multiplicity of the block exceeds the ploidy of the chromosome, it is decreased to the ploidy of the chromosome. In such cases, HAT eliminates combinations of alleles with low support until each seed has the same number of combinations as the chromosome's ploidy. HAT creates a separate phase matrix for each multiplicity block. Each row of the phase matrix corresponds to one of the haplotypes, and each column is representative of an SNP locus within the multiplicity block.

Finally, HAT generates the first phased blocks. First, HAT removes the seeds with fewer combinations of alleles than the estimated multiplicity of the block. Next, we use the combinations of alleles of the seeds to fill the phase matrix in the columns the seed is covering. Each seed creates a separate phased block because the relation of combinations of alleles of different seeds is unclear to one another. The SNP loci that do not belong to any block are added to the closest block to them.

ITERATION

The iterative part of HAT (Figure 5.1B and Supplementary Figure D.1B) continues until there is only one block and the phase matrix is full, or if the blocks and the phase matrix stop updating. We run the first iteration with short reads and the rest with long reads.
An essential step of the iterative HAT algorithm is assigning short and long reads to haplotypes in blocks. Each stage of the iterative part uses these assigned reads. Therefore, after both *Fill blocks* and *Connect and merge* steps, reads are reassigned to the haplotype blocks based on the latest changes.
First, for every read we check the phased SNP loci that it covers within a block. If the combinations of alleles at those loci are unique for each haplotype, the read is assigned to the block. Then, the alleles of the read located at the phased SNP loci the read is covering within the block are compared with the alleles of each row of the phase matrix. The read is assigned to the haplotype if the Hamming distance to the row is less than hamming_parameter, which changes with each run of the algorithm. When assigning long reads to the haplotypes, the hamming_parameter is small (1) to accommodate sequencing errors. As the phasing algorithm proceeds, we increase hamming_parameter to 3 to be less strict with the assignments, because there are more shared phased SNP locus within the block and the read.
The next stage of the iterative component is connecting and merging consecutive phased blocks. To connect two blocks, we use reads assigned to both blocks. We iterate over the

```
Parameters: : d ;                                   /* distance_parameter */
Inputs:     : S ;                                   /* sorted set of seeds */
Returns:    : MB ;                                  /* Multiplicity blocks */
FindMultiplicityBlocks (S,d)
    cores ← [ ];
    foreach seed ∈ S do
        if length(cores) = 0 then
            new_cores ← [seed];
            cores.append(new_cores);
            continue;
        end
        closest_core ← The closest core to the seed;
        if Distance(seed, closest_core) ≤ d then
            if Alleles(seed) = Multiplicity(closest_core) then
                cores[closest_core].add(seed);
            end
        else
            new_cores ← [seed];
            cores.add(new_cores);
        end
    end
    Multiplicity_blocks ← [[ ] ∗ length(cores)];
    foreach seed ∈ S do
        closest_core ← The closest core to the seed;
        Multiplicity_blocks[closest_core] ← seed;
    end
    MB ← {};
    foreach m_b ∈ Multiplicity_blocks do
        l ← Most left position of m_b;
        r ← Most right position of m_b;
        m ← Maximum multiplicity of m_b;
        MB[(l,r)] ← m;
    end
    return MB;
```

Algorithm 1: Find multiplicity blocks algorithm. The find multiplicity blocks algorithm takes the sorted set of seeds and a distance_parameter as input, and it returns multiplicity blocks as output. The multiplicity blocks is a dictionary that with multiplicity blocks as keys and multiplicity of the region as values. Each multiplicity block has a start and end position as well as an estimated multiplicity for that region. The *Alleles* function in the algorithm gets a seed as input and returns the number of combinations of alleles the seed has. The *Multiplicity* function in the algorithm gets a list of seeds as input and returns the highest number of combinations of alleles within them.

blocks based on their location. If there is a one-to-one connection between all the haplotypes of two blocks with enough support, the blocks are merged, and the rows of the second block are switched so that the connected haplotypes of the first and second haplotypes are in the same row. Two haplotypes are connected if the number of reads supporting the connection is more than 1 in the first and second iteration, and 3 in the rest.
In the blocks' filling step, we use all reads assigned to haplotypes of a block as input and process them to find the allele of unphased SNPs within the block by a majority voting between the reads of the haplotype. If the number of the reads supporting the majority vote allele is greater than 2, the allele is assigned to the haplotype's SNP locus, and that cell of the phase matrix is filled. This phase might lead to some SNP loci being phased in some haplotypes but not in others. When iteration converges, HAT assigns long and short reads to haplotypes of each phased block using the read assignment module.

ASSEMBLY
Optionally, HAT can assemble the reads to reconstruct sequence of the haplotypes using miniasm [16] version 0.3-r179 and the clustered long reads, then polish the assemblies using Pilon [17] version 1.24 and the clustered short reads. This part of HAT is optional. HAT uses miniasm and Pilon with default parameters. Users can use HAT to only cluster the reads and create the phase matrix and then use a tool of their choice to reconstruct sequence of the haplotypes.

5.2.3. OUTPUT
HAT outputs the following files:

- A multiplicity block figure which illustrates the multiplicity blocks and their level over the chromosome.
- The clustered reads files which contain the IDs of clustered reads for the haplotypes of each phased block.
- The phase matrix file which lists the alleles of haplotypes within each phased block.
- The haplotype sequences within each phased block. This output is optional, and it is produced only if the user also requests assembly.

5.2.4. EVALUATING HAT
We run HAT version 0.1.7, nPhase version 1.1.10, and Whatshap polyphase version 0.19.dev161+g7660dcf from the polyploid-haplotag branch on the simulated datasets and compare the phasing and read clustering accuracy. Unlike HAT and nPhase, Whatshap polyphase does not cluster the reads by default and after phasing with Whatshap polyphase, we use Whatshap haplotag to cluster the reads for evaluation purposes. The parameters of Whatshap haplotag is mentioned in Supplementary Table D.2. To calculate phasing accuracy of HAT, first we find a one-to-one mapping between the haplotypes HAT identifies within each block and the real haplotypes. Then we compare the allele of each haplotype at the SNP loci from the phase matrix to the ground truth. We count the number of correct SNPs for all the blocks and haplotypes and calculate accuracy as the count of correct SNPs divided by the total number of SNPs within the multiplicity blocks. To calculate the phasing

Table 5.1: The percentage of SNPs that are inside multiplicity blocks

Dataset	Percentage of SNPs inside multiplicity blocks
Triploid low heterozygosity	92%
Triploid high heterozygosity	99%
Tetraploid low heterozygosity	87%
Tetraploid high heterozygosity	99%
Pentaploid low heterozygosity	78%
Pentaploid high heterozygosity	98%

accuracy of nPhase and Whatshap, we first find the haplotype which is the most similar to each cluster based on the cluster's and haplotype's alleles at the SNP loci the cluster covers. Then we divide the count of correctly phased SNPs by the total number of SNPs within the clusters.

Then, we assess the accuracy of read clustering. Since the simulated reads are already labeled with their native haplotype, we calculate the clustering accuracy by counting the number of reads clustered correctly. We also count the number of phased blocks to evaluate the reconstructed haplotypes' completeness.

In addition to simulated data, we investigate the haplotypes HAT creates for the real CBS1483 and GB54 data.

5.3. RESULTS

5.3.1. CONCEPTUAL OVERVIEW OF HAT USING THE EXAMPLE OF A TRIPLOID CHROMOSOME

To provide an overview of the HAT algorithm, we consider the triploid chromosome ScII of *Saccharomyces pastorianus* CBS1483 (ChrSc2), for a step-by-step discussion of HAT. The HAT algorithm consists of three main steps: (i) initialization, (ii) iteration and (iii) assembly. The input is a combination of both short and long reads, along with a reference genome. HAT will produce read clusters per haplotype when run in default settings. If the optional assembly parameter is supplied by the user HAT will also generate the haplotype sequences.

In the initialization step, HAT builds prototype phased blocks from seeds within multiplicity blocks. Phased blocks are fully resolved haplotype segments, while multiplicity blocks are genomic regions presenting sufficient variants for phasing and have an estimated ploidy associated. The initialization consists of three steps. First, HAT uses the alignment of short reads to the reference to find well-supported combinations of variant alleles, called seeds. In our example of ChrSc2, 528 SNPs were used to create 25335 combinations, which are filtered down to 119 by removing the combinations with low support (see Methods). Next, HAT constructs multiplicity blocks from the seeds with Algorithm 1. Finally, HAT uses seeds with matching number of combinations of alleles to create the first phased blocks within each multiplicity block. In the example of ChrSc2, HAT found 16 multiplicity blocks (see Supplementary Figure D.3).

During the iterative phase HAT processes each multiplicity block to phase the remaining

SNPs and create bigger phased blocks within a multiplicity block. It consists of two sections: (i) filling blocks, and (ii) merging blocks. Before running each section, HAT assigns reads to blocks and haplotypes based on the SNPs each read covers and their similarity to the phased SNPs. Supplementary Table D.3 shows how each step of the iterative algorithm improves the phasing of ChrSc2. The iteration stops when there is no improvement over the previous step. The first iteration uses the short reads, while the remaining iterations use long reads. In our experiments with real and simulated data, HAT converges in less than four iterations. Increasing the number of iterations for the short reads does not change the overall phasing performance, because the blocks are bigger than the linking range of the short reads.

Upon convergence, there are 23 phased blocks and only 23 unphased alleles from the SNP loci within the multiplicity blocks. We use miniasm on the long reads assigned to haplotypes of each phased block to assemble them. Then, we polish the assemblies with the short reads assigned to haplotypes using Pilon.

5.3.2. HAT OUTPERFORMS STATE-OF-THE-ART ON SIMULATED DATA

To evaluate HAT we use simulated datasets, consisting of short and long reads, and alignments to the haplotypes. Details of simulation are described in the Methods section. Summary statistics of the simulated data sets are reviewed in Table 5.2.

Figure 5.2: Multiplicity blocks HAT finds for chromosome ScI of CBS1483. The output of finding multiplicity blocks algorithm on real data, chromosome ScI of CBS1483. The long, black vertical lines at the bottom show the SNPs and their positions on the chromosome found by FreeBayes. From these SNPS, HAT finds the seeds shown in short, black vertical lines in panel above the SNPs. The seeds are placed vertically based on the number of combination of alleles they have, ranging from 1 to 6 (y axis). HAT uses these seeds to find multiplicity blocks, which are shaded regions encapsulating the seeds and the color of the region indicates the estimated multiplicity level. See the legend for the colors corresponding to different multiplicity levels. The green box covers the multiplicity block that contains the UIP3 gene.

Table 5.2: Descriptive statistics of the simulated datasets and ChrSc2, the base chromosome used for simulations.

Dataset	Ploidy	Simulated SNPs	# SNPs found by Freebayes	# short reads	# long reads
Triploid low heterozygosity	3	1230	687	194910	3295
Triploid high heterozygosity	3	6398	4143	194910	3441
Tetraploid low heterozygosity	4	1144	506	259880	4423
Tetraploid high heterozygosity	4	12072	7512	259880	4358
Pentaploid low heterozygosity	5	1606	504	324850	5433
Pentaploid high heterozygosity	5	17802	7232	324850	5394
CBS1483 chromosome ScII	3	–	528	428802	8051

We compare HAT to nPhase and Whatshap polyphase using the various metrics (see Methods); Table 5.3 summarizes the performance of the tools on the simulated data sets. First, we compare long read clustering accuracy. The number of long reads clustered incorrectly by HAT is lower than that of both nPhase and Whatsap for all ploidy levels: HAT's error rate ranges from less than 1% (triploid high heterozygosity) to 24% (pentaploid low heterozygosity), whereas for nPhase the range is 5% (tetraploid high heterozygosity) to 38% (pentaploid low heterozygosity) and for Whatshap it is 7% (tetraploid high heterozygosity) to 22% (triploid high heterozygosity).

For all datasets, HAT successfully phases at least 90% of the SNPs, and the accuracy is the highest at 98% for the triploid high heterozygous genome (last column in Table 5.3). In all datasets, Whatshap has the lowest accuracy and HAT has the highest. Note that the phasing accuracy of HAT is calculated only for the SNPs inside the multiplicity blocks, but the multiplicity blocks cover almost all of the SNPs on the chromosome, with the lowest coverage being 78% for the pentaploid low heterozygous genome (Table 5.1). Similarly, for nPhase and Whatshap, we calculated the phasing accuracy based on the clusters each tool generates.

As shown in Table 5.3, HAT phases fewer total SNPs than nPhase and Whatshap. That is because HAT does not attempt to phase the areas far from the seeds. A few reads cover both the core of the multiplicity block and these regions that are far, making phasing of these regions less reliable. That means the result of HAT can be incomplete, but it has higher accuracy because it only works in reliable regions.

To assess the phasing contiguity we checked the number of phased blocks in the HAT output, and report that for highly heterozygous cases HAT can phase almost all of the haplotypes. HAT creates 2 phased blocks for the triploid case and 3 phased blocks for the tetraploid one. For cases with low heterozygosity, HAT creates 15, 30 and 23 phased blocks for the triploid, tetraploid and pentaploid genomes. This is expected because these genomes are largely identical and it is not possible to connect the phased blocks. In contrast, we also note that for the highly heterozygous pentaploid dataset, HAT creates 33 phased blocks although it has 96% phasing accuracy, an outcome likely caused by the high ploidy level.

Table 5.3: HAT outperforms nPhase and Whatshap in phasing accuracy on simulated data. The first two columns show number the number of reads clustered and phased, third column lists the number of SNPs within multiplicity blocks that were phased correctly/incorrectly and unphased. The final column is the SNP phasing accuracy.

Dataset	**Tool**	**Read clustering error**		**Total reads phased**		**SNP phasing performance**			**Accuracy**[2]
		short[1]	long	short[1]	long	correct	incorrect	unphased	
Triploid low	HAT	14	65	5400	2122	1813	20	35	98%
	nPhase	–	307	–	1268	1936	379	0	84%
	Whatshap	–	225	–	1943	759	1215	0	61%
Triploid high	HAT	55	13	37291	2138	11895	185	19	98%
	nPhase	–	218	–	2829	12701	1464	0	90%
	Whatshap	–	680	–	3060	7106	4663	0	60%
Tetraploid low	HAT	135	187	2466	1580	1549	34	68	94%
	nPhase	–	297	–	1439	2023	335	0	86%
	Whatshap	–	254	–	2229	1434	538	0	72%
Tetraploid high	HAT	62	32	17968	2252	29039	493	37	98%
	nPhase	–	219	–	4053	28980	1990	0	94%
	Whatshap	–	287	–	4142	20550	7942	0	72%
Pentaploid low	HAT	545	518	2350	2141	1341	51	74	91%
	nPhase	–	662	–	1726	1714	287	0	86%
	Whatshap	–	348	–	2396	1803	547	0	76%
Pentaploid high	HAT	1041	266	9360	6804	31980	1479	224	95%
	nPhase	–	449	–	5264	35450	2676	0	93%
	Whatshap	–	708	–	5246	27929	7301	0	79%

[1] The short read cluster error was calculated only for HAT because nPhase and whatshap are designed specifically to cluster the long reads.
[2] Accuracy is defined in Methods.

Table 5.4: HAT results on CBS1483 real data. These chromosomes are a representative subset of all chromosomes of CBS1483.

Chromosome name	Ploidy	# SNPs	% phased regions	Alleles within blocks	
				Phased	unphased
ScI	3	619	54%	1384	60
ScII	3	528	14%	922	23
ScIV	3	1643	20%	3424	168
ScIX	2	195	10%	335	43
ScVIII	5	417	14%	687	4
SeI	2	21	<1%	8	0
SeVII-ScVII	3	341	4%	444	5

5.3.3. HAT SHOWS ROBUST PERFORMANCE ON REAL DATA

Since there are not many chromosome-level polyploid assemblies available, the disparity between simulated and real genomes can be significant. Hence, we are evaluating HAT on the real *Saccharomyces pastorianus* CBS1483 dataset to corroborate the results from the simulated datasets as well. CBS1483 is a valid test model because it is aneuploid and has various ploidies ranging from one to five. Additionally, the chromosomes are small and easy to investigate. We report read clustering and phasing results for seven chromosomes of CBS1483 representing various levels of ploidy, heterozygosity and length in Table 5.4. For the highly heterozygous chromosome ScI (see Figure 5.2), multiplicity blocks that HAT finds cover 54% of the whole sequence and within these blocks HAT phased 96% of the alleles. Although ScIV contains a large number of SNPs, it is the largest chromosome and all the SNPS are concentrated around the centromere and thus, the % of phased regions is lower. SeI, on the other hand, is one of the shortest chromosomes (185kb long) and there are very few SNPs, meaning that the haplotypes are identical in most positions on the chromosome. For that reason, HAT phases less than 1% of the chromosome.

We observe the haplotype sequences created by miniasm and polished by Pilon for the multiplicity block 153738,163604 in chromosome ScII (Figure 5.3). This multiplicity block is only 8kb long, and the estimated ploidy for that region is 2. To visually investigate the accuracy of haplotype reconstruction, we map the clustered reads to haplotype 1 and haplotype 2 reconstructed by HAT and view the alignment in Integrative Genome Viewer (iGV). Figure 5.3 depicts the alignment of clustered reads of CBS1483 Chromosome ScII to the sequence of the first haplotype. The reads that belong to each haplotype have matching alleles that can differentiate them from reads of other haplotypes. We, therefore, demonstrate that the HAT algorithm for read clustering and finding multiplicity blocks works on real data.

Previous studies shows that the UIP3 gene is removed in some of the haplotypes of CBS1483 chromosome ScI [10]. We investigate the same gene in HAT output by aligning the reads HAT clustered in the multiplicity block covering the positions from 169765 to 178549 where UIP3 is located (see Figure 5.2), to UIP3 sequence. As expected, only haplotype 2 reads align to the gene.

Finally, we test the performance of HAT on GB54, a triploid *Brettanomyces bruxellensis* yeast strain, which Abou Saada et al. also uses to evaluate nPhase [6]. GB54 is an interesting test set because it has longer chromosomes and the chromosomes are more het-

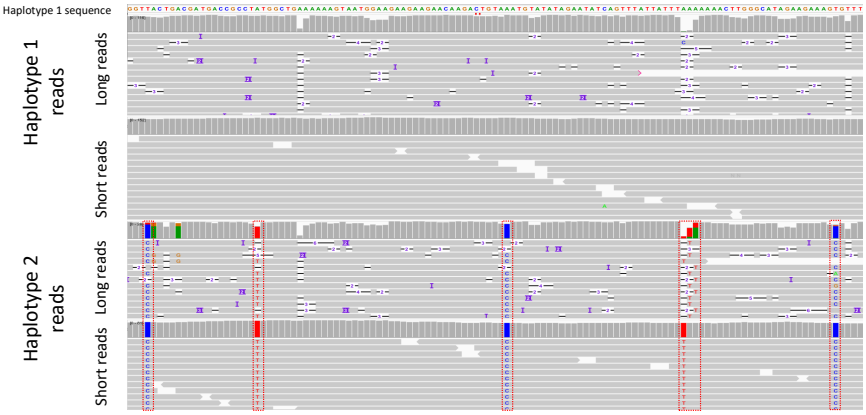


Figure 5.3: HAT can accurately cluster reads to reconstructed haplotypes. We aligned short and long reads to haplotype 1 (top two rows) and haplotype 2 (bottom two rows) phased by HAT for the multiplicity block covering the positions from 153738 to 163604 of ChrSc2 and visualized the alignment using iGV. Haplotype 2 reads differ significantly from haplotype 1 reads at five positions (outlined with red rectangles).

Table 5.5: HAT results on GB54 real data.

