

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Next-Generation Transcriptome Assembly: Strategies and Performance Analysis

Adam Voshall and Etsuko N. Moriyama

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/intechopen.73497>

Abstract

Accurate and comprehensive transcriptome assemblies lay the foundation for a range of analyses, such as differential gene expression analysis, metabolic pathway reconstruction, novel gene discovery, or metabolic flux analysis. With the arrival of next-generation sequencing technologies, it has become possible to acquire the whole transcriptome data rapidly even from non-model organisms. However, the problem of accurately assembling the transcriptome for any given sample remains extremely challenging, especially in species with a high prevalence of recent gene or genome duplications, those with alternative splicing of transcripts, or those whose genomes are not well studied. In this chapter, we provided a detailed overview of the strategies used for transcriptome assembly. We reviewed the different statistics available for measuring the quality of transcriptome assemblies with the emphasis on the types of errors each statistic does and does not detect. We also reviewed simulation protocols to computationally generate RNAseq data that present biologically realistic problems such as gene expression bias and alternative splicing. Using such simulated RNAseq data, we presented a comparison of the accuracy, strengths, and weaknesses of nine representative transcriptome assemblers including *de novo*, genome-guided, and ensemble methods.

Keywords: RNAseq, transcriptome, assembly, *de novo*, genome-guided, ensemble approach

1. Introduction

Transcriptome assembly from high-throughput sequencing of mRNA (RNAseq) is a powerful tool for detecting variations in gene expression and sequences between conditions, tissues, or

strains/species for both model and non-model organisms [1, 2]. However, the ability to accurately perform such analyses is crucially dependent on the quality of the underlying assembly [3]. Especially for the detection of sequence variations, but also for isoform detection and transcript quantification, mis-assembly of genes of interest can increase both the false positive and false negative rates, depending on the nature of the mis-assembly [4]. These problems are exacerbated in non-model organisms where genomic sequences that can be used as the references, if available at all, are sufficiently different than those from the individuals sequenced [5].

Transcripts can be mis-assembled in several ways [6]. Two of the most drastic assembly errors are fragmentation, where a single transcript is assembled as one or more smaller contigs, and chimeras, where a contig is assembled using part or all of more than one transcript. Fragmentation errors tend to result from fluctuations in the read coverage along a transcript, with the breaks in the transcript sequence occurring in regions that have lower coverage. By contrast, chimera errors often occur because of ambiguous overlaps within the reads, coupled with algorithms that choose the longest possible contig represented by the data, or by adjacent genes on the genome being merged. Both of these types of errors can have major impacts especially on gene identification. Small (single or few) nucleotide alterations to the contig sequence also happen as mis-assemblies. Sequence mistakes are often the result of mis-sequenced reads, but can also result from ambiguity for highly similar reads e.g. from heterozygous genes and from duplicated genes. In some cases, these errors can shift the reading frame for the contig, which can have significant impacts on the translated protein sequence. Finally, transcripts can be mis-assembled when alternative transcripts are collapsed into a single contig [6].

In the following sections, we will first review strategies used for transcriptome assembly as well as how their performance can be assessed. We then compare the performance of representative transcriptome assembly methods using a simulated human transcriptome and RNAseq. Finally we discuss a possible strategy to improve transcriptome assembly accuracy.

2. Transcriptome assembly strategies

2.1. *De novo* assemblers

De novo assemblers generate contigs based solely on the RNAseq data [7–13]. Most of the *de novo* assemblers rely on de Bruijn graphs generated from kmer decompositions of the reads in the RNAseq data [14]. The reads are subdivided into shorter sequences of a given length k (the kmers) and the original sequence is reconstructed by the overlap of these kmer sequences. One major limitation of the de Bruijn graphs is the need for a kmer to start at every position along the original sequence in order for the graph to cover the full sequence [13]. This limitation creates a tradeoff in regard to the length of the kmers. Shorter kmers are more likely to fully cover the original sequence, but are more likely to be ambiguous, with a single kmer corresponding to multiple reads from multiple transcripts. While by using longer kmers such ambiguity can be avoided, those kmers may not cover the entire sequence of some transcripts causing e.g. fragmented assembly. Consequently, each transcript, with its unique combination of expression level (corresponding to the number of reads in the RNAseq data generated from

that transcript) and sequence will have a different best kmer length for its assembly [15]. As a result, even using the same *de novo* assembly algorithm, performing two assemblies with different kmer lengths will generate a different set of contigs, inevitably with a varying set of correctly assembled contigs [16].

Examples of popularly used *de novo* assemblers include idba-Tran [9], SOAPdenovo-Trans [8], rnaSPAdes [12], and Trinity [7]. Idba-Tran is unique among these *de novo* assemblers, as it runs individual assemblies across a range of kmer lengths and merges the results to form the final prediction. The remaining assemblers use only the results of a single kmer length. For SOAPdenovo-Trans and Trinity, a kmer length needs to be chosen (default kmer: 23 and 25, respectively), while rnaSPAdes dynamically determines the kmer length to be used based on the read data. While all of these tools use the same fundamental strategies to construct, revise, and parse the de Bruijn graph for the assemblies, each method uses different thresholds and different assumptions to make decisions. These differences lead to different subsets of transcripts being correctly assembled by each method. An example of how these tools produce different sets of contigs is shown in Section 4.2.

2.2. Genome-guided assemblers

Genome-guided assemblers avoid the ambiguity of kmer decompositions used in de Bruijn graphs by mapping the RNAseq data to the reference genome. In order to account of introns, mapping of the reads for genome-guided assembly needs to allow them to be split, where the first part of the read maps to one location (an exon), and the other half maps to a downstream location (another exon). This mapping is done by split-read mappers such as TopHat [17], STAR [18], HISAT [19], or HPG-aligner [20]. Each of these methods maps the reads slightly differently, which may impact the quality of subsequent assembly.

This read mapping greatly reduces the complexity of transcript assembly by clustering the reads based on genomic location rather than relying solely on overlapping sequences within the reads themselves [3]. However, this approach still has some major drawbacks. The most obvious drawback is that genome-guided assemblers require a reference genome, which is not available for all organisms. The quality of the reference genome, if it is available, also impacts the quality of the read mapping and, by extension, the assembly. This impact is particularly noteworthy when genes of interest contain gaps in the genome assembly, preventing the reads necessary to assemble those genes from mapping to part or all of the transcript sequence. Ambiguity occurs also when reads map to multiple places within a genome. How the specific algorithm handles choosing which potential location a read should map to can have a large impact on the final transcripts predicted [6]. This problem is expounded when working with organisms different from the reference, where not all of reads map to the reference without gaps or mismatches.

Examples of popularly used genome-guided assemblers include Bayesembler [21], Cufflinks [22], and StringTie [23]. While each of these methods uses the mapped reads to create a graph representing the splice junctions of the transcripts, how they select which splice junctions are real differs fundamentally. Cufflinks constructs transcripts based on using the fewest number of transcripts to cover the highest percentage of mapped reads. StringTie uses the number of reads that span each splice junction to construct a flow graph, constructing the transcripts

based in order of the highest flow. Bayesemblem constructs all viable transcripts for each splice junction and uses a Bayesian likelihood estimation based on the read coverage of each potential transcript to determine which combination of transcripts is most likely. Due to these fundamentally different approaches, each of these tools produces different sets of transcripts from the same set of reads. An example of assemblies produced by these methods and how the assembled contigs differ is described in Section 4.3.

2.3. Ensemble approach

While a core set of transcripts are expected to be assembled correctly by many different assemblers, many transcripts will be missed by any individual tool [24] (also see Section 4). Through combining the assemblies produced by multiple methods, ensemble assemblers such as EvidentialGene [25] and Concatenation [26] attempt to address the limitations of individual assemblers, ideally keeping contigs that are more likely to be correctly assembled and discarding the rest. Both of EvidentialGene and Concatenation filter the contigs obtained from multiple assemblers (usually *de novo*) by clustering the contigs based on their sequences, predicting the coding region of the contig, and using features of the overall contig and the coding region to determine the representative sequence for each cluster. EvidentialGene recommends using several different tools across a wide range of kmer lengths. It uses the redundancy from multiple tools generating nearly identical sequences, clusters them, scores the sequences in each cluster based on the features of the sequence (e.g. lengths of the 5' and 3' untranslated regions), and returns one representative sequence from each cluster (keeping also some alternative sequences). In contrast, Concatenation recommends using only three assemblers, with one kmer length each. Concatenation merges nucleotide sequences that are identical or perfect subsets, only filters contigs with no predicted coding region.

