Data Driven Methods for Scaffolding Genomes with Hi-C

Aakash Sur

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:

Peter Myler, Chair

William Noble

Shawn Sullivan

Program Authorized to Offer Degree:

Biomedical and Health Informatics

University of Washington

Abstract

Data Driven Methods for Scaffolding Genomes with Hi-C

Aakash Sur

Chair of the Supervisory Committee:
Peter Myler
Biomedical Informatics and Medical Education

High-quality reference genomes are once again in vogue with the publication of the telomere-to-telomere human genome and several challenging plant and animal genomes. Recent efforts in genome assembly have coalesced around two key technologies – ultra-long reads and genome chromatin conformation capture (Hi-C). Here, we used both to complete the protist genomes of *Leishmania donovani, Leishmania tarentolae, Crithidia fasciculata, and Euglena gracilis*, shedding light on their genomic organization and evolutionary history. To navigate the many Hi-C genome scaffolding methods, we benchmarked the most popular methods against a set of high-quality reference genomes. We found that while most can operate well under ideal circumstances, many struggle with using modern high-quality assemblies which contain near chromosome length contigs. Finally, we attempted to overcome these limitations using a machine learning approach by leveraging the recent bounty of genomes that have been published

with Hi-C. Using an innovative convolutional neural network, we demonstrated a proof of concept for a data-driven approach to scaffolding genomes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

My career has been a journey involving countless people, each of whom have left a lasting impression and an impetus for changing my path through life. Naturally, my graduate school career is strongly centered around my lab so I would like to thank Peter and all the members of the Myler lab for their many years of support. The bioinformatics group with Isabelle and Sandhya were always around for work questions, but most fun of all was to just there to hang out at the Evoke café with Samira. Sandhya has been incredibly supportive and somehow always knows how to navigate the various challenges life throws at you.

As part of my committee, Peter, Bill and Shawn helped to guide the ideation and completion of my thesis. Peter has been an unparalleled resource for *Leishmania* genomics, and I think we were able chip away at a few persistent questions and contribute to the field together. Bill's group has been a second home for me, and with all the strong computational researchers there, it really helped push my skills to the next level. Bill was also instrumental in the drafting and submission process for several of the thesis related manuscripts. Working with Hi-C data and assemblies remains somewhat of an art form, and Shawn's experience with hundreds of libraries and species proved invaluable.

I'd like to thank my friends and partners through the years. Without their support and encouragement, I wouldn't have made it to where I am today. My college friends helped me become more competitive academically, which eventually led to taking several graduate level classes. One thing led to another and here I am at the finish line (perhaps after a few extra years).

Martin, Jerry, Brian, and everyone else, thanks for coming to my defense, I couldn't ask for a better crew. Loc, I envy your math skills, and love that we actually pulled the trigger on the fintech ideas, let's see where the future takes us. Chelsea, thanks for all the years of friendship, let's not lose any more of our best guys. My graduate school friends helped strike an enviable work life balance and helped my hobbies and teaching blossom. Will, we really put a great class together, and your work is always an inspiration to stay close to the cutting edge. Graham, Chethan, Jimmy, Hannah, and everyone, graduate school is impossible without the support of friends and colleagues, thanks for being there. Sam, thanks for pushing my climbing to the next level and introducing me to such a talented group of climbers, I'm so stoked to crush all the projects. To my Seattle friends including Kate and Steve, thanks for getting me into painting and gardening, it really helped get through those pandemic years. Tressa, I know you never doubted I would make to the finish line eventually, it feels like such a relief to finally make it. Jessie, thanks for all the support when I had to put my head down and grind out the work, I'm looking forward to seeing you cross the finish line next. I'd also like to thank my family, they undoubtedly set me down the path of science, long before I even knew it. I suppose in some ways, I can carry the legacy of two grandparents with PhDs in biochemistry. Not the least, thank you to all the pet people in my life - Lily, Beaker, Emmie, and Samira, you all have such indominable smiles.

I'd like to thank my teachers and professors through the years. Mrs. Goodwin cultivated my writing skills, something I didn't know I had before, and pushed me to join Plan II at UT. Prof. Hoad had the most enthralling Plan II class and a personality to rival Hemingway. Prof. Iverson, your Organic Chemistry class should be a model for how to run a college level course, I

modeled aspects of my own teaching style from yours. Prof. Marcotte, there's a direct line from your class to graduate school, I got so hooked I never stopped.

Much of the computational work was facilitated through the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system and funded by the STF at the University of Washington. During my thesis work, I used over 91 CPU years of time on Hyak, so it really was an integral part of the project.

DEDICATION

"We are such stuff as dreams are made on,
and our little life is rounded with a sleep."

Chapter 1. INTRODUCTION

The genetic material of a cell contains the vast majority of information needed to orchestrate its functions. In addition to containing the sequences of all RNA and protein molecules, the genome contains a myriad of genetic elements including regulatory sequences, mobile genetic regions, and repetitive sequences, which affect the operation of a cell. To decipher these complex phenomena using modern methods, researchers commonly use genome-wide interrogation assays. Such experiments fundamentally rely on mapping sequencing data back to a reference genome. Consequently, understanding the genetic sequence of a species is key to uncovering how it operates on a cellular level. The process of genome assembly aims to create accurate base-level representations of genomes using sequencing information.

1.1 GENOME ASSEMBLY

Experimental methods of genome assembly have dramatically changed in the last three decades. After Sanger sequencing was streamlined, the genomes of several key organisms were sequenced, including *Escherichia coli, Saccharomyces cerevisiae,* and *Homo sapiens*[1]. These sequencing efforts relied heavily on a divide and conquer approach, where the genome was split into a vast library of bacterial artificial chromosomes (BACs), which could be reliably grown and independently sequenced. In this approach, fragments of a genome generated from a restriction digest are cloned into plasmids which can be transfected and replicated in bacterial strains. The Human Genome Project made extensive use of this method, and over the course of

13 years and $3 billion, about 60,000 BACs were sequenced across multiple institutions[2]. This initial period of genome assembly was marked by high cost and high-quality genomes for a few species.

In 2006, Solexa released a commercial sequencer which utilized the sequencing by synthesis approach and ushered in the "second generation" of sequencing[3]. This breakthrough massively increased the throughput of sequencing and dramatically reduced the costs involved. As a result, individual labs could bypass the time-consuming BAC centered approaches and sequence the entire genome at one time. One critical limitation was the sequence length generated by these technologies. Compared to Sanger sequencing, which could produce close to 1,000 bp reads, second-generation sequencers initially produced reads as short as 35 bp, and later up to 150bp. With such short reads and assembly required beyond the purview of a given BAC, the computational task of genome assembly became considerably more challenging. Consequently, genome assemblers produced shorter contigs, which in turn led to draft assemblies of species with far more sequence gaps than the previous generation of reference genomes.

In 2011, third generation sequencing finally debuted with the release of PacBio's RS instrument[4]. Noticeably different were the incredibly long reads these instruments produced, on the order of 10kb. Later, Oxford Nanopore released their MinION sequencer in 2014 which relied on directly detecting bases through ionic changes as DNA passed through a channel, and produced read lengths of around 15kb[5]. The primary drawback of third generation sequencing, at least initially, was the comparably poorer sequence quality. The number of errors per base was dramatically higher, and in addition to having substitution errors, third generation sequencing often included indels. These limitations have gradually been overcome, with the latest iteration from PacBio known as HiFi, specializing in high fidelity long reads and producing accurate 10-

25kb reads[6]. In the last few years, Oxford Nanopore (ONT) has optimized their protocols to produce average read lengths upwards of 100kb, though they are perhaps the most error prone of the third-generation sequencing technologies[7]. These enormous leaps in sequencing length and accuracy have led to a new set of low cost, high-quality genomes.

Through the evolution of genome sequencing, the computational methods used to assemble reads have largely relied on the same theoretical framework: assembly graphs[8]. The crux of the method is to treat reads as nodes in a graph and the overlaps between them as the edges connecting nodes. This mathematical formulation allows us to keep track of all possible sequences a given set of reads could produce. Progress in assembly software has included developing more efficient data structures, better heuristics to find paths through the graph, and filtering or correcting low confidence reads. Despite improvements through the years, researchers have found limitations in the quality of assemblies produced by shorter reads. Such assemblies are invariably fragmented, contain collapsed repeats, and are unable to phase alleles properly[9].

1.2 Genome Scaffolding

With the surge of third generation sequencing, we are overcoming many of these challenges. Long reads can span complex sequencing regions with confidence, and as a result, assemblies are able to produce longer and fewer contigs. And yet, the ultimate goal of genome sequencing—to produce chromosome length sequences—remains out of reach. Even with the latest sequencing technology, assemblies for large genomes are likely to produce thousands of pieces[10]. To bridge the gap between contig and chromosomes, a scaffolding approach is required,

whereby contigs are grouped into their parent chromosomes, and then ordered and oriented correctly. There are five categories of scaffolding technologies: restriction mapping, linked reads, sequencing reads, reference-guided alignment, and chromosome conformation[11] (Figure 1.1).

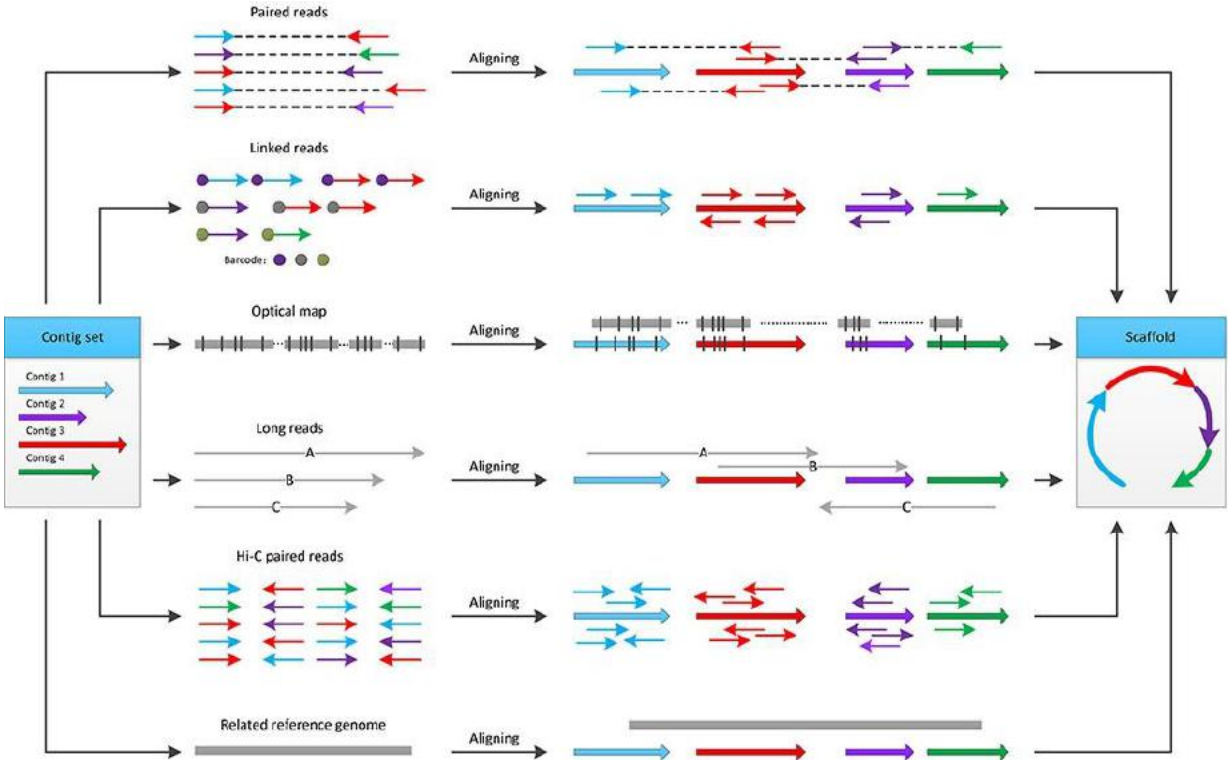

Figure 1.1: An overview of genome scaffolding methods[11]. There are five different categories of scaffolding technologies: paired end reads, linked reads, optical maps, long reads, Hi-C reads, and syntenic alignments. All methods attempt to bridge between contigs by using sequencing information.

Originally, the most common method to scaffold genomes was using restriction maps, where restriction digestion would reliably create patterns of fragments which could then be used to "fingerprint" segments of DNA and elucidate overlaps. This tended to be laborious, though

recent advances in automating the process have been significant[12]. Nevertheless, the method relies heavily on accurate representation of restriction sites in contigs, as well as access to specialized machines to perform optimal mapping. For several years, 10x Genomics offered another scaffolding approach, linked-read technology, which generated short reads from barcoded DNA molecules, offering the ability to trace back reads to contiguous sequences. With this information, contigs that were within 100kb of one another could be spanned, though the remaining ones still could not be placed [13]. Another approach was to simply reuse regular sequencing data, whether short reads, long reads, or mate pairs, to scaffold contigs. Rather than focusing on the underlying sequence, scaffolders would only attempt to organize contigs using sequence overlap information. Though they can create larger scaffolds from contigs in assemblies, these methods generally do not provide enough information to completely scaffold genomes and are always limited by the length of the molecules being sequenced. Reference-guided approaches to scaffolding use the high-quality genome of a closely related species to scaffold contigs based on their alignment to chromosomes. Though powerful, the fundamental assumption of this method is that the chromosomal organization of two species is exactly the same, which does not always hold true. Indeed. However, countless examples in evolutionary genomics point to even close relatives undergoing some level of genetic rearrangement.

One of the most promising scaffolding methods in the last decade uses genome-wide chromosome conformation methods, particularly Hi-C. In this approach, information about proximal regions of the genome is used to scaffold contigs, and distances between every pair of regions of the genome can be determined. Hi-C has widely overtaken other scaffolding approaches and has become a standard procedure in many genome sequencing initiatives[14]. The T2T initiative recently published the complete sequence of the human genome without any gaps,

and they used Hi-C as part of their validation process (though not part of the scaffolding step)15. The Vertebrate Genome Project (VGP) has a dedicated pipeline that scaffolds genomes with Hi-C[16], and the 10KP plant sequencing initiative incorporates many of these processes as well[17]. Currently, the sequencing formula for producing the highest quality genomes possible involves three pillars: HiFi PacBio reads, ultra-long Nanopore reads, and Hi-C reads[18].

1.3 Genome-Wide Chromatin Conformation Capture

Though we have grown familiar with reference genomes as a linear sequence of characters, we know that in fact they reside in the cells as three-dimensional structures. In 1879 Walter Fleming described mitosis, perhaps the first account of chromosomal organization of genetic information[19]. Since then, our understanding of the organization of DNA has continued to improve, with new discoveries being made every few years and new technologies developed to interrogate the three-dimensional structure. One of the most recent innovations in the field was the development of genome-wide chromosome conformation capture (Hi-C)[20]. The method relies on chemically cross linking the cell such that its DNA is frozen in space (Figure 1.2). The genome is then digested, exposing breaks near regions where two pieces of DNA are proximal in 3D space. These breaks are ligated, and through a clever series of steps, a library is generated to produce reads where each half of the read is from a neighboring region in 3D space.

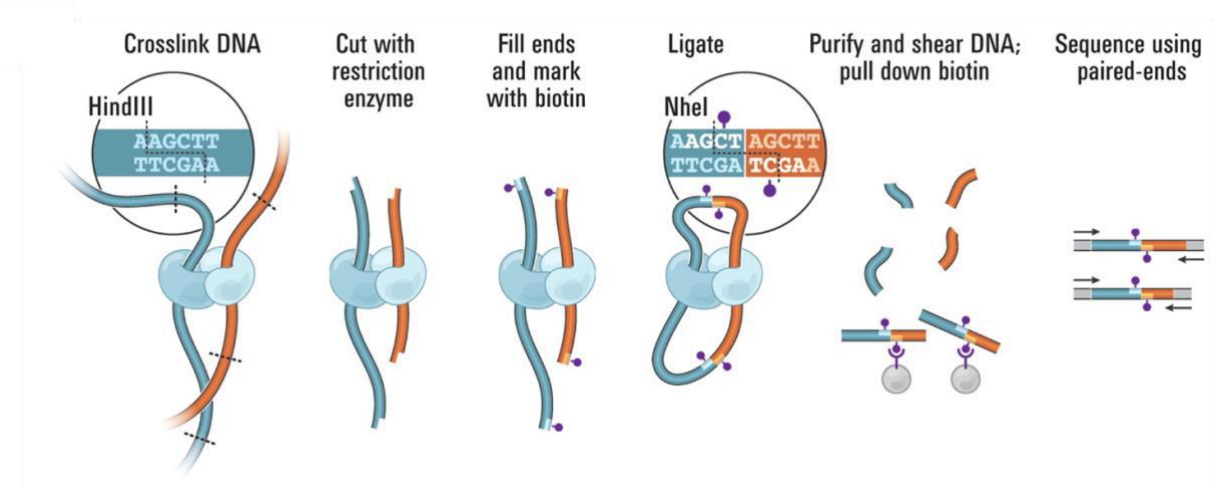

Figure 1.2: The steps involved in conducting Hi-C[20]. The cell is first chemically frozen in space using a crosslinking agent. Then, DNA is digested with a restriction enzyme and dangling ends are repaired with biotinylated nucleotides. Next, proximal DNA fragments which have been crosslinked are ligated. Finally, DNA is sheared and purified by streptavidin beads to isolated junctions, yielding a classic next-generation sequencing library.

One of early insights derived from Hi-C was that cells maintain structure in their chromosomes by co-locating regions of the genome that linearly neighbor each other. In other words, the one-dimensional positioning of a sequence is strongly related to its three-dimensional positioning. This key finding, known as the genomic distance effect, allows us to utilize Hi-C data to scaffold genomes. Since 2013, with the release of Lachesis[21], a number of Hi-C scaffolding methods have been published. They largely rely on creating a graph where contigs are the nodes and edges are weighted by the number of Hi-C reads spanning contigs. Each method is distinguished by a slightly different set of heuristics and what parameters they require, as well as the algorithmic approach to generate scaffolds from the graph. Given the proliferation of Hi-C scaffolding methods, and their importance in creating reference genomes, the Hi-C scaffolding process and the many software tools warrant further study.

In the course of this thesis work, we explored the Hi-C scaffolding process by completing the genomes of several neglected tropical parasites and organisms - *Leishmania tarentolae, Leishmania donovani,* and *Crithidia fasciculata*. Our work with *Euglena gracilis* explores the limits of the genome assembly process and some of the experimental hurdles when it comes to preparing Hi-C libraries. We then conducted a literature search to identify the most widely used Hi-C scaffolders. Using the five most popular methods, we devised a comprehensive set of tests to benchmark scaffolding accuracy. Surprisingly, we found that several methods performed worse with higher quality input assemblies, suggesting they are not compatible with the most recent generation of ultra-long read assemblies. Finally, we developed a machine learning approach to scaffolding genomes with Hi-C, leveraging the increasing number of genomes that are being published after manually curated corrections. This supervised approach shows promising results in learning correct scaffoldings from solved reference genomes and transferring them to new genome assemblies. We hope this body of work can guide those who aim to scaffold new genomes, and those who are building new scaffolding methods.

Chapter 2. SCAFFOLDING GENOMES WITH HI-C

2.1 INTRODUCTION

The Trypanosomatidae family contains a diverse set of single-celled parasites that can infect a wide range of hosts, including humans, domestic animals, lizards, insects, and even plants. After malaria, leishmaniasis causes the second-most number of deaths for parasitic organisms[22], with several forms of the disease also resulting in non-lethal, but debilitating, symptoms. Consequently, trypanosomes have been studied for several decades, not only for their disease burden, but also for their unique molecular biology. In 2005, the first *Leishmania* genome was completely sequenced[23], though subsequent genomes came at a slow pace. With the widespread adoption of third generation sequencing, renewed efforts have resulted in a number of recent genome publications, including the gap-free genome sequences for two strains of *Leishmania donovani*[24,25] and a phased assembly of *Trypanosoma brucei*[26]. Here, we present completed genomes of *L. donovani* 1S, *L. tarentolae* Parrot, and *Crithidia fasciculata* C1, as well as a near-complete genome of *Euglena gracilis*.

The *Leishmania* genus contains a number of clinically important species known to cause several forms of leishmaniasis. *L. tropica, L. major,* and *L. aethiopica* primarily cause cutaneous leishmaniasis, the most common form of the disease. The *Viannia* subgenus is responsible for mucosal leishmaniasis, most often by *L. braziliensis* and *L. panamensis*. Finally, *L. donovani* and *L. infantum* cause the most severe form of disease, known as visceral leishmaniasis, which has a fatality rate of over 95% if left untreated[27]. Historically, *L. tarentolae,* a species isolated from

reptiles has been used as a model organism given its strong ability to grow *in vitro*. Additionally, since *L. tarentolae* has evolved to infect geckos, its non-pathogenicity towards humans makes it especially easy to work with in most laboratory conditions.

The genus *Crithidia* contains species that exclusively infect insects, with several (including *C. bombi* and *C. mellificae*) being associated with bee colony losses[28], and *C. fasciculata* infecting mosquitoes. Similar to *L. tarentolae, C. fasciculata* has also been used extensively to study the biochemistry and unique transcriptional control of trypanosomes. Perhaps the two most identifying molecular features of trypanosomes are their extensive use of polytranscriptional units (PTUs) and presence of interlocking circular mitochondrial DNA (kDNA)[29]. Several key findings regarding RNA editing, *trans*-splicing, and kDNA replication were worked out in *L. tarentolae* and *C. fasiculata*.

Euglena gracilis, is a free living single-celled flagellate in the same phylum (Euglenozoa) as the trypanosomatids. Though this species is quite divergent from the trypanosomes, since it containing chloroplasts and lacks any pathogenicity, it still contains a base modification unique to this section of the tree of life. Base J is a glycosylated thymine that is thought to be responsible for transcriptional termination and silencing[30]. *E. gracilis* has been shown to have potential utility in biofuel production, environmental remediation, and biomolecule synthesis. Despite the international interest the species has garnered, its genome remains in an abysmal state, and basic facts about its genetic organization remain unknown. In addition to an unknown chromosome count, its ploidy has also not been established, and estimates on genome size vary widely, from as low as 332mb[31] to as high as 9gb[32].

Using long read sequencing and genome-wide chromatin conformation capture (Hi-C), strides have been made towards producing chromosome length reference genomes. Combining

these technologies with traditional short read sequencing, we can correct the frameshift errors that are commonly present in long read sequencing. With Hi-C sequencing, we are able to confirm the chromosome organization of *L. donovani* and *L. tarentolae*, which were originally accomplished using an alignment map. In addition, we make several corrections to the previously proposed chromosomes of *C. fasciculata* using the Hi-C data. Finally, we encountered experimental challenges while using Hi-C with *E. gracilis*, and although we are unable to complete its chromosomes, we present high-quality draft assembly with approximately 1000x less fragmentation than the preexisting draft assembly.

2.2 METHODS

2.2.1 *Culturing and DNA Extraction*

The *L. tarentolae* Parrot-TarII wild-type and *L. donovani* 1S strains were grown in SDM-79 medium supplemented with 10% fetal bovine serum. *E. gracilis* samples were obtained from Mark Field's lab at the University of Dundee, and *C. fasciculata* C1 samples were obtained from Steve Beverley's group at the University of Washington St. Louis. Cells were resuspended in a breaking buffer (2% Triton X-100, 1% SDS, 100mM NaCl, 10mM Tris, 5mM EDTA) with proteinase K and incubated at 37 C for 2 hours. DNA was extracted in phenol, chloroform, isoamyl alcohol (PCIA) and precipitated in ethanol. DNA was treated with RNase A, extracted with PCIA, extracted again with chloroform and isoamyl alcohol, precipitated with ethanol, and resuspended in Tris EDTA buffer pH 8.5.

2.2.2 *Library Preparation and Sequencing*

For standard Illumina libraries, DNA was sheared to 400 bp using the Covaris S2. Libraries were prepared using the NEBNext Ultra II library prep kit (NEB, E7645S) following the manufacturers protocol with indexed adaptors ordered from IDT and annealed by incubating at 95°C in annealing buffer (10mM Tris, 1mM EDTA, 50mM NaCl) for 5 minutes and slowly cooling to room temperature. Long-read libraries were generated using PacBio's DNA Template Prep Kit and sequenced on RSII P5/C3 instruments. For Hi-C libraries, strains were grown until 10^8 cells, then crosslinked with 1% freshly prepared methanol-free formaldehyde at room temperature for 20 minutes, and then quenched with 125mM glycine. Cells were washed in PBS and used with the Phase Genomics ProxiMeta kit to generate libraries. Both standard Illumina and Hi-C libraries were sequenced on HiSeq instruments to generate 75bp paired end reads. We also incorporated several libraries prepared by other groups including publicly available Roche 454 libraries from *C. fasciculata* generated by Steve Beverley's group at University of Washington St. Louis. For *E. gracilis*, we used the published libraries generated by Thankgod Ebenezer from Cambridge University containing several standard and mate pair Illumina libraries, as well as PacBio libraries generated from Purificación Lopez at the University of Paris-Sud, and Nanopore libraries from Pierre Cardol at the University of Liege. For scaffolding *E. gracilis*, we incorporated 10x link-read libraries generated from Neil Hall's group at the Earlham Institute, and all available RNA-seq libraries from the SRA database.

2.2.3 *Genome Properties and Assembly*

To estimate genome size, ploidy, levels of heterozygosity, and repeat content, we used GenomeScope2 and Smudgeplot[33]. Using short read data, k-mer counts were computed to

estimate various genome properties. For genome assembly, we first error-corrected long reads with short read data using FMLRC2[34], and in the case of *E. gracilis*, the long reads included nanopore data generated by Pierre Cardol's group at the University of Liege. Error-corrected long reads were passed to Canu using default parameters[35]. We removed haplotypes from the resulting assemblies using purge_dups[36]. Finally, assemblies were scaffolded using Hi-C by following the Juicer and 3d-dna pipelines, and then corrected *via* manual curation[37]. For *E. gracilis*, due to the challenges in assembling a genome, we aggregated all publicly available genome sequencing which included mate pair libraries (Table 2.1). To utilize these in a genome assembler, we additionally ran Masurca[38], which took as input two standard Illumina libraries, a mate pair Illumina library, a PacBio library, and a nanopore library. To scaffold *E. gracilis*, we first used long read data using LongStitch[39], then with RNA-seq data using p_rna_scaffolder[40], and finally with 10x genomics data using Arcs[13].

Species	Type	Location	Reads	Base Pairs
L. tarentolae	PacBio	Seattle	1,360,815	7,198,339,498
L. tarentolae	Illumina	Seattle	144,238,646	21,635,796,900
L. tarentolae	Hi-C	Seattle	59,964,841	9,594,374,560
L. donovani	PacBio	Seattle	838,529	6,574,356,845
L. donovani	Illumina	Seattle	355,821,859	53,373,278,850
L. donovani	Hi-C	Seattle	78,641,799	11,815,666,800
C. fasciculata	PacBio	Seattle	412,577	3,126,269,594
C. fasciculata	Illumina	St. Louis	41,508,521	21,922,416,870
C. fasciculata	454	St. Louis	9,816,376	3,641,141,835
C. fasciculata	Hi-C	Seattle	87,425,815	13,113,872,250
E. gracilis	PacBio	Seattle	3,533,322	31,793,245,156
E. gracilis	PacBio	Paris	403,289	3,540,114,295
E. gracilis	Nanopore	Liege	410,593	1,870,759,526
E. gracilis	Illumina	Seattle	64,526,539	9,678,980,850
E. gracilis	Illumina	Cambridge	183,462,358	29,233,357,879
E. gracilis	Matepair	Cambridge	36,182,723	3,618,272,300
E. gracilis	10x	Norwich	270,669,346	64,825,308,367
E. gracilis	Hi-C	Seattle	451,398,791	67,709,818,650
E. gracilis	RNA-seq	Tokyo	681,091,367	127,112,768,228
E. gracilis	RNA-seq	Cambridge	44,229,922	8,576,704,580
E. gracilis	RNA-seq	Norwich	191,708,318	38,341,663,600
E. gracilis	RNA-seq	Ceske Budejovice	20,879,285	3,929,902,443
E. gracilis	RNA-seq	Liege	951,113,100	190,222,620,000

Table 2.1: A summary of all the sequencing data used for genome assembly and scaffolding of *L. tarentolae, L. donovani, C. fasciculata,* and *E. gracilis*. Though we generated most of the data for the trypanosomes, the complexity of *E. gracilis* led us to collate all publicly available genomic and transcriptomic data for the species, in addition to several unreleased datasets from collaborators.

2.3 RESULTS

2.3.1 *Genome Properties*

We generated several libraries of Illumina sequencing, which is known to provide short but accurate sequencing information. We utilized these short reads using reference free k-mer analysis methods, which can characterize the genome without using a genome assembly, and establish genome size, ploidy, repeat content and allelic variation (Table 2.2, Figure A.4 - Figure A.11). We confirmed that all the trypanosomes were diploid and had haploid lengths of close to 36mb. We found that *E. gracilis* likely contains three copies of each chromosome and has a haploid genome length of 616mb. These results represent a significant advance as neither the ploidy nor genome size have been historically established for *E. gracilis*. The repeat content in *Euglena* has long been suspected to be fairly high, and we confirm that 46% of the genome is repetitive. Additionally, *E. gracilis* has a very high heterozygosity rate of almost 5%, which in combination with the high degree of repetitiveness, could account for previous challenges in assembling this genome.

Species	Size (bp)	Chromosomes	Ploidy	Repeat	Heterozygosity
Leishmania donovani	34,640,000	36	2	25%	0.1%
Leishmania tarentolae	36,790,000	36	2	31%	0.2%
Crithidia fasciculata	37,080,000	29	2	29%	2.4%
Euglena gracilis	616,060,000	Unknown	3	46%	4.9%

Table 2.2: The genome properties of each species. We first determined ploidy using Smudgeplot, and then used that ploidy in Genomescope to estimate genome size, heterozygosity, and repeat

content. The repeat column refers to what percent of 21-mers have been repeated elsewhere in the genome. The chromosome count was derived from the scaffolded Hi-C maps.

2.3.2 *Genome Assembly*

We relied heavily on long read technologies to improve the contiguity of our assemblies. By running Canu with the error-corrected long reads, we generated assemblies for each species. At this stage, *L. tarentolae* had 341 contigs, *L. donovani* had 221 contigs, *C. fasciculata* had 451 contigs, and *E. gracilis* had 37,085 contigs. The N50 of a genome assembly is the size at which contigs of equal or greater length cover half the assembly. After using purge_dups to remove haplotigs, *L. tarentolae* had 77 contigs with an N50 of 694kb, *L. donovani* had 62 contigs with an N50 of 702kb, and *C. fasciculata* had 95 contigs with an N50 of 613kb. Using 3d-dna, we scaffolded each genome into chromosomes, and after manual correction of the results, we found that *L. tarentolae* had 15 gap-free chromosomes and a genome size of 31,943,030bp, *L. donovani* had 13 gap-free chromosomes and a genome size of 33,075,856bp, and *C. fasciculata* had 4 gap-free chromosomes and a genome size of 33,579,047bp. We also generated Masurca assemblies for these species, but we found the assemblies to aggressively collapse repeat regions compared to Canu.

We generated several draft assemblies for *E. gracilis*, the most promising versions were the Canu assembly with 9,689 contigs and a 152kb N50, and the Masurca assembly which had 2,791 contigs and a 669kb N50 (Table A.1). These results represent a quantum leap in genome quality, as the currently published draft assembly of *E. gracilis* contains 2,066,288 contigs and has a 955bp N50[31]. Unfortunately, we found that our Hi-C library contained too many PCR duplicates to provide informative scaffolding information. Instead of scaffolding with Hi-C, we

used a combination of strategies for *E. gracilis*. Using the long-read data, the RNA-seq transcriptome data, and 10x genomics linked read data, we made three successive rounds of scaffolding (Table A.2).

To explore the quality of these genome assemblies using an orthogonal method, we employed BUSCO to determine gene content[41]. BUSCOs are representations of conserved genes across eukaryotes and are expected to be found in every genome. For the three trypanosomes, we found that the published genomes had very similar BUSCO scores to our assemblies, indicating that existing assemblies already had fairly good base level representations of the genomes. However, we found a dramatic increase in BUSCO scores for *E. gracilis*, going from 4 complete BUSCOs to 53 for the Canu assembly, and 49 for the Masurca assembly (Table A.3, Table A.4). This 10-fold increase in core gene content likely means that there are fewer errors at the base level in addition to the increased contig lengths. In addition, we investigated how well the existing and independently derived transcriptome aligned to our assemblies using Magic Blast, a newer version of Blast streamlined to align RNA (Table A.5). We found that on average, 90% of the sequence of a gene was able to align to the genome, compared to just 61% on the previous assembly. Finally, as a manual curation step, we aligned the gene for JBP1[30], and the three largest genes in the transcriptome with continuous open reading frames to both of our assemblies. In general, we found that both assemblies were able to align well to these genes and aligned to each other around these regions. However, on flanking regions, there was far less sequence identity, suggesting a more careful look with Hi-C is required.

