Plant Transcriptome Assembly: Review and Benchmarking

Sairam Behera[1] • Adam Voshall[1,2,3] • Etsuko N. Moriyama[3,4]

[1]Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA; [2]Boston Children's Hospital/Harvard Medical School, Boston, MA, USA; [3]School of Biological Sciences, University of Nebraska-Lincoln, Lincoln, NE, USA; [4]Center for Plant Science Innovation, University of Nebraska-Lincoln, Lincoln, NE, USA

Author for correspondence: Etsuko N. Moriyama, School of Biological Sciences and Center for Plant Science Innovation, University of Nebraska-Lincoln, Lincoln, NE, USA. Email: emoriyama2@unl.edu

Doi: https://doi.org/10.36255/exonpublications.bioinformatics.2021.ch7

Abstract: Transcriptome assembly using next-generation sequencing data is an important step in a wide range of biological studies at the molecular level. The quality of computationally assembled transcriptomes affects various downstream analyses, such as gene structure prediction, isoform identification, and gene expression analysis. However, the actual accuracy of assembled transcriptomes is usually unknown. Furthermore, assembly quality depends on various factors such as the method used, the parameters (for example, *k*-mers) used with the method, and the transcript to be assembled. Users often choose an assembly method based solely on availability without considering differences among methods, as well as choices of the parameters. This is partly due to the lack of suitable benchmarking datasets. In this chapter, we provide a review of computational approaches used for transcriptome assembly (genome-guided, *de novo*, and ensemble), factors that affect assembly performance including those particularly important for plant transcriptomes, and how the transcriptome assembly performance can be assessed. Using examples from plant transcriptomes, we further illustrate how simulated benchmark datasets can be generated and used to

In: *Bioinformatics*. Nakaya HI (Editor). Exon Publications, Brisbane, Australia.
ISBN: 978-0-6450017-1-6; Doi: https://doi.org/10.36255/exonpublications.bioinformatics.2021

compare the quality of transcriptome assemblies and how the performance of transcriptome assemblers can be assessed using various metrics.

Keywords: benchmarking; isoform; plants; simulation; transcriptome assembly

INTRODUCTION

A transcriptome is the entire set of transcripts in a cell. The content of a transcriptome varies between cell types and developmental stages. Understanding the content of transcriptomes and tracking their spatial and temporal differentiation is important when we study the mechanisms of cellular differentiation, carcinogenesis, and gene regulation. RNA-sequencing (RNA-seq) is a transcriptome profiling technology that utilizes high-throughput next-generation sequencing. The majority of RNA-seq data are generated from the complementary DNAs (cDNAs) converted from messenger RNAs (mRNAs) by using the Illumina short-read sequencing platform (1, 2). More recently, long-read and direct-RNA sequencing has also become available for RNA-seq using third-generation sequencing platforms, such as Pacific Biosciences (PacBio) and Oxford Nanopore (ONT) (3).

RNA-seq provides a quantitative snapshot of a transcriptome of the cells at a given time point. RNA-seq data can be used to reconstruct transcriptomes and also to analyze differential gene expression and differential splicing of mRNAs. However, many challenges remain in assembling transcripts correctly using the available assembly algorithms (4). Sequencing errors and the presence of repetitive sequences are often the cause of mis-assembly of transcripts. Shared exon regions and different expression levels among alternatively spliced transcripts (isoforms) make the identification and quantification of genes and isoforms challenging for transcriptome assembly and quantification tools (5). For many plant species, polyploidy adds another level of complexity for transcriptome assembly. The high sequence similarity among sub-genomes, duplicated genes, and isoforms all make the *de novo* transcriptome assembly a significant challenge (6, 7).

In the following sections, we first review three transcriptome assembly strategies: genome-guided, *de novo*, and ensemble. Next, we describe how the transcriptome assembly performance can be evaluated. We discuss the advantages of using simulated benchmark data instead of actual data and outline how such simulated benchmark transcriptome datasets can be generated. Finally, we demonstrate how transcriptome assemblies generated from different methods can be compared and how the transcriptome assembly quality can be evaluated using simulated plant transcriptomes with varied complexity.

TRANSCRIPTOME ASSEMBLY STRATEGIES

Transcriptome assembly is a process of reconstructing the complete set of full-length transcripts from RNA-seq data, which often include tens of millions of short-read sequences. Genome assembly methods cannot be used for transcriptome assembly due to drastically varied sequencing depth among transcripts (due to gene-expression variation), strand-specific experiments with RNA-seq, and existence of isoforms. For transcriptome assembly, genome-guided or reference-based assembly methods are preferred when a high-quality reference genome is available (2, 8). *De novo* or reference-free transcriptome assembly methods do not require reference genomes. These methods are particularly useful for non-model organisms where high-quality reference genomes are often not available (9–11).