These approaches greatly reduce the number of contigs by removing redundant and highly similar sequences. However, there is no guarantee that the correct representative sequence is kept for a given cluster or that each cluster represents one unique gene. Because they require multiple assemblies to merge, they also come at a far greater computational cost. An example of how these ensemble assembly strategies perform compared to individual *de novo* and genome-guided methods is shown in Section 4.4.

2.4. Third generation sequencing

All of the methods described so far primarily use short but highly accurate reads from Illumina sequencing for assembly, with or without a reference. With the rise of third-generation sequencing technologies from Pacific Biosciences (PacBio SMRT) and Oxford Nanopore Technologies (ONT MinION), it is becoming possible to sequence entire mRNA molecules as one very long read, though with a high error rate [27]. The ability to sequence the entire mRNA molecule is especially beneficial for detecting alternative splice forms, which remain a challenge for short-read only assembly, and potentially for more accurate transcript quantification if there is no bias in the mRNA molecules sequenced.

While many tools exist to perform genome assemblies using either these long reads alone or by combining long reads and Illumina reads, at present no short read transcriptome assemblers

take advantage of long-reads in transcriptome assembly. If these long reads can be sufficiently error-corrected (e.g. [28, 29]), they can be used for a snapshot of the expressed transcriptome, without requiring assembly or external references [30, 31]. Alternatively, after an independent *de novo* assembly of short reads, the long reads can be used to confirm alternative splice forms present in the assembly [32]. The long reads can be also mapped to a reference genome similar to the split-read mapping methods used for genome-guided short-read assemblers discussed above [27, 33–35]. With their accuracy increasing, in the future, long reads can be used more to improve transcriptome assembly quality.

3. Performance metrics used for transcriptome assembly

In this section, we will discuss commonly used metrics to assess the quality of transcriptome assemblies.

3.1. Metrics based on contig count and lengths

The most straightforward assembly metrics are those based on the number and lengths of the sequences produced [36]. The number of sequences can be presented either or both of:

- the number of contigs
- the number of scaffolds

where for contigs no further joining of the sequences is performed after assembly, and for scaffold contigs that have some support for being from the same original sequence are combined together with a certain number of gaps between them.

Several different statistics are available for presenting the lengths of the sequences (either contigs or scaffolds). The most commonly reported metrics are:

- minimum length (bp): the length of the shortest sequence produced
- maximum length (bp): the length of the longest sequence produced
- mean length (bp): the average length of the sequences produced
- median length (bp): the length where half of the sequences are shorter, and half of the sequences are longer
- N50 (bp): a weighted median where the sum of the lengths of all sequences longer than the N50 is at least half of the total length of the assembly
- L50: the smallest number of sequences whose combined length is longer than the N50

Additional metrics similar to N50 (e.g. N90) based on different thresholds are also used.

For genome assemblies where the target number of sequences is known (one circular genome plus any smaller plasmids for prokaryotic organisms and the number of chromosomes for eukaryotic organisms), these metrics provide an estimate for the thoroughness of the assembly

[36]. For instance, in prokaryotic assemblies, the vast majority of the sequence is expected to be in one long sequence, and having many shorter sequences indicates fragmentation of the assembly [15]. In this context, longer sequences (e.g. larger N50) tend to indicate higher quality assemblies. For transcriptome assemblies, however, the length of the assembled contigs varies depending on the lengths of the transcripts being assembled. For the human transcriptome, for example, while the longest transcript (for the gene coding the Titin protein) is over 100 kb, the shortest is only 186 bp, with a median length of 2787 bp [37]. Emphasizing longer contigs also rewards assemblers that over-assemble sequences, either by including additional sequence incorrectly within a gene, or by joining multiple genes together to form chimeric contigs. Therefore, for transcriptome assembly, metrics based on contig lengths do not necessarily reflect its quality.

3.2. Metrics based on coded protein similarity

Rather than focusing on the number or length of the sequences produced by the assembly, performing similarity searches with the assembled sequences can provide an estimate of the quality of the contigs or scaffolds [24, 38]. Typically, the process consists of either similarity searches against well annotated databases (such as the protein datasets of related genomes or targeted orthologs, the BLAST non-redundant protein database [39] or the UniProt/Swiss-Prot database [40]), conserved domain search within the contig sequence that determines the potential function of the gene (such as PFAM or Panther [41, 42]), or a search against a lineage specific conserved single-copy protein database (such as BUSCO [43]). These similarity searches are usually performed on the predicted protein sequences for the contigs (e.g. using GeneMarkS [44]), but can also be performed directly from the assembled nucleotide sequences using BLASTX where translated nucleotide sequences are used to search against a protein database [38]. If the organism being sequenced is closely related to a model organism with a well-defined transcriptome, nearly all of the contigs that are not erroneously assembled and code proteins should have identifiable potential homologs in the database. If a large percentage of the contigs do not have similar proteins identified in the database, there is a high probability that the sequences are incorrectly assembled, regardless of the length of the sequences. By performing similarity searches, over-assemblies or chimera contigs (those covering more than one gene) can be also detected as large gaps in the alignment between the query and the hits. As protein sequence annotations are necessary for most downstream analyses, they also provide a convenient metric without the need for additional, otherwise unnecessary analyses.

Despite these advantages, there are some limitations to using protein-similarity based metrics for assembler performance. First, the more divergent the organism being sequenced is from the sequences in the database searched and the more species-specific genes in the transcriptome, the lower the percentage of contigs with hits will be. This can result in some organisms appearing to have a lower quality assembly solely due to their divergence from those well represented in the databases. By extension, assemblies that recover more transcripts whose coded proteins have few similar sequences in the database will appear worse than assemblies that only recover conserved genes. This limitation can be somewhat mitigated by comparing

only genes that are universally single-copy across different species, which are more likely to be conserved and similar enough to be identified. This is the strategy used in BUSCO [43]. However, this comparison at best uses only a subset of the assembled contigs. Second, and more problematic, this metric rewards assemblies that artificially duplicate conserved genes with only small differences in the nucleotide sequence. In the extreme, this can result in several times as many contigs in the assembly than were present in the actual transcriptome, but with nearly all of the contigs coding conserved protein sequences. This is particularly an issue when the analysis depends on identifying the gene copy numbers in the assembly. It also has a large impact on the accuracy of contig quantification and differential expression analyses [45].

3.3. Assembly metrics based on benchmark transcriptomes

The only way to overcome the limitations of the metrics described in the previous sections is to compare the assembly output against a benchmark transcriptome where correct sequences of all transcripts are known. When an RNAseq data generated from a well-established model organism is used for assembly, many of correctly assembled contigs can be identified. However, variability in the transcriptome among e.g. cell types limits the amount of information that can be gained for incorrectly assembled contigs. It is also not possible to determine whether sequences from the reference that are missing from the assembled transcriptome are due to assembly errors, or whether they were not expressed in the library sequenced. Transcriptome sequences may also vary between the individual under study and the reference. Such variations can mask assembly errors that affect the contig sequences. Although this limitation can be mitigated by sequencing an individual that is genetically identical to the reference, it severely limits the types of organisms that can be used for the benchmark.

To comprehensively assess all of the assembly errors, we need to obtain RNAseq data from a transcriptome where all transcript sequences and expression patterns are known. Ideally, such a benchmark transcriptome would be synthetically produced and sequenced using standard protocols. However, currently no such synthetic mRNA library exists. An alternative approach is to simulate the sequencing of a given benchmark transcriptome. There are several tools that can generate simulated reads modeling short Illumina reads [46, 47] and/or long third-generation sequencing reads such as PacBio SMRT and ONT MinION [48, 49]. These tools typically either focus on identifying the statistical distribution of reads across the sequences and errors within the reads, as is the case for RSEM [46], PBSIM [48], and Nanosim [49], or attempt to reconstruct each step of the library preparation and sequencing pipeline, mimicking the errors and biases introduced at each step, as is the case for Flux Simulator [47].

