THE DEVELOPMENT AND APPLICATION OF COMPUTATIONAL METHODS FOR GENOME ANNOTATION

by
Alaina Shumate

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
March, 2022

Abstract

Improvements in DNA sequencing technology and computational methods have led to a substantial increase in the creation of high-quality genome assemblies of many species. To understand the biology of these genomes, annotation of gene features and other functional elements is essential; however, many genomes, especially eukaryotic genomes have not yet been annotated. *Ab-initio* gene prediction is notoriously hard in eukaryotic genomes due to the sparse gene content and introns interrupting genes. Two more-promising strategies for annotating eukaryotic genomes are RNA-sequencing followed by transcriptome assembly and/or mapping genes from a closely related species. Current transcriptome assembly methods can assemble either short or long RNA-sequencing reads, which each have their own weaknesses that limit assembly accuracy. Additionally, there are no standalone tools that can accurately map gene annotations from one assembly to another. Therefore, in this work we first present hybrid-read transcriptome assembly with StringTie where we combine long and short reads to mitigate the weaknesses of each datatype. We show that hybrid-read assembly achieves better accuracy than long or short-read only assembly on simulated as well as real RNA-sequencing data from human, *Mus musculus, and Arabidopsis thaliana*. We then introduce Liftoff, which is a standalone tool that can map gene annotations between assemblies of the same or closely related species. As a proof of concept, we map genes between two versions of the human reference genome and then between the human reference genome and the chimpanzee reference genome. We then describe the results of using Liftoff to annotate 3 new reference-quality human genome

assemblies and a new assembly of the bread wheat genome. Lastly, we introduce LiftoffTools, which is a toolkit that compares the sequence, synteny, and copy number of genes lifted from one assembly to another.

Committee: Steven Salzberg (Primary Advisor), Mihaela Pertea, Winston Timp

Acknowledgements

There are so many people in my professional life and personal life that have made this work possible. First and foremost, I would like to thank my advisor, Dr. Steven Salzberg, for taking a chance on me and providing me every opportunity to pursue my interests and the guidance to be successful. I am incredibly grateful to have been advised by someone who prioritizes intellectual curiosity and has such a high standard of scientific integrity. I am also incredibly grateful for the guidance from my other committee members Dr. Mihaela (Ela) Pertea and Dr. Winston Timp. Ela has guided me through so much of my research and has been a role model to me as a highly successful female scientist. Winston was the first professor to reach out to me from Hopkins and sparked my interest in the amazing genomics research going on here. I would also like to thank Dr. Jennifer Lu for all of the guidance she provided me on writing my thesis proposal and dissertation. Also thank to my undergraduate mentee Jakob Heinz, who was an important part of some of the work presented here and is the only other person I've met in Baltimore who understands my Wyoming references. To all of the Salzberg Lab members, thank you not only for your guidance and collaboration over the years but also your friendship. I feel incredibly lucky to call you all colleagues and friends. I would also like to thank my collaborators outside of our lab. Thank you to Dr. Michael Alonge for introducing me to the world of plant genomics. I never imagined I could find wheat so exciting. Thank you to Dr. Justin Zook and the rest of the Genome in a Bottle team for helping us improve our assemblies with your benchmarking resources. And lastly, thank you to Dr. Adam Phillippy, Dr. Karen Miga, Mark Diekhans, Marina Haukness, and the

rest of the Telomere-to-Telomere Consortium for allowing me to contribute to such a major milestone in human genomics.

The people in my life outside of research were just as important in helping me reach this milestone. To my Hopkins BME cohort, Pum Wiboonsaksakul, Neha Thomas, Mason Chen, Wangui Mbuguiro, Marlen Tagle Rodriguez, Savannah Est-Witte, and Dr. Julie Shade, thank you for always being up for taking a break from lab and exploring new restaurants and bars in Baltimore. Thank you to my college friends, Molly Gerrity, Sachie Weber, Johannah Brady, and Leslie Knueven for being a constant support of support and laughter despite being on the opposite side of the country. Our Sunday night zoom calls are always a highlight of my week. To my partner Andrianna Ayiotis, thank you for showing me what true love and partnership is. Even when the world is falling apart you still manage to make me laugh. I can't imagine doing this PhD without having you by my side every step of the way. Of course, I also have to acknowledge our cat Rocket (Rocky) Cat Ayiotis-Shumate who selflessly donated her eye to corneal transplant research in her former life as a Hopkins research animal. Lastly, I would like to thank my family who has loved and supported me my entire life. To my brother, Ben Shumate, thank you for being my first best friend and coming to every dance recital, concert, and graduation ceremony. To my parents Angela Ollila and Roger Shumate, thank you for teaching me the value of hard work, encouraging me to pursue my dreams, and most importantly, loving me unconditionally. I would not have made it here without you.

In memory of my dad

Roger Shumate

March 16th, 1958 – April 19th, 2010

Thank you for giving me a lifetime of love and wisdom in our short 15 years together.

I miss you everday.

Table of Contents

List of Tables

Chapter 2

Chapter 4

Chapter 6

List of Figures

A note about collaborations: Much of the work presented here has been conducted collaboratively with other scientists. While I try to focus on my individual contributions, some results and writing from my collaborators/mentors have been included for the sake of context and completeness. At the beginning of each chapter, I will list any results or text that were not generated directly by me and the names of those who contributed. I am incredibly grateful to the many wonderful scientists I have had the opportunity to work with throughout the course of my PhD.

Chapter 1: Introduction

The field of genomics seeks to answer 2 simple questions about the DNA of any organism. What is the sequence? And how does it function? The first question is answered with DNA sequencing and genome assembly, and the second is answered with genome annotation. Significant progress has been made in our ability to determine the sequence of a genome over the last 2 decades due to major improvements in DNA sequencing technology. In 2001, the cost of DNA sequencing was over 5,000 dollars per 1 million bases. The cost fell to approximately 15 dollars per million bases around the year 2008, when sequencing labs began switching from Sanger-based sequencing to next-generation sequencing. Today the cost is a mere 0.6 cents per million bases [1]. As genomic scientists love to point out, this steep decline in cost means DNA sequencing has outpaced Moore's Law – the observation in the computer hardware industry that computing power doubles every 2 years. Consequently, sequencing and assembling genomes, even large eukaryotic genomes, is routinely completed from start

to finish by individual labs. This dramatically improved ability to sequence genomes is evident in NCBI where the number of eukaryotic assemblies has increased from just 3 in 2001 to 23,680 in early 2022. While this is without a doubt an impressive accomplishment for the field of genomics, the sequence of a genome by itself is of little use when trying to understand the biology of an organism. For this, it is imperative to have complete and accurate annotation of the structure and function of genes and other genomic elements. Unfortunately, our ability to annotate eukaryotic genomes has not kept pace with sequencing and assembly, as shown in Figure 1.1.

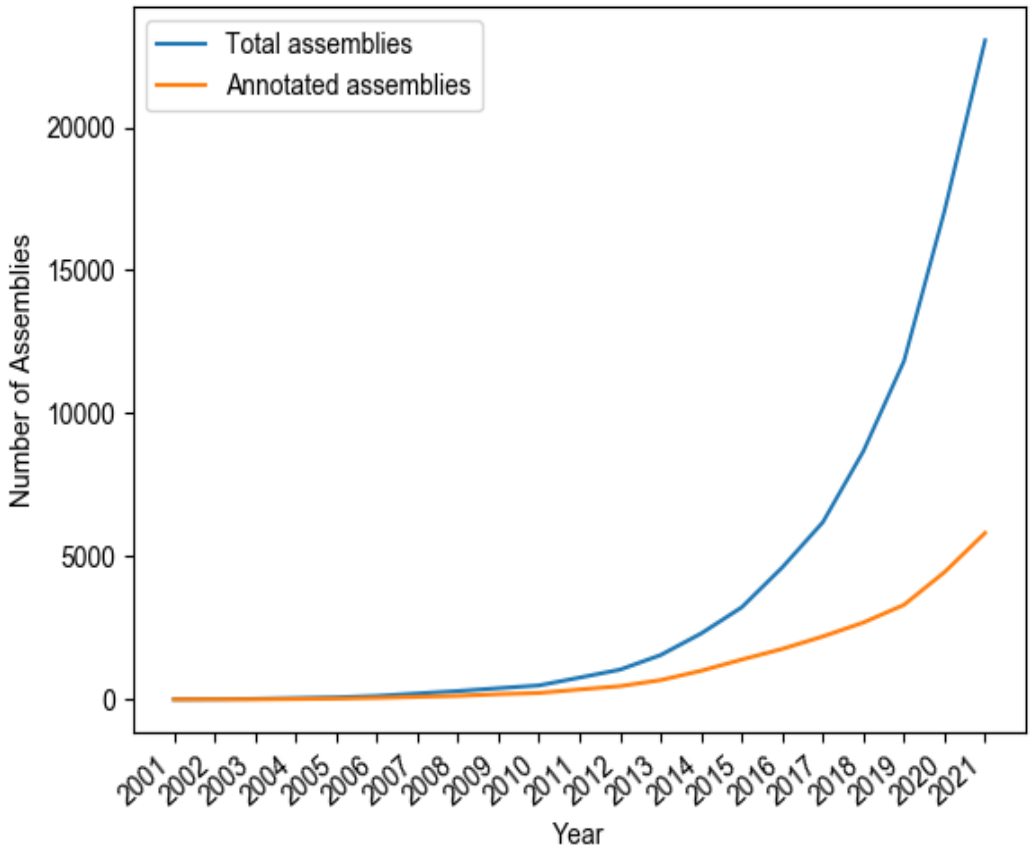

Figure 1.1. Number of genome assemblies in NCBI over time. The total number of eukaryotic genome assemblies in NCBI from 2001 to 2021 (blue) and the number of those assemblies which have annotation (orange).

There are many computational methods that try to automate gene-finding in eukaryotes[2–5] which generally rely on 3 types of information including *ab-initio* predictions, experimental transcriptomic evidence (RNA-sequencing), and homology-based information from gene models of closely-related species[6]. *Ab-initio* prediction in eukaryotes is notoriously difficult because unlike in prokaryotes, genes are interrupted by introns, and genes make up very little of the total genomic sequence[7]. Transcriptomic and homology-based evidence are both more promising but come with their own challenges. Short-read RNA-sequencing produces reads that can rarely span more than 1 exon, and long-read RNA-sequencing has a high error rate. This makes it challenging to accurately assemble transcripts which is essential for gene annotation. Homology-based evidence is generally derived from either local alignments of gene sequences or whole genome alignments; however, there are no standalone tools that can take these alignments and automatically produce a complete and accurate annotation. The focus of this work is the development and application of computational methods which improve gene annotation using RNA-sequencing and homology-based methods. First, we introduce a new release of StringTie[8] and show that it improves transcriptome assembly by using both long and short RNA-sequencing reads. Next, we introduce Liftoff [9] which is a standalone tool that maps gene annotations from a well-annotated reference genome to a target genome of the same or closely related species. We then show the results of using Liftoff to annotate 3 new human genome assemblies[10–12] and a new bread wheat assembly[13]. Finally, we describe LiftoffTools, which is a toolkit to compare genes mapped between different genome assemblies.

Chapter 2: Hybrid-read transcriptome assembly with StringTie

A version of chapter 2 has appeared in:

A. Shumate, B. Wong, G. Pertea, M. Pertea (2021). “Improved Transcriptome Assembly Using a Hybrid of Long and Short Reads with StringTie” *bioRxiv* (submitted for publication).

Additional Contributors:

This work was conducted jointly with the authors listed above. Mihaela Pertea designed and implemented the hybrid read StringTie algorithm and wrote the section ‘StringTie algorithm for Hybrid Data’. Geo Pertea assisted in the development and implementation of the algorithm and created Figure 2.2. Brandon Wong aligned and assembled RNA-seq data, conducted the coverage analysis, and created Figure 2.4.

2.1 Introduction

Uncovering the transcriptome of an organism is crucial to understanding the functional elements of the genome. This requires being able to accurately identify transcript structure and quantify transcript expression levels. In eukaryotes, this task is more challenging due to alternative splicing, which occurs frequently with an estimated 92%-94% of human genes undergoing alternative splicing[14]. Short-read RNA-sequencing (RNA-seq) has been a useful tool in uncovering the transcriptome of many organisms

when coupled with computational methods for transcriptome assembly and abundance estimation. Short-read sequencing provides the advantage of deep coverage and highly accurate reads. Second-generation sequencers such as those from Illumina can produce millions of reads with an error rate of less than 1%[15]. While second-generation sequencers produce very large numbers of reads, their read lengths are typically quite short, in the range of 75-125 bp for most RNA-seq experiments today. These short reads often align to more than one location in the genome, and suffer the limitation that they rarely span more than two exons, resulting in a difficult and sometimes impossible task of constructing an accurate assembly of genes with multiple exons and many diverse isoforms, no matter how deeply those genes are sequenced. These issues can be alleviated by third-generation sequencing technologies such as those from Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT). Reads from these technologies can be greater than 10 kilobases long, allowing full-length transcripts to be sequenced. However, practical limitations often impede the ability to capture full-length transcripts. These include the rapid rate of RNA degradation, shearing of the RNA during library preparation, or incomplete synthesis of cDNA[16]. Additionally, long reads have a high error rate relative to Illumina short reads, and the throughput of long-read RNA-seq is much lower than that of short-read RNA-seq. This can make it difficult in some cases to define precise splice sites. Using a combination of short reads and long reads for transcriptome assembly allows us to take advantage of the strengths of each technology and mitigate the weaknesses. While there are many tools that use either short reads or long reads for transcriptome assembly and quantification, there are very few that use a hybrid of the two. These tools include Trinity[17], IDP-denovo[18], and

rnaSPAdes[19], which only perform *de novo* transcriptome assembly. If a high-quality reference genome of the target organism is available, as it is for human and for a large number of plants, animals, and other species, *de novo* transcriptome assembly usually produces lower-quality assemblies compared to reference-based approaches. This is due to technical challenges resulting from the presence of gene families, large variations in gene expression, and extensive alternative splicing[20]. StringTie is a reference-based transcriptome assembler that can assemble either long reads or short reads and has been shown to be more accurate than existing short and long read assemblers[21]. In this work we present a new release of StringTie which allows transcriptome assembly and quantification using a hybrid dataset containing both short and long reads. We show with simulated data from the human transcriptome that hybrid-read assemblies result in more accurate assembly and coverage estimates than using long reads or short reads alone. Additionally, we evaluate the assembly accuracy on 9 real datasets from 3 well-studied species (human, *Mus musculus*, and *Arabidopsis thaliana*) and demonstrate that the hybrid-read assemblies are more accurate than both the long-read only and short-read only assemblies. We also demonstrate that hybrid-read assembly is more accurate and substantially faster than a strategy of correcting long reads prior to assembly.

2.2 Results

Our hybrid transcriptome assembly algorithm takes advantage of the strengths of both long and short read RNA sequencing, by combining the capacity of long reads to capture longer portions of transcripts with the high accuracy and coverage of short-read

data to produce better transcript structures as well as better expression estimates. Figure 2.1 shows examples of alignment artifacts that are often present in long reads because of the high error rate. These include “fuzzy” splice sites as well as retained introns, spurious extra exons, falsely skipped exons, and false alternative splice sites.

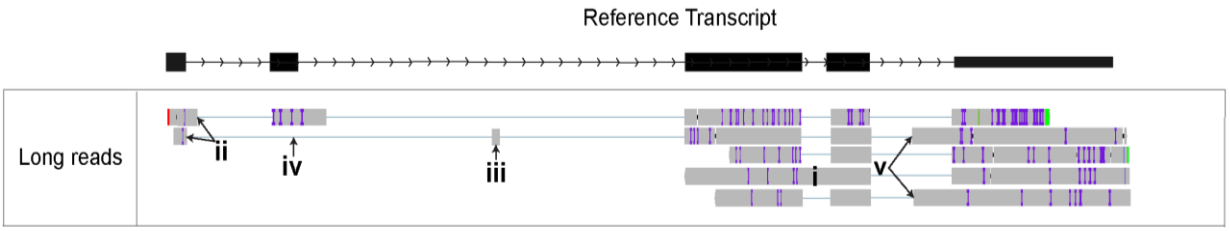

Figure 2.1. Examples of alignment artifacts in long reads. Artifacts present the in long read alignments: i) retained introns; ii) disagreement around the splice sites; iii) spurious extra exons; iv) falsely skipped exons; v) false alternative splice sites.

Figure 2.2 shows a specific example of a 9-exon isoform of a human gene that can only be correctly assembled using both long and short reads. There are no long reads mapped to the first 3 exons of this isoform, and we see a retained intron in the alignment. The short reads do not have any reads that span more than two exons, and for 2 splice junctions, there is only one spliced read spanning the junction. In both cases, the read does not fully cover both exons and consequently the transcript is assembled in three fragments. Using both long and short reads we were able to correctly assemble the transcript by using the short reads to support the splice sites found in the long-read alignments (See Methods). The adequate coverage of the short reads mapped to exons 1-3 also allow us to capture these in the assembly despite the lack of coverage in the long reads.

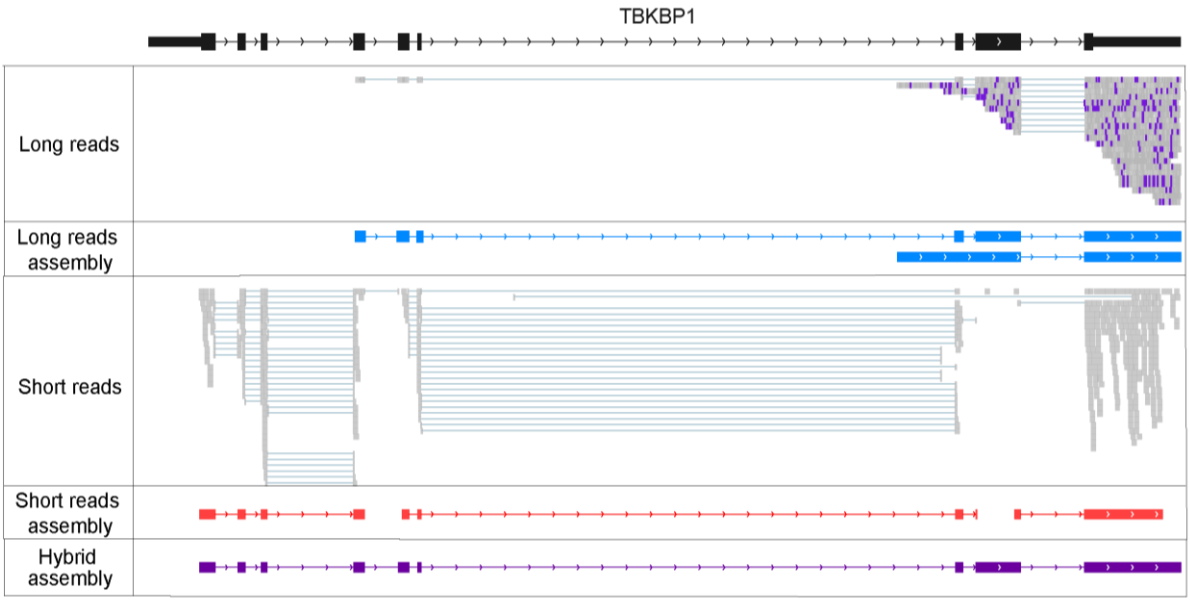

Figure 2.2. Hybrid-read transcript example. Example of a human transcript that can only be correctly assembled using both the long and short reads. This is transcript ENST000000361722.7 from the TBKBP1 gene. Blue lines in the middle of the reads (gray boxes) indicate a splice alignment. The long reads do not have coverage of exons 1-3 and the short reads lack adequate splice-site support across the 1st intron and the 7th intron.

Next, we present results for StringTie's performance with hybrid, long, and short-read sequences on simulated data as well as on three real RNA-seq data sets, from human, mouse, and the model plant *Arabidopsis thaliana*.

Simulated Data

Since it is not possible to know the true transcripts that are present in real RNA-seq datasets, we first used simulated data to assess the accuracy of hybrid-read assembly and quantification across the transcriptome. To this end, we simulated two human RNA-seq datasets, one with short-reads and one with ONT direct RNA long reads (see Methods) and assembled them with StringTie.

To evaluate the accuracy of hybrid-read assemblies compared to long-read only and short-read only assemblies, we generated 4 different assemblies of each read type (long, short, and hybrid) with 4 different sets of parameters (Figure 2.3). We then computed the precision and sensitivity for each assembly. Precision is defined as the percent of assembled transcripts that match true transcripts, and sensitivity is defined as the percent of true transcripts that match an assembled transcript (see Methods). For these calculations, we considered a transcript to be truly expressed only if it was fully covered by either the short or long simulated reads. For each hybrid-read assembly, we calculated the relative percent increase in precision and sensitivity over the long-read and short-read assemblies with the same parameters (see Methods). When we report the percent increase of any metric, we are referring to the *relative* percent increase. Averaging these results, we saw that hybrid-read assemblies had an increase in precision of 9.8% over the long-read assemblies, and an increase in sensitivity of 24.4%. As compared to the short-read assemblies, the hybrid-read assemblies had an increase in precision of 12.5% and an increase in sensitivity of 22.1%.

We also compared the coverage computed by StringTie to the actual coverage of long-read only, short-read only, and hybrid-read assemblies created with default parameters (Figure 2.4). StringTie's computed coverage of the hybrid-read assembly was closest to the true coverage. We found that the correlation between true and calculated coverage for hybrid-read assembly yielded an R^2 value of 0.966, higher than the R^2 values for both the short-read (0.959) and long-read (0.933) only assemblies.

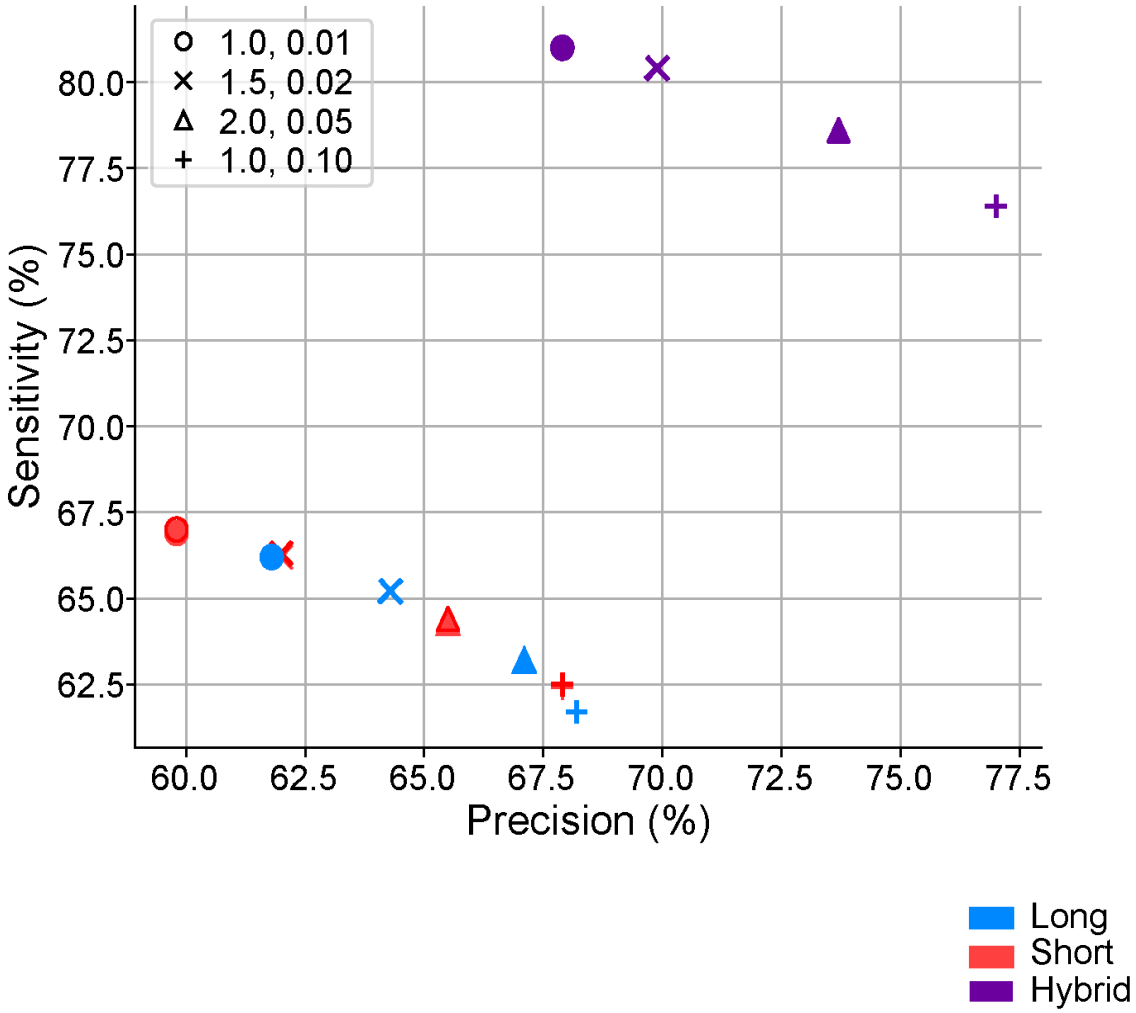

Figure 2.3. Simulated data accuracy. Sensitivity and precision for StringTie assemblies of simulated data with varying sensitivity parameters. The two StringTie parameters varied were the minimum read coverage allowed for a transcript (-c) and the minimum isoform abundance as a fraction of the most abundant transcript at a given locus (-f). Each shape represents a different combination of -c,-f parameters with the values indicated in the legend. The default values of -c and -f are 1.0 and 0.01 respectively and are represented by the circle marker.