Genome-guided approach

The genome-guided approach of transcriptome assembly makes use of a genome sequence while reconstructing the transcripts (12). These approaches first map the sequenced reads to the reference genome using a splice-aware aligner such as TopHat2 (13), HISAT2 (14), or STAR (15). The mapping information is then used to construct a graph that represents the splice junction of the transcripts (splice graph). The final transcripts are extracted by traversing the graph. Bayesembler (16), Cufflinks (17), StringTie (18), and Scallop (19) are some examples of commonly used genome-guided assembly tools. To handle the presence of introns in the genome, the aligners take splice-junction sites into consideration and allow split-mapping where one part of a read is mapped to one exon and another part to another exon. One issue with using short reads is that they can be mapped to multiple locations in the genome due to the existence of repetitive sequences or highly similar duplicated genes. The read-mapping strategies used by different aligners handle such ambiguities differently (20). The techniques used to construct the graph and the contig sequences from the mapping information are also different among the methods. Selection of aligners and assembly methods, therefore, has a significant impact on the assembly results. The availability of a high-quality reference genome is also necessary for accurate assembly. If the read sequences and the reference genome are not from the same strain of the same species, the resulting divergence in the read and reference sequences could also cause assembly mistakes.

Cufflinks is one of the most widely used genome-guided transcriptome assemblers (17). It can be used not only to assemble transcripts but also to estimate their abundance and to test differential expression. Cufflinks constructs an overlap graph based on the alignments of the overlapping reads on the genome. Transcripts are identified by traversing the minimal paths that cover all alignments in the graph (each path represents a different isoform). Since Cufflinks performs transcriptome assembly and expression-level estimation separately, it does not consider transcript abundance when finding the minimal set of transcripts. StringTie simultaneously assembles transcripts and estimates their expression levels (18). From the clusters of reads mapped to the genome, it creates a splice graph for each cluster. It then traverses the splice graph to construct transcripts. For each transcript, it creates a flow network to estimate its expression level using an optimization technique known as the maximum flow algorithm. This information is iteratively used to update the splice graph. Scallop, a more recent genome-guided tool, also creates a splice graph from the clustered reads mapped on the genome (19). It preserves phasing paths using the reads that span more than two exons. By iteratively decomposing each splice graph, it reduces false transcripts. By incorporating phasing information, Scallop achieves improved assembly of multi-exon transcripts and lowly expressed transcripts.

De novo approach

The *de novo* approach of transcriptome assembly reconstructs transcript sequences from short reads without using a reference genome. Most of the *de novo* transcriptome assembly techniques use the de Bruijn graph based on *k*-mers (21), which include Trinity (22, 23), IDBA-Tran (24), SOAPdenovo-Trans (25), and rnaSPAdes (26). A *k*-mer of a sequence is a subsequence of length *k*, that is, *k* consecutive nucleotides. During the assembly process, each sequence is decomposed into all possible fixed size *k*-mers. The nodes or vertices of a de Bruijn graph are represented by the *k*-mers. An edge is created between two nodes if the corresponding *k*-mers have a suffix-prefix overlap of length *k*-1, that is, the last *k*-1 nucleotides of one *k*-mer exactly match with first *k*-1 nucleotides of the other *k*-mer. Two consecutive *k*-mers of a sequence, therefore, can be represented as two nodes with an edge between them. Thus, a de Bruijn graph represents a set of reads as each read induces a sequence of edges that joins a sequence of vertices, that is, a path. If two read sequences share a subsequence, then a common path is induced in the graph. If two read sequences have a suffix-prefix overlap, then a single path is induced for both sequences. After a de Bruijn graph is constructed, different paths are traversed to generate the putative transcripts. Note that if the reads are derived from highly similar (but not identical) sequences, they create isolated nodes and loops, which affects the accuracy of the graph construction. Sequencing errors can also cause false *k*-mers (those containing erroneous nucleotides) to participate in the graph construction by creating false nodes. The false nodes either break the path or create a false path if overlapped with another *k*-mer.