Using simulated RNAseq data with a known transcriptome as a benchmark gives the most detailed and close to true performance metric for assemblies. Specifically, this strategy allows the quantification of each of the following categories:

- correctly assembled sequences (true positives or TPs)
- sequences that are assembled with errors (false positives or FPs)
- sequences in the reference that are missing from the assembly (false negatives or FNs)

“Correctness” and “incorrectness” (or error) can be defined using varying degrees of sequence similarities. Using the strictest threshold, a contig sequence is assembled “correctly” only if the entire nucleotide or coded protein sequence is identical to a reference transcript. All other contigs found in the assembly, including those whose sequences have no similarity in the reference transcriptome (missing contigs), are considered to be assembled “incorrectly” (FPs) regardless of the similarity against the reference sequences.

Note that true negatives (TNs) can be counted only if the assembly experiments are done including reads that are derived from transcripts that are not part of the reference transcriptome (negative transcripts). Using these categories, following assembly metrics can be calculated:

- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- Sensitivity (or recall) = $\frac{TP}{TP+FN}$
- Specificity = $\frac{TN}{TN+FP}$
- Precision = $\frac{TP}{TP+FP}$
- F-measure (or F_1 score) = $\frac{2(TP)}{2(TP)+FP+FN}$
- False discovery rate (FDR) = $\frac{FP}{FP+TP}$

Often in an RNAseq simulation, negative transcripts are not included; hence TN cannot be counted. In such cases, we can calculate an alternative metric as the accuracy:

- Accuracy* = $\frac{TP}{TP+FP+FN}$

Despite the added benefits of simulation for measuring the performance of assemblers, these metrics assume that the simulation accurately reflects the nature of real RNAseq data. Differences in the distribution of reads or errors between the simulations and real data can impact the relative performance of the assemblers. Assemblers that perform well on simulated data may perform poorly on real data if those assumptions are not met. Consequently, great care must be taken to ensure that the simulated data captures the features of real data as accurately as possible to best characterize the performance of different assembly strategies.

4. Performance analysis of transcriptome assemblers

In this section, as an example, we compare the performance of transcriptome assemblers using a simulated benchmark transcriptome dataset.

4.1. Benchmark transcriptome and simulated RNAseq

RNAseq datasets were generated by Flux Simulator [47] using the hg38 human genome (available at <https://genome.ucsc.edu/cgi-bin/hgGateway?db=hg38>) as the reference. The older hg19 human genome (available at <http://genome.ucsc.edu/cgi-bin/hgGateway?db=hg19>) was

also used as an alternate reference genome to assess the impact of using a different reference with genome-guided assemblers. The gene expression profile was generated by Flux Simulator using the standard parameters from the hg38 reference genome and transcriptome model. Approximately 250 million pairs of reads were computationally generated with the given expression model with no PolyA tail. The simulated library construction was fragmented uniformly at random, with an average fragment size of 500 (± 180) nucleotides (nt). Because reads overlapping within read pairs can cause problems for some assemblers, fragments shorter than 150 nt were removed. The simulated sequencing was performed using paired-end reads of length of 76 nt using the default error model based on the read quality of Illumina-HiSeq sequencers. Note that only reference transcripts with full coverage of RNAseq data were included in the benchmarking, as transcripts without full coverage cannot be correctly assembled as a single contig. This filtering removed 2700 transcripts expressed in the benchmark transcriptome, leaving 14,040 unique sequences derived from 8557 genes (5309 with no alternative splicing; on average 1.64, ranging up to 13, isoforms per gene).

The read pairs generated by Flux Simulator were quality filtered using Erne-filter version 2.0 [50]. The reads were filtered using ultra-sensitive settings with a minimum average quality of q20 (representing a 99% probability that the nucleotide is correctly reported). The filtering was performed in paired-end mode to ensure that both reads of the pair were either kept or discarded concurrently to keep the pairs together. The remaining reads were normalized using Khmer [51] with a kmer size of 32 and an expected coverage of 50 \times . The normalization was also performed in paired-end mode to maintain pairs.

4.2. *De novo* assemblies

We compared the performance among four *de novo* transcriptome assemblers: idba-Tran (version 1.1.1) [9], SOAPdenovo-Trans (version 1.03) [8], rnaSPAdes (version 3.11.0) [12], and Trinity (version 2.5.1) [7], using the simulated human RNAseq dataset as described in the previous section. The resulted assemblies were compared against the benchmark transcriptome. As shown in **Table 1**, all of the tools underestimated the number of transcripts present, generating fewer contigs than the number of transcripts expected (14,040). The best performing tool among the four compared was Trinity with the most correct contigs (5782) and the highest correct/incorrect ratio ($C/I = 0.84$). However, even with Trinity, still only 41% (5782/14,040) of transcripts in the benchmark were correctly assembled; the remaining almost 60% of contigs either contained errors in the sequence or were missed entirely. rnaSPAdes assembled the largest number of transcripts (874 more unique transcripts compared to Trinity). The number of unique transcripts generated, 13,513, is also the closest to the expected total number of transcripts (96% of 14,040). However, fewer of those sequences (36%) were correctly assembled, lowering the overall performance across all statistics than Trinity.

Performance statistics for each assembler is given in **Table 2**. Precision is a measure of how likely an assembled contig is to be correct, and recall is a measure of how likely the assembler is to correctly assemble a contig. In these terms, for assemblers with high precision, the contigs produced are more likely to be correct, but the assembly may miss a large number of sequences present in the sample. Conversely, assemblers with high recall values correctly assemble more of

Methods	Total ^a	Unique ^a	Correct ^a (%) ^b	Incorrect ^a	C/I ^c
[Default]					
idba-Tran	11,943	11,941	3504 (24.96)	8437	0.4153
SOAPdenovo-Trans	12,902	11,830	3754 (26.74)	8076	0.4648
rnaSPAdes	15,670	13,513	5014 (35.71)	8499	0.5900
Trinity	14,044	12,639	5782 (41.18)	6857	0.8432
[Pooled] ^d					
idba-Tran	170,358	41,849	6391 (45.52)	35,458	0.1802
SOAPdenovo-Trans	297,192	50,504	6059 (43.16)	44,445	0.1363
rnaSPAdes	765,525	113,975	6665 (47.47)	107,310	0.0621
Trinity	89,126	25,045	6452 (45.95)	18,593	0.3470

^aNumber of contigs assembled.
^bProportion (%) of transcripts in the benchmark that were correctly assembled.
^c(Number of correctly assembled contigs)/(number of incorrectly assembled contigs).
^dPooled results from using multiple kmers as follows: 15, 19, 23, 27, and 31 for Trinity; 15 kmer values ranging from 15 to 75 in increments of 4 for SOAPdenovo-Trans and rnaSPAdes; 20, 30, 40, 50, and 60 for idba-Tran.

Table 1. Performance of individual *de novo* assemblers on simulated RNAseq library using default parameters or pooled across multiple kmer lengths.

Methods	Precision	Recall	Accuracy*	F ₁	FDR
idba-Tran	0.2934	0.2496	0.1559	0.2697	0.7066
SOAPdenovo-Trans	0.3173	0.2674	0.1697	0.2902	0.6827
rnaSPAdes	0.3711	0.3571	0.2225	0.3640	0.6289
Trinity	0.4575	0.4118	0.2767	0.4334	0.5425

Table 2. Performance statistics of individual *de novo* assemblers using default parameters on simulated RNAseq library.

the sequences present in the sample, but may do so at the cost of accumulating a large number of incorrectly assembled contigs. In these statistics, both the modified accuracy score (accuracy*; see Section 3.3) and the F₁ score are a measure of the number of correctly assembled contigs relative to the number of missing and incorrectly assembled contigs. FDR is the proportion of assembled reads that are incorrect. Based on these statistics, Trinity is the best performing *de novo* assembler with the highest precision, recall, accuracy* and F₁ score, and the lowest FDR, followed by rnaSPAdes then SOAPdenovo-Trans. Despite idba-Tran running multiple kmers and merging the results, it performed worst across every metric.