2.3.3 *Chromatin Organization*

After the scaffolding and manual correction was complete, we generated Hi-C maps for each species (Figure A.1, Figure A.2, Figure A.3). Immediately, it is striking how the trypanosomes lack a classic checkboard pattern indicative of higher-level organization such compartments or topologically associated domains (TADs). For the most part, there are no organizational compartments in the genome, aside from the usual separation of chromosomal arms. Relatedly, the centromeres appear to strongly interact between all the chromosomes, which allowed us to annotate the precise location of every centromere for the first time in these species. Several repeat structures appear as diagonal lines, indicating that there is a similar but not identical sequence for the repeat. Notably, one such instance is the ribosomal RNA locus of chromosome 22 on *C. fasciculata* (Figure 2.1). This chromosome also displays a divergence of signal between its arms, as if they were completely separate chromosomes. In the yeast genome, a similar phenomenon occurs for Chromosome XII, where the half of the chromosome containing ribosomal RNA repeats is segregated in the nucleolus and does not interact with the other half of the chromosome[42]. In the *C. fasciculata* genome, there are additionally several sections of missing sequencing which are apparent from the drop in Hi-C signal. One such instance is on chromosome 5, which contains the spliced leader RNA locus, which is likely present in a far greater copy number than is currently represented in the genome (Figure 2.2).

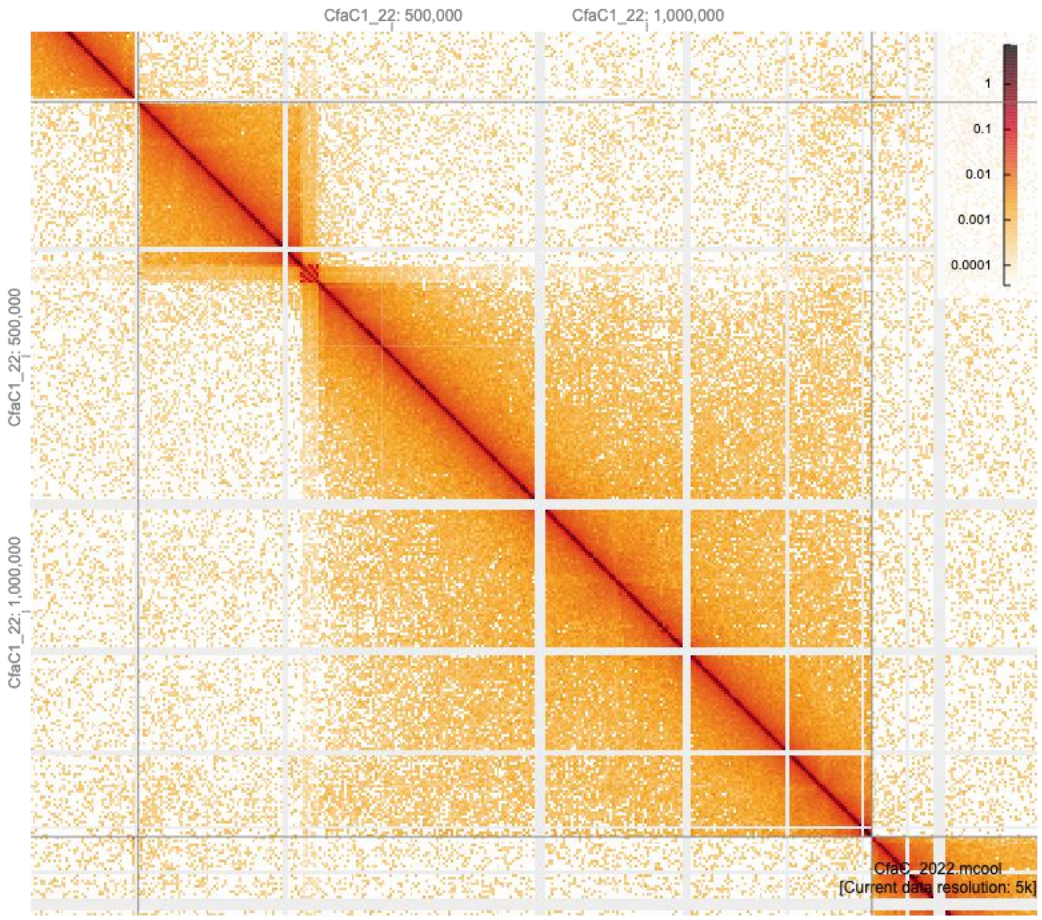

Figure 2.1: The Hi-C interaction map of Chromosome 22 of *C. fasciculata*. Around 320kb, the ribosomal RNA locus begins and contains several repeats, each indicated by another diagonal striation. Interestingly, this region segregates the chromosome into two sections that do not interact with each other.

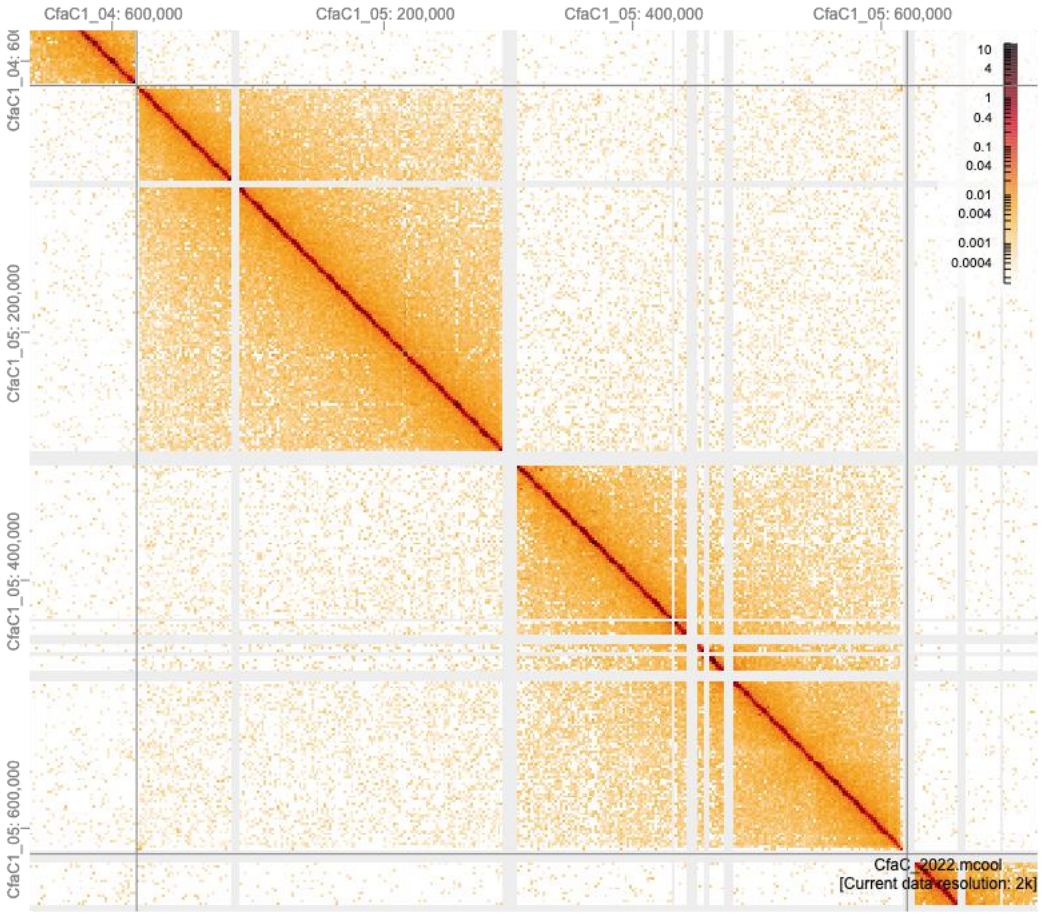

Figure 2.2: The Hi-C interaction map of Chromosome 05 of *C. fasciculata*. The greyed-out section is the splice leader locus which contains the sequence that is required to be trans-spliced on to all genes being transcribed in the organism. From the sequence coverage map, there is strong evidence that this region is significantly underrepresented, suggesting that this chromosome is missing a substantial amount of repetitive sequence in its current state.

2.4 DISCUSSION

We performed long read, short read, and Hi-C sequencing to produce high-quality reference genomes for four euglenozoa species. We succeeded in producing chromosome length scaffolds for three of these species (*L. tarentolae, L. donovani,* and *C. fasciculata*) and produced

a high-quality draft of *E. gracilis*. Though these are all polyploid species, we sought to produce a haploid representation of each genome. We used purge_dups to identify and exclude haplotigs and focus on scaffolding the primary contigs. However, several groups have demonstrated the ability to phase haplotypes using Hi-C data[43], and future work could conceivably transform this into a polyploid representation.

One of the reasons we were unable to complete the chromosomes of *E. gracilis* was due to the Hi-C library not containing a sufficient diversity of molecules. We found that our Hi-C library had a PCR duplication rate of 66%, suggesting a poor DNA extraction step. We are currently collaborating with Pierre Cardol's group at the University of Liege to repeat the Hi-C library preparation and sequencing. Should this replicate succeed, we expect to be able to validate the existing assemblies and their contigs, scaffold contigs into chromosomes, and definitively determine the number of chromosomes.

One interesting finding from the Hi-C maps of the three trypanosomes was the general lack of topologically associating domains (TADs) and chromatin compartments. These organisms are thought to have evolved early in the history of eukaryotes, and we speculate that complex chromatin organization had not yet evolved on the tree of life at this point. As such, the Hi-C contact maps show only low levels of defined organization, much like those of bacterial genomes[44]. Nevertheless, it has not been studied whether the promastigotes and amastigote forms of trypanosomes have significant differences in chromatin architecture, though we do not have reason to suspect they do.

The final step towards publishing these genomes and producing high-quality reference sequences for the community is to generate annotations of commensurate quality. Often, protein coding sequence annotations are merely transferred based on sequence alignment, and transcript

and gene boundaries are left out altogether. We plan to use RNA-seq libraries in conjunction with splice leader seq libraries to accurately determine the 5` and 3` ends of each of the transcripts. Since there can be multiple splice sites in trypanosomes, we can use the furthest splice site for a given CDS to define gene boundaries and use the most common site to define transcript boundaries.

With ubiquitous use of genome-wide sequencing experiments to answer biological questions, the absence of a high-quality genome for any given species hampers research. With accurate DNA, transcript, and coding sequences, omics experiments can capture a wider range of information with better correspondence to the underlying truth. We hope the publication of these species will further parasitology and help reveal the origins of this unusual branch on the tree of life.

Chapter 3. MEASURING SCAFFOLDING ACCURACY WITH EDIT DISTANCE

This chapter is adapted from the following work:

> Sur, Aakash, William Stafford Noble, Shawn Sullivan, and Peter Myler. "Edison: measuring scaffolding accuracy with edit distance." *bioRxiv* (2022).

3.1 INTRODUCTION

The reference genome of a species is the starting point for the many types of sequencing experiments. Accordingly, errors in the reference genome often propagate through subsequent analyses. The critical task of constructing the reference genome is handled by genome assemblers, which distill large sets of genomic reads into stretches of contiguous sequence. Ideally, a genome assembly contains chromosome-length sequences, but in practice only subregions of high confidence, known as “contigs,” can be independently assembled. Arranging these contigs in the correct chromosome grouping, order, and orientation is known as the scaffolding problem. Although the scaffolding problem remains challenging, advances in experimental methods such as chromatin conformation capture have allowed researchers to publish high-quality scaffolds for historically difficult genomes[45–47].

Despite the importance of scaffolding to the assembly process, there is little agreement on how to assess the quality of a given scaffold. Since 1995, when the term “scaffolding” was introduced[48], there has been a steady flow of new scaffolding algorithms, each with its own evaluation criteria (Table B.1). Typically, evaluation criteria fall into three categories: length

metrics, visual plots, and error counts. The most common length metric is N50, which is the size at which contigs of equal or greater length cover half the assembly. Though useful, since this metric only characterizes the length of the scaffolds, it does not evaluate the placement of contigs, which can lead to overly aggressive scaffolders receiving higher scores for otherwise inaccurate scaffolds. Visual inspection using dot plots and linkage maps can confer a sense of agreement with a reference genome but does not yield quantitative measurements. In contrast, enumerating the errors in scaffolds creates quantifiable values, but there are many choices in defining these errors.

We suggest using the concept of edit distance, which measures the minimum number of edits (splitting scaffolds, joining scaffolds, moving contigs, and inverting contigs) required to fix misplaced contigs, to encompass all these flavors of accuracy. Edit distance has been studied in many contexts[49], and in the field of evolutionary genomics it has generally been defined as the most parsimonious series of rearrangements in gene order that would explain the evolution of one species into another[50–53]. In 2006, an elegant formulation called the Double Cut and Join (DCJ) model was introduced, which accounted for chromosomal fusions, fissions, translocations, and inversions[54]. However, edit distance has rarely been applied to the evaluation of scaffolds[55,56], and to the best of our knowledge, there is no current software tool to perform this task.

We have developed Edison (Edit Distance Scaffolding), an open-source Python program that uses the DCJ edit distance between a scaffolded assembly and a high-quality reference genome as a measure of scaffolding accuracy. This software package calculates the overall length-weighted edit distance (relative to the correct placement of contigs), along with individual scores for grouping, ordering, and orientation accuracy. By focusing strictly on contig placement

instead of sequence error, we can disentangle errors in the genome assembly step and the scaffolding step. As such, our measurement of scaffolding accuracy allows researchers to benchmark existing scaffolders and test new algorithms against known genomes to gain a better under- standing of performance compared to traditional metrics such as N50 and base level errors. Our findings show that on random permutations of the yeast genome, scaffolding accuracy better evaluates the state of an assembly compared to N50.

3.2 RESULTS

Edison begins by breaking scaffolds into their constituent contigs at regions with a stretch of Ns. Next, contig sequences are aligned to the reference genome using MUMmer4[57]. This alignment allows us to determine the optimal organization of contigs into scaffolds. To determine the edit distance, we compare this alignment-based scaffolding to the input scaffolds using the DCJ algorithm. The algorithm begins by constructing an adjacency graph that maps the positions of contigs in both assemblies onto a graph. The edit distance then becomes a simple relationship between the number of contigs and the number of even and odd paths in this graph (Appendix B). The edit distance is converted into accuracy by taking the total length of correctly placed contigs compared to the overall length of all the contigs (Figure 3.1).

To assist in the interpretation of the accuracy and edit distance, we break down three of its contributing factors, the grouping, ordering, and orientation scores. The grouping score represents what fraction of a scaffold belongs to a single chromosome, averaged across all scaffolds by length. The ordering score is the length-weighted percent of contigs that are next to their expected neighbors. The orientation score is similar to the ordering score, but contigs are

required to be in the correct orientation in addition to being ordered correctly. Finally, Edison also produces a visualization which displays the MUMmer4 alignments of the contigs to further illuminate how a particular assembly compares to the reference genome.

In the presence of a reference genome, we observe that accuracy is a much better indicator of correctness than the N50 of the assembly. To demonstrate this, we simulated a genome assembly by splitting the *S. cerevisiae* reference genome into equal sized 100kb contigs and scaffolding them according to their true chromosomal assignments. We then randomly made between 0 and 30 permutations to the assembly by moving contigs, merging scaffolds, and breaking scaffolds to create 1,000 permuted assemblies. Several of these scaffolds had considerably higher N50s than even the reference genome, which would have made them attractive candidates in a *de novo* setting. However, the accuracy of these assemblies is demonstrably lower due to the incorrect joins required to create these longer scaffolds.

Developing new scaffolders is a challenging task, which has to balance producing longer scaffolds with producing correctly joined scaffolds, and it has been shown that more aggressive scaffolding parameters accumulate more errors[58]. Because these scaffolders are most often used in the *de novo* setting, where a reference genome is absent, correctness is quite difficult to characterize. With Edison, we propose that researchers benchmark and test their scaffolders on assemblies of species with known reference genomes in order to compare performance between scaffolders and evaluate the strengths and weaknesses of competing algorithms.

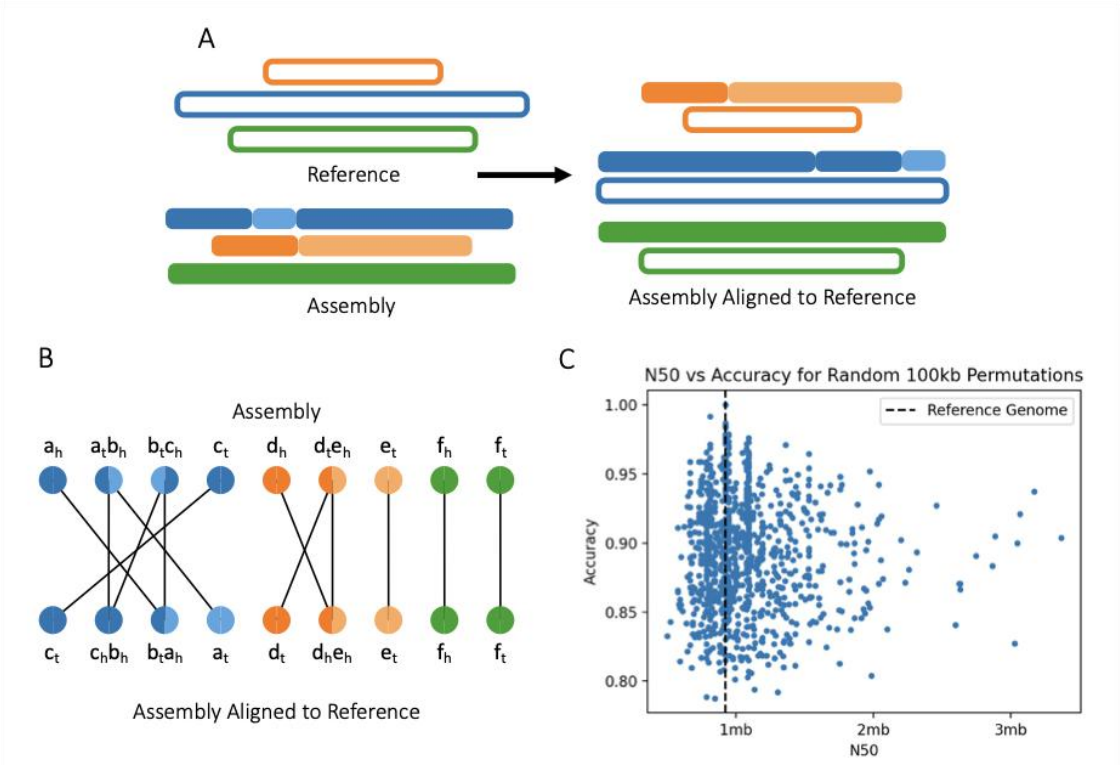

Figure 3.1: An overview of the Edison pipeline, and experimental results. A) Scaffolds in the assembly are first aligned to the reference genome to determine their optimal positions. B) We compare the original assembly and the assembly aligned to the reference by creating an adjacency graph. As described in the Double Cut and Join model, the adjacency graph can be used to compute the edit distance between these two layouts. C) Random permutations of the *S. cerevisiae* genome indicate that while the N50 can be artificially inflated, the accuracy cannot. The dashed vertical line represents the N50 of the reference genome.

Chapter 4. BENCHMARKING HI-C SCAFFOLDERS USING REFERENCE GENOMES AND *DE NOVO* ASSEMBLIES

This chapter is adapted from the following work:

> Sur, Aakash, William Stafford Noble, and Peter J. Myler. "A benchmark of Hi-C scaffolders using reference genomes and de novo assemblies." *bioRxiv* (2022).

4.1 INTRODUCTION

Robust genome sequences are foundational to molecular biology, yet many reference genomes remain in the purgatory of the draft assembly. Producing complete chromosome-length sequences of organisms has long been the goal for genome assembly, but progressing from high-confidence contigs to scaffolded chromosomes has proven challenging. Recent experimental advances in interrogating the three-dimensional structure of genomes through chromatin conformation capture (Hi-C) has provided valuable new information to help solve the assembly problem[21]. Several groups have developed scaffolding algorithms that utilize Hi-C data to group, order, and orient contigs into completed genomes. Despite the proliferation of such methods, the accuracy of these methods has never been comprehensively benchmarked.

A search of the current literature revealed ten Hi-C scaffolding methods, of which five (Lachesis, HiRise, 3d-dna, SALSA, and AllHiC) have been used in a publication more than three times (Table 4.1). In each of their respective publications, the authors test their scaffolder by solving a genome assembly, but the species and task vary considerably among papers.

Occasionally, a reference is artificially split into an arbitrary number of pieces and the scaffolder tasked with reassembling it (Salsa, AllHiC). In some cases, assemblies of known genome are scaffolded (Lachesis, HiRise, 3d-dna, Salsa, AllHiC), and other times unpublished genomes are solved (3d-dna, Salsa, AllHiC).

Our primary objective in this study is to evaluate the performance of existing Hi-C scaffolders using a uniform set of tests and evaluation metrics. To this end, we evaluated the five most commonly used Hi-C scaffolders against four diverse eukaryotic species — *Saccharomyces cerevisiae*, *Leishmania tarentolae*, *Arabidopsis thaliana*, and *Homo sapiens*. Each of these genomes has unique characteristics, including widely varying genome size, base composition, rate of interchromosomal interaction, and even evolutionary differences in chromosome packing.

Name	Publication Date	Citation Count	Application Count
Lachesis[21]	December 1, 2013	481	89
Dna-triangulation[59]	December 1, 2013	138	0
Graal[60]	December 17, 2014	108	1
HiRise[61]	February 4, 2016	349	17
3d-dna[37]	April 7, 2017	297	40
Salsa[62]	July 12, 2017	71	5
ALLHiC[63]	August 5, 2019	20	4
HiCAssembler[64]	October 10, 2019	8	0
HiC-Hiker[65]	May 5, 2020	0	0
Manual Annotation*	N/A	N/A	5

Table 4.1: The Hi-C scaffolding tools identified in the literature search along with their publication date, citation count, and number of times they were used in a genome publication as of September 15, 2020. Although scaffolders published earlier tend to have more citations, their actual application count varies significantly. Our selection criteria required scaffolders to have

three or more applications. *"Manual annotation" refers to studies that complete the genome by hand using the Hi-C map as a visual guide.

We tested the scaffolders with two different sets of contigs: split reference and *de novo* assemblies. The split assembly divides a reference genome into equal-sized pieces, presenting an artificially pristine test that is the best-case scenario for each scaffolder. Because real genome assembly is usually complicated by repeat ambiguity, haplotypes, and low complexity sequences, we created several *de novo* assemblies from long read datasets using the Canu assembler[35]. In both settings, scaffolders are tasked with joining a given set of contigs to produce scaffolds, which are then compared against the existing reference genome. We observed that scaffolders performed better on average on the split reference task than the *de novo* assembly task. Additionally, we found that the accuracy of the scaffolders changed with the species being tested, suggesting that sequence characteristics such as repeat content and heterozygosity levels may affect performance. On average, HiRise and Lachesis performed the best, with HiRise and Salsa working best on less fragmented assemblies, and HiRise, Lacheis, or AllHiC being better choices for more fragmented assemblies. Although scaffolders can perform well under ideal circumstances, our results suggest that existing Hi-C scaffolders are still expected to make mistakes, requiring manual correction before new reference genomes can be published.

4.2 METHODS

4.2.1 *Literature Search*

To find all available Hi-C scaffolders, we conducted a literature search on PubMed for publications between January 1, 2010, to September 15, 2020. The search terms "(hi-c scaffolding) or (hi-c assembly) or (hi-c genome assembly) or (hic scaffolding) or (hic assembly) or (hic genome assembly)" yielded 370 results, of which 171 were ultimately deemed relevant (Figure C.13). Ten scaffolding methods were identified, as well as the frequency with which they have been used to publish genomes. Methods with three or more cited applications were selected for benchmarking.

4.2.2 *Split reference and de novo assemblies*

To generate the split assemblies, we partitioned the established reference genomes of *S. cerevisiae*, *L. tarentolae*, *A. thaliana*, and *H. sapiens* into equal sized pieces of 10kb, 50kb, 100kb, 500kb, and 1mb to create a total of sixteen split reference assemblies. To generate a diverse set of *de novo* assemblies for each of the four organisms, we collected a large repository of long-read data from NCBI's SRA database, as well as our previously generated PacBio reads for *L. tarentolae* (Table C.7). To normalize for different genome sizes, we down-sampled the number of reads to achieve a theoretical coverage of 10x, 20x, …, 100x for each reference genome. We ran the Canu assembler on each of these down-sampled datasets using the default parameters to create forty different *de novo* assemblies (Table C.8). Canu was selected given its

popularity as a long-read assembler as well as its robust support for clusters and scheduling frameworks such as Slurm.

4.2.3 *Hi-C Alignments*

Hi-C scaffolders use the alignment of Hi-C reads against a genome assembly to optimally place contigs. We downloaded publicly available datasets of Hi-C reads from the SRA database for *S. cerevisiae*, *A. thaliana*, and *H. sapiens*, and used our previously generated Hi-C reads for *L. tarentolae* (Table 4.2). The Hi-C dataset for the human genome proved too large to easily work with so we opted to down-sample that dataset to 100 reads per kilobase. All reads were aligned against *de novo* and split reference assemblies using BWA[66], and duplicate reads were filtered with samblaster[67]. The resulting alignment files were sorted with samtools[68] into either coordinate-sorted files or read-sorted files depending on scaffolder requirements. 3d-dna required the use of the Juicer pipeline[69], so reads were extracted from the filtered BAM alignments to be input to Juicer.

Organism	**Hi-C Reads**	**Hi-C Bases**	**Hi-C Coverage***	**Hi-C Enzyme**	**BioProject**
S. cerevisiae	1,542,620,558	128,833,195,802	127,489	DpnII	PRJNA525842
L. tarentolae	59,964,841	4,797,187,280	1,862	Sau3AI	PRJNA818795
A. thaliana	790,614,589	159,575,826,006	5,856	DpnII	PRJNA227546
H. sapiens	15,307,281,379	3,088,242,262,764	4,938	MboI	PRJNA268125

Table 4.2: Overview of the Hi-C data collected for each of the four organisms. All the Hi-C enzymes cut at the same site, allowing the pipeline to remain consistent. The *H. sapiens* data proved to be computationally unwieldy, it was down sampled to 100 reads/kb. *Hi-C coverage is

reported as reads per kilobase since the number of bases in a particular read do not contribute towards the contact count.

4.2.4 *Hi-C Scaffolding*

Of the five scaffolders selected for benchmarking, two initially failed to build due to errors in the source code (Lachesis and HiRise), one required deprecated software dependencies (Salsa), and two installed as intended (3d-dna and AllHiC). To work with this diverse set of software tools and with an eye towards providing a community resource, we containerized the four methods with installation hurdles in Docker and have made them freely available (https://hub.docker.com/u/aakashsur). Scaffolders were then tasked with joining the contigs of a genome assembly using the alignment of Hi-C reads to that assembly. Lachesis and AllHiC require an expected number of chromosomes when running, representing a potential limitation of those methods. Additionally, HiRise was originally developed for use with *in vitro* chromatin proximity ligation reads (Chicago) rather than Hi-C data.

To determine how many Hi-C reads were required to effectively scaffold genomes, we selected a single assembly for each species and down-sampled the number of aligned and de-duplicated Hi-C reads to 1, 50, 100, 500, and 1000 reads per kilobase. Since the difficulty of the scaffolding problem is often related to the fragmentation of the underlying genome assembly, we also wanted to determine how assembly quality affects scaffolding. To test the effects of different N50s, the number of Hi-C reads were normalized by down-sampling each run to 100 aligned Hi-C reads per kilobase.

4.2.5 *Scaffolding Accuracy*

Each scaffolder outputs a FASTA file where the appropriate joins have been made to the input contigs. To evaluate how closely this layout matches the optimal layout, we used MUMmer4 to map the assembly contigs to the known reference genome, and then compared them[57]. Using the python package Edison, we calculate the edit distance, overall accuracy, grouping accuracy, ordering accuracy, and orientation accuracy[70]. The edit distance is the number of edits needed to alter a given set of scaffolds such that they most closely resemble a reference genome. This distance is calculated using the Double Cut and Join (DCJ) algorithm for genomic rearrangements, which guarantees finding the theoretical minimum edit distance[54]. The overall accuracy is obtained from the DCJ model by determining what fraction of sequence in the assembly has been correctly placed. The grouping accuracy measures how effectively a scaffolder can partition contigs into their associated chromosomes and is computed using a length-weighted Jaccard index between scaffolds and chromosomes. The ordering accuracy is the length-weighted frequency of finding a pair of adjacent contigs in the assembly that are also adjacent in the reference. Finally, the orientation accuracy is similar to the ordering accuracy, but also requires that adjacent contigs be correctly oriented relative to each other.

4.3 RESULTS

We benchmarked the five most utilized Hi-C scaffolders: Lachesis, HiRise, 3d-dna, SALSA, and AllHiC. To simulate ideal conditions, we challenged the scaffolders to reproduce the high-quality reference genomes of *S. cerevisiae, L. tarentolae, A. thaliana,* and *H. sapiens* that had been split into equal size pieces. In addition, to assess performance in a more realistic

setting, we benchmarked each scaffolder using *de novo* assemblies which approximate the reference genomes, but contain the ambiguities and complexities of real assemblies.

Several of the scaffolders proved challenging to install and run, so we have released patched versions as Docker containers at https://hub.docker.com/u/aakashsur. Despite these patches, there are inherent limitations to some of the software tools that lead to failed runs. Most commonly, Lachesis fails to finish if it is unable to build the specified number of chromosomes. Additionally, several scaffolders failed to complete the 10kb N50 split reference assembly of the human genome in the allotted 10 days — the maximum our cluster allows. In most cases, HiRise tends to have a run time that is an order of magnitude higher than the other scaffolders, and Lachesis consistently had the fastest time, even being able to scaffold human-sized genomes within an hour (Figure C.18).

To determine the impact of Hi-C read coverage on scaffolding ability, we down-sampled the number of Hi-C reads. As a baseline, we chose the 100kb split reference assembly to represent a plausible, modern assembly attempt. We found that the majority of the time, performance degrades as the amount of Hi-C data is reduced (Figure 4.1). Although we observed heterogeneity as to when this shift occurs, 50 reads/kilobase appears to be the point below which at least some of the scaffolders begin to perform worse. Lachesis and AllHiC appear to have both the smallest drop in performance suggesting that they are the most tolerant of less data. These results are broadly replicated in the *de novo* assembly setting as well (Table C.6).