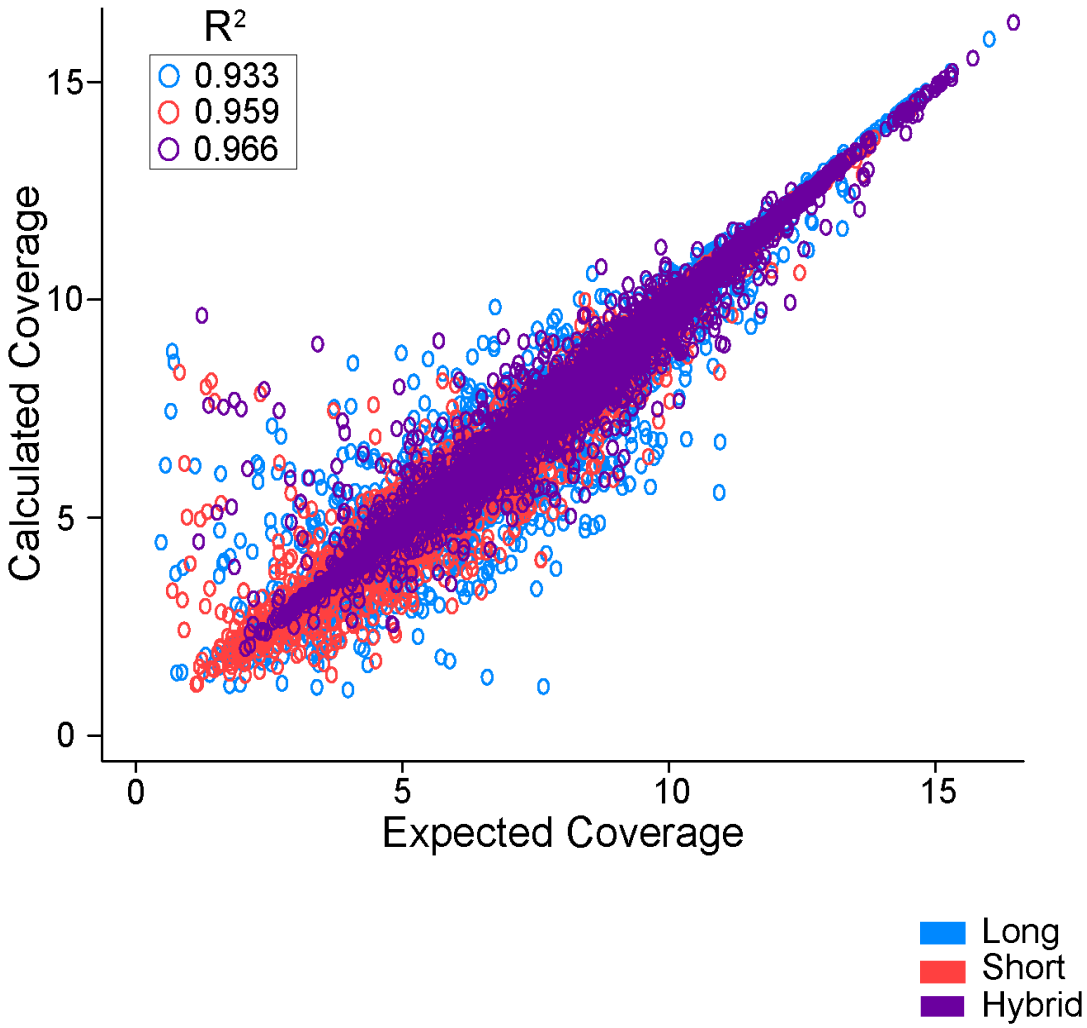

Figure 2.4. Transcript coverage correlation. Calculated coverage vs. expected coverage for long-read, short-read, and hybrid-read assemblies of simulated data.

If the reference genome annotation is reliable, some methods (including StringTie) can use that annotation to improve the accuracy of the transcriptome assembly. Note that not all transcripts in the reference annotation will be expressed in the data, therefore the assembler needs to accurately determine which of the transcripts are present in the data. Moreover, reference annotations are usually incomplete, so StringTie's default behavior when annotation is provided is to assume that novel transcripts could be

present as well. We wanted to assess if StringTie's performance improves on hybrid data if the human reference annotation is provided. As shown in Figures 2.5 A and 2.5 B, both precision and sensitivity improved when the reference annotation was provided, and hybrid data assembly had the highest sensitivity and precision regardless of whether the reference annotation was provided or not. The use of hybrid-read data plus annotation had an increase in precision of 10.7% and an increase in sensitivity of 23.5% as compared to using short reads plus annotation, which in turn was superior to using long reads plus annotation.

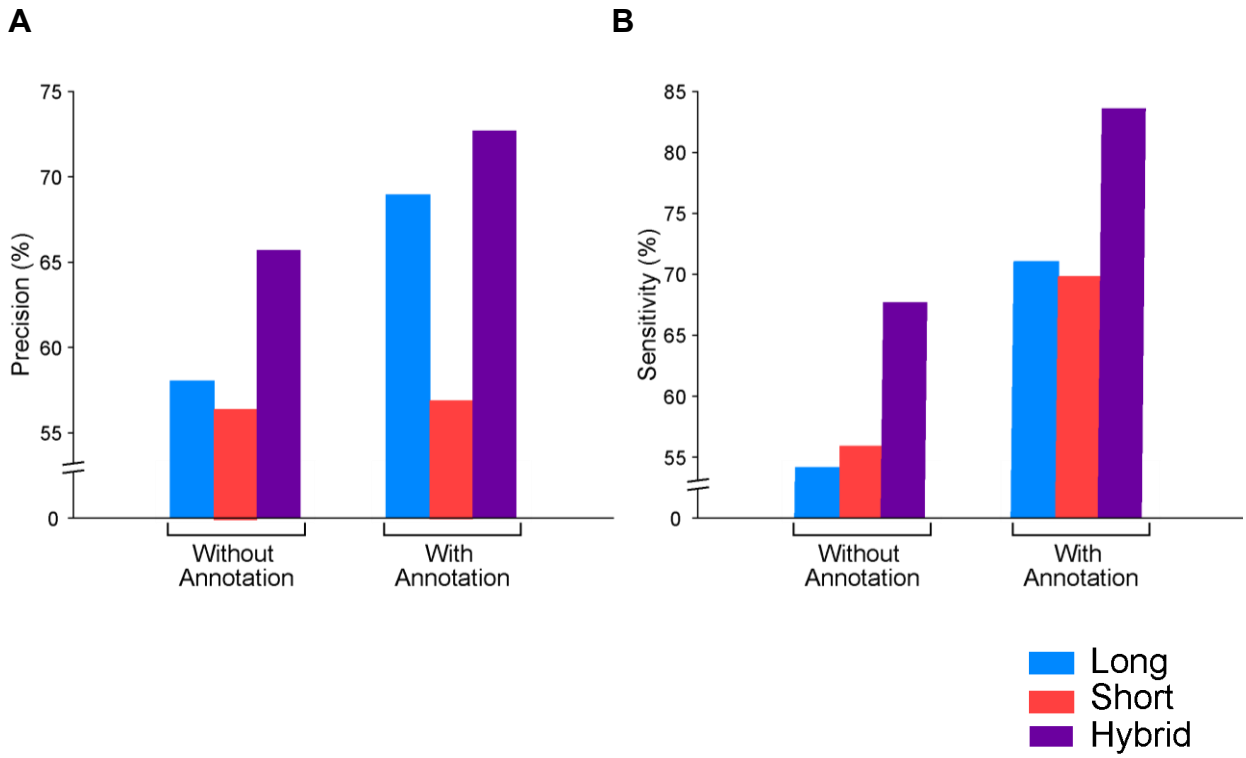

Figure 2.5. Accuracy of reference-guided assemblies of simulated data. A) Precision of long-read, short-read, and hybrid-read assemblies of simulated data with and without guide annotation. **B)** Sensitivity of long-read, short-read, and hybrid-read assemblies of simulated data with and without guide annotation

An important observation is that the coverage of the hybrid reads is the sum of the coverage of the short and long reads. Here we define coverage simply as the sum of the read lengths divided by the sum of the expressed transcript lengths. To confirm that the improvements we saw in the hybrid-read assemblies were not simply due to having deeper coverage, we conducted another experiment where we simulated additional short reads and long reads such that the coverage of each dataset approximately matched the coverage of the hybrid-read dataset (See Methods). We then repeated the same analysis as in Figure 2.3 where we created 4 assemblies of each read set with varying parameters and computed the accuracy. To effectively compare the results of assembling the new simulated dataset (equal coverage) to the assemblies of the original simulated dataset (unequal coverage), we computed accuracy of both sets using the full set of expressed transcripts as the reference. Unlike in Figure 2.3, we did not filter the reference transcripts for those transcripts fully covered by long or short reads as this set is not the same for both simulations. The results for the unequal coverage assemblies are shown in Figure 2.6 A and the results for the equal coverage assemblies are shown in Figure2.6 B. From these results, it is clear that the improvement seen with hybrid-read assembly is not simply due to the increased coverage.

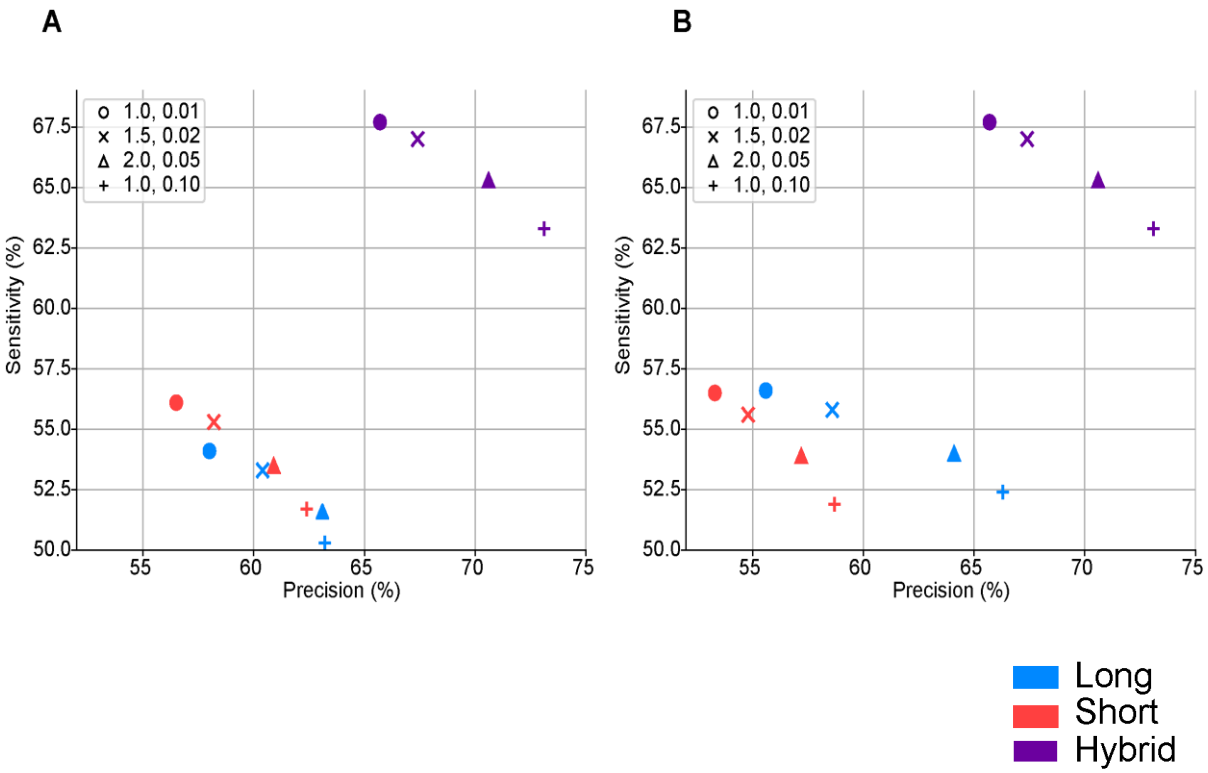

Figure 2.6. Accuracy of simulated data with equal coverage. Sensitivity and precision for StringTie assemblies of simulated data with varying sensitivity parameters. The two StringTie parameters varied were the minimum read coverage allowed for a transcript (-c) and the minimum isoform abundance as a fraction of the most abundant transcript at a given locus (-f). Each shape represents a different combination of -c,-f parameters with the values indicated in the legend. **A)** Sensitivity and precision of the assemblies created from the original dataset where the hybrid read coverage is the sum of the long read and the short read coverage**. B)** Sensitivity and precision of the assemblies created from the dataset where the coverage of the short, long, and hybrid reads is approximately equal.

Real Data

Next, we evaluated the accuracy of hybrid-read assemblies on real data, which is in general much more challenging than simulated data, in part because the real data may contain biases or other artifacts not always captured by simulated data. From publicly available data, we chose a total of 9 combinations of long and short reads from 3 well-

studied species: *Arabidopsis thaliana*, *Mus musculus*, and human. Each combination of long and short reads is derived from the same sample. All three species have well-characterized reference annotation available, even though their level of completeness is not fully established [22]. The short read libraries were all generated through poly-A selection and sequenced with Illumina sequencers. The long reads were generated by a variety of technologies including ONT direct RNA, ONT cDNA, and PacBio cDNA (Table 2.1). The quality of these long reads varies with error rates ranging from 3.2% to 17.2% (Table 2.2) and the percentage of full-length isoforms sequenced ranging from 25.4% to 67.2% (Table 2.3).

Table 2.1. RNA-seq data summary. Availability of real RNA-seq datasets and descriptions of sequencing technology used including chemistry and base-caller version for ONT datasets.

Accession Number	Database	Species	Sequencing Technology
ERR3486096	European Nucleotide Archive	*A. thaliana*	Illumina HiSeq 4000
ERR3764345	European Nucleotide Archive	*A. thaliana*	ONT direct RNA SQK-RNA001 MinION Guppy v2.3.1
ERR3486098	European Nucleotide Archive	*A. thaliana*	Ilumina HiSeq 4000
ERR3764349	European Nucleotide Archive	*A. thaliana*	ONT direct RNA SQK-RNA001 MinION Guppy v2.3.1
ERR3486099	European Nucleotide Archive	*A. thaliana*	Ilumina HiSeq 4000
ERR3764351	European Nucleotide Archive	*A. thaliana*	ONT direct RNA SQK-RNA001 MinION Guppy v2.3.1

ERR2680378	European Nucleotide Archive	*M. musculus*	Illumina HiSeq 4000
ERR2680375	European Nucleotide Archive	*M. musculus*	ONT direct RNA SQK-RNA001 MinION Albacore 2.1.10
ERR2680377	European Nucleotide Archive	*M. musculus*	ONT cDNA MinION SQK-PCS108 Albacore 2.1.10
ERR2680380	European Nucleotide Archive	*M. musculus*	Illumina HiSeq 4000
ERR2680379	European Nucleotide Archive	*M. musculus*	ONT direct RNA SQK-RNA001 MinION Albacore 2.1.10
SRR4235527	Sequence Read Archive	*H. sapiens*	Illumina Genome Analyzer IIx
NA12878-dRNA	github.com/nanopore-wgs-consortium	*H. sapiens*	ONT direct RNA SQK-RNA001 MinION Guppy v3.2.6
NA12878-cDNA	github.com/nanopore-wgs-consortium	*H. sapiens*	ONT cDNA SQK-PCS108 MinION Albacore 2.1
SRR1153470	Sequence Read Archive	*H. sapiens*	Illumina HiSeq 2000
SRR1163655	Sequence Read Archive	*H. sapiens*	PacBio cDNA PacBio RS

Table 2.2. Long read error rates. Error rates of all long read datasets before and after correction with TALC.

Sample	Species	Sequencing Type	Error Rate Before Correction (%)	Error Rate After Correction (%)
ERR2680375	*M. musculus*	ONT dRNA	17.2	4.7
ERR2680377	*M. musculus*	ONT cDNA	14.3	4.8
ERR2680379	*M. musculus*	ONT dRNA	15.8	4.7
ERR3764345	*A. thaliana*	ONT dRNA	15.7	5.1
ERR3764349	*A. thaliana*	ONT dRNA	16.2	4.3
ERR3764351	*A. thaliana*	ONT dRNA	16.2	4.1
NA12878-cDNA	Human	ONT cDNA	15.5	6.4
NA12878-DirectRNA	Human	ONT dRNA	10.8	3.1
SRR1163655	Human	PacBio cDNA	3.2	1.8
Simulated-dRNA	Human	Simulated ONT dRNA	10.4	N/A

Table 2.3. Full-length isoforms in long-read data. The percentage of reads that are full-length isoforms and the number of unique full-length isoforms captured in each long-read dataset. We define a full-length isoform as a read that spans all exon/intron boundaries of a multi-exon transcript or a read that spans at least 80% of a single-exon transcript.

Sample	Species	Sequencing Type	% Full-length Isoforms	Number of Unique Full-length Isoforms
ERR2680375	*M. musculus*	ONT dRNA	25.4	22882
ERR2680377	*M. musculus*	ONT cDNA	36.2	34441
ERR2680379	*M. musculus*	ONT dRNA	40.0	14824
ERR3764345	*A. thaliana*	ONT dRNA	60.6	31479
ERR3764349	*A. thaliana*	ONT dRNA	66.5	33132
ERR3764351	*A. thaliana*	ONT dRNA	67.2	29618
NA12878-cDNA	Human	ONT cDNA	51.1	40368
NA12878-DirectRNA	Human	ONT dRNA	47.3	57563
SRR1163655	Human	PacBio cDNA	42.8	67900
Simulated-dRNA	Human	Simulated ONT dRNA	35.1	32324

Although we cannot know exactly which transcripts are present in the samples, it can be assumed that an assembly with more transcripts matching known annotations is more sensitive, and an assembly is more precise if known transcripts comprise a higher percentage of the total number of assembled transcripts. Therefore, to evaluate the

accuracy of the assemblies of real data, we report two values: (1) the number of assembled transcripts matching an annotated transcript, and (2) precision, which we define as the percentage of assembled transcripts matching known annotations. We chose to report the number of transcripts matching the annotation instead of sensitivity, because it is impossible to know exactly which transcripts are truly expressed in real experimental data. As with the simulated data, we report the relative percent increase/decrease of both metrics. Since short-read data offers much higher coverage of the expressed transcriptome, for these calculations we only consider loci with long-read coverage.

We also compare hybrid-read assembly to the strategy of correcting long reads prior to assembly, which is a common approach to handling the high error rate of long reads. Multiple previous algorithms have been proposed to combine long and short reads into high-accuracy long reads [23], but those approaches were primarily intended to be applied to whole-genome data with the aim to improve the quality of genome assemblies. Only recently a new method, called TALC [24], was developed for long-read correction in the context of RNA-seq data by incorporating coverage analysis throughout the correction process. Using the corresponding short-read sample, we corrected each long-read sample with TALC. On average TALC decreased the error rate by 9.5% (Table 2.2). We created additional long-read and hybrid-read assemblies with the TALC-corrected reads and then compared the accuracy of the hybrid-read assemblies to the corrected long-read assemblies. We also assessed whether using corrected long reads in a hybrid-read assembly substantially improved the accuracy. As

we show below, TALC is quite effective at correcting errors; however, it is far slower than StringTie (running on a single RNA-seq samples takes TALC a day or longer, compared to less than one hour for StringTie), and it does not improve transcript assembly as compared to our new hybrid assembly algorithm.

Arabidopsis thaliana

The hybrid-read assemblies of the *Arabidopsis thaliana* samples achieved higher precision and contained more annotated transcripts than both the long-read and short-read assemblies (Figure 2.7 A-C). The average percent increase in precision in the hybrid-read assemblies was 8.0% over the long-read assemblies, and 4.1% over the short-read assemblies. The increase in the number of annotated transcripts was 21.7% and 5.0% over the long-read and short-read assemblies respectively. When comparing the results of hybrid-read assembly to an assembly of corrected long reads, the hybrid-read assembly had a very small decrease in precision of 1.0%, but an increase in the number of annotated transcripts of 14.4%. Finally, using the TALC-corrected long reads instead of the uncorrected long reads in a hybrid-read assembly only increased precision by 0.5% and increased the number of annotated transcripts by 0.4%.

Mus musculus

In the *Mus musculus* samples, the hybrid-read assemblies showed an even greater improvement in precision versus the long-read only and short-read only assemblies (Figure 2.7 D-F). The percent increase was 38.6% over the long-read assemblies and 18.9% over the short-read assemblies. The number of annotated transcripts assembled

increased substantially over the long-read assemblies with a relative increase of 118%; however, there was a slight decrease over the short-read assemblies of 0.6%. As with *Arabidopsis thaliana*, we saw that the hybrid-read assemblies outperform the corrected long-read assemblies with a 24.3% increase in precision and a 96.0% increase in the number of annotated transcripts. Hybrid-read assemblies using the TALC-corrected long reads again did not appear considerably different than the hybrid-read assemblies with the uncorrected reads: precision decreased by 0.5% while the number of annotated transcripts increased by 0.8%.

Human

In the human data, we saw an increase in precision of 26.0% in the hybrid-read assemblies over the long-read assemblies, and an increase of 22.7% over the short-read assemblies (Figure 2.7 G-I). The number of annotated transcripts was also higher in the hybrid-read assemblies with an increase of 47.2% over the long-read assemblies and 36.5% over the short-read assemblies. As with the *Arabidopsis thaliana* and *Mus musculus* samples, the hybrid-read assemblies were still better than corrected long-read assemblies with 21.4% greater precision and 45.0% more annotated transcripts. The increase in precision and number of annotated transcripts in the hybrid-read assembly with corrected long reads compared to hybrid-read assembly with the uncorrected reads was again small, at 1.1% and 1.0% respectively. Because the human genome is the largest of the 3 genomes, we also compared the runtime of hybrid-read assembly to that of TALC. On average, hybrid-read assembly of the human samples took 48.8 minutes

using 1 thread. In comparison, TALC took an average of 7143 minutes using 12 threads.

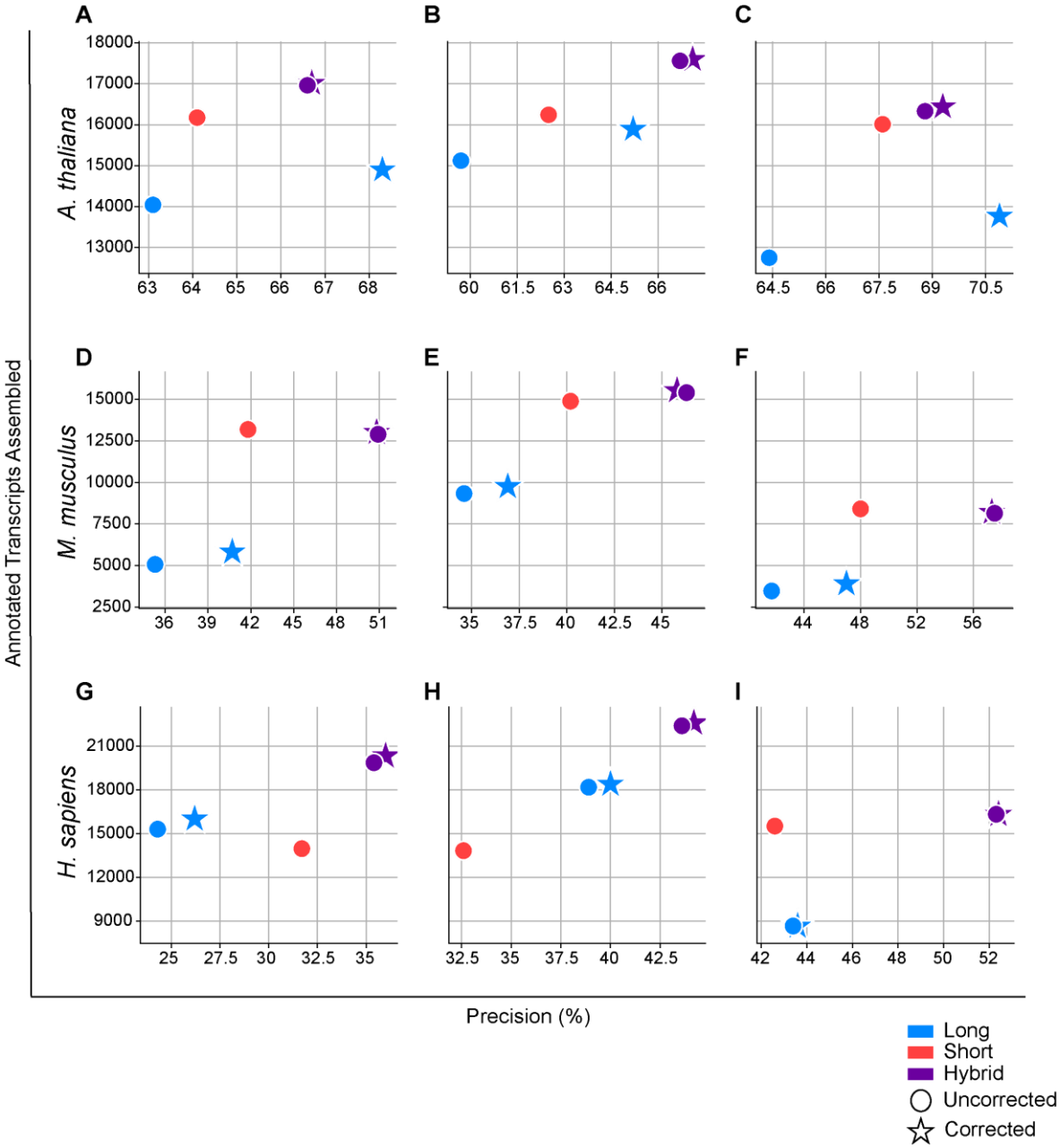

Figure 2.7. Accuracy of real data assemblies at loci with long-read expression. Sensitivity and the number of annotated transcripts assembled for 9 real datasets from *Arabidopsis thaliana*, *Mus musculus*, and human. Only loci with long read expression are considered for these calculations. The circle markers represent assemblies created from uncorrected reads, and the stars represent assemblies created from long-reads corrected with TALC. The long and short read combinations analyzed from *Arabidopsis thaliana* were **A)** ERR3486096 and ERR3764345 **B)** ERR3486098 and ERR3764349 **C)** ERR3486099 and ERR3764351. The long and short read combinations analyzed from

Mus musculus were **D)** ERR2680378 and ERR2680375 **E)** ERR2680378 and ERR2680377 **F)** ERR2680380 and ERR2680379. The long and short read combinations analyzed from human were **G)** SRR4235527 and NA12878-cDNA **H)** SRR4235527 and NA12878-dRNA **I)** SRR1153470 and SRR1163655.

While examining the accuracy at only loci with long-read coverage provides the fairest comparison between long, short, and hybrid-read assemblies, in practice it may be useful to know the outcome of using all of the data. We ran the same analysis considering all loci (Figure 2.8), and we observe similar trends where hybrid-read assemblies are superior when considering both precision and the number of annotated transcripts assembled.