In **Table 1**, the results from pooling (taking the union of) the outputs of multiple runs of each assembler across a range of kmer lengths are also shown. With these pooled assemblies, the proportion of correctly assembled transcripts in the benchmark for Trinity increased from 41 to 46%, and for rnaSPAdes from 36 to 47%. However, the pooling process also accumulated several times more unique incorrect sequences than additional correct sequences recovered.

For Trinity, the C/I decreased from 0.8432 to 0.3470, and for rnaSPAdes this ratio decreased from 0.5900 to 0.0621.

Although the four *de novo* assembly methods use the same core approach, each method assembled a different set of sequences correctly (**Figure 1A**). Only a set of 5331 contigs were correctly assembled by all of the four *de novo* assemblers with at least one kmer length. Additional 813, 567, and 670 contigs were correctly assembled by at least three, at least two, and only one of the assemblers, respectively. In contrast, the vast majority of the incorrectly assembled contigs were produced by only one assembler (**Figure 1B**). For these contigs, 3764 were produced by all four assemblers, while an additional 2692, 7977 and 166,720 were produced by at least three, at least two or only one of the assemblers, respectively.

4.3. Genome-guided assemblies

We next compared the transcriptome assembly performance among three genome-guided assemblers: Bayesembler (version 1.2.0) [21], Cufflinks (version 2.2.1) [22], and StringTie (version 1.0.4) [23]. To demonstrate the impact of using different reference genomes on genome-guided transcriptome assemblies, we used both of the hg38 as well as hg19 genomes as the references. Assembly assessment was done against the hg38 benchmark transcriptome.

Table 3 shows the performance of each of these tools in the two scenarios (RNAseq data and the reference were derived from the same or different genomes). As observed with *de novo* methods, all of these genome-guided methods underestimated the number of transcripts present, even more severely than *de novo* methods. In terms of the number of contigs correctly assembled, StringTie performed slightly better than other two methods. All three methods had comparable percent correct (36–41% with the same reference) and C/I (0.87–0.88 with the same

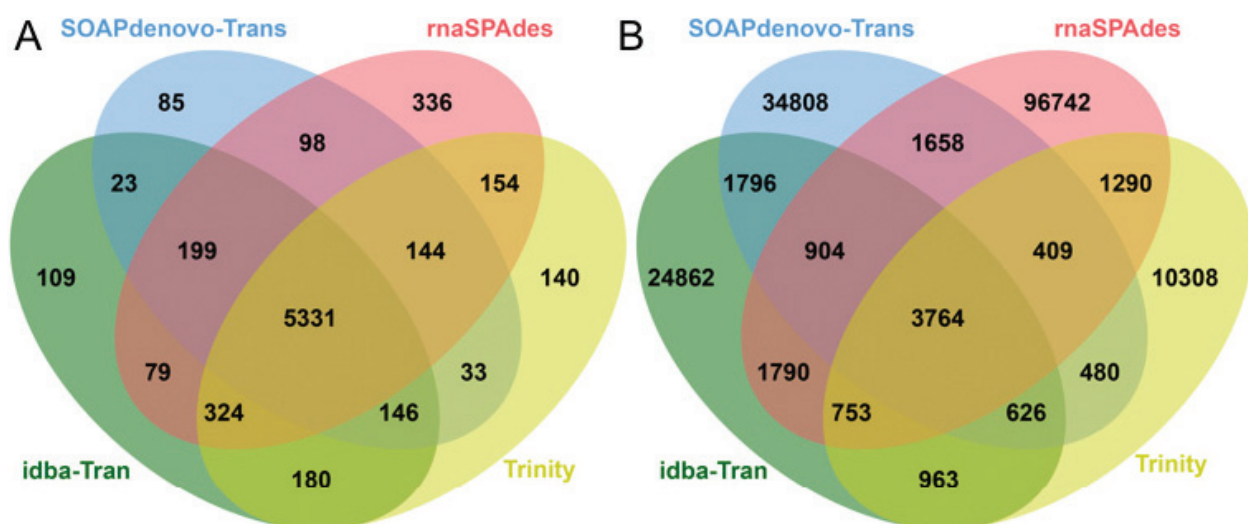


Figure 1. Comparisons of the contigs correctly (A) and incorrectly (B) assembled among four *de novo* assemblers. For each assembler, results from multiple kmers were pooled. Correctly assembled sequences were identified when the protein sequence of the contig matched the protein sequence in the benchmark transcriptome. Incorrectly assembled sequences were identified when the protein sequence of the contig did not exactly match any protein sequence in the benchmark transcriptome.

Methods	Total	Unique	Correct (%)	Incorrect	C/I
[Same reference]					
Bayesembler	12,989	11,482	5327 (37.94)	6155	0.8655
Cufflinks	11,257	10,733	4992 (35.56)	5741	0.8695
StringTie	13,218	12,147	5696 (40.57)	6451	0.8830
[Different reference]					
Bayesembler	8536	7479	3345 (23.82)	4134	0.8091
Cufflinks	7234	6906	3078 (21.92)	3828	0.8041
StringTie	8608	7867	3466 (24.69)	4401	0.7875

Table 3. Performance of individual genome-guided assemblers using default parameters on simulated RNAseq library with both the same and different references genome as the benchmark. See **Table 1** for the description of numbers shown.

reference). While none of the genome-guided assemblers produced as many correctly assembled contigs as the best performing *de novo* assembler (Trinity), proportions of correctly assembled contigs were higher with genome-guided methods ($C/I = 0.87\text{--}0.88$ with the same reference) than with the four *de novo* methods ($C/I = 0.41\text{--}0.84$). When the performance metrics are compared between the best performing *de novo* assembler (Trinity) and genome-guided assembler (StringTie) (**Table 4**), while both methods showed similar accuracy, StringTie (when using the same reference) showed slightly higher precision, accuracy* and F_1 and lower FDR compared to Trinity, but a slightly lower recall. It reflects fewer FPs and FNs produced by StringTie.

As with the *de novo* assemblers, each of these tools correctly assembled a different set of transcripts (**Figure 2A** and **C**). When the assemblies were performed using the same reference as the simulation, all of the genome-guided tools correctly assembled a core set of 4013 transcripts (**Figure 2A**). There were nearly a quarter as many (936) that were unique to only one genome-guided tool. When a different reference was used, the number of sequences correctly assembled by all of the tools dropped to 2546 (**Figure 2C**). Similar to the *de novo* assemblers, most of the

Methods	Precision	Recall	Accuracy*	F_1	FDR
[Same reference]					
Bayesembler	0.4639	0.3794	0.2638	0.4174	0.5361
Cufflinks	0.4651	0.3556	0.2524	0.4030	0.5349
StringTie	0.4689	0.4057	0.2780	0.4350	0.5311
[Different reference]					
Bayesembler	0.4473	0.2382	0.1841	0.3109	0.5527
Cufflinks	0.4457	0.2192	0.1723	0.2939	0.5543
StringTie	0.4406	0.2469	0.1880	0.3164	0.5594

Table 4. Performance statistics of individual genome-guided assemblers using default parameters on simulated RNAseq library with both the same and different references genome as the benchmark.

