Improved Transcriptome Assembly Using a Hybrid of Long and Short Reads with StringTie

Alaina Shumate^{1,2}, Brandon Wong^{1,2,3,4}, Geo Pertea⁵, Mihaela Pertea^{1,2,3}

1. Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD
2. Center for Computational Biology, Johns Hopkins University, Baltimore, MD
3. Department of Computer Science, Johns Hopkins University, Baltimore, MD
4. Department of Applied Math and Statistics, Johns Hopkins University, Baltimore, MD
5. The Lieber Institute for Brain Development, Baltimore, MD

Abstract

Short-read RNA sequencing and long-read RNA sequencing each have their strengths and weaknesses for transcriptome assembly. While short reads are highly accurate, they are unable to span multiple exons. Long-read technology can capture full-length transcripts, but its high error rate often leads to mis-identified splice sites, and its low throughput makes quantification difficult. Here we present a new release of StringTie that performs hybrid-read assembly. By taking advantage of the strengths of both long and short reads, hybrid-read assembly with StringTie is more accurate than long-read only or short-read only assembly, and on some datasets it can more than double the number of correctly assembled transcripts, while obtaining substantially higher precision

than the long-read data assembly alone. Here we demonstrate the improved accuracy on simulated data and real data from *Arabidopsis thaliana*, *Mus musculus*, and human. We also show that hybrid-read assembly is more accurate than correcting long reads prior to assembly while also being substantially faster. StringTie is freely available as open source software at <https://github.com/gperte/stringtie>.

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 higher eukaryotes, this task is more challenging, as the proteomic diversity and phenotypic complexity is driven by alternative splicing. It occurs frequently, with an estimated 92%-94% of human genes undergoing alternative splicing [1]. 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% [2]. 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 also 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 [3]. Additionally, long reads have a high error rate between 13% and 15% [4], 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[5], IDP-denovo [6], and rnaSPAdes [7], 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 [8]. 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 [9].

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 also substantially faster than a strategy of correcting long reads prior to assembly.

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 1 shows an example of a single transcript that can only be correctly assembled using both long and short reads. The long reads provide the advantage of spanning several exons; however, they contain many errors and the coverage is low. As a result, only two reads map to the first exon (with disagreeing splice sites), only one read maps to the second exon, and the first two exons are omitted from the long-read assembly.

“Fuzzy” splice sites are a common problem with aligning long reads to a reference genome, but other confusing alignment artifacts are often present as well, such as

retained introns, spurious extra exons, falsely skipped exons, and false alternative splice sites, all of them illustrated in Fig 1. While the short reads have better coverage and accuracy, there is only one spliced read mapped to the first exon, leading to incorrect assembly because the coverage of the rest of the transcript is much higher. In the hybrid-read assembly, StringTie uses this one read as support for the first intron in the long-reads (See Methods). The short reads also support the rest of the correct splice sites observed in the long reads, enabling StringTie to correctly assemble the transcript.

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.

Figure 1: Example of a transcript that can only be correctly assembled in the hybrid-read assembly. The gray rectangles represent aligned reads with dashed lines indicating the read is spliced. The colored lines represent errors in the reads. The blue, red and purple transcripts are the StringTie assemblies created from the long, short, and hybrid datasets respectively. Artifacts present in the 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.

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 2A). We then computed the precision and sensitivity for each assembly. Precision is defined as the percent of assembled transcripts that match true transcript, 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 2B). 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.

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 2C and 2D, 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.

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. All three species have well-characterized reference annotation available, even though their level of completeness is not fully established [10]. The short reads in these data sets were generated by Illumina sequencers and the long reads were generated by a variety of technologies including ONT direct RNA, ONT cDNA, and PacBio cDNA (Table 1). 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, but the results are similar when we look at all loci (data not shown).

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 [11], 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 [12], 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, and then created additional long-read and hybrid-read assemblies with the corrected reads. We 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.

Figure 2: **A)** 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. **B)** Calculated coverage vs. expected coverage for long-read, short-read, and hybrid-read assemblies of simulated data. **C)** Precision of long-read, short-read, and hybrid-read assemblies of simulated data with and without guide annotation. **D)** Sensitivity of long-read, short-read, and hybrid-read assemblies of simulated data with and without guide annotation

Table 1: Descriptions and availability of read RNA-seq datasets.

Accession Number	Database	Species	Sequencing Technology
ERR3486096	European Nucleotide Archive	<i>A. thaliana</i>	Illumina
ERR3764345	European Nucleotide Archive	<i>A. thaliana</i>	ONT direct RNA
ERR3486098	European Nucleotide Archive	<i>A. thaliana</i>	Illumina
ERR3764349	European Nucleotide Archive	<i>A. thaliana</i>	ONT direct RNA
ERR3486099	European Nucleotide Archive	<i>A. thaliana</i>	Illumina
ERR3764351	European Nucleotide Archive	<i>A. thaliana</i>	ONT direct RNA
ERR2680378	European Nucleotide Archive	<i>M. musculus</i>	Illumina
ERR2680375	European Nucleotide Archive	<i>M. musculus</i>	ONT direct RNA
ERR2680377	European Nucleotide Archive	<i>M. musculus</i>	ONT cDNA
ERR2680380	European Nucleotide Archive	<i>M. musculus</i>	Illumina
ERR2680379	European Nucleotide Archive	<i>M. musculus</i>	ONT direct RNA
SRR4235527	Sequence Read Archive	<i>H. sapiens</i>	Illumina
NA12878-dRNA	github.com/nanopore-wgs-consortium	<i>H. sapiens</i>	ONT direct RNA
NA12878-cDNA	github.com/nanopore-wgs-consortium	<i>H. sapiens</i>	ONT cDNA
SRR1153470	Sequence Read Archive	<i>H. sapiens</i>	Illumina
SRR1163655	Sequence Read Archive	<i>H. sapiens</i>	PacBio cDNA

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 3 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 3 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 3 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.