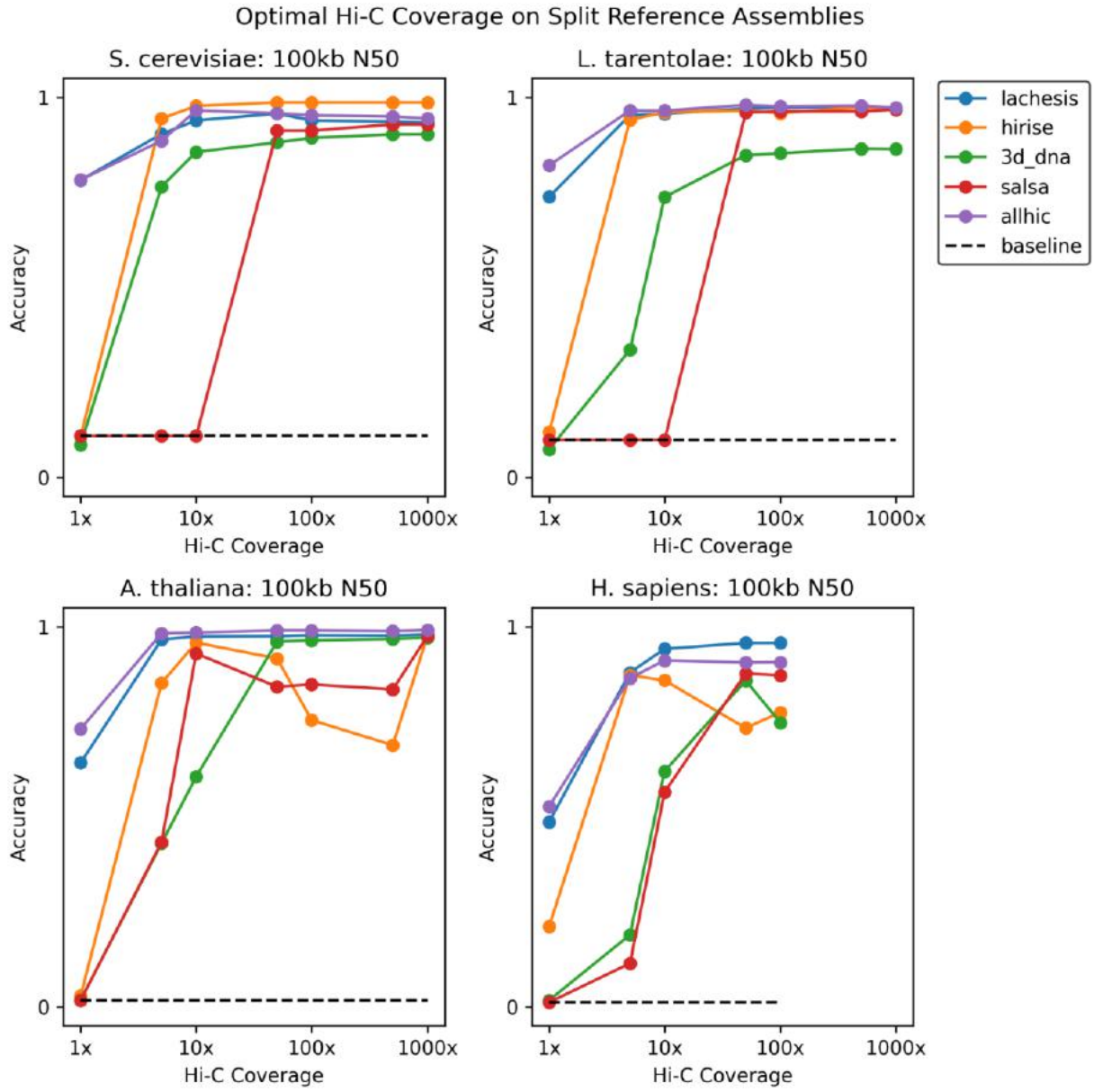

Figure 4.1: The effect of down-sampling Hi-C coverage on scaffolding accuracy. While using the 100kb N50 split reference assembly, reads were down sampled to target densities. Past 50 reads per kilobase, performance of all scaffolders tends to degrade, though some scaffolders are more resistant to decline than others.

To measure accuracy as a function of assembly size, we varied the N50 of the genome assemblies while keeping a constant 100 reads per kilobase of Hi-C data. For the split reference assemblies, we found that scaffolders can reach upwards of 80% accuracy in many cases, but occasionally perform much worse (Figure 4.2). Indeed, the best scaffolder for one species was not necessarily the best for another, making the calculus of ranking more challenging. Nevertheless, a consistent trend was lower performance on 10kb N50 assemblies, suggesting their high degree of fragmentation makes them difficult to scaffold. Though this might suggest that more contiguous assemblies ought to fare better, we were surprised to find that several of the scaffolders had dramatic decreases in performance with the highest N50s. In fact, on five occasions, scaffolders perform worse than the baseline of no scaffolding, indicating the errors they have created exceed the starting errors. For AllHiC, this decrease occurs because the method produces a single scaffold containing all the contigs (Figure C.22). Similarly, 3d-dna yielded consistently low grouping scores for these two species and also showed a decline in ordering accuracy at the high N50 range with similar problems in "over-scaffolding," *i.e.*, producing fewer scaffolds than there are chromosomes.

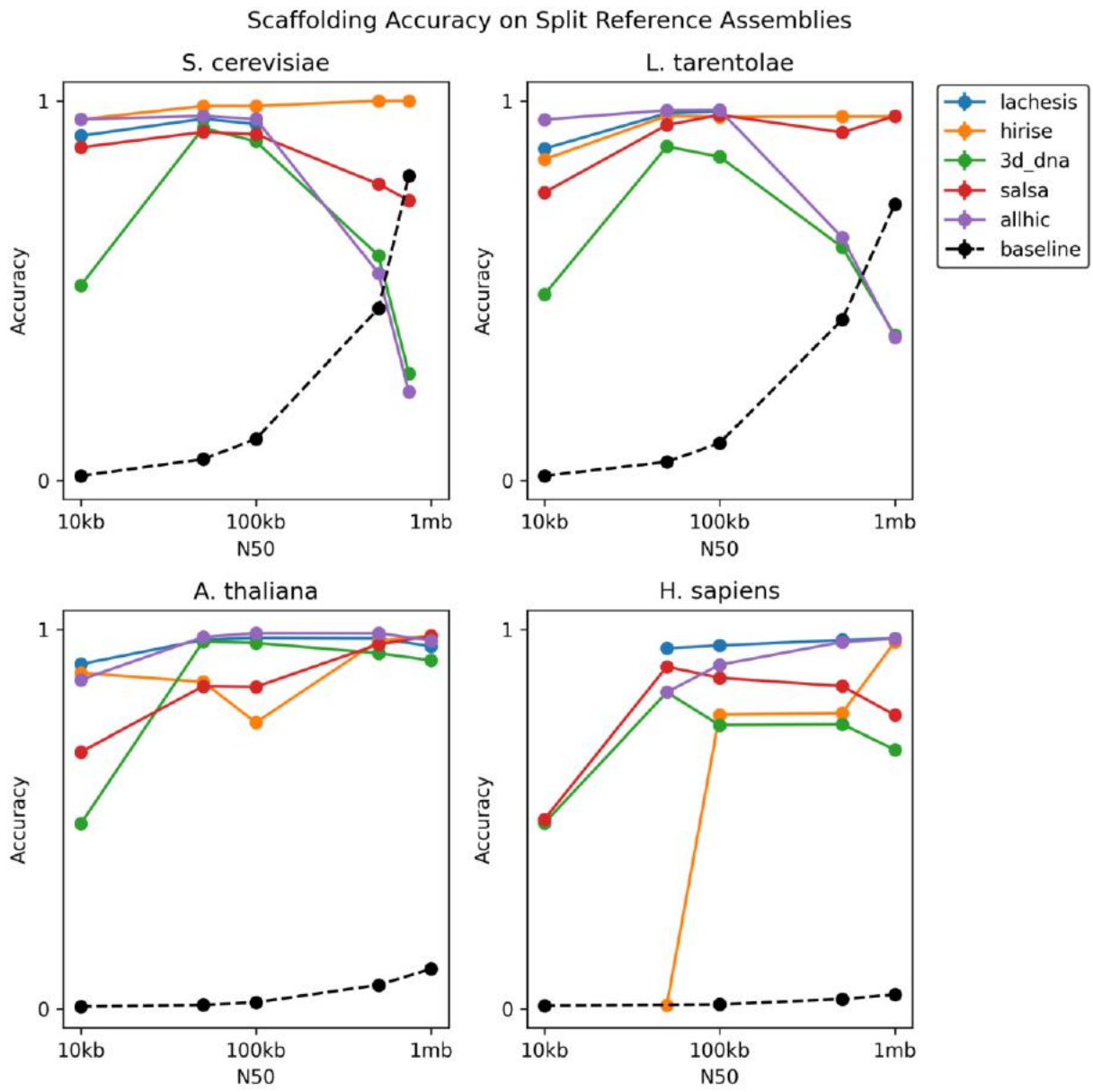

Figure 4.2: The accuracy of Hi-C scaffolders on four reference genomes. For each species, we created five different assemblies by splitting the reference in equal sized parts. High performance on a particular organism does not guarantee high performance on others. Performance decreases by 10kb N50 for all species, but also decreases at the high N50 range for *S. cerevisiae* and *L. tarentolae*.

For the *de novo* assemblies, significant variability in accuracy was observed across scaffolders and species, presumably due to the inherent complexities of the assembly process

(Figure 4.3). As a general trend, the scaffolders tended to perform worse on *de novo* assemblies than they did on split references. Particularly for *A. thaliana* and *H. sapiens*, accuracies are much lower and closer to baseline than in the split reference setting. Overall, we still see the trend where accuracy decreases at both extremes of the N50 spectrum, albeit in a less consistent fashion. Again, AllHiC and 3d-dna perform significantly worse on the more contiguous *S. cerevisiae* assemblies, similar to their trends on the split reference task.

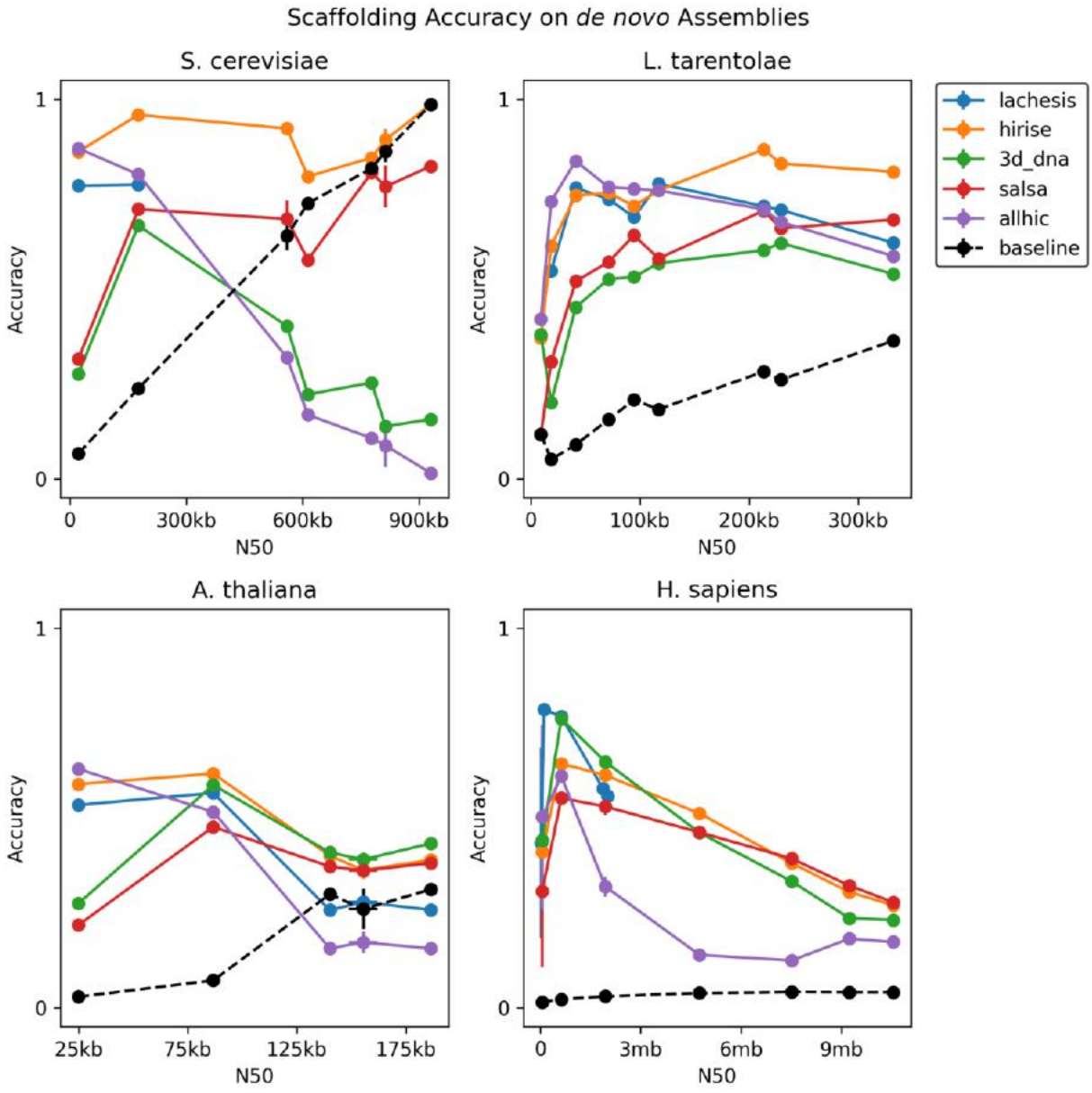

Figure 4.3: The accuracy of Hi-C scaffolders on the *de novo* assemblies of four species. The scaffolders exhibited significant variability across species, as well as an overall lower performance compared to the split reference reconstruction task. Nevertheless, a similar trend of poorer performance at either end of the N50 spectrum remains, with highly fragmented assemblies causing poorer performance, and highly contiguous assemblies causing a drop as well.

While investigating the lower performance on the *de novo* assemblies of *A. thaliana* and *H. sapiens* compared to the split reference setting, we discovered that approximately fifty percent of the Hi-C reads for the *A. thaliana* assemblies mapped to more than one location, in comparison to about ten percent for the other species (Figure C.24). Because multi-mapping reads cannot be reliably used in the scaffolding process, Hi-C scaffolders typically mask these regions during preprocessing. We speculate that this led to the poor scaffolding performance on *A. thaliana* assemblies.

While the culprit for low accuracy on *de novo A. thaliana* assemblies was the high repeat content, the same cannot be said for *H. sapiens*, which shows a multi-mapping rate on par with the other two species. Instead, our Mummer alignments of scaffolds hinted at a high frequency of small, overlapping contigs, known as haplotigs, which are typically caused by allelic variation (Figure C.21). Haplotigs are known to interfere in scaffolding since they create a scenario in which two different sequences belong to the same location along the genome. Since the goal of most assembly projects is to produce a haploid representation of the genome, haplotigs can be reasonably omitted from analysis. We found that most Hi-C scaffolders automatically exclude haplotigs in scaffolding, because they are small and therefore contribute a relatively small amount of Hi-C signal compared to the primary contig in a region. Consequently, excluding unscaffolded regions of the assembly led to only a 5% reduction in assembly size on average. The scaffolding accuracy of these assemblies improved dramatically for HiRise, Salsa, and 3d-dna, suggesting that the presence of haplotigs can obfuscate the assessment of *de novo* assemblies (Figure 4.4). As such, future studies should attempt to remove haplotypes before scaffolding. We have had preliminary success along these lines using the purge_dups pipeline[36] (Figure C.23).

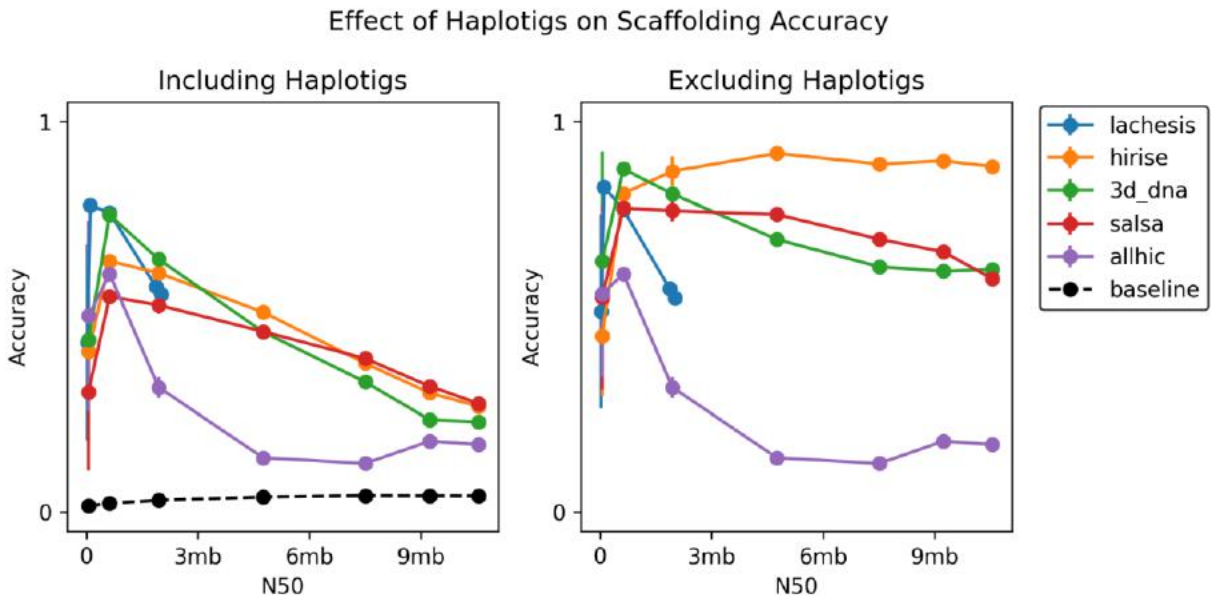

Figure 4.4: The improvement in accuracy by removing unscaffolded contigs of *H. sapiens*. We found that small overlapping contigs known as haplotigs are often excluded from scaffolds by the methods we tested. This suggests that the remaining scaffolded contigs show much higher accuracy, and is more consistent with results from the split reference setting.

4.4 Discussion

Given the proliferation of methods to scaffold genomes using Hi-C, it is critical to understand the landscape of their performance and assess the state of the field. Overall, we found that the performance of existing Hi-C scaffolding tools varies with species and assembly size, but with two salient trends across Hi-C scaffolders. First, accuracy decreases at low N50 values. Most scaffolders join contigs by pairing the two contigs with the highest number of connecting Hi-C reads. In a context where the contigs are both small and numerous, this approach leads to

ambiguities in edge strength and subsequent erroneous adjacencies. However, more surprising is the second trend, where scaffolders also perform worse in the high N50 regime. For example, when the yeast genome is broken into 1mb contigs, there are only four joins necessary to complete the genome. Yet only one assembler is successful at this task, whereas the others perform worse than the baseline of no scaffolding. Often it seems that over-grouping is the culprit, with AllHiC and 3d-dna prone to producing mega-scaffolds by grouping all of its contigs into a single scaffold. This suggests that single-contig scaffolds were not considered as a unique and important edge case for scaffolders. Recent advances in long read technologies have spurred the growth of near chromosome-length contigs in some species, and care should be taken to ensure that Hi-C scaffolders can also function as a polishing tool in this setting.

Overall, we found that HiRise offers the best performance on average across all conditions. (Table C.6) Though it was the slowest of all scaffolders, it was one of the only scaffolders not to experience any significant performance decay at large N50s. It should be noted that the original software release of HiRise contains several errors in the source code, and that subsequent development has been taken on by a private company (Dovetail Genomics). Lachesis was the second-best scaffolder on average, though it was the tool that most commonly failed to run under default settings. Its initial release also contains several installation bugs, and its later development has been taken on by another private company (Phase Genomics). AllHiC and Salsa yielded slightly worse performance than Lachesis. Interestingly, though 3d-dna had some of the lowest performing runs of the group under certain conditions, it remains one of the most heavily used methods. We traced its largest drop in performance to low grouping accuracy, which in turn was caused by a strong tendency to place most contigs into one or two scaffolds.

We offer several recommendations to consider when developing new Hi-C scaffolders. First, the most common starting point for Hi-C scaffolders is the BAM alignment file and assembly FASTA file. Since these files are both straightforward to parse and routinely produced, they offer excellent starting points for scaffolding. We found that deviations from this workflow, such as 3d-dna's requirement of the Juicer pipeline, created additional barriers to use. Second, we found the requirement for chromosome count to be unnecessary to achieve good performance. The two scaffolders which required this parameter, Lachesis and AllHiC, did not perform substantially better than other methods. Because Hi-C scaffolding is most often used in a *de novo* context, chromosome count is often unknown or unreliable, and therefore should be estimated by the method itself. Third, several scaffolders include integrated methods to break scaffolds at positions where a misassembly may have occurred. This step should be optional, because in some situations it erroneously breaks contigs that have been assembled with the aid of additional sequencing information such as mate pairs or optical mapping. Finally, scaffolders should output an AGP file to describe the organization of contigs[71]. Several scaffolders omit this information, and some create their own bespoke file formats which lead to problems comparing and understanding scaffolding outputs.

The current state of Hi-C scaffolding remains a two-step process: first, assemblies are passed through scaffolding software, and then the errors are fixed by hand. Opportunities for improvement lie on both ends of the workflow - better algorithms to scaffold genomes with Hi-C, and more modern tools for the manual correction of scaffolds. Despite any shortcomings of current methods, Hi-C scaffolding is a powerful tool in the evolving science of building reference genomes, and we hope that future developers can look to our study to help select an appropriate scaffolder for their own assembly tasks.

4.5 Availability of Data and Materials

All the scripts used in this study are available at https://github.com/Noble-Lab/hic_scaffolder_benchmarks under the MIT license. All the docker containers created containing the Hi-C scaffolders are also available under open licenses.

Chapter 5. A MACHINE LEARNING APPROACH TO SCAFFOLDING GENOMES WITH HI-C

5.1 INTRODUCTION

A vast amount of information is locked away in the genome of a species until it is first sequenced and assembled. With a high-quality reference genome, we can predict genes, transcripts, and proteins, and open the door to multi-omics experiments. Historically, efforts to produce reference sequences required large consortia and a careful delegation of work, but several key advances in sequencing technologies have enabled individual labs to publish chromosome-length sequences. Among these advances are improvements in scaffolding capabilities due to the development of genome-wide chromatin conformation capture (Hi-C).[72] Hi-C allows scaffolders to algorithmically organize contigs into chromosomes, but these methods fall short of human performance.[73] The current state of Hi-C scaffolding separates the task into two steps: first, a scaffolding algorithm provides a partially completed scaffold and second, mistakes are manually corrected by hand. Here, we attempt to leverage the growing number of genomes manually assembled with Hi-C to train a classifier to identify adjacent contigs and close the gap between computer and human performance.

The goal of scaffolding in genome assembly is to find a path of contigs to form chromosome length sequences. Contigs fall short of spanning chromosomes due to the presence of challenging genomic segments, such as repeat regions, allelic variation, and low-complexity sections.[74] Using proximity information, Hi-C scaffolders determine which contigs are adjacent

to each other. Typically, a scaffolder will start by creating a contig graph where each contig is represented by a node. Then, edges are created to connect nodes, with weights proportional by the number of Hi-C reads connecting two contigs. The underlying assumption of this approach is that distances in three-dimensions is highly correlated with distances in distances in one dimension. Accordingly, the most common approach is to find the maximum spanning tree of the resulting graph, where each node is connected with its highest edge weight partners. Finally, to produce a linear set of sequences equivalent to a set of chromosomes, scaffolders must identify a set of paths from this maximum spanning tree. Branches are impermissible in a scaffolding, because a branch would indicate that one contig has three adjacent neighbors rather than two, a property not allowed by the linear nature of the genome. Ultimately, this approach is inherently error prone since the relationship between the one-dimensional and three-dimensional genomics positions is complicated by the cell organizing its chromatin in particular patterns.

Hi-C scaffolders mostly differ in the heuristics they utilize to find paths in the tree. Lachesis[21], HiRise[61], and AllHiC[63] utilize a clustering step to partition the contig graph before moving on to subsequent steps. 3d-dna[37] and Salsa[62] attempt to resolve orientation of contigs by further splitting each contig into two halves before computing a maximum spanning tree. HiRise and AllHiC optimize an objective function to determine orientation, and AllHiC extends this approach to determine order as well. However, even with the supplement of heuristics, our prior investigation into benchmarking Hi-C scaffolders revealed that they fall short of human performance. Human curators identify and correct scaffolding errors by visually evaluating the patterns present in the Hi-C interaction matrix between two contigs rather than their edge weight.

Given that human curators do not use additional data for the scaffolding task beyond the Hi-C matrix, we reason that all the requisite information for scaffolding is present in the

interaction matrix. To mimic the manual approach of visually assessing regions of the Hi-C matrix, we opted to use an image-based machine learning approach to learn how to discriminate between adjacent and non-adjacent contigs. We used the DNA Zoo repository of manually curated Hi-C based genome assemblies as our training set.[14] By extracting millions of examples of adjacent and non-adjacent pairs of contigs from ten species, we trained a convolutional neural network to classify contig adjacencies. One of the key challenges in applying a machine learning approach was to feed the variable-sized Hi-C matrices into models that most often require fixed input sizes. We developed a novel approach to classify a given Hi-C matrix by jointly analyzing the four different corners of the matrix. We show that with this approach, we achieved high precision and recall on several of our validation species. Finally, we found that our proof-of-concept approach to generating scaffolds using the model produce comparable results to current Hi-C scaffolding algorithms. With further optimization of our model and path finding algorithm, we believe a machine learning approach could be a feasible method of exceeding current scaffolding capabilities.

5.2 METHODS

5.2.1 *Data Collection*

We relied on the DNA Zoo for high-quality, manually curated, and publicly available genomes that have associated Hi-C reads. For our approach to be broadly applicable to new species, we sought to gather training data diversely across the tree of life.[75] We mapped all available species across the DNA Zoo on the tree of life and selected 10 diverse species that covered as many distinct branches as possible (Figure 5.1). We chose two plants (the large white petunia *Petunia axillaris* and 'Hillquist' blackberry *Rubus argutus*), a fungus (the common mushroom *Agaricus bisporus*), a nematode (the brown stomach worm *Teladorsagia circumcincta*), an insect (the orange-legged furrow bee *Halictus rubicundus*), a fish (yellowfin tuna *Thunnus albacares*), a reptile (the Argentine black and white tegu *Salvator merianae*), a bird (the emu *Dromaius novaehollandiae*), and two mammals (the black-footed cat *Felis nigripes* and bearded seal *Erignathus barbatus*). We then downloaded for each species the input assembly FASTA, the Hi-C reads, and the assembly file which indicates how contigs were ultimately organized. After correcting misnamed sequences, we aligned the Hi-C reads against the assembly file and created cooler files from the resulting alignments[76]. These cooler files store the Hi-C contact matrix, offer fast programmatic access, and are compatible with the HiGlass viewer.[77]

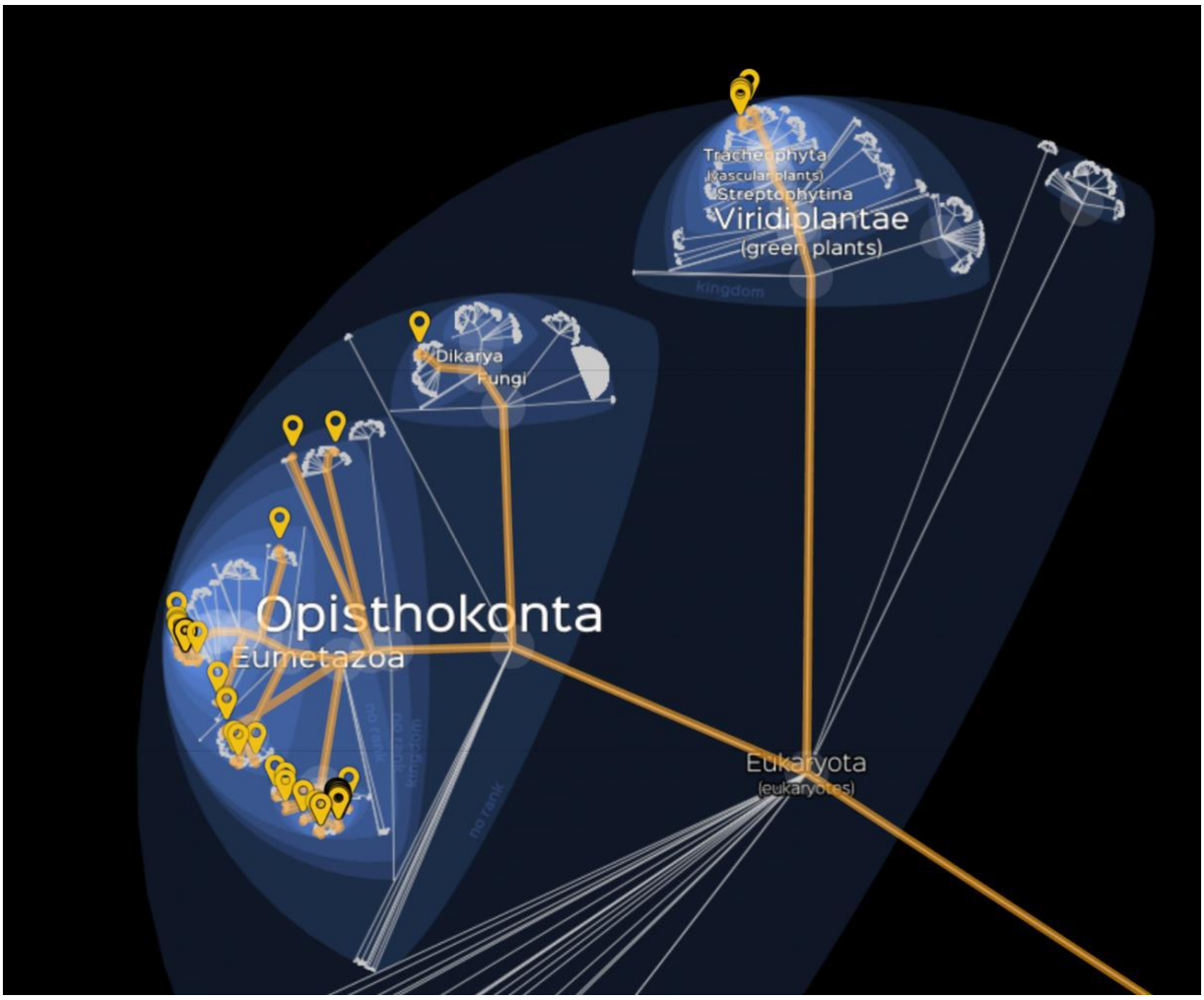

Figure 5.1: A mapping of all the species available in DNA Zoo on the eukaryotic section of the tree of life. The project is geared towards completed mammalian genomes so the vast majority of species are concentrated in a few branches. We attempted to pick our training species to as broadly represent the tree of life.

5.2.2 *Generating Training Data*

To generate positive and negative training data, we extracted information from the cooler files. Each example corresponds to a pair of contigs. Positive examples were extracted for any contigs that were adjacent to each other. In theory, the set of all interactions minus positive

examples could be negative examples; however, because that number is related to the square of the number of contigs, it is prohibitive to extract all negative examples. Instead, we focused on only extracting negative examples which were between highly connected contigs. For each contig, the top 50 most connected contigs which were not the positive connection were used as negative examples. To determine connectivity, we normalized by contig length by dividing the number of reads between contigs by the multiple of both contig lengths. We also limited contig sizes to ≥10kb to ensure that interaction matrices would at least be of a certain size. Starting with the 2kb resolution balanced Hi-C matrix, we extracted positive and negative examples by taking the 5x5 pixel corners of each matrix. For each interaction matrix, we first trimmed any rows and columns on the edges which contain no reads. To ensure that for larger contigs adequate signal was captured, we dynamically lower the resolution . For example, if the interaction matrix was 20x40 pixels, we would run a pooling operation to generate the 10x20 pixel version of that matrix before extracting the corners. The 4x5x5 arrays were stored along with their labels for use during training. Each array was normalized such that the highest value in the matrix was 1 and augmented with random gaussian noise with a standard deviation of 0.1 each time it was drawn from the set of examples.

5.2.3 *Model Architecture and Training*

Our model has two stages: a grouped convolution stage, then a connected layer (Figure 5.2). The very first layer contains four sets of convolutional kernels, one set for each of the four corners of the interaction matrix. After that, there is a joint convolution layer and several dense layers, before finally outputting a vector of length five with each position undergoing a softmax. The

first four positions in the output vector correspond to the four different orientations two contigs can be in, and the fifth position indicates they are not connected.

There is a deep class imbalance due to the presence of far more negative samples than positive ones. To ensure that there is sufficient training on positive examples, we created training batches with equal numbers of up-sampled positive and negative examples per species. Additionally, since each species has a different number of contigs, the total number of examples from each varied considerably. To avoid learning from one species more than another, we also up-sampled examples from underrepresented species such that an equal number of examples from each species were included in each batch.

To understand how well the model was generalizing its predictions, we used a ten-fold cross validation approach in which we withheld one species during each run. We recorded training and validation loss, accuracy, recall, and precision. During inference on the validation sets, we did not up-sample positive examples so that the inference would mimic the ratio of classes we might expect to see in the real world. Due to the class imbalance, a precision recall (PR) curve was more appropriate than a ROC curve. We measured the PR curve of two properties: the order and orientation success. The order PR curve shows the ability to correctly determine the adjacency of two contigs, regardless of their orientation. The orientation PR curve takes the average PR curve of each of the four orientations to generate an overall understanding of how well the model orients contigs.