Chromosome	# SNP loci	% of phased regions	Alleles within blocks	
			Phased	Unphased
Chr 1	36628	84%	96764	2001
Chr 2	23756	82%	60066	2586
Chr 3	18902	86%	35728	1807
Chr 4	19377	89%	50616	884
Chr 5	10609	63%	17867	1673
Chr 6	15762	72%	32877	1295
Chr 7	3581	92%	10482	216
Chr 8	1327	72%	2949	305

erozygous compared to CBS1483. Table 5.5 shows HAT's performance in phasing GB54, and Supplementary Figure D.4 illustrates the multiplicity blocks HAT finds. As expected, the percentage of phased regions is much larger than that of CBS1483 (Table 5.4), since GB54 is more heterozygous. Additionally, when we visualize the multiplicity blocks in GB54 we observe multiple long regions in chromosome 2, 3, and 5 where two of the haplotypes are identical. For instance, on Chr 3 the genomic region from 909325 to 1172321, all of the seeds have only two combination of alleles, and the average ratio of the read support for combination of alleles of seeds at these regions is 1.7. This is in line with our expectation that two of the haplotypes are identical in this region, and we get on average near twice as much read coverage for one of the haplotypes as there would be if there were three different haplotypes. Morever, when Abou Saada et al. phased Chr 4 using nPhase they reported that two of the haplotypes were identical at the end region of Chr 4 since they could phase only two haplotypes. However, when we look at the same location on Chr

4, we observe two small genomic regions (from 1407542 to 1430976 and from 1533678 to 1546306) where HAT can successfully phase all three haplotypes (see Supplementary Figure D.4).

The running time of HAT depends on the size of the multiplicity blocks. The bigger the multiplicity blocks are, the more reads are assigned to them. The computational complexity of running HAT is currently $O(SKn^2)$ where S is the number of SNP loci, K is the ploidy of the chromosome, and n is the number of long reads within a multiplicity block. The process of phasing each multiplicity block can be run in parallel, but in the current implementation it is not parallelized. It took HAT less than an hour to run for each chromosome of Saccharomyces pastorianus, because it has small multiplicity blocks. However, the Brettanomyces bruxellensis took longer to run because it has longer multiplicity blocks (in some cases, chromosome-scale). It took HAT 24 hours to phase the longest chromosome of Brettanomyces bruxellensis, which is 3.5 Mbp. HAT memory usage is minimal; HAT uses less than 8GB of memory for all datasets. We executed HAT on a system with 16 cores of CPU and 32GB of memory. It is worth mentioning that HAT is a proof of concept implementation and not optimized for speed.

5.4. CONCLUSION

HAT is a haplotype assembly tool that reconstructs haplotypes and phases genomes using NGS and TGS data. It is impossible to phase entire homologous chromosomes when there are large variation deserts. To address this, HAT identifies regions where some of the haplotypes are identical so they are taken into account when phasing. We show that NGS and TGS provide enough information to phase high heterozygosity genomes on a chromosome-scale and more than 90% of the alleles in a low heterozygosity genomes.

We evaluate the performance of HAT on six simulated datasets based on an aneuploid yeast strain *Saccharomyces pastorianus* CBS1483, and compare it to nPhase and Whatshap, the state-of-the-art algorithms. We observe that HAT presents higher phasing accuracy, which results from starting with seeds created by accurate short reads. While all tools have decent performance in highly heterozygous genomes, HAT performs remarkably well in phasing and read clustering of low heterozygote genomes. However, in the latter case, haplotypes created by HAT are fragmented since it does not attempt to connect the multiplicity blocks, because there is not enough information to link them. While we did not evaluate HAT on any diploid dataset directly, we observed that HAT successfully phases blocks with multiplicity level of 2 which shows that it can also be applied to diploid genomes.

The value of the distance_parameter affects HAT's result. A smaller distance_parameter affects HAT's result, for example a smaller distance_parameter will lead to smaller multiplicity blocks. That means the final phasing is more accurate because only the seeds and the areas close to them are phased and connected. However, simultaneously, the final result is separated into more disjoint blocks because HAT considers multiplicity blocks disconnected from each other and never attempts to merge them. A larger distance_parameter will lead to larger multiplicity blocks, which means HAT connects areas that are further apart together. However, there will be less read support for some of these areas because they are far, meaning some areas remain unphased. On top of that, errors might affect the phasing if the distance_parameter is too large because there will be less read support, and the sequencing error might affect the majority voting. Based on our experience, the average read

length of the long reads is a good trade-off between accuracy and the block length.

The main limitation of HAT is that it uses of alignments of short and long reads to the reference. Similar to other haplotype assembly tools, HAT's performance greatly depends on the quality of this alignment and the subsequent variant calling. Moreover, HAT uses only the SNPs for phasing, thus it may not be able to reconstruct haplotypes in genomes with high levels of insertions, deletions, and structural variations. Meanwhile, we do not expect different long reads error rates to affect HAT's accuracy, since HAT starts with seeds created using NGS accurate reads, and requires high support at all steps.

Like all other reference-based haplotype reconstruction methods, HAT suffers from reference genome errors. Errors in the reference genome can lead to inaccurate variant calling, which immediately affects the haplotype reconstruction, as it is the primary source of information that HAT uses for the phasing. As an example, collapsed repeats can affect the ploidy estimation. The region with the collapsed repeat can have seeds with more combinations of alleles than the actual multiplicity of the region. That will create a small multiplicity block with a higher multiplicity than the region and lead to extra, wrong haplotypes for that region. However, it is worth mentioning that this region will be small because the seed with more combinations of alleles will not be joined with any other seed to create a more extended multiplicity block, and the multiplicity block will be around two times the distance_parameter.

Another potential limitation is that in rare cases, HAT may incorrectly assign a lower than actual multiplicity number. This occurs when there is a group of seeds in close proximity where the number of combinations of alleles in any of the seeds is smaller than the actual multiplicity level. When each seed is viewed separately, some of the haplotypes appear to be identical in that region. However, it is possible that these are different groups of identical haplotypes, and the ploidy level of the region may be higher if all of these seeds are viewed as a whole. This can be solved by creating seeds and identifying multiplicity blocks using long reads. However, considering all consecutive SNP loci in the reads as seeds requires significant computing power since each long read might cover hundreds of SNPs. Additionally, allelic combinations in the seeds may be affected by the high error rate of long reads. Another way to mitigate erroneous multiplicity assignment is to adjust multiplicity levels during the iterative part of HAT when long reads are used to phase the SNPs. In principle, by solving the mentioned problem it should be possible to create the seeds with HiFi reads, which will lead to longer multiplicity blocks and higher contiguity.

There are not many polyploid haplotype resolved genomes at the chromosome scale, which hinders the development of novel haplotype assembly algorithms. Hence, haplotype simulators are limited and the simulated haplotypes differ significantly from the real ones. We observed this when we compared the multiplicity blocks of real and simulated data (compare Supplementary Figure D.2 and Supplementary Figure D.3). There are many regions in the real data where the multiplicity level is smaller than the chromosome's actual ploidy level, contrary to simulated data. This might change with HiC reads since they provide long range information and link regions of the chromosome that are far apart. In addition to inconsistencies in the ploidy levels, the large variation deserts in CBS1483 genome cannot be simulated due to limitations of current haplotype simulators.

Although we demonstrate the performance of HAT on only two yeast strains *Saccharomyces pastorianus* CBS1483 and *Brettanomyces bruxellensis* GB54, HAT can also phase

different polyploid genomes. Since HAT performed consistently well on various levels of ploidy and heterozygosity, we expect our results to generalize to other genomes of varying ploidy and that HAT can readily be adopted to different use-cases. Moreover, we presume that HAT can find applications in metagenomics assembly since the haplotype and metagenomics assembly problems are comparable at the strain level. In metagenomics assembly, the goal is to reconstruct the genome of every single strain of the metagenomics community, which can be up to thousands of genomes. These strains, like haplotypes, are quite similar to each other. Furthermore, as a result of horizontal gene transfers, some of the species within the community share genomic content, complicating their read separation. Moreover, the sequencing coverage of strains varies significantly, which might lead to the underrepresented strains not being reconstructed in the assembly process.

Ultimately, HAT enables us to reconstruct haplotypes of polyploid genomes reliably, investigate the relationship of phenotypic features to the underlying haplotype alleles, and gain a better understanding of genetic diversity.

BIBLIOGRAPHY

[1] J. Ramsey and D. W. Schemske. "Pathways, mechanisms, and rates of polyploid formation in flowering plants". In: *Annual review of ecology and systematics* 29.1 (1998), pp. 467–501.

[2] D. C. Crawford and D. A. Nickerson. "Definition and clinical importance of haplotypes". In: *Annu. Rev. Med.* 56 (2005), pp. 303–320.

[3] S. Garg. "Computational methods for chromosome-scale haplotype reconstruction". In: *Genome biology* 22.1 (2021), pp. 1–24.

[4] S. B. Ng et al. "Targeted capture and massively parallel sequencing of 12 human exomes". In: *Nature* 461.7261 (2009), pp. 272–276.

[5] J. A. Bhat et al. "Features and applications of haplotypes in crop breeding". In: *Communications biology* 4.1 (2021), p. 1266.

[6] O. Abou Saada et al. "nPhase: an accurate and contiguous phasing method for polyploids". In: *Genome Biology* 22.1 (2021), pp. 1–27.

[7] S. D. Schrinner et al. "Haplotype threading: accurate polyploid phasing from long reads". In: *Genome biology* 21.1 (2020), pp. 1–22.

[8] M.-H. Moeinzadeh et al. "Ranbow: a fast and accurate method for polyploid haplotype reconstruction". In: *PLOS Computational Biology* 16.5 (2020), e1007843.

[9] E. Motazedi et al. "Exploiting next-generation sequencing to solve the haplotyping puzzle in polyploids: a simulation study". In: *Briefings in bioinformatics* 19.3 (2018), pp. 387–403.

[10] A. N. Salazar et al. "Chromosome level assembly and comparative genome analysis confirm lager-brewing yeasts originated from a single hybridization". In: *BMC genomics* 20 (2019), pp. 1–18.

[11] R. R. Wick. "Badread: simulation of error-prone long reads". In: *Journal of Open Source Software* 4.36 (2019), p. 1316.

[12] W. Huang et al. "ART: a next-generation sequencing read simulator". In: *Bioinformatics* 28.4 (2012), pp. 593–594.

[13] E. Garrison and G. Marth. "Haplotype-based variant detection from short-read sequencing". In: *arXiv preprint arXiv:1207.3907* (2012).

[14] E. Garrison et al. *Vcflib and tools for processing the VCF variant call format*. Tech. rep. Section: New Results Type: article. bioRxiv, 2021.

[15] H. Li. "Minimap2: pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18 (2018), pp. 3094–3100.

[16] R. Vaser et al. "Fast and accurate de novo genome assembly from long uncorrected reads". In: *Genome research* 27.5 (2017), pp. 737–746.

[17] B. J. Walker et al. “Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement”. In: *PloS one* 9.11 (2014), e112963.

6

GENOMES OF FOUR *Streptomyces* STRAINS REVEAL INSIGHTS INTO PUTATIVE NEW SPECIES AND PATHOGENICITY OF SCAB-CAUSING ORGANISMS

Genomes of four Streptomyces isolates, two putative new species (Streptomyces sp. JH14 and Streptomyces sp. JH34) and two non thaxtomin-producing pathogens (Streptomyces sp. JH002 and Streptomyces sp. JH010) isolated from potato fields in Colombia were selected to investigate their taxonomic classification, their pathogenicity, and the production of unique secondary metabolites of Streptomycetes inhabiting potato crops in this region. The average nucleotide identity (ANI) value calculated between Streptomyces sp. JH34 and its closest relatives (92.23%) classified this isolate as a new species. However, Streptomyces sp. JH14 could not be classified as a new species due to the lack of genomic data of closely related strains. Phylogenetic analysis based on 231 single-copy core genes, confirmed that the two pathogenic isolates (Streptomyces sp. JH010 and JH002) belong to Streptomyces pratensis and Streptomyces xiamenensis, respectively, are distant from the most well-known pathogenic species, and belong to two different lineages. We did not find orthogroups of protein-coding genes characteristic of scab-causing Streptomycetes shared by all known pathogenic species. Most genes involved in biosynthesis of known virulence factors are not present in the scab-causing isolates (Streptomyces sp. JH002 and Streptomyces sp. JH010). However, Tat-system substrates likely involved in pathogenicity in Streptomyces sp. JH002 and Streptomyces sp. JH010 were identified. Lastly, the presence of a putative mono-ADP-ribosyl transferase, homologous to the virulence factor scabin, was confirmed in Streptomyces sp. JH002. The described pathogenic isolates likely produce virulence factors uncommon in Streptomyces species, including a histidine phosphatase and a met-

This chapter has been published in BMC Genomics,
https://doi.org/10.1186/s12864-023-09190-y

alloprotease potentially produced by Streptomyces sp. JH002, and a pectinesterase, potentially produced by Streptomyces sp. JH010. Biosynthetic gene clusters (BGCs) showed the presence of clusters associated with the synthesis of medicinal compounds and BGCs potentially linked to pathogenicity in Streptomyces sp. JH010 and JH002. Interestingly, BGCs that have not been previously reported were also found. Our findings suggest that the four isolates produce novel secondary metabolites and metabolites with medicinal properties.

6.1. INTRODUCTION

In general, *Streptomyces* species are characterized by the production of interesting secondary metabolites; many of them are used for the treatment of a wide range of diseases. Therefore *Streptomyces* spp. are often considered a primary source of drug compounds [1–3]. In the environment, these metabolites may increase the fitness of *Streptomyces* spp. [4]. These natural compounds are involved in nutrient or niche competition, mutualism, and symbiotic relationships between the microorganisms and plants or insects [4–6].

Under laboratory culture conditions, however, *Streptomycetes* often only produce a small part of the secondary metabolites they can synthesize [7]. The discovery of metabolites by traditional methods requires the detection of these compounds in culturable conditions, reducing the chances of finding novel metabolites [2, 7]. Biosynthetic gene cluster (BGC) evaluation by genome mining and bioinformatics enables the identification and characterization of metabolites that cannot be found otherwise through traditional methods [7]. BGCs encoding secondary metabolites diverge between species and even strains [5, 7], likely due to acquisition through horizontal gene transfer or deletion [5]. These differences in BGCs often lead to adaptation of these microorganisms to the ecosystem, inducing lineage divergence by subsequent niche differentiation or antagonism [5]. Since BGCs are highly diverse at the strain level, even genome mining of strains belonging to the same species are key for the discovery of novel secondary metabolites [8].

Most *Streptomyces* species are saprophyte organisms and few have been described as plant pathogens [1, 9]. Pathogenic *Streptomyces* spp. are not host-specific and can infect potato tubers and taproot crops producing scab disease [10]. In these crops, pathogenic *Streptomyces* deteriorate tuber and root vegetable appearance decreasing their commercial value and causing high economic losses worldwide [11, 12]. Pathogenic *Streptomyces* species use different strategies to infect plants and to cause scab disease, including phytotoxic secondary metabolites, phytohormones, and secreted proteins [9].

Most studies aimed at understanding virulence mechanisms in scab-causing species have focused on strains that produce thaxtomin phytotoxins, including *Streptomyces scabiei* 87-22, *Streptomyces scabiei* EF-35, *Streptomyces europaeiscabiei* 89-04, *Streptomyces acidiscabies* 84-104, *Streptomyces stelliscabiei* NRRL B-24447, and *Streptomyces turgidiscabies* Car8. Among the virulence factors also identified in these pathogens are cytokinins, scabin, indole-3-acetic acid, concanamycins, coronafacoyl phytotoxins, Nec1 protein, TomA, ethylene, and suberinases [9, 13–21]. In contrast, little is known about the infection mechanisms employed by non-thaxtomin producing *Streptomyces* species. Virulence factors of *Streptomyces luridiscabiei*, *Streptomyces puniciscabiei*, *Streptomyces niveiscabiei*, and *Streptomyces reticuliscabiei* have not been stablished so far. Few virulence factors have been described for some non-thaxtomin producing pathogens, including Fridamycin E, FD-891, Borrelidin and non-diketopiperazine; however, their role in disease development remains unclear [22–25].

Recently, several *Streptomyces* isolates from potato crops in Colombia were characterized [26]. The authors identified several scab-causing isolates that did not produce thaxtomin A. Virulence factors responsible for the pathogenic phenotype in these organisms were not identified. Within these isolates, *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010, were identified as *Streptomyces xiamenensis* and *Streptomyces pratensis*, respectively, based on a multilocus sequence analysis (MLSA). In addition, two isolates (*Strepto-*

myces sp. JH14 and *Streptomyces* sp. JH34) could not be classified into specific taxa and were considered potentially new species [26]. From inoculation of sporulated isolates on potato tuber slices and radish seedling bioassays, *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010 were classified as pathogens [26]. However, the pathogenicity of *Streptomyces* sp. JH14 and *Streptomyces* sp. JH34 could not be established, as these microorganisms did not sporulate on ISP2 or oatmeal agar [26] or GYM (glucose, yeast and malt extract) agar.

In this study we wanted to establish the taxonomic classification of *Streptomyces* sp. JH14, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002 and investigate if JH14 and JH34 could be new species based on genomic data. In addition, we wanted to evaluate the pathogenic isolates to find potential virulence factors produced by these strains. Finally, we wanted to search for putative BGCs in the genomes of the four isolates, looking for potentially interesting metabolites. Our results highlight the importance of focusing scab disease research on non-thaxtomin-producing scab-causing species to provide new insights into the emergence of novel pathogenic *Streptomyces* species. In addition, our results contribute to the study of the diversity of *Streptomycetes* and may lead to the discovery of new medicinal compounds.

6.2. RESULTS

6.2.1. GENOME CHARACTERIZATION OF FOUR *Streptomyces* SPP

The genome assemblies of the isolates were nearly complete, with more than 98% of the single copy orthologs from the actinobacteria_odb9 BUSCO database represented and between 1 and 2 contigs representing each genome (Table 6.1). The genome sizes of *Streptomyces sp.* JH002, *Streptomyces sp.* JH34, *Streptomyces sp.* JH010, and *Streptomyces sp.* JH14 isolates ranged between 6.24 Mbp and 7.72 Mbp. *Streptomyces sp.* JH010 had the largest genome (7.72 Mbp). The genomic GC content of *Streptomyces sp.* JH002, *Streptomyces sp.* JH010, *Streptomyces sp.* JH34, and *Streptomyces sp.* JH14 isolates, ranged from 70.2% to 72%. All genome assemblies and annotations are available at Genbank under Genebank identifiers CP087989, JAJSOO000000000, JAJNMN000000000, and JAJNEL000000000, for JH002, JH34, JH010, and JH14, respectively. Raw sequencing data has been submitted to NCBI with BioProject PRJNA657491.

Table 6.1: Results of assembly, annotation, and completeness analysis for the genomes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010 and *Streptomyces* sp. JH14.

Isolate	**JH002**	**JH34**	**JH010**	**JH14**
Assembly				
Number of contigs	1	2	2	2
Contigs N50 (bp)	6,242,747	7,279,370	7,656,094	6,580,952
Assembled genome size (bp)	6,242,747	7,292,977	7,718,394	6,928,808
Coverage (X)	530	562	507	479
G⊕C (%)	72	70.9	71	70.2
BUSCO (actinobacteria_odb9)				
Complete and single copy	348	351	351	351
Complete and duplicate	0	0	0	0
Fragmented	2	0	0	0
Missing	2	1	1	1
Total BUSCO genes	352	352	352	352
BUSCO completeness (%)	98.9	99.7	99.7	99.7
Annotation				
Number of CDS	5,676	6,606	6,883	6,294
Hypothetical gene annotations (%)	31	34	33	33
CDSs classified into a subsystem (%)	35	34	34	35
Number of RNAs	72	88	85	85
Number of rRNAs (16S, 23S, 5S)	15	18	18	21
Genbank identifier	CP087989	JAJSOO000000000	JAJNMN000000000	JAJNEL000000000

The number of coding sequences (CDSs) predicted by the RAST server in the four isolates ranged from 5,676 to 6,883. About 34%-35% of CDSs found in the *Streptomyces* genomes were classified into a subsystem by the RAST server. Isolates showed to be very different among each other in terms of their metabolism (Table 6.2). There were important differences between the number of CDSs associated with each subsystem in the different isolates. Within the four isolates, *Streptomyces sp.* JH14 has the highest number of CDSs in the "Phages, Prophages, Transposable elements, Plasmids" subsystem (14 CDSs), "Cofactors, Vitamins, Prosthetic Groups, Pigments" (335 CDSs) and "Metabolism of Aromatic Compounds" (86 CDSs). *Streptomyces sp.* JH010 has the highest number of CDSs in the "iron acquisition and metabolism" subsystem (64 CDSs) and "stress response" (180 CDSs). In the latter subsystem, *Streptomyces sp.* JH34 also has a high number of CDSs (171 CDSs). In contrast, *Streptomyces sp.* JH14 has the lowest number of CDSs associated with "Dormancy and Sporulation" (2 CDSs), and no CDSs were classified into "Secondary metabolism". *Streptomyces sp.* JH002 contains the highest number of CDSs linked to "Secondary metabolism" (27 CDSs) and "Virulence, disease and defense" (68 CDSs) subsystems.