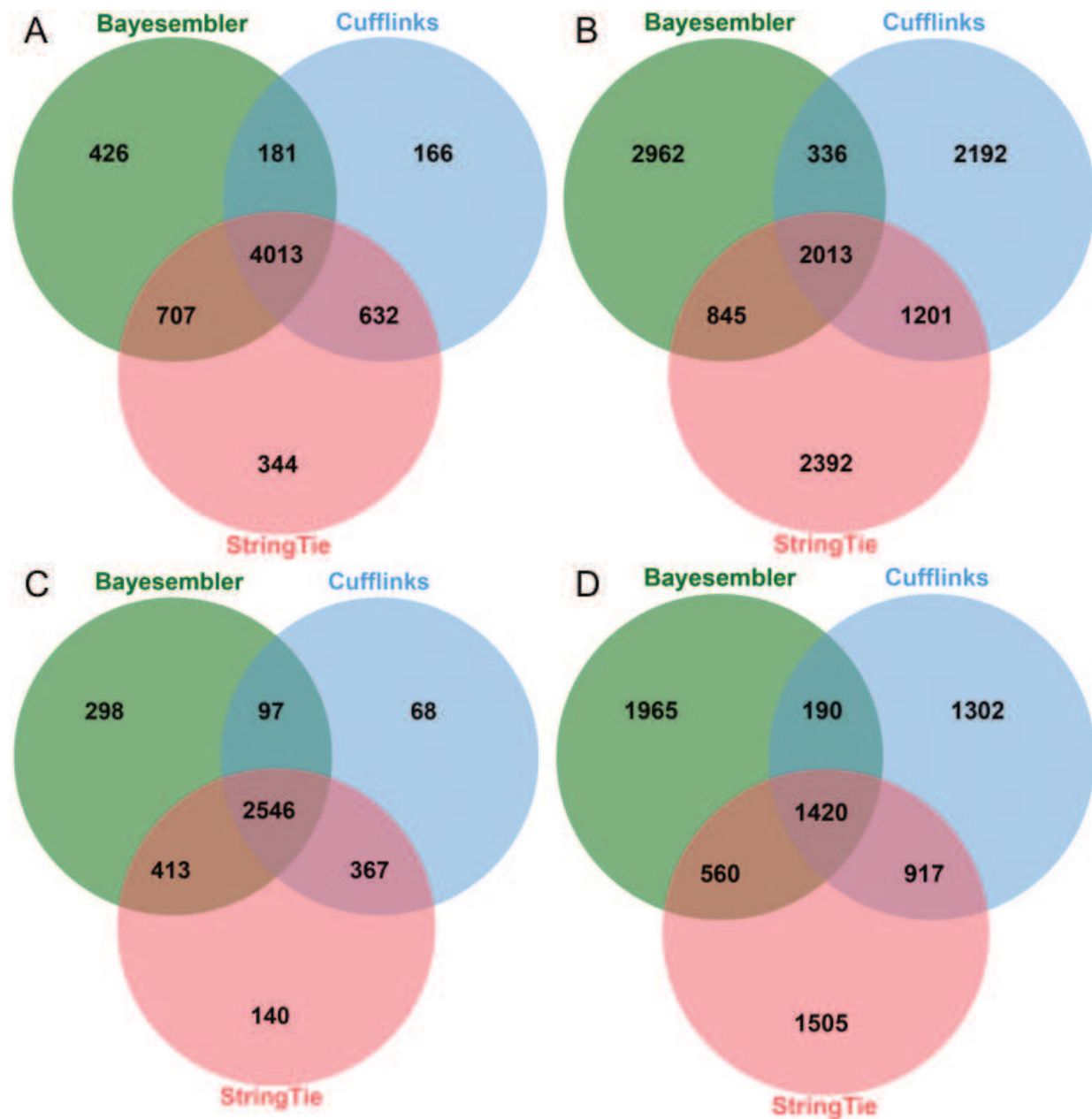


Figure 2. Comparisons of the contigs correctly (A and C) and incorrectly (B and D) assembled among three genome-guided assemblers. Correctly assembled sequences were identified when the protein sequence of the contig matches the protein sequence in the same (A) or different (C) reference genome. Incorrectly assembled sequences were identified when the protein sequence of the contig does not exactly match any protein sequence in the same (B) or different (D) reference genome.

incorrectly assembled contigs produced by each of the genome-guided assemblers were produced by only one assembler regardless of the reference genome used (**Figure 2B** and **D**). For assemblies using the same reference genome, 2013 incorrectly assembled contigs were produced by all of the tools, while an additional 2382 and 7546 were produced by any two or only one tool, respectively (**Figure 2B**). For assemblies using a different reference genome, 1420 incorrectly assembled contigs were produced by all of the tools, while an additional 1667 and 4772 were produced by any two or only one tool, respectively (**Figure 2D**).

4.4. Comparison of *de novo* and genome-guided assemblers

While the overall statistics are comparable between the best *de novo* assemblies and the genome-guided assemblies using the same reference genome, these tools produced different sets of contigs. The overlap of correctly assembled contigs between the assemblers from *de novo* with pooled kmers lengths and the three genome-guided assemblers are shown in **Figure 3A**. All of the *de novo* assemblers and at least one genome-guided assembler correctly assembled 4605 contigs. An additional 629 were assembled by at least three *de novo* and at least one genome-guided assembler and 427 assembled by at least two *de novo* and at least one genome-guided assembler. Conversely, 3861 contigs were correctly assembled by all of the three genome-guided assemblers and at least one *de novo* assembler, with 1338 assembled by at least two genome-guided assemblers and at least one *de novo* assembler (**Figure 3B**). Additionally, these tools produced only 602 correctly assembled contigs that were not predicted by any *de novo* assembly, while 1514 sequences were correctly assembled by at least one *de novo* assembly, but no genome-guided assemblies.

As with the individual assemblies, fewer incorrectly assembled contigs were produced by all of the tools, and most are assembler specific (**Figure 3C and D**). In particular, only 1387 incorrectly assembled contigs were produced by all of the *de novo* assemblers and at least one genome-guided assembler (**Figure 3C**), and only 1593 contigs were produced all of the genome-guided assemblers and at least one *de novo* assembler (**Figure 3D**). In contrast, 4823 incorrectly assembled contigs were produced by at least one genome-guided assembler but no *de novo* assemblers, and 176,397 incorrectly assembled contigs were produced by at least one *de novo* assembler but no genome-guided assemblers.

Overall, these results suggest that genome-guided assemblies provide relatively few correctly assembled contigs relative to performing multiple *de novo* assemblies, even when using the same reference genome. However, they produce far fewer incorrectly assembled contigs than the pooled *de novo* assemblies. If the correctly assembled contigs produced by each of the *de novo* assemblies can be retained while filtering out the incorrectly assembled contigs, *de novo* assemblies can outperform all of the genome-guided assemblies. This result forms the motivation of ensemble assembly strategies, discussed in the next section.

4.5. Ensemble assemblies

We compared the two ensemble transcriptome assembly methods, EvidentialGene (version 2017.03.09) [25] and Concatenation (version 1) [26] using the simulated RNAseq data. The strategies for these assemblies followed the recommendations by each method. For EvidentialGene, the pooled results from all of the four *de novo* assemblies performed across the full range of kmer lengths (described in Section 4.2) were used. For Concatenation, the results of a single assembly each from idba-Tran (using kmer length of 50), rnaSPAdes (with default kmer selection), and Trinity (with default kmer length) were used. These assemblers were chosen to match the assemblies used in [26], substituting the commercial CLC Assembly Cell (<https://www.qiagenbioinformatics.com/products/clc-assembly-cell/>) with freely available rnaSPAdes.