Annotation-Guided Assembly

As with the simulated data, we also performed annotation-guided assembly for each species and evaluated the precision (Figure 2.9 A) and number of annotated transcripts assembled (Figure 2.9 B) considering only loci with long-read expression. We compared these results to the hybrid-read assemblies created without guide annotation. The average precision of the *Arabidopsis thaliana* hybrid-read assemblies increased from 67.3% to 80.2%, and the average number of annotated transcripts assembled increased from 16,952 to 26,214. The average precision of the *Mus musculus* hybrid-read assemblies increased from 51.6% to 80.6% and the number of annotated transcripts increased from 12,150 to 34,809. Lastly the precision of the human assemblies increased from 43.8% to 75.8% and the number of annotated transcripts assembled more than doubled, increasing from 19,543 to 40,903. Across all samples in all species, the annotation-guided hybrid-read assemblies had greater precision than the annotation-guided long and short read assemblies. In all *Mus musculus* and human

samples, the hybrid-read assemblies also contain a greater number of annotated transcripts. The *Arabidopsis thaliana* assemblies contain more annotated transcripts than the long-read assemblies, but slightly fewer than the short-read assemblies.

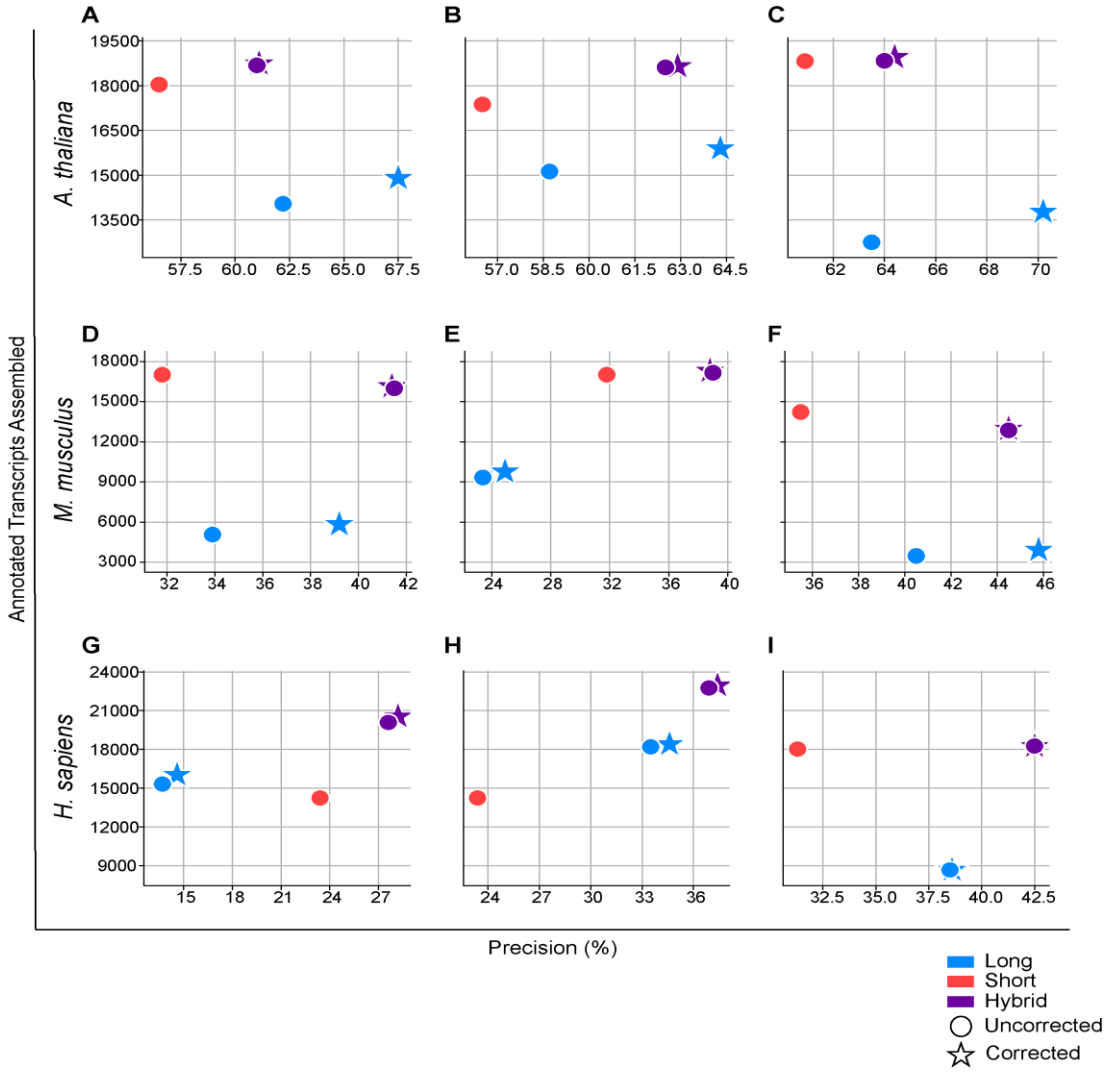

Figure 2.8. Accuracy of real data assemblies at all loci. Sensitivity and the number of annotated transcripts assembled for 9 real datasets from *Arabidopsis thaliana*, *Mus musculus*, and human. The circle markers represent assemblies created from uncorrected reads, and the stars represent assemblies created from long-reads corrected with TALC. The long and short read combinations analyzed from *Arabidopsis thaliana* were **A)** ERR3486096 and ERR3764345 **B)** ERR3486098 and ERR3764349 **C)** ERR3486099 and ERR3764351. The long and short read combinations analyzed from *Mus musculus* were **D)** ERR2680378 and ERR2680375 **E)** ERR2680378 and ERR2680377 **F)** ERR2680380 and ERR2680379. The long and

short read combinations analyzed from human were **G)** SRR4235527 and NA12878-cDNA **H)** SRR4235527 and NA12878-dRNA **I)** SRR1153470 and SRR1163655.

2.3 Discussion

The new StringTie algorithm described here uses the strengths of both long and short-read RNA-seq data to improve transcriptome assembly. By using the short reads to support or adjust splice sites identified in the long-read alignments, we were able to reduce noise caused by the high error rate of long reads. Using simulated data, we demonstrated that hybrid-read assemblies achieve greater precision and sensitivity than both the long-read only and short-read only assemblies across a range of sensitivity parameters. We also showed that the calculated transcript coverage correlates better with the true coverage in the hybrid-read assemblies. Lastly, we confirmed that these improvements are not simply due to the increased coverage of the hybrid reads. Using real data from 3 different species, we showed that hybrid-read assemblies are more precise than long and short-read assemblies across all samples in all species. The hybrid-read assemblies also contained more transcripts that precisely matched the reference annotations as compared to the long and short-read assemblies in all but 2 *Mus musculus* datasets (Figure 2.7 D & F). In these 2 datasets, the hybrid-read assemblies contained more annotated transcripts than the long-read assemblies, but slightly fewer than the short-read assemblies. The lowest error rate out of any of the long reads was 3.2% in the PacBio cDNA human reads. Even with this low error rate, the hybrid-read assembly was still more precise and contained more annotated transcripts than the long-read assembly. This suggests that even as error rates of ONT

data inevitably decline, hybrid-read assembly will still be preferable for the foreseeable future.

Performing hybrid assembly with the new StringTie algorithm is akin to correcting the long reads prior to assembly; therefore, we compared StringTie's hybrid assembly to assembling long reads corrected by TALC. Notably, read correction with TALC took 146 times longer to run than StringTie on human data. Furthermore, we found that all of the hybrid-read assemblies contained more annotated transcripts than the assemblies of TALC-corrected long reads. All but 2 *Arabidopsis thaliana* hybrid-read assemblies also achieved greater precision. We also tested whether using corrected long reads in a hybrid-read assembly would be more accurate than using uncorrected reads. As shown in Figure 2.7, the difference between using corrected versus uncorrected long reads with StringTie's hybrid algorithm is very small, ranging from ~0.5% to 1% for both precision and the number of annotated transcripts assembled. When considering the substantial increase in runtime and the marginal increase in accuracy, we conclude that using StringTie's hybrid assembly algorithm with uncorrected long reads is the preferable method of transcriptome assembly.

Because *Arabidopsis thaliana, Mus musculus*, and human are well-studied organisms, they have high-quality reference annotations. This allowed us to perform separate experiments in which StringTie was run with a guide annotation. Across all datasets among the simulated and real data we saw substantial improvements in accuracy. This evidence indicates that the best results are achieved with annotation-guided hybrid

assembly for species with high-quality reference annotations. We have demonstrated that hybrid-read assembly with StringTie is better than long-read, short-read, or corrected long-read assemblies. As the first reference-based, hybrid-read transcriptome assembler, we believe this new release of StringTie will be a valuable tool leading to improvements in transcriptomic studies of many species.

2.4 Methods

StringTie Algorithm for Hybrid Data

As previously described, StringTie takes as input an alignment file of all reads from a sample in either SAM or BAM format[25], and uses these alignments to create a splice graph[20]. This new release of StringTie also supports input alignment data in CRAM format as it now makes use of the HTSlib C library[26] and can operate in *hybrid data* mode, enabled by the **`--mix`** option. In this new mode of operation, StringTie takes as input two alignment files, the first file on the command line containing the short-read alignment data and the second one having the long-read alignments. These two alignment files are parsed in parallel to identify clusters of reads that represent potential gene loci. Errors in the reads or the alignments, which are commonly present in the long-read data, propagate to the construction of the splice graph, creating vastly more paths through the graph, which not only slows down the algorithm, but also makes it much more difficult to choose the correct set of isoforms (each of which corresponds to a path) at a particular gene locus. As illustrated in Figure 2.10, each mis-aligned long read can create a "noisy" transcript that appears to have alternative donor and acceptor

sites, extra exons, or skipped exons. In the figure, we show two noisy transcripts, one with an extra exon and an erroneous acceptor (AG) site, and the other with two erroneous donor (GT) sites. These two noisy transcripts together contribute four additional exons to the splice graph, shown on the right side of the figure. These additional exons then generate 8 additional, erroneous edges in the graph, shown in orange. Thus, while the clean splice graph has only 4 nodes and 4 edges, the noisy splice graph has 8 nodes and 12 edges. Because every possible path through a splice graph is a possibly valid isoform, the number of isoforms grows exponentially as we add edges. In this simplified example, the clean splice graph shown on the upper right, based on 2 error-free transcripts, has only 2 paths, each representing a correct transcript. The noisy splice graph, in contrast, has 10 possible paths, only 2 of which correspond to genuine transcripts. Note that a splicing graph implicitly assumes independence of local events, and thus it typically contains many more legal paths than the number of transcripts used to create it.

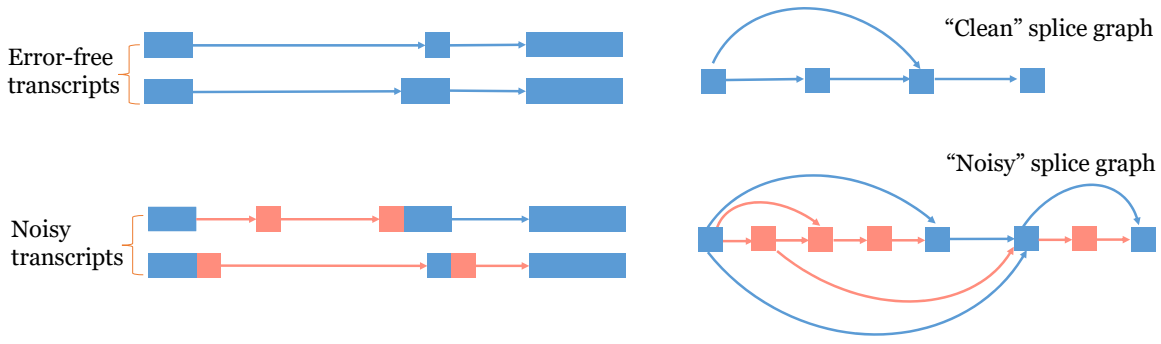

Figure 2.10. Example of noisy splice graph. Noisy alignments make the splice graph vastly more complicated. The clean splice graph on the upper right is based on the two error-free transcripts, while the noisy splice graph is based on all four of the transcripts shown on the left. Regions shown in orange are errors due to mis-alignments.

With a hybrid data set containing both long and short reads, we can take advantage of highly accurate short reads to fix most of these problems. The strategy we employ is to scan all the splice sites at a locus in order to evaluate how well-supported each site is by the read alignments. If a splice site is not well-supported (e.g., by at least one short read, or by most of the long reads that have splice sites in a small window around that particular splice site), we will search for a nearby splice site with the best support (i.e. one that has the largest number of alignments agreeing with it), and adjust the long-read alignment correspondingly. We found that this strategy can greatly reduce the number of spurious splice sites. Relying on short read data, we can also fix other long read alignment artifacts. For instance, one common problem that we and others [27] noticed is the ambiguity of strand of origin for long reads. Due to their high error rate, the aligner sometimes infers the wrong strand for the long-read alignment. We can fix this by scanning nearby splice sites, and choose the strand of the alignment that is best supported by the short-read data. Another common problem is the presence of false "exons" introduced by insertions in the long reads. These insertions tend to be small (usually less than 35bp), so to address this issue, we remove exons that have support only from long reads and that are contained within introns that are well supported by short read alignments.

After the splice graph has been pruned to remove erroneous splice sites and nodes, the hybrid version of StringTie will execute the next two steps:

1. First, it will cluster all compatible long-read alignments. We can do this efficiently by taking advantage of the sparse bit vector representation of the

splice graph already employed by StringTie, where each node or edge in the graph corresponds to a bit in the vector. A read or a paired read (in the case of short read data) will therefore be represented by a vector of bits where only the bits that represent the nodes or edges spanned by the read and its pair are set to 1. The bit representation provides a quick way to check compatibilities between long reads. Each cluster will represent a path in the splice graph that will have an initial expression level estimate $E(l)$ based on the number of long reads covering that path. Note that a cluster does not always have to be a full transcript (i.e. if all long reads in the cluster come from a truncated cDNA molecule), although in most cases it will be.

2. For each cluster path P inferred in the previous step, starting from the one with the largest number of long reads supporting it, StringTie will use the short-read alignment to output an assembled transcript and expression level estimate. First, StringTie will choose the heaviest path in the splice graph that includes P. This will represent a candidate transcript. Then StringTie will use its maximum flow algorithm to compute an expression level estimate $E(s)$ based on short-read data only. The final expression level of the transcript will be equal to $E(l)+ E(s)$, and short-read alignments that contribute to $E(s)$ will be removed from the subsequent expression level computations.

Note that some gene loci might have either only long-read or short-read alignments present. For those cases, StringTie will follow its previously implemented algorithms to assemble those loci [21].

Data and Commands

Reference Genomes and Annotations

The human reads (simulated and real) were aligned to GRCh38 and compared to the RefSeq annotation version GRCh38.p8 for accuracy. The *Mus musculus* reads were aligned to GRCm39 (GenBank Accession GCA_000001635.9) and accuracy was computed using the GENCODE annotation version M26. The *Arabidopsis thaliana* reads were aligned to TAIR10.1 (GenBank Accession GCA_000001735.2)

Simulated Data Generation

We used the same short read simulated data from FluxSim [28] as was used to evaluate StringTie2 [21] . We used NanoSim[29] to simulate ONT direct RNA sequencing reads. Using the NA12878-dRNA reads, we built a model of the reads by using the read_analysis.py module of NanoSim in transcriptome mode with the following command:

```
read_analysis.py transcriptome -i ONT_dRNA_reads.fq -rg GRCh38.fa -rt
transcripts.fa -annot  hg38c_protein_and_lncRNA_sorted.gtf -o training
```

where `transcripts.fa` is the human reference transcriptome obtained by using gffread [30].

We simulated 13,361,612 reads (the same number of reads in the NA12878-dRNA sample used build the model) by running the simulator.py module of NanoSim in transcriptome mode with the following command:

```
simulator.py transcriptome -rt transcripts.fa -rg GRCh38.fa -e
expression_levels.tpm -r dRNA -n 13361612 –fastq -o simulated_dRNA -
b guppy -c training
```

To match the expression levels of the long reads to the short reads, we used the .pro file generated by FluxSim to calculate the TPM of each transcript. These values were given as input to the NanoSim simulation with the -e parameter.

Equal Coverage Simulation

To control for coverage in the simulated data, we first calculated the coverage of each dataset simply by summing the lengths of every read and dividing by the sum of the lengths of the transcripts expressed. Doing this we found that the coverage of the short reads was 164.9, the coverage of the long reads was 195.7, and the coverage of the hybrid reads was 356.1. The long reads had slightly fewer transcripts expressed which is why the hybrid read coverage is not exactly the sum of long and short read coverage. To increase the coverage of the short reads, we reran FluxSim and increased the number of simulated reads from 150,000,000 to 323,636,363, and this resulted in a coverage of 355.9. To increase the coverage of the long reads, we re-ran NanoSim and increased the number of simulated reads from 13,361,612 to 24,306,253. This resulted in a coverage of 342. Because the lengths of the long reads vary extensively, unlike the short reads, the increase in coverage is not always proportional to the increase in the

number of reads. Nonetheless the coverage of this dataset is much higher than the original coverage of 195.7 and quite close to the target value of 356.1.

Alignment and Assembly

All short reads were aligned with HISAT2 with default parameters [31] using the following command:

```
hisat2 -x hisat2_index -1 short_reads_R1.fastq -2
short_reads_R2.fastq -S short_aligned.sam
```

Long reads were aligned with Minimap2[32] using the default parameters for spliced alignment with the following command:

```
minimap2 -ax GRCh38.fa long_reads.fastq -o long_aligned.sam
```

Alignment files were sorted and converted to BAM format using samtools[25]. The StringTie commands used to assemble the input alignment file for each assembly type were:

- For long-read data: `stringtie -L long_reads.bam`
- For short-read data: `stringtie short_reads.bam`
- For hybrid data: `stringtie --mix short_reads.bam long_reads.bam`

In the case of annotation-guided assembly, we added to all commands above the following option: `-G reference_annotation.gtf`

Accuracy Analysis

We define sensitivity as TP/(TP + FN) and precision as TP/(TP + FP) where TP (true positives) are correctly assembled transcripts, FP (false positives) are transcripts that are assembled but do not match the reference annotation, and FN (false negatives) are expressed transcripts that are missing from the assembly. We used gffcompare [30] to obtain these metrics in addition to the number of annotated transcripts assembled. All numbers reported are at the 'transcript' level (as opposed to the intron or base level accuracy also reported by gffcompare). The 'true positive' reference sets provided to gffcompare (with the -r option) are as follows:

Simulated data with varying sensitivity parameters (Figure 2.3): Human reference transcripts fully covered by either the long or short simulated reads. We define full coverage for multi-exon transcripts as coverage across all splice sites. For single-exon transcripts, it is considered fully covered if there is coverage across >=80% of the length.

Simulated data with default sensitivity parameters (Figure 2.5): The full set of expressed transcripts in the simulated data.

Simulated data with equal coverage of long, short, and hybrid reads (Figure 2.6). The full set of expressed transcripts in the simulated data.

Real data (Figure 2.7 and 2.9.): The reference annotation for the given species filtered to only include loci covered by at least one long read.

Real Data (Figure 2.8): the full reference annotation for the given species

The -Q option was used with gffcompare to only consider loci present in the reference set provided.

Our main metric used to compare the accuracy of the long, short, and hybrid-read assemblies is relative percent increase in sensitivity and precision which is defined as $(S_1-S_2)/S_2$ and $(P_1-P_2)/P_2$ where S_1 and P_1 are the sensitivity and precision of the hybrid-read assembly and S_2 and P_2 are the sensitivity and precision of the assembly we are comparing it to. For example, a 10% absolute increase in sensitivity from S_2 = 20% to S_1 = 30% results in a relative increase of 50% [21]. For the real data, S is the number of annotated transcripts assembled.

Coverage Analysis of Simulated Data

The expected coverage for the long-read only and short-read only assemblies was obtained by taking the sum of the lengths of all the reads covering a transcript and dividing it by the transcript length. For the hybrid-read assemblies, the expected coverage was calculated by taking the sum of the short-read and long-read expected coverages of each transcript. The computed read coverages were taken from StringTie's output for each type of assembly. All coverages were exported to R and

normalized to $\log_2(1 + \text{coverage})$. To make the comparison fair, we only plotted the coverages and calculated the R^2 for the transcripts that were shared between the long-read only, short-read only, and hybrid-read assemblies.

Long-read Correction with TALC

For each set of long reads, we first counted all 21-mers in the short reads from the sample using Jellyfish [33]. The kmer counts were obtained with the following commands:

```
jellyfish count --mer 21 -s 100M -o kmers.jf -t 8
$short_reads_1.fa $short_reads_2.fa
jellyfish dump -c kmers.jf > kmers.dump
```

Using the Jellyfish output, we ran TALC with the following command:

```
talc $long_reads.fa --SRCounts kmers.dump -k 21 -o
$long_reads_TALC.fa   -t 12
```

Error-rate Calculations and Full-length Isoform Analysis

The mature transcript sequences for each species were extracted from the reference genome using gffread with the following command:

```
gffread -w transcripts.fa -g reference_genome.fa
reference_annotation.gtf
```

We then aligned each long-read dataset to the transcript sequences using Minimap2 and output the alignments in PAF format. To calculate the error rate, we selected the primary alignment for each read and divided the number of matches by the alignment length. These values are in columns 10 and 11 respectively in the PAF alignment output.

To identify full-length isoforms, we filtered for reads that spanned all intron/exon boundaries of a multi-exon transcript or 80% of the length of a single-exon transcript. The reference annotations were used to identify the coordinates of the intron/exon boundaries.

Chapter 3: Liftoff: an accurate gene annotation mapping tool

A version of chapter 3 previously appeared as:

A. Shumate, S.L. Salzberg (2020). “Liftoff: accurate mapping of gene annotations” *Bioinformatics*, 1639-1643, 37(12).

3.1 Introduction

The declining cost of sequencing has allowed us to sequence and assemble the genomes of new organisms, but it has also allowed us to update and improve existing assemblies. The most well-known example of this is the human genome, but other model organisms such as mouse, zebrafish[34], rhesus macaque[35], maize[36], and many others have had a series of gradually improved assemblies. Beyond updating the reference genomes, it is also now feasible to sequence and assemble multiple members of the same species. For example, the Human Pangenome Reference Consortium[37] has released assemblies of 47 different individuals[38] and plans to release many more in the coming years. Rather than repeating the annotation process from scratch for each updated or additional genome for a given species, a more scalable approach is to take the annotation from a previously-annotated member of the same or closely-related species, and then map or ‘lift over’ gene models from the annotated genome onto the new assembly.

Current strategies for this task use tools such as UCSC liftOver[39] or CrossMap[40] to convert the coordinates of genomic features between assemblies; however, these tools only work with a limited number of species, and they rely only on sequence homology to find a one-to-one mapping between genomic coordinates in the reference and coordinates in the target. This strategy is often inadequate when converting genomic intervals, like a gene feature, rather than a single coordinate. If the interval is no longer continuous in the target genome, current strategies will either split the interval and map it to different locations or map the spanned interval to the target genome[41]. In many cases, this disrupts the biological integrity of the genomic feature; for example, if the interval is split and mapped to different chromosomes or strands, or spans a large genomic distance, it may not be possible for it to represent a single gene feature. Furthermore, prior tools convert each feature independently, so while every exon from one transcript may be lifted over to a continuous interval, the combination of exons in the target genome may not necessarily form a biologically meaningful transcript. Mapping each feature independently also often results in multiple paralogous genes incorrectly mapping to a single locus.

Here we introduce Liftoff, an accurate tool that maps annotations described in General Feature Format (GFF) or General Transfer Format (GTF) between assemblies of the same, or closely related species. Unlike current coordinate lift-over tools which require a pre-generated “chain” file as input, Liftoff is a standalone tool that takes two genome assemblies and a reference annotation as input and outputs an annotation of the target genome. Liftoff uses Minimap2[32] to align the gene sequences from a reference genome

to the target genome. Rather than aligning whole genomes, aligning only the gene sequences allows genes to be lifted over even if there are many structural differences between the two genomes. For each gene, Liftoff finds the alignments of the exons that maximize sequence identity while preserving the transcript and gene structure. If two genes incorrectly map to overlapping loci, Liftoff determines which gene is most-likely mis-mapped and attempts to re-map it. Liftoff can also find additional gene copies present in the target assembly that are not annotated in the reference.

Here, we describe the Liftoff algorithm as well as present more examples demonstrating the accuracy and versatility of Liftoff. First, we map genes between two versions of the human reference genome. Next, to demonstrate a cross-species lift over, we map protein-coding genes from the human reference genome to a chimpanzee genome assembly.

3.2 Methods

Liftoff is implemented as a python command-line tool. The main goal of Liftoff is to align gene features from a reference genome to a target genome and use the alignment(s) to optimally convert the coordinates of each exon. An optimal mapping is one in which the sequence identity is maximized while maintaining the integrity of each exon, transcript, and gene. While our discussion of Liftoff here focuses on lifting over genes, transcripts, and exons, it will work for any feature, or group of hierarchical features present in a GFF or GTF file.

As input, Liftoff takes a reference genome sequence and a target genome sequence in FASTA format, and a reference genome annotation in GFF or GTF format. The reference annotation is processed with gffutils[42], which uses a sqlite3 database to track the hierarchical relationships within groups of features (e.g., gene, transcript, exon). Using pyfaidx[43] Liftoff extracts gene sequences from the reference genome, and then invokes Minimap2 to align the entire gene sequence including exons and introns to the target. The Minimap2 parameters are set to output up to 50 secondary alignments for each sequence in SAM[25] format. Additionally, the end bonus parameter in Minimap2 is set to 5 to favor end-to-end alignments as opposed to soft-clipping mismatches at the end of alignments. While these parameters work well for the examples presented here, Liftoff allows the user to change or add any additional Minimap2 options. By default, genes are aligned to the entire target genome, but for chromosome-scale assemblies, the user can enable an option to align genes chromosome by chromosome. Under that option, only those genes which fail to map to their expected chromosome are then aligned to the entire genome.