6.2.2. TAXONOMIC ANALYSIS SUGGESTS THAT JH34 AND JH14 ARE PUTATIVE NEW SPECIES

Pairwise comparison results between 16S rRNA sequences from *Streptomyces* sp. JH34 and EZBioCloud database showed 30 species with similarity values≥98.7% (see Table S2 in [27]), including two species with 100% of similarity (*Streptomyces* clavifer CGMCC 4.1604 and *Streptomyces* mutomycini NRRL B-65393). The 16S rRNA gene from *Streptomyces* sp. JH14 had a similarity≥98.7% to 36 species (see Table S3 in [27]). Within these species, *Streptomyces* yanii NBRC 14,669 had the highest 16S rRNA gene similarity value (99.9%).

Although the genomic sequences of all the species with 16S rRNA gene similarity values≥98.7% are supposed to be included for the calculation of ANI calculations [28], the genomes of several of these species are not available. Therefore, an MLSA was conducted to evaluate which species within the available genome sequences were closest to *Streptomyces* sp. JH34 and *Streptomyces* sp. JH14. The closest species to the isolates were selected for ANI calculation.

MLSA placed the isolate *Streptomyces* sp. JH34 in a well-supported clade (Bootstrap value=99%) along with *Streptomyces* pratensis ch24, 'Kitasatospora papulosa' NRRL B-16504 (considered as a member of S. pratensis [28]), *Streptomyces* atroolivaceus CGMCC 4.1405, and *Streptomyces* mutomycini NRRL B-65393 species, being more closely related to S. pratensis and 'K. papulosa' (Figure 6.1). MLSA results differed from the 16S rRNA similarity analysis, which indicated that *Streptomyces* sp. JH34 was most closely related to S. clavifer and S. mutomycini. Nevertheless, the ANI values confirmed MLSA results, being higher between *Streptomyces* sp. JH34 and S. pratensis ATCC 33,331 (92.23 and 'K. papulosa' (92.30%), than between *Streptomyces* sp. JH34 and S. mutomycini (89.24%), S. atroolivaceus (89.15%), and S. clavifer (86.18%). The ANI values between *Streptomyces* sp. JH34 and its closest relatives are lower than 95%, indicating that this isolate is a new species.

MLSA grouped the isolate *Streptomyces* sp. JH14 with *Streptomyces* yanii CGMCC

Table 6.2: Number of genes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010, and *Streptomyces* sp. JH14 isolates distributed by subsystem based on RAST annotation server.

Subsystem	**Number of genes**			
	Streptomyces sp. JH002	**Streptomyces sp. JH34**	**Streptomyces sp. JH010**	**Streptomyces sp. JH14**
Motility and Chemotaxis	5	7	5	6
Phages, Prophages, Transposable elements, Plasmids	6	6	6	14
Dormancy and Sporulation	6	12	12	2
Potassium metabolism	16	18	18	23
Sulfur Metabolism	25	63	56	72
Secondary Metabolism	27	9	4	0
Miscellaneous	31	42	39	45
Nitrogen Metabolism	33	27	30	29
Metabolism of Aromatic Compounds	34	60	54	86
Cell Division and Cell Cycle	35	34	37	34
Phosphorus Metabolism	42	38	43	50
Iron acquisition and metabolism	52	50	64	30
Regulation and Cell signaling	59	63	65	57
Virulence, Disease and Defense	68	46	43	55
Membrane Transport	101	113	109	96
RNA Metabolism	103	119	117	105
Cell Wall and Capsule	107	138	123	158
DNA Metabolism	110	124	130	115
Nucleosides and Nucleotides	113	122	133	124
Respiration	117	129	126	151
Stress Response	127	171	180	138
Fatty Acids, Lipids, and Isoprenoids	240	258	248	248
Cofactors, Vitamins, Prosthetic Groups, Pigments	258	287	269	335
Protein Metabolism	290	306	309	283
Carbohydrates	476	564	563	488
Amino Acids and Derivatives	500	591	644	553
Total number of genes	2981	3397	3427	3297

4.1146, *Streptomyces* sanglieri CGMCC 4.1146, *Streptomyces* gelaticus CGMCC 4.1444 and *Streptomyces* atratus CGMCC 4.1632 in a well-supported clade (Bootstrap=92%), with the three latter species distantly related from *S. yanni* and *Streptomyces* sp. JH14 (Figure 6.1). *Streptomyces yanni* was selected for ANI calculation as it is the closes relative to JH14 based on MLSA. However, it could not be calculated as the genome of *S. yanii* has not been sequenced. Consequently, even though some results suggest that *Streptomyces sp.* JH14 could be a new species, *S. yanni*'s full genomic information is needed to confirm this hypothesis based on genomic data.

6.2.3. SCAB CAUSING *Streptomyces* SP. JH010 AND *Streptomyces* SP. JH002 ARE PHYLOGENETICALLY DISTANT FROM OTHER PHYTOPATHOGENIC *Streptomyces* SPECIES

Our data show that *Streptomyces sp.* JH002 and *Streptomyces sp.* JH010 are distantly related to most of the scab-causing species (Figure 6.2). A phylogenetic analysis based on concatenated sequences of 231 single-copy core genes from pathogenic isolates *Streptomyces sp.* JH010 and *Streptomyces sp.* JH002 and well-known pathogenic *Streptomyces* species showed that these isolates belong to two further different lineages. The isolates *Streptomyces sp.* JH002 and *Streptomyces sp.* JH010 were grouped with the non-pathogenic species, *S. xiamenensis*, and *S. pratensis* (Bootstrap value=100%). Pathogenic *Streptomyces spp.* were mainly clustered in three well-supported clades (Bootstrap values=100%), two of them constituted by previously described pathogenic species (Clade 1 and Clade 2) (Figure 6.2). Most of the well-known *Streptomyces* pathogenic species are placed in clade 2, including *S. scabiei*, *S. acidiscabies*, *S. europaeiscabiei*, and *S. turgidiscabieis*. This clade groups all thaxtomin-producing species; however, it also contains species that do not produce this type of toxins (i.e., *Streptomyces reticuliscabiei*, *Streptomyces sp.* ST1015, and *Streptomyces sp.* ST1020). This clade is distantly related to clades 1 and 3. Clade 1 was constituted by two pathogens, *Streptomyces sp.* JH010 and *S. luridiscabiei* NRRL B-24455, and clade 3 contained only one pathogen, *Streptomyces sp.* JH002. This clade is the most distant clade from the well-known pathogenic *Streptomyces* species.

6.2.4. BIOSYNTHETIC GENE CLUSTERS

Using antiSMASH, we found several putative biosynthetic gene clusters in the genomes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010, and *Streptomyces* sp. JH14. The four isolates contain BGCs associated with the production of secondary metabolites with antimicrobial and antitumoral activities and iron chelators used for the treatment of different diseases (see Table 6.3). In addition, we found BGCs probably linked to the synthesis of novel natural compounds; in the genomes of the pathogenic isolates, we also found several BGCs that might be related to the pathogenesis of these organisms, including the BGC for ectoine, melanin, and several siderophores (i.e., Desferrioxamin B/E and coelichelin).

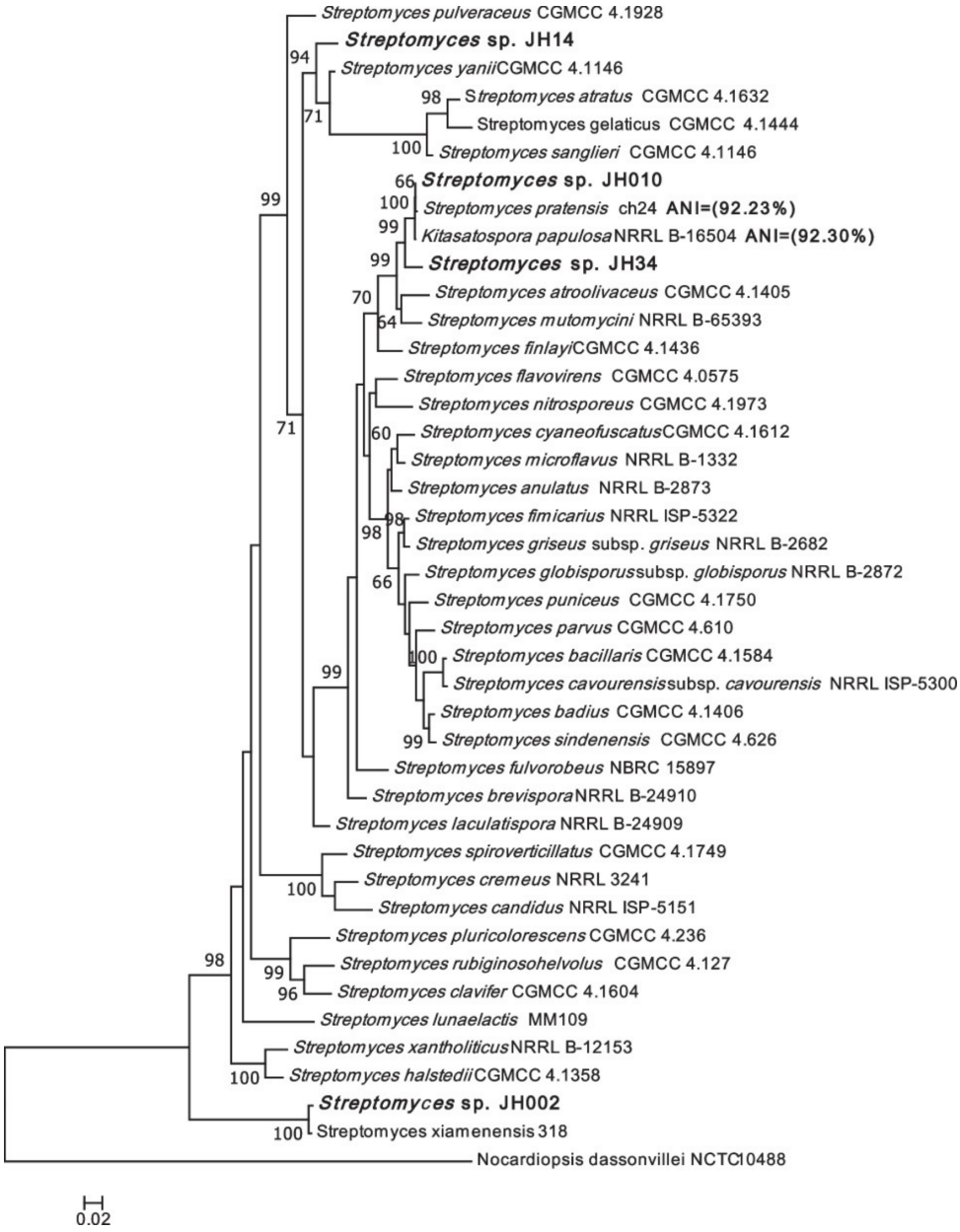


Figure 6.1: Phylogenetic analysis based on concatenated sequences of atpD, gyrB, recA, rpoB and trpB genes of the *Streptomyces* sp. JH34, *Streptomyces sp.* JH14, *Streptomyces* sp. JH002, and *Streptomyces sp.* JH010 and 37 *Streptomyces* reference strains. Phylogenetic tree was constructed using the ML method. *Nocardiopsis dassonvillei* NCTC 10,488 was chosen as the outgroup. The data were resampled 1000 times for Bootstrap test. Only bootstrap values higher than 60% are shown. As previously described [26] *Streptomyces sp.* JH10 and *Streptomyces sp.* JH002 belong to *S. pratensis* and *S. xiamenensis* species. ANI values between *Streptomyces sp.* JH34 and *S. pratensis* and 'K. papulosa' (closest relatives) are shown in parentheses. ANI value between JH14 and *S. yanii* (closest relative) could not be calculated as its genome has not been sequenced.

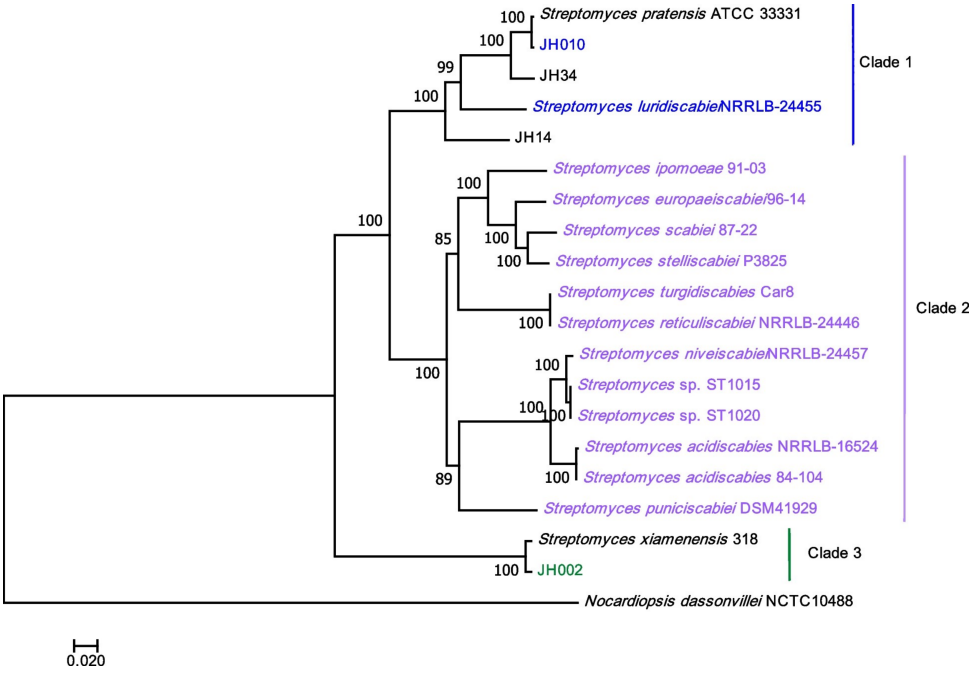


Figure 6.2: Phylogenetic analysis of *Streptomyces* species based on concatenated sequences of 231 single-copy core genes of isolates *Streptomyces* sp. JH002, *Streptomyces* sp. JH010, *Streptomyces* sp. JH14, *Streptomyces* sp. JH34, and 13 known pathogenic *Streptomyces* species. Pathogenic organisms are highlighted in the colors blue, purple, and green. Also, the type strains S. pratensis ATCC 33,331 and S. xiamenensis 318 were included in this analysis. The phylogenetic tree was constructed using the ML method. Nocardiopsis dassonvillei NCTC 10,488 was chosen as the outgroup. Data were resampled 1000 times for bootstrapping.

Table 6.3: Putative biosynthetic gene clusters found by antiSMASH from the genomes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010, and *Streptomyces* sp. JH14. Only clusters with similarity (percentage of genes with significant BLAST hit)≥60% are shown. Sm (%)=similarity percentage. NA=Metabolite function not fully established, or without therapeutic activity.

Isolate	Cluster name	Sm (%)	Secondary metabolite function
***Streptomyces* sp. JH002**	Ikarugamycin	84	Antimicrobial activity
	Ectoine	100	NA
	Moomysin	75	NA
	Nenestatin	66	NA
	Desferrioxamine B	60	Iron chelator
***Streptomyces* sp. JH34**	Isorenieratene	100	NA
	Desferrioxamine B/ E	100	Iron chelator
	Ectoine	100	NA
	Melanin	100	NA
	Sceliphrolactam	92	Antifungal metabolite
	Coelichelin	90	Iron chelator
	Spore pigment	83	NA
	Chromomycin A3	88	Antitumoral metabolite
***Streptomyces* sp. JH010**	Melanin	100	NA
	Ectoine	100	NA
	Ectoine	100	NA
	Isorenieratene	100	NA
	Coelichelin	90	Iron chelator
	Sceliphrolactam	88	Antifungal metabolite
	Spore pigment	83	NA
	Desferrioxamine B /E	83	Iron chelator
	Terpene	69	NA
	Carbapenem MM4550	65	Antimicrobial activity
***Streptomyces* sp. JH14**	Desferrioxamine B	100	Iron chelator
	Naringenin	100	Antimicrobial, anti-inflammatory, and antitumoral metabolite
	Amycomicin	100	Antimicrobial metabolite
	Hopene	84	NA

The genomes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010, and *Streptomyces* sp. JH14 contained 23, 27, 27, and 11 BGCs for secondary metabolites, respectively, based on antiSMASH annotation (see Tables S6-S9 in [27]). The BGCs predicted by antiSMASH comprised genes classified in several subsystems by RAST annotation server, including "secondary metabolism", "stress response", "iron acquisition and metabolism", "dormancy and sporulation", and "virulence, disease and defense" subsystems. Only between 22 and 36% of these clusters had similarity values$\geq$60% to known biosynthetic clusters.

Streptomyces sp. JH14 contains putative BGCs to produce the antibiotic amycomicin, flavanone naringenin and desferrioxamin B. *Streptomyces* sp. JH34 contains BGCs like those involved in the production of Chromomycin A3, desferrioxamin B, and sceliphrolactam. In *Streptomyces* sp. JH010 we also found a BGC associated with the production of sceliphrolactam. No differences were found between the clusters identified by antiSMASH in the genomes of *Streptomyces* sp. JH010 and *Streptomyces* pratensis ATCCC 33,331. The genome of *Streptomyces* sp. JH010 contains putative BGCs for ectoine and melanin, metabolites that may be associated with the pathogenicity of this isolate.

Streptomyces sp. JH002 has a gene cluster similar to the one for the production of the antibiotic ikarugamycin. The genome of *Streptomyces* sp. JH002 also contains four clusters that are not present in *Streptomyces* xiamenensis 318. Most of the clusters annotated in this genome had low percentages of similarity to known biosynthesis gene arrays (10%-75%). In JH002 we also found ectoine and desferrioxamin B BGCs that may be involved in the pathogenicity/virulence of this isolate.

6.2.5. FACTORS IN *Streptomyces* SP. JH010 AND *Streptomyces* SP. JH002

Orthologous gene analysis results revealed that *Streptomyces* species do not share unique orthologous gene clusters characteristic of pathogenic organisms (Figure 6.3). Furthermore, BlastP search showed that most virulence factors identified in the pathogenic *Streptomyces* species are not present in the genomes of *Streptomyces* sp. JH002 or *Streptomyces* sp. JH010.

Orthofinder assigned 192,325 genes, 94.2% of the identified genes, into 15,607 orthogroups. In total, 43 and 115 of the orthogroups identified in *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002, respectively, were specific for pathogenic *Streptomyces* species. *Streptomyces* sp. JH010 shared most orthogroups with S. scabiei 87.22 (11) and S. turgidiscabies Car8 (7), and *Streptomyces* sp. JH002 shared most orthogroups with S. acidiscabies (35) and *Streptomyces* sp. ST1020 (31) (Figure 6.3).