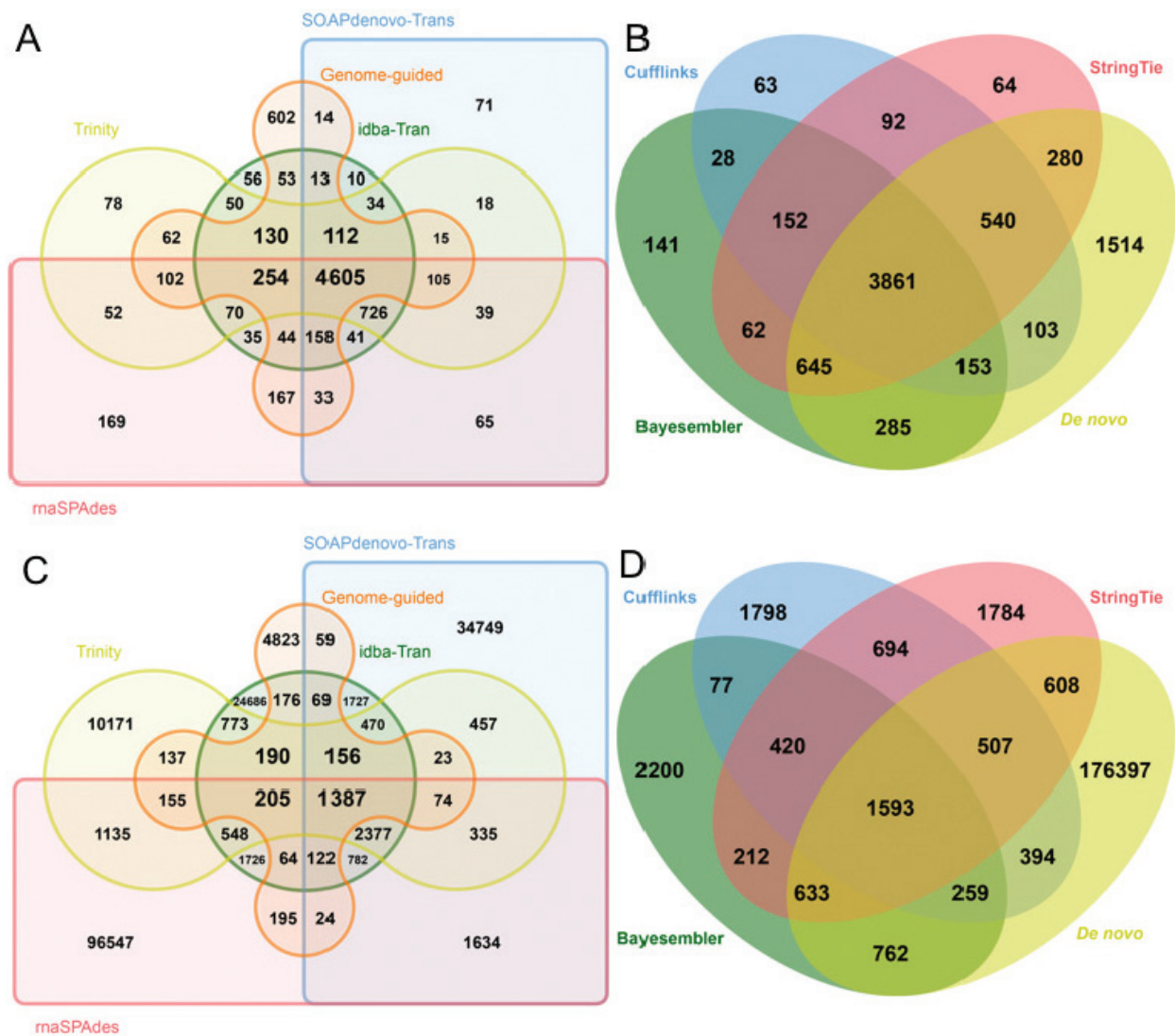


Figure 3. Comparisons of the results among *de novo* and genome-guided transcriptome assemblers. For each *de novo* assembler, results from multiple kmers were pooled. Correctly (A) and incorrectly (C) assembled sequences for each *de novo* assembler are compared with the combined results from genome-guided assemblers. Correctly (B) and incorrectly (D) assembled sequences for each genome-guided assembler are compared with the combined results from *de novo* assemblers.

In addition to the two ensemble methods, we also included three “consensus” approaches taking the consensus of the pooled *de novo* methods. These consensus assemblies involve keeping all of the unique protein sequences produced by any two, three and four tools (named Consensus 2, Consensus 3 and Consensus 4, respectively). Note that Consensus 4 is a subset of Consensus 3, and Consensus 3 is a subset of Consensus 2.

The performance of these ensemble strategies is shown in **Table 5**. Both of EvidentialGene and Concatenation resulted in an over-estimation in the number of transcripts present. Interestingly, while Concatenation produced a larger total number of transcripts (19,767) than EvidentialGene (19,177), ~2300 of those sequences were redundant, leading to fewer unique sequences (17,497 by

Concatenation). Additionally, Concatenation both kept more of the correctly assembled contigs from the individual *de novo* assemblies, and removed more of the incorrectly assembled contigs than EvidentialGene. These differences lead Concatenation to outperform EvidentialGene across every statistic (Table 6). The performance of the consensus approach varied based on the number of assemblers required.

Consensus 2 produced the most correctly assembled contigs of any method (6711), but at the cost of more incorrectly assembled contigs than Concatenation (14,433). However, both Consensus 3 and Consensus 4 kept the majority of the correctly assembled contigs while reducing the number of incorrectly assembled contigs by roughly half or three quarters, respectively. Consensus 4 had the highest precision (0.5861) and lowest FDR (0.4139) of any method. However, the additional reduction in the number of correctly assembled contigs lead to Consensus 3 having slightly higher accuracy* (0.2998) and F₁ score (0.4613).

In Figure 4 all individual methods (both *de novo* and genome-guided) as well as ensemble methods are compared. Concatenation performed more poorly than Trinity despite the Trinity assembly forming part of the ensemble. In contrast, Consensus 3 kept more correctly assembled contigs than any individual assembly, with fewer incorrectly assembled than any approach except Consensus 4. This test highlights the weakness of ensemble assembly strategies to retain the incorrect version of a transcript, even if the correct version of the transcript exists in the individual assemblies. More robust methods, such as the consensus approaches we presented here, are needed to reliably improve over individual assemblies.

Methods	Total	Unique	Correct (%)	Incorrect	C/I
EvidentialGene	19,177	19,175	2267 (16.15)	16,908	0.1341
Concatenation	19,767	17,497	4697 (33.45)	12,800	0.3670
Consensus 2	21,444	21,444	6711 (47.80)	14,433	0.4650
Consensus 3	12,600	12,600	6144 (43.76)	6456	0.9517
Consensus 4	9095	9095	5331 (37.97)	3764	1.416

Table 5. Performance of individual genome-guided assemblers using default parameters on simulated RNAseq library with both the same and different references genome as the benchmark transcriptome. See Table 1 for the description of numbers shown.

Methods	Precision	Recall	Accuracy*	F ₁	FDR
EvidentialGene	0.1182	0.1615	0.0733	0.1365	0.8818
Concatenation	0.2684	0.3345	0.1750	0.2979	0.7316
Consensus 2	0.3174	0.4780	0.2357	0.3815	0.6826
Consensus 3	0.4876	0.4376	0.2998	0.4613	0.5124
Consensus 4	0.5861	0.3797	0.2994	0.4609	0.4139

Table 6. Performance statistics of ensemble assembly strategies using *de novo* assemblies on simulated RNAseq library.

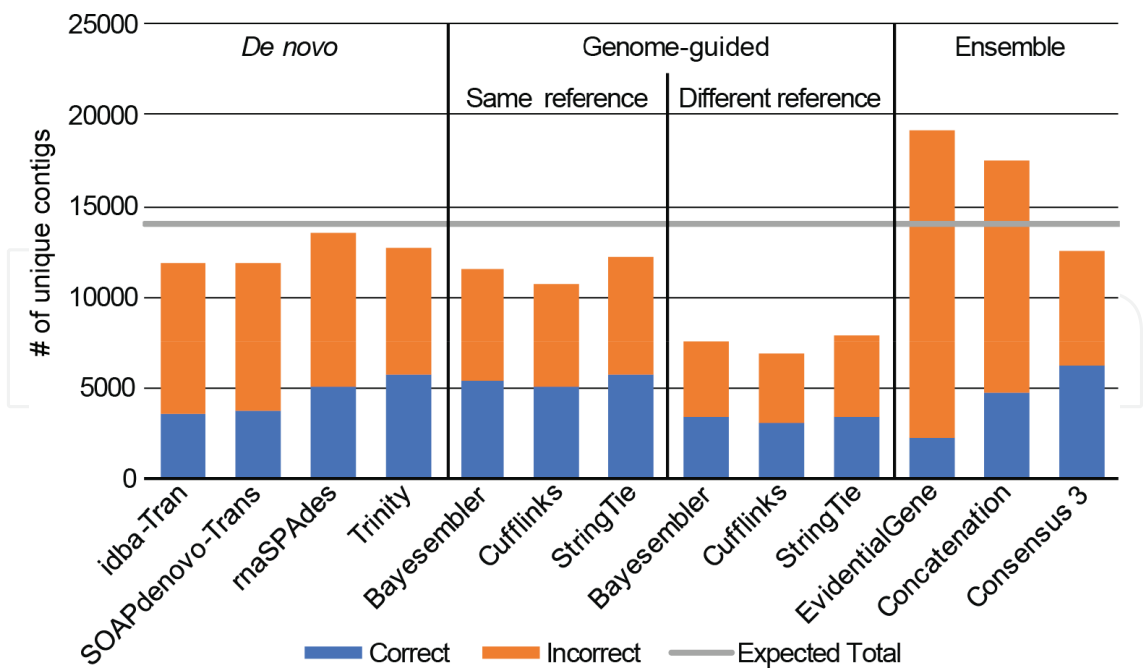


Figure 4. Performance comparison among all assemblers including *de novo*, genome-guided, and ensemble strategies. Simulated RNAseq data were used for testing, and the default parameters were used for each assembler. See **Tables 1, 3, and 5** for the actual numbers. The expected number of contigs is 14,040.