In many cases, a gene has a single complete alignment to the target genome, which makes finding the optimal mapping trivial. In other cases, differences between the two genomes cause the gene to align in many fragmented pieces, and the optimal mapping is some combination of alignments. To find this combination, Liftoff uses networkx[44] to build a directed acyclic graph representing the alignments as follows. Using Pysam[45] to parse the Minimap2 alignments, each alignment is split at every insertion and deletion in order to form a group of gapless alignment blocks. Blocks not containing any part of an

exon are discarded, and the remaining blocks are represented by nodes in the graph. Two nodes *u* and *v* are connected by an edge if the following conditions are true.

1) *u* and *v* are on the same chromosome or contig
2) *u* and *v* are on the same strand
3) *u* and *v* are in the correct 5' to 3' order
4) The distance from the start of *u* to the end of *v* in the target genome is no greater than 2 times that in the reference genome

Nodes in the graph are weighted according to mismatches within exons. By default, a mismatch within an exon incurs a penalty of 2. Edges are assigned a weight according to the length of gaps within exons. By default, opening a gap in an exon incurs a penalty of 2, and extending it incurs a penalty of 1. Mismatches and gaps within introns are not counted. The mismatch, gap open, and gap extend parameters can be changed by the user. A source and sink are added to the graph representing the start and end of the gene respectively, and the shortest path from source to sink is found using Dijkstra's algorithm [46] where the weight function between two nodes *u* and *v* is

$$\frac{weight_u + weight_v}{2} + weight_{edge}$$

The shortest path represents the combination of aligned blocks that is concordant with the original structure of the gene and minimizes the number of mismatches and indels within exons. The alignments in this path define the final placement of the gene. Using the coordinates of the aligned blocks in the shortest path, the coordinates of each exon

are converted to their respective coordinates in the target genome. A simple example of this process is shown in Figure 3.1**,** which illustrates lifting over a 5-exon transcript from the human reference genome (GRCh38) to a chimpanzee genome (PTRv2). This gene has a large intronic deletion in PTRv2 and does not have and end-to-end alignment, but it can still be successfully lifted over using our algorithm.

One of the main challenges with gene annotation lift over is correctly mapping homologous genes from multi-gene families. Two different genes may optimally map to the same locus if they are identical or nearly identical. To handle this situation, after Liftoff maps all genes to their best matches, it checks for pairs of genes on the reference genome that have incorrectly mapped to overlapping (or identical) locations on the target genome, and it then attempts to find another valid mapping for one of the genes. Liftoff first tries to remap the gene with the lower sequence identity. If the genes mapped with the same sequence identity, Liftoff considers the neighboring genes and tries to remap the gene that appears out of order according to the reference annotation. When remapping the gene, Liftoff rebuilds the graph of aligned blocks excluding any blocks that overlap the homologous gene. The shortest path through this new graph represents the best mapping for this gene that does not overlap its homolog. If another valid mapping does not exist, the gene with lower identity is considered unmapped. This process is repeated until there are no genes mapped to overlapping loci. Liftoff then outputs a GFF or GTF file with the coordinates on the target genome of all of the features from the original annotation, and a text file with the IDs of any genes that could not be lifted over.

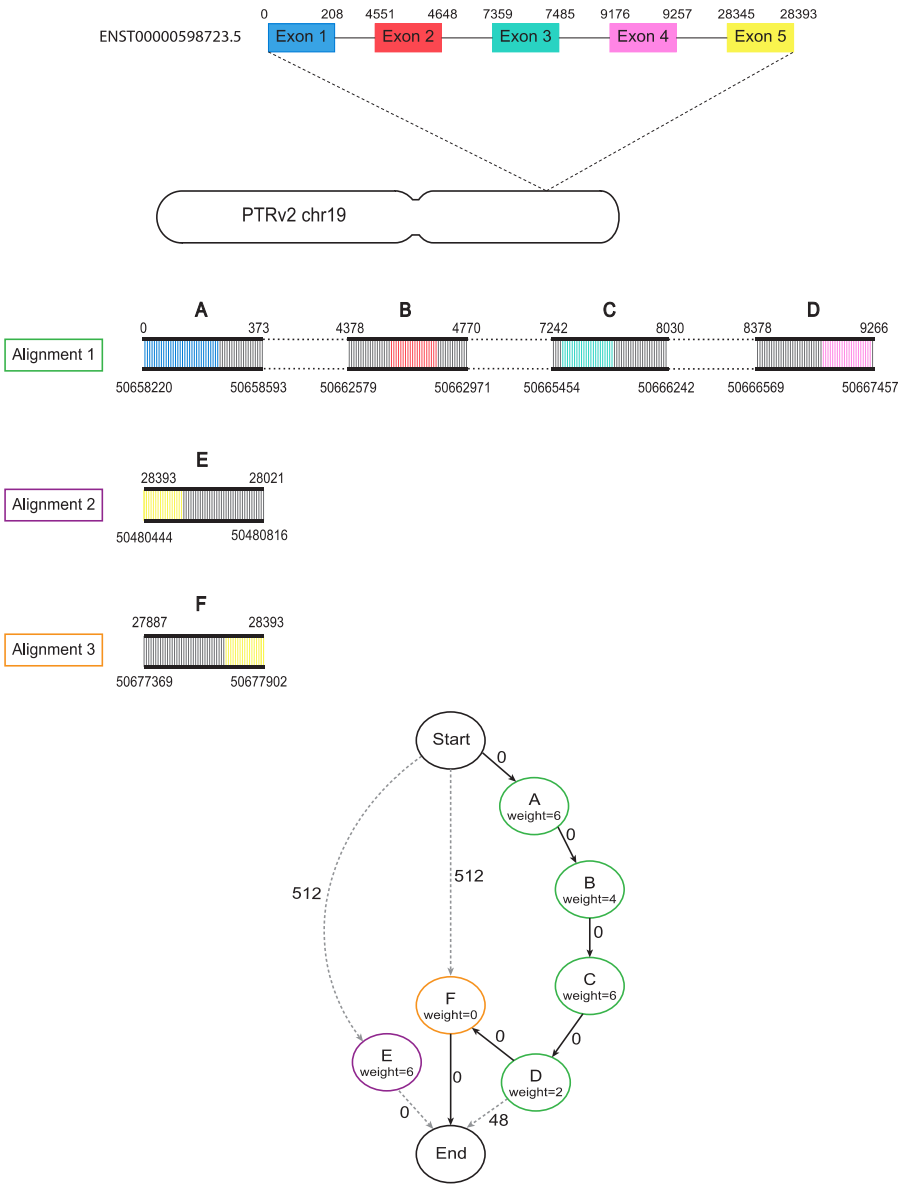

Figure 3.1. Example of the lift-over process. Diagram showing the steps taken by Liftoff when mapping human transcript ENST00000598723.5 to the chimpanzee (PTRv2) homolog on chromosome 19. Minimap2 produces 3 partial alignments of this gene to PTRv2. Alignment 1 (green) has 4 gapless blocks containing exons 1-4 which are represented by nodes A-D in the graph. The dashed lines in between blocks of the alignment represent gaps/introns. Alignments 2 (purple) and 3 (orange) each have 1 gapless block containing exon 5 represented by nodes E and F respectively. Node E is not on the same strand as alignments 1 and 2 and is therefore only connected to the start and end. The node weights correspond to the exon mismatch penalties (default of 2 per mismatch) and the edge weights are the sum of the exon gap open penalty (2) and gap extension penalty (1). An edge weight of zero means the gaps did not occur within an exon. The shortest path (A,B,C,D,F) is shown with bold arrows and contains complete alignments of all 5 exons with a total of 9 mismatches and 0 gaps.

Note that differences in the genome sequences themselves may result in Liftoff mapping a gene to a paralogous location. For example, consider a gene family with 5 members on the reference genome but only 4 members on the target. The fifth gene might simply be unmapped, but if the target has a paralogous copy elsewhere, *and* if that copy is not matched by a homolog on the reference, then Liftoff will map the fifth gene to the paralogous location.

Another feature unique to Liftoff is the option to find additional copies of genes in the target assembly not annotated in the reference. With this option enabled, Liftoff maps the complete reference annotation first, and then repeats the lift-over process for all genes. An extra gene copy is annotated if another mapping is found that does not overlap any previously-annotated genes, and that meets the user-defined minimum sequence identity threshold. The lift-over procedure is repeated until all valid mappings have been found.

3.3 Results

Here we demonstrate Liftoff's ability to lift an annotation to an updated reference genome by lifting genes from the two most recent versions of the human reference genome, GRCh37 and GRCh38. We also demonstrate Liftoff's ability to lift genes between genomes of closely-related species by lifting genes from GRCh38 to the chimpanzee genome Clint_PTRv2. To assess the accuracy of Liftoff in each example, we evaluate the sequence identity and order of mapped genes.

GRCh37 to GRCh38

We attempted to map all protein-coding genes and lncRNAs on primary chromosomes (excluding alternative scaffolds) in the GENCODE v19 annotation[47] from GRCh37 to GRCh38. Out of 27,459 genes, we successfully mapped 27,422 (99.87%). We consider a gene to be successfully mapped if at least 50% of the reference gene maps to the target assembly. Genes that failed to map according to this threshold are listed in Supplementary Table 1 of Shumate and Salzberg 2020[9]. An overwhelming majority of the gene sequences in GRCh38 were nearly identical to the sequences in GRCh37, with an average sequence identity in exons of 99.97% (Figure 3.2).

To visualize the co-linearity of the gene order between the two assemblies, we plotted each gene as a single point on a 2D plot where the X coordinate is the ordinal position of the gene in GRCh37 and the Y coordinate is the ordinal position in GRCh38 (Figure 3.3). The gene order appears perfectly co-linear; however, there are some exceptions not visible at the scale of the whole genome. To calculate the number of genes out of order in GRCh38 with respect to GRCh37, we calculated the edit distance between the gene order in each assembly. This revealed 361 genes (1.3%) in a different relative position in GRCh38 compared to GRCh37.

To compare Liftoff to an existing commonly used method, we lifted over genes between the same 2 assemblies using the UCSC liftOver tool. UCSC liftOver failed to map 122 genes. 63 of these genes mapped end-to-end with Liftoff (Supplementary Table 2 in

Shumate and Salzberg 2020) and 27 mapped partially with an alignment coverage less than 100% but greater than the 50% threshold mentioned above.

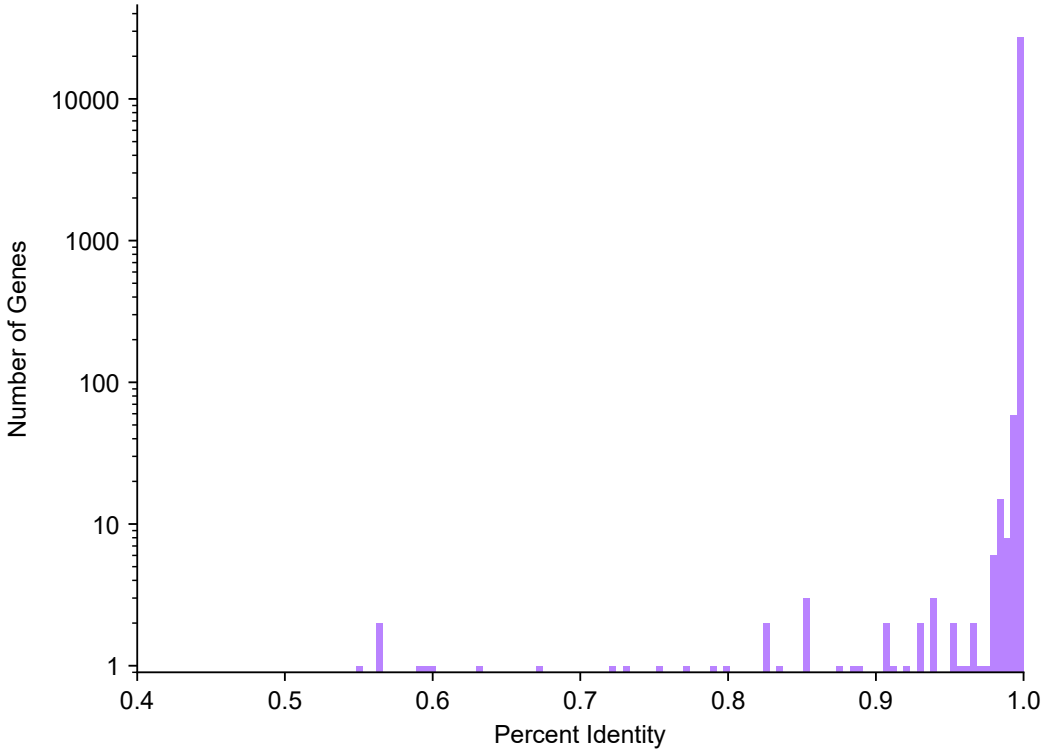

Figure 3.2. Distribution of GRCh37 and GRCh38 sequence identity. Histogram showing the distribution of exon sequence identity of protein-coding and lncRNA genes in GRCh37 and GRCh38. Log scale used to make the counts of just 1 or 2 genes visible; all bins below 97% identity contain at most 4 genes.

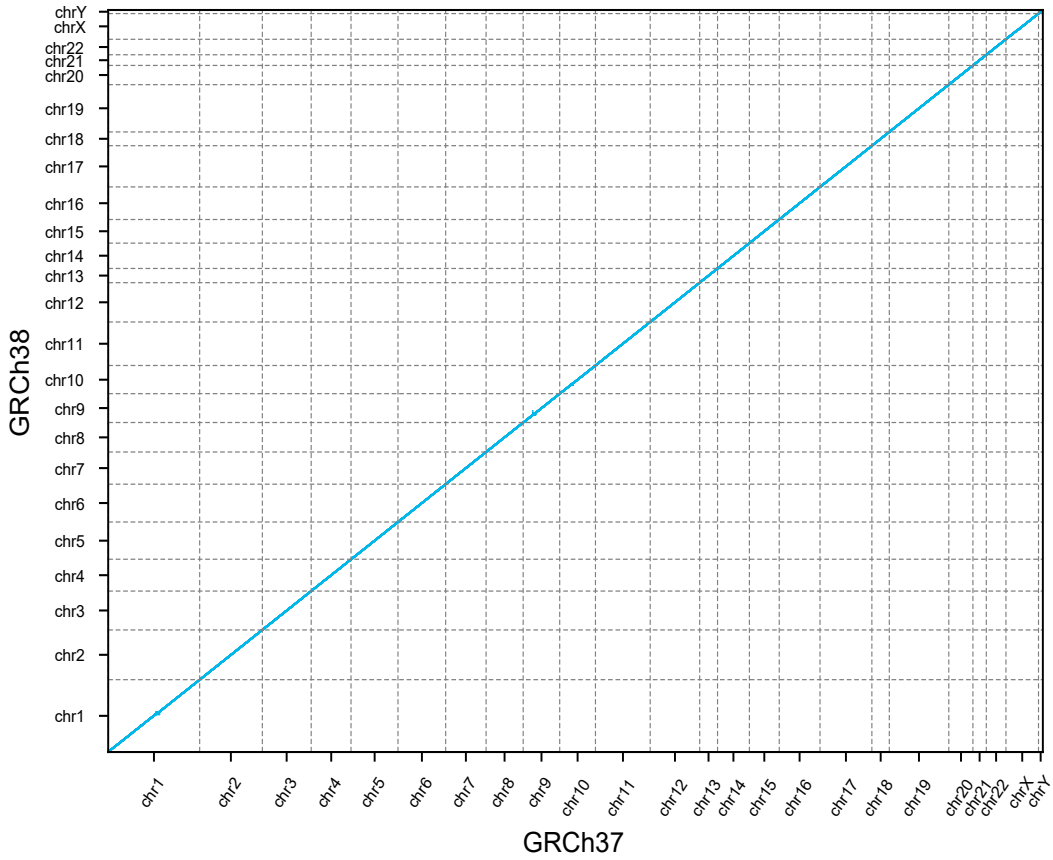

Figure 3.3. GRCh37 and GRCh38 gene order. Dot plot showing the ordinal position of each gene in GRCh37 on the x-axis and the ordinal position in GRCh38 on the y-axis.

GRCh38 to PTRv2

We attempted to map all protein-coding genes on chromosomes 1-22 and chromosome X in the GENCODE v33 annotation [48] from GRCh38 to an assembly of the chimpanzee (*Pan troglodytes*), PTRv2 (GenBank accession GCA_002880755.3). Out of 19,878 genes, we were able to map 19,543 (98.31%). Genes that failed to map according to this threshold are listed in Supplementary Table 4 of Shumate and Salzberg 2020. The average sequence identity in exons of successfully mapped genes was 98.21% (Figure 3.4).

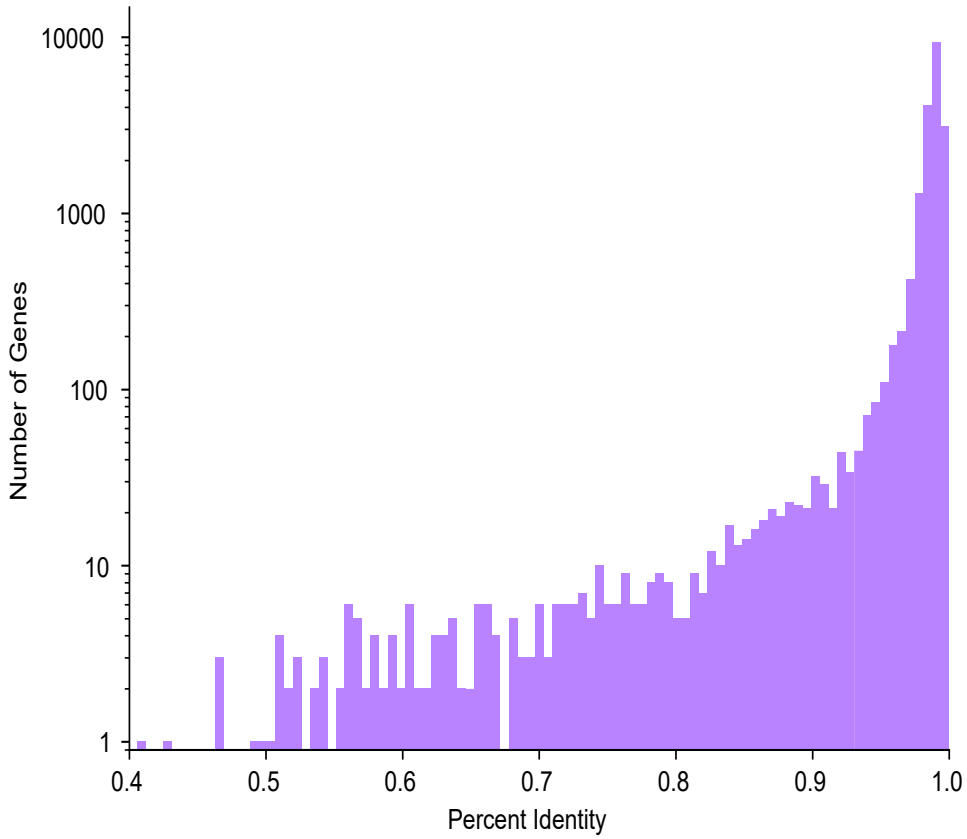

Figure 3.4. Distribution of GRCh38 and PTRv2 sequence identity. Histogram showing the distribution of exon sequence identity of protein-coding genes in GRCh38 and PTRv2. Note that the y-axis is shown on a log scale.

As was done with the GRCh37 to GRCh38 lift-over, we compared the gene order in GRCh38 to that in PTRv2 and found 2,477 genes in PTRv2 to be in a different relative position. Some of these ordinal differences are visible at the whole-genome scale (Figure 3.5) including 4 large regions on the chimpanzee homologues of chromosomes 4, 5, 12, and 17 where the gene order is inverted due to large-scale chromosomal inversions.

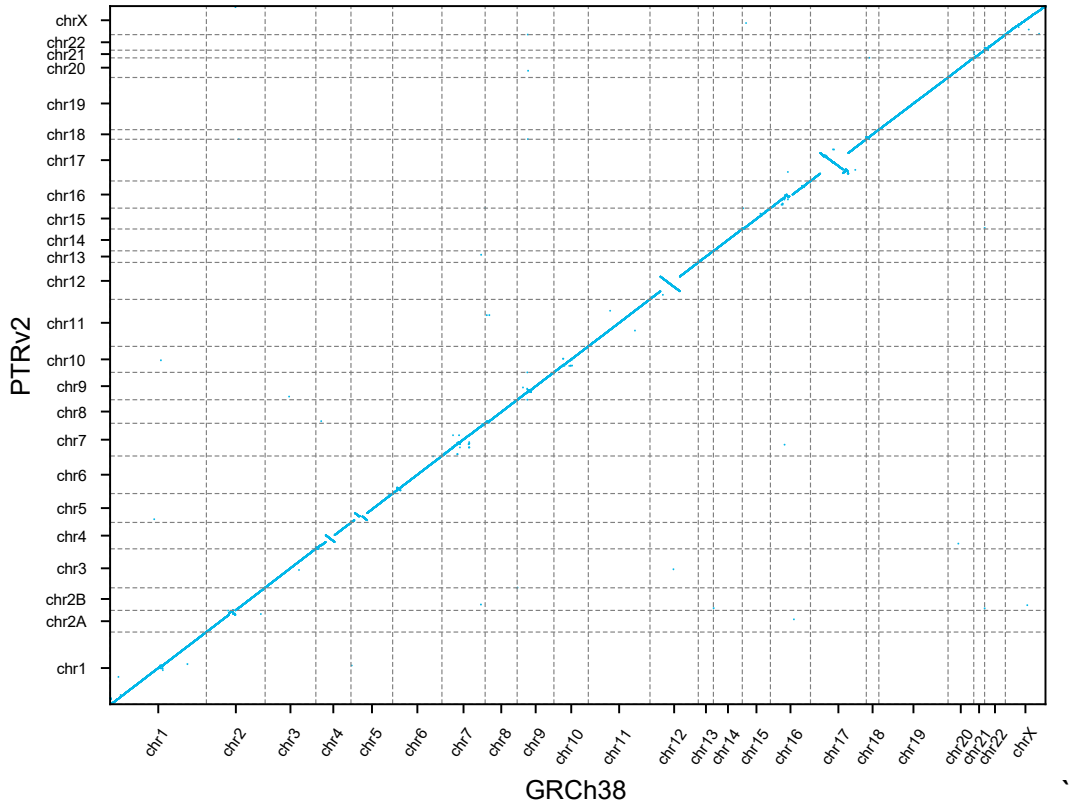

Figure 3.5. GRCh38 and PTRv2 gene order. Dot plot showing the ordinal position of each gene in GRCh38 on the x-axis and the ordinal position in PTRv2 on the y-axis.

We again compared our results to UCSC liftOver. We found that UCSC liftOver failed to map 597 genes. 78 of these genes mapped end-to-end with Liftoff (Supplementary Table 4 in Shumate and Salzberg 2020) and 417 mapped with an alignment coverage less than 100%, but greater than the 50% threshold.

3.4 Discussion

The rapidly growing number of high-quality genome assemblies has greatly increased our potential to understand sequence diversity, but accurate genome annotation is

needed to understand the biological impact of this diversity. Rather than annotating genomes *de novo*, we can take advantage of the extensive work that has gone into creating reference annotations for many well-studied species. We developed Liftoff as an accurate tool for transferring gene annotations between genomes of the same or closely-related species. Unlike current coordinate lift-over strategies which only consider sequence homology, Liftoff considers the constraints between exons of the same gene and constraint that distinct genes need to map to distinct locations. We demonstrate that this approach can map more genes than sequence homology-based approaches.

We showed that we were able to lift over nearly all genes from GRCh37 to GRCh38. The gene sequences and order are very similar between the two assemblies, with an average sequence identity of >99.9% and only 361 genes appearing in a different order. GRCh38 fixed a number of mis-assemblies and single base errors present in GRCh37[49], so it is expected that the gene sequence and order are not entirely identical. This demonstrates Liftoff's ability to accurately annotate an updated reference assembly, making it a useful tool as reference assemblies are continuously updated.

We also showed that we could lift-over nearly all protein-coding genes from GRCh38 to the chimpanzee genome, PTRv2, with an average sequence identity of 98.2%. This is consistent with previous work showing the human genome and chimpanzee genome are approximately 98% identical[50]. Comparing the gene order revealed 4 large regions on the homologs of chromosomes 4, 5, 12, and 17 where the gene order is inverted. These regions are consistent with previous reports: the chimpanzee genome has 9 well-

characterized pericentric inversions on chromosome homologs 1, 4, 5, 9, 12, 15, 16, 17[51]. The 4 largest of these inversions are on 4, 5, 12, and 17[52] hence their visibility at this scale. Additionally, the co-linear mapping of genes from human chromosome 2 to chimpanzee chromosomes 2A and 2B is consistent with the known telomeric fusion of these chromosomes[51]. The consistency of the gene sequence identity with the known genome sequence identity between chimpanzee and human, and the consistency of the gene order with the known structural differences between the two genomes demonstrate the accuracy of Liftoff's gene placements in a cross-species lift-over.

There are some limitations with annotating new assemblies using a lift-over strategy rather than *de novo.* First of all, the success of the lift-over is limited by the divergence between the reference and target genomes (Figure 3.6). Secondly, the annotation of the new assembly will only be as complete as the reference. However, as more genomes are sequenced and assembled, and reference annotations continue to improve through manual curation, experimental validation, or improved computational methods, Liftoff will enable easy integration of these improvements across many genomes. We anticipate that Liftoff will be a valuable tool in improving our understanding of the biological function of the large and rapidly growing number of sequenced genomes.