In *Streptomyces* sp. JH010, the orthogroups shared with other pathogens did not contain homologous genes implied in the pathogenicity/virulence of phytopathogenic organisms. In contrast, in JH002, we found two orthogroups for genes encoding proteins associated with the virulence of plant pathogenic bacteria, including a histidine phosphatase, and a metalloprotease [29–31].

Key proteins for the production of thaxtomins (thaxtomin synthases A and B) are not encoded in the isolates' genomes (Tables 6.4 and 6.5). Homologs to proteins required for the synthesis of other phytotoxins associated with the pathogenesis of scab-causing *Streptomyces* species were also not found. Within the proteins recognized as potential virulence factors in the *Streptomyces* genus, only a homolog of scabin was found in *Streptomyces* sp.

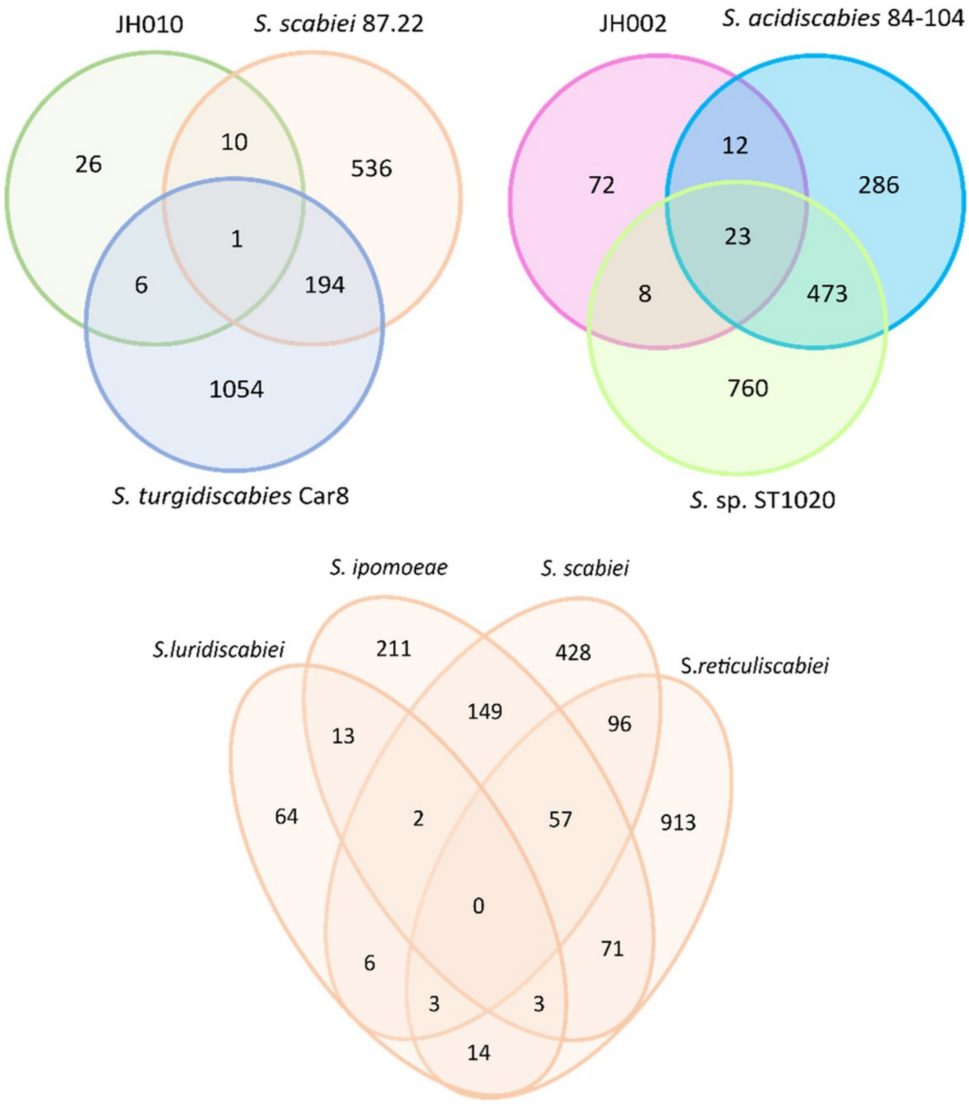


Figure 6.3: Venn diagram for orthogroups of protein-coding genes unique in pathogenic *Streptomyces* species.

JH002. In addition, homologs of the IAM hydrolase (iaaH gene) required for IAA production in the indole-3-acetamide pathway were found in both pathogenic isolates *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002; however, a Trp monooxygenase-like protein, necessary for IAA production in this pathway, was not found. The twin-arginine translocation (Tat) system was also found encoded in the genomes of both isolates; we found TatA, TatB, and TatC homologs.

Table 6.4: Blastp results. Proteins encoded in the genome of *Streptomyces* sp. JH002 with significant alignments to proteins involved in virulence mechanisms in pathogenic *Streptomyces* species are shown. Protein family was identified for each sequence using the HMMER server. Protein sequences with query cover≥80% are shown in bold.

Name	**Reference protein**		**Putative Protein encode in Streptomyces sp. JH002 genome**					
	Protein Family	**Length**	**Protein ID (Project ID PRJNA657491)**	**Protein family**	**Length**	**Query Cover (%)**	**Positive (%)**	**Identity (%)**
Trp monooxygenase	Flavin-containing amine oxidoreductase	565	H7827_04665	Flavin-containing amine oxidoreductase	470	23	41	31
			H7827_23025	Flavin-containing amine oxidoreductase	433	10	60	48
IAM hydrolase	Carbon-nitrogen hydrolase	262	H7827_03825	Carbon-nitrogen hydrolase	282	63	49	36
			H7827_06430	**Carbon-nitrogen hydrolase**	**267**	**93**	**48**	**32**
scabin	-	208	H7827_06625	-	**197**	**80**	**77**	**62**
TatA	mttA/Hcf106 family	97	H7827_02105	mttA/Hcf106 family	86	50	89	53
			H7827_26400	**mttA/Hcf106 family**	**101**	**96**	**68**	**49**
TatB	mttA/Hcf106 family	168	**H7827_09295**	**mttA/Hcf106 family**	**157**	**100**	**83**	**78**
TatC	Sec-independent protein translocase protein (TatC)	317	**H7827_26405**	**Sec-independent protein translocase protein (TatC)**	**299**	**100**	**67**	**52**

Table 6.5: Blastp results. Proteins encoded in the genome of *Streptomyces* sp. JH010 with significant alignments to proteins involved in virulence mechanisms in pathogenic *Streptomyces* species are shown. Protein family was identified for each sequence using the HMMER server. Protein sequences with query cover≥80% are shown in bold.

Protein	**Reference protein**		**Putative Protein encode in Streptomyces sp. JH10 genome**					
	Protein Family	**Length**	**Protein ID (Project ID PRJNA657491)**	**Protein family**	**Length**	**Query Cover (%)**	**Positive (%)**	**Identity (%)**
Trp monooxygenase	Flavin-containing amine oxidoreductase	565	H8R03_26010	Flavin-containing amine oxidoreductase	495	0	0	0
			H8R03_31270	Flavin containing amine oxidoreductase	421	7	66	42
IAM hydrolase	Carbon-nitrogen hydrolase	262	**H8R03_27235**	**Carbon-nitrogen hydrolase**	**280**	**100**	**46**	**31**
			H8R03_30185	**Carbon-nitrogen hydrolase**	**292**	**87**	**46**	**38**
			H8R03_05035	**Carbon-nitrogen hydrolase**	**265**	**100**	**83**	**74**
			H8R03_16235	**Carbon-nitrogen hydrolase**	**265**	**98**	**47**	**33**
TatA	mttA/Hcf106 family	97	H8R03_29530	mttA/Hcf106 family	79	57	69	55
			H8R03_06065	**mttA/Hcf106 family**	**97**	**96**	**75**	**65**
			H8R03_04485	**mttA/Hcf106 family**	**90**	**85**	**61**	**45**
TatB	mttA/Hcf106 family	168	**H8R03_04500**	**mttA/Hcf106 family**	**147**	**100**	**54**	**42**
			H8R03_21560	**mttA/Hcf106 family**	**162**	**100**	**64**	**53**
TatC	Sec-independent protein translocase protein (TatC)	317	**H8R03_06060**	**Sec-independent protein translocase protein (TatC)**	**319**	**100**	**72**	**58**

TAT-SYSTEM AND ITS EFFECT ON VIRULENCE

Homologs to TatA, TatB, and TatC, involved in the twin-arginine translocation (Tat) system, were found in the genomes of *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010. To evaluate if any Tat-transported substrates might be associated with the pathogenicity/virulence in *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010, we found several bona fide Tat-substrates using TATFIND 1.4 and TatP 1.0 servers.

Forty-two and sixty putative proteins secreted by the Tat-system were found in *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010, respectively, including several plant cell wall degrading enzymes (see Table S12 in [27]). In *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010 we found a putative endo-1,4-beta-xylanase A precursor and a putative endo-1,4-beta-xylanase, respectively. A putative aldose 1-epimerase was also found in both isolates. In *Streptomyces* sp. JH010 we also found three putative enzymes involved in the breakdown of plant cell wall, including several glycosyl hydrolases, and a pectinesterase [32, 33]. In *Streptomyces* sp. JH002 we found a putative rhamnogalacturonan lyase. In this strain we also found a lon-like protease, and a peptidase containing the S8/S53 domain.

SCABIN HOMOLOG IN *Streptomyces* SP. JH002

In the genome of *Streptomyces* sp. JH002, we found a scabin homolog (mART-JH002), which can be involved in pathogenicity. The prediction of the 3D structure of the mART-JH002, through LOMETS and RaptorX servers, revealed that this protein can be folded into a shape like other mART toxins. Results generated by LOMETS showed that 10 out of 10 servers predicted the crystal structure of scabin as the best 3D model for mART-JH002 with a coverage of 77-83%. Furthermore, the structure predicted by RaptorX showed that the putative 3D structure of mART-JH002 is similar to scabin (Figure 4). The quality scores of the predicted structure indicated that it has a correct fold (p-value=1.21E-10, Global Distance Test normalized (uGDT)=148, and the number of identical residues in the alignment normalized (SeqID)=50%). Moreover, SignalP 5.0 and SecretomeP 2.0 scores (SignalP 5.0 Likelihood=0.8342; SecP score=0.591) higher than 0.45, indicated that mART-JH002 is not associated with the bacterial membrane or cell wall and that mART-JH002 might be secreted following the signal peptide pathway [34].

Two out of three key active sites characteristic of mARTs are conserved in mART-JH002, including the Arg required for NAD+ binding and the Gln-X-Glu motif necessary for transferase activity (see Figure S1 in [27]). The third active site, commonly constituted by the Ser-Thr-[Ser-Gln-Thr] motif and involved in the scaffold of the NAD+ binding pocket formation [17], is replaced by Ser-Ala-Thr- motif in mART-JH002. Despite of the substitution of threonine by an alanine in this active site, this protein can still have DNA ADP-ribosyltransferase activity according to the molecular function predicted by COFACTOR server (Cscore=0.83).

The phylogenetic analysis showed that mART-JH002 is closely related to the Pierisin-like protein family with a high Bayesian support value (Posterior probability=1.0). Within this group of mARTs, mART-JH002 was relatively more closely related to mARTs from other *Streptomyces* species (scabin and ScARP) than to those described in other organisms (Figure 6.5). However, mART-JH002 was distantly related to scabin and ScARP.

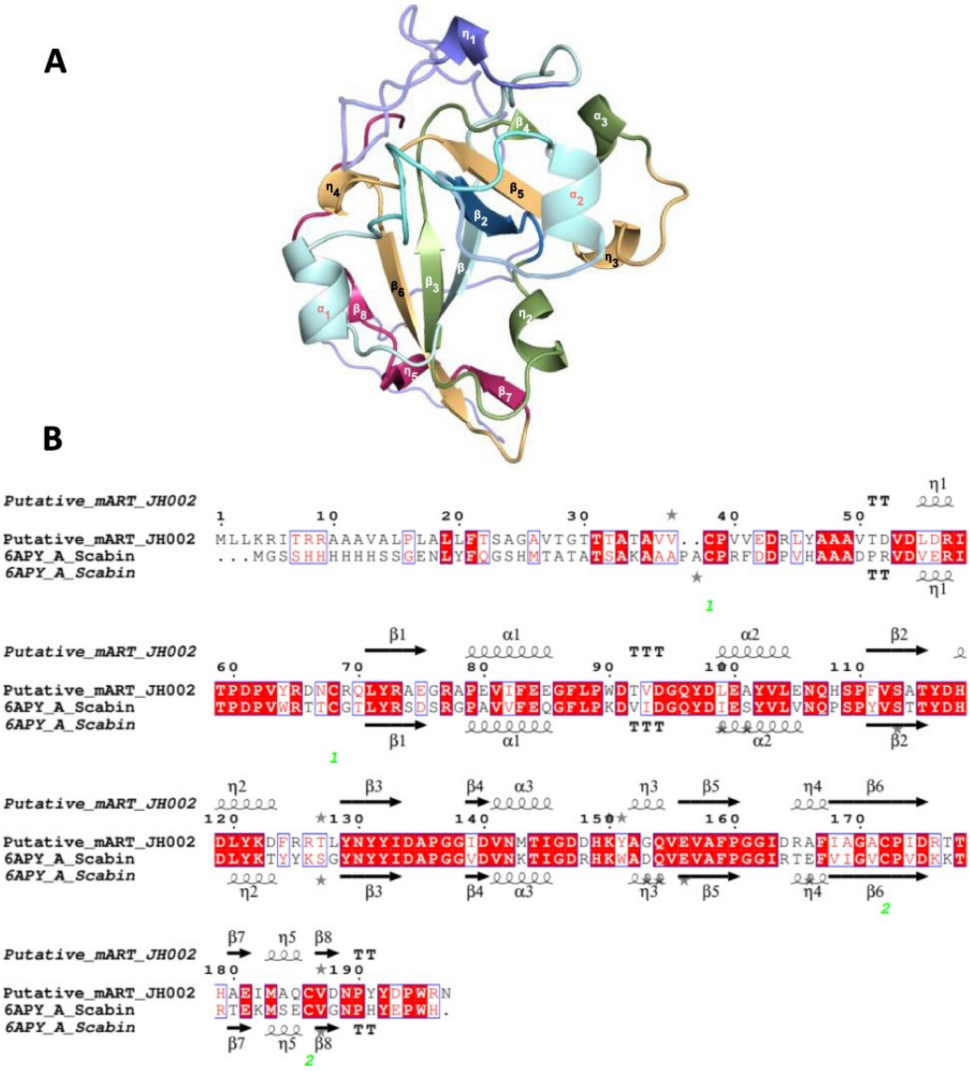


Figure 6.4: A 3D structure predicted in RaptorX from putative mART sequence. The image was obtained by using Chimera 1.15rc. B Alignment of putative mART-JH002 and Scabin sequences. Secondary structures of each protein are also shown. The image was obtained through ESPript 3.0 server under default parameters [35]. Identical residues are highlighted in red and similar residues are in blue frames. β strands are shown as arrows, α helices as squiggles, strict β-turns as TT and strict α-turns as TTT.

6.3. DISCUSSION

The genome sizes of *Streptomyces* sp. JH002, *Streptomyces* sp. JH34, *Streptomyces* sp. JH010 are consistent with the sizes of the genomes of other *Streptomyces* strains previously reported (ranging from 5.93 Mbp to 10.13 Mbp) [36, 37]. The genome of *Streptomyces* sp. JH010, which was identified as *Streptomyces pratensis*, is consistent with those described in

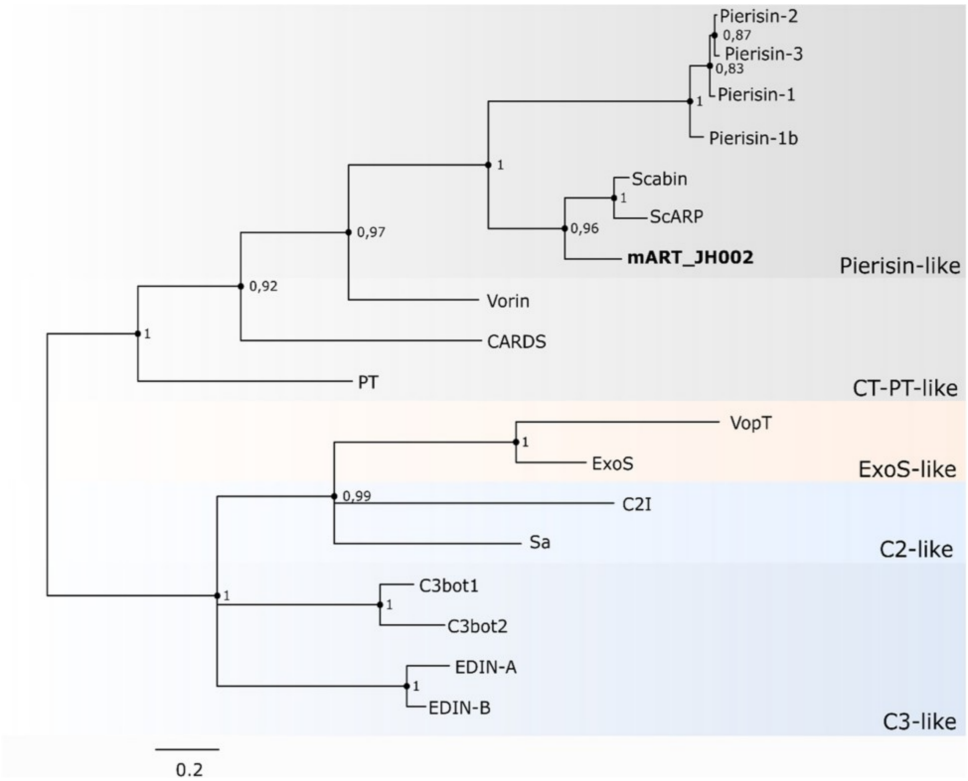


Figure 6.5: Phylogenetic tree of mARTs and putative mARTs. Tree was constructed by using Bayesian inference.

other *Streptomyces pratensis* strains (7.52-7.62 Mbp) [38]. Intraspecies genome size variations might be associated with the number of duplicate genes present in each organism. In fact, a positive correlation between the number of paralogues and genome size in several *Streptomyces* species has been previously reported [37]. The genomic GC content of *Streptomyces* sp. JH002, *Streptomyces* sp. JH010, *Streptomyces* sp. JH34, and *Streptomyces* sp. JH14 is consistent with the GC content reported for other *Streptomyces* species, usually over 70% [36, 37]. The number of coding sequences (CDSs) predicted by the RAST is consistent with CDS reported in other *Streptomyces* species (5,491-8,396) and is positively correlated with genome size, a hallmark of bacterial genomes [36, 37].

The phylogenetic relationships observed in this study were consistent with a previous MLSA constructed from sequences of more than 600 *Streptomyces* strains [28]. The fact that *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010 were grouped with the non-pathogenic species, S. xiamenensis, and S. pratensis with a Bootstrap value of 100%, confirms the previous taxonomic classification [26].

Based on our genome-based taxonomic analysis, we have good evidence that *Streptomyces* sp. JH34 is a putative new species and *Streptomyces* sp. JH14 is likely a candidate to be a putative new species. However, some species with high similarity values based on the 16S rRNA sequence to *Streptomyces* sp. JH34 and *Streptomyces* sp. JH14 were not

included in the MLSA, as their genome sequences are not available. These species are *Streptomyces sundarbansensis* MS1/7 closely related to *Streptomyces* sp. JH34 and *Streptomyces* sp. JH14 with 16S rRNA pairwise similarity of 98.96% and 98.89%, respectively, and Pilimelia columellifera subsp. pallida, closely related to *Streptomyces* sp. JH14 with 98.89% of 16S rRNA pairwise similarity. Also, the genomic sequence of *Streptomyces* yanni is needed for ANI calculation to evaluate if *Streptomyces* sp. JH14 is in fact a new species.