For de Bruijn graph-based assembly methods, the choice of the *k*-mer size plays an important role on the quality of the assembly, and also creates trade-offs between several effects (27). While short *k*-mers are expected to cover the original transcript fully and resolve the problems caused by errors in the sequences, they also create ambiguity because they can be shared among multiple transcripts. If repeats are longer than *k*, it creates forks in the graph, which causes the contig to break up. Longer *k*-mers, on the other hand, are expected to have higher chances of containing sequence errors. Errors in the *k*-mers cause the loss of overlap information, which affects the accuracy of the de Bruijn graph construction. In reality, it is difficult to determine which *k*-mer size generates the optimal assembly for a given data using a given assembler. Different assemblers result in different sets of transcripts even if they are used with *k*-mers of the same size. When the same method is used with different *k*-mer sizes, assembly outputs can be also different.

Trinity includes three modules: Inchworm, Chrysalis, and Butterfly (22, 23). Inchworm removes erroneous *k*-mers from the read sequences and then uses a greedy-extension based overlap method to assemble reads into contigs. Chrysalis clusters the contigs and constructs a de Bruijn graph for each cluster. Finally, Butterfly traverses the graphs to construct transcripts. SOAPdenovo-Trans is an extension of the SOAPdenovo2 genome assembler (25, 28). It uses the error removal methods of Trinity to remove edges representing the erroneous *k*-mers. The contigs extracted from the de Bruijn graphs are mapped to reads to build linkage between them, and the contigs are clustered into subgraphs based on the linkage information. Finally, each subgraph is traversed to generate the transcripts. The default *k*-mer sizes for Trinity and SOAPdenovo-Tran are 23 and 25, respectively.

IDBA-Tran uses a unique assembly strategy (24). It iterates *k*-mers from small to large *k* (*k*=20 to 60 in every 10 in default) to balance the advantages and limitations of *k*-mer sizes. For each *k*-mer, it constructs a de Bruijn graph and then travers the graph to generate contigs. The results from different *k*-mer sizes are merged by including the contigs generated with smaller *k*-mers as part of the input in the next iteration with a larger *k*-mer. rnaSPAdes is an extension of the SPAdes genome assembler (26, 29). The de Bruijn graph used in SPAdes was modified for transcriptome assembly to handle paired-end reads, uneven coverage, and multiple insert sizes. Similar to IDBA-Tran, iterative de Bruijn graph construction was used but with only two *k*-mer sizes (one small and one large) dynamically selected using the input read data information.

Ensemble approach

No single assembler is considered to be the optimal for a wide range of input data (8, 30). While it is possible to increase the true transcript reconstruction by combining the assembly results of multiple assemblers, this approach can also increase the number of mis-assembled transcripts. The ensemble approach of transcriptome assembly attempts to reduce the number of mis-assembled transcripts without removing correctly assembled transcripts. EvidentialGene (31) and the method proposed in (32) (we call this method "Concatenation") merge multiple *de novo* assemblies and cluster contigs using either CD-HIT (33) or BLAST (34, 35) and select the representative sequences for the final assembly set. We previously reported a consensus strategy where multiple *k*-mers are considered for assembly and simple voting is used to select the contigs that are assembled by at least three out of four *de novo* assemblers for the final assembly set (8). TransBorrow (36) is an ensemble approach that combines the results from different genome-guided assemblers. TransBorrow first extracts reliable subpaths supported by paired-end reads from a splice graph. Transcripts assembled by multiple genome-guided methods are merged and colored graphs representing the merged transcripts are built. Reliable assembly subpaths are further extracted based on the number of assemblers that detected each subpath (transcript). After combining reliable assembly subpaths and reliable subpaths on the splicing graphs, the final transcripts are assembled.

HOW TO EVALUATE TRANSCRIPTOME ASSEMBLY PERFORMANCE

To evaluate the transcriptome assembly performance, quantification of the accuracy of assembled transcriptomes is necessary. Assembly performance metrics can be grouped into two classes: reference-free and reference-based. The reference-based metrics are further grouped into those based on real biological data and those based on simulated benchmark data.

Performance metrics without references

When high-quality reference sequences are not available to provide the ground truth, some assembly statistics can be used as reference-free performance statistics. Some commonly used assembly statistics include: (i) number of contigs; (ii) median contig length (bp); and (iii) N50 (or Nx), a length-weighted median where the sum of the lengths (bp) of all contigs longer than the N50 (or Nx) is at least 50% (or x%) of the total length of the assembly.

rnaQUAST (37), for example, can be used to obtain these metrics. Higher values of N50 (Nx) indicate that a greater number of reads are overlapped to form longer contigs. In contrast to genome assembly, where longer contigs (for example, larger N50) indicate a higher quality assembly, a transcriptome includes transcripts with varied lengths. The longer contigs in a transcriptome assembly could also represent over-assembly or chimeric contigs. Therefore, for a transcriptome assembly, the length-based metrics are not always useful as accuracy measures (38).