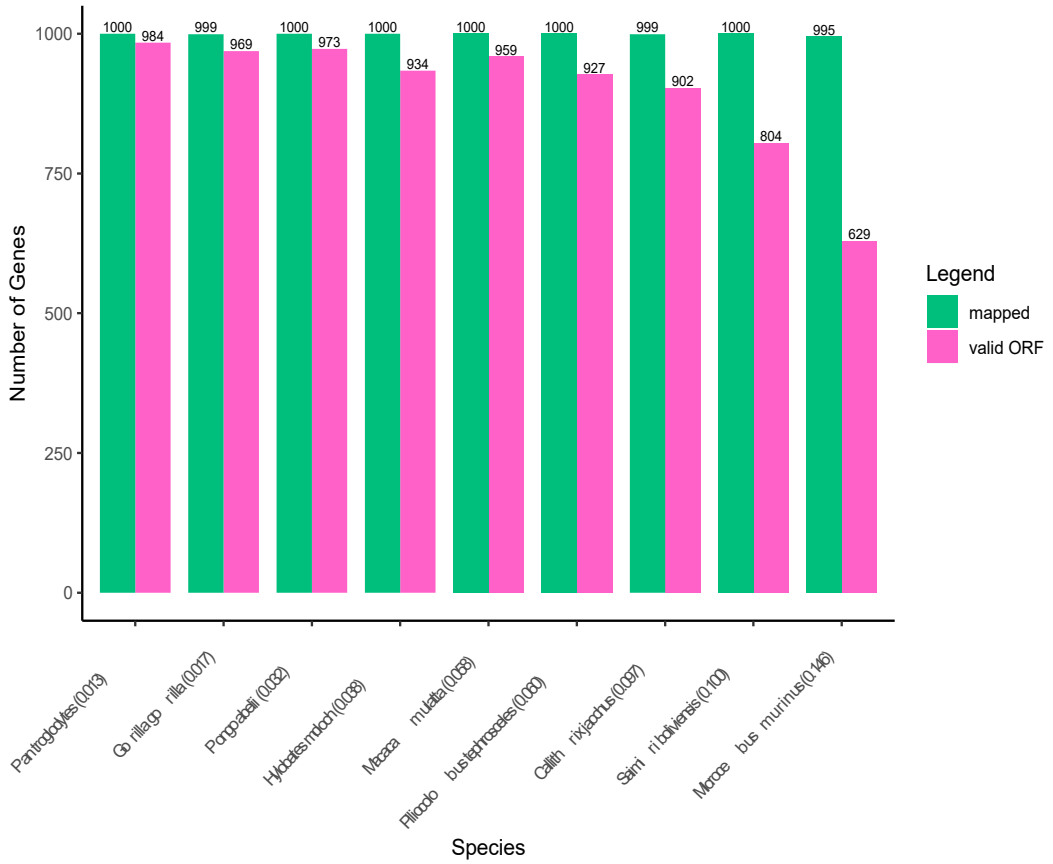

Figure 3.6. Gene mapping results on 9 primate genomes. 9 primate genome assemblies and reference annotations were downloaded from the NCBI RefSeq database. We then used the reference annotations to identify genes common to all 9 primates and human. We selected a random subset of 1000 genes from this list to lift-over from GRCh38 to each primate genome. By selecting a test set of genes present in all genomes, we control for lift-over failures due to genuine biological differences and/or incomplete assemblies. The x-axis shows the species name sorted by Mash distance[53] from human, which is shown in parenthesis. The y-axis is the number of genes out of 1000 that successfully mapped onto the primate assembly (green) and the number of these mapped genes which have at least 1 valid open reading frame (pink). Valid open reading frames were found by using gffread[54] with the -P option to identify and then filter out CDS's that did not begin with a start codon, end with a stop codon, or contained a premature stop codon.

Chapter 4: Annotation of 3 human genome assemblies

This chapter contains sections from the following publications:

1) A. Shumate *, A.V. Zimin*, R.M. Sherman,…& S.L. Salzberg (2020). “Assembly and annotation of an Ashkenazi human reference genome.” *Genome Biology* 21,129.

2) A.V. Zimin*, A. Shumate*, I. Shinder, J. Heinz, D. Puiu, M. Pertea, S.L. Salzberg (2022). “A reference-quality, fully annotated genome from a Puerto Rican individual.” *Genetics* 220(2).

3) S. Nurk, S. Koren, A. Rhie, M. Rautianen,…& K. H. Miga, A.M. Phillippy (2021). “The complete sequence of a human genome.” *bioRxiv.* (Submitted for publication)

* indicates equal contribution

These 3 publications describe in detail the assembly and annotation of human genomes. My work was focused just on the annotations, so the assemblies are described only briefly here in the introduction for context. For more information about the assembly results and methods, refer to the publications.

Additional Contributors:

The introduction contains a small section from publication 1 written by Steven Salzberg. The translocation analysis of Ash1 was conducted by Steven Salzberg and Aleksey

Zimin. The analyses of the unmapped genes in PR1 were done by my undergraduate mentee Jakob Heinz. The annotation of the T2T-CHM13 genome was conducted jointly with Mark Diekhans and Marina Haukness. Specifically, they created the annotation with the Comparative Annotation Toolkit. The section here describing the annotation results of the T2T-CHM13 genome is directly from publication 3 written by Sergey Nurk, Sergey Koren , Karen Miga, Adam Phillippy, and other members of the Telomere-to-Telomere consortium.

4.1 Introduction

Since 2001, the international community has relied on a single reference genome (currently GRCh38 released in 2013). While this reference is significantly improved from the original version published in 2001, some major problems persist.

First and foremost, GRCh38 is incomplete. The reference genome was created primarily through Sanger sequencing of bacterial artificial chromosome (BAC) clones[55], and the limitations of this technology restricted the assembly to only the euchromatic regions. After 20 years of work, the current reference genome still has 151 Mbp of missing sequence which amounts to around 8% of the genome. These missing regions include the short arms of the 5 acrocentric chromosomes, the ribosomal DNA (rDNA) arrays, and large satellite arrays[12]. The second major problem is that the current reference represents a tiny fraction of the total population. The sequence is a mosaic from a small number of individuals with about 65% originating from a single person[56]. Many studies have pointed out that a single genome is inadequate for a variety of reasons, such as inherent bias towards the reference genome[57–59]. The availability of

reference genomes from multiple human populations would greatly aid attempts to find genetic causes of traits that are over- or under-represented in those populations, including susceptibility to disease[60]. Another drawback of relying on a single reference genome is that it almost certainly contains minor alleles at some loci, which in turn confounds studies focused on variant discovery and association of those variants with disease[60–63].

These two major limitations of the reference genome have been improved upon in recent work. In 2021, enabled by long-read sequencing technology, the Telomere-to-Telomere (T2T) consortium released the first truly complete sequence of a human genome[12]. This genome assembly was derived from the sequence of a complete hydatidiform mole cell line (referred to as CHM13) due to its essentially haploid nature. The CHM13 genome added 200 Mbp of novel sequence and fixed numerous errors in the current GRCh38 reference. Additionally, in an effort to reduce the reliance on a single reference, we have created reference-quality assemblies of an Ashkenazi individual (called Ash1)[10] and Puerto Rican individual (called PR1)[11] in 2019 and 2021 respectively. These assemblies were created from high-quality data provided by the Human Pangenome Reference Consortium[37]. In addition to being more complete than GRCh38, they come from populations that are not well-represented in GRCh38.

In order for these genomes to function as effective references, they need to be annotated. Because the current human reference is well-annotated, this represents the ideal use case for Liftoff [9] where we map genes from a reference genome to a target

genome of the same species. Ash1 and PR1 were annotated entirely with Liftoff, and CHM13 was annotated with both the Comparative Annotation Toolkit (CAT) [64] and Liftoff. In the following sections I describe the results and methods of the annotation process for each genome.

4.2 Results

Ash1

To annotate Ash1, we used the CHESS2.2 database[22] because it is comprehensive, including all protein-coding genes from both GENCODE[48] and RefSeq[65]. We attempted to map all 310,901 transcripts from 42,167 gene loci on the primary chromosomes in GRCh38 to Ash1. In total, we successfully mapped 309,900 (99.7%) transcripts from 42,070 gene loci onto the main chromosomes. We considered a transcript to be mapped successfully if the mRNA sequence in Ash1 is at least 50% as long as the mRNA sequence on GRCh38. However, the vast majority of transcripts greatly exceed this threshold, with 99% of transcripts mapping at a coverage greater than or equal to 95% (Figure 4.1). The sequence identity of the mapped transcripts is similarly high, with 99% of transcripts mapping with a sequence identity greater than or equal to 94% (Figure 4.2).

Of those genes with at least one successfully mapped isoform, 42,059 (99.7%) mapped to the corresponding locations on the same chromosome in Ash1. Of the 108 genes that initially failed to map, 11 genes mapped to a different chromosome in 7 distinct blocks (shown in Table 4.1), suggesting a translocation between the two genomes. Interestingly, 16 of the 22 locations involved in the translocations were in subtelomeric

regions, which occurred in 8 pairs where both locations were near telomeres. This is consistent with previous studies reporting that rearrangements involving telomeres and subtelomeres may be a common form of translocation in humans [66–68].

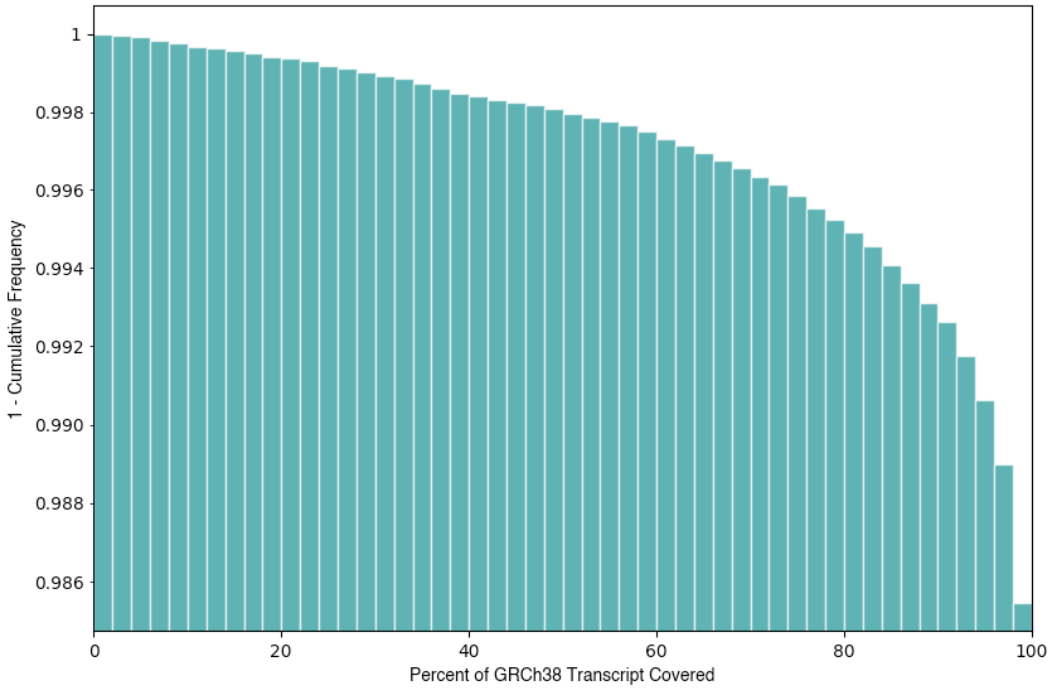

Figure 4.1. Ash1 cumulative distribution of coverage. Cumulative distribution showing how much of the GRCh38 transcripts map onto Ash1. The Y axis shows the fraction of transcripts with percent coverage greater than or equal to coverage on the X axis; e.g., the next-to-last bar at 98% on the X axis shows that 98.9% of GRCh38 transcripts (Y axis) mapped for at least 98% of their length onto Ash1.

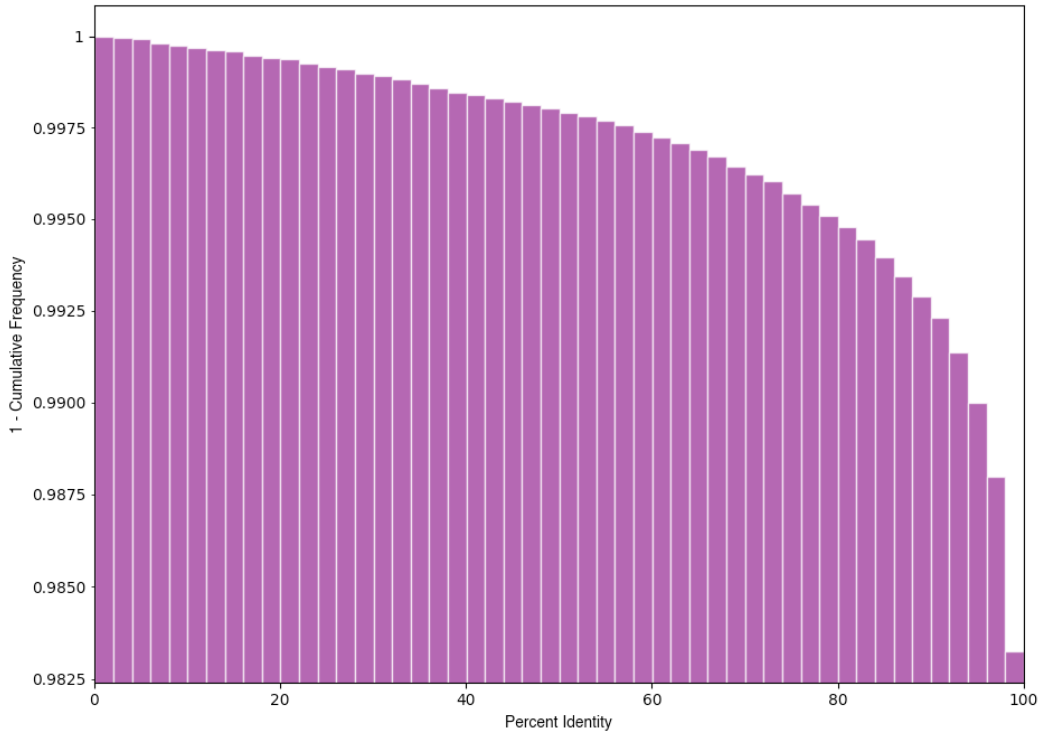

Figure 4.2. Ash1 cumulative distribution of sequence identity. Cumulative distribution of the sequence identity of transcripts mapped onto Ash1. The Y axis shows the fraction of transcripts that aligned between GRCh38 and Ash1 with DNA sequence identity greater than or equal to the percent identity on the X axis. E.g., the next-to-last vertical bar at 98% on the X axis shows that 98.75% of the GRCh38 transcripts aligned at 98% or greater identity to Ash1.

Table 4.1. Translocated genes in Ash1. 11 genes from GRCh38, 4 of them protein coding, that map to a different chromosome on Ash1. Genes are sorted by their position on GRCh38. Genes that appear to have moved in a block via a single translocation are highlighted in colored rows. Sub-telomeric coordinates are indicated by (T) next to the coordinates. Abbreviations: NC, noncoding.

CHESS ID	Gene Name	Gene Type	GRCh38 Location	Ash1 Location
CHS.460	HNRNPCL4	protein	chr1:13164555-13165482	chr6:113726526-113727453
CHS.39870	USP17L11	protein	chr4:9215405-9216997	chr11:71983132-71984724
CHS.39871	USP17L12	protein	chr4:9220152-9221744	chr11:71978387-71979979
CHS.54932	WASH1	protein	chr9:14475-30487 (T)	chr20:50732-69104 (T)
CHS.54933	LOC107987041	NC	chr9:27657-30891 (T)	chr20:65950-69493 (T)
CHS.54934	FAM138C	NC	chr9:34394-35864 (T)	chr20:65083816-65085286 (T)
CHS.18492	Unnamed	NC	chr15:101959848-101960582 (T)	chr20:65088782-65089512 (T)
CHS.18493	WASH3P	NC	chr15:101960813-101976605 (T)	chr20:65089741-65105526 (T)
CHS.18494	DDX11L9	NC	chr15:101976558-101979093 (T)	chr20:65105479-65108014 (T)
CHS.20775	LOC107987240	NC	chr16:90199813-90211886 (T)	chr20:2-12021 (T)
CHS.59387	DDX11L16	NC	chrY:57212178-57214703 (T)	chr20:48248-50782 (T)

We examined the translocation between chromosomes 15 and 20, which contains three of the genes in Table 4.1, by looking more closely at the alignment between GRCh38 and Ash1. The translocation is at the telomere of both chromosomes, from position 65,079,275 to 65,109,824 (30,549 bp) of Ash1 chr20 and 101,950,338 to 101,980,928 (30,590 bp) of GRCh38 chr15. To confirm the translocation, we aligned an independent set of very long PacBio reads, all from HG002, to the Ash1 v1.7 assembly (See Methods) and evaluated the region around the breakpoint on chr20. These alignments show deep, consistent coverage extending many kilobases on both sides of the breakpoint, supporting the correctness of the Ash1 assembly (Figure 4.3).

Sixty-two genes failed entirely to map from GRCh38 onto Ash1, and another 32 genes mapped only partially (below the 50% coverage threshold), as shown in Table 5 of Shumate, Zimin *et al.* 2020[10].

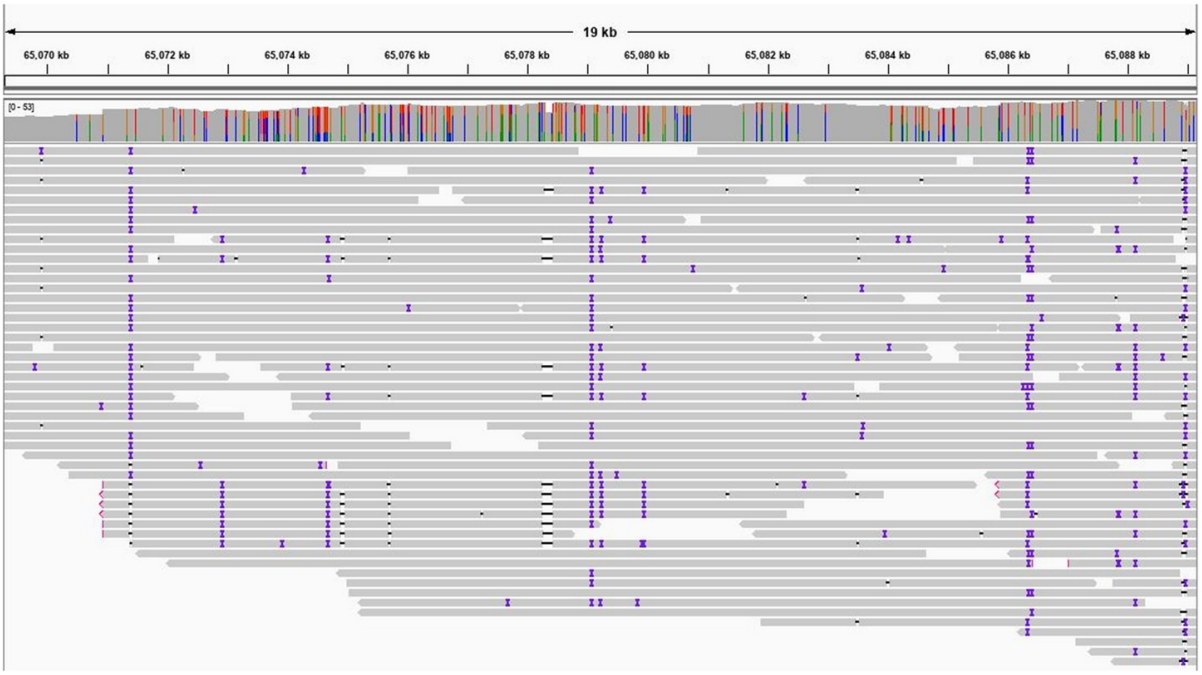

Figure 4.3. Chromosome 20 translocation. Snapshot showing alignments of long PacBio reads to the Ash1 genome, centered on the left end of the location in chromosome 20 (position 65,079,275) where a translocation occurred between chromosome 15 (GRCh38) and 20 (Ash1). The top portion of the figure shows the coordinates on chr20. Below that is a histogram of read coverage, and the individual reads fill the bottom part of the figure. The indels in the reads, shown as colored bars on each read, are due to the relatively high error rate of the long reads.

All of the genes that failed to map or that mapped partially were members of multi-gene families, and in every case, there was at least one other copy of the missing gene present in Ash1, at an average identity of 98.5%. Thus, there are no cases at all of a gene that is present in GRCh38 and that is entirely absent from Ash1. Three additional genes (2 protein coding, 1 lncRNA) mapped to two unplaced contigs, which will provide a guide to placing those contigs in future releases of the Ash1 assembly.

After mapping the genes onto Ash1, we extracted the coding sequences from transcripts that mapped completely (coverage equal to 100%), aligned them to the coding sequences from GRCh38, and called variants relative to GRCh38 (see the "Methods" section). Within the 35,513,365 bp in these protein-coding transcripts, we found 20,864 single-nucleotide variants and indels. We then compared these variants to the Genome in a Bottle (GIAB) benchmark set for HG002 [69]. 14,499 of these variants fell within the GIAB "callable" regions for high-confidence variants, although 3963 of these were in GIAB "difficult" repetitive regions, for which alignments are often ambiguous. Of the 10,536 variants not in these difficult regions, 10,456 (99.2%) agreed with the GIAB high-confidence variant set. In the difficult regions, 3804/3963 (96.0%) agreed with the GIAB set.

We then annotated the changes in amino acids caused by variants and incomplete mapping for all protein-coding sequences. Out of 124,238 protein-coding transcripts from 20,197 genes, 92,600 (74.5%) have 100% identical protein sequences. Another 26,566 (21.4%) have at least one amino acid change but yield proteins with the identical length, and 1,561 (1.3%) have frame-preserving mutations that insert or delete one or more amino acids, leaving the rest of the protein unchanged. Table 4.3 shows statistics on all of the changes in protein sequences. If a protein had more than 1 variant, we counted it under the most consequential variant, i.e., if a protein had a missense variant and a premature stop codon, we include it in the "stop gained" group.

Table 4.2. Comparison of coding sequences between Ash1 and GRCh38. Here, "insertion" means an insertion in Ash1 relative to GRCh38, and other terms are similarly referring to changes in Ash1 compared to GRCh38. "Truncated" indicates the transcript was only partially mapped. "Stop gained" refers to premature stop codons caused by a SNP.

Variant Type	Number of Coding Sequences
identical	92,600
mis-sense variant	26,566
in-frame deletion	956
in-frame insertion	605
frameshift variant	2,158
start lost	169
stop gained	416
stop lost	58
truncated	564
unmapped	138
Total	**124,230**

Of particular interest are those transcripts with variants that significantly disrupt the protein sequence and may result in loss of function. These include transcripts affected by a frameshift (2158), stop loss (58), stop gain (416), start loss (58), or truncation due to incomplete mapping (564). These disrupted isoforms represent 885 gene loci; however, 505 of these genes have at least 1 other isoform that is not affected by a disrupting variant. This leaves 380 genes in which all isoforms have at least one disruption.

PR1

The PR1 assembly used CHM13, the first truly complete human genome, for scaffolding. Thus, we used Liftoff to map all of the genes from CHM13 onto PR1, including protein-coding and noncoding RNA genes. The CHM13 annotation contains 37,670 genes in total, of which 19,829 protein-coding genes and 16,818 lncRNAs (36,647 genes) were mapped onto CHM13 from the GENCODE annotation of

GRCh38[48]. CHM13 also contains 804 additional paralogs (140 protein coding and 664 lncRNAs) and 219 additional rDNA genes not present in GRCh38 (See section 4.2.3). Because the CHM13 genome does not have a Y chromosome, we mapped the GENCODE genes from GRCh38's chromosome Y onto PR1 chromosome Y. Out of the 37,670 genes from CHM13 and 142 genes from GRCh38 chrY (37,812 total), Liftoff successfully mapped 37,743 (99.8%). Of the 69 unmapped genes (Supplementary Table 1 in Zimin, Shumate *et al.* 2021[11]) 42 are protein coding and 27 are noncoding. The vast majority of genes mapped well above Liftoff's minimum 50% threshold with 93% of genes mapping with ≥99% coverage and sequence identity.

Out of the 69 genes that failed to map, 29 aligned end-to-end with another copy of the gene present elsewhere in the assembly (*i.e.*, a paralog), suggesting that PR1 simply has fewer copies (Supplementary Table 1 in Zimin, Shumate *et al.* 2021). Another 28 genes had partial copies present in the assembly (see Methods). Of the 12 remaining unmapped genes, all but 3 genes mapped partially but did not meet the 50% minimum coverage and sequence identity threshold. The three genes completely missing from PR1 are all lncRNAs whose function is unknown.

We looked at all 86,335 protein-coding transcripts that were mapped from CHM13 to PR1 to determine if the protein sequence was preserved. In the vast majority of cases, the sequences either were identical or had nonsynonymous mutations that preserved the protein sequence length. Specifically, 71,699 transcripts (83.0%) had identical sequence, 13,544 (15.7%) had amino acid changes but identical lengths and an average protein sequence identity of 99.5%, and 828 (0.96%) had insertions or

deletions that preserved the reading frame. Only 196 transcripts had frame-shifting mutations, and 68 were truncated on one end or missing the start codon.

To identify genes with a higher copy number in PR1 than CHM13, we used an optional feature of Liftoff to identify additional paralogs. We found 12 additional paralogs including 8 paralogs of protein-coding genes and 4 paralogs of lncRNAs (Supplementary Table 2 in Zimin, Shumate *et al.* 2021). Six of these paralogs occur in tandem, defined as a gene that occurs within 100 kbp of another copy. All isoforms of the additional copies are 100% identical at the mRNA level to the original copy in CHM13. In general, a finding of additional paralogs is either the result of increased assembly completeness or copy number variation. Given that CHM13 is a complete, gap-free assembly, these 12 paralogs appear to represent genuine copy number variation between PR1 and CHM13. Also, worth noting here is that CHM13 contains 140 additional copies of protein-coding genes by comparison to GRCh38 all of which are also present in PR1.

Because GRCh38 is currently the primary human reference genome, we also mapped the annotation from GRCh38 onto PR1, using CHESS v2.2[22] just as we did with Ash1. We successfully mapped 42,172 out of 42,306 genes (99.7%) from the CHESS annotation. Seventy-three out of the 134 unmapped genes are protein coding and the other 61 are noncoding. We also identified 159 additional gene copies (paralogs) present in PR1 and missing from GRCh38. These include 30 paralogs of protein-coding genes and 129 paralogs of noncoding genes. The CHESS genes that failed to map, including all gene types, are shown in Supplementary Table 3 in Zimin, Shumate *et al.*

2021. All extra gene copies in PR1 compared to GRCh38, along with the gene names and chromosomal locations on PR1, are shown in Supplementary Table 4 in Zimin, Shumate *et al.* 2021.