Annotation-Guided Assembly

As with the simulated data, we also performed annotation-guided assembly for each species and evaluated the precision (Figure 4A) and number of annotated transcripts assembled (Figure 4B). 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 3: 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 SRR4235527 **I)** SRR1153470 and SRR1163655.

Figure 4: A) Precision of annotation-guided assemblies of all real datasets B) The number of annotated transcripts assembled in annotation-guided assemblies of all real datasets.

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. 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 3 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.

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 3, 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.

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 [13], and uses these alignments to create a splice graph [8]. This new release of StringTie also supports input alignment data in CRAM

format as it now makes use of the HTSlib C library [14] 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 5, 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 5. 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 [15] 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 $E(I)$ 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 [9]

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 [16] as was used to evaluate StringTie2 [9]. We used NanoSim [17] 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` [18].

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.

Alignment and Assembly

All short reads were aligned with HISAT2 with default parameters [19]. Long reads were aligned with Minimap2 [20] using the default parameters for spliced alignment.

Alignment files were sorted and converted to BAM format using samtools [13].

Transcriptome assembly and quantification was done with StringTie version 2.2.0. We used the following StringTie commands to assemble the input alignment file for each assembly type:

- 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 [18] 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 1A): 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 $\geq 80\%$ of the length.

Simulated data with default sensitivity parameters (Figure 1C and 1D): The full set of expressed transcripts in the simulated data.

Real data (Figure 3 and 4): The reference annotation for the given species filtered to only include loci covered by at least one long read.

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% [9]. 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 [21]. 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
```

Availability

StringTie is freely available as open source software at

<https://github.com/gpertea/stringtie>. An up-to-date documentation, and usage manual

can be consulted at <https://ccb.jhu.edu/software/stringtie/index.shtml>.

Acknowledgments

Authors want to thank Steven Salzberg for proofreading the manuscript.

Funding

This work was supported in part by NSF grant DBI-1759518 and by NIH grant R01 MH123567.

References

1. Wang ET, Sandberg R, Luo S, Khrebtkova I, Zhang L, Mayr C, et al. Alternative isoform regulation in human tissue transcriptomes. *Nature*. 2008;456.
2. Stoler N, Nekrutenko A. Sequencing error profiles of Illumina sequencing instruments. *NAR Genomics and Bioinformatics*. 2021;3.
3. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. *Nature Reviews Genetics*. 2019.
4. Buck D, Weirather JL, de Cesare M, Wang Y, Piazza P, Sebastiano V, et al. Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis. *F1000Research*. 2017;6.

5. Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, et al. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*. 2011;29.
6. Fu S, Ma Y, Yao H, Xu Z, Chen S, Song J, et al. IDP-denovo: De novo transcriptome assembly and isoform annotation by hybrid sequencing. *Bioinformatics*. 2018.
7. Prjibelski AD, Puglia GD, Antipov D, Bushmanova E, Giordano D, Mikheenko A, et al. Extending rnaSPAdes functionality for hybrid transcriptome assembly. *BMC Bioinformatics*. 2020;21.
8. Perte M, Perte GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*. 2015;33.
9. Kovaka S, Zimin A v., Perte GM, Razaghi R, Salzberg SL, Perte M. Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biology*. 2019;20.
10. Perte M, Shumate A, Perte G, Varabyou A, Breitwieser FP, Chang YC, et al. CHESS: A new human gene catalog curated from thousands of large-scale RNA sequencing experiments reveals extensive transcriptional noise. *Genome Biology*. *BioMed Central*; 2018;19:1–14.
11. Amarasinghe SL, Su S, Dong X, Zappia L, Ritchie ME, Gouil Q. REVIEW Open Access Opportunities and challenges in long-read sequencing data analysis.
12. Broseus L, Thomas A, Oldfield AJ, Severac D, Dubois E, Ritchie W. TALC: Transcript-level Aware Long-read Correction. *Bioinformatics*. 2020;36.
13. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*. 2009;25.

14. Bonfield JK, Marshall J, Danecek P, Li H, Ohan V, Whitwham A, et al. HTSlib: C library for reading/writing high-throughput sequencing data. *GigaScience*. Gigascience; 2021;10.
15. Wilks C, Schatz MC. LongTron: Automated Analysis of Long Read Spliced Alignment Accuracy. *bioRxiv*. 2020;
16. Griebel T, Zacher B, Ribeca P, Raineri E, Lacroix V, Guigó R, et al. Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research*. 2012;40.
17. Yang C, Chu J, Warren RL, Birol I. NanoSim: Nanopore sequence read simulator based on statistical characterization. *GigaScience*. 2017.
18. Perteu M, Perteu G. GFF Utilities: GffRead and GffCompare. *F1000Research*. 2020;9.
19. Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*. 2019;37.
20. Li H. Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*. 2018;34.
21. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*. 2011;27.