DETONATE provides a model-based score, RSEM-EVAL (39). It combines the compactness of an assembly and the support of the assembly from the RNA-seq reads into a single score based on their joint probability. Higher RSEM-EVAL scores indicate better assembly performance.

TransRate (30) provides an assembly score, (v) as shown below, based on the four contig scores (i)–(iv):

(i) $s(C_{\text{nuc}})$: measures the extent to which the nucleotides in the mapped reads are the same as those in the assembled contig
(ii) $s(C_{\text{cov}})$: measures the proportion of nucleotides in the contig that have zero coverage
(iii) $s(C_{\text{ord}})$: measures the extent to which the order of the bases in contig are correct
(iv) $s(C_{\text{seg}})$: measures the probability that the coverage depth of the transcript is univariate, which represents a single-transcript assembly, not a hybrid/chimeric assembly
(v) TransRate assembly score (T): the geometric mean of the four contig scores multiplied by the proportion of RNA-seq reads that provide positive supports for the assembly (that map to the assembly)

Performance metrics using actual biological data

When the references (either genome or transcriptome sequences) are available, reference-based metrics can be calculated. rnaQUAST (37), for example, provides the gene-level metrics (for example, numbers of assembled genes, isoforms, or exons and their lengths) as well as the alignment metrics (for example, numbers of aligned, unaligned, or misassembled transcripts).

DETONATE provides a tool kit, REF-EVAL (39), which computes a number of reference-based scores including:

(i) Recall, Precision, and F_1: calculated at contig or nucleotide-level (see the equations [3] - [5] below)

(ii) KC (*k*-mer compression) score: measures the accuracy of the assembly based on the weighted *k*-mer recall and the compression ratio between the assembly and the RNA-seq data.

The quality of the assembly can be also evaluated based on the proportion of the predicted gene or protein sequences matched with those in the database of known genes or proteins. BUSCO (40), for example, provides a quantitative assessment of the completeness of an assembly in terms of the expected content of the lineage-specific gene dataset. The Benchmarking Universal Single-Copy Orthologs (BUSCO) is extracted from OrthoDB (41). Orthologous gene candidates are searched at the protein level in the assembly and the results are summarized into four categories: complete and single-copy, complete and duplicated, fragmented, and missing. In the comprehensive study reported in (42), these metrics were used to compare ten *de novo* assemblers using nine actual RNA-seq datasets.

Performance metrics using simulated benchmark data

Simulation can provide a way to generate benchmark datasets where the ground truth is known. This is advantageous over using actual biological data as the reference, where the ground truth cannot be known completely. For a transcriptome analysis, RNA-seq can be simulated to generate short reads derived from a set of transcripts whose sequences are known. The simulated reads are used with assembly methods and the assembled contigs are compared with the original transcripts. This is also the only way where the information about the transcripts that are not assembled (missing transcripts) can be fully evaluated.

A contig generated by an assembler is considered to be correctly assembled (positive) if the identical sequence is present in the reference transcriptome in the benchmark dataset. A contig is considered to be mis-assembled (negative) if the identical sequence is not present in the reference transcriptome in the benchmark dataset. Note that less stringent evaluation can be performed by using a lower threshold (< 100%) to identify positive contigs. It is also possible to use a protein-level similarity instead of a nucleotide-level similarity to identify positive contigs. The test results are categorized as the following three outcomes:

(i) True positive (TP): a correctly assembled contig
(ii) False positive (FP): a mis-assembled contig (including both partially correctly assembled and those with no similarity with the reference)
(iii) False negative (FN): a benchmark transcript that is missing in the assembly

Note that true negative (TN) can be counted only if the benchmark dataset includes a negative transcript set (transcript sequences that do not belong to the reference set) and the assembly experiments are done including reads that are derived from negative transcripts.

The performance of each assembler is evaluated by the following metrics:

- Correct/incorrect ratio $(\mathrm{C/I}) = \frac{TP}{FP}$ [1]

- $$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \text{ or Accuracy*} = \frac{TP}{TP+FP+FN} \qquad [2]$$

- $$\text{Recall}\left(\text{or Sensitivity}\right) = \frac{TP}{TP+FN} \qquad [3]$$

- $$\text{Precision} = \frac{TP}{TP+FP} = 1-\text{False Discovery Rate}\left(\text{FDR}\right) \qquad [4]$$

- $$\text{F}-\text{measure}\left(\text{F or F}_1\right) = \frac{2(TP)}{2(TP)+FP+FN} \qquad [5]$$

In the equations above, *TP*, *FP*, *TN*, and *FN* are the numbers of instances in those categories. As shown in the equation [2], when *TN* is not counted, Accuracy cannot be calculated. In such cases, we define a modified accuracy (Accuracy*) without using *TN*.