T2T-CHM13

To provide an initial annotation, we used both the Comparative Annotation Toolkit (CAT)[64] and Liftoff to project the GENCODE v35[48] reference annotation onto the T2T-CHM13 assembly. Additionally, CHM13 Iso-Seq transcriptome reads were assembled into transcripts and provided as complementary input to CAT. A comprehensive annotation was built by combining the CAT annotation with genes identified only by Liftoff.

The draft T2T-CHM13 annotation totals 63,494 genes and 233,615 transcripts, of which 19,969 genes (86,245 transcripts) are predicted to be protein coding, with 683 predicted frameshifts in 385 genes (469 transcripts) (Supplementary Tables S1, S6, S8 and Supplementary Figure S3 in Nurk *et al.* 2021[12]). Only 263 GENCODE genes (448 transcripts) are exclusive to GRCh38 and have no assigned ortholog in the CHM13 annotation (Supplementary Tables S9 and S10 in Nurk *et al.* 2021). Of these, 194 are due to a lower copy number in the CHM13 annotation (Supplementary Figure S31 in Nurk *et al.* 2021), 46 do not align well to CHM13, and 23 correspond to known false duplications in GRCh38[70] (Supplementary Figure S32 in Nurk *et al.* 2021). The majority of these genes are non-coding and associated with repetitive elements. Only 4 are annotated as being medically relevant (*CFHR1*, *CFHR3*, *OR51A2*, *UGT2B28*) [71] all of

which are due to lower copy number, and the only protein coding genes that align poorly are immunoglobulin and T-cell receptor genes, which are known to be highly diverse.

In comparison, a total of 3,604 genes (6,693 transcripts) are exclusive to CHM13 (Supplementary Tables S11 and S12 in Nurk *et al.* 2021). Most of these genes represent putative paralogs and localize to pericentromeric regions and the short arms of the acrocentrics, including 876 rRNA transcripts. Only 48 of the CHM13-exclusive genes (56 transcripts) were predicted solely from the de novo assembled transcripts. Of all genes exclusive to CHM13, 140 are predicted to be protein coding based on their GENCODE paralogs and have a mean of 99.5% nucleotide and 98.7% amino acid identity to their most similar GRCh38 copy (Supplementary Table S13 in Nurk *et al.* 2021). While some of these additional paralogs may be present (but unannotated) in GRCh38, 1,956 of the genes exclusive to CHM13 (99 protein coding) are in regions with no primary alignment to GRCh38 (Supplementary Table S11 in Nurk *et al.* 2021). A broader set of 182 multi-exon protein coding genes fall within non-syntenic regions, 36% of which were confirmed to be expressed in CHM13 [72].

4.3 Methods

Ash1

Aligning long PacBio reads for validation

We downloaded a recently released set of PacBio HiFi reads, generated on the Sequel II System, from the HG002 Data Freeze (v1.0) at Human Pangenome Reference Consortium github site (https://github.com/human-

pangenomics/HG002_Data_Freeze_v1.0#hg002-data-freeze-v10-recommended-downsampled-data-mix, also available from the NCBI SRA database under accessions SRX7083054, SRX7083055, SRX7083058, SRX7083059). These data, which were not used in our assembly of Ash1, were size selected for 15-kb and 20-kb fragments, and they represent ~34x genome coverage of the genome. We aligned them to Ash1 v1.7 genome using bwa-mem[73] with default parameters. We filtered the alignments using samtools[74] to include only reads aligning with a quality of 40 and above.

Comparing variants in mapped genes

Gffread [54] was used to extract the coding sequences from GRCh38 and Ash1. Sequences were aligned pairwise using the EMBOSS Stretcher command line interface [75] from Biopython 1.75. The alignments were used to determine the GRCh38 location, sequence, and functional consequence of each variant. When comparing GIAB HG002 V3.3.2 benchmark set, we excluded any transcripts that did not map with an alignment coverage of 100%. We compared the variants to the benchmark set using vcfeval from RealTimeGenomics tools[76]. We used bedtools[77] to intersect the false positive variants in Ash1 genes with the union set of difficult regions from GIAB (ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/release/genome-stratifications/v2.0/GRCh38/union/GRCh38_alldifficultregions.bed).

Aligning transcripts between GRCh38 and Ash1

To compute the cumulative distributions shown in Figure 4.1 and 4.2, the mRNA sequences of the Ash1 transcripts and GRCh38 transcripts were extracted using

gffread. The sequences were then aligned pairwise using the EMBOSS Stretcher command line interface from Biopython 1.75, and the resulting alignments were used to calculate the percent of GRCh38 transcript lengths covered and the percent identity between the pairs of transcripts.

PR1

To annotate the PR1 genome, we mapped the CHM13, GRCh38 chromosome Y, and CHESS v2.2 annotations using Liftoff version 1.6.1 with the following parameters: -copies -polish -exclude_partial -chroms <chroms.txt>. After the initial mapping, we aligned every unmapped transcript to every mapped transcript using Blastn[78] to determine if the unmapped genes were copies of mapped genes (where we define a copy as an end-to-end alignment with a mapped transcript). We also re-ran Liftoff allowing for overlapping genes (-overlap 1.0). By comparing the results to the initial Liftoff output, we were able to identify genes that only mapped when allowed to overlap another gene. These overlapping genes are either complete copies of one another if they map to exactly the same locus or partial copies if they map to different but overlapping loci. The overlapping genes were identified by intersecting the Liftoff-generated annotation file with itself using bedtools intersect[77]. The output file was then filtered to remove self-overlaps, and genes identified by this process were classified as partial copies. We further attempted to identify partial copies at the protein level by using gffread to extract the protein sequences and blastp[78] to align them to mapped proteins, using an *e*-value threshold of 10^{-6} to filter the results.

T2T-CHM13

Annotation

The Cactus[79] alignment between the v1.0 assembly and the primary contigs of GRCh38, with chimp as an outgroup, was created with the following command:

```
cactus aws:us-west-2:t2t-jobstore-chm13 cactus-config-
chm13- t2t.draft_v1.v2.txt t2tChm13.draft_v1.v2.hal --
maxCores 80 -- binariesMode local
```

Using the following config file for cactus:

```
 Chimp:0.00655,(GRCh38:0.0005,CHM13:0.0005)); Chimp
GCF_002880755.1_Clint_PTRv2_fixed.fa GRCh38
GRCh38.primary.fa CHM13 t2t-chm13-v1.0.fa
```

Iso-Seq reads were aligned using minimap2[32] using the following command:

```
minimap2 -ax splice -f 1000 --sam-hit-only --secondary=no -
-eqx - K 100M -t 4 --cap-sw-mem=3g mmdb/0.mmi
iso_fastas/0_0.fasta
```

Stringtie2[21] assembled the transcriptome using available Iso-Seq reads:

```
stringtie -p 8 chm13_1.t2t.sorted.bam.filtered.bam -L >
chm13_1.t2t.TM.stringtie.gtf
```

CAT[64] v2.2.1 (commit 96e7550f22387a669f0b98dfc0c94be825192e24) was run in TransMap[80] mode using the following command:

```
 luigi --module cat RunCat --hal=t2tChm13.draft_v1.v2.hal -
- target-genomes='("CHM13",)' --ref-genome=GRCh38 --
workers=10 -- config=cat.t2t.draft_v1.full.isoseq.config --
work-dir work-chm13- t2t --out-dir out-chm13-t2t --local-
scheduler --assembly-hub -- maxCores 5 --binary-mode local
```

Using GENCODEv35[48] and following config file for CAT:

```
 [ANNOTATION] GRCh38 = gencode.v35.annotation.gff3.noPAR
CHM13 = CHM13.TM.stringtie.merged.gff3 [ISO_SEQ_BAM] CHM13
= data/chm13_1.t2t.sorted.bam,data/chm13_2.t2t.sorted.bam
```

We removed genes from the CAT annotation that had overlapping annotations from multiple genes in the same family, leaving the gene that was correct based on synteny. The Liftoff annotation was created with the following command using version 1.6.0:

```
liftoff chm13.draft_v1.0.fasta GRCh38.fa -sc 0.95 -copies -
g gencode.v35.annotation.gff3 -polish -chroms chroms.txt
```

To create the final annotation, we complemented the CAT result with missed GENCODE genes and putative additional paralogs (with minimum sequence identity of 95%) from the Liftoff annotation. Only predictions that did not overlap any CAT annotations were added. The annotation set on the v1.0 assembly was lifted over to the v1.1 assembly using liftover with the command:

```
liftOver -gff CHM13.combined.v4.gff3 v1_to_v1.0423.chain
CHM13.combined.v4.liftover.v1.1.gff3 unmapped.txt
```

The rDNAs were annotated by mapping an assembly of an rDNA unit isolated from chromosome 21[81] onto v1.1 with Liftoff. Using GenBank entry KY962518.1, the rDNA sequence was obtained and a gff3 file created with the coordinates of the 45S, 18S, 5.8S, and 28S subunits. Liftoff was then run with the following command to annotate all rDNAs within the assembly

```
liftoff chm13.draft_v1.1.fasta KY962518.1.fasta -g
KY962518.1.gff3 -copies -sc 0.95 -mm2_options="-N 300"
```

All annotations that have been lifted over and that overlapped the newly added rDNA regions were removed. The rDNA annotations (876 new genes) were added to create a final annotation set.

I

Identifying falsely duplicated sequence in GRCh38

To identify falsely duplicated regions in GRCh38, we compared copy number estimates of GRCh38 to copy number estimates of 268 genomes from the SGDP dataset using short reads, using a method analogous to comparative read-depth approaches described previously[82,83]. We first averaged the copy number estimates for each genome across 1 kbp windows. For each 1 kbp region, we flagged it as a potential false duplication if the copy number in GRCh38 was greater than the copy number in 99% of the other genomes. Flagged regions were assigned a value of 1 and unflagged regions were assigned a value of 0. To filter the flagged regions, we used a median filter approach with a window size of 3 kbp, where the binary value of each 1 kbp region was replaced with the median value of the complete window. We then merged all adjacent flagged regions and reported the start and end coordinates with respect to T2T-CHM13. To find the corresponding locations of the duplications on GRCh38, we used minimap2[32] version 2.17-r941 with parameter -p 0.25. Some regions mapped to more than two locations on GRCh38 due to true SDs in the genome. We curated these regions with more than two alignments and identified the incorrect region(s) as the region(s) that did not have an assembly-assembly alignment from T2T-CHM13 or the HG002 haplotype. We identified the affected, correct region as the region that aligned most closely to the T2T-CHM13 region, which also had reduced HG002 read coverage. Upon curation of the regions with only two alignments on GRCh38, we selected as correct the region that was on the same chromosome arm as the corresponding T2T-CHM13 region. When both regions were on the same chromosome arm, we selected as correct the region that was not adjacent to or between gaps in GRCh38. One false duplication was a tandem duplication, and we arbitrarily selected one copy as correct.

Upon curation, we also removed one small 8 kb region (chr19:14,359,000-14,367,000 on T2T-CHM13) that was incorrectly identified as falsely duplicated.

Chapter 5: Annotation of an improved bread wheat assembly

Parts of chapter 5 previously have appeared in:

M. Alonge*, A.Shumate*, D. Puiu, A.V. Zimin, S.L. Salzberg (2020). “Chromosome-Scale Assembly of the Bread Wheat Genome Reveals Thousands of Additional Gene Copies.” *Genetics,* 599-608, 216(2).

* indicates equal contribution

Additional Contributors:

This work was conducted jointly with Michael Alonge and the other authors listed above. This publication details the assembly and annotation of a bread wheat genome. My main contribution was the annotation, so the assembly (conducted by Michael Alonge) is described only in the introduction and discussion of this chapter. Refer to the publication for more details about the assembly results and methods. In the following chapter, the introduction and discussion are from the publication and were written by Michael Alonge. Michael Alonge also created figures 5.4, 5.6 and 5.7 and wrote the ‘Gene duplications affecting traits’ section and the associated methods.

5.1 Introduction

Bread wheat (*Triticum aestivum*) is a crop of significant worldwide nutritional, cultural, and economic importance. As with most other major crops, there is a strong interest in applying advanced breeding and genomics technologies toward crop improvement. Key to these efforts are high-quality reference genome assemblies and associated gene annotations, which are the foundations of genomics research. However, the bread

wheat genome has some notable features that make it especially technically challenging to assemble. One such feature is allohexaploidy (2*n* = 6× = 42, AABBDD), a result of wheat's dynamic domestication history[84,85]. This polyploidy results from the hybridization of domesticated emmer (*Triticum turgidum*, AABB) with *Aegilops tauschii* (DD). Domesticated emmer—also an ancestor of durum wheat—is itself an allotetraploid resulting from interspecific hybridization between *Triticum urartu* and a relative of *Aegilops speltoides*.

The resulting bread wheat genome is immense, with flow cytometry studies estimating the genome size to be ~16 Gbp[86]. As with most other large plant genomes, repeats, including mostly retrotransposons, make up the majority of the genome, which is estimated to be ~85% repetitive[87]. These repeats make this genome especially difficult to assemble, even given the recent improvements in long-read sequencing and algorithmic advancements in genome assembly technology. Nonetheless, early efforts were made to establish *de novo* reference genome assemblies for wheat. In 2014, the International Wheat Genome Sequencing Consortium (IWGSC) used flow cytometry-based sorting to sequence and assemble individual chromosome arms, thus removing the repetitiveness introduced by homologous chromosomes (IWGSC 2014). In spite of this approach, this short-read based assembly was highly fragmented, and only reconstructed ~10.2 Gbp of the genome. Subsequent short-read assemblies using alternate strategies were also developed by the community, though each also struggled to achieve contiguity and completeness[88,89].

In 2017, we released the first-ever long-read-based assembly for bread wheat (Triticum_aestivum_3.1), representing the Chinese Spring variety[90]. With an N50 contig size of 232.7 kbp, Triticum_aestivum_3.1 was far more contiguous than any previous assembly of bread wheat, and with a total assembly size of 15.34 Gbp, it reconstructed the highest percentage of the expected wheat genome size of any assembly. Though this assembly provided a more complete representation of the Chinese Spring genome, its contigs were not mapped onto chromosomes, and, notably, it did not include gene annotation.

In 2018, the IWGSC published a chromosome-scale reference assembly and associated annotations for bread wheat (IWGSC CS v1.0, Chinese Spring), providing the best-annotated reference genome yet[87]. Because that assembly was entirely derived from short reads, it was less complete and more fragmented than Triticum_aestivum_3.1, having a total size of 14.5 Gbp and an N50 contig size of 51.8 kbp. However, a collection of long-range scaffolding data, including physical (BACs, Hi-C), optical (Bionano), and genetic maps, enabled most of the assembled scaffolds to be mapped onto wheat's 21 chromosomes. These pseudomolecules served as a foundation for comprehensive *de novo* gene and repeat annotation, facilitating investigations into the genomic elements that drove the evolution of genome size, structure, and function in wheat.

Here, we used the IWGSC CS v1.0 assembly (GenBank accession GCA_900519105.1) to inform the scaffolding and annotation of the more complete Triticum_aestivum_3.1 assembly. The new assembly, Triticum_aestivum_4.0, contains 1.1 Gbp of additional

nongapped sequence compared to IWGSC CS v1.0, while localizing 97.9% of sequence to chromosomes. Comparative analysis revealed that Triticum_aestivum_4.0 more accurately represents the Chinese Spring repeat landscape, which is heavily collapsed in IWGSC CS v1.0. Our more-complete assembly allowed us to anchor ~2000 genes that were previously annotated on unlocalized contigs in IWGSC CS v1.0. We also found 5799 additional gene copies in Triticum_aestivum_4.0, showing extensive collapsing of gene duplicates in the IWGSC CS v1.0 assembly. We highlighted specific examples of these extra gene copies, including at the *Ppd-B1* locus, where Triticum_aestivum_4.0 accurately reflects the expected four copies of pseudo-response regulator (PRR) genes influencing photoperiod sensitivity. We additionally found three extra copies of a MADS-box transcription factor gene in T4, demonstrating the potential to find new gene copy number variants (CNVs) that influence traits.

5.2 Results

Annotation

We mapped the IW v1.1 high-confidence annotation onto T4 using Liftoff [9]. Out of 130,745 transcripts from 105,200 gene loci annotated on primary chromosomes in IW, we successfully mapped 124,579 transcripts from 100,831 gene loci. We define a transcript as successfully mapped if the mRNA sequence in T4 is at least 50% as long as the mRNA sequence in IW. However, the vast majority of transcripts greatly exceed this threshold, with 92% of transcripts having an alignment coverage of 98% or greater (Figure 5.1). Sequence identity is similarly high with 92% of transcripts aligning at an identity of 95% or greater (Figure 5.2). Of the transcripts that failed to map, 4634 had a

partial mapping with an alignment coverage <50%, and the remaining transcripts failed to map entirely.

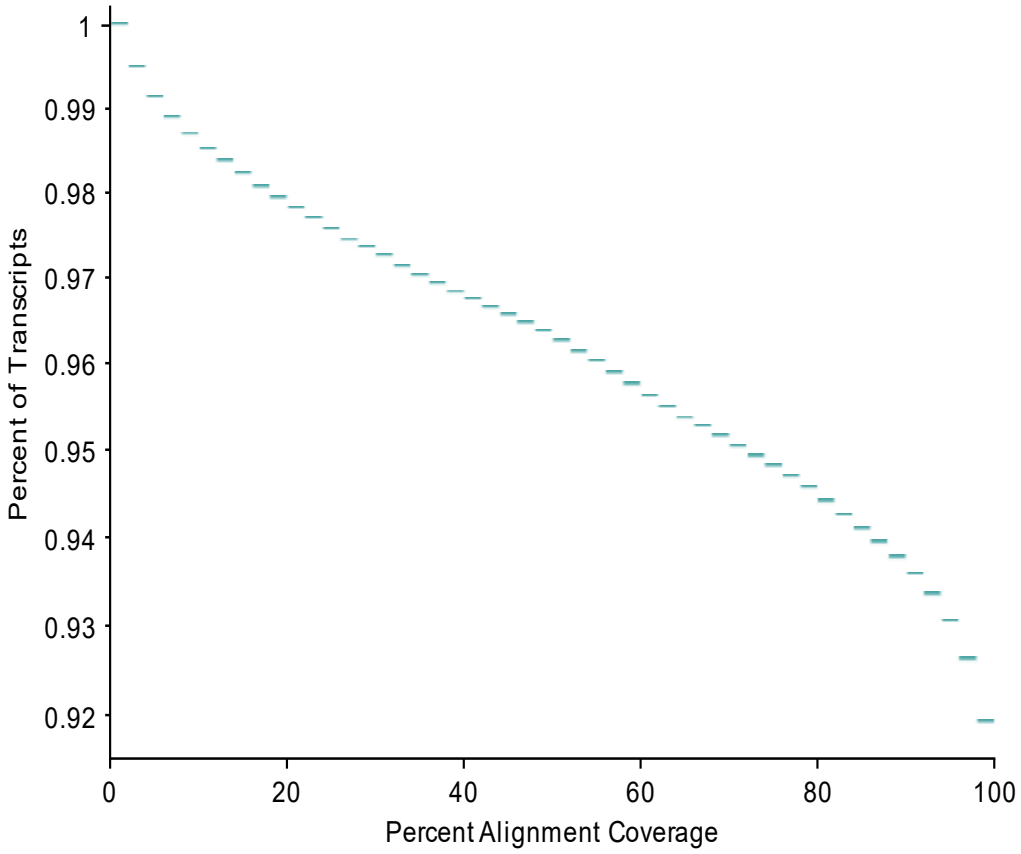

Figure 5.1. T4 cumulative distribution of coverage. Cumulative distribution showing how much of the IW transcripts map onto T4. The y-axis shows the fraction of transcripts with percent coverage greater than or equal to coverage on the x-axis.

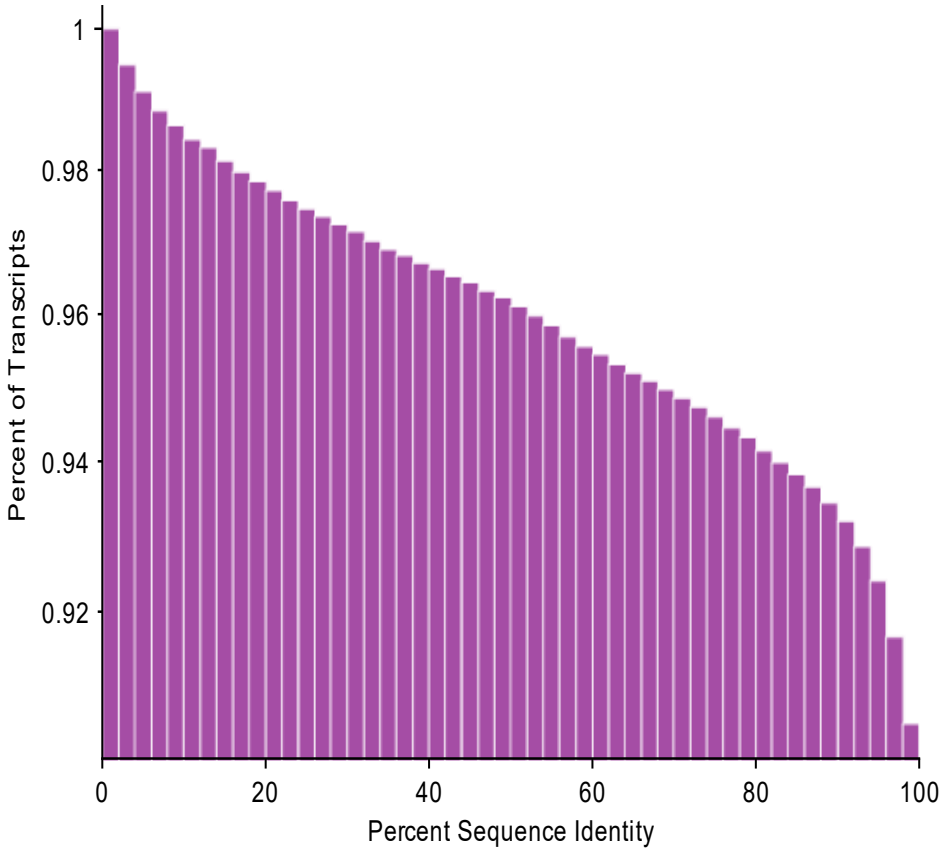

Figure 5.2. T4 cumulative distribution of sequence identity. Cumulative distribution showing the sequence identity of IW transcripts mapped onto T4. The y-axis shows the fraction of transcripts with sequence identity greater than or equal to sequence on the x-axis.

As expected, we observed strong gene synteny between T4 and IW (Figure 5.3). Of the 100,831 mapped IW genes, 96,148 mapped to the same chromosome in T4. The remaining 4683 mapped to a different chromosome after failing to map to their expected chromosome. There is a clear pattern showing many of these genes mapped to a similar location on the same chromosome of a different subgenome. We also found that the sequence identity of genes mapped to different chromosomes is much lower, with an average identity of 90.7% compared to 99.3% in genes mapped to the same

chromosome. We therefore hypothesize that these genes are missing in the T4 assembly and have instead mapped to paralogs in T4 that are not annotated in IW.

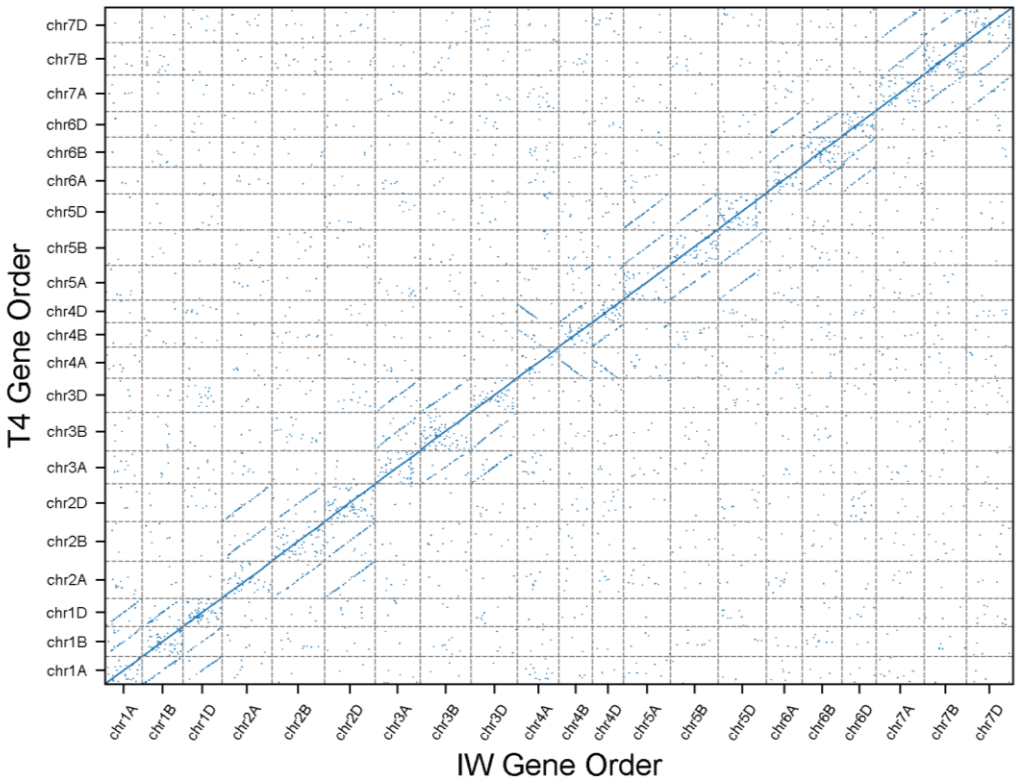

Figure 5.3. IW and T4 gene order. Gene synteny dot plot showing the ordinal position for each gene in IW on the x-axis and the ordinal position in T4 on the y-axis.

The IW v1.1 annotation also contains 2691 genes annotated on unplaced contigs ("chrUn"). Using Liftoff, we were able to map 2001 of these genes onto a primary chromosome in T4 (Figure 5.4); 1767 genes were confidently placed with a sequence identity of at least 98% while the remaining 234 mapped with a lower identity (Supplementary Table S2 in Alonge, Shumate *et al.* 2020[13]). To control for differences in annotation pipelines between IW and T4, we used Liftoff to map chrUn genes onto

the primary IW chromosomes to look for additional, unannotated, gene copies. Of the 2001 chrUn genes mapped to T4 pseudomolecules, 78 of these were also mapped to primary IW chromosomes. This suggests that ≥1923 genes were placed due to improved assembly completeness rather than differences in annotation methods.