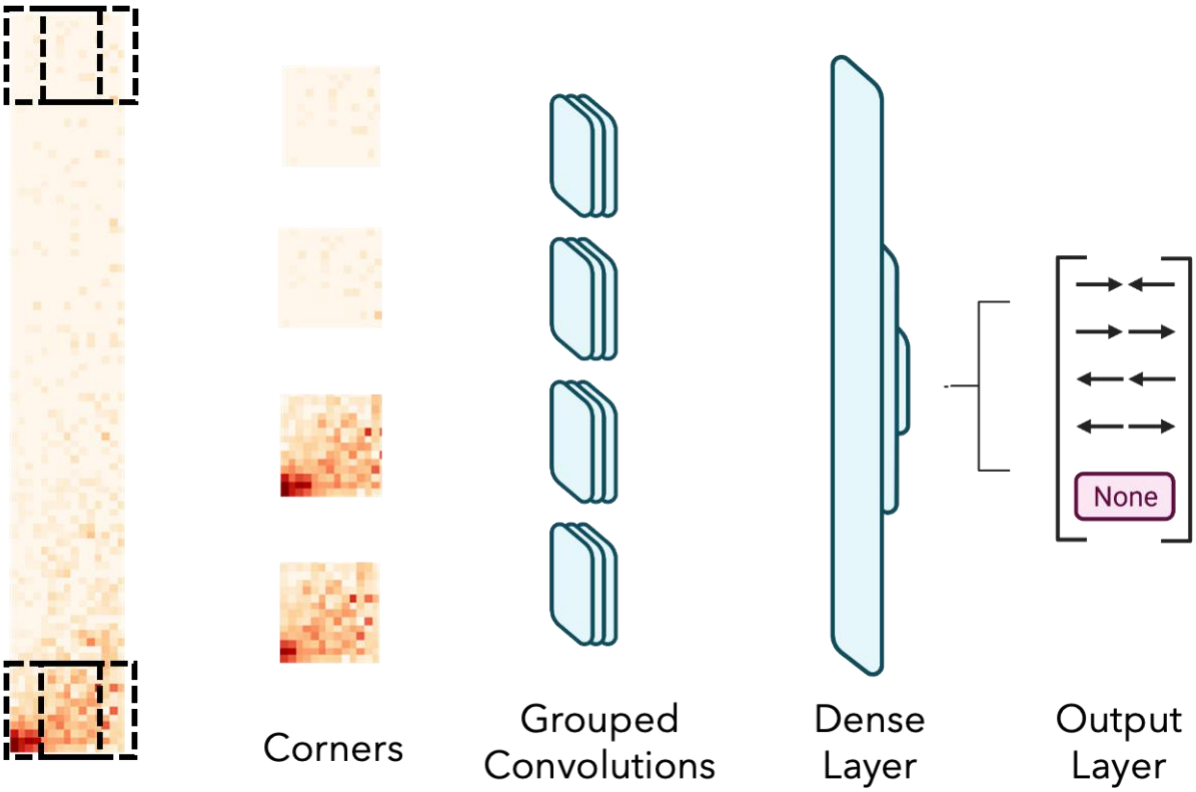

Figure 5.2: An overview of the architecture of our model. For any two contigs, there exists an interaction matrix between them. To maintain a fixed size input to our model, we focus on extracting the four corners of this image. They are then jointly fed into a grouped convolution layer such that each corner maintains a unique set of convolutional filters. After a series of dense layers, the output is a vector of length five, where the first four positions encode the orientation and the fifth position encodes the possibility of not being connected.

5.2.4 *Path Algorithm*

To generate a global scaffolding, we need to bridge the gap between classifying individual contig adjacencies and complete contig paths. On inference, the Hi-C interaction matrix is extracted in an identical fashion to the training dataset. For each contig, we gathered images from the top 50 most connected contigs as determined by read density. The model was then used to determine the probability of each of the edges in the top 50 nearest neighbor graph.

The first step in our path algorithm was to identify the maximum spanning tree in the probability graph. To resolve branches that might form in the spanning tree, we implemented a rudimentary tree cutting algorithm. For every vertex with a degree greater than 2, we only retain the top two edges. This guarantees that the resulting graph will only contain non-overlapping paths.

5.3 RESULTS

5.3.1 *Training Data*

We extracted images in the form of Hi-C interaction matrices from ten different species for our training and validation sets. For each positive example, we had approximately 50 negative examples involving the same contig, creating a class imbalance of about 98% negative. We found that more than 99% of the time, the top 50 connected contigs contained the true adjacency. Our initial training data extraction focused on pulling 10x10 pixel corners at a fixed 1kb resolution. However, by dynamically changing the resolution, trimming empty rows and columns, and extracting a smaller 5x5 pixel square from each corner, we improved the area under the precision recall curve four-fold. We also found that balancing the number of examples between classes and species greatly improved the generalizability between validation folds.

5.3.2 *Model Performance*

We trained our convolutional neural network on 2,608,964 examples over 5 epochs, and found the model converged in the first few epochs. Overall, we found that our model performed

well on half of the species we validated. Using the *F. nigripes* genome as validation, the model achieved ~70% precision and 80% recall when classifying contigs as adjacent or non-adjacent (Figure 5.3). During our cross validation, we found wide variation in performance between species. Performance was high on *A. bisporus*, *T. albacares*, *F. nigripes*, *E. barbatus*, and *D. novaehollandiae*, moderate on *R. argutus* and *S. merianae*, and low on *P. axillaris*, *T. circumcincta*, and *H. rubicundus* (Figure D.1 – Figure D.9). On a high-performing validation species such as *F. nigripes*, we found that the precision-recall curve exceeds that of even the training data set, suggesting the validation task was not as difficult as the training task (Figure 5.4).

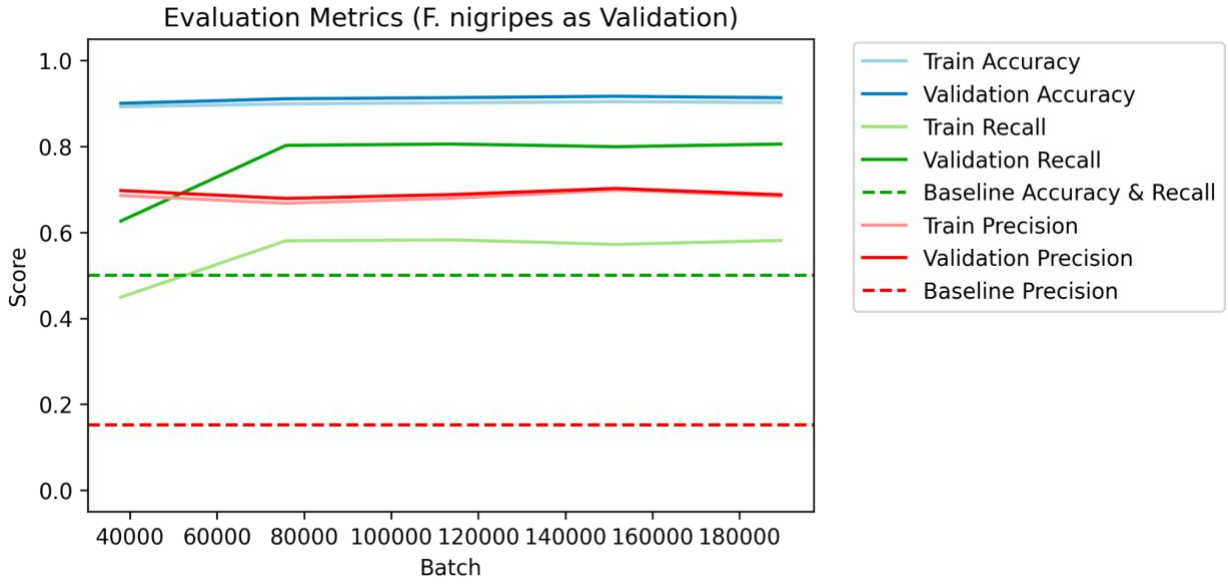

Figure 5.3: The evaluation metrics of the model during training. The dashed lines represent the baseline model of a random classifier. Since there is a large class imbalance, with a skew towards negative examples, precision and recall values are particularly revealing.

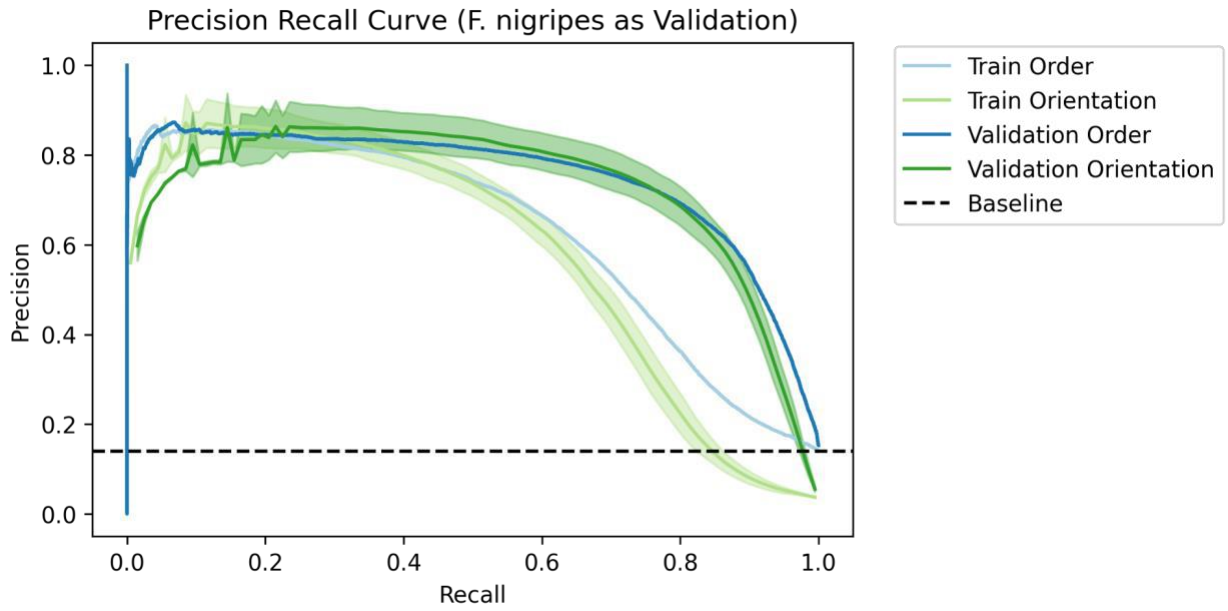

Figure 5.4: The precision recall curve when using *F. nigripes* as the validation set. The validation precision recall performance is better than even the training set, recapitulating the training loss observations. The model performs the best on this particular species compared to the other cross validation folds.

5.3.3 *Scaffolding Accuracy*

Using *D. novaehollandiae* as our validation species, we tested the scaffolding accuracy of our approach. We implemented our path finding approach to generated scaffolds and used Edison[70] to determine scaffolding accuracy (Figure 5.5). Scaffolds generated in this manner were 70% accurate when compared to the manually curated genome. The major contributor in loss of accuracy was the low grouping score of 6.3%, suggesting that scaffolds were short but highly accurate. We tested several other ,scaffolders and found that most had an accuracy of around 70% - 80%, indicating that the machine learning approach is a viable method (Table 5.1).

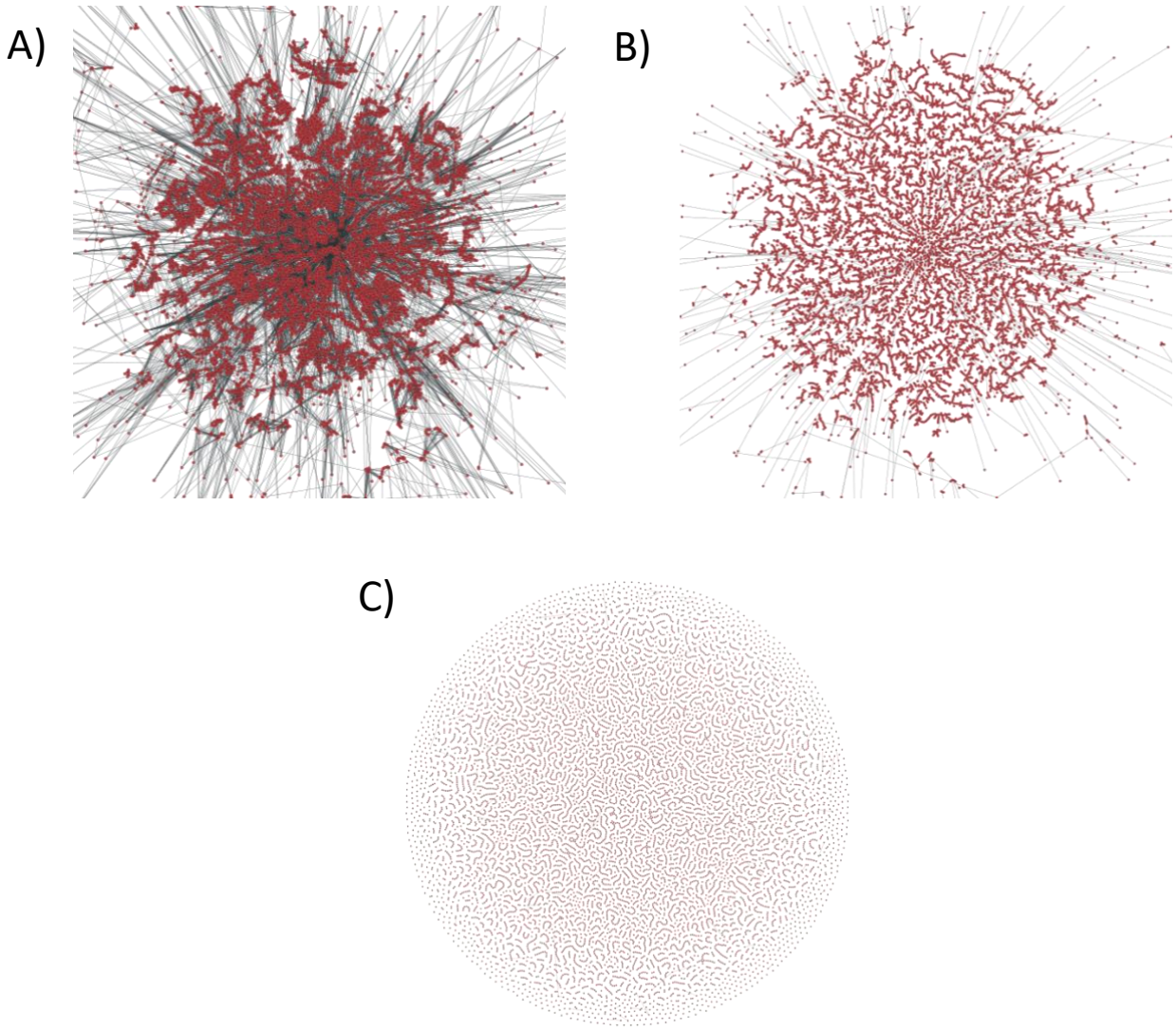

Figure 5.5: The progression of graphs during the path finding phase of the machine learning approach. A) The probability graph after each edge in the contig graph has been evaluated by the convolutional neural network and the probability of adjacency is determined. B) The maximum spanning tree of the contig probability graph. Branches are impermissible when producing scaffolds, so this tree needs to be reduced to a set of paths. C) The paths generated by the tree trimming algorithm, where each path represents an independent scaffold.

Method	Accuracy
CNN	70.05%
Lachesis	81.25%
Hirise	68.79%
3d-dna	78.99%
Salsa	1.11%
Allhic	77.08%

Table 5.1: The accuracy of various Hi-C scaffolding methods on *D. novaehollandiae*. Our machine learning based approach (CNN) does not exceed state of the art but does appear to produce comparable results using a simple and conservative path finding algorithm.

5.4 DISCUSSION

Hi-C based scaffolding offers an exciting new chapter in genome assembly, though current state-of-the-art methods all require extensive manual correction. We attempt to close this gap by applying a machine learning approach and learning patterns of Hi-C data between neighboring contigs. To the best of our knowledge, this is the first data-driven approach to Hi-C based scaffolding. In a two-step approach, we use our trained model to compute the probability of contig adjacencies, and then use a path finding algorithm to traverse the resulting graph and produce scaffolds.

For classifying adjacent contigs, our novel convolutional neural network approach shows promising results, though further refinement is necessary. On half of the species tested the model performed well, though in other cases only moderate or low performance was achieved. We are still exploring why this is the case, though we found that performance does not seem to be related to size of the contigs or genome. We also want to explore the information content of negative examples, and determine if down sampling those examples and subsequently increasing

the number of epochs would lead to better performance. One potentially limiting factor in our current approach is the use of dynamic Hi-C resolutions. For large contigs, a lower resolution is chosen such that more information can be summarized, and for smaller contigs, a higher resolution is used so that there are enough pixels to input into the model. It is possible that this change in Hi-C resolution could introduce size-based biases during classification. We are currently exploring methods to use a fixed resolution and minimize our preprocessing of the data, thereby moving towards a more end-to-end approach.

The last step of the scaffolding process in our framework is to convert edge weights, that correspond to the probability of being connected, into a set of paths which map to chromosomes. During our exploratory work into this step, we found that we achieve a level of accuracy comparable with existing Hi-C scaffolders. Considering that we used a simple and highly conservative approach to our path finding algorithm, we believe there is much room for improvement. The maximum spanning tree of the contig probability graph shows many components with long paths that have intervening short branches. Rather than cutting at all branch points, we could first identify the longest path between terminal nodes in a component, then cut all branches not on the path to yield longer scaffolds. Interestingly, Lachesis currently uses a similar approach, and it would be quite interesting to see if it works better with a probability graph than a read count graph.

With the excitement around sequencing initiatives such as the T2T human genome project and vertebrate genome project, Hi-C based scaffolding has come to the forefront of genome assembly. We have shown that a machine learning based approach holds promise, and with further optimization could close the gap between computer and human performance.

Additionally, a machine learning paradigm would allow us to continuously improve future scaffolding efforts by retraining on future releases of manually curated genomes.

Chapter 6. CONCLUSION

The first Hi-C based scaffolder was introduced almost a decade ago. At the time, the genome-wide version of chromatin conformation capture had only recently been described. Conducting the Hi-C experiment was challenging as it required specialized reagents, lacked a streamlined kit, and even the underlying procedure was in relative flux with various competing versions of the protocol. News of a novel method to scaffold genomes was only just percolating into the genome assembly community, and only a few ambitious groups ventured to attempt the Hi-C experiment and scaffolding process.

Since then, the field has leapt forward and matured, with the genesis of several initiatives and the widespread adoption of Hi-C. The 4DN initiative is a dedicated NIH initiative to study the three-dimensional organization of chromatin. It has spurred greater collaboration in the community, as well as a push toward standardization of data and methods. The DNA Zoo is an initiative to conduct Hi-C and scaffold mostly mammalian genomes and has greatly popularized Hi-C as a scaffolding method. Even initiatives initially unrelated to Hi-C have incorporated it as a key element in their work. Notably, the telomere-to-telomere initiative for the human genome utilized Hi-C to validate their sequence of challenging regions. The vertebrate genome project has incorporated Hi-C as a necessary step to scaffold genomes.

Perhaps most telling, is the development of a new genome assembly “recipe” by the T2T initiative. In their approach, sequencing efforts are directed towards generating PacBio HiFi

reads, which provide high quality long reads, Oxford Nanopore reads, which provide ultra-long sequencing data, and Hi-C reads, which allow the completion of chromosomes and validation of contig accuracy. Our work with assembling the genomes of various species suggests that long reads are critical to the assembly process. Previous efforts to assemble *E. gracilis* relied on short read data and ultimately did not yield a workable result. In addition, we found that with the longer contigs generated by utilizing long read data, the process of manually correcting the scaffolds was greatly simplified. The longer contigs yielded greater Hi-C signal, making the patterns of correct and incorrectly scaffolded contigs far more obvious than in more fragmented assemblies.

Our efforts in understanding the state of the field for algorithmic based Hi-C scaffolded yielded a surprising irony - that the rise of third generation sequencing was at odds with many Hi-C scaffolders. We found that several scaffolders struggled on the high-quality assemblies we might expect to see from groups utilizing the long read technologies we suggest adopting. It seems that many of these scaffolders were geared toward scaffolding assemblies with thousands of pieces rather than a few dozen. This gap between the state of genome assembly and the state of genome scaffolding reveals a key opportunity for researchers to address.

One potential avenue of addressing differences in human performance on Hi-C scaffolding and algorithmic performance is to leverage the growing amount of publicly available Hi-C data. We utilized the repository of assemblies and Hi-C data available through the DNA zoo initiative in a machine learning approach. To our knowledge, there are no supervised learning methods that have attempted to address the Hi-C scaffolding problem. Our novel approach performs well on some species, though we were not able to achieve more general

performance. The landscape of possibilities for data processing, models, and training regimens is large, so we cannot discount the possibility of a better approach.

In conclusion, Hi-C scaffolding has become a critical component of modern genome assembly. In conjunction with the long-read data generated by third generation sequencing technologies, Hi-C has enabled the completion of giga-base sized genomes. We expect that its use will continue to grow in the coming years, giving urgent need for high performance tools in the ecosystem. With this body of work, we hope we can contribute to the field and enable others to navigate the many paths and methods available in the space.

BIBLIOGRAPHY

1. Giani, A. M., Gallo, G. R., Gianfranceschi, L. & Formenti, G. Long walk to genomics: History and current approaches to genome sequencing and assembly. *Computational and Structural Biotechnology Journal* **18**, 9–19 (2020).
2. Abdellah, Z. *et al.* Finishing the euchromatic sequence of the human genome. *Nature 2004 431:7011* **431**, 931–945 (2004).
3. Bennett, S. T., Barnes, C., Cox, A., Davies, L. & Brown, C. Toward the $1000 human genome. *http://dx.doi.org/10.1517/14622416.6.4.373* **6**, 373–382 (2005).
4. English, A. C. *et al.* Mind the Gap: Upgrading Genomes with Pacific Biosciences RS Long-Read Sequencing Technology. *PLOS ONE* **7**, e47768 (2012).
5. Lu, H., Giordano, F. & Ning, Z. Oxford Nanopore MinION Sequencing and Genome Assembly. *Genomics, Proteomics & Bioinformatics* **14**, 265–279 (2016).
6. Hon, T. *et al.* Highly accurate long-read HiFi sequencing data for five complex genomes. *Scientific Data 2020 7:1* **7**, 1–11 (2020).
7. Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nature Biotechnology 2018 36:4* **36**, 338–345 (2018).
8. Pop, M., Salzberg, S. L. & Shumway, M. Genome sequence assembly: Algorithms and issues. *Computer (Long Beach Calif)* **35**, 47–54 (2002).
9. Salzberg, S. L. *et al.* GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Research* **22**, 557–567 (2012).
10. Rhie, A. *et al.* Towards complete and error-free genome assemblies of all vertebrate species. *Nature 2021 592:7856* **592**, 737–746 (2021).

11. Luo, J. *et al.* A comprehensive review of scaffolding methods in genome assembly. *Briefings in Bioinformatics* **22**, (2021).

12. Yuan, Y., Chung, C. Y. L. & Chan, T. F. Advances in optical mapping for genomic research. *Comput Struct Biotechnol J* **18**, 2051–2062 (2020).

13. Yeo, S., Coombe, L., Warren, R. L., Chu, J. & Birol, I. ARCS: scaffolding genome drafts with linked reads. *Bioinformatics* **34**, 725–731 (2018).

14. Dudchenko, O. *et al.* The Juicebox Assembly Tools module facilitates de novo assembly of mammalian genomes with chromosome-length scaffolds for under $1000. *bioRxiv* 254797 (2018) doi:10.1101/254797.

15. Nurk, S. *et al.* The complete sequence of a human genome. *Science (1979)* **376**, 44–53 (2022).

16. Rhie, A. *et al.* Towards complete and error-free genome assemblies of all vertebrate species. *Nature 2021 592:7856* **592**, 737–746 (2021).

17. Cheng, S. *et al.* 10KP: A phylodiverse genome sequencing plan. *Gigascience* **7**, 1–9 (2018).

18. Raae, M. *et al.* Verkko: telomere-to-telomere assembly of diploid chromosomes. *bioRxiv* 2022.06.24.497523 (2022) doi:10.1101/2022.06.24.497523.

19. Flemming, W. & Flemming, W. *Zellsubstanz, Kern und Zelltheilung*. *Zellsubstanz, Kern und Zelltheilung /* (F.C.W. Vogel, 1882). doi:10.5962/bhl.title.168645.

20. Lieberman-Aiden, E. *et al.* Comprehensive mapping of long-range interactions reveals folding principles of the human genome. *Science* **326**, 289–293 (2009).

21. Burton, J. N. *et al.* Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions. *Nat Biotechnol* **31**, 1119 (2013).

22. Naghavi, M. *et al.* Global, regional, and national age-sex specific all-cause and cause-specific mortality for 240 causes of death, 1990-2013: A systematic analysis for the Global Burden of Disease Study 2013. *The Lancet* **385**, 117–171 (2015).

23. Ivens, A. C. *et al.* The Genome of the Kinetoplastid Parasite, Leishmania major. *Science* **309**, 436 (2005).

24. Camacho, E. *et al.* Complete assembly of the Leishmania donovani (HU3 strain) genome and transcriptome annotation. *Scientific Reports 2019 9:1* **9**, 1–15 (2019).

25. Lypaczewski, P. *et al.* A complete Leishmania donovani reference genome identifies novel genetic variations associated with virulence. *Scientific Reports 2018 8:1* **8**, 1–14 (2018).

26. Müller, L. S. M. *et al.* Genome organization and DNA accessibility control antigenic variation in trypanosomes. *Nature 2018 563:7729* **563**, 121–125 (2018).

27. Hotez, P. J. *et al.* The Global Burden of Disease Study 2010: Interpretation and Implications for the Neglected Tropical Diseases. *PLOS Neglected Tropical Diseases* **8**, e2865 (2014).

28. Runckel, C., DeRisi, J. & Flenniken, M. L. A Draft Genome of the Honey Bee Trypanosomatid Parasite Crithidia mellificae. *PLOS ONE* **9**, e95057 (2014).

29. Shapiro, T. A. Kinetoplast DNA maxicircles: networks within networks. *Proc Natl Acad Sci U S A* **90**, 7809–7813 (1993).

30. van Luenen, H. G. A. M. *et al.* Glucosylated hydroxymethyluracil, DNA base J, prevents transcriptional readthrough in Leishmania. *Cell* **150**, 909–921 (2012).

31. Ebenezer, T. E. *et al.* Transcriptome, proteome and draft genome of Euglena gracilis. *BMC Biology 2019 17:1* **17**, 1–23 (2019).

32. Ebenezer, T. G. E., Carrington, M., Lebert, M., Kelly, S. & Field, M. C. Euglena gracilis Genome and Transcriptome: Organelles, Nuclear Genome Assembly Strategies and Initial Features. *Adv Exp Med Biol* **979**, 125–140 (2017).

33. Ranallo-Benavidez, T. R., Jaron, K. S. & Schatz, M. C. GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes. *Nature Communications 2020 11:1* **11**, 1–10 (2020).

34. Wang, J. R., Holt, J., McMillan, L. & Jones, C. D. FMLRC: Hybrid long read error correction using an FM-index. *BMC Bioinformatics* **19**, 1–11 (2018).

35. Koren, S. *et al.* Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Research* **27**, gr.215087.116 (2017).

36. Guan, D. *et al.* Identifying and removing haplotypic duplication in primary genome assemblies. *Bioinformatics* **36**, 2896–2898 (2020).

37. Dudchenko, O. *et al.* De novo assembly of the Aedes aegypti genome using Hi-C yields chromosome-length scaffolds. *Science (1979)* **356**, 92–95 (2017).

38. Zimin, A. v. *et al.* The MaSuRCA genome assembler. *Bioinformatics* **29**, 2669–2677 (2013).

39. Coombe, L. *et al.* LongStitch: high-quality genome assembly correction and scaffolding using long reads. *BMC Bioinformatics* **22**, 1–13 (2021).

40. Zhu, B. H. *et al.* P_RNA_scaffolder: A fast and accurate genome scaffolder using paired-end RNA-sequencing reads. *BMC Genomics* **19**, 1–13 (2018).

41. Manni, M., Berkeley, M. R., Seppey, M., Simão, F. A. & Zdobnov, E. M. BUSCO Update: Novel and Streamlined Workflows along with Broader and Deeper Phylogenetic

Coverage for Scoring of Eukaryotic, Prokaryotic, and Viral Genomes. *Molecular Biology and Evolution* **38**, 4647–4654 (2021).

42. Duan, Z. *et al.* A three-dimensional model of the yeast genome. *Nature* **465**, 363–367 (2010).
43. Kronenberg, Z. N. *et al.* Extended haplotype-phasing of long-read de novo genome assemblies using Hi-C. *Nature Communications 2021 12:1* **12**, 1–10 (2021).
44. Hoencamp, C. *et al.* 3D genomics across the tree of life reveals condensin II as a determinant of architecture type. *Science* **372**, (2021).
45. Beier, S. *et al.* Construction of a map-based reference genome sequence for barley, Hordeum vulgare L. *Scientific Data 2017 4:1* **4**, 1–24 (2017).
46. Zhang, L. *et al.* The Tartary Buckwheat Genome Provides Insights into Rutin Biosynthesis and Abiotic Stress Tolerance. *Mol Plant* **10**, 1224–1237 (2017).
47. Zhang, L. *et al.* Improved Brassica rapa reference genome by single-molecule sequencing and chromosome conformation capture technologies. *Horticulture Research 2018 5:1* **5**, 1–11 (2018).
48. Roach, J. C., Boysen, C., Wang, K. & Hood, L. Pairwise end sequencing: a unified approach to genomic mapping and sequencing. *Genomics* **26**, 345–353 (1995).
49. Berger, B., Waterman, M. S. & Yu, Y. W. Levenshtein Distance, Sequence Comparison and Biological Database Search. *IEEE Transactions on Information Theory* **67**, 3287–3294 (2021).
50. Hannenhalli, S. & Pevzner, P. A. Transforming men into mice (polynomial algorithm for genomic distance problem). *Annual Symposium on Foundations of Computer Science - Proceedings* 581–592 (1995) doi:10.1109/SFCS.1995.492588.

51. Tesler, G. Efficient algorithms for multichromosomal genome rearrangements. *Journal of Computer and System Sciences* **65**, 587–609 (2002).

52. Ozery-Flato, M. & Shamir, R. Two notes on genome rearrangement. *J Bioinform Comput Biol* **1**, 71–94 (2003).

53. Yancopoulos, S., Attie, O. & Friedberg, R. Efficient sorting of genomic permutations by translocation, inversion and block interchange. *Bioinformatics* **21**, 3340–3346 (2005).

54. Bergeron, A., Mixtacki, J. & Stoye, J. A unifying view of genome rearrangements. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **4175 LNBI**, 163–173 (2006).

55. Darling, A. E., Mau, B. & Perna, N. T. progressiveMauve: Multiple Genome Alignment with Gene Gain, Loss and Rearrangement. *PLoS ONE* **5**, (2010).

56. Alonge, M. *et al.* RaGOO: Fast and accurate reference-guided scaffolding of draft genomes. *Genome Biology* **20**, 1–17 (2019).

57. Marçais, G. *et al.* MUMmer4: A fast and versatile genome alignment system. *PLOS Computational Biology* **14**, e1005944 (2018).

58. Coombe, L. *et al.* ARKS: Chromosome-scale scaffolding of human genome drafts with linked read kmers. *BMC Bioinformatics* **19**, 1–10 (2018).

59. Kaplan, N. & Dekker, J. High-throughput genome scaffolding from in vivo DNA interaction frequency. *Nature Biotechnology 2013 31:12* **31**, 1143–1147 (2013).

60. Marie-Nelly, H. *et al.* High-quality genome (re)assembly using chromosomal contact data. *Nature Communications 2014 5:1* **5**, 1–10 (2014).

61. Putnam, N. H. *et al.* Chromosome-scale shotgun assembly using an in vitro method for long-range linkage. *Genome Research* **26**, 342 (2016).

62. Ghurye, J. *et al.* Integrating Hi-C links with assembly graphs for chromosome-scale assembly. *PLOS Computational Biology* **15**, e1007273 (2019).

63. Zhang, X., Zhang, S., Zhao, Q., Ming, R. & Tang, H. Assembly of allele-aware, chromosomal-scale autopolyploid genomes based on Hi-C data. *Nature Plants 2019 5:8* **5**, 833–845 (2019).

64. Renschler, G. *et al.* Hi-C guided assemblies reveal conserved regulatory topologies on X and autosomes despite extensive genome shuffling. *Genes Dev* **33**, 1591–1612 (2019).

65. Nakabayashi, R. & Morishita, S. HiC-Hiker: a probabilistic model to determine contig orientation in chromosome-length scaffolds with Hi-C. *Bioinformatics* **36**, 3966–3974 (2020).

66. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arxiv* (2013) doi:10.48550/arxiv.1303.3997.

67. Faust, G. G. & Hall, I. M. SAMBLASTER: fast duplicate marking and structural variant read extraction. *Bioinformatics* **30**, 2503–2505 (2014).

68. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *Gigascience* **10**, (2021).

69. Durand, N. C. *et al.* Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. *Cell Syst* **3**, 95–98 (2016).

70. Sur, A., Noble, W. S., Sullivan, S. & Myler, P. Edison: measuring scaffolding accuracy with edit distance. *bioRxiv* 2022.03.25.484952 (2022) doi:10.1101/2022.03.25.484952.

71. NCBI. AGP Specification v2.1. *National Center for Biotechnology Information* https://www.ncbi.nlm.nih.gov/assembly/agp/AGP_Specification/ (2019).

72. Rice, E. S. & Green, R. E. New Approaches for Genome Assembly and Scaffolding. (2018) doi:10.1146/annurev-animal-020518.

73. Sur, A., Noble, W. S. & Myler, P. J. A benchmark of Hi-C scaffolders using reference genomes and de novo assemblies. *bioRxiv* 2022.04.20.488415 (2022) doi:10.1101/2022.04.20.488415.

74. Pop, M., Salzberg, S. L. & Shumway, M. Genome sequence assembly: Algorithms and issues. *Computer (Long Beach Calif)* **35**, 47–54 (2002).

75. de Vienne, D. M. Lifemap: Exploring the Entire Tree of Life. *PLOS Biology* **14**, e2001624 (2016).

76. Abdennur, N. & Mirny, L. A. Cooler: scalable storage for Hi-C data and other genomically labeled arrays. *Bioinformatics* **36**, 311–316 (2020).

77. Kerpedjiev, P. *et al.* HiGlass: Web-based visual exploration and analysis of genome interaction maps. *Genome Biology* **19**, 1–12 (2018).

78. Pop, M., Kosack, D. S. & Salzberg, S. L. Hierarchical scaffolding with Bambus. *Genome Res* **14**, 149–159 (2004).

79. Nagarajan, N., Read, T. D. & Pop, M. Scaffolding and validation of bacterial genome assemblies using optical restriction maps. *Bioinformatics* **24**, 1229 (2008).

80. Phillippy, A. M., Schatz, M. C. & Pop, M. Genome assembly forensics: Finding the elusive mis-assembly. *Genome Biology* **9**, 1–13 (2008).

81. Darling, A. E., Tritt, A., Eisen, J. A. & Facciotti, M. T. Mauve Assembly Metrics. *Bioinformatics* **27**, 2756–2757 (2011).

82. Boetzer, M., Henkel, C. v., Jansen, H. J., Butler, D. & Pirovano, W. Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics* **27**, 578–579 (2011).

83. Magoc, T. *et al.* GAGE-B: an evaluation of genome assemblers for bacterial organisms. *Bioinformatics* **29**, 1718–1725 (2013).

84. Hunt, M., Newbold, C., Berriman, M. & Otto, T. D. A comprehensive evaluation of assembly scaffolding tools. *Genome Biology* **15**, 1–15 (2014).

85. Li, M., Tang, L., Wu, F. X., Pan, Y. & Wang, J. SCOP: a novel scaffolding algorithm based on contig classification and optimization. *Bioinformatics* **35**, 1142–1150 (2019).

86. Qin, M. *et al.* LRScaf: Improving draft genomes using long noisy reads. *BMC Genomics* **20**, 1–12 (2019).

87. Luo, J. *et al.* SLR: a scaffolding algorithm based on long reads and contig classification. *BMC Bioinformatics* **20**, 539 (2019).

88. Zhao, Z. *et al.* LDscaff: LD-based scaffolding of de novo genome assemblies. *BMC Bioinformatics* **21**, 1–15 (2020).

APPENDIX A

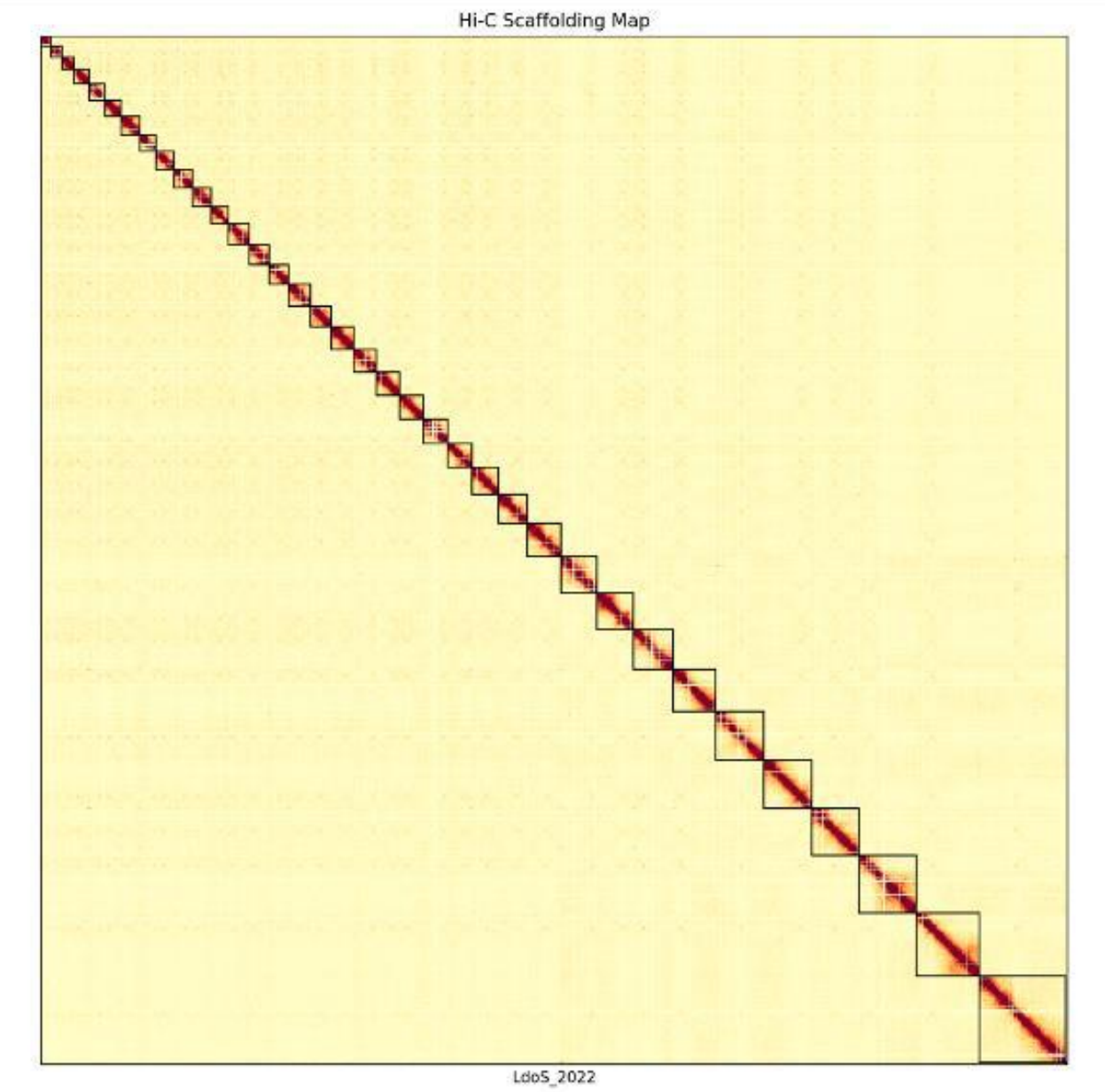

Figure A.1: The Hi-C interaction map of the scaffolded and completed genome for *L. donovani*. The Hi-C map did not show a significant presence of TADs or compartments, indicating that the species lack the more advanced chromatin organization typically seen in higher eukaryotes. However, it does show a classic centromeric interaction, allowing us to annotate these positions for the first time.

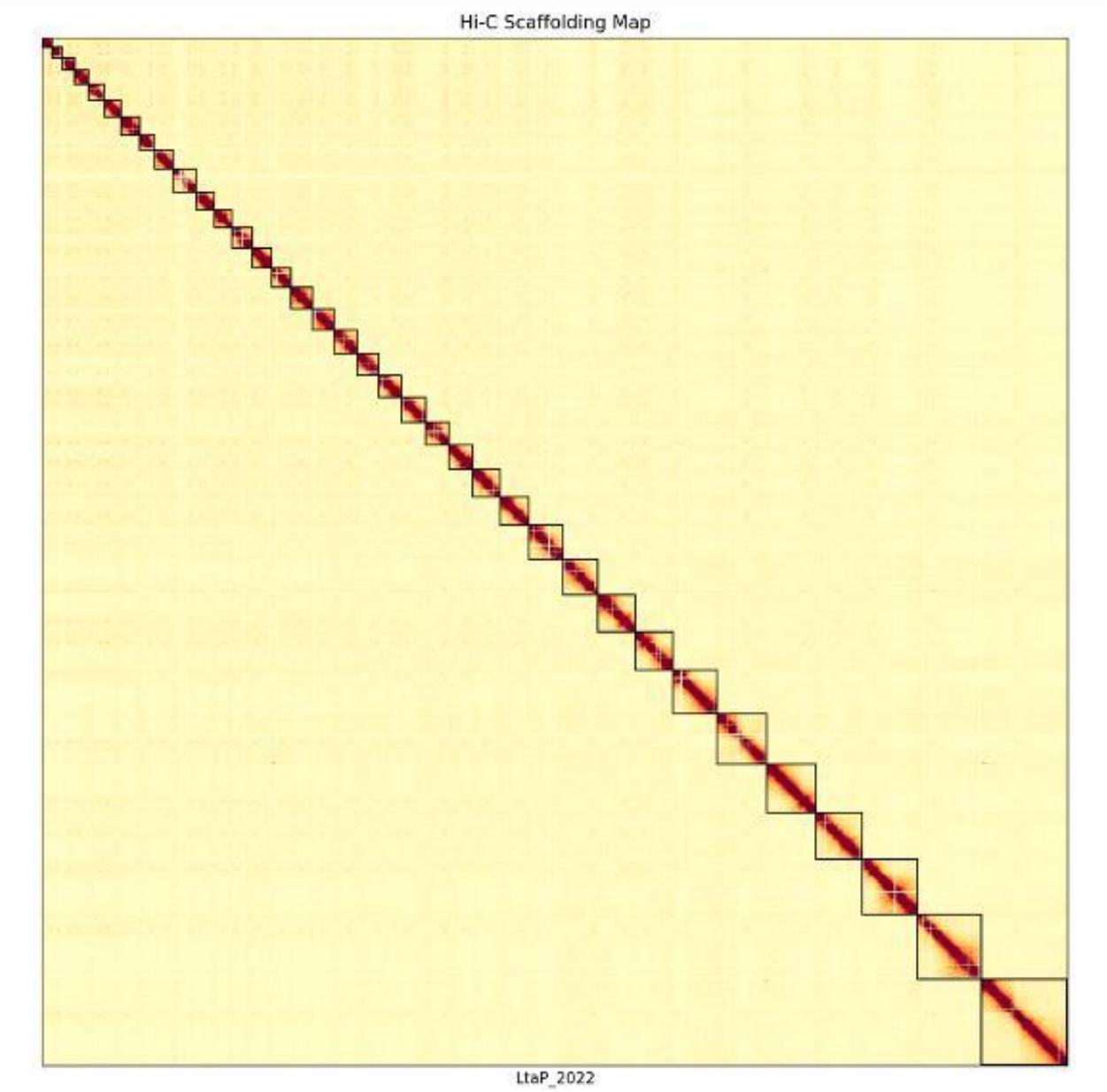

Figure A.2: The Hi-C interaction map of the scaffolded and completed genome for *L. tarentolae*.

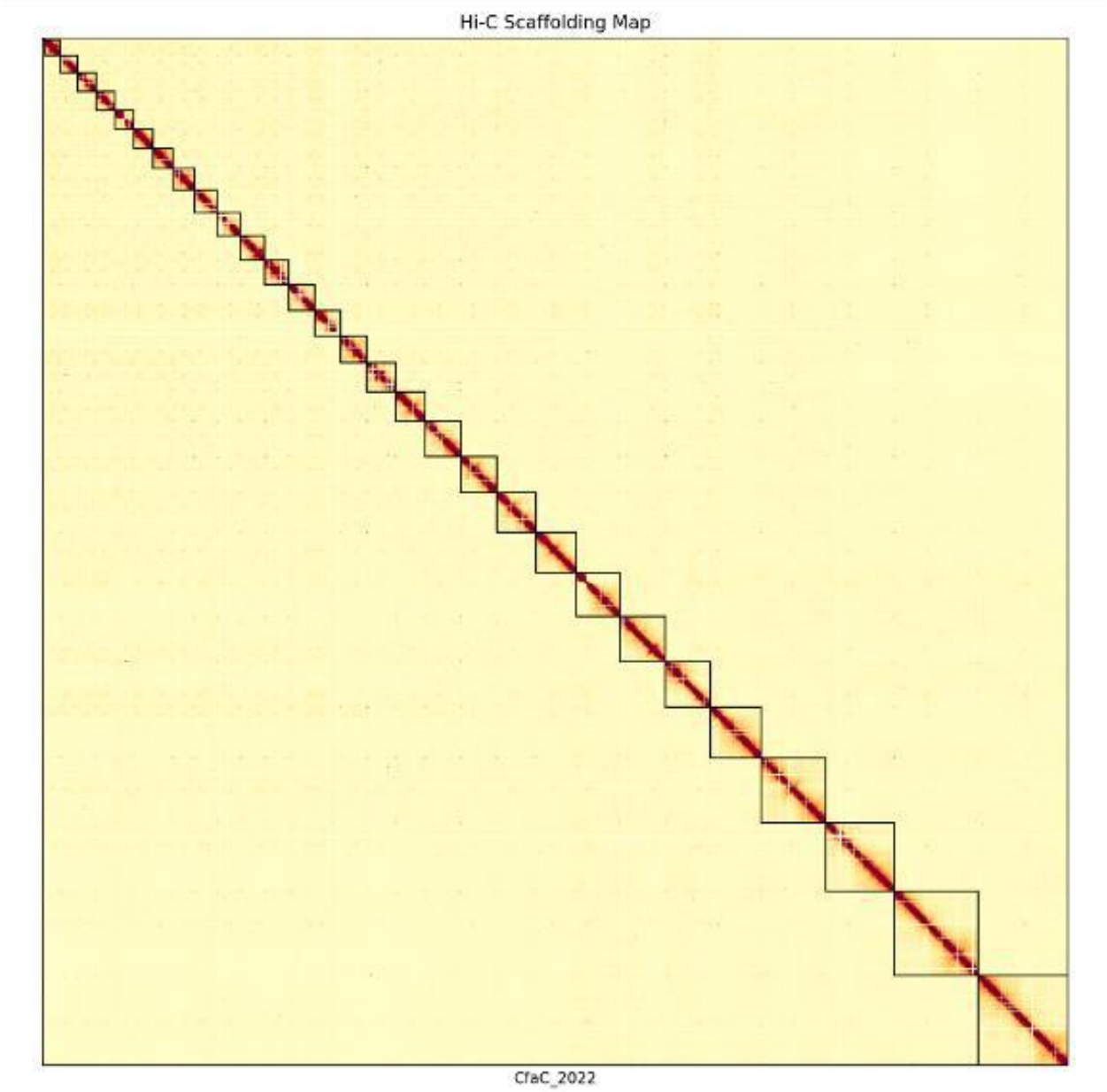

Figure A.3: The Hi-C interaction map of the scaffolded and completed genome for *C. fasciculata*.

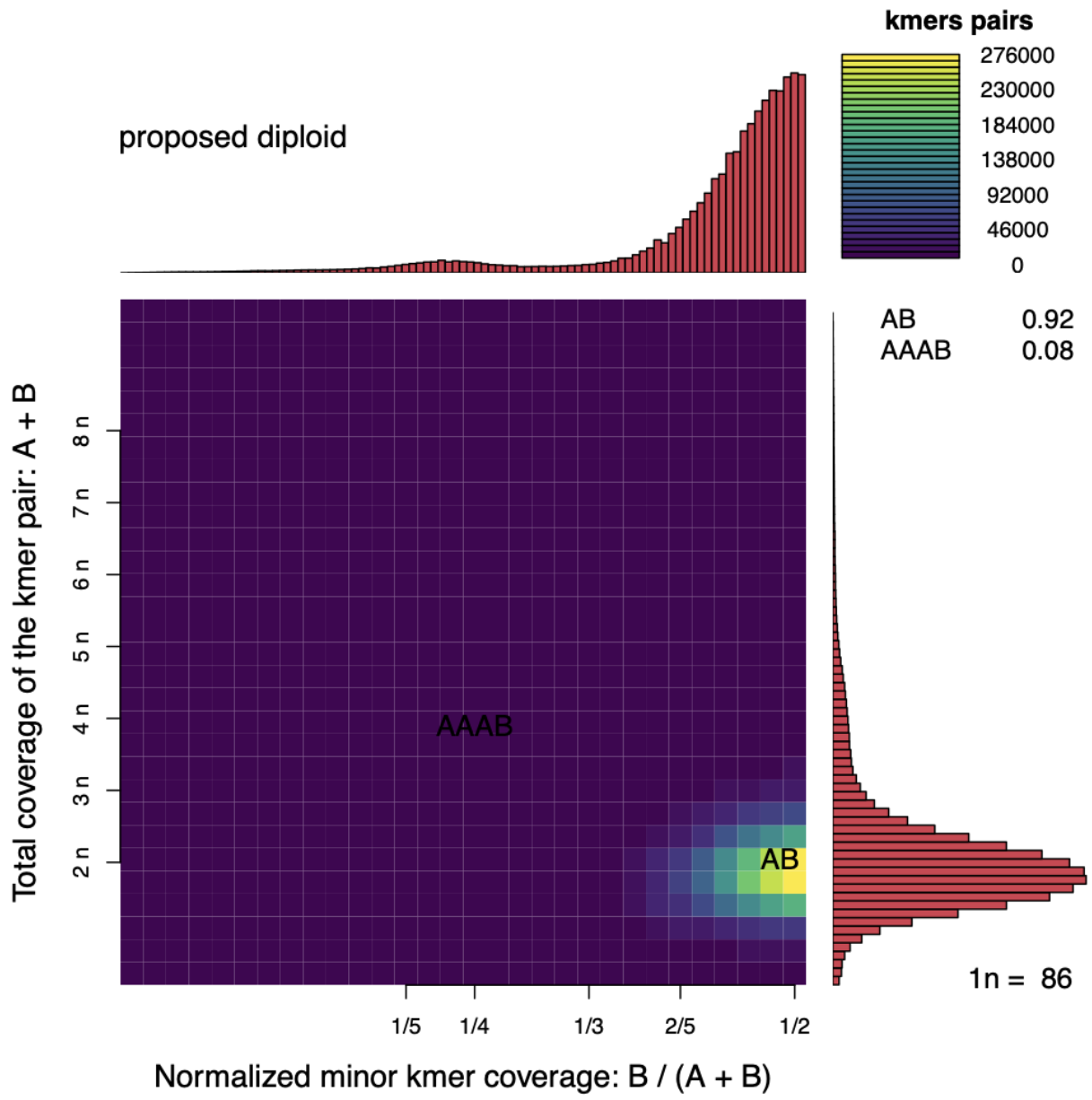

Figure A.4: The smudgeplot of *C. fasciculata* showing a strong indication of a diploid genome.

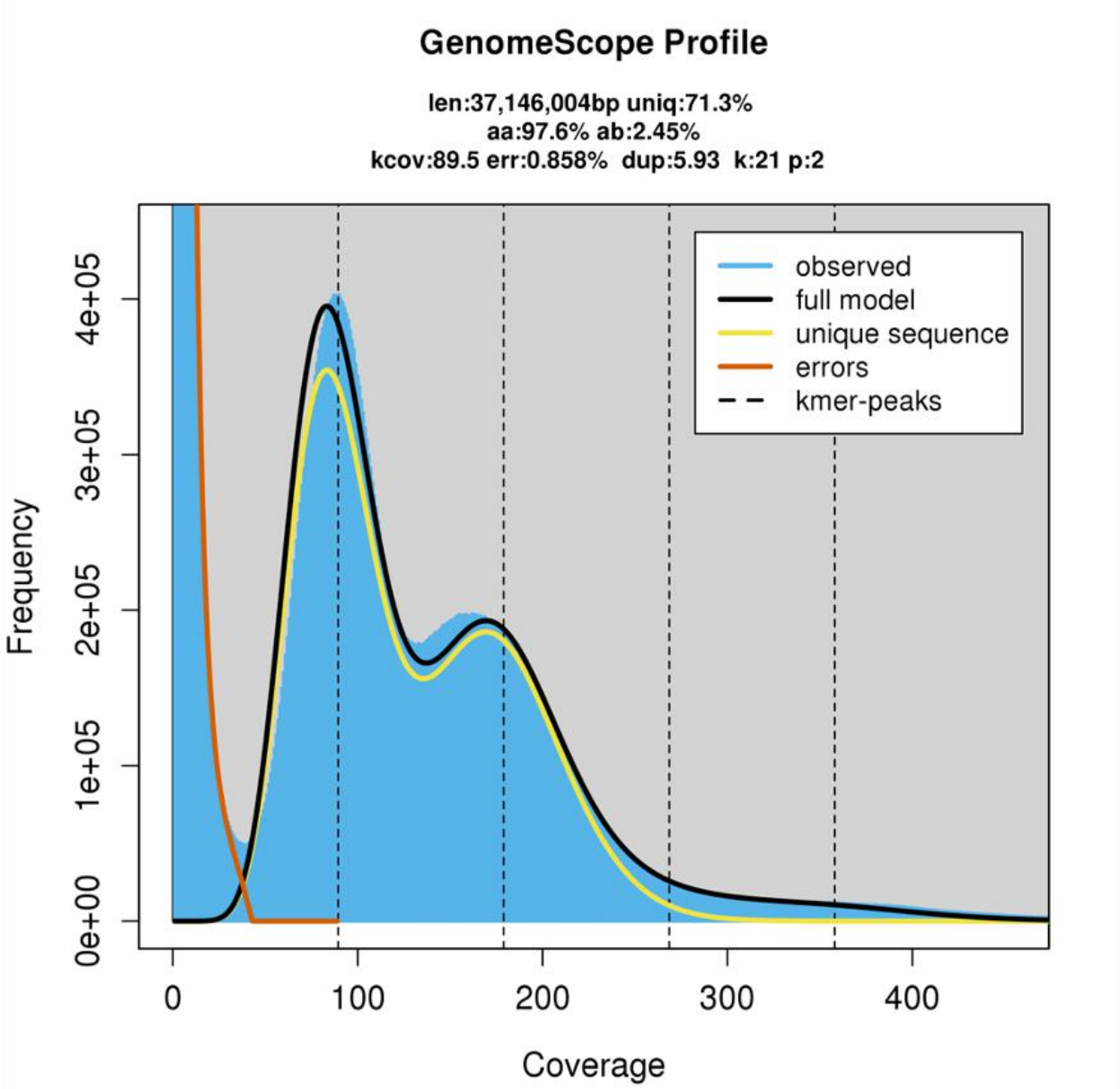

Figure A.5: The k-mer distribution from *C. fasciculata* short read libraries.

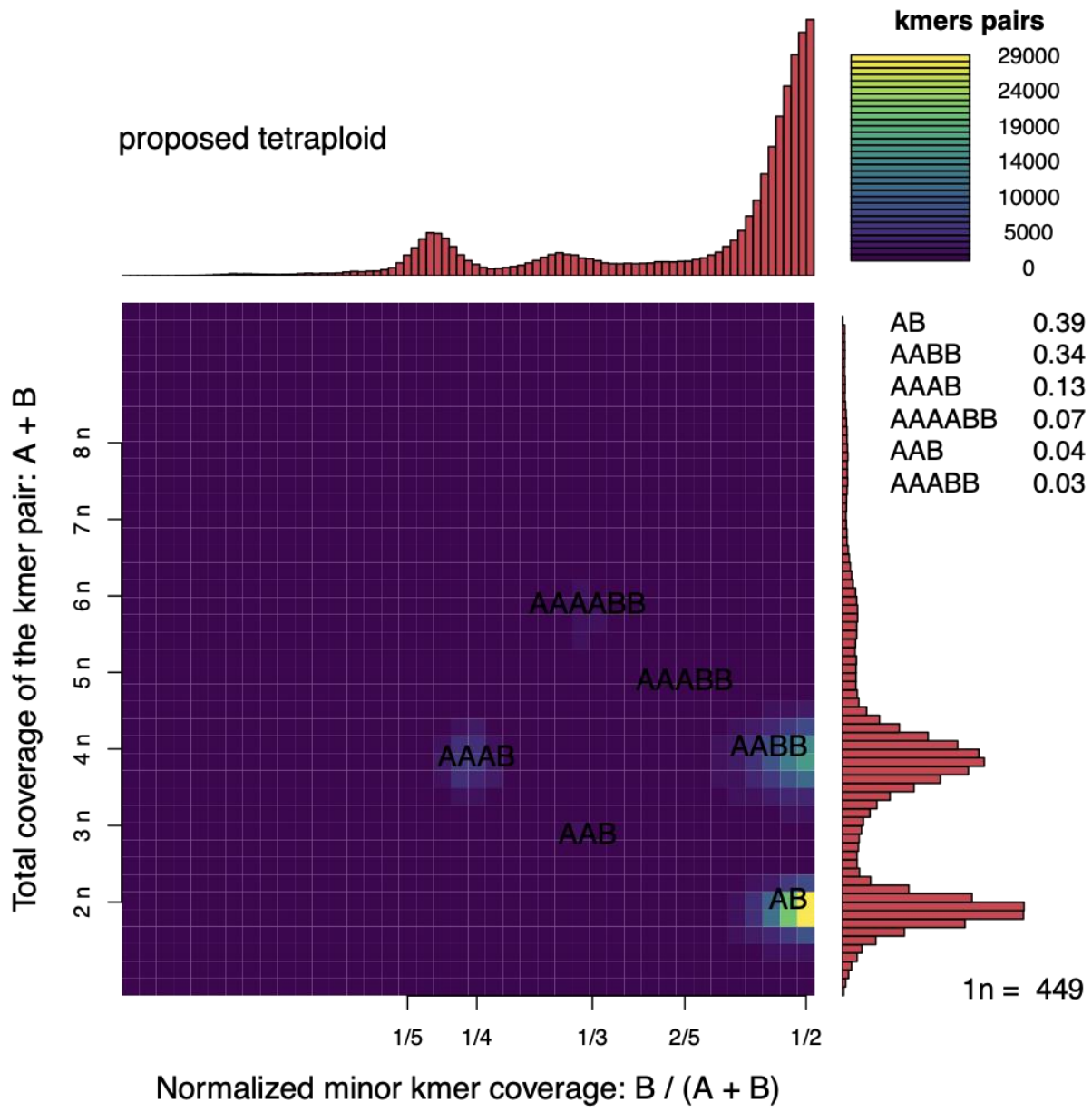

Figure A.6: The smudgeplot of *L. donovani*. Though the plot indicates the possibility of tetraploidy, the authors state that under conditions of low allelic variation, there can be a conflation of ploidies. We believe that *L. donovani* is diploid with species with aneuploidies for individual chromosomes.

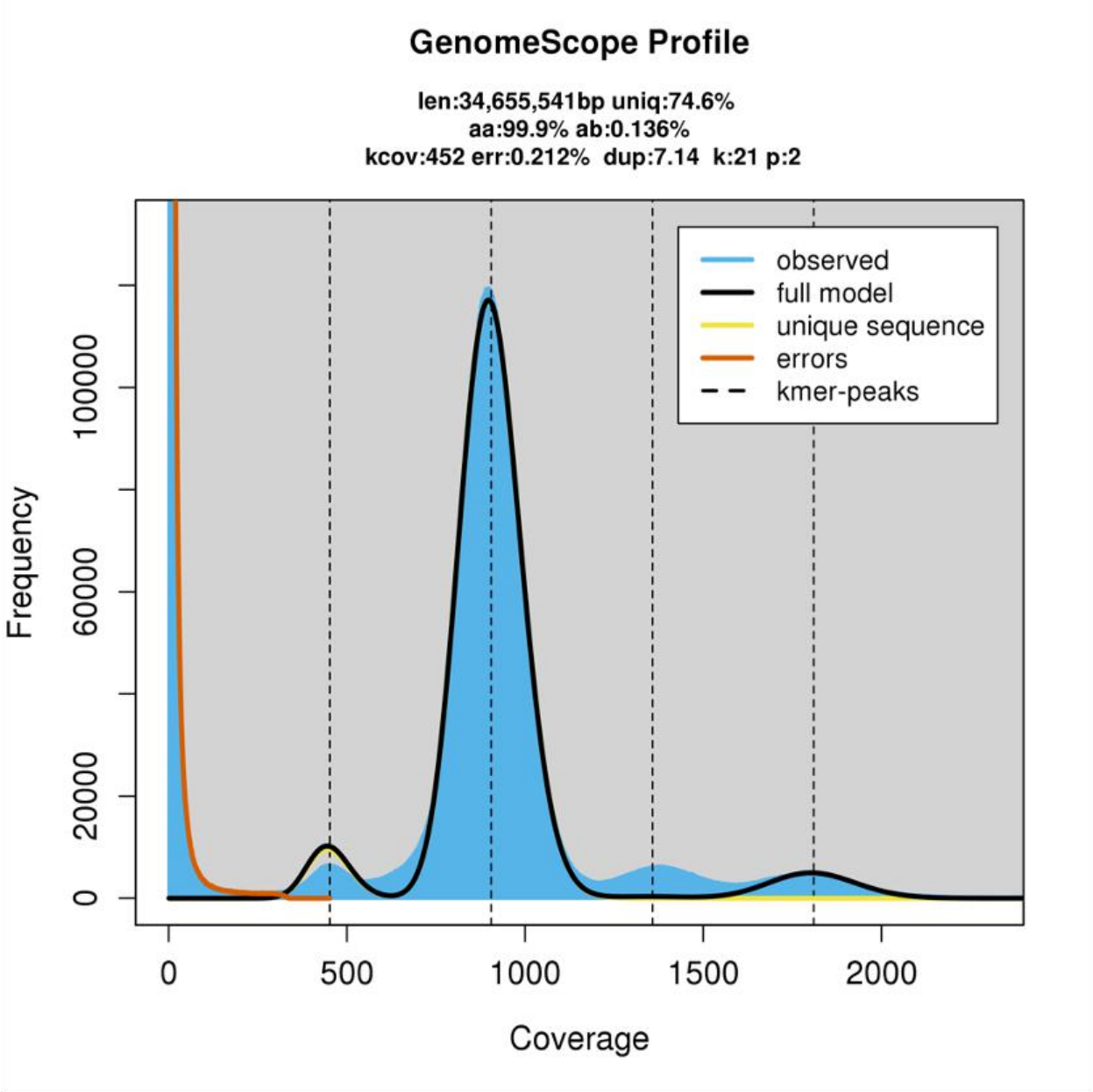

Figure A.7: The k-mer distribution from *L. donovani* short read libraries.