Although the hypothesis that JH14 and JH34 are new species cannot be confirmed based on the available genomic information alone, the phenotypic differences between these strains and their closest relatives make this hypothesis stronger. A previous characterization of *Streptomyces* sp. JH34 and *Streptomyces* sp. JH14 showed that these microorganisms do not sporulate in oatmeal agar or ISP2 medium [26]. However, the *Streptomyces* species most closely related to *Streptomyces* sp. JH34 and *Streptomyces* sp. JH14 can sporulate on these culture media. *Streptomyces* pratensis, and S. atroolivaceus produce spores on both oatmeal agar and ISP2, and S. mutomycini on oatmeal agar [39, 40]. Likewise, *Streptomyces* yanni can sporulate on oatmeal agar, *Streptomyces* gelaticus sporulates on ISP2 agar and *Streptomyces* gelaticus in both media [40, 41]. In addition, we compared the phylogenetic relationships obtained from the single-copy core genes and MLSA. Both phylogenetic analyses were consistent showing that *Streptomyces* sp. JH34 and *S. pratensis* are closely related and share a common ancestor. These results encourage the hypothesis that *Streptomyces* sp. JH34 is a new species.

It is interesting to find that Scab causing *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002 are not closely phylogenetically related to other scab causing *Streptomyces* spp. In fact, most of the potential virulence factors found in *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002 are different from those reported for other scab-causing *Streptomyces* spp. Based on PCR analysis, it was previously reported that genes involved in the synthesis of thaxtomins and the Nec1 protein, common in pathogenic *Streptomyces* species [9], were not found in *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002 [26]. The absence of genes related to the production of virulence factors described in most pathogenic *Streptomyces* species in *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010, as well as evidence that there are no shared orthogroups specific for scab between all well-known pathogenic organisms and *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010, suggest that the later *Streptomyces* species have gone through different evolutionary paths leading to the pathogenic phenotype. These species may have evolved mainly through horizontal transfer events. Indeed, horizontal transfer has been described as a key process in the evolution of plant pathogenic bacteria, leading to the adaptation of the bacteria to the host [42, 43]. The acquisition of mobile elements (i.e., phages, and integrative and conjugative elements) has enabled the adaptation of phytopathogens to a specific host and has been associated with the development of different symptoms in plants [44].

Streptomyces sp. JH010 and *Streptomyces* sp. JH002 contain interesting gene clusters associated with potential virulence factors. In particular, the genome of *Streptomyces* sp. JH010 contains putative BGCs for ectoine and melanin, metabolites that may be associated with the pathogenicity of this isolate. Ectoine and melanin protect pathogens from environmental changes generated during plant infection. Ectoine helps bacteria resist in environments with high osmolarity [44]. Melanin also plays an important role in the sur-

vival of microorganisms under adverse environmental conditions and has been implicated broadly in bacterial pathogenesis [45]. *Streptomyces* sp. JH010 also possesses gene arrays similar to siderophore BGCs (i.e., Desferrioxamin B/E and coelichelin). Siderophores in *Streptomyces* species trigger diverse biological processes, including growth, cellular differentiation, and the production of antibiotics [46]. These metabolites have also been associated with the pathogenic phenotype of several plant pathogenic bacteria [47]. Histidine phosphatases, which are potentially produced by JH 002, have been described as virulence regulators of Xanthomonas campestris pv. campestris [31]. Metalloproteases, also potentially produced by this strain, have been characterized in several pathogenic bacteria, including Pectobacterium carotovorum, Dickeya dadantii, and Xanthomonas campestris [30]. Although the role of these proteins in pathogenicity has not been fully elucidated, it has been proposed that histidine phosphatases might be involved in the breakdown of the plant cell wall or/and helping to counter the plant's immune response [30].

The Tat-system secretes different proteins associated with virulence in *Streptomyces scabiei* [48] and it was previously established that Δtatc *Streptomyces scabiei* (clone in which the gene tatc was mutated) was less virulent than the wild type. The Tat-system is known to be involved in virulence in *Streptomyces scabiei* [48] through the secretion of various proteins. We found that the twin-arginine translocation system is encoded in the genomes of both isolates, *Streptomyces* sp. JH010 and *Streptomyces sp.* JH002; we also found several interesting putative Tat substrates in the pathogenic isolates. Together, these two findings suggest that the Tat-system may be involved in the pathogenicity/virulence of these organisms, yet further experimental analyses are required to better evaluate this hypothesis. Of the over hundred putative proteins secreted by the Tat-system in JH002 and JH010, there were several plant cell wall degrading enzymes, which are frequently used in phytopathogenic organisms to make the host susceptible to infection and release nutrients during plant colonization [33, 49]. Endo-1,4-beta-xylanases A and endo-1,4-beta-xylanases, similar to those found in JH002 and JH010, respectively, are involved in xylan degradation, which is a structural polymer found in plant cells [50]. An aldose 1-epimerase from Phytophthora species, similar to those potentially produced by JH002 and JH010, has been showed to trigger cell death in Nicotiana benthamiama [51]. Other enzymes found in JH010, annotated as glycosyl hydrolases, and a pectinesterase, could be involved in the breakdown of plant cell wall [32, 33]. Rhamnogalacturonan lyases, similar to the one potentially produced by JH002, degrade rhamnogalacturonan I, a structural component of pectin in the cell wall of plants [52, 53]. Lon-like proteases, also potentially produced by this strain, are generally required in pathogenic bacteria for full virulence (i.e., *Pseudomonas syringae* and *Rhizobium radiobacter*) [30]. It was previously reported that the disruption of a gene encoding a protein belonging to the S8 peptidase protein family produced a decrease in the virulence of the fungal pathogen Penicillium expansum on apple fruit [54].

In the genome of *Streptomyces* sp. JH002, we found a scabin homolog (mART-JH002). Scabin is a mono-ADP-ribosyltranferase (mART) belonging to the Pierisin family [55]. These enzymes transfer an ADP ribose to DNA [56]. Pierisin-like toxins can induce cell apoptosis by labeling a guanine base with an ADP-ribose moiety [56, 57]. Although the role of scabin in the pathogenicity of S. scabiei has not been fully elucidated, it has been observed that scabin modifies DNA and shows a high affinity for the DNA of Solanum tuberosum [58], suggesting this scabin homolog may also play an important role in pathogenicity

in JH002. Considering that mARTs conserve the reaction mechanism and that mARTs are classified according to substrate target type [34], the divergence of mART-JH002 from other *Streptomyces* pierisin-like enzymes indicate that the targets of this protein might vary from those described for scabin (double or single stranded DNA) and ScARP (mononucleotides and nucleosides) [55, 59].

As expected, we found that all isolates contain gene clusters associated with production of interesting compounds, yet only few of the BCGs we found, had high similarity values to known clusters, suggesting the potential for identification of several novel metabolites in these isolates. Flavanone naringenin, one of the substances potentially produced by *Streptomyces* sp. JH14, has diverse therapeutic properties, including antimicrobial, anti-inflammatory, and antitumor activities [60, 61]; desferrioxamin B, slso potentially produced by this strain, is a siderophore used to treat iron overdose in humans [62]. Chromomycin A3, a metabolite potentially produced by *Streptomyces* sp. JH34 is an antitumoral substance [63] and sceliphrolactam, also potentially produced by this strain is an antifungal [64]. JH002 does not have the region that contains the genes associated with the production of xiamenmycin, an anti-fibrotic drug candidate known to be produced by S. xiamenensis 318 [37].

All isolates contain other interesting putative BGCs involved in the production of medicinal substances, including known antimicrobial, anti-inflammatory, and anti-tumoral metabolites; further analysis of these isolates, for example using metabolomic tools, could lead to the identification and isolation of novel compounds or compounds with medicinal and industrial uses.

6.4. CONCLUSIONS

Streptomyces spp. are very diverse, and there are still many unknowns regarding their pathogenicity and their capacity to produce medicinal substances. Especially in some countries in Latin America, known for their biodiversity, little is known about *Streptomyces* spp. Four strains of *Streptomyces* spp. previously isolated from potato fields in Colombia, were investigated in this study. Based on genomic data, and considering phenotypic differences with closest relatives, we were able to establish that *Streptomyces* sp. JH34 is likely new species. *Streptomyces* sp. JH14 could not be classified as a new species from ANI calculation, because its closest relative has not been sequenced; however, MSLA and phenotypic characteristics suggest it could be a new species as well. We confirmed previous findings that *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010 belong to *Streptomyces* pratensis and *Streptomyces* xiamenensis, respectively, and that they are phylogenetically distant from the most well-known pathogenic species. In fact, no orthogroups of protein-coding genes characteristic of scab-causing *Streptomycetes* were found in the pathogenic isolates and most of the genes involved in the biosynthesis of known virulence factors were also not found in these scab-causing isolates. However, we did find several Tat-system substrates that are probably involved in the pathogenicity of *Streptomyces* sp. JH002 and *Streptomyces* sp. JH010 as well as the presence of a putative mono-ADP-ribosyl transferase, a homolog to scabin, in *Streptomyces* sp. JH002.

We found BGCs for secondary metabolites associated with pathogenicity in the pathogenic isolates (*Streptomyces* sp. JH010 and *Streptomyces* sp. JH002) and BGCs associated with the synthesis of interesting medicinal compounds, including antibiotics,

antifungal, and antitumoral substances, in all isolates. Our results provide new insights about pathogenicity in *Streptomyces* species, highlighting the importance of focusing scab disease research on non-thaxtomin-producing scab-causing species and highlights the key role of horizontal transfer in the emergence of new scab-causing organisms. Our results may also contribute to the discovery of new therapeutic agents.

6.5. METHODS

6.5.1. MICROBIAL ISOLATES

The *Streptomyces* species analyzed (*Streptomyces* sp. JH34, *Streptomyces* sp. 14, *Streptomyces* sp. JH010 and *Streptomyces* sp. JH002) were isolated in the department of Cundinamarca, Colombia, from potato tubers. The isolates were phenotypically characterized in a previous study and are deposited at the Museo de Historia Natural ANDES [26].

6.5.2. DNA ISOLATION

Cultures of isolates *Streptomyces sp.* JH34, *Streptomyces sp.* 14, *Streptomyces sp.* JH010 and *Streptomyces sp.* JH002 were grown in 100 mL ISP2 broth ((Dextrose (4 g/L); Yeast Extract (4 g/L); Malt Extract (10 g/L); pH 7.0-7.2) [65]) for 5 days at 30 °C in constant shaking (250 rpm). After growth, cultures were centrifuged at 11,000×g for 15 min. The supernatant was carefully removed, and *Streptomyces mycelia* were recovered and used for DNA isolation using the DNeasy PowerSoil Kit following the manufacturer's protocol with the following modifications: i) approximately 0.20 g of mycelium sample was added to the PowerBead Tube instead of a soil sample; ii) three mycelia samples for each isolate were processed separately up until the addition of solution C4, a highly concentrated salt solution used in the DNA isolation in the PowerSoil Kit. Then, the three samples were loaded into the same MB spin column. Washing and elution steps were carried out according to the manufacturer's protocol.

6.5.3. GENOME SEQUENCING, ASSEMBLY, AND ANNOTATION

We sequenced, assembled, and annotated the genomes of four *Streptomyces* isolates (two pathogens and two putative new species) isolated from potato fields in Colombia. Whole-genome sequencing of the four isolates (*Streptomyces sp.* JH002, *Streptomyces sp.* JH34, *Streptomyces sp.* JH010, and *Streptomyces sp.* JH14) was carried out at the University of Minnesota Genomics Center using Single-Molecule Real-Time (SMRT) Pacific Bio-Sciences (PacBio) technology. Samples were sequenced in one Sequel SMRT Cell 1 M v3. Demultiplexed data was provided and used for de novo assembly of the genomes of *Streptomyces sp.* JH002, *Streptomyces sp.* JH34, *Streptomyces sp.* JH010, and *Streptomyces sp.* JH14; for this, we used the Flye assembler 2.6 [66] using plasmid flag and three polishing iterations; the remaining parameters were set to default. Genome assembly completeness was analyzed by assessing the presence of single-copy ortholog genes using BUSCO 3.01 [67]. The genome sequences obtained were compared to Actinobacteria genes from the OrthoDB database (actinobacteria_odb9). After assembly and BUSCO analyses, we annotated the four genomes on the RAST 2.0 annotation server by the ClassicRAST scheme [68] using default parameters. Finally, Barrnap v.0.9 (https://github.com/tseemann/barrnap) was used to determine the number of rRNAs in each genome.

6.5.4. TAXONOMIC CLASSIFICATION OF *Streptomyces sp.* JH34 AND *Streptomyces sp.* JH14 ISOLATES FROM GENOME DATA

Taxonomic classification of *Streptomyces sp.* JH34 and *Streptomyces sp.* JH14 was conducted from the calculation of the Average Nucleotide Identity (ANI) between the isolates and their closest relatives, because ANI differentiates closely related species based on a comparison of genome sequences [69]. To identify the species close to *Streptomyces sp.* JH34 and *Streptomyces sp.* JH14, similarity values between 16S rRNA sequences of the isolates and 16S rRNA sequences available on the EZBioCloud 16S database (https://help.ezbiocloud.net/ezbiocloud-16s-database/) were obtained by pairwise comparison [69]. Species with similarity values$\geq$98.7% are chosen for ANI calculation [69]. Here, we identified the closest species to *Streptomyces sp.* JH34 and *Streptomyces sp.* JH14, based on Multilocus Sequence Analysis (MLSA) of the concatenated sequences of five housekeeping genes (atpD, gyrB, recA, rpoB, trpB). MLSA has shown a high resolution in the differentiation of close *Streptomyces* species [28]. All the species with 16S rRNA similarity values$\geq$98.7% [69] were selected for the MLSA. The gene sequences for the isolates were obtained from genome assemblies, and the sequences for the reference *Streptomyces* species were retrieved from the NCBI database. The homologous sequences for each housekeeping gene were aligned by using Multiple Sequence Alignment (MUSCLE) [70] and trimmed manually to the same position by using MEGA7 [71]. The resulting alignments were joined head-to-tail in a frame, obtaining 2532 bp sequences, including gaps. Subsequently, the phylogenetic tree was constructed using the Maximum Likelihood (ML) method and the GRT$\oplus$G$\oplus$I substitution model in MEGA7 [71]. Pairwise distances were calculated under default parameters. The confidence of the phylogenetic tree and the pairwise distance calculation was estimated by bootstrapping method, resampling the sequences 1000 times. In total, 36 species were included in MLSA, and Norcadopsis dassonvillei NCTC 10,488 was chosen as the outgroup. Genbank accession numbers of housekeeping genes for all strains included in MLSA are shown in Table S1 in [27]. After conducting the MLSA, the closest species to *Streptomyces sp.* JH34 and *Streptomyces sp.* JH14 were chosen based on the phylogenetic analysis results. ANI values between the chosen species and *Streptomyces sp.* JH34 and *Streptomyces sp.* JH14 were obtained by using the ANI Calculator on the EZBioCloud platform [72]. Accession numbers of *Streptomyces* species assemblies used for ANI calculation are shown in Table S3 in [27].

6.5.5. PHYLOGENETIC ANALYSIS

Phylogenetic analysis was conducted based on concatenated sequences of 231 single-copy core genes of the pathogenic isolates *Streptomyces sp.* JH002 and *Streptomyces sp.* JH010, their closest relatives (*S. pratensis* and *S. xiamenensis*), and previously reported pathogenic *Streptomyces* species. The set of single-copy core genes was selected by comparison of the gene identifiers obtained after RAST annotation. Sequences of the homologous genes were aligned using MUSCLE, and the alignments were cleaned by G-block implementation to improve the phylogenetic reconstruction [73]. Subsequently, phylogenetic tree topology was constructed based on aligned sequences using the Maximum Likelihood method with the RAxML program on CIPRES Science Gateway [74, 75]. The data was resampled 1000 times for bootstrap analyses, and the GRTGAMMA model was used as the substitution model. The *Streptomyces* species included in the analysis and the accession numbers of

genome assemblies are shown in Table S5 in [27].

6.5.6. SEARCH FOR PUTATIVE BIOSYNTHETIC GENE CLUSTERS (BGCS)

Biosynthetic gene clusters for secondary metabolites encoded in the genomes of the four isolates (*Streptomyces sp.* JH14, *Streptomyces sp.* JH34, *Streptomyces sp.* JH010, and *Streptomyces sp.* JH002), *S. pratensis* ATCC 3333, and *S. xiamenensis* 318 were identified using the antiSMASH 5.0 online tool [76]. The two latter strains are considered saprophytic bacteria; however, they are the closest phylogenetically related strains to the pathogenic isolates (JH002 and JH010). Hence, the BGCs found in the pathogenic isolates and their closest relatives were compared to determine differences in secondary metabolism of these microorganisms.

6.5.7. INVESTIGATION OF POTENTIAL VIRULENCE FACTORS IN *STREPTOMYCES SP.* JH002 AND *STREPTOMYCES SP.* JH010 GENOMES

Here we aimed to find genes that might be involved in the pathogenesis of these isolates by using two different approaches: (i) search for orthogroups of protein-coding genes unique in pathogenic *Streptomyces* species through Orthofinder, and (ii) identification of homologs of putative proteins involved in the synthesis of virulence factors commonly described in *Streptomyces* species through BlastP.

Orthofinder v2.4.0 with default parameters was used to obtain orthogroups for protein-coding genes from the genomes of pathogenic and non-pathogenic species [77]; specifically, it was used to find orthogroups from pathogenic species that are absent in non-pathogenic organisms. Subsequently, one protein sequence from each group was chosen randomly, and homologs were determined through BlastP (Version 2.11.0) search on NCBI under default parameters [78]. The *Streptomyces* species and accession numbers of genomes included in the Orthofinder analysis are shown in Table S5 in [27].

Sequences of proteins involved in the biosynthesis of virulence factors that have been described in pathogenic *Streptomyces* species were retrieved from the NCBI database and searched in the genome annotation of the pathogenic isolates using BlastP 2.5.0 with default parameters and E-value cut-off of 1e-4. Query cover, identity, and positive substitutions were obtained by BlastP on NCBI. Sequences from isolates with query cover$\geq$80% and identity$\geq$40% to protein sequences involved in virulence in pathogenic species were chosen for further analysis. Protein sequences chosen were analyzed with the HMMER 3.3.2 webserver to confirm their putative function [79]. Table S5 in [27] shows accession numbers for the protein sequences retrieved from the NCBI database.

Finally, Tat substrates homologous to proteins involved in the pathogenesis of phytopathogenic organisms were found. Putative proteins secreted through the Tat-system were predicted through TATFIND 1.4 [80] and TatP 1.0 servers [81] under default parameters. Only the proteins predicted by both servers were considered as bona fide Tat substrates. The function of bona fide Tat substrates was obtained from RAST annotation.

6.5.8. ANALYSIS OF PUTATIVE MART TOXIN ENCODED IN THE *Streptomyces sp.* JH002 GENOME

Analyses were conducted to confirm that a putative mono-ADP-ribosyltransferase (mART) toxin is encoded in the genome of the pathogenic isolate *Streptomyces sp.* JH002 by following the mART toxin discovery pipeline described by Tremblay et al., [34], with some modifications as follows: (i) we evaluated whether this protein has a similar folding to other mART toxins by analyzing the sequence in Local Meta-Threading Server (LOMETS). In addition, the putative protein 3D structure was predicted by using RaptorX template-based protein structure modeling under default parameters [82, 83]; (ii) the presence of secretion signal peptides or indicators of non-classical secretion and the lack of transmembrane domains in the sequences was determined by SignalP 5.0 and SecretomeP 2.0 with default parameters [84, 85]; (iii) conserved catalytic mART motifs from the sequence were identified through mART toxin sequence alignments conducted by MUSCLE; and (iv) the molecular function of the putative mART was predicted by using COFACTOR server under default parameters [86].