5. Conclusions

Transcriptome assembly can be approached from multiple different strategies. Historically, these approaches have revolved around assembling short but highly accurate Illumina reads with or without an existing genome assembly as a reference, referred to as genome-guided or *de novo* assemblers, respectively. All of the widely used *de novo* assemblers decompose the short reads into smaller kmers and use de Bruijn graphs built on these kmers to attempt to reconstruct the original transcripts. Due to the limitations of the de Bruijn graphs, this approach presents a trade-off between the uniqueness of the longer kmers and increased coverage of the shorter kmers. As a result, different kmer lengths can produce drastically different graphs, leading to large differences in the final assemblies.

Genome-guided assemblers avoid the limitations of the de Bruijn graphs by mapping the reads to the reference genome. This mapping, however, introduces its own limitations and trade-offs. Reads that are ambiguous between splice forms in the same genomic locations or across multiple genomic locations create similar challenges to the de Bruijn graphs. These ambiguities are compounded when the mapping must take into account mismatches due to sequencing errors as well as biological variations.

The limitations of the individual tools can potentially be overcome by combining multiple different assemblies in ensemble. As each tool and set of parameters results in a different set of correctly assembled contigs, accurately selecting these correctly assembled contigs without

selecting any redundant incorrectly assembled contigs would leverage the strengths of each methods without the weaknesses of any. However, currently available ensemble strategies cannot guarantee that the correct sequence is chosen, leading to ensemble assemblies that are less accurate than individual assemblies. As the selection criteria for ensemble methods improve, such as with the “Consensus” approach shown here, these methods can also leverage new assembly approaches that can better handle certain subsets of transcripts (e.g. alternative splice forms) that may have other weaknesses that prevent them from being competitive as a general transcript assembly tool.

Overall, as our results demonstrated, transcriptome assemblers can still be improved, regardless of the approach used. While the genome-guided assemblers generally perform best when the assembly is performed against the same reference sequence that the RNAseq data was generated from, this is not always possible. When these sequences differ, the genome-guided assemblers may have lower accuracy than the *de novo* assemblers. While ensemble assembly strategies can potentially improve on accuracy over individual assemblies, it is also possible that they instead reduce the accuracy. Improving the performance of these tools, whether individual assemblers, ensemble strategies, or combined with long-read sequencing, will improve not only the accuracy of the reconstructed transcriptome but also the accuracy of downstream analyses, such as sequence annotation, quantification, and differential expression.

Acknowledgements

This work was supported by a grant from the National Science Foundation to ENM (Award #: 1339385).

Author details

Adam Voshall and Etsuko N. Moriyama*

*Address all correspondence to: emoriyama2@unl.edu

School of Biological Sciences and Center for Plant Science Innovation, University of Nebraska-Lincoln, Lincoln, NE, USA

References

- [1] Wang Z, Gerstein M, Snyder M. RNA-Seq: A revolutionary tool for transcriptomics. *Nature Reviews. Genetics*. 2009;**10**:57-63. DOI: 10.1038/nrg2484
- [2] Ozsolak F, Milos PM. RNA sequencing: Advances, challenges and opportunities. *Nature Reviews. Genetics*. 2011;**12**:87-98. DOI: 10.1038/nrg2934

- [3] Huang X, Chen XG, Armbruster PA. Comparative performance of transcriptome assembly methods for non-model organisms. *BMC Genomics*. 2016;**17**:523. DOI: 10.1186/s12864-016-2923-8
- [4] Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, Szczesniak MW, Gaffney DJ, Elo LL, Zhang X, Mortazavi A. A survey of best practices for RNA-seq data analysis. *Genome Biology*. 2016;**17**:13. DOI: 10.1186/s13059-016-0881-8
- [5] Simonis M, Atanur SS, Linsen S, Guryev V, Ruzius FP, Game L, Lansu N, de Bruijn E, van Heesch S, Jones SJ, Pravenec M, Aitman TJ, Cuppen E. Genetic basis of transcriptome differences between the founder strains of the rat HXB/BXH recombinant inbred panel. *Genome Biology*. 2012;**13**:r31. DOI: 10.1186/gb-2012-13-4-r31
- [6] Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S. TransRate: Reference-free quality assessment of de novo transcriptome assemblies. *Genome Research*. 2016;**26**:1134-1144. DOI: 10.1101/gr.196469.115
- [7] Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, et al. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*. 2011;**29**:644-652. DOI: 10.1038/nbt.1883
- [8] Xie Y, Wu G, Tang J, Luo R, Patterson J, Liu S, Huang W, He G, Gu S, Li S, Zhou X, Lam TW, Li Y, Xu X, Wong GK, Wang J. SOAPdenovo-trans: *De novo* transcriptome assembly with short RNA-Seq reads. *Bioinformatics*. 2014;**30**:1660-1666. DOI: 10.1093/bioinformatics/btu077
- [9] Peng Y, Leung HC, Yiu SM, Lv MJ, Zhu XG, Chin FY. IDBA-tran: A more robust *de novo* de Bruijn graph assembler for transcriptomes with uneven expression levels. *Bioinformatics*. 2013;**29**:i326-i334. DOI: 10.1093/bioinformatics/btt219
- [10] Zerbino DR, Birney E. Velvet: Algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Research*. 2008;**18**:821-829. DOI: 10.1101/gr.074492.107
- [11] Schulz MH, Zerbino DR, Vingron M, Birney E. Oases: Robust *de novo* RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*. 2012;**28**:1086-1092. DOI: 10.1093/bioinformatics/bts094
- [12] Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, Lesin VM, Nikolenko SI, Pham S, Prjibelski AD, Pyshkin AV, Sirotkin AV, Vyahhi N, Tesler G, Alekseyev MA, Pevzner PA. SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. *Journal of Computational Biology*. 2012;**19**:455-477. DOI: 10.1089/cmb.2012.0021
- [13] Chevreux B, Pfisterer T, Drescher B, Driesel AJ, Muller WEG, Wetter T, Suhai S. Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs. *Genome Research*. 2004;**14**:1147-1159. DOI: 10.1101/gr.1917404