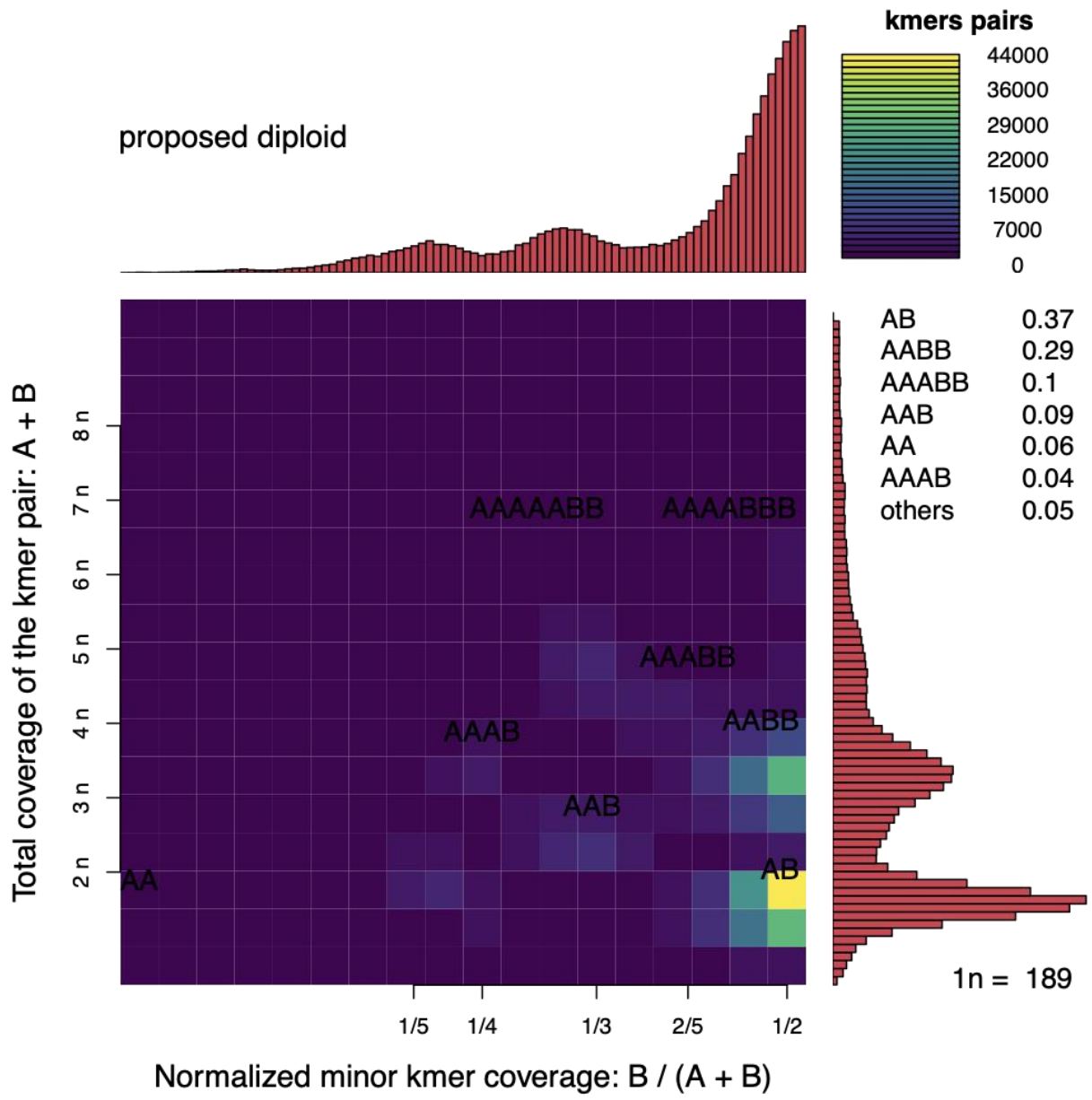

Figure A.8: The smudgeplot of *L. tarentolae.* Though similar in pattern to the *L. donovani* smudgeplot, this plot yields a proposal of a diploid genome. We believe that this is further evidence that both species in the genus are in fact diploid.

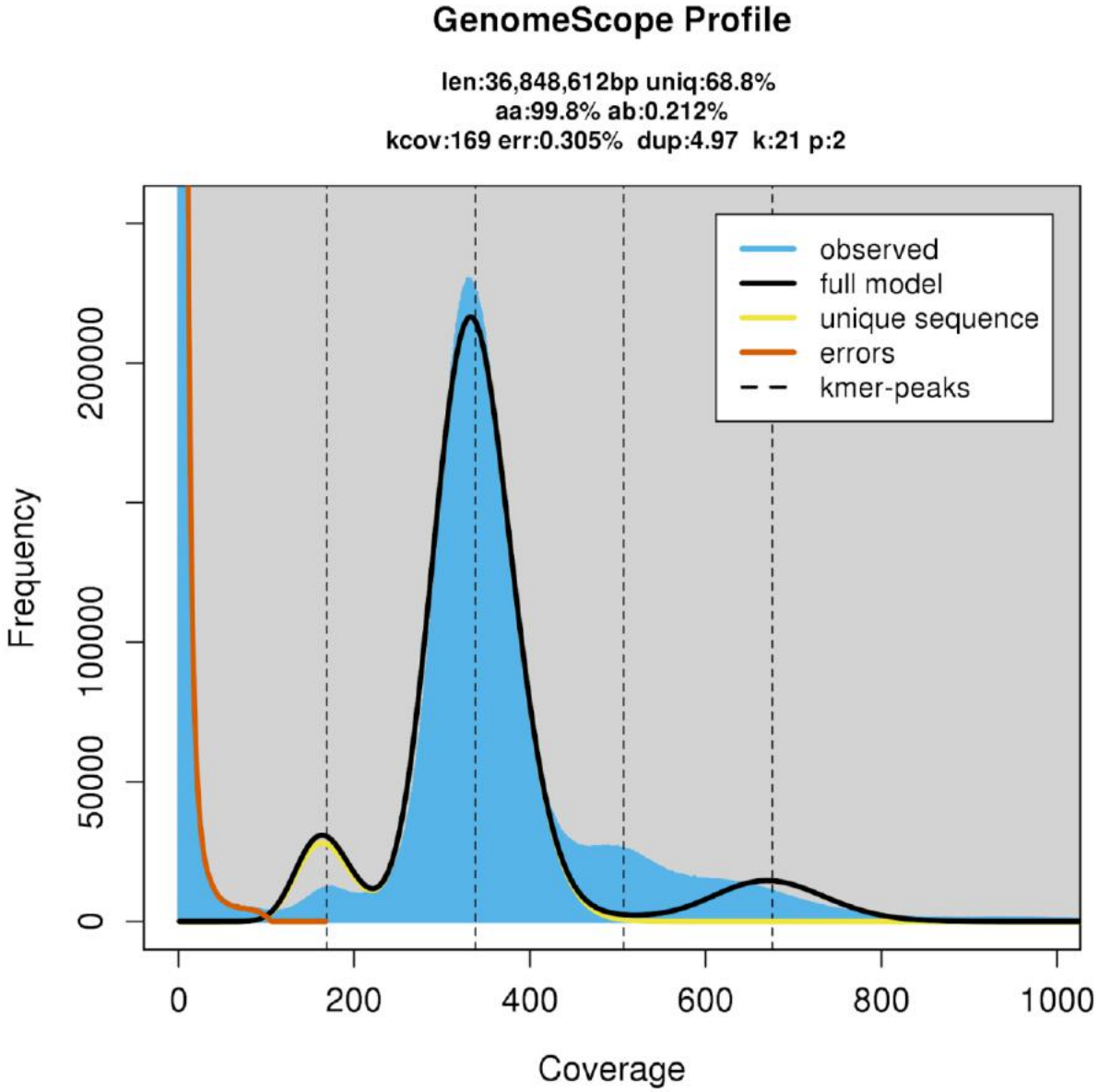

Figure A.9. The k-mer distribution from *L. tarentolae* short read libraries.

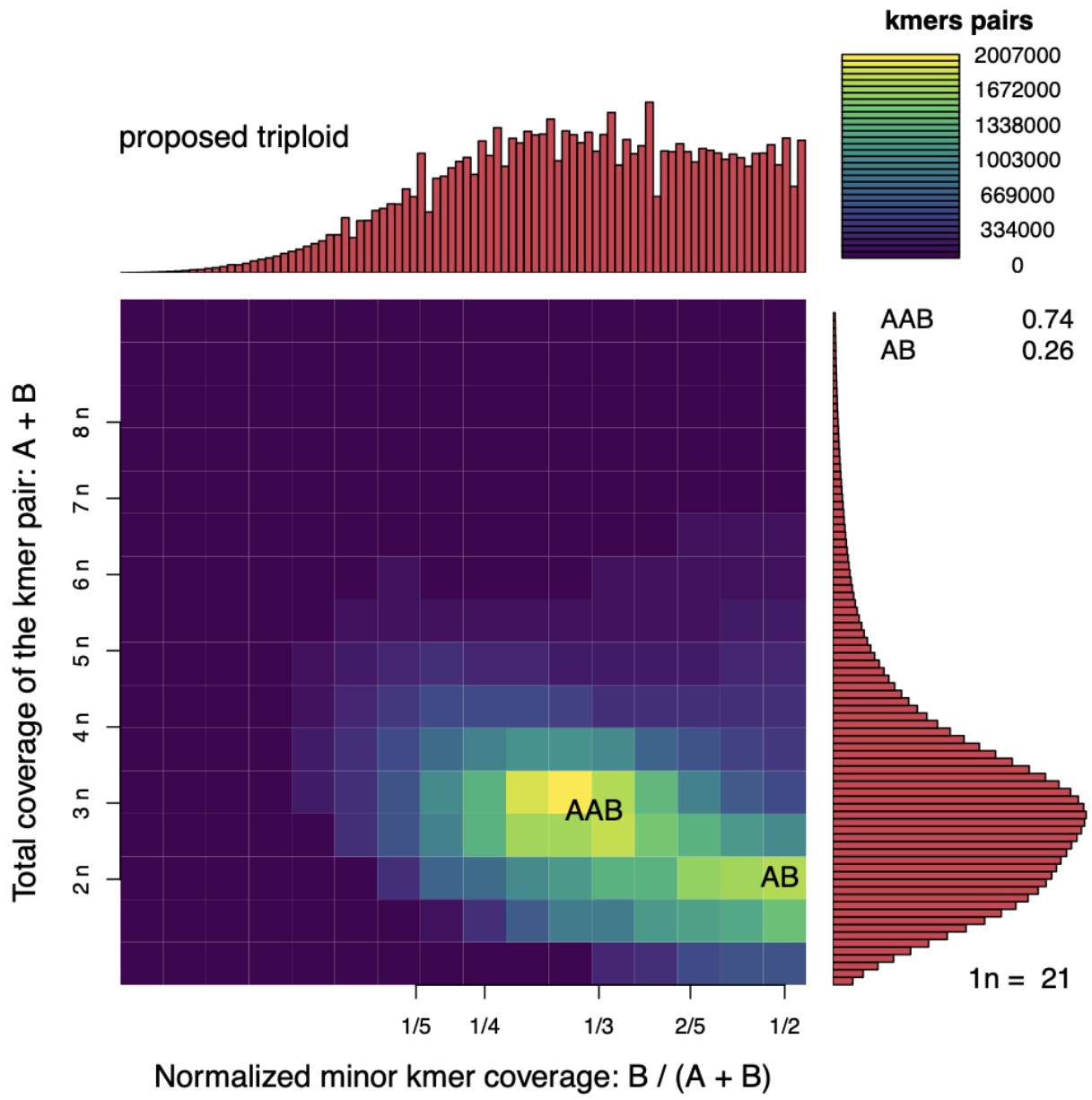

Figure A.10: The Smudgeplot for *E. gracilis* showing a strong possibility of a triploid genome. Smudgeplot computes single base pair changes between k-mers, and from the distribution of these allelic variations suggests a ploidy and heterozygosity rate.

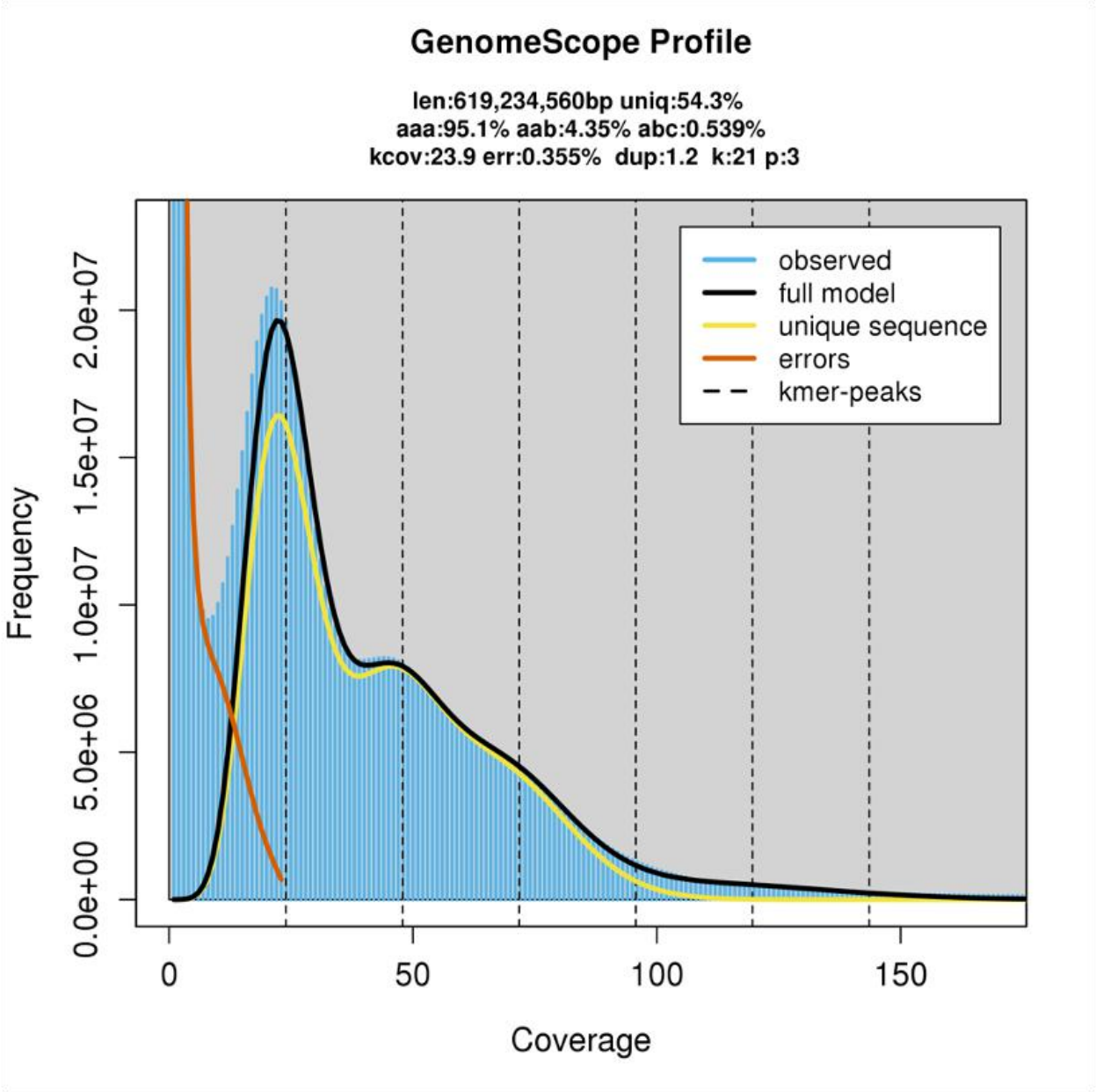

Figure A.11: The k-mer distribution from *E. gracilis* short read libraries. Using a ploidy three, as estimated by Smudgeplot, Genomescope2 estimates genome size, heterozygosity rates, and repetitiveness.

Method	N50	Contigs	Largest Contig	Total Sequence
Cambridge	955	2,066,288	166,587	1,435,499,417
Spades	2,109	2,460,937	127,613	1,628,627,972
Haslr	10,786	35,293	99,257	247,420,723
Canu	23,984	37,085	334,076	697,360,445
Masurca	293,344	6,114	2,352,880	1,022,182,428

Table A.1: A summary of results from various assemblers. These represent our best attempt with each assembler. The Cambridge row refers to the existing published draft assembly of *E. gracilis*. We found that Masurca offers the assembly metrics, though with a total sequence length that exceeds the expected haploid genome size.

Method	N50	Contigs	Largest Contig	Total Sequence
Canu	28,601	25,960	334,076	554,925,320
Canu + Long Read	75,575	13,390	532,820	558,430,984
Canu + Long Read + RNA	99,357	11,283	865,718	558,644,154
Canu + Long Read + RNA + 10x	151,983	9,689	1,312,997	558,803,554
Masurca	281,066	5,425	2,343,329	899,150,175
Masurca + Long Read	370,471	4,319	2,978,512	901,439,185
Masurca + Long Read + RNA	565,909	3,090	3,621,805	901,562,186
Masurca + Long Read + RNA + 10x	668,595	2,791	3,621,805	901,592,086

Table A.2: Scaffolding results after utilizing long reads, 10x genomics linked reads, and RNA-seq data. We focused on scaffolding the Canu assembler and Masurca assembler, as they had the highest BUSCO scores and best assembly metrics. We used LongStitch for the long-read scaffolding, p_rna_scaffolder for RNA-seq based scaffolding, and Arcs for linked read scaffolding. For each scaffolder, we did a hyperparameter search to find the best parameters.

Method	Complete	Single Copy	Duplicated	Fragmented	Missing	Searched
Cambridge	4	3	1	42	209	255
Spades	14	13	1	56	185	255
Haslr	5	5	0	14	236	255
Canu	53	47	6	60	142	255
Masurca	49	45	4	59	147	255

Table A.3: BUSCO results for each of the assembly attempts for *E. gracilis*. Canu and Masurca offer the highest BUSCO scores indicating that they are the most complete genomes, and that they likely contain the least number of base level errors.

Location	Complete	Single Copy	Duplicated	Fragmented	Missing	Searched
Cambridge	185	177	8	39	31	255
Tokyo	90	72	18	3	162	255
Liege	212	87	125	18	25	255

Table A.4: BUSCO results for each of the available transcriptomes for *E. gracilis*. The newest transcriptome from the University of Liege contains the most number of BUSCOs, but also a higher duplication rate. We chose to use the Cambridge transcriptome when assessing the quality of our genome assemblies due to its higher single copy count.

Method	Missing Genes	Gene Alignment Length
Cambridge	4.02%	60.69% ± 31.32
Spades	1.93%	72.19% ± 26.36
Haslr	50.28%	61.36% ± 29.30
Canu	13.59%	83.87% ± 21.19
Masurca	8.69%	89.56% ± 16.15

Table A.5: Transcript alignment results for each *E. gracilis* genome assembly. Taking the Cambridge transcriptome, we mapped all the genes to each genome assembly to determine the quality of alignments. Missing genes refers to the number of transcripts that do not have

APPENDIX B

Computing the accuracy of an assembly requires determining whether the grouping, order, and orientation of each contig within a scaffold matches its position in the corresponding chromosome of the reference genome. Here, we outline the steps necessary to do so, along with the equations to compute accuracy metrics.

B.1 MAPPING

All scaffolding methods output a FASTA file, in which contigs are joined in some particular order. We produce an "A Golden Path" (AGP) file to record this order, and then disassemble it into its constituent contigs by splitting scaffolds at runs of Ns (default: 10 Ns)[71]. MUMmer 4 is then used to determine where contigs map on the reference genome. Alignments must have a minimum alignment length (default: 1000bp) and a minimum percent of the query sequence aligning to the reference (default: 20%), otherwise they are excluded from subsequent steps. From this coordinate map, we generate another AGP file representing the ideal scaffolding. Evaluation of scaffolding accuracy can then be computed solely from these two AGP files.

B.2 EDIT DISTANCE

The method for computing the edit distance between two assemblies A and B begins by constructing the adjacency graph. In this bipartite graph, the two sets of vertices are the adjacencies in each genome, and edges connect vertices with overlapping adjacencies. The algorithm for creating the adjacency graph is outlined in greater detail in the original Double Cut and Join edit distance paper[54]. The distance can then be computed as a function of N, the number of contigs, C, the number of cycles and I, the number of odd paths in the adjacency graph:

$$f_{distance} = N - \left(C + \frac{I}{2}\right)$$

In two assemblies that are identical, all contigs are involved in cycles of length two or odd paths of length one. This observation lets us compute a length weighted version of edit distance, our notion of accuracy, where the longest two contigs in each cycle and odd path are taken to represent the number of bases that do not have to be moved:

$$f_{accuracy} = len(C) + \frac{len(I)}{2}$$

Intuitively, we can divide scaffolding into three tasks: grouping contigs into chromosomes, ordering contigs, and orienting them such that contiguous ends are touching. Though a scaffolder may not explicitly perform these tasks, they are always implicit in the output, allowing any scaffolder to be compared on these common fronts. Here, we further define the metrics for each of these sub-tasks.

B.2.1 Grouping

To evaluate grouping performance, we must determine the degree to which scaffolds overlap with reference chromosomes. Suppose there exists some reference chromosome A_i and some assembly scaffold B_j, then the intersection of these two sets of contigs are those contigs which belong to both the chromosome and the scaffold. We define the length weighted Jaccard index as the sum of contig lengths in the intersection divided by the sum of contig lengths in the union for any two sets:

$$J(A_i, B_j) = \frac{len(A_i \cap B_j)}{len(A_i \cup B_j)}$$

We then find the maximum length weighted Jaccard index for each reference chromosome by iterating through all the assembly scaffolds. These maximum Jaccard values are then weighted by the length of the reference chromosome it corresponds to, such that smaller chromosomes get weighed less. The sum of these weighted Jaccard maximums are then divided by the length of the reference genome:

$$f_{grouping}(A,B) = \frac{\sum_{i=0}^{|A|} argmax_j \left(J(A_i, B_j)\right) * len(A_i)}{len(A)}$$

Since the Jaccard index is a value between 0 and 1, the grouping score is also a value between 0 and 1.

B.2.2 Ordering

The ordering performance can be evaluated by determining how many contigs were next to their expected adjacency. Given contigs k and l in the reference A, let their adjacency be kl. The two adjacencies kl and lk are then the same. The length weighted adjacency is then the length of k plus the length of l. If we record all the length weighted adjacencies in the reference A and the assembly B, then the ordering score is the sum of adjacencies in the intersection divided by the sum of adjacencies in the reference A:

$$f_{ordering}(A,B) = \frac{len(A_{edges} \cap B_{edges})}{len(A_{edges})}$$

B.2.3 Orientation

In a similar fashion, we can compute the orientation accuracy if we construct our adjacency set to also include orientation. Here, we retain which end of each contig is adjacent when recording the set of adjacencies such that contigs i and j might create an edge $i_h j_t$ indicating the head of i

is contiguous with the tail of j. Again, the adjacency $i_h j_t$ is equivalent to $j_t i_h$. Then the orientation score is the sum of adjacencies in the intersection divided by the sum of adjacencies in reference A:

$$f_{ordering}(A, B) = \frac{len\left({A_{edges}}' \cap {B_{edges}}'\right)}{len\left({A_{edges}}'\right)}$$

B.3 Software

We implement the methods in a python package which is freely available under the MIT license (https://github.com/Noble-Lab/edison). It takes in two inputs, a reference FASTA and scaffolded FASTA file, and chiefly produces five metrics of accuracy. While running, it also produces a plot of how the contigs map to the reference genome and the scaffolds to which they belong (Figure B.12). Additionally, it produces two AGP files representing how the assembly places contigs and how they ought to be placed to most closely match the reference.

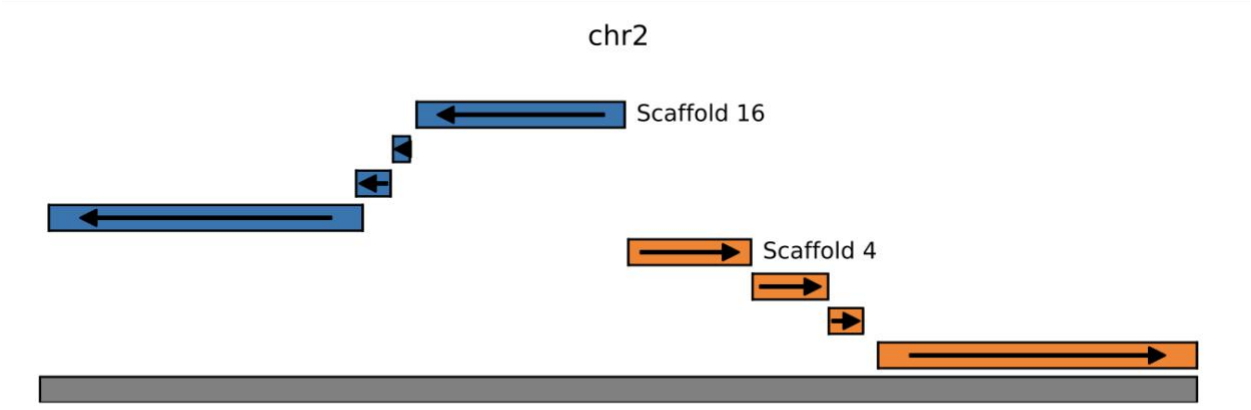

Figure B.12: Visualizing contig alignments to a reference genome. Here, each row represents a new contig, where the horizontal coordinates indicate where on the reference chromosome they aligned, the arrow indicates the orientation of alignment, and the color corresponds to scaffold membership.

Method	Year	N50	Visual Plots	Misassembly	Grouping	Ordering	Orientation	Edit Distance
Bambus[78]	2004	✓				✓*	✓*	
Soma[79]	2008					✓*		
Amos[80]	2008			✓				
Mauve[81]	2011		✓	✓				✓*
Sspace[82]	2011	✓						
Gage[83]	2013	✓	✓	✓*				
Lachesis[21]	2013	✓	✓		✓	✓	✓	
Hunt's[84]	2014					✓*	✓*	
Hirise[61]	2016	✓				✓*	✓*	
3d-dna[37]	2017	✓	✓					
Arks[58]	2018	✓	✓	✓*				
Salsa[62]	2019	✓			✓*	✓*	✓*	
Scop[85]	2019					✓*	✓*	
Lrscaf[86]	2019	✓		✓*				
Slr[87]	2019	✓		✓*				
Allhic[63]	2019	✓	✓					
Ragoo[56]	2019	✓	✓	✓*	✓*	✓*	✓*	✓*
Ldscaff[88]	2020	✓	✓	✓				
Edison[70]	2022	✓			✓	✓	✓	✓

Table B.1: Scaffolding methods and the metrics they use to assess the accuracy. “Visual plots” refers to any kind of visual analysis, such as dot plots, linkage plots, and circle plots. “Misassemblies” refers to the counting of structural variants such as inversions, deletions, substitutions, and translocations. *Indicates that the metric is a count and is not weighted by the length of the contigs.

APPENDIX C

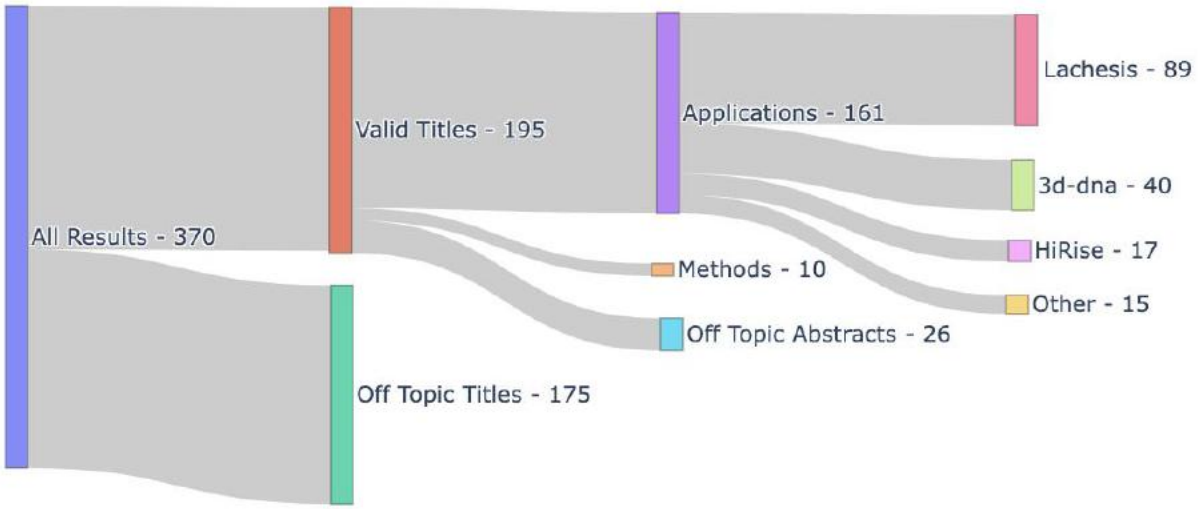

Figure C.13: A sankey diagram depicting the literature search process. We identified ten different Hi-C scaffolding methods in the literature and found that their usage varied significantly, with only five methods showcasing more than three published genomes.

	Grouping		Order		Orientation		Accuracy	
Method	**Split**	**De novo**	**Split**	**De novo**	**Split**	**De novo**	**Split**	**De novo**
lachesis	**0.91 ± 0.1**	**0.71 ± 0.2**	**0.95 ± 0.03**	**0.66 ± 0.1**	**0.95 ± 0.03**	**0.55 ± 0.2**	**0.95 ± 0.03**	0.55 ± 0.2
hirise	0.74 ± 0.3	0.69 ± 0.2	0.87 ± 0.2	0.49 ± 0.3	0.87 ± 0.2	0.44 ± 0.3	0.87 ± 0.2	**0.62 ± 0.2**
3d_dna	0.35 ± 0.2	0.38 ± 0.3	0.66 ± 0.3	0.34 ± 0.2	0.66 ± 0.3	0.30 ± 0.2	0.71 ± 0.2	0.41 ± 0.2
salsa	0.52 ± 0.3	0.47 ± 0.3	0.85 ± 0.2	0.44 ± 0.2	0.84 ± 0.2	0.41 ± 0.2	0.85 ± 0.1	0.50 ± 0.2
allhic	0.75 ± 0.4	0.52 ± 0.3	0.95 ± 0.1	0.58 ± 0.3	0.94 ± 0.1	0.49 ± 0.3	0.84 ± 0.2	0.39 ± 0.3
baseline	0.18 ± 0.3	0.36 ± 0.3	0.03 ± 0.1	0	0.03 ± 0.1	0	0.16 ± 0.2	0.28 ± 0.3

Table C.6: An overview of performance of each of the methods based on their average accuracy determined by Edison. The split column refers to the task of scaffolding equal sized pieces of the reference genome. The *de novo* column refers to the task of scaffolding the assemblies created by Canu. Baseline represents the score for contigs without scaffolding.

Organism	Genome Size	Reads	Bases	Coverage	BioProject
S. cerevisiae	12,100,000	313,114	1,701,530,052	141	PRJEB7245
L. tarentolae	32,200,000	1,360,815	7,198,339,498	224	PRJNA821548
A. thaliana	135,000,000	7,353,356	49,942,606,909	370	PRJNA314706
H. sapiens	3,100,000,000	47,885,330	328,978,598,683	106	PRJNA301527

Table C.7: Overview of data collected for *de novo* genome assemblies. The amount of data is roughly proportional to the size of the genome such that they can be down sampled to a similar read coverage.

Coverage	Yeast N50	Leishmania N50	Arabidopsis N50	Human N50
10	21,825	9,250	24,823	18,711
20	176,516	18,595	86,731	43,231
30	551,752	41,537	159,879	105,563
40	568,123	40,610	147,845	626,886
50	614,056	94,577	149,418	1,873,143
60	813,309	213,275	161,138	2,018,914
70	777,771	71,335	140,209	4,761,131
80	813,629	117,224	149,909	7,508,518
90	813,427	229,090	164,531	9,231,632
100	930,538	332,478	186,445	10,553,285

Table C.8: Overview of the *de novo* assemblies created by Canu. We generated ten assemblies for each species and down sampled reads used to create them to vary their N50s. As a general trend, we observed that increased read coverage led to long contigs. Arabidopsis appeared to be an outlier of this trend, and its genome assembly appeared to be challenging and indicative of high rates of heterozygosity.