In addition, phylogenetic analysis based on protein sequences of mART and putative mART toxins was conducted. Accession numbers of protein sequences are shown in Table S6 in [27]. The sequence alignment was conducted using MUSCLE in MEGA 7.0 and trimmed manually to the same position. The phylogenetic tree was carried out on Phylotree.fr by Mr. Bayes 3.2.6 with default parameters, except the substitution model implemented was Poisson and the number of generations was set to 100,000 parameters that yielded the highest Bayesian support values.

6.6. AVAILABILITY OF DATA AND MATERIALS

All genome assemblies and annotations are available at Genbank (see Table 6.1). Raw sequencing data has been submitted to NCBI with BioProject PRJNA657491.

BIBLIOGRAPHY

[1] R. Seipke, M. Kaltenpoth, and M. Hutchings. "Streptomyces as symbionts: An emerging and widespread theme?" In: *FEMS Microbiology Reviews* 36.4 (2012), pp. 862–876.

[2] A. Craney, S. Ahmed, and J. Nodwell. "Towards a new science of secondary metabolism". In: *Journal of Antibiotics* 66.7 (2013), pp. 387–400.

[3] R. de Lima Procópio et al. "Antibiotics produced by Streptomyces". In: *Brazilian Journal of Infectious Diseases* 16.5 (2012), pp. 466–471.

[4] J. O'Brien and G. Wright. "An ecological perspective of microbial secondary metabolism". In: *Current Opinion in Biotechnology* 22.4 (2011), pp. 552–558.

[5] M. Choudoir, C. Pepe-Ranney, and D. Buckley. "Diversification of secondary metabolite biosynthetic gene clusters coincides with lineage divergence in Streptomyces". In: *Antibiotics* 7.1 (2018), pp. 1–15.

[6] P. Vaz Jauri et al. "Subinhibitory antibiotic concentrations mediate nutrient use and competition among soil streptomyces". In: *PLoS ONE* 8.12 (2013).

[7] N. Lee et al. "Mini review: Genome mining approaches for the identification of secondary metabolite biosynthetic gene clusters in Streptomyces". In: *Computational and Structural Biotechnology Journal* 18 (2020), pp. 1548–1556.

[8] K. Belknap et al. "Genome mining of biosynthetic and chemotherapeutic gene clusters in Streptomyces bacteria". In: *Scientific Reports* 10.1 (2020).

[9] Y. Li et al. "Virulence mechanisms of plant-pathogenic streptomyces species: An updated review". In: *Microbiology (United Kingdom)* 165.10 (2019), pp. 1025–1040.

[10] R. Loria, J. Kers, and M. Joshi. "Evolution of plant pathogenicity in Streptomyces". In: *Annual Review of Phytopathology* 44 (2006), pp. 469–487.

[11] R. Loria et al. "Plant pathogenicity in the genus Streptomyces". In: *Plant Disease* 81.8 (1997), pp. 836–846.

[12] S. Lerat, A.-M. Simao-Beaunoir, and C. Beaulieu. "Genetic and physiological determinants of Streptomyces scabies pathogenicity". In: *Molecular Plant Pathology* 10.5 (2009), pp. 579–585.

[13] D. Bignell et al. "What does it take to be a plant pathogen: Genomic insights from Streptomyces species". In: *Antonie van Leeuwenhoek, International Journal of General and Molecular Microbiology* 98.2 (2010), pp. 179–194.

[14] J. Fyans et al. "Characterization of the coronatine-like phytotoxins produced by the common scab pathogen streptomyces scabies". In: *Molecular Plant-Microbe Interactions* 28.4 (2015), pp. 443–454.

[15] D. Komeil, A.-M. Simao-Beaunoir, and C. Beaulieu. “Detection of potential suberinase-encoding genes in streptomyces scabiei strains and other actinobacteria”. In: *Canadian Journal of Microbiology* 59.5 (2013), pp. 294–303.

[16] Y. Zhang et al. “Promiscuous pathogenicity islands and phylogeny of pathogenic streptomyces spp.” In: *Molecular Plant-Microbe Interactions* 29.8 (2016), pp. 640–650.

[17] B. Lyons et al. “Scabin, a novel DNA-acting ADP-ribosyltransferase from Streptomyces scabies”. In: *Journal of Biological Chemistry* 291.21 (2016), pp. 11198–11215.

[18] G. Legault et al. “Tryptophan regulates thaxtomin a and indole-3-acetic acid production in Streptomyces scabiei and modifies its interactions with radish seedlings”. In: *Phytopathology* 101.9 (2011), pp. 1045–1051.

[19] M. Natsume et al. “Effects of concanamycins produced by Streptomyces scabies on lesion type of common scab of potato”. In: *Journal of General Plant Pathology* 83.2 (2017), pp. 78–82.

[20] M. Joshi et al. “Streptomyces turgidiscabies secretes a novel virulence protein, Nec1, which facilitates infection”. In: *Molecular Plant-Microbe Interactions* 20.6 (2007), pp. 599–608.

[21] R. Seipke and R. Loria. “Streptomyces scabies 87-22 possesses a functional tomatinase”. In: *Journal of Bacteriology* 190.23 (2008), pp. 7684–7692.

[22] M. Natsume et al. “Phytotoxin produced by the netted scab pathogen, Streptomyces turgidiscabies strain 65, isolated in Sweden”. In: *Journal of General Plant Pathology* 84.2 (2018), pp. 108–117.

[23] M. Natsume et al. “Phytotoxin produced by Streptomyces sp. causing potato russet scab in Japan”. In: *Journal of General Plant Pathology* 71.5 (2005), pp. 364–369.

[24] Z. Cao et al. “Isolation of borrelidin as a phytotoxic compound from a potato pathogenic Streptomyces strain”. In: *Bioscience, Biotechnology and Biochemistry* 76.2 (2012), pp. 353–357.

[25] M. Lapaz et al. “Isolation and structural characterization of a non-diketopiperazine phytotoxin from a potato pathogenic Streptomyces strain”. In: *Natural Product Research* 33.20 (2019), pp. 2951–2957.

[26] L. Henao et al. “Genotypic and phenotypic characterization of Streptomyces species associated with potato crops in the central part of Colombia”. In: *Plant Pathology* 71.3 (2022), pp. 750–761.

[27] L. Henao et al. “Genomes of four Streptomyces strains reveal insights into putative new species and pathogenicity of scab-causing organisms”. In: *BMC genomics* 24.1 (2023), p. 143.

[28] D. Labeda et al. “Phylogenetic relationships in the family Streptomycetaceae using multi-locus sequence analysis”. In: *Antonie van Leeuwenhoek, International Journal of General and Molecular Microbiology* 110.4 (2017), pp. 563–583.

[29] K. Pirc et al. “Nep1-like proteins as a target for plant pathogen control”. In: *PLoS Pathogens* 17.4 (2021).

[30] D. Figaj et al. “The role of proteases in the virulence of plant pathogenic bacteria”. In: *International Journal of Molecular Sciences* 20.3 (2019).

[31] F.-F. Wang and W. Qian. “The roles of histidine kinases in sensing host plant and cell-cell communication signal in a phytopathogenic bacterium”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 374.1767 (2019).

[32] M. Fries et al. “Molecular basis of the activity of the phytopathogen pectin methylesterase”. In: *EMBO Journal* 26.17 (2007), pp. 3879–3887.

[33] V. Rafiei, H. Vélëz, and G. Tzelepis. “The role of glycoside hydrolases in phytopathogenic fungi and oomycetes virulence”. In: *International Journal of Molecular Sciences* 22.17 (2021).

[34] O. Tremblay et al. “Several New Putative Bacterial ADP-Ribosyltransferase Toxins Are Revealed from In Silico DataMining, Including the Novel Toxin Vorin, Encoded by the Fire Blight Pathogen Erwinia amylovora”. In: *Toxins* 12.12 (2020).

[35] P. Gouet, X. Robert, and E. Courcelle. “ESPript/ENDscript: Extracting and rendering sequence and 3D information from atomic structures of proteins”. In: *Nucleic Acids Research* 31.13 (2003), pp. 3320–3323.

[36] N. Lee et al. “Thirty complete Streptomyces genome sequences for mining novel secondary metabolite biosynthetic gene clusters”. In: *Scientific Data* 7.1 (2020).

[37] M.-J. Xu et al. “Deciphering the streamlined genome of Streptomyces xiamenensis 318 as the producer of the anti-fibrotic drug candidate xiamenmycin”. In: *Scientific Reports* 6 (2016).

[38] J. Doroghazi and D. Buckley. “Intraspecies comparison of Streptomyces pratensis genomes reveals high levels of recombination and gene conservation between strains of disparate geographic origin”. In: *BMC Genomics* 15.1 (2014).

[39] X. Rong et al. “Classification of Streptomyces phylogroup pratensis (Doroghazi and Buckley, 2010) based on genetic and phenotypic evidence, and proposal of Streptomyces pratensis sp. nov.” In: *Systematic and Applied Microbiology* 36.6 (2013), pp. 401–407.

[40] W. Ludwig et al. “Road map of the phylum Actinobacteria”. In: *Bergey’s manual® of systematic bacteriology*. Springer, 2012, pp. 1–28.

[41] Z. Liu et al. “Classification of Streptomyces griseus (Krainsky 1914) Waksman and Henrici 1948 and related species and the transfer of ’Microstreptospora cinerea’ to the genus Streptomyces as Streptomyces yanii sp. nov.” In: *International Journal of Systematic and Evolutionary Microbiology* 55.4 (2005), pp. 1605–1610.

[42] E. Goss, N. Potnis, and J. Jones. “Grudgingly sharing their secrets: New insight into the evolution of plant pathogenic bacteria”. In: *New Phytologist* 199.3 (2013), pp. 630–632.

[43] C. Straub, E. Colombi, and H. McCann. “Population genomics of bacterial plant pathogens”. In: *Phytopathology* 111.1 (2021), pp. 23–31.

[44] A. Richter et al. “Biosynthesis of the Stress-Protectant and Chemical Chaperon Ectoine: Biochemistry of the Transaminase EctB”. In: *Frontiers in Microbiology* 10 (2019).

[45] M. Pavan, N. López, and M. Pettinari. “Melanin biosynthesis in bacteria, regulation and production perspectives”. In: *Applied Microbiology and Biotechnology* 104.4 (2020), pp. 1357–1370.

[46] A. Arias et al. “Growth of desferrioxamine-deficient Streptomyces mutants through xenosiderophore piracy of airborne fungal contaminations”. In: *FEMS Microbiology Ecology* 91.7 (2015).

[47] R. Seipke et al. “The plant pathogen Streptomyces scabies 87-22 has a functional pyochelin biosynthetic pathway that is regulated by TetR- and AfsR-family proteins”. In: *Microbiology* 157.9 (2011), pp. 2681–2693.

[48] M. Joshi et al. “The twin arginine protein transport pathway exports multiple virulence proteins in the plant pathogen Streptomyces scabies”. In: *Molecular Microbiology* 77.1 (2010), pp. 252–271.

[49] C. Santos et al. “Molecular mechanisms associated with xylan degradation by xanthomonas plant pathogens”. In: *Journal of Biological Chemistry* 289.46 (2014), pp. 32186–32200.

[50] S. Saka and H.-J. Bae. *Secondary Xylem for Bioconversion*. 2016, pp. 213–231.

[51] Y. Xu et al. “Phytophthora sojae apoplastic effector AEP1 mediates sugar uptake by mutarotation of extracellular aldose and is recognized as a MAMP”. In: *Plant Physiology* 187.1 (2021), pp. 321–335.

[52] V.-A. Ochoa-Jiménez et al. “Functional analysis of tomato rhamnogalacturonan lyase gene Solyc11g011300 during fruit development and ripening”. In: *Journal of Plant Physiology* 231 (2018), pp. 31–40.

[53] B. Wachananawat et al. “Diversity of Pectin Rhamnogalacturonan I Rhamnosyltransferases in Glycosyltransferase Family 106”. In: *Frontiers in Plant Science* 11 (2020).

[54] B. Li et al. “Molecular basis and regulation of pathogenicity and patulin biosynthesis in Penicillium expansum”. In: *Comprehensive Reviews in Food Science and Food Safety* 19.6 (2020), pp. 3416–3438.

[55] B. Lyons et al. “Characterization of the catalytic signature of Scabin toxin, a DNA-targeting ADP-ribosyltransferase”. In: *Biochemical Journal* 475.1 (2018), pp. 225–245.

[56] M. Watanabe et al. “Mono(ADP-ribosyl)ation of DNA by apoptosis-inducing protein, pierisin.” In: *Nucleic acids research. Supplement (2001)* 2 (2002), pp. 243–244.

[57] T. Takamura-Enya et al. “Mono(ADP-ribosyl)ation of 2’-deoxyguanosine residue in DNA by an apoptosis-inducing protein pierisin-1 from cabbage butterfly”. In: *Proceedings of the National Academy of Sciences of the United States of America* 98.22 (2001), pp. 12414–12419.

[58] M. Lugo et al. “Dynamics of Scabin toxin. A proposal for the binding mode of the DNA substrate”. In: *PLoS ONE* 13.3 (2018).

[59] T. Yoshida and H. Tsuge. “Substrate N2 atom recognition mechanism in pierisin family DNA-targeting, guanine-specific ADP-ribosyltransferase ScARP”. In: *Journal of Biological Chemistry* 293.36 (2018), pp. 13768–13774.

[60] G. Pishchany et al. “Amycomicin is a potent and specific antibiotic discovered with a targeted interaction screen”. In: *Proceedings of the National Academy of Sciences of the United States of America* 115.40 (2018), pp. 10124–10129.

[61] B. Salehi et al. “The therapeutic potential of naringenin: A review of clinical trials”. In: *Pharmaceuticals* 12.1 (2019).

[62] K. Zdyb et al. *Siderophores: Microbial tools for iron uptake and resistance to other metals*. 2017, pp. 247–266.

[63] N. Menéndez et al. “Biosynthesis of the Antitumor Chromomycin A3 in Streptomyces griseus: Analysis of the Gene Cluster and Rational Design of Novel Chromomycin Analogs”. In: *Chemistry and Biology* 11.1 (2004), pp. 21–32.

[64] D.-C. Oh et al. “Sceliphrolactam, a polyene macrocyclic lactam from a wasp-associated Streptomyces sp.” In: *Organic Letters* 13.4 (2011), pp. 752–755.

[65] E. Shirling and D. Gottlieb. “Methods for characterization of Streptomyces species”. In: *Int. J. Syst. Bacteriol.* 16 (1966), pp. 313–340.

[66] M. Kolmogorov et al. “Assembly of long, error-prone reads using repeat graphs”. In: *Nature biotechnology* 37.5 (2019), pp. 540–546.

[67] F. A. Simão et al. “BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs”. In: *Bioinformatics* 31.19 (2015), pp. 3210–3212.

[68] R. Aziz et al. “The RAST Server: Rapid annotations using subsystems technology”. In: *BMC Genomics* 9 (2008).

[69] J. Chun et al. “Proposed minimal standards for the use of genome data for the taxonomy of prokaryotes”. In: *International Journal of Systematic and Evolutionary Microbiology* 68.1 (2018), pp. 461–466.

[70] R. Edgar. “MUSCLE: Multiple sequence alignment with high accuracy and high throughput”. In: *Nucleic Acids Research* 32.5 (2004), pp. 1792–1797.

[71] S. Kumar, G. Stecher, and K. Tamura. “MEGA7: Molecular Evolutionary Genetics Analysis Version 7.0 for Bigger Datasets”. In: *Molecular Biology and Evolution* 33.7 (2016), pp. 1870–1874.

[72] S.-H. Yoon et al. “Introducing EzBioCloud: A taxonomically united database of 16S rRNA gene sequences and whole-genome assemblies”. In: *International Journal of Systematic and Evolutionary Microbiology* 67.5 (2017), pp. 1613–1617.

[73] J. Castresana. “Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis”. In: *Molecular Biology and Evolution* 17.4 (2000), pp. 540–552.

[74] A. Stamatakis. “RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies”. In: *Bioinformatics* 30.9 (2014), pp. 1312–1313.

[75] J. R. Miller, S. Koren, and G. Sutton. “Assembly algorithms for next-generation sequencing data”. In: *Genomics* 95.6 (2010), pp. 315–327.

[76] K. Blin et al. “AntiSMASH 5.0: Updates to the secondary metabolite genome mining pipeline”. In: *Nucleic Acids Research* 47.W1 (2019), W81–W87.

[77] D. Emms and S. Kelly. “OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy”. In: *Genome Biology* 16.1 (2015).

[78] M. Johnson et al. “NCBI BLAST: a better web interface.” In: *Nucleic acids research* 36.Web Server issue (2008), W5–9.

[79] I. Letunic, S. Khedkar, and P. Bork. “SMART: Recent updates, new developments and status in 2020”. In: *Nucleic Acids Research* 49.D1 (2021), pp. D458–D460.

[80] K. Dilks et al. “Prokaryotic utilization of the twin-arginine translocation pathway: a genomic survey”. In: *Journal of bacteriology* 185.4 (2003), pp. 1478–1483.

[81] J. D. Bendtsen et al. “Prediction of twin-arginine signal peptides”. In: *BMC bioinformatics* 6.1 (2005), pp. 1–9.

[82] M. Källberg et al. “Template-based protein structure modeling using the RaptorX web server”. In: *Nature protocols* 7.8 (2012), pp. 1511–1522.

[83] S. Wu and Y. Zhang. “LOMETS: a local meta-threading-server for protein structure prediction”. In: *Nucleic acids research* 35.10 (2007), pp. 3375–3382.

[84] J. J. Almagro Armenteros et al. “SignalP 5.0 improves signal peptide predictions using deep neural networks”. In: *Nature biotechnology* 37.4 (2019), pp. 420–423.

[85] J. D. Bendtsen et al. “Non-classical protein secretion in bacteria”. In: *BMC microbiology* 5.1 (2005), pp. 1–13.

[86] C. Zhang, P. L. Freddolino, and Y. Zhang. “COFACTOR: improved protein function prediction by combining structure, sequence and protein–protein interaction information”. In: *Nucleic acids research* 45.W1 (2017), W291–W299.

7

DISCUSSION

This thesis investigated critical issues in genome and haplotype assembly based on Third Generation Sequencing (TGS) technologies. First, it aimed to identify optimal approaches for achieving comprehensive genome and haplotype assembly. This was done through literature review of haplotype assembly and benchmarking of available third generation sequencing-based genome assembly tools. Then, it tackled one of the most difficult challenges of genome assembly; handling of repetitive sequences. It demonstrated how removing repeat-induced overlaps could significantly enhance the assembly process. Finally, it introduced an innovative approach for haplotype assembly that emphasized the interpretability of the results.

Chapter 2 provided provided a fundamental understanding of the problems, data, and methods associated with haplotype assembly. It detailed the difficulties that prevented accurate haplotype assembly, such as variation deserts, and set the stage for the following chapters to investigate specific methods to overcome these issues. This chapter acted as a stepping stone, guiding the research towards more effective strategies for haplotype assembly.

In Chapter 3, the focus shifted towards de novo genome assembly and showed the potential of long-read sequencing technologies to address the shortcomings of short-read sequencing. In this chapter, we compared different third generation sequencing technologies and de novo assembly tools using 76 datasets. Our results showed that although there was no assembler that always performed the best in all quality metrics, Flye and Hifiasm were the most consistent assemblers for error prone and HiFi reads, respectively. Furthermore, we investigated the effect of using even longer reads in de novo assembly and showed how the longer reads could benefit the more complicated genomes with higher repetitive content while they didn't add much for simpler organisms. Together, Chapters 2 and 3 created a better understanding of the challenges in genome reconstruction.

Chapter 4 extended the discussion on genome assembly, specifically focusing on the issues caused by interspersed repeats. It examined how these repeats affected the standard long read de novo assembly pipeline. We showed that by removing repeat-induced overlaps, significant improvements were achieved in the assembly process, enhancing the correctness and contiguity of the reconstructed genomes. Then we introduced multiple approaches, such as training a machine learning model, to detect and remove repeat-induced overlaps. This chapter brought valuable insights into the refinement of de novo assembly pipelines,

which could lead to improved genome sequences.