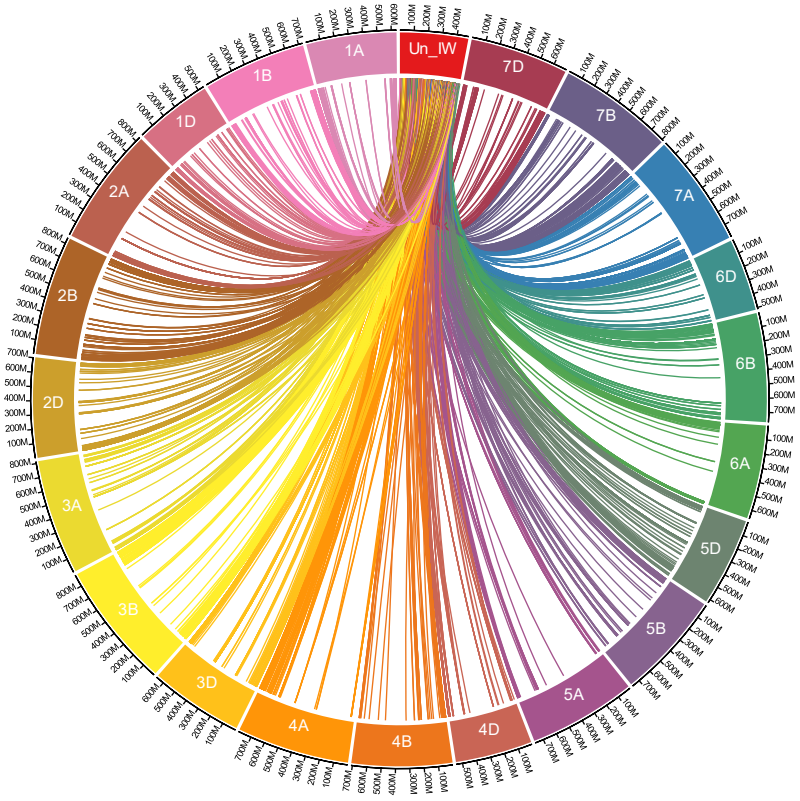

Figure 5.4. Circos plot of previously unplaced genes. Circos plot (http://omgenomics.com/circa/) showing where unplaced genes in IW were mapped in T4.

After mapping the IW v1.1 annotation onto T4, we used Liftoff to look for additional gene copies in T4. We required 100% sequence identity in exons and splice sites to map a gene copy. We found 5799 additional gene copies in T4 that are not annotated in IW

v1.1. Of these, 4158 genes have one extra copy, and 567 genes have two or more additional copies, with a maximum of 84 additional copies (Figure 5.5).

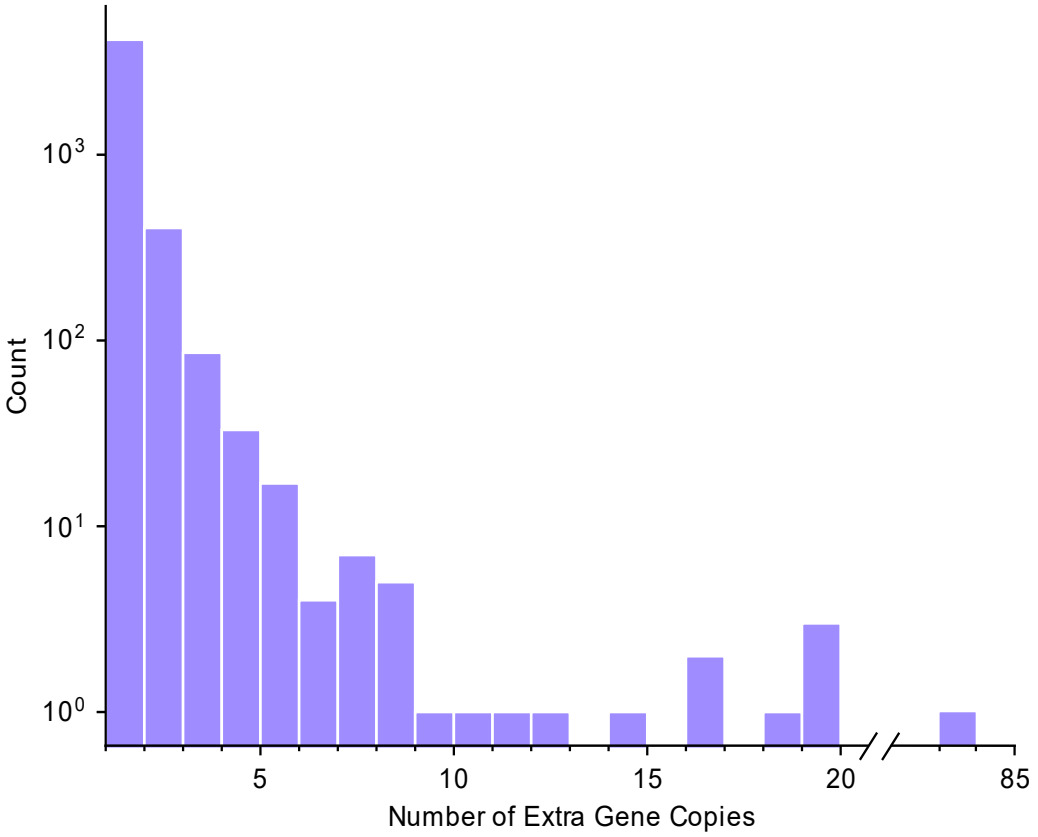

Figure 5.5. Copy number histogram. Histogram depicting the distribution of the number of additional gene copies found in T4.

IW collapsed most gene copies on the same chromosome rather than across homeologous chromosomes, with 4062 of the 5799 additional gene copies occurring on the same chromosome, and 97 copies occurring on the same chromosome of a different subgenome (Figure 5.6); 915 gene copies were placed on different chromosomes. The remaining 725 are extra copies of chrUn genes placed on

chromosomes. The location and functional annotation of all additional copies is provided in Supplementary Table S3 in Alonge, Shumate *et al.* 2020. As was done for unplaced genes, we also looked for additional IW gene copies present elsewhere in IW. Of our 5799 additional gene copies, 159 were also present in IW, suggesting that at least 5640 of T4 copies are strictly the result of improved assembly completeness.

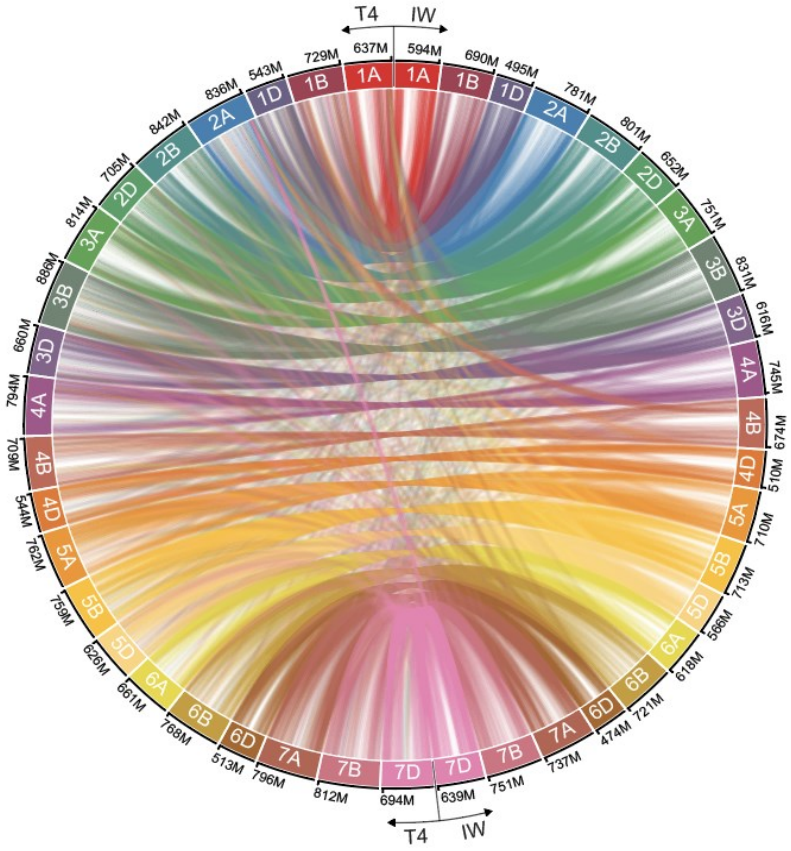

Figure 5.6. Circos plot of extra gene copies. Circos plot (http://omgenomics.com/circa/) showing the locations of all additional gene copies. Lines are drawn from the location of the gene in IW on the right half of the diagram to the location of each copy in T4 on the left half.

Gene duplications affecting traits

We searched T4 for specific examples of functionally relevant gene duplications previously collapsed or missing in IW. We focused on the *Ppd-B1* locus on chr2B because copy number variation of PRR genes at this locus underlies variation in photoperiod sensitivity among hexaploid wheat varieties[91]. Others have shown that the Chinese Spring variety has four PRR genes at the *Ppd-B1* locus, with one of the copies being truncated[92]. Because the entire ~200 kbp Chinese Spring *Ppd-B1* locus was previously cloned and sequenced, we were able to assess this region had been accurately assembled in both T4 and IW. IW lacks any PRR genes at the *Ppd-B1* locus, with fragments of three of the four expected paralogs (*TraesCSU02G196100*, *TraesCSU02G221500*, *TraesCSU02G199500*) residing on unplaced chrUn sequence. In contrast, T4 localizes four PRR genes (*T4021472*, *T4021473*, *T4021474*, and *T4021475*) at *Ppd-B1*, matching the expected Chinese Spring copy number state. Alignment of this T4 locus to the known Chinese Spring *Ppd-B1* sequence indicated that the entire locus had been accurately assembled, even correctly representing the three, highly similar, intact PRR genes (Figure 5.7). The successful assembly of *Ppd-B1* served as a validation that T4 accurately resolves duplications with high sequence similarity.

The successful resolution of the *Ppd-B1* locus suggested that new functionally relevant CNVs may be discovered among the large number of localized or duplicated genes in T4. One notable example was a MADS-box transcription factor

gene, *TraesCS6A02G022700*, which had three additional tandem copies (T4 genes *T4081597*, *T4081598*, *T4081599*, and *T4081600*) on T4 chr6A (Figure 5.8).

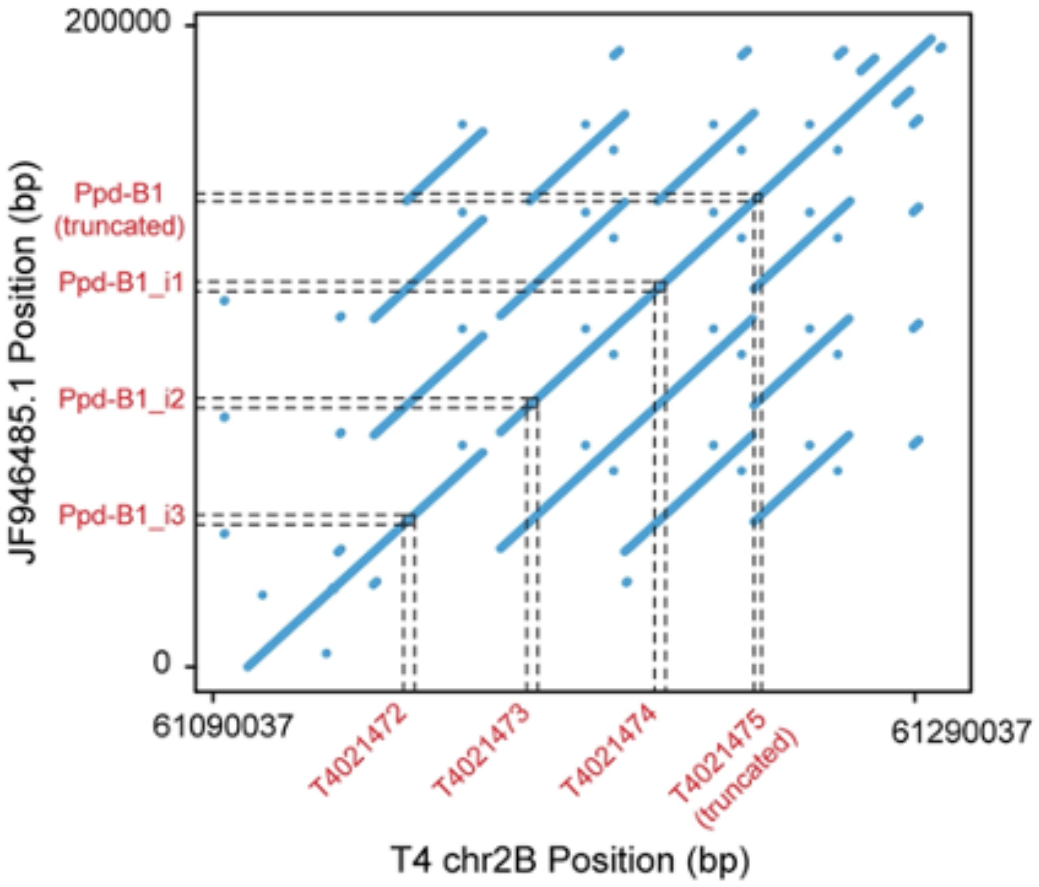

Figure 5.7. *Ppd-B1* dot plot. Dot plot depicting maximal exact matches (MEMs) between T4 *Ppd-B1* (x-axis) and a publicly available Chinese Spring *Ppd-B1* sequence (GenBank accession JF946485.1) (y-axis). Dashed lines indicate the co-linear positions of four PRR genes (red labels).

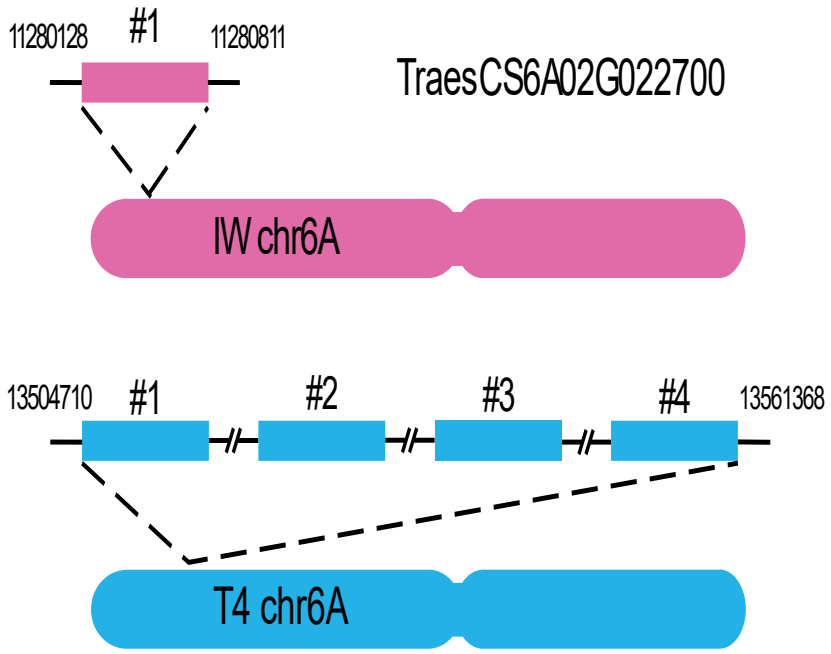

Figure 5.8. Extra copies of MADS-box genes. Diagram of the MADS-box transcription factor gene, *TraesCS6A02G022700,* present in 3 additional tandem copies in T4 as relative to IW. Ideograms are not drawn to scale.

MADS-box transcription factors are known to influence traits such as flowering time and floral organ development[93,94]. Furthermore, MADS-box gene duplications can quantitatively impact gene expression and domestication phenotypes in a dosage-dependent manner[95]. To provide further evidence that this gene is part of a collapsed repeat in IW, we aligned Chinese Spring Illumina reads to IW and calculated the coverage across the gene ±50 kbp of flanking sequence. We observed a spike in coverage indicating a collapsed repeat in IW containing *TraesCS6A02G022700* (Figure 5.9). We further note that this region contains 10,205 bp of gap sequence, suggesting that this locus had been misassembled in IW. This duplication of a MADS-box transcription factor gene, as well as our analysis of the *Ppd-B1* locus, highlights how T4,

with its superior genome completeness, resolves functionally relevant genic sequence previously misassembled, missing, or unlocalized in IW.

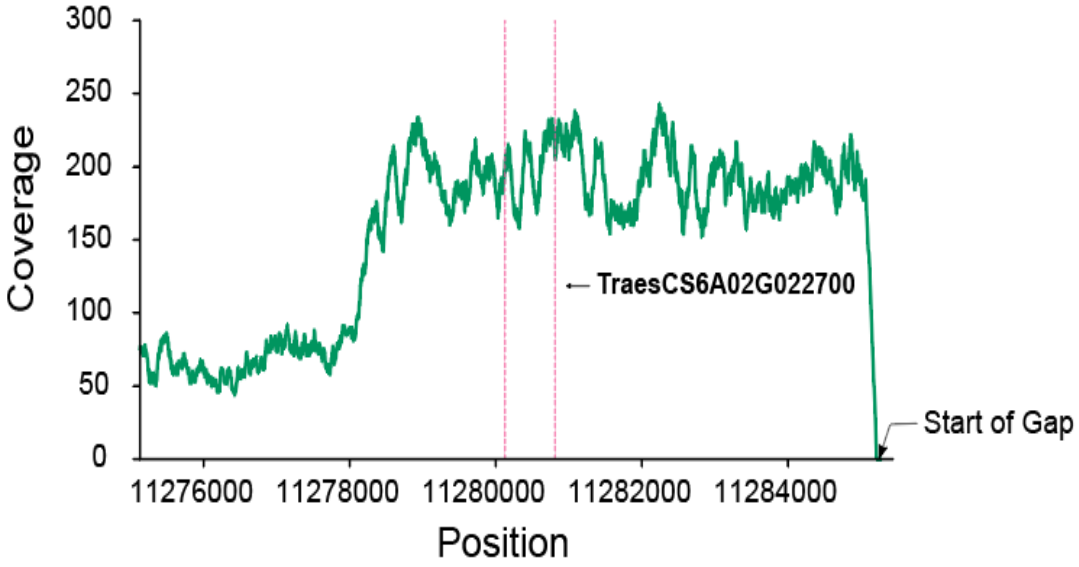

Figure 5.9. TraesCS6A02G02270 short-read coverage. Plot of the short-read coverage in IW starting 5kb upstream of *TraesCS6A02G02270* and extending to the first gap downstream of the gene. The pink dashed lines show the location of the gene.

5.3 Discussion

In one critical aspect, the bread wheat genome exemplifies the challenge of eukaryotic genome assembly. Repeats, which remain difficult to assemble, are pervasive in this transposon-rich allohexaploid plant genome. Therefore, the accurate and complete resolution of the bread wheat genome and the subsequent study of genomic structure especially depends on high-quality data and advanced genome assembly techniques. In 2017, we published the first near-complete and highly contiguous representation of the bread wheat genome (Triticum_aestivum_3.1), demonstrating the value of long reads for wheat genome assembly[90]. In our efforts described here, we used

Triticum_aestivum_3.1 as our foundation, while leveraging the strengths of the IWGSC CS v1.0 reference genome to establish the most complete chromosome-scale and gene-annotated reference assembly yet created for bread wheat. By scaffolding and annotating our contigs, we created the genomic context needed to quantify and qualify the completeness of the Triticum_aestivum_4.0 assembly, especially relative to its predecessors. Compared to the IWGSC CS v1.0 assembly, Triticum_aestivum_4.0 resolves more repeat sequence, exemplified by the improved centromere localization and by the many additional gene copies. The discovery of these extra gene copies, as well as the localization of 2001 previously unplaced genes, also demonstrates how Triticum_aestivum_4.0 provides an enhanced representation of Chinese Spring genic sequence.

Gene CNVs are pervasive in hexaploid wheat and are associated with traits such as frost tolerance (*Fr-A2*), vernalization requirement (*Vrn-A1*), and photoperiod sensitivity (*Ppd-B1*)[96–99]. These and other CNVs contributed to the adaptive success of domesticated wheat, which now thrives in diverse conditions and geographies. This is exemplified by the *Ppd-B1* locus, where variation of PRR gene copy number influences photoperiod sensitivity. Our successful assembly of the *Ppd-B1* locus, which was unanchored and incomplete in IWGSC CS v1.0, highlights a specific example where our improved assembly accurately reflected a known CNV genotype in Chinese Spring. This validation suggests that other functional gene duplications may also be directly encoded in the Triticum_aestivum_4.0 assembly and identifiable by our annotation of extra gene copies. We indicated one such potential candidate, the MADS-box transcription factor gene, which appears with three extra copies in Triticum_aestivum_4.0. We expect that

further investigation of the extensive gene duplications presented in this work will provide additional insights into the role of CNVs in wheat phenotypes.

Structural variants (SVs), including CNVs, comprise a vast source of natural genetic variation influencing traits. As sequencing technologies continue to advance, plant scientists are increasingly using pan-genome analyses to study genome structure among diverse varieties and ecotypes[100–102]. These studies rely especially on structurally accurate reference genomes to discover SVs. Our work introduces Triticum_aestivum_4.0 as an improved reference genome resource ideal for future structural variant analyses in wheat. Furthermore, our comparative genomics analysis showed that a substantial portion of the Chinese Spring genome was collapsed, missing, or misrepresented when assembled with short reads. This emphasizes the utility of long reads in future wheat pan-genome analyses, where structural accuracy is key. Generally, our work provides a preview of the computational genomics analyses that are possible with an accurate wheat reference genome.

5.4 Methods

Annotation

We used Liftoff to annotate the T4 genome using the IW v1.1 gene models. Genes were aligned to their same chromosome in T4 using BLASTN v.2.9.0[103] (-soft_masking False -dust no -word_size 50 -gap_open 3 -gapextend 1 -culling_limit 10). The blast hits were filtered to include only those that contained one or more exons. For each gene, the optimal exon alignments were chosen according to sequence identity and concordance with the exon/intron structure of the gene model in IW. These alignments were used to

define the boundaries of each exon, transcript, and gene in T4. We excluded any transcripts that did not map with at least 50% alignment coverage. Any genes without at least one mapped isoform were then aligned against the entire T4 genome using BLASTN with the same parameters and placed given they did not overlap an already placed gene.

To place the chrUn genes, we aligned the genes to the entire T4 genome using the same parameters. We excluded any transcripts that did not meet the 50% alignment coverage threshold or overlapped an already annotated gene.

To find additional gene copies, we aligned all genes (query) to the complete T4 genome (reference) using BLASTN v2.9.0[103] (-soft_masking False -dust no -word_size 50 -gap_open 3 -gapextend 1 -culling_limit 100, qcov_hsp_perc 100). The notable differences in these parameters are qcov_hsp_perc, which requires 100% query coverage, and culling_limit, which has been increased from 10 to 100 to increase the number of reported alignments for genes with a highly increased copy number. We excluded any alignments that did not have 100% exonic sequence identity or overlapped a previously placed gene. We used gffread to filter out genes with noncanonical splice sites[54].

Finally, using the same methods as described for high confidence genes above, we also used Liftoff to map the IW v1.1 low-confidence annotation onto T4. We successfully mapped 152,900 out of 161,537 low-confidence genes. Another 1581 genes mapped partially below the 50% alignment coverage threshold.

Ppd-B1 haplotype comparison

To find the approximate location of the *Ppd-B1* locus in the T4 and IW assemblies, we aligned a *Ppd-B1* PRR gene sequence (GenBank accession DQ885757.1) to T4 and IW with blastn v2.6.0 (-perc_identity 95) [91]. No matches were found on IW chr2B, though partial matches were found on chrUn. In contrast, four strong matches were found on T4 chr2B, corresponding to genes *T4021472*, *T4021473*, *T4021474*, and *T4021475*. We also aligned the entire Chinese Spring haplotype for this locus, which had been previously cloned and sequenced (GenBank accession JF946485.1), to T4 using blastn v2.6.0 (-perc_identity 95)[92]. We used these alignments to approximately define the genomic coordinates of *Ppd-B1* in T4. In order to further validate the accuracy of this locus in T4, we aligned the GenBank JF946485.1 sequence to the T4 locus ±10 kbp flanking sequence in order to find pairwise maximal exact matches (MEMs) at least 50 bp in length. These alignments are depicted in Figure 5.7 and were generated with mummer v3.23 (-maxmatch -l 50 -b -c). Prior to alignment, the GenBank JF946485.1 sequence was reverse complemented in order to refer to the same strand as our T4 chr2B.

Because the PRR gene annotations used to define T4 *Ppd-B1* PRR genes were incomplete in IW, they were also initially incomplete in T4. To correctly annotate these T4 PRR genes, we used Liftoff to lift-over the GenBank JF946485.1 PRR gene annotations to T4. These genes are labeled *T4021472*, *T4021473*, *T4021474*, and *T4021475* in the final annotation.

Chapter 6: LiftoffTools: a toolkit for comparing genes lifted between genome assemblies.

6.1 Introduction

As we have seen in the previous 2 chapters, our work on gene annotation goes beyond just simply lifting over the reference annotation with Liftoff[9]. To gain additional biological insight about the new genome assembly, we have conducted analyses such as identifying variants and their effects on genes, comparing gene order, and evaluating extra genes and missing genes. All of these analyses presented in the previous 2 chapters were done somewhat manually, requiring a variety of different tools and custom scripts. Automating these steps will greatly reduce the time and effort needed to do this analysis, which will become increasingly important as gene annotation lift-over becomes more routine. To this end, we introduce LiftoffTools which is a toolkit to compare genes mapped from one assembly to another. LiftoffTools provides 3 different modules for comparing genes. The first identifies variants in protein-coding genes and their effects on the genes. The second compares the gene synteny, and the third clusters genes into groups of paralogs to evaluate the expansion and collapse of gene families. While LiftoffTools is designed to analyze the output of Liftoff, it is also compatible with the output of other annotation lift-over tools such as UCSC liftOver[39]. Here we provide a description of each module and a simple proof of concept example

using annotations lifted from a reference yeast genome assembly (GCA_000146045.2) to a target yeast assembly (GCA_003086655.1) with Liftoff v1.6.3.