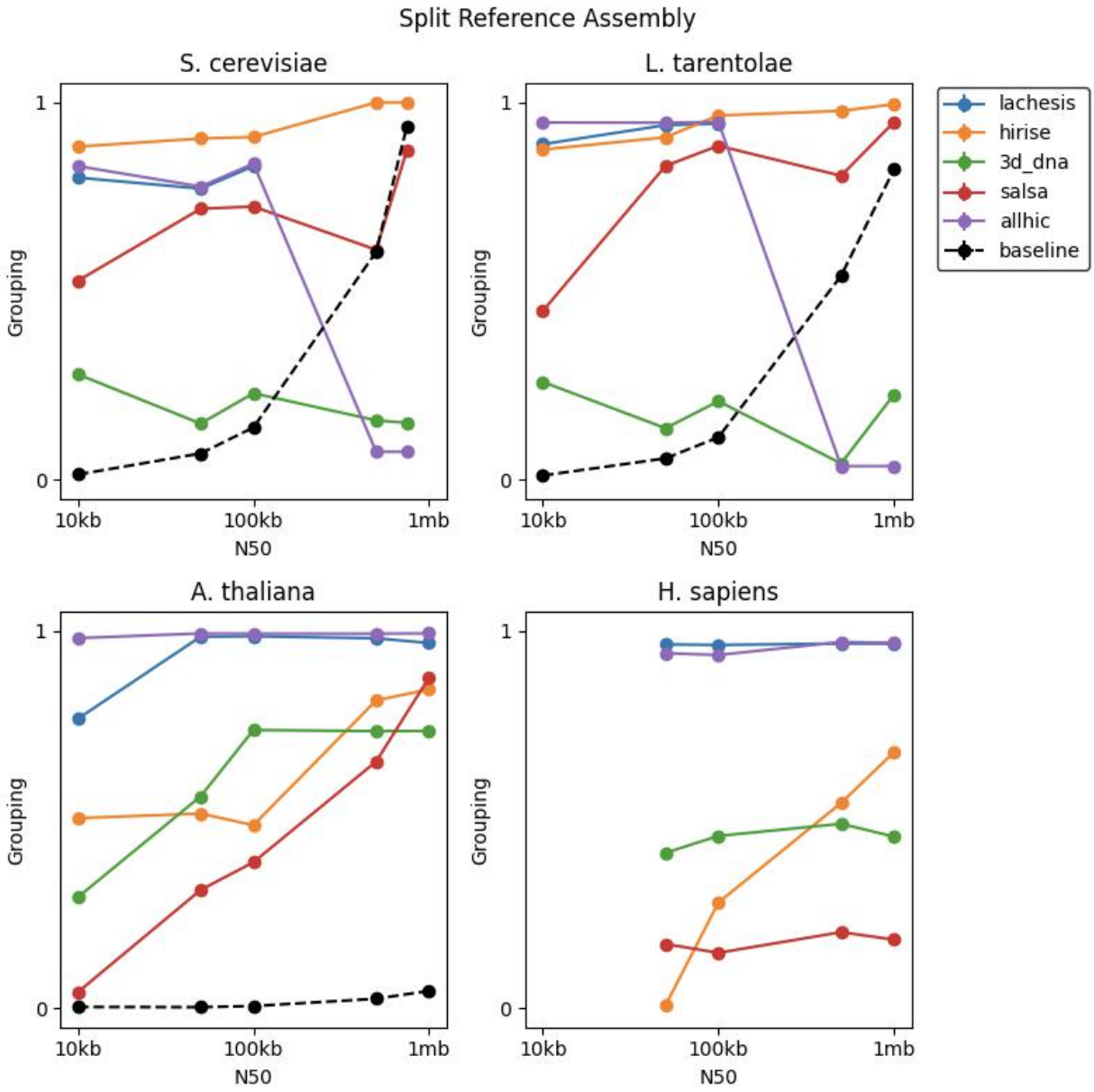

Figure C.14: The grouping scores of scaffolders on split reference assemblies. There is wide variation in grouping performance, with trends pointing to difficulty with small assemblies with large N50s and large assemblies with small N50s.

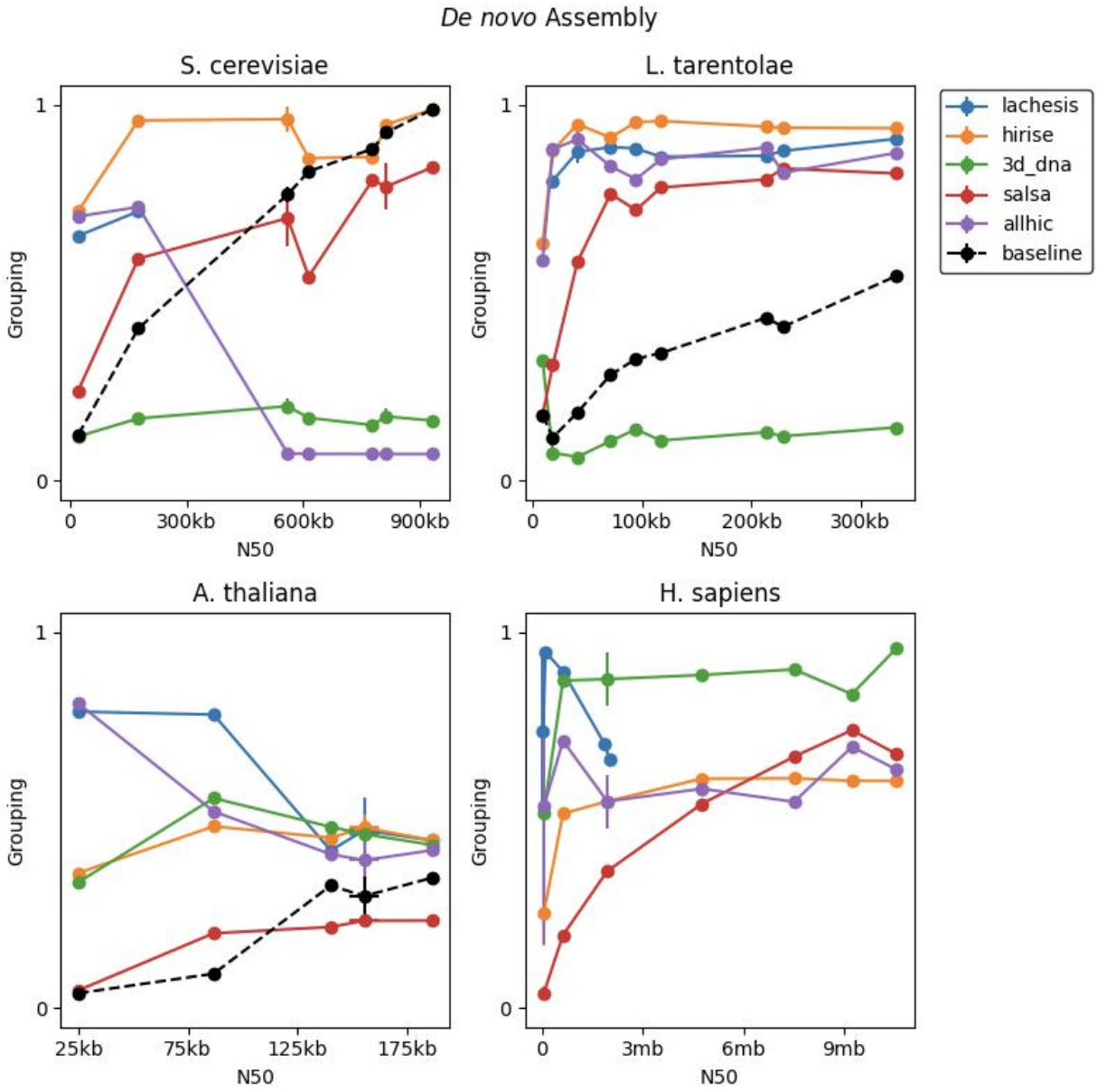

Figure C.15: The grouping scores for Hi-C scaffolders on *de novo* assemblies. Higher grouping accuracy indicates that scaffolders were able to uniquely isolate contigs belonging to the same chromosome within scaffolds.

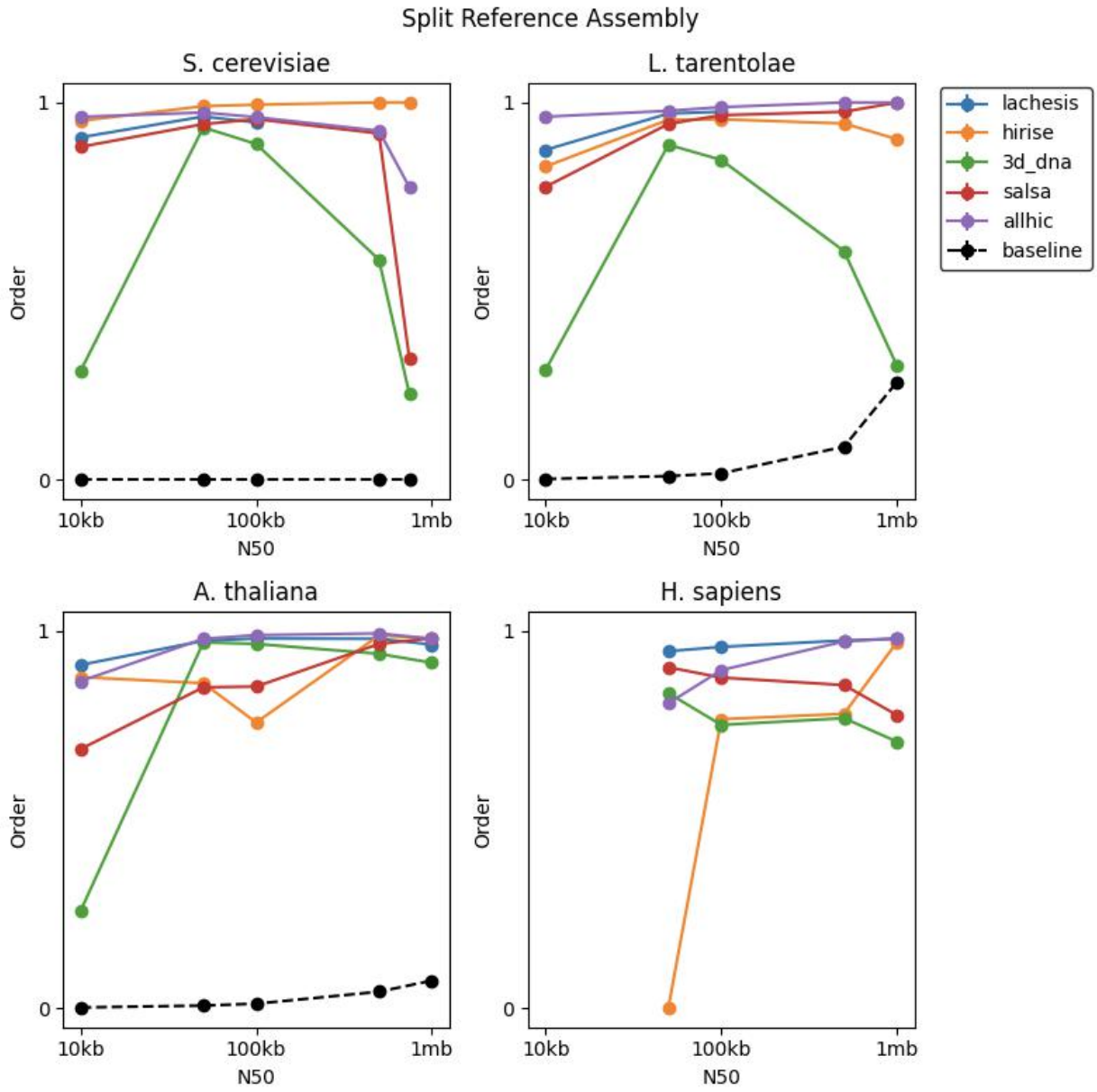

Figure C.16: The order scores for Hi-C scaffolders on split reference assemblies.

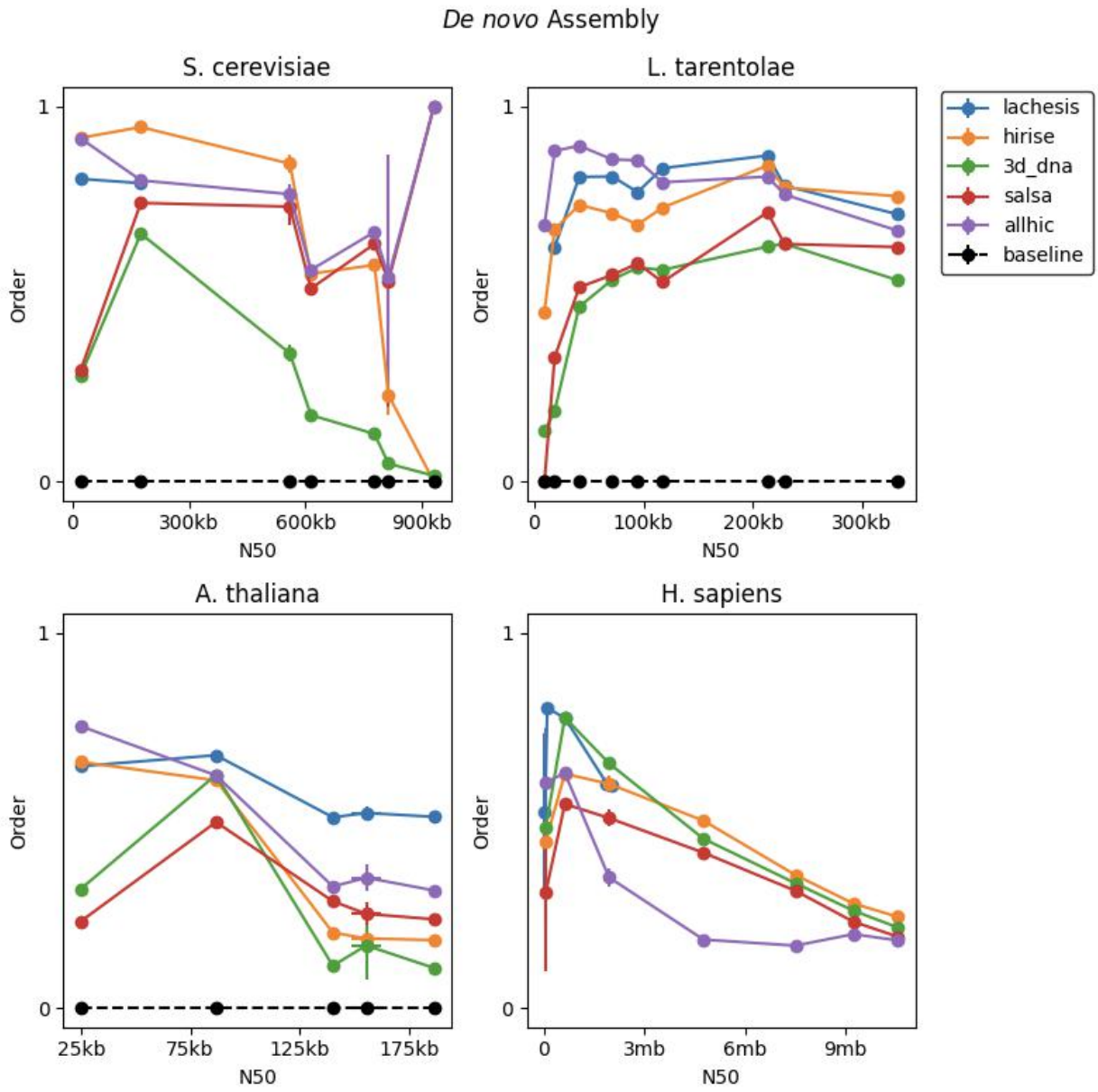

Figure C.17: The order scores for Hi-C scaffolders on *de novo* assemblies. Higher order accuracy indicates that scaffolders were able to correctly place contigs next to their expected neighbors.

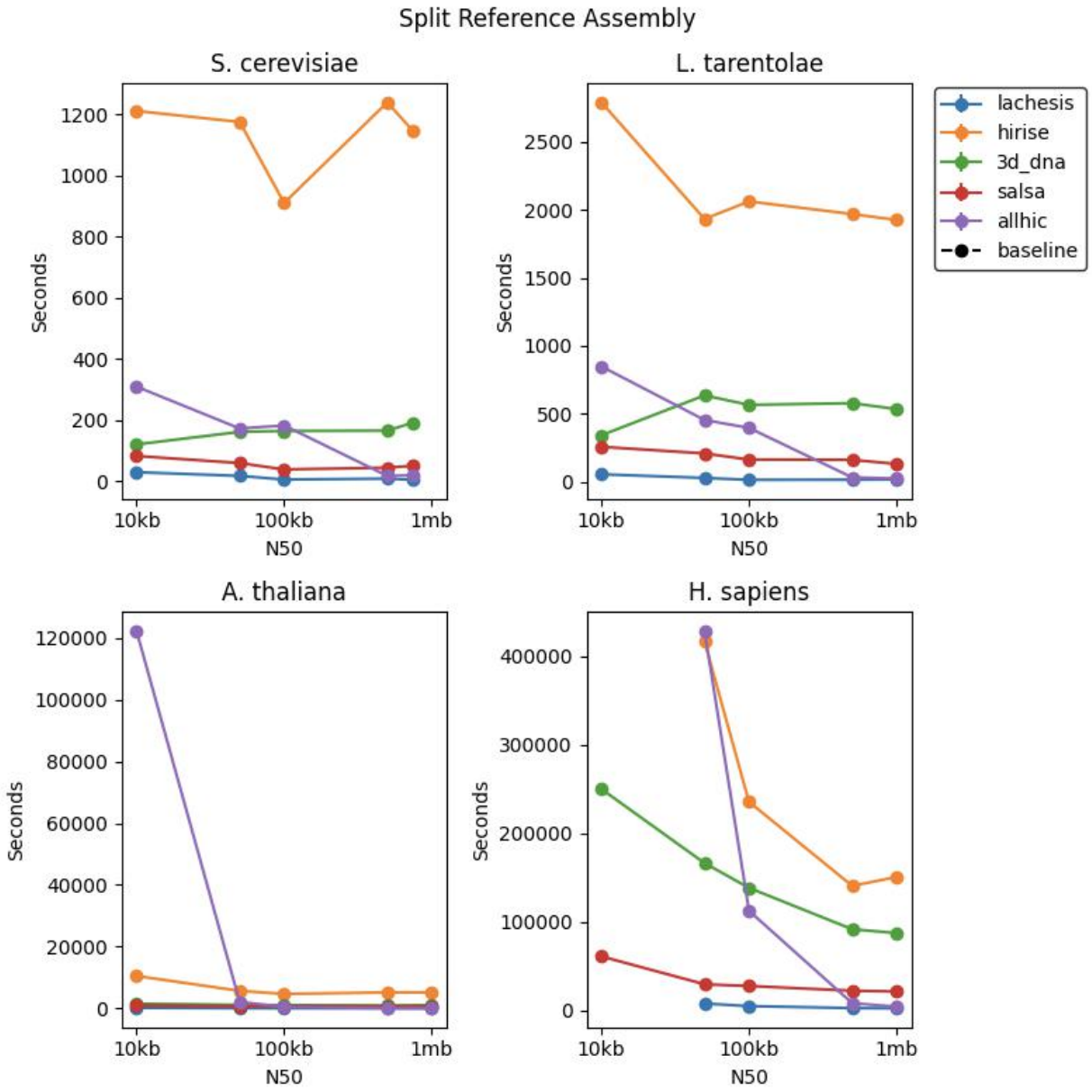

Figure C.18 The runtime of Hi-C scaffolders on split assemblies.

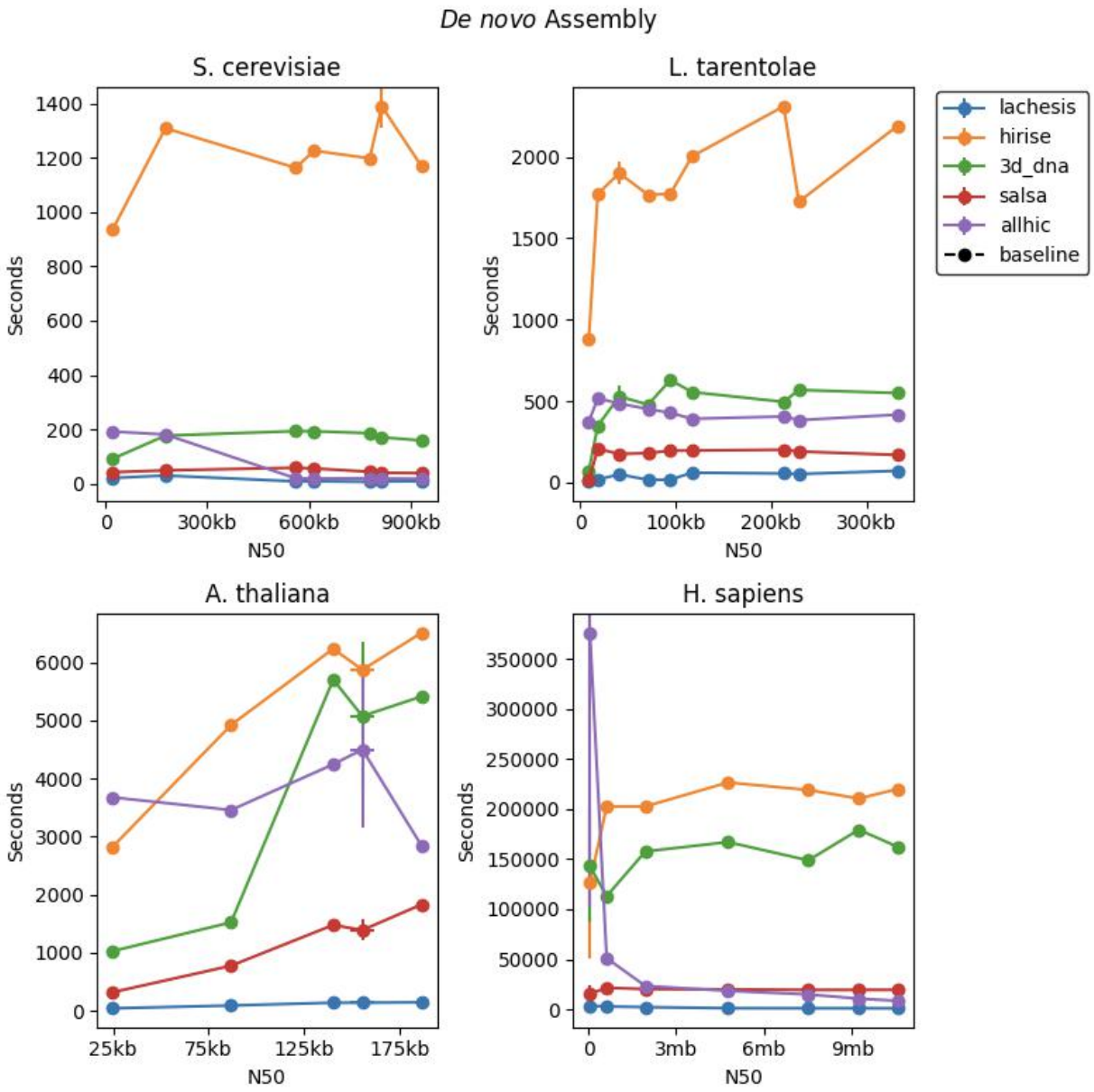

Figure C.19: The runtime of Hi-C scaffolders on *de novo* assemblies. Hirise is generally the slowest and Lachesis the fastest scaffolder.

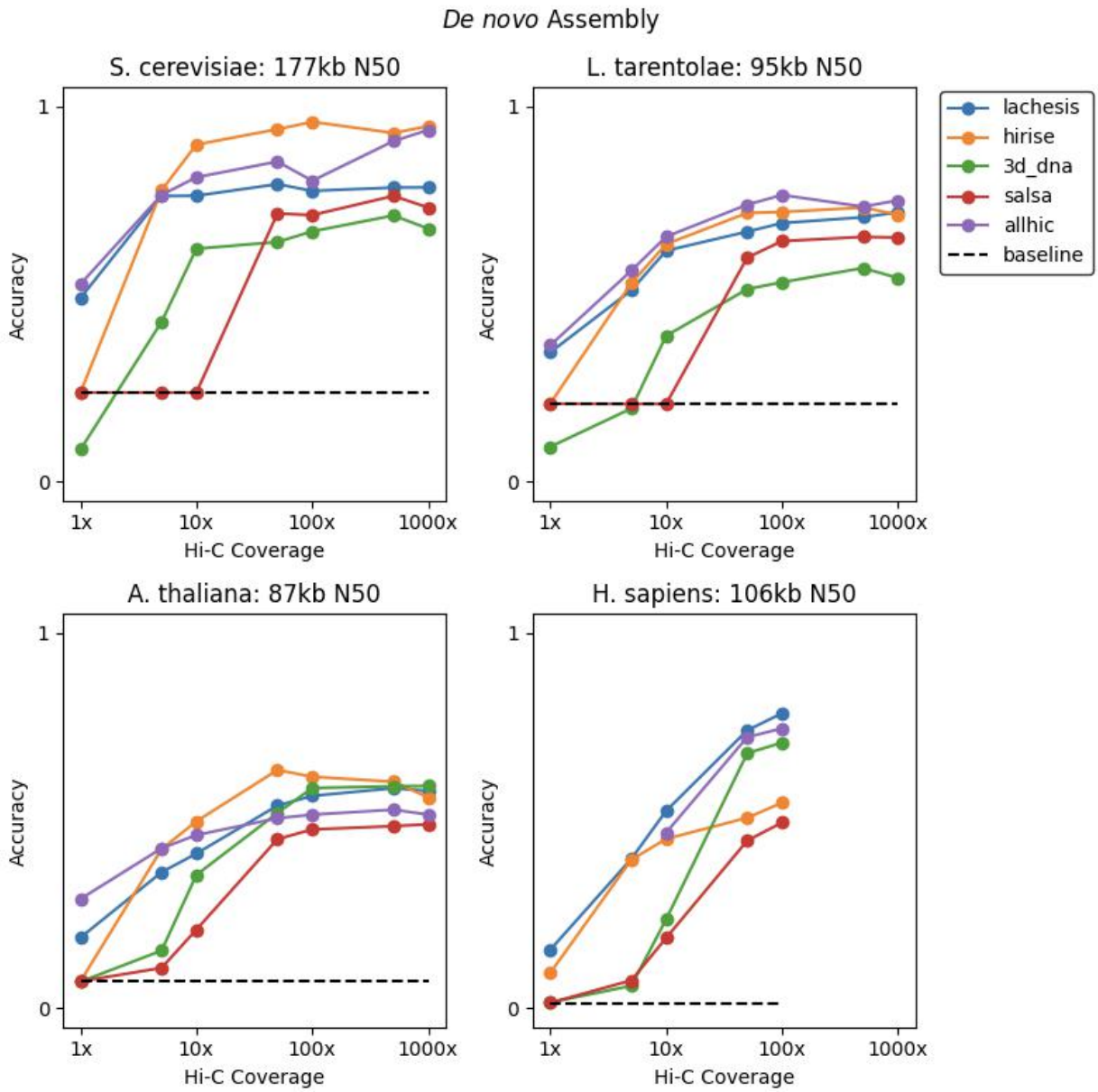

Figure C.20: Downsampling of Hi-C reads on *de novo* assemblies. The same trend as the split references is seen here, where Hi-C read densities below 50 reads per kilobase lead to a decline in performance.

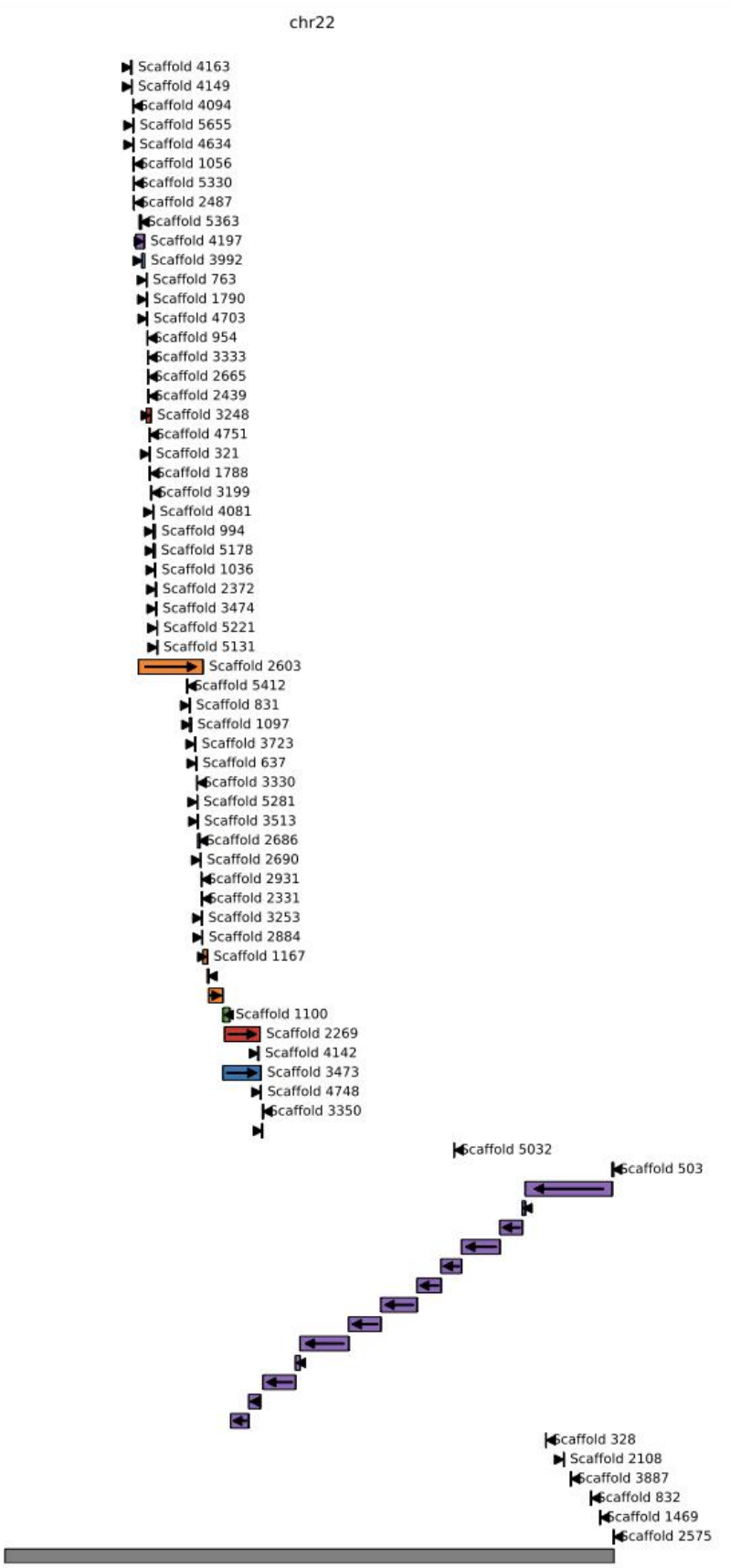
chr22
Scaffold 4163
Scaffold 4149
Scaffold 4094
Scaffold 5655
Scaffold 4634
Scaffold 1056
Scaffold 5330
Scaffold 2487
Scaffold 5363
Scaffold 4197
Scaffold 3992
Scaffold 763
Scaffold 1790
Scaffold 4703
Scaffold 954
Scaffold 3333
Scaffold 2665
Scaffold 2439
Scaffold 3248
Scaffold 4751
Scaffold 321
Scaffold 1788
Scaffold 3199
Scaffold 4081
Scaffold 994
Scaffold 5178
Scaffold 1036
Scaffold 2372
Scaffold 3474
Scaffold 5221
Scaffold 5131
Scaffold 2603
Scaffold 5412
Scaffold 831
Scaffold 1097
Scaffold 3723
Scaffold 637
Scaffold 3330
Scaffold 5281
Scaffold 3513
Scaffold 2686
Scaffold 2690
Scaffold 2931
Scaffold 2331
Scaffold 3253
Scaffold 2884
Scaffold 1167
Scaffold 1100
Scaffold 2269
Scaffold 4142
Scaffold 3473
Scaffold 4748
Scaffold 3350
Scaffold 5032
Scaffold 503
Scaffold 328
Scaffold 2108
Scaffold 3887
Scaffold 832
Scaffold 1469
Scaffold 2575

Figure C.21: An overview of how HiRise scaffolded the 10mb N50 *H. sapiens* assembly for Chromosome 22. Each row represents a contig, and each label and color represents a scaffold. The x-axis represents the alignment based position of the contig, and the y-axis represents the scaffolder based order of the contigs. Here, HiRise picks out a set of larger contigs and scaffolds them in the correct order relative to each other. However it leaves out a number of the smaller contigs, including ones that overlap with its primary scaffold (Scaffold 503) for this chromosome.