- [14] Martin JA, Wang Z. Next-generation transcriptome assembly. *Nature Reviews. Genetics*. 2011;**12**:671-682. DOI: 10.1038/nrg3068
- [15] Koren S, Treangen TJ, Hill CM, Pop M, Phillippy AM. Automated ensemble assembly and validation of microbial genomes. *BMC Bioinformatics*. 2014;**15**:126. DOI: 10.1186/1471-2105-15-126
- [16] Deng X, Naccache SN, Ng T, Federman S, Li L, Chiu CY, Delwart EL. An ensemble strategy that significantly improves *de novo* assembly of microbial genomes from metagenomic next-generation sequencing data. *Nucleic Acids Research*. 2015;**43**:e46. DOI: 10.1093/nar/gkv002
- [17] Trapnell C, Pachter L, Salzberg SL. TopHat: Discovering splice junctions with RNA-Seq. *Bioinformatics*. 2009;**25**:1105-1111. DOI: 10.1093/bioinformatics/btp120
- [18] Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*. 2013;**29**:15-21. DOI: 10.1093/bioinformatics/bts635
- [19] Kim D, Langmead B, Salzberg SL. HISAT: A fast spliced aligner with low memory requirements. *Nature Methods*. 2015;**12**:357-360. DOI: 10.1038/nmeth.3317
- [20] Medina I, Tarraga J, Martinez H, Barrachina S, Castillo MI, Paschall J, Salavert-Torres J, Blanquer-Espert I, Hernandez-Garcia V, Quintana-Orti ES, Dopazo J. Highly sensitive and ultrafast read mapping for RNA-seq analysis. *DNA Research*. 2016;**23**:93-100. DOI: 10.1093/dnares/dsv039
- [21] Maretty L, Sibbesen JA, Krogh A. Bayesian transcriptome assembly. *Genome Biology*. 2014;**15**:501. DOI: 10.1186/s13059-014-0501-4
- [22] Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*. 2010;**28**(5):511. DOI: 10.1038/nbt.1621
- [23] Pertea M, Pertea GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*. 2015;**33**:290-295. DOI: 10.1038/nbt.3122
- [24] Nakasugi K, Crowhurst R, Bally J, Waterhouse P. Combining transcriptome assemblies from multiple *de novo* assemblers in the allo-tetraploid plant *Nicotiana benthamiana*. *PLoS One*. 2014;**9**:e91776. DOI: 10.1371/journal.pone.0091776
- [25] Gilbert D. Gene-omes built from mRNA seq not genome DNA. 7th Annual Arthropod Genomics Symposium Notre Dame. 2013
- [26] Cerveau N, Jackson DJ. Combining independent *de novo* assemblies optimizes the coding transcriptome for nonconventional model eukaryotic organisms. *BMC Bioinformatics*. 2016;**17**:525. DOI: 10.1186/s12859-016-1406-x

- [27] Abdel-Ghany SE, Hamilton M, Jacobi JL, Ngam P, Devitt N, Schilkey F, Ben-Hur A, Reddy AS. A survey of the sorghum transcriptome using single-molecule long reads. *Nature Communications*. 2016;**7**:11706. DOI: 10.1038/ncomms11706
- [28] Salmela L, Walve R, Rivals E, Ukkonen E. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics*. 2017;**33**:799-806. DOI: 10.1093/bioinformatics/btw321
- [29] Salmela L, Rivals E. LoRDEC: Accurate and efficient long read error correction. *Bioinformatics*. 2014;**30**:3506-3514. DOI: 10.1093/bioinformatics/btu538
- [30] Hargreaves AD, Mulley JF. Assessing the utility of the Oxford Nanopore MinION for snake venom gland cDNA sequencing. *PeerJ*. 2015;**3**:e1441. DOI: 10.7717/peerj.1441
- [31] Cheng B, Furtado A, Henry RJ. Long-read sequencing of the coffee bean transcriptome reveals the diversity of full-length transcripts. *Gigascience*. 2017;**6**:1-13. DOI: 10.1093/gigascience/gix086
- [32] Mei W, Liu S, Schnable JC, Yeh CT, Springer NM, Schnable PS, Barbazuk WB. A comprehensive analysis of alternative splicing in paleopolyploid maize. *Frontiers in Plant Science*. 2017;**8**:694. DOI: 10.3389/fpls.2017.00694
- [33] Sharon D, Tilgner H, Grubert F, Snyder M. A single-molecule long-read survey of the human transcriptome. *Nature Biotechnology*. 2013;**31**:1009-1014. DOI: 10.1038/nbt.2705
- [34] Minoche AE, Dohm JC, Schneider J, Holtgrawe D, Viehover P, Montfort M, Sorensen TR, Weisshaar B, Himmelbauer H. Exploiting single-molecule transcript sequencing for eukaryotic gene prediction. *Genome Biology*. 2015;**16**:184. DOI: 10.1186/s13059-015-0729-7
- [35] Zhang SJ, Wang C, Yan S, Fu A, Luan X, Li Y, Sunny Shen Q, Zhong X, Chen JY, Wang X, Chin-Ming Tan B, He A, Li CY. Isoform evolution in primates through independent combination of alternative RNA processing events. *Molecular Biology and Evolution*. 2017;**34**:2453-2468. DOI: 10.1093/molbev/msx212
- [36] Earl D, Bradnam K, St John J, Darling A, Lin D, Fass J, Yu HO, Buffalo V, Zerbino DR, Diekhans M, Nguyen N, Ariyaratne PN, Sung WK, Ning Z, Haimel M, Simpson JT, Fonseca NA, Birol I, Docking TR, Ho IY, et al. Assemblathon 1: A competitive assessment of *de novo* short read assembly methods. *Genome Research*. 2011;**21**:2224-2241. DOI: 10.1101/gr.126599.111
- [37] Piovesan A, Caracausi M, Antonaros F, Pelleri MC, Vitale L. GeneBase 1.1: A tool to summarize data from NCBI gene datasets and its application to an update of human gene statistics. *Database: The Journal of Biological Databases and Curation*. 2016;**2016**: Article number: baw153. DOI: 10.1093/database/baw153
- [38] O'Neil ST, Emrich SJ. Assessing *de novo* transcriptome assembly metrics for consistency and utility. *BMC Genomics*. 2013;**14**:465. DOI: 10.1186/1471-2164-14-465
- [39] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. *Journal of Molecular Biology*. 1990;**215**:403-410. DOI: 10.1016/S0022-2836(05)80360-2

- [40] The UniProt Consortium. UniProt: The universal protein knowledgebase. *Nucleic Acids Research*. 2017;**45**:D158-D169. DOI: 10.1093/nar/gkw1099
- [41] Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR, Heger A, Hetherington K, Holm L, Mistry J, Sonnhammer EL, Tate J, Punta M. Pfam: The protein families database. *Nucleic Acids Research*. 2014;**42**:D222-D230. DOI: 10.1093/nar/gkt1223
- [42] Thomas PD, Campbell MJ, Kejariwal A, Mi H, Karlak B, Daverman R, Diemer K, Muruganujan A, Narechania A. PANTHER: A library of protein families and subfamilies indexed by function. *Genome Research*. 2003;**13**:2129-2141. DOI: 10.1101/gr.772403
- [43] Waterhouse RM, Seppey M, Simao FA, Manni M, Ioannidis P, Klioutchnikov G, Kriventseva EV, Zdobnov EM. BUSCO applications from quality assessments to gene prediction and phylogenomics. *Molecular Biology and Evolution*. 2018;**35**:543-548. DOI: 10.1093/molbev/msx319
- [44] Besemer J, Lomsadze A, Borodovsky M. GeneMarkS: A self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*. 2001;**29**:2607-2618
- [45] Wang S, Gribskov M. Comprehensive evaluation of *de novo* transcriptome assembly programs and their effects on differential gene expression analysis. *Bioinformatics*. 2017;**33**: 327-333. DOI: 10.1093/bioinformatics/btw625
- [46] Li B, Dewey CN. RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*. 2011;**12**:323. DOI: 10.1186/1471-2105-12-323
- [47] Griebel T, Zacher B, Ribeca P, Raineri E, Lacroix V, Guigo R, Sammeth M. Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research*. 2012;**40**:10073-10083. DOI: 10.1093/nar/gks666
- [48] Ono Y, Asai K, Hamada M. PBSIM: PacBio reads simulator—Toward accurate genome assembly. *Bioinformatics*. 2013;**29**:119-121. DOI: 10.1093/bioinformatics/bts649
- [49] Yang C, Chu J, Warren RL, Birol I. NanoSim: Nanopore sequence read simulator based on statistical characterization. *Gigascience*. 2017;**6**:1-6. DOI: 10.1093/gigascience/gix010
- [50] Del Fabbro C, Scalabrin S, Morgante M, Giorgi FM. An extensive evaluation of read trimming effects on Illumina NGS data analysis. *PLoS One*. 2013;**8**:e85024. DOI: 10.1371/journal.pone.0085024
- [51] Crusoe MR, Alameldin HF, Awad S, Boucher E, Caldwell A, Cartwright R, Charbonneau A, Constantinides B, Edverson G, Fay S, Fenton J, Fenzl T, Fish J, Garcia-Gutierrez L, Garland P, Gluck J, Gonzalez I, Guermond S, Guo J, Gupta A, et al. The khmer software package: Enabling efficient nucleotide sequence analysis. *F1000Research*. 2015;**4**:900. DOI: 10.12688/f1000research.6924.1