Chapter 5 synthesized the knowledge gained from previous chapters and revisited the complexities involved in reconstructing individual haplotypes discussed in Chapter 2. We introduced HAT, a haplotype assembly tool that used NGS and TGS data to reconstruct haplotypes on a chromosome scale. Unlike other haplotype assembly tools, HAT uniquely addressed variation deserts by identifying multiplicity blocks at the start of phasing. These multiplicity blocks represented regions that were not identical across the haplotypes in the genome to be phased by HAT. Subsequently, we demonstrated that HAT outperformed existing methods for haplotype reconstruction based on TGS data, while significantly improving interpretability by providing multiplicity blocks.

At last, Chapter 6 narrowed down to the taxonomic classification of Streptomyces species and their pathogenicity. Based on the knowledge acquired from the previous chapters, we assembled two new Streptomyces species. Next, by analyzing the genomic aspects of these pathogens and identifying their virulence factors, this research deepened our knowledge of microbial genomics. Finally, this chapter offered potential therapeutic applications of Streptomyces species.

7.1. USING MULTIPLE SEQUENCING TECHNOLOGIES FOR HAPLOTYPE ASSEMBLY

The use of multiple sequencing technologies significantly boosts our ability to deal with complex genomic tasks such as haplotype assembly. Illumina's NGS and other short-read technologies provide accurate detection of single nucleotide variations and small insertions or deletions, forming a solid foundation for identifying haplotypes. However, their ability to explore large structural changes and complex genomic regions is limited. Here, long-read technologies such as Oxford Nanopore or PacBio fill in the gaps, helping to unravel these larger elements that are key to building complex haplotypes. PacBio's HiFi sequencing blends the strengths of both, offering long reads with high accuracy. Furthermore, 'linked-reads' from 10x Genomics extend our vision, providing a wider genomic context that helps in separating out haplotypes.

Chapter 3 presented a performance comparison of various long-read assemblers using both simulated and real data from Third Generation Sequencing technologies. However, hybrid assemblies, which combine different types of data, are becoming increasingly popular. A more holistic approach might have been to compare combinations of TGS assemblers with other technologies that enhanced the assembly's quality and provided a structural overview, like Illumina HiSeq for fine-tuning the assembly and HiC for linking larger pieces. This integrated approach could be valuable for complex genomes. For simpler genomes like the streptomyces DNA we assembled in Chapter 6 using only PacBio TGS data, the gains are not significant.

Chapter 4 further explored how repetitive regions impacted TGS genome assembly. We found that when a genome has many repetitive regions, many of the pairwise alignments between the TGS reads are due to these repeats. In this chapter, we only used computational methods to identify these repeat-induced overlaps, but using long-range sequencing technologies such as HiC could be another way to deal with repetitive regions, as they can link together sections up to several million bases apart.

In Chapter 5, we demonstrated the advantages of using multiple sequencing technologies with the Haplotype Assembly Tool (HAT). HAT starts by using short-reads to produce accurate starting points for the assembly, then uses long-read data to extend these and handle complex regions. But there is still room for improvement. We only used TGS to connect the phased blocks in HAT, but adding other types of long-range data might give even better results. Likewise, while we used Illumina short reads to call variants and create initial seeds, using HiFi reads could have allowed us to create longer and more accurate seeds, making the task easier.

7.2. EXPLORING MICROBIAL COMMUNITIES THROUGH HAPLOTYPE ASSEMBLY

Investigating microbial communities often involves challenging complexities. These communities are made up of multiple species and strains that coexist harmoniously. The substantial similarity among genomes of different strains, along with the complexities of individual sequencing due to cost and the presence of multiple species, can make it challenging to accurately separate and sequence individual genomes. In this context, haplotype assembly can be a beneficial tool.

Considering different strains as unique haplotypes allows us to tackle this challenge using the techniques of polyploid haplotype assembly. The objective shifts from attempting to separate entirely distinct species or strains to assembling distinct haplotypes within a mixed sample. This change allows the reconstruction of individual strain genomes from mixed samples, a task that would be difficult without adopting a haplotype perspective. This perspective would allow us to adapt and use all haplotype assembly techniques for strain level metagenomics assembly. For instance, the HAT multiplicity block finder module can be used in a metagenomics setting to identify the number of different strains of specific species in metagenomic communities.

Once the genomes of individual strains are identified, we can further explore the roles and relationships within the microbial community. These genomes can reveal the functional abilities of individual strains, highlighting their role in the microbial ecosystem. They also provide insights into the genomic diversity within a community, which is essential for evolutionary studies.

Consequently, this view will enhance our ability to decode the complex genetic interactions within microbial communities.

7.3. HAPLOTYPE ASSEMBLY: DIRECTIONS FOR FUTURE RESEARCH

Two major changes are currently transforming the field of genomics and setting new directions for future research.

The first big shift involves developing computer methods specially tailored for long-read sequencing data. As our grasp of genomics expands and technology improves, we're seeing a massive rise in new tools, each based on a unique concept, which results in a wide range of methods. In Chapter 2, we briefly presented this. This diversity, however, complicates tool comparison and performance measurement, making comprehensive, gold-standard simu-

lated datasets necessary. Simulated data provide us with a known ground truth, facilitating more accurate evaluation of the outputs, a common challenge in bioinformatics. As an example, in the area of haplotype assembly, this dataset should include genomes with different ploidies, haplotype variations, areas lacking variation, and repetitive content. This resource would enable researchers to compare tools under uniform conditions, assisting in choosing the most appropriate tool. As these tools become more advanced, our testing methods need to keep pace, further emphasizing the need for a standardized, rigorous benchmarking system. An intriguing idea is to present reads from simulated references with varied attributes in a Kaggle-style competition. Contestants would receive reads from select references and then develop methods to assemble them. Later, these methods would be assessed using reads from other simulated references to select the winner. As previously mentioned, because the references are simulated, the ground truth is known, ensuring a fair evaluation of the outputs.

The second shift comes with the development of highly accurate long-read sequencing technologies like HiFi [1, 2]. This change is reshaping many areas, including haplotype assembly. Tools that used to depend on short reads, e.g., HAT, can now switch to accurate long-read technologies. These long reads cover larger parts of the genome, providing extensive and accurate starting points for assembly algorithms. This change can simplify the haplotype assembly process and lead to a more accurate reconstruction of the haplotypes. This change comes with its own set of difficulties. Replacing short reads with accurate long reads presents challenges due to the increased length. Methods tailored for short reads don't necessarily handle the accurate long reads seamlessly. While switching from long reads to accurate long reads is somewhat more straightforward, it's not without complications; accurate long reads lack the ultra-length, potentially resulting in missing information and new challenges. Thus, adopting accurate long reads compels us to reassess and adjust our current methodologies. For instance, long accurate reads can enhance HAT in developing longer and more exact haplotype blocks, as they cover more SNPs and can create longer initial seeds for phasing. But with short reads, we could just take all consecutive SNPs covered by short reads as the starting seeds. This approach, however, doesn't work when using long accurate reads since it would create too many initial seeds which would increase the runtime of the tool significantly. Therefore, a more thoughtful strategy is needed to create initial seeds. In conclusion, it is not possible to effortlessly swap short reads with accurate long reads; we also need to modify the base algorithm to work with these long accurate reads. On top of that, HiFi reads cannot entirely replace other long-read technologies due to their lower throughput and the absence of extra-long reads, which are vital for resolving complex genomic regions. The combination of HiFi sequencing and HiC technologies has demonstrated significant performance in haplotype assembly, as shown by the pstools algorithm [3]. This novel approach uses the precision of HiFi sequencing to construct sequence graphs that retain haplotype information, while HiC data contributes to global phasing, ordering, and genome orientation. However, pstools faced challenges in resolving certain complex centromic regions. The authors suggested that incorporating ultra-long reads could address this issue, showing that HiFi reads still cannot fully replace other long-read technologies. Furthermore, while the synergy between HiFi and HiC is promising, challenges persist, particularly in resolving highly complex regions with a high multiplicity and long repetitive regions. This highlights that achieving the complete and

accurate assembly of highly complex polyploid genomes remains a formidable challenge that requires further technological and computational progress.

These two developments - the advancement of computer methods for long-read data and the emergence of long accurate read technologies - are leading us into a new era in haplotype assembly and genomics. By acknowledging these changes, we can actively contribute to the refinement of more precise, adaptable, and widely applicable haplotype assembly methods. The utilization of HiFi, HiC, and ultra-long read technologies holds the promise of delivering fully haplotype-resolved genomes, particularly in simpler cases like diploid genomes such as humans, potentially revolutionizing healthcare by enabling the assignment of compound genetic disorders to specific haplotypes when they underlie a disease. It is important to note that, as discussed in pstools [3], ultra-long reads from TGS technologies will still be necessary to address complex genomic regions, suggesting that the methods introduced in this dissertation could find future applications in achieving fully resolved haplotypes. Nonetheless, achieving fully resolved haplotypes will require new tools for downstream analysis. For instance, in polyploid genomes, traditional read mapping methods would require the mapping of reads to each haplotype separately, a process that is not efficient. As a result, innovative approaches will be essential for downstream analysis of haplotype-resolved genomes, such as the adoption of pangenome read mapping techniques. In summary, these transformative developments, including both methodology and technology, mark a new chapter in haplotype assembly and genomics.

BIBLIOGRAPHY

[1] M. R. Vollger et al. “Improved assembly and variant detection of a haploid human genome using single-molecule, high-fidelity long reads”. In: *Annals of human genetics* 84.2 (2020), pp. 125–140.

[2] S. Hotaling et al. “Highly accurate long reads are crucial for realizing the potential of biodiversity genomics”. In: *BMC genomics* 24.1 (2023), pp. 1–9.

[3] S. Garg. “Towards routine chromosome-scale haplotype-resolved reconstruction in cancer genomics”. In: *Nature Communications* 14.1 (2023), p. 1358.

A

Supplementary materials - A review of computational methods to reconstruct polyploid haplotypes based on DNA sequencing data

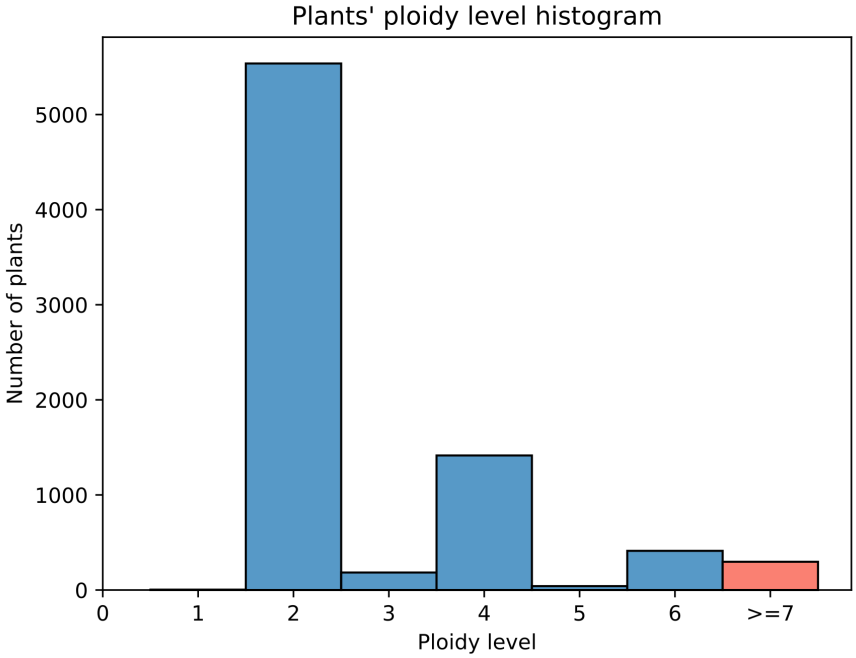


Figure A.1: Histogram of plants' ploidy level based on Plant DNA C-values database [113]. The red bar indicates the number of plants with ploidy equal or greater than seven. The red bar amounts to 287 out of 7890 plant species, i.e. 3.6%.

B

SUPPLEMENTARY MATERIALS - EVALUATING LONG READ DE NOVO ASSEMBLY TOOLS FOR EUKARYOTIC GENOMES: INSIGHTS AND CONSIDERATIONS

Due to the extensive nature of supplementary materials associated with this chapter, they are not included in this thesis. Interested readers are encouraged to refer to the published paper for supplementary materials (see `https://doi.org/10.1093/gigascience/giad100`).

C

SUPPLEMENTARY MATERIALS - THE EFFECT OF REMOVING REPEAT-INDUCED OVERLAPS IN *de novo* ASSEMBLY SEQUENCING DATA

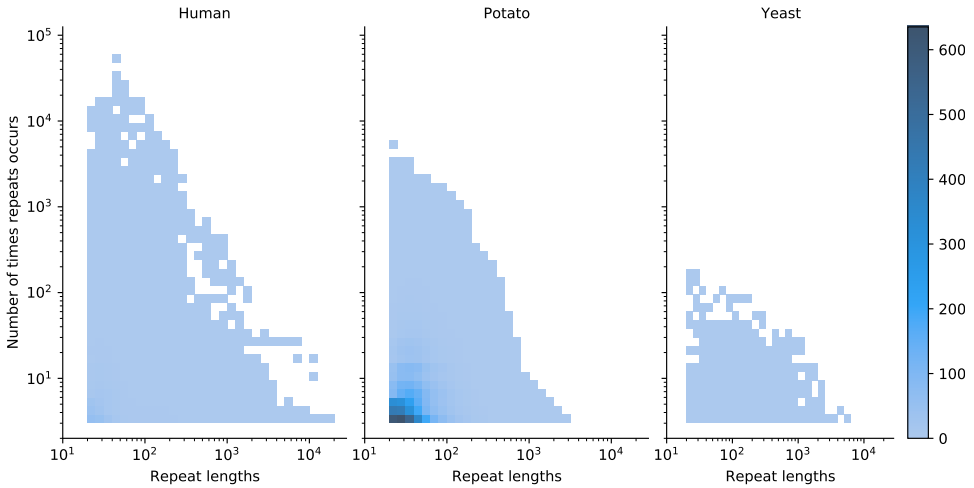


Figure C.1: This plot shows the two dimensional histogram of repeat-lengths and number of times repeats occurs in human, potato, and yeast genomes. This plot shows that the smaller repeats occur more in the genome.

Table C.1: The metrics that we used in chapter 4 for assembly evaluation.

Metrics	Definition
Coverage	Coverage is a measure of the fraction of the reference genome that is present in an assembly. It is determined as the ratio between the summed length of the coverage islands and the summed length of the reference sequences. Coverage values are between 0 and 1, and a higher coverage is preferred.
Validity	Validity is the ratio between the summed length of the alignments and the assembled scaffolds, measuring how much of the assembly is aligned to the reference. The validity values closer to 1 are considered better.
Multiplicity	Multiplicity is the ratio of the summed length of the alignments and the summed length of the coverage islands. This metric describes whether the assembler collapsed or replicated repeats within the genome. The multiplicity values closer to 1 are considered better.
Number of contigs	The total number of contigs in the assembly. The number of contigs is a metric that describes the contiguity of the assembly. The ideal value for the number of contigs is the number of chromosomes of the organism.
Longest contig	The size of the longest contig in the assembly. This metric also describes the contiguity of the assembly. The ideal value for the longest contig is the size of the largest chromosome of the organism.

D

SUPPLEMENTARY MATERIALS - HAT: HAPLOTYPE ASSEMBLY TOOL USING SHORT AND ERROR-PRONE LONG READS

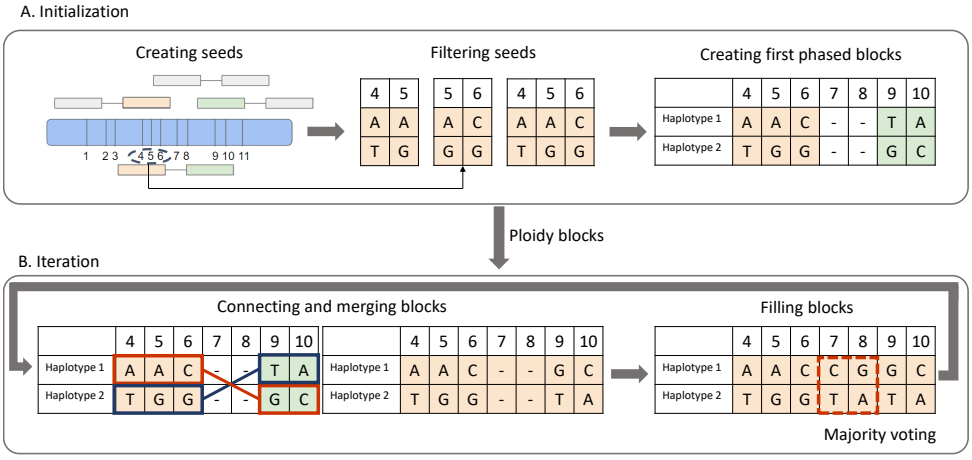


Figure D.1: A. Based on the alignment of the reads that are covering SNPs 4,5 and 6, HAT creates three seeds. In this scenario, because the support of the combinations of alleles of these three seeds are equal, HAT keep the longer one in the filtering seeds step. After that, HAT use the remaining seeds and the combinations of alleles to create first phased blocks. B. Based on the read assignment, the reads that belong to haplotype 1 of block 1, also belong to haplotype 2 of block 2. This means, these two haplotypes are the same, and in the connecting and merging blocks, a bigger block is created, and the mentioned haplotypes are linked together. After that, based on the assignment of reads to each haplotype, and a majority voting between those reads, HAT finds the allele of the unphased SNPs.

Figure D.2: Multiplicity blocks of Triploid low heterozygosity dataset.

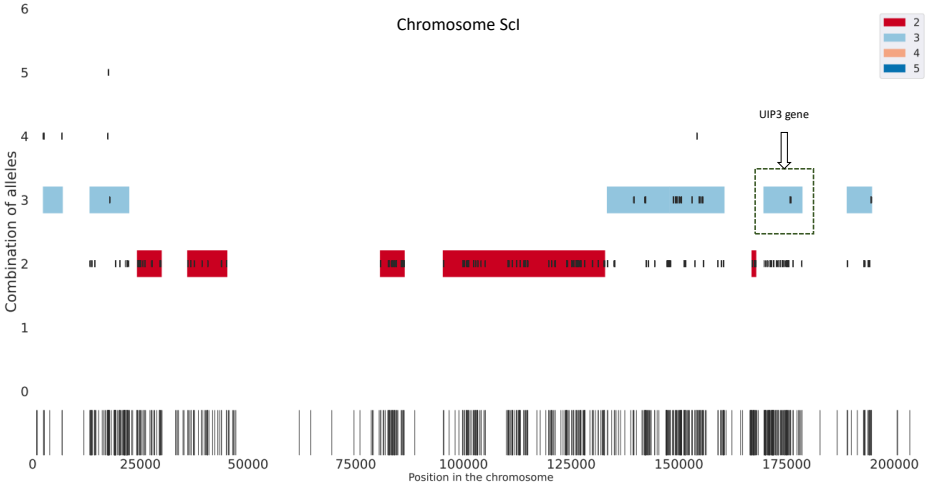


Figure D.3: Multiplicity blocks of triploid ChrSc2 of CBS1483.