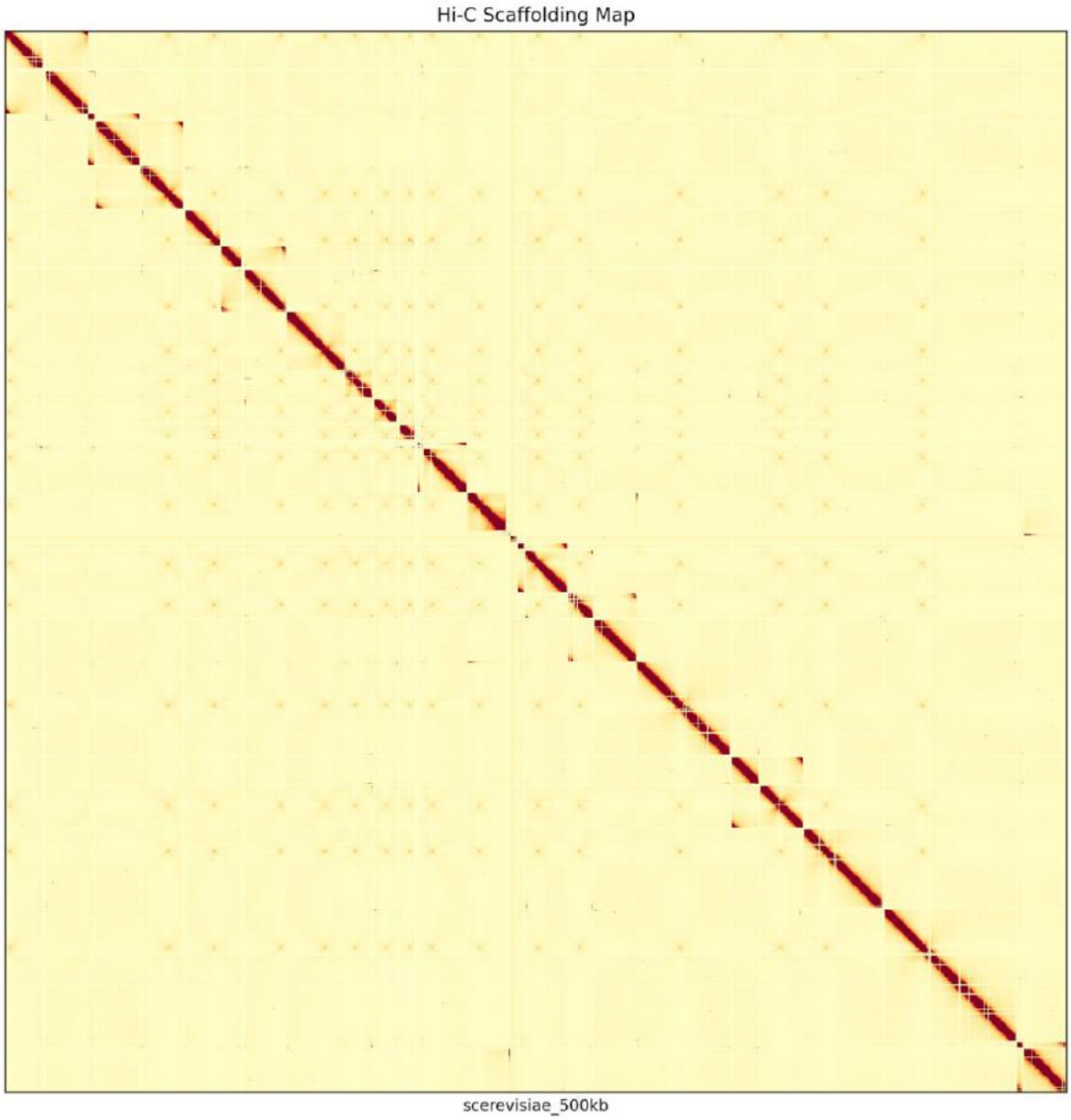

Figure C.22: AllHiC scaffolding 500kb contigs from the split reference assembly of *S. cerevisiae.* While the contigs have been placed mostly in the correct order and orientation, all the contigs were placed in a single mega-scaffold causing the overall accuracy to dramatically decrease for this particular scaffolding.

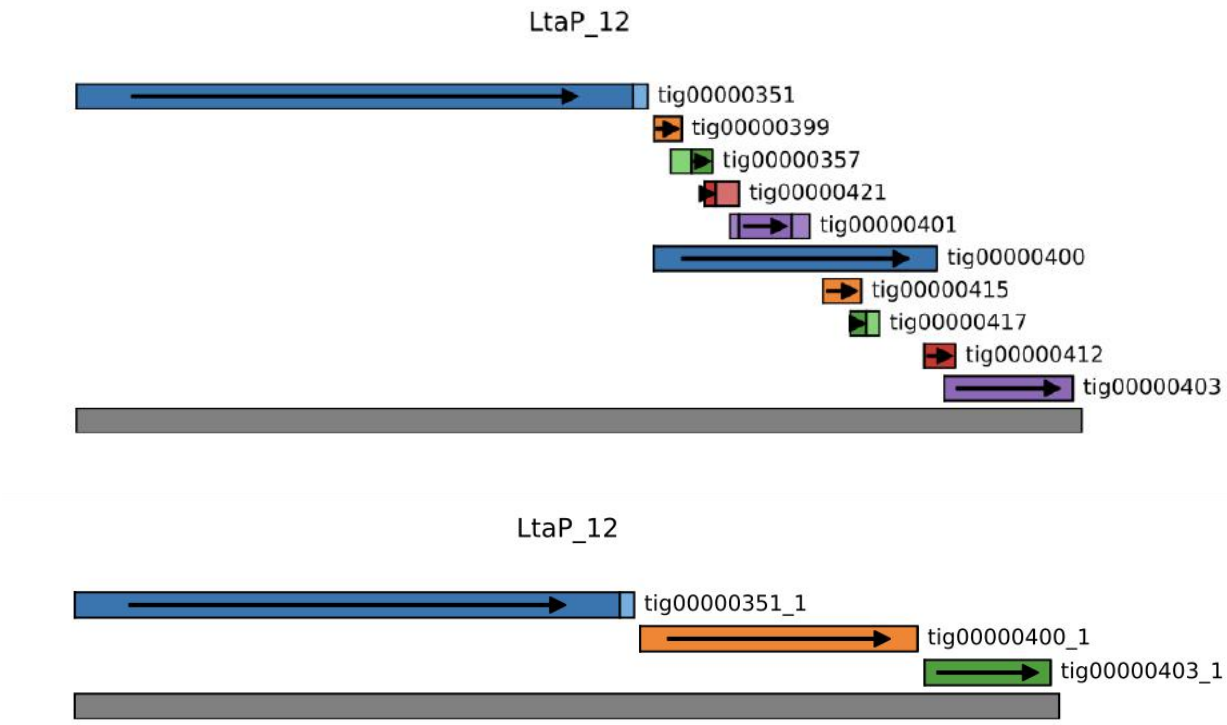

Figure C.23: Using purge_dups to remove halpotigs. The top section shows the contigs of the original assembly for *L. tarentolae* that map to chromosome 12. The bottom section shows the remaining contigs after the purging of haplotigs.

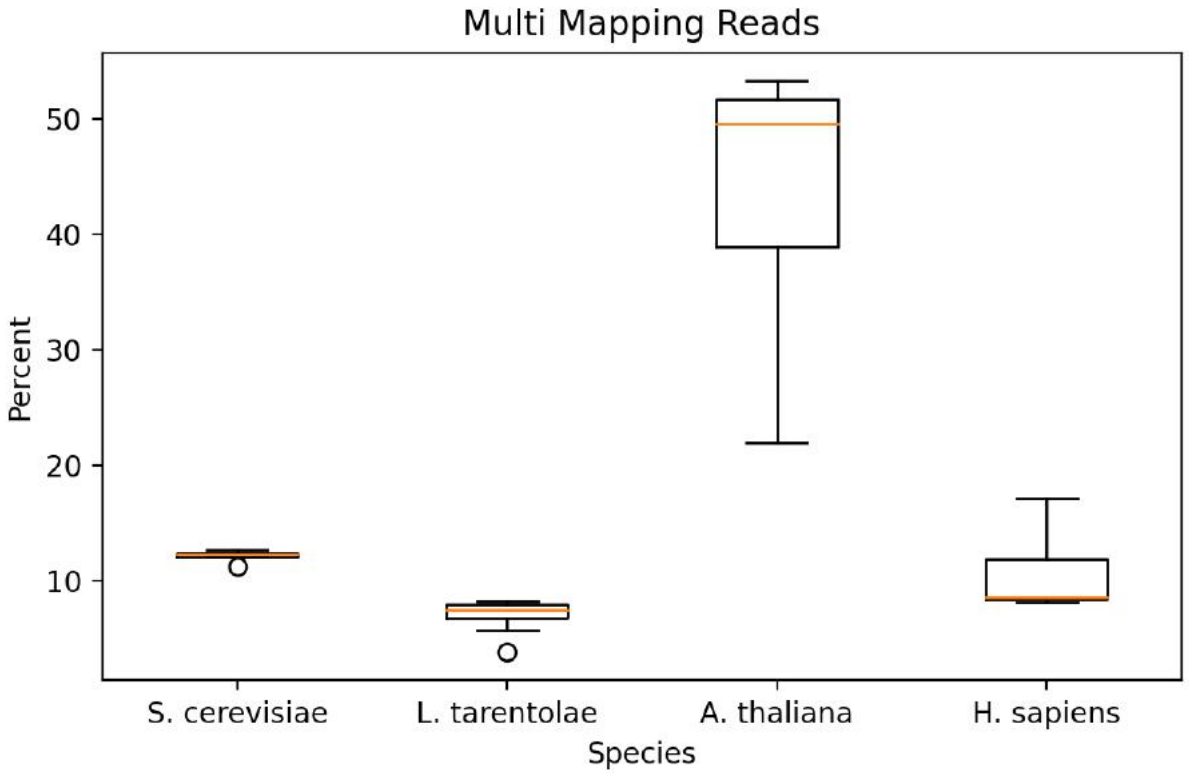

Figure C.24: The percent of reads that map to multiple positions in the *de novo* assembly. We found that as the number of reads used to create the *de novo* assembly goes up, the repetitive content of the genome goes up. The uniformly low accuracy against *A. thaliana* assemblies can likely be attributed to a high percentage of multi-mapping reads, which cannot be used by Hi-C scaffolders.

APPENDIX D

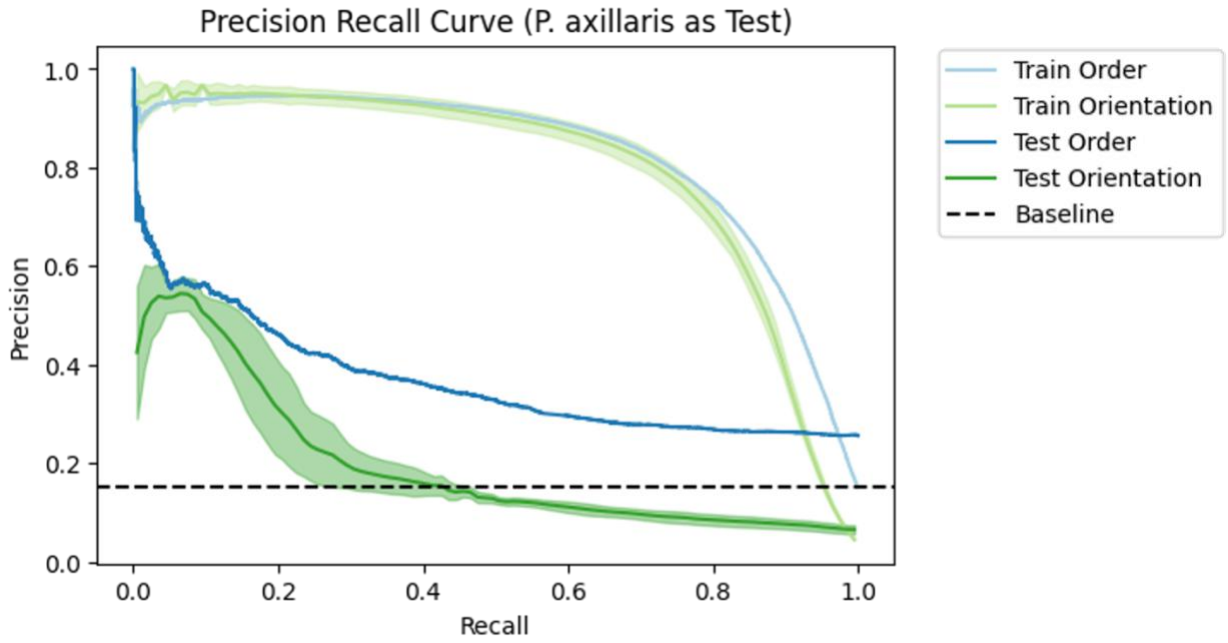

Figure D.25: The precision recall curve while holding *P. axillaris* as the validation set.

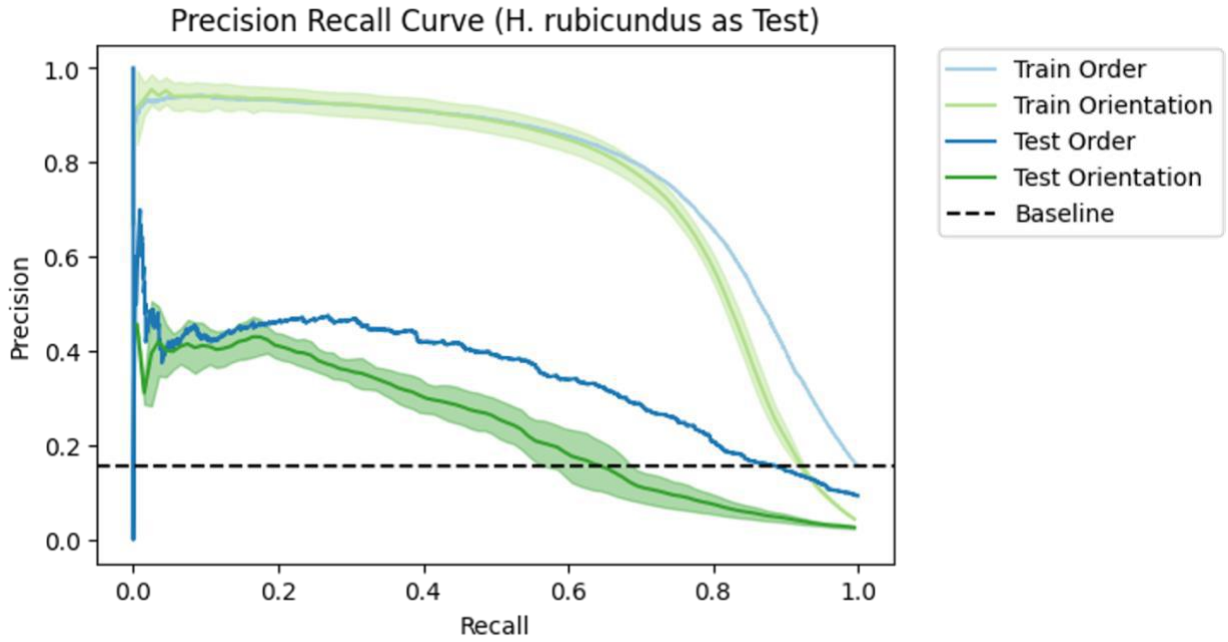

Figure D.26: The precision recall curve while holding *H. rubicundus* as the validation set.

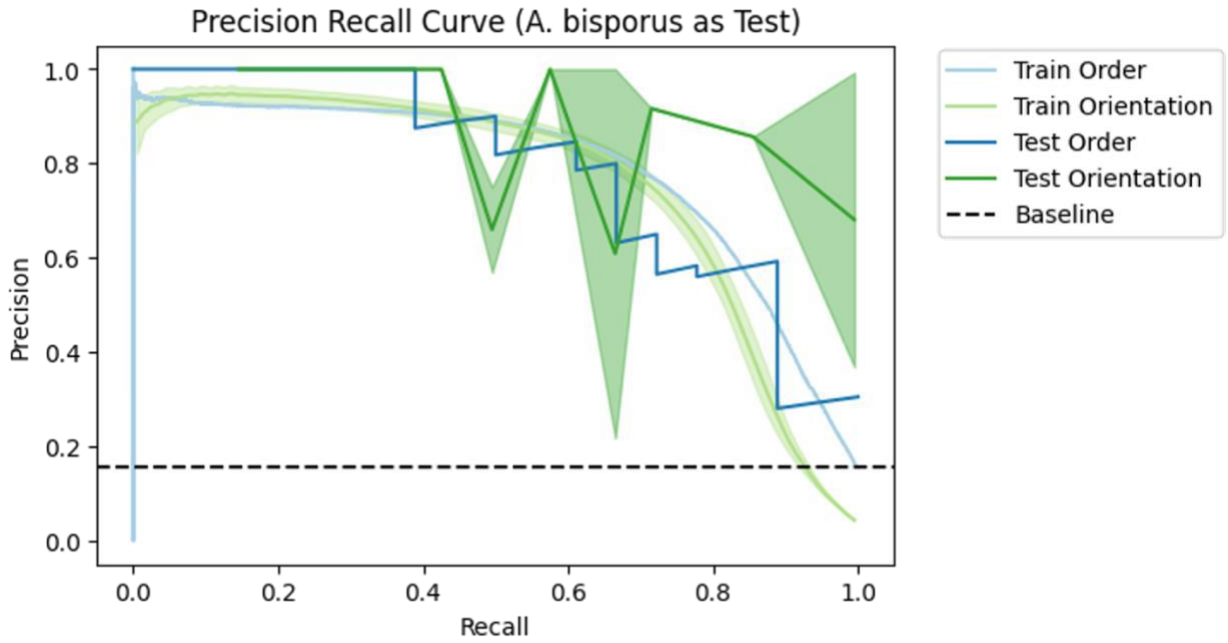

Figure D.27: The precision recall curve while holding *A. bisporus* as the validation set.

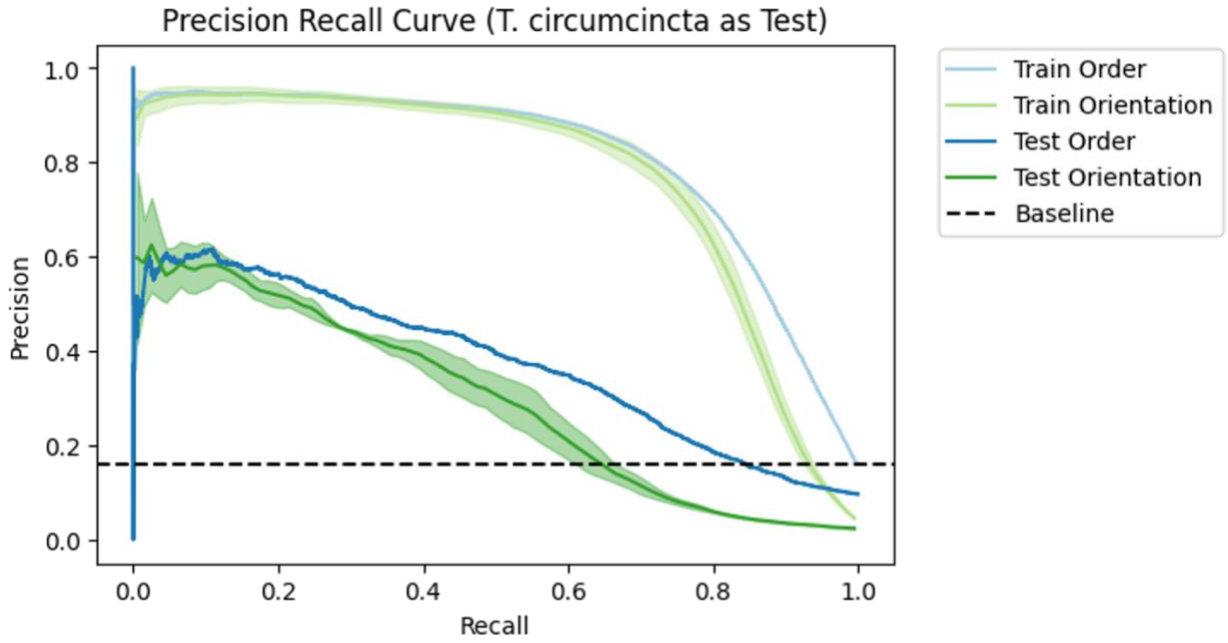

Figure D.28: The precision recall curve while holding *T. circumcincta* as the validation set.

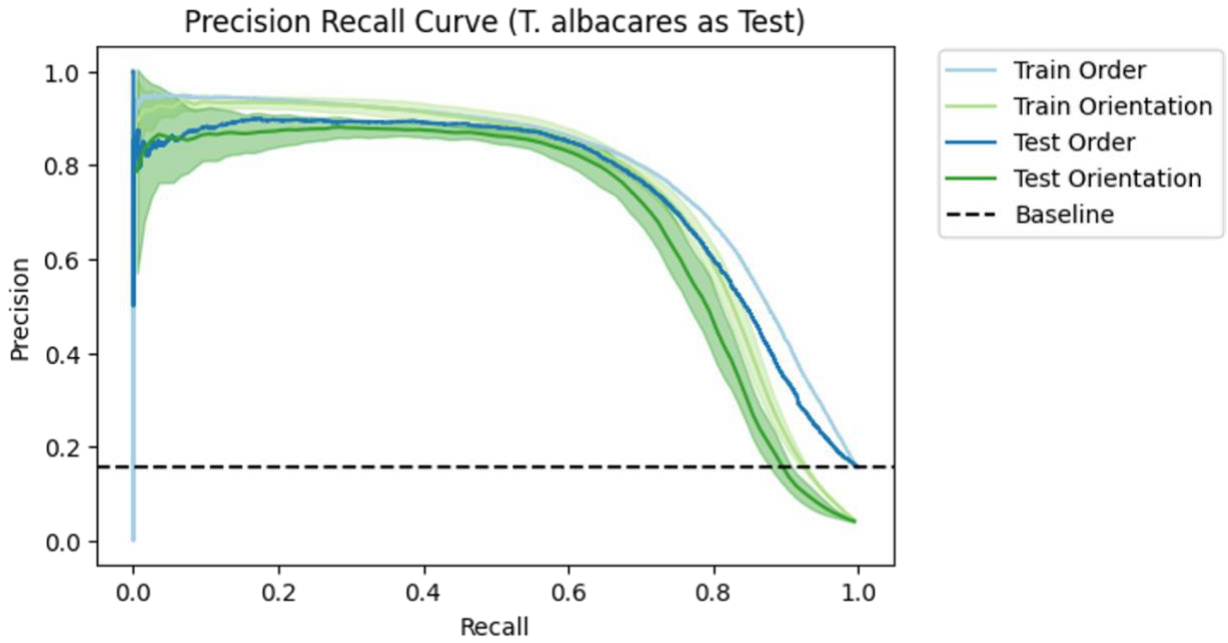

Figure D.29: The precision recall curve while holding *T. albacares* as the validation set.

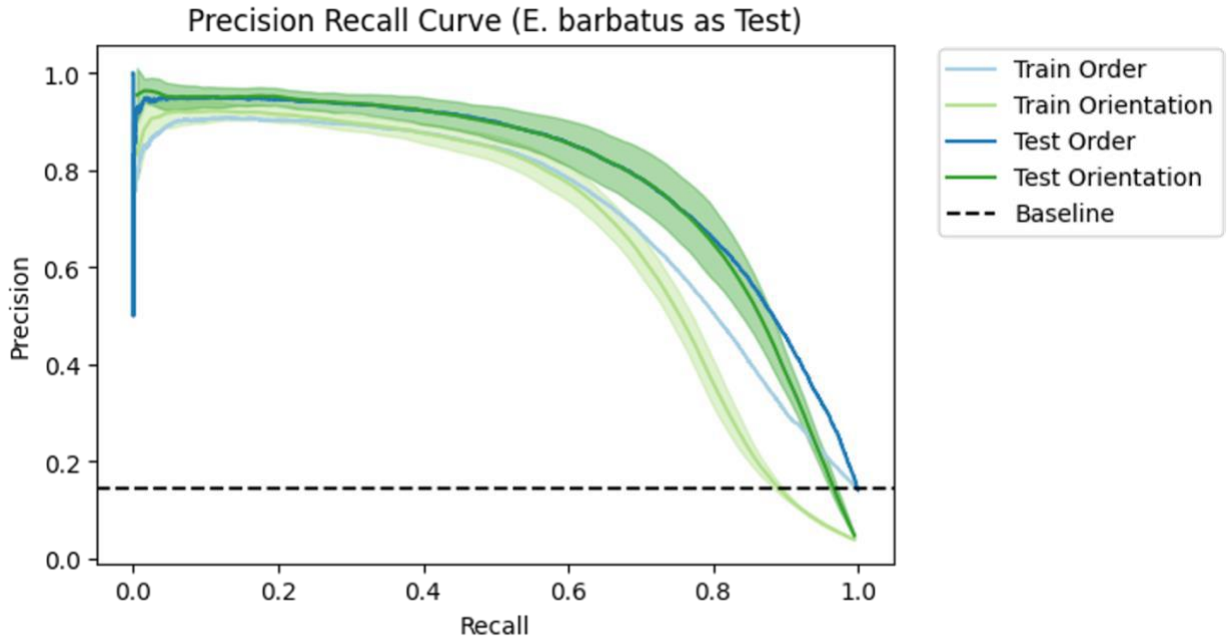

Figure D.30: The precision recall curve while holding *E. barbatus* as the validation set.

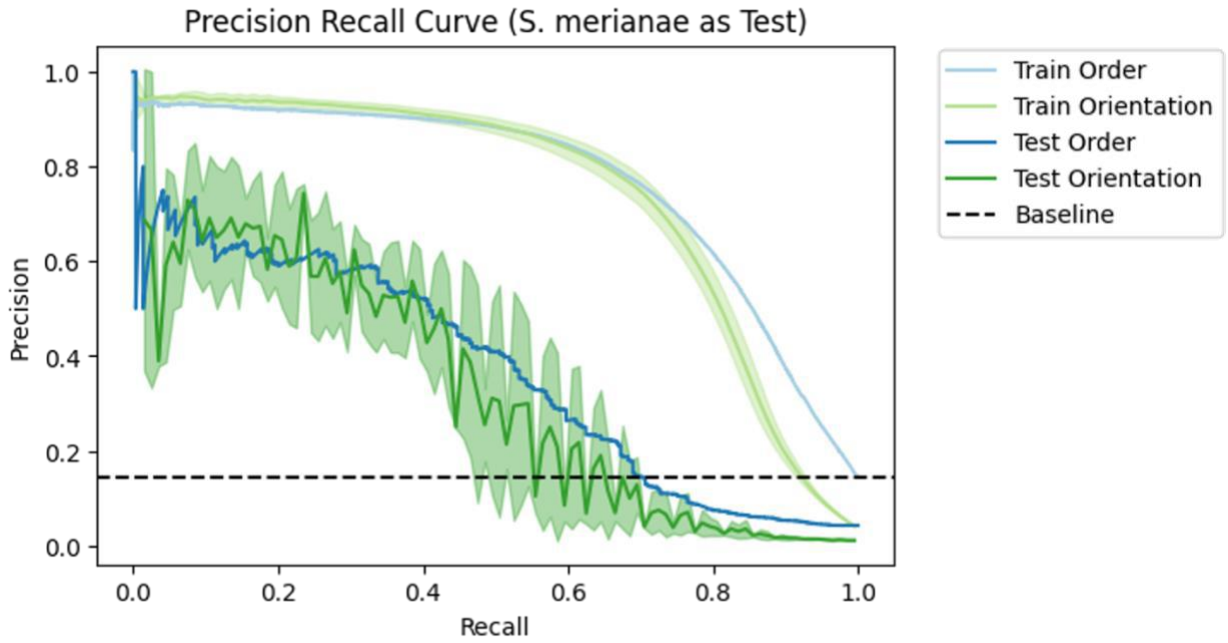

Figure D.31: The precision recall curve while holding *S. merianae* as the validation set.

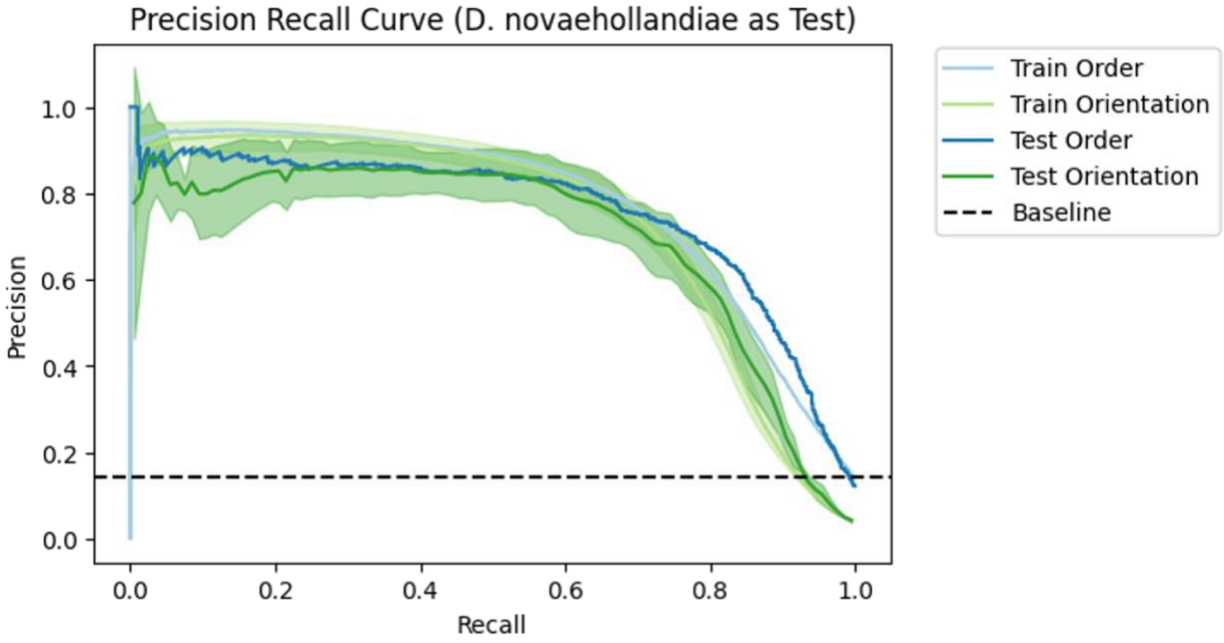

Figure D.32: The precision recall curve while holding *D. novaehollandiae* as the validation set.

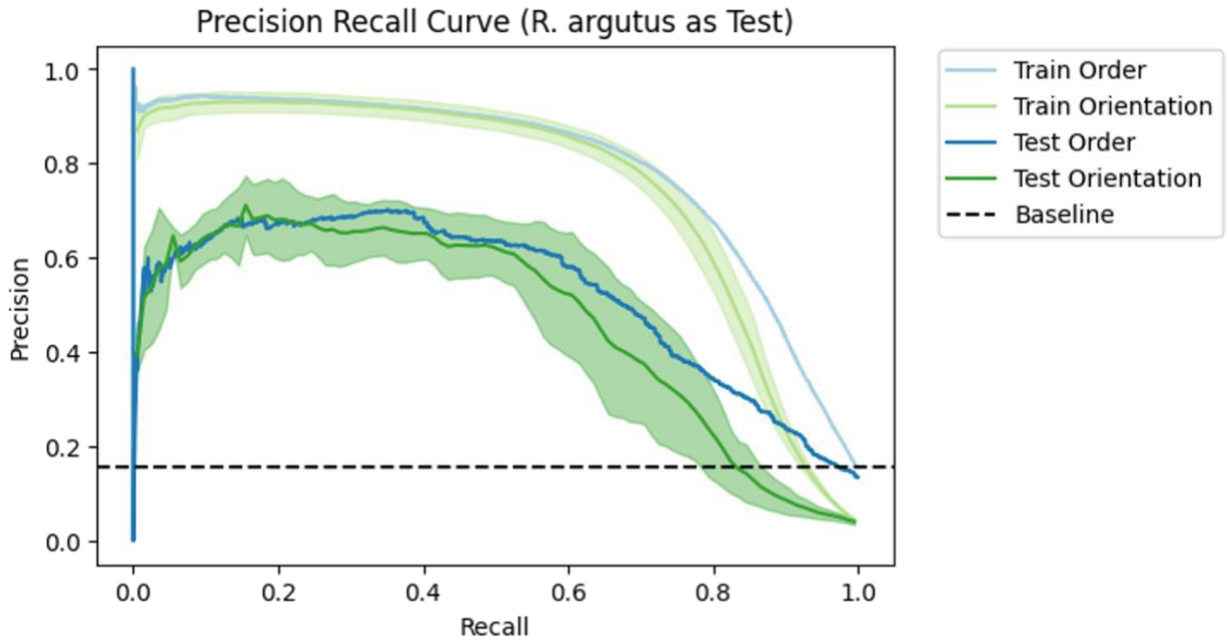

Figure D.9: The precision recall curve while holding *R. argutus* as the validation set.

VITA

Aakash Sur was born in Rochester, Minnesota and his family moved to various parts of the country, and even the world, throughout the years. Most of his primary education was in Houston, and after graduating high school Aakash attended the University of Texas at Austin and received a Bachelor of Sciences in Biochemistry and a Bachelor of Arts in Plan II. With a sense of restlessness and readiness to jump into the next chapter of life, he went straight from undergraduate to graduate school at the University of Washington, Seattle. This thesis represents the culmination of that graduate work. Outside of graduate school, he likes to climb, play frisbee, ski, game, cook, and spend time with his dog, Samira.