6.2 Features

LiftoffTools offers 3 modules to compare annotations. The input required for all 3 is the sequences of the reference and target assemblies (in FASTA format), and the annotation of the reference and target assemblies (in GFF3 or GTF format). The target annotation can be derived from other lift-over tools besides Liftoff as long as the feature IDs in the reference and target annotations are the same.

Variants

This module calculates the DNA sequence identity of transcripts in the reference genome and the corresponding transcript in the target genome and identifies variants and their effect on the gene. First, we globally align the nucleotide sequences of the reference transcripts to the target transcripts using the Needleman-Wunsch algorithm implemented by Parasail[104],which is a single instruction/multiple data (SIMD) C library for sequence alignment. If the transcript has an annotated CDS, we also align the protein sequences again using Parasail. We then identify mismatches and gaps in the alignments and evaluate the effect on the protein sequence. The potential effects we look for are synonymous mutations, nonsynonymous mutations, in-frame deletions, in-frame insertions, start lost, 5' truncations, 3' truncations, frameshifts, and stop gained. For all transcripts we output the percent identity at the nucleotide level. For protein-coding transcripts we also output the protein percent identity and the variant effect if

applicable. If there is more than one variant, we report only the most severe. For example, if a transcript has a synonymous mutation and a frameshift mutation, we output 'frameshift' for that transcript as this would be more disruptive to gene function.

Running this module on the yeast annotations, we found that out of 6,003 protein-coding transcripts in the reference genome, 5897 were identical, 5 failed to map entirely, and the others had variants with the effects shown in Table 6.1. This module completed in 39.9 seconds on a personal laptop.

Table 6.1. Variants module results. Variant effects and the number of transcripts affected in the target yeast assembly identified by the LiftoffTools sequences module.

Variant Effect	Number of Transcripts
Synonymous	12
Nonsynonymous	47
In-frame deletion	8
In-frame insertion	7
Start lost	5
5' truncation	0
3' truncation	1
Frameshift	21
Stop gained	0

Synteny

This module compares the gene order in the reference annotation to the order in the target annotation. The genes are sorted first by chromosome and then by start coordinate in each annotation. Each gene is then plotted as a point on a 2D plot where the x-coordinate is the ordinal position (e.g., 1st, 2nd, 3rd, etc.) in the reference genome and the y-coordinate is the ordinal position in the target genome. The color of the point corresponds to the sequence identity of the reference gene and the target gene where

green indicates higher identity and red indicates lower identity. Note this color feature is only available for target annotations created by Liftoff which have the sequence identity information in the GTF/GFF3. The plot and a file with the ordinal positions and sequence identities of each gene will be output. The user also has the option to calculate the edit distance between the reference order and the target order. This gives an estimate of the number of genes that are in a different order in the target genome with respect to the reference.

We ran this module on the yeast assemblies and the dot plot in Figure 6.1 shows that the genes are co-linear and nearly identical in sequence. The target has a small insertion on chromosome 8 (CP026287.1) that appears as a vertical shift in the plot.

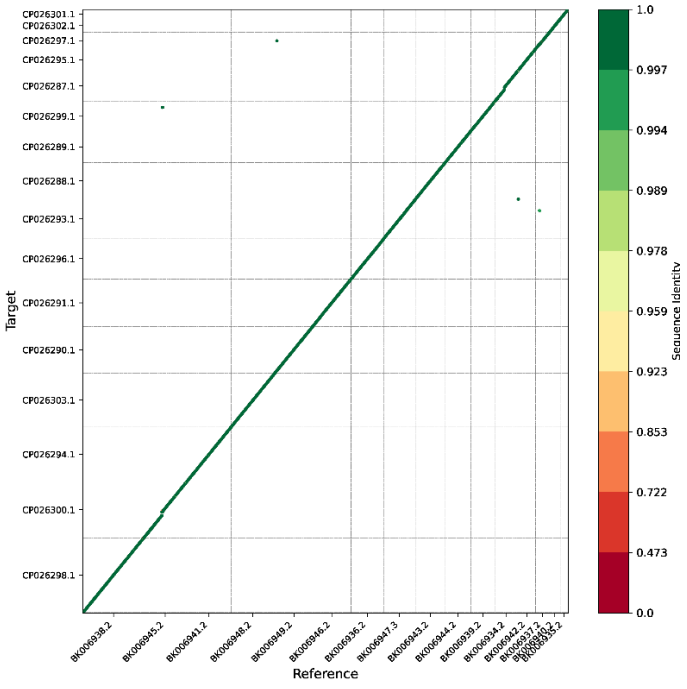

Figure 6.1: Yeast gene order dot plot. Dot plot showing the ordinal position of each gene in the reference assembly on the x-axis and the ordinal position in the target assembly on the y-axis. The color of each point indicates the sequence identity, and the gray lines separate the chromosomes. The labels on the x and y axes are the names of the chromosomes in the reference genome and target genome respectively.

When enabling the option to calculate the edit distance, we found 70 genes in the target genome that are in a different order with respect to the reference. Finding the gene order and creating the dot plot took 24.6 seconds and calculating the edit distance took 81.6 seconds on a personal laptop.

Clusters

This module clusters the genes into paralogous groups to evaluate the expansion and contraction of gene families. LiftoffTools first invokes MMSeqs2[105] to cluster the reference gene sequences. MMSeqs2 clusters the amino acid sequences of the protein-coding genes, and the nucleotide sequences of noncoding genes. For each gene we select only the longest isoform to be included in the clustering. The minimum sequence identity for clustering is 90% by default, but this parameter can be adjusted by the user. After clustering the reference genes, we define the target gene clusters by first iterating through each reference cluster and removing any gene that failed to map to the target. Next, if the -copies option was used with Liftoff to identify extra gene copies in the target genome, we add the extra copies to the same cluster as their closest paralog. For each cluster, we output the number of reference genes and the number of target genes belonging to that cluster as well as the gene IDs of the cluster members. Additionally, for each reference gene that failed to map to the target genome, we find the closest mapped paralog and output the gene ID and the sequence identity. This is only applicable for unmapped genes that clustered with one or more genes in the reference.

This module grouped the 6,418 reference yeast genes into 5,923 clusters. Only 183 clusters contain 2 or more genes (clusters with only 1 gene indicate that the gene does not have any paralogs with >= 90% sequence identity). 8 of the 183 clusters contain fewer genes in the target genome than in the reference indicating possible gene family contraction. 25 of the 167 clusters contain more genes in the target genome indicating possible gene family expansion. The clusters module completed in 150.8 seconds.

Chapter 7: Conclusions

In the not-too-distant future, we will likely see the fully automated assembly of telomere-to-telomere genomes. To prevent the gap between assembled genomes and annotated genomes growing larger, improved annotation methods are imperative. In this work we presented two such methods. First, we introduced hybrid-read transcriptome assembly with StringTie. Using simulated data and real data from human, *Mus musculus*, and *Arabidopsis thaliana*, we showed that hybrid-read assembly is more precise and assembles more known transcripts than long or short-read only assembly. We also demonstrate that it is substantially faster than correcting long reads before assembly while maintaining comparable accuracy. Next, we introduced Liftoff which maps gene annotations between assemblies of the same or closely-related species. As a proof of concept, we showed that Liftoff was able to map nearly all genes between the GRCh37 and GRCh38 human reference genomes as well as from GRCh38 to the chimpanzee reference genome. We also applied Liftoff to 3 novel human assemblies and a novel bread wheat assembly where in all cases, we mapped over the vast majority of known genes from the reference genomes. In 2 of the human assemblies (T2T-CHM13 and PR1) and the bread wheat assembly, we also detected and annotated many novel paralogs. Lastly, we introduced LiftoffTools. While not an annotation method itself, LiftoffTools automates downstream analysis of annotations mapped with Liftoff which can provide further insight into biological function.

While we have explored StringTie and Liftoff as separate methods, gene annotation will likely continue to require a combination of methods and datatypes for the foreseeable future. RNA-sequencing approaches like StringTie and homology-based approaches like Liftoff each have inherent limitations. RNA-sequencing is limited in that it will only ever capture expressed transcripts. Even if we reach a future where every transcript in a sample can be sequenced end-to-end without error, we will have to sequence a large number of samples in many conditions to obtain comprehensive annotation. Furthermore, it has been shown that not all expressed sequences are functional[22,106]. With the end goal of annotation being the identification of all *functional* elements in a genome, we cannot simply label any transcript that is expressed as functional.

The main limitation of homology-based approaches is that accuracy and completeness of the annotation is entirely dependent on the accuracy and completeness of the annotations of closely-related species. Even species that are considered "well-annotated" often have errors or inconsistencies in their annotations. Take the human genome, for example, where the two most commonly used annotations, RefSeq[65] and GENCODE[48], have far more unique transcripts than transcripts in common[22]. Additionally, homology-based approaches will miss any de novo genes which are characterized by their lack of homologous genes in other species and have been identified in a wide-range of organisms[107] .

Combining both RNA-sequencing and homology evidence likely will mitigate some of these limitations and produce better annotations. A way to achieve this with the methods presented here would be to first use Liftoff to map gene annotations from a genome of the same or closely related species. Then we can use StringTie to assemble long and short-read RNA-sequencing data to identify novel transcripts or genes not annotated by Liftoff. To distinguish novel functional transcripts from noise we can check whether the transcript is conserved in other species. Conservation in many species is strong evidence that the sequence has some function, even if it is not annotated in any other species. Future work may explore the benefits of using both methods in combination as just described.

While the methods presented here offer improvements over existing methods, there is still work to be done before accurate and complete genome annotation is fully automated. However, with the speed at which the field of genomics has evolved, the day where we can determine the sequence and function of the genome of every organism may be closer than we think.

References

1. DNA Sequencing Costs: Data. https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data.
2. Burge, C. & Karlin, S. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology* **268**, 78–94 (1997).
3. Stanke, M., Schöffmann, O., Morgenstern, B. & Waack, S. Gene prediction in eukaryotes with a generalized hidden Markov model that uses hints from external sources. *BMC Bioinformatics* **7**, 1–11 (2006).
4. Cantarel, B. L. *et al.* MAKER: An easy-to-use annotation pipeline designed for emerging model organism genomes. *Genome Research* **18**, 188 (2008).
5. Majoros, W. H., Pertea, M. & Salzberg, S. L. TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics* **20**, 2878–2879 (2004).
6. Keilwagen, J., Hartung, F., Paulini, M., Twardziok, S. O. & Grau, J. Combining RNA-seq data and homology-based gene prediction for plants, animals and fungi. *BMC Bioinformatics* **19**, 1–12 (2018).
7. Salzberg, S. L. Next-generation genome annotation: We still struggle to get it right. *Genome Biology* **20**, 1–3 (2019).
8. Shumate, A., Wong, B., Pertea, G. & Pertea, M. Improved Transcriptome Assembly Using a Hybrid of Long and Short Reads with StringTie. *bioRxiv* 2021.12.08.471868 (2021) doi:10.1101/2021.12.08.471868.

9. Shumate, A. & Salzberg, S. L. Liftoff: accurate mapping of gene annotations. *Bioinformatics* **37**, 1639–1643 (2021).

10. Shumate, A. *et al.* Assembly and annotation of an Ashkenazi human reference genome. *Genome Biology* **21**, 1–18 (2020).

11. Zimin, A. v *et al.* A reference-quality, fully annotated genome from a Puerto Rican individual. *Genetics* **220**, (2022).

12. Nurk, S. *et al.* The complete sequence of a human genome. *bioRxiv* 2021.05.26.445798 (2021) doi:10.1101/2021.05.26.445798.

13. Alonge, M., Shumate, A., Puiu, D., Zimin, A. v. & Salzberg, S. L. Chromosome-Scale Assembly of the Bread Wheat Genome Reveals Thousands of Additional Gene Copies. *Genetics* **216**, 599–608 (2020).

14. Wang, E. T. *et al.* Alternative isoform regulation in human tissue transcriptomes. *Nature* **456**, (2008).

15. Stoler, N. & Nekrutenko, A. Sequencing error profiles of Illumina sequencing instruments. *NAR Genomics and Bioinformatics* **3**, (2021).

16. Stark, R., Grzelak, M. & Hadfield, J. RNA sequencing: the teenage years. *Nature Reviews Genetics* vol. 20 (2019).

17. Grabherr, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology* **29**, (2011).

18. Fu, S. *et al.* IDP-denovo: De novo transcriptome assembly and isoform annotation by hybrid sequencing. in *Bioinformatics* vol. 34 (2018).

19. Prjibelski, A. D. *et al.* Extending rnaSPAdes functionality for hybrid transcriptome assembly. *BMC Bioinformatics* **21**, (2020).

20. Pertea, M. *et al.* StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology* **33**, (2015).

21. Kovaka, S. *et al.* Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biology* **20**, (2019).

22. Pertea, M. *et al.* CHESS: A new human gene catalog curated from thousands of large-scale RNA sequencing experiments reveals extensive transcriptional noise. *Genome Biology* **19**, 1–14 (2018).

23. Amarasinghe, S. L. *et al.* REVIEW Open Access Opportunities and challenges in long-read sequencing data analysis. doi:10.1186/s13059-020-1935-5.

24. Broseus, L. *et al.* TALC: Transcript-level Aware Long-read Correction. *Bioinformatics* **36**, (2020).

25. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, (2009).

26. Bonfield, J. K. *et al.* HTSlib: C library for reading/writing high-throughput sequencing data. *GigaScience* **10**, (2021).

27. Wilks, C. & Schatz, M. C. LongTron: Automated Analysis of Long Read Spliced Alignment Accuracy. *bioRxiv* (2020).

28. Griebel, T. *et al.* Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research* **40**, (2012).

29. Yang, C., Chu, J., Warren, R. L. & Birol, I. NanoSim: Nanopore sequence read simulator based on statistical characterization. *GigaScience* vol. 6 (2017).

30. Pertea, M. & Pertea, G. GFF Utilities: GffRead and GffCompare. *F1000Research* **9**, (2020).

31. Kim, D., Paggi, J. M., Park, C., Bennett, C. & Salzberg, S. L. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology* **37**, (2019).

32. Li, H. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, (2018).

33. Marçais, G. & Kingsford, C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* **27**, (2011).

34. Church, D. M. *et al.* Modernizing Reference Genome Assemblies. *PLOS Biology* **9**, e1001091 (2011).

35. He, Y. *et al.* Long-read assembly of the Chinese rhesus macaque genome and identification of ape-specific structural variants. *Nature communications* **10**, (2019).

36. Jiao, Y. *et al.* Improved maize reference genome with single-molecule technologies. *Nature 2017 546:7659* **546**, 524–527 (2017).

37. Miga, K. H. & Wang, T. The Need for a Human Pangenome Reference Sequence. *https://doi.org/10.1146/annurev-genom-120120-081921* **22**, 81–102 (2021).

38. human-pangenomics/HPP_Year1_Assemblies: Assemblies from HPP Year 1 production. https://github.com/human-pangenomics/HPP_Year1_Assemblies.

39. Kuhn, R. M., Haussler, D. & James Kent, W. The UCSC genome browser and associated tools. *Briefings in bioinformatics* **14**, 144–161 (2013).

40. Zhao, H. *et al.* CrossMap: a versatile tool for coordinate conversion between genome assemblies. *Bioinformatics (Oxford, England)* **30**, 1006–1007 (2014).

41. Gao, B., Huang, Q. & Baudis, M. segment_liftover : a Python tool to convert segments between genome assemblies. *F1000Research* **7**, 319 (2018).

42. daler/gffutils: GFF and GTF file manipulation and interconversion. https://github.com/daler/gffutils.

43. Shirley, M. D., Ma, Z., Pedersen, B. S. & Wheelan, S. J. Efficient “pythonic” access to FASTA files using pyfaidx. (2015) doi:10.7287/PEERJ.PREPRINTS.970V1.

44. networkx/networkx: Network Analysis in Python. https://github.com/networkx/networkx.

45. pysam-developers/pysam: Pysam is a Python module for reading and manipulating SAM/BAM/VCF/BCF files. It’s a lightweight wrapper of the htslib C-API, the same one that powers samtools, bcftools, and tabix. https://github.com/pysam-developers/pysam.

46. Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik 1959 1:1* **1**, 269–271 (1959).

47. Harrow, J. *et al.* GENCODE: the reference human genome annotation for The ENCODE Project. *Genome research* **22**, 1760–1774 (2012).

48. Frankish, A. *et al.* GENCODE reference annotation for the human and mouse genomes. *Nucleic acids research* **47**, D766–D773 (2019).

49. Guo, Y. *et al.* Improvements and impacts of GRCh38 human reference on high throughput sequencing data analysis. *Genomics* **109**, 83–90 (2017).

50. Mikkelsen, T. S. *et al.* Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature 2005 437:7055* **437**, 69–87 (2005).

51. Yunis, J. J. & Prakash, O. The origin of man: a chromosomal pictorial legacy. *Science (New York, N.Y.)* **215**, 1525–1530 (1982).

52. Soto, D. C. *et al.* Identification of Structural Variation in Chimpanzees Using Optical Mapping and Nanopore Sequencing. *Genes* **11**, (2020).

53. Ondov, B. D. *et al.* Mash: Fast genome and metagenome distance estimation using MinHash. *Genome Biology* **17**, 1–14 (2016).

54. Pertea, G. & Pertea, M. GFF Utilities: GffRead and GffCompare. *F1000Research* **9**, (2020).

55. McPherson, J. D. *et al.* A physical map of the human genome. *Nature 2001 409:6822* **409**, 934–941 (2001).

56. Green, R. E. *et al.* A Draft Sequence of the Neandertal Genome. *Science (New York, N.Y.)* **328**, 710 (2010).

57. Need, A. C. & Goldstein, D. B. Next generation disparities in human genomics: concerns and remedies. *Trends in genetics : TIG* **25**, 489–494 (2009).

58. Popejoy, A. B. & Fullerton, S. M. Genomics is failing on diversity. *Nature 2016 538:7624* **538**, 161–164 (2016).

59. Ballouz, S., Dobin, A. & Gillis, J. A. Is it time to change the reference genome? *Genome Biology* **20**, 1–9 (2019).

60. Wong, K. H. Y., Levy-Sakin, M. & Kwok, P. Y. De novo human genome assemblies reveal spectrum of alternative haplotypes in diverse populations. *Nature Communications 2018 9:1* **9**, 1–9 (2018).

61. Magi, A. *et al.* Characterization and identification of hidden rare variants in the human genome. *BMC Genomics* **16**, 1–16 (2015).

62. Ferrarini, A. *et al.* The Use of Non-Variant Sites to Improve the Clinical Assessment of Whole-Genome Sequence Data. *PLOS ONE* **10**, e0132180 (2015).

63. Barbitoff, Y. A. *et al.* Catching hidden variation: Systematic correction of reference minor allele annotation in clinical variant calling. *Genetics in Medicine* **20**, 360–364 (2018).

64. Fiddes, I. T. *et al.* Comparative Annotation Toolkit (CAT)-simultaneous clade and personal genome annotation. *Genome Research* **28**, 1029–1038 (2018).

65. O'Leary, N. A. *et al.* Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic acids research* **44**, D733–D745 (2016).

66. Liddiard, K. *et al.* Sister chromatid telomere fusions, but not NHEJ-mediated inter-chromosomal telomere fusions, occur independently of DNA ligases 3 and 4. *Genome Research* **26**, 588–600 (2016).

67. Muraki, K. & Murnane, J. P. The DNA damage response at dysfunctional telomeres, and at interstitial and subtelomeric DNA double-strand breaks. *Genes & Genetic Systems* **92**, 135–152 (2017).

68. Bailey, S. M. & Murnane, J. P. Telomeres, chromosome instability and cancer. *Nucleic Acids Research* **34**, 2408–2417 (2006).

69. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nature Biotechnology 2014 32:3* **32**, 246–251 (2014).

70. Aganezov, S. *et al.* A complete reference genome improves analysis of human genetic variation. *bioRxiv* 2021.07.12.452063 (2021) doi:10.1101/2021.07.12.452063.

71. Wagner, J. *et al.* Curated variation benchmarks for challenging medically relevant autosomal genes. *Nature Biotechnology 2022* 1–9 (2022) doi:10.1038/s41587-021-01158-1.

72. Vollger, M. R. *et al.* Segmental duplications and their variation in a complete human genome. *bioRxiv* 2021.05.26.445678 (2021) doi:10.1101/2021.05.26.445678.

73. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. (2013).

74. H, L., B, H., A, W., T, F. & J, R. The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078–2079.

75. Madeira, F. *et al.* The EMBL-EBI search and sequence analysis tools APIs in 2019. *Nucleic Acids Research* **47**, W636–W641 (2019).

76. Cleary, J. G. *et al.* Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines. *bioRxiv* 023754 (2015) doi:10.1101/023754.

77. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).

78. Altschul, S. F. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research* **25**, 3389–3402 (1997).

79. Paten, B. *et al.* Cactus: Algorithms for genome multiple sequence alignment. *Genome Research* **21**, 1512–1528 (2011).

80. Stanke, M., Diekhans, M., Baertsch, R. & Haussler, D. Using native and syntenically mapped cDNA alignments to improve de novo gene finding. *Bioinformatics* **24**, 637–644 (2008).

81. Kim, J. H. *et al.* Variation in human chromosome 21 ribosomal RNA genes characterized by TAR cloning and long-read sequencing. *Nucleic acids research* **46**, 6712–6725 (2018).

82. Dennis, M. Y. *et al.* The evolution and population diversity of human-specific segmental duplications. *Nature ecology & evolution* **1**, (2017).

83. Sudmant, P. H. *et al.* Diversity of human copy number variation and multicopy genes. *Science (New York, N.Y.)* **330**, 641–646 (2010).

84. Dubcovsky, J. & Dvorak, J. Genome plasticity a key factor in the success of polyploid wheat under domestication. *Science* **316**, 1862–1866 (2007).

85. Petersen, G., Seberg, O., Yde, M. & Berthelsen, K. Phylogenetic relationships of Triticum and Aegilops and evidence for the origin of the A, B, and D genomes of common wheat (Triticum aestivum). *Molecular Phylogenetics and Evolution* **39**, 70–82 (2006).

86. Arumuganathan, K. & Earle, E. D. Nuclear DNA content of some important plant species. *Plant Molecular Biology Reporter 1991 9:3* **9**, 208–218 (1991).

87. Appels, R. *et al.* Shifting the limits in wheat research and breeding using a fully annotated reference genome. *Science* **361**, (2018).

88. Chapman, J. A. *et al.* A whole-genome shotgun approach for assembling and anchoring the hexaploid bread wheat genome. *Genome Biology* **16**, 1–17 (2015).

89. Clavijo, B. J. *et al.* An improved assembly and annotation of the allohexaploid wheat genome identifies complete families of agronomic genes and provides genomic evidence for chromosomal translocations. *Genome Research* **27**, 885–896 (2017).

90. Zimin, A. v. *et al.* The first near-complete assembly of the hexaploid bread wheat genome, Triticum aestivum. *GigaScience* **6**, 1–7 (2017).

91. Beales, J., Turner, A., Griffiths, S., Snape, J. W. & Laurie, D. A. A Pseudo-Response Regulator is misexpressed in the photoperiod insensitive Ppd-D1a mutant of wheat (Triticum aestivum L.). *Theoretical and Applied Genetics* **115**, 721–733 (2007).

92. Díaz, A., Zikhali, M., Turner, A. S., Isaac, P. & Laurie, D. A. Copy Number Variation Affecting the Photoperiod-B1 and Vernalization-A1 Genes Is Associated with Altered Flowering Time in Wheat (Triticum aestivum). *PLOS ONE* **7**, e33234 (2012).

93. Coen, E. S. & Meyerowitz, E. M. The war of the whorls: genetic interactions controlling flower development. *Nature 1991 353:6339* **353**, 31–37 (1991).

94. Ng, M. & Yanofsky, M. F. Function and evolution of the plant MADS-box gene family. *Nature Reviews Genetics 2001 2:3* **2**, 186–195 (2001).

95. Soyk, S. *et al.* Duplication of a domestication locus neutralized a cryptic variant that caused a breeding barrier in tomato. *Nature Plants 2019 5:5* **5**, 471–479 (2019).

96. Díaz, A., Zikhali, M., Turner, A. S., Isaac, P. & Laurie, D. A. Copy number variation affecting the photoperiod-B1 and vernalization-A1 genes is associated with altered flowering time in wheat (Triticum aestivum). *PLoS ONE* **7**, (2012).

97. Würschum, T., Boeven, P. H. G., Langer, S. M., Longin, C. F. H. & Leiser, W. L. Multiply to conquer: Copy number variations at Ppd-B1 and Vrn-A1 facilitate global adaptation in wheat. *BMC Genetics* **16**, (2015).

98. Würschum, T., Longin, C. F. H., Hahn, V., Tucker, M. R. & Leiser, W. L. Copy number variations of CBF genes at the Fr-A2 locus are essential components of winter hardiness in wheat. *Plant Journal* **89**, 764–773 (2017).

99. Würschum, T., Langer, S. M., Longin, C. F. H., Tucker, M. R. & Leiser, W. L. A three-component system incorporating Ppd-D1, copy number variation at Ppd-B1, and numerous small-effect quantitative trait loci facilitates adaptation of heading time in winter wheat cultivars of worldwide origin. *Plant Cell and Environment* **41**, 1407–1416 (2018).

100. Alonge, M. *et al.* Major Impacts of Widespread Structural Variation on Gene Expression and Crop Improvement in Tomato. *Cell* **182**, 145-161.e23 (2020).

101. Liu, Y. *et al.* Pan-Genome of Wild and Cultivated Soybeans. *Cell* **182**, 162-176.e13 (2020).

102. Song, J. M. *et al.* Eight high-quality genomes reveal pan-genome architecture and ecotype differentiation of Brassica napus. *Nature Plants* **6**, 34–45 (2020).

103. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology* **215**, 403–410 (1990).

104. Daily, J. Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinformatics* **17**, 1–11 (2016).

105. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology 2017 35:11* **35**, 1026–1028 (2017).

106. Palazzo, A. F. & Lee, E. S. Non-coding RNA: What is functional and what is junk? *Frontiers in Genetics* **5**, 2 (2015).

107. Ruiz-Orera, J. *et al.* Origins of De Novo Genes in Human and Chimpanzee. *PLOS Genetics* **11**, e1005721 (2015).