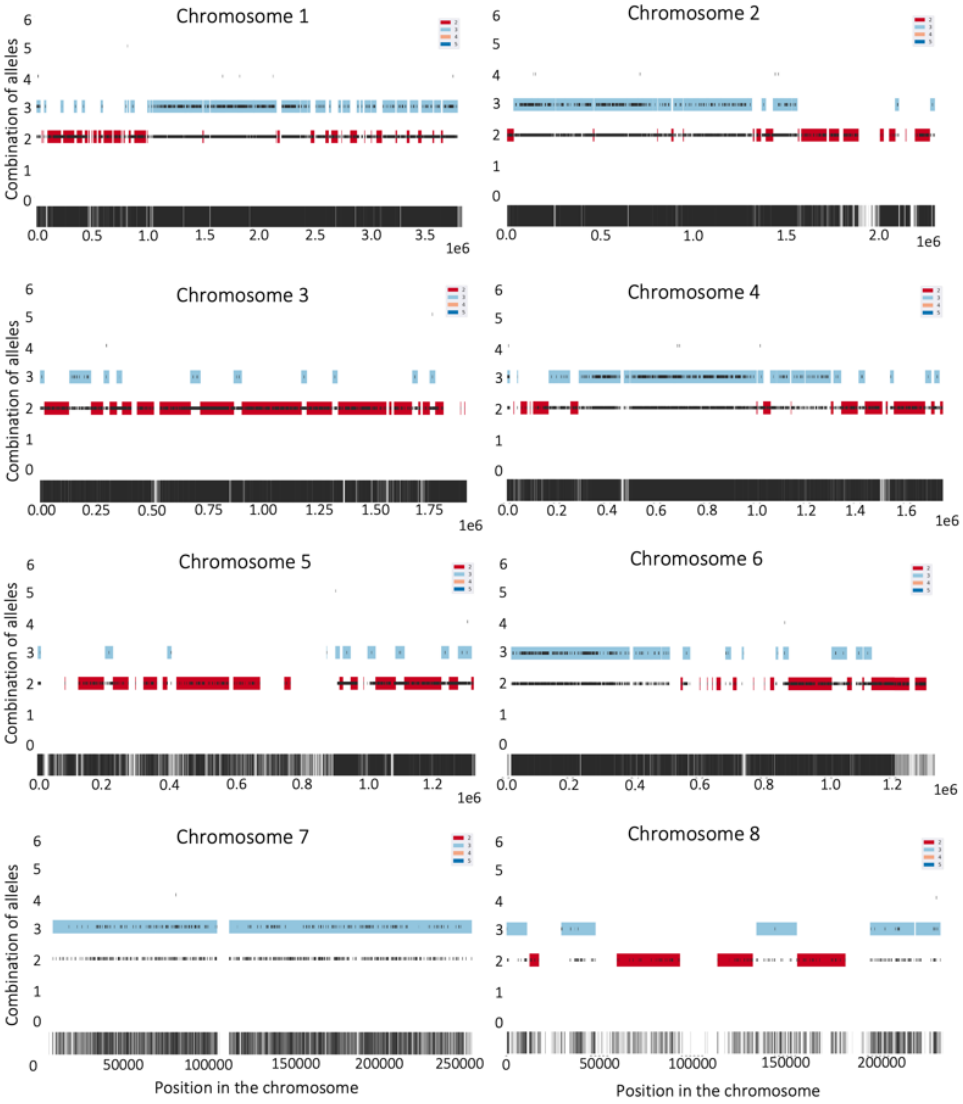


Figure D.4: Multiplicity blocks of all chromosomes of triploid GB54.

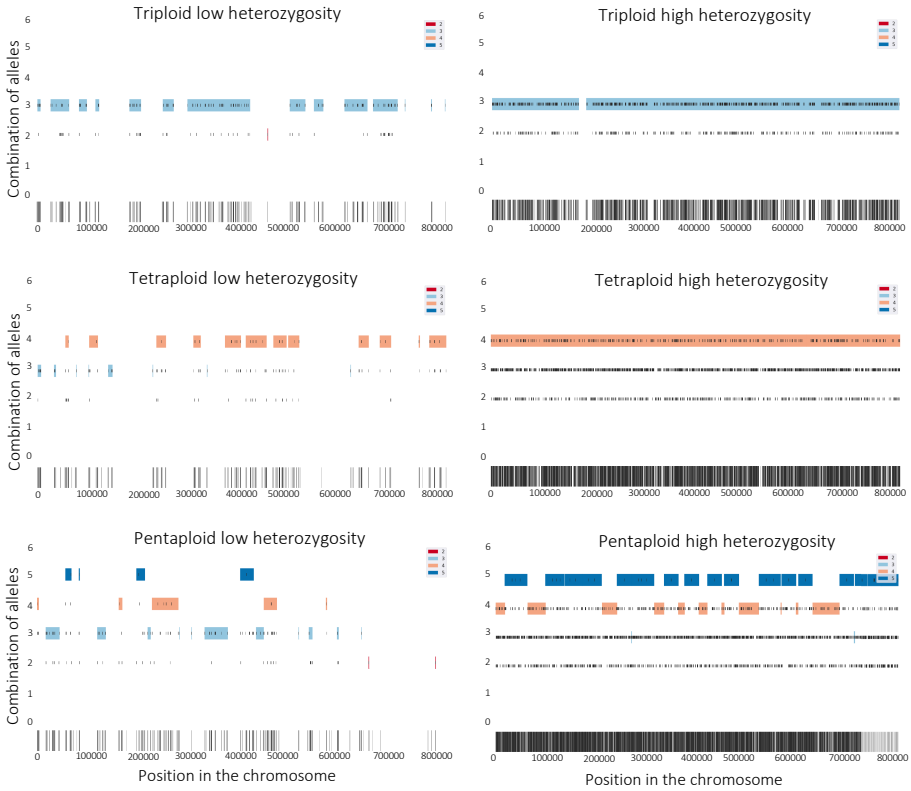


Figure D.5: Multiplicity blocks of all simulated datasets.

Table D.1: Haplogenerator parameters for simulating datasets.

Dataset	SNP		Insertion		Deletion	
	Mean	STD	Mean	STD	Mean	STD
Triploid low heterozygosity	3	2.2	7	2.8	9	3
Triploid high heterozygosity	3	2	7	2	9	3
Tetraploid low heterozygosity	3	2.4	7	2	9	12
Tetraploid high heterozygosity	3	1.6	6	2	9	12
Pentaploid low heterozygosity	3	2.4	7	2	9	3
Pentaploid high heterozygosity	3	2	7	2	9	12

Table D.2: Parameters of the tools we use in this study.

Tool name	Parameter name	Parameter value
ART	Read length	125
	Mean insertion size	400
	Standard deviation of insertion size	20
	coverage	20
Badread	quantity	20
Minimap2	Secondary	No
BWA mem	Default	-
Vcffilter	-f	TYPE = SNP
FreeBayes	Ploidy	Ploidy of the dataset
	Mean alternate count	5
Miniasm	Default	-
Pilon	Default	-
Whatshap haplotag	–ploidy	Ploidy of the chromosome

Table D.3: The haplotype reconstruction improvement after each step: we ran HAT on Chromosome ScII of CBS1483 and calculated the number of blocks and number of variants after each iteration of the iterative part of HAT. For all ploidy blocks HAT converges in at most 4 iterations.

Ploidy block	Initialization		Iteration 1		Iteration 2		Iteration 3		Iteration 4	
	# of blocks	# Phased variants	# blocks	# Phased variants	# blocks	# Phased variants	# blocks	# Phased variants	# blocks	# Phased variants
4473-6122	3	7	3	7	3	7	Converge			
153738-163604	8	16	6	18	1	29	1	29	Converge	
171517-197986	19	38	14	50	3	83	3	83		
213369-231934	12	24	9	27	1	41	1	41		
246873-265346	16	32	8	37	1	62	1	63	1	63
271864-274237	2	4	2	5	1	7	1	7	Converge	
284309-287742	4	8	3	9	1	12	1	12		
294099-298274	8	16	4	23	1	33	1	33		
308600-325784	26	52	13	71	1	93	1	93		
334284-334379	1	2	1	2	1	2	Converge			
344243-344336	1	2	1	2	1	2				
450369-450450	1	2	1	2	1	2				
766447-766450	1	2	1	2	1	2				
787679-795363	9	18	5	22	3	31	3	31	Converge	
800096-804047	1	2	1	2	1	8	1	8		
805511-810793	3	7	3	8	2	13	2	13		

Table D.4: Real and simulated long reads error rate. The error rates are calculated based on the alignment of reads to the dataset reference genome and samtools stats.

Dataset	Error rate percentage
CBS1483 real dataset	15.15%
GB54 real dataset	14.71%
Triploid low heterozygosity	12.82%
Triploid high heterozygosity	12.84%
Tetraploid low heterozygosity	12.61%
Tetraploid high heterozygosity	13.07%
Pentaploid low heterozygosity	12.73%
Pentaploid high heterozygosity	12.94%

E

SUPPLEMENTARY MATERIALS - GENOMES OF FOUR *Streptomyces* STRAINS REVEAL INSIGHTS INTO PUTATIVE NEW SPECIES AND PATHOGENICITY OF SCAB-CAUSING ORGANISMS

Due to the extensive nature of supplementary materials associated with this chapter, they are not included in this thesis. Interested readers are encouraged to refer to the published paper for supplementary materials (see `https://doi.org/10.1186/s12864-023-09190-y`).

ACKNOWLEDGEMENTS

Completing a PhD is undeniably a tough journey, and I'd like to express my sincere appreciation to everyone who played a role in helping me navigate it. Your support, whether it was through guidance, encouragement, or simply being there when I needed it most, made all the difference in reaching this milestone.

Yosra, the love of my life, your constant support was crucial during my PhD. Without it, I wouldn't have been able to finish this journey. **Maman** and **Baba**, your hard work allowed us to grow, for which I am deeply grateful. **Aida**, thank you for being the best sister in the world. **Somayeh**, thanks for all the last-minute help with the thesis cover. **Loki**, though you entered my life recently, I cannot imagine life without you now.

A special thanks to my supervisors: **Thomas**, for wrestling with me over the past 5 years, being patient when I missed my deadlines, and listening to my naggings. **Marcel**, thank you for creating such a friendly group at TUDelft with many bright researchers and for the excellent and critical feedback you gave me at my yearly meetings. I'm glad that we are still working together through my postdoc.

Appreciation extends to other PIs in the DBL group. **Ahmed**, I always enjoyed chatting and partying with you, and thanks for teaching us those beautiful Egyptian dance moves. **Erik**, talking about science with you was always enjoyable during my PhD. I have always admired your ambition for science, and now I enjoy working with you in my postdoc. **Jasmijn**, we had similar topics for our PhDs, and I always looked up to you; thanks for being a great role model. **Jana**, talking to you is always amazing and easy; I'm glad you joined the group. **Joana**, I remember going to lunch with you in the first week of my PhD; thanks for always being so friendly. **Marco**, drinking with you at borrels and then going out to pubs to continue the fun is one of the highlights of my PhD. **David**, chatting with you next to the coffee machine was always fun. **Jan**, thanks for always keeping your smile and positivity.

My gratitude also goes to the C++ group: **Stephanie**, thanks for always providing me with first-hand gossip and keeping my office clean. **Mostafa**, talking with you is always enjoyable, whether it be about football, politics, science, or culture. I'm extremely happy to have you next to me for my defense as a paranymph. **Amelia**, your visit to Delft was one of the best times of my PhD. Talking to you, gaming with you, and having you as a paranymph in my defense is heartwarming. **Mo**, thanks for not letting me down when I asked you to help organize the retreat. That was the beginning of our closer friendship. You and **Zaytoona** are very dear to me. **Yasin**, I haven't seen you in years, but I still have your picture on my monitor. Thanks for always being down for all activities. **Kirti**, you have such a talent in gaming. Thanks for being a formidable opponent in Overcooked.

Then, I want to thank the OG DBL team. **Alex**, thanks for teaching me the 2008 Netherlands' word of the year early in my PhD; it helped me integrate well in the Netherlands. **Soufiane**, hearing "Dastshooyi kojast?" from you always made me smile. Talking to you about French and Iranian culture was always enjoyable. **Stavros**, my trip to Greece for

your wedding is still my best memory from the PhD. Talking to you is always fun and entertaining. I'm still waiting for the trip to the specific island you promised me. **Niki**, thanks for being so cute :)). **Valentina**, thanks for throwing such nice parties with amazing Greek food. **Tamim**, your supportive words with a deep voice always encouraged me to do what I was afraid of doing. **Tom Mokveld**, thanks for organizing amazing board game nights and welcoming me to the group in the first month of my PhD. **Arlin**, I don't know how you manage to do all these things and be successful in all of them, PhD, playing in a band, shooting, sports, and now that you are a mother, I'm pretty sure you will be a wonderful one. Also, thanks for taking me to lunch on the first day of my PhD. **Christine**, you were always very supportive; talking to you has always been fun. **Christian**, I still remember the nights we stayed at the social room up to 9:00 pm, drinking and talking; it was really enjoyable. **Aysun**, we had so much fun together playing board games, having dinner plans, and talking.

Next, I want to thank the best office mates someone could ask for. **Ekin**, with amazing stories. **Laura**, you defended your PhD in the first two weeks of my PhD and threw a fantastic party afterward, which allowed me to connect with other PhDs. **YinCung**, thanks for always being there in the office; with you, I always knew I wouldn't be alone if I went to the office. **Stephanie Tan**, our short chats were always excellent and funny. **Bernd**, thanks for inviting me to your house in the first months of my PhD; it was such a memorable night.

Thanks to the DBL lab members for your friendship and collaboration. **Chengyao**, thanks for introducing me to yùh hēung ké jí, which is one of my favorite foods. I had such a lovely time at your wedding. **Paul**, I don't know why I always have these funny random interactions with you, but believe me, I enjoy them. **Swier**, take care of your shoulder; you need that to complete your PhD. **Roy**, we should go together to restaurants more often. **Colm**, thanks for introducing me to Irish names, history, and culture. **Gabriel**, thanks for always having a smile. **Gerard**, thanks for such a nice retreat you organized. **Jasper**, discussing video/board games with you is very enjoyable. We should definitely finish BG3 campaign together. **Sara**, **Timo**, **Ivan**, **Bram**, and **Niek**, you all started your PhD very recently; I hope you enjoy your time at DBL as much as I did. **Sander**, we talked about you wanting to bike the Silk Road; I wish you fulfill your dream. **Docu**, we had a nice drunken night after Soufiane's party; it was such a memorable night. **Daniyal**, I still believe no one is pronouncing your name correctly. **Akash** and **Alvaro**, you had a short time in DBL, but we wrote a paper together in that time; thanks for helping with the ECCB submissions. **Lieke**, we were searching for a house at the same time; it was so nice that I could share my disappointment with you. **Lucas**, it was amazing that your master's thesis was a nice document for me to refer to at the start of my PhD.

Many thanks to my LUMC colleagues for the fun times we've had, and I'm sure there are even better days to come. **Onur**, talking to you about games, cultures, and food is always fun. I'm sure we will have successful collaborations in the near future. **Jeppe**, you made a tough decision recently; I hope everything works out the best for you. **Marieke**, collaborating with you is one of the fantastic parts of my PostDoc; thanks for all your bright feedback. **Redmar**, you just started your PhD after years of being a bioinformatician at LUMC; good luck with that. **Dani**, I'm jealous of you for eating original cacio e pepe every holiday.

Appreciation also goes to the pattern recognition and computer vision groups. First,

Osman, I always look at you like an older brother. Having coffee chats, tea chats, juice chats, cola chats, sports chats, and lunch chats with you was always very fun and enjoyable. I'm sad that our schedules no longer align, and we cannot spend time together as much as before. **Taygun**, we used to visit very lovely Turkish restaurants together; I miss that. **Ombretta**, talking to you was always fun, and I was very surprised that I saw you as the paranymph for Kiarash. **Attila**, playing Pandemic at your place was extremely fun. **Robert-Jan**, thanks for taking the initiative for everything. **Nargis**, it was such a fun night when we watched Doctor Stone at your place; I followed up on the anime and watched all the episodes later. **Seyran**, having another Iranian in the group at the start of my PhD was heartwarming. **Hesam**, I wish you a very successful PhD. **Tom Viering**, you definitely have the loudest laughs I have ever heard; it was very fun seeing you at Sunakchi's party. **Arman**, I really want to take you to the AnimeCon next year. **Rickard**, thanks for being a great DITO representative. **Ramin Ghorbani**, I'm happy that someone with my name is going to stay in the group and keep my name alive. **Mahdi**, chatting with you is always fun, and thanks for the wall painting tips. **Hayley**, thanks for hiring such interesting PhDs and PostDocs.

Then, I want to thank the other PhDs who started at the same time as me. First, **Chirag**, my friend. It always amazed me how easy it is for you to connect with other people. Finding common words in Persian and Hindi with you was always so interesting for me. **Yeshwanth**, I'm still waiting for your promised Dungeons and Dragons session. **Meng**, we always had a special connection since we were the two DBL PhDs that started together. I hope to see you again. **Jose** and **Ziqi**, I wish you guys the best for the next steps of your career.

I would like to thank my collaborators, who helped me with the projects I did in my PhD. **Bianca**, supervising your bachelor's end project led to our paper. Thanks for sticking to the project and investing all that time in it. **Ekaterina**, supervising your master's project was a very fun experience. **Kirsten**, I would really like to be around to see your flash mob live. **Erin**, working with you on drug-producing yeast was very interesting. I hope the paper gets accepted. **Laura**, it was such a fantastic opportunity that you got stuck in the Netherlands at the start of COVID, so we got all the time to work on our project.

Next, I would like to thank **Marunka**, **Ruud**, **Saskia**, and **Azza** for your constant support in this journey.

Lastly, I would like to thank my Iranian friends. **Zivar**, I have known you for a long time, and we have amazing fun memories; thanks for supporting my enthusiasm for the Black Cats band. **Khatere**, you are always next to us in the important moments of Yosra's and my life; thanks for that. **Amirmasoud**, we had such nice trips to Paris and Prague together; it's time to decide our next destination. I wish to see you every day like I used to in the first year of my PhD. **Leila**, I'm very honored to have found such a nice person during my time in Delft. **Parviz** and **Zahra**, stop avoiding us. **Mousa** and **Farinaz**, I wish you two a happy life. **Hoda**, stop blaming my laziness and visit us. **Lotfol**, mesle chi shodi emshab? **Ahmadreza**, being your neighbor, was one of my amazing experiences during my PhD. Thanks for all the cooking you did.

To everyone who has been part of my PhD journey, your role has been invaluable. Thank you for being part of this significant chapter in my life.

Curriculum Vitæ

Ramin Shirali Hossein Zade

26-03-1992	Born in Ahvaz, Iran.
2010–2015	**Bachelor of Science in Chemistry** Qom University, Iran
2015–2018	**Master of Science in Information Technology** Sharif University of Technology, Iran
2018–2022	**Doctoral Candidate** Delft University of Technology, The Netherlands

LIST OF PUBLICATIONS

- **Shirali Hossein Zade R**, Urhan A, Assis de Souza A, Singh A, Abeel T. HAT: haplotype assembly tool using short and error-prone long reads. *Bioinformatics*. 2022 Dec 15;38(24):5352-9.
- Cosma BM, **Shirali Hossein Zade R**, Jordan EN, van Lent P, Peng C, Pillay S, Abeel T. When do longer reads matter? A benchmark of long read de novo assembly tools for eukaryotic genomes. *bioRxiv*. 2023:2023-01. *Accepted in Gigascience journal*
- **Shiarli Hossein Zade R**, Abeel T. The effect of removing repeat-induced overlaps in de novo assembly. *bioRxiv*. 2023:2023-04. *Under review at Plos One*
- Henao L, **Shirali Hossein Zade R**, Restrepo S, Husserl J, Abeel T. Genomes of four Streptomyces strains reveal insights into putative new species and pathogenicity of scab-causing organisms. *BMC genomics*. 2023 Mar 23;24(1):143.

List of Presentations

Oral presentations

- Data structures in Bioinformatics, July 2022, Düsseldorf, Germany
- Bioinformatics and Systems Biology, July 2022, Lunteren, The Netherlands
- VIB International Conference on Polyploidy. June 2019, Gent, Belgium

Poster presentations

- Bioinformatics and Systems Biology, July 2019, Lunteren, The Netherlands