The higher C/I shows that among the assembled contigs (predicted positives) there are more correctly assembled contigs (*TP*) than mis-assembled contigs (*FP*). This is similar to Precision where the proportion of correctly assembled contigs (*TP*) is shown relative to all assembled contigs. Recall also shows the proportion of correctly assembled contigs (*TP*) but relative to the number of transcripts in the reference (actual positives). Accuracy (or Accuracy*) and F-measure are combined metrics. F-measure is useful because it balances the concerns of Recall and Precision and does not require *TN* to be counted.

All the above metrics can be calculated at both the nucleotide and protein sequence levels. Depending on the transcriptome assembly algorithms, the 5' and 3'-ends of contigs are defined differently. Such small differences at the 5' and 3'-ends could have significant effects on the TP counts. By using the protein-level accuracy, this issue can be avoided. However, the performance metrics can also be affected depending on how the gene-prediction algorithm used to identify the open reading frame (ORF) from each contig works.

Although the assembly performance metrics calculated using simulated benchmark datasets are expected to provide better evaluation of the performance of transcriptome assemblers, challenges remain on how biologically realistic the simulation of RNA-seq data can be. If the read distribution and sequencing errors, for example, are not modeled properly, assemblers may perform well on simulated data but poorly on real data or *vice versa*.

HOW TO GENERATE SIMULATED BENCHMARK TRANSCRIPTOME DATASETS

To analyze the performance of transcriptome assemblies, each of the benchmark transcriptome datasets should include the annotated genome, the transcriptome from which simulated RNA-seq is performed, and the RNA-seq data. In this section, we first briefly describe the methods that can be used to simulate RNA-seq. We then discuss protocols to generate simulated benchmark datasets.

RNA-seq simulation methods

There are several tools that can simulate RNA-seq with short-read sequencing using the Illumina platform and/or third-generation long-read sequencing using the PacBio SMRT and ONT MinION platforms (43). Many short-read simulators developed for benchmarking transcript abundance and differential expression tools, such as RSEM (44), SimSeq (45), SPsimSeq (46), and seqgendiff (47), model the error distribution and changes in transcript expression found in real RNA-seq datasets. This modeling can include sequence specific bias, such as producing fewer GC-rich reads (48), as in an extension to Polyester (49). Some short-read simulators, such as Flux Simulator (50), attempt to reconstruct each step of the library preparation and sequencing pipeline, mimicking the errors and biases introduced at each step. Long-read simulators, including PBSIM (51), LongISLND (52), Badread (53), and Trans-Nanosim (54), focus on identifying the statistical distribution of read lengths and errors within the reads, especially the prevalence of insertions or deletions, which are common in long reads but rare in short reads. Note that while Trans-Nanosim is the only long-read simulator specifically built for RNA-seq data, all of these simulators have been applied to introduce sequencing errors to model transcriptomic data.

Examples of RNA-seq simulation

To illustrate how the RNA-seq simulation is done, for this example, we used Flux Simulator (50). To model a range of transcriptome complexity, six genomes from four plant species including both monocots (*Oryza sativa* and *Zea mays*) and dicots (*Glycine max* and *Arabidopsis thaliana*) were chosen. The reference genome each simulation was based is listed in Table 1. Using these genome sequences and gene annotations provided in .gff files, RNA-seq simulation was performed as follows:

(i) The expression profile was generated by Flux Simulator using the reference genome. Flux Simulator in default assigns random expression levels to genes and transcripts.
(ii) Fragmentation of the expressed transcripts was done using a uniform random distribution. For this example, the lengths were set to 300 bp ± 150 bp. The fragments ≥ 150 bp were retained.
(iii) For sequencing, the Illumina Hi-Seq sequencing profile, which models sequencing errors, insert size, and transcript coverage, was used to generate 76 bp paired-end reads. For each transcriptome, a total of ~495 million reads were generated with more than 50X coverage for most transcripts.
(iv) For the reference set of transcripts, those that are mapped with sequenced reads with no gap in the coverage were chosen.
(v) ORFfinder (59) was used to identify the ORFs from each reference transcript, and the longest ORFs was chosen.
(vi) After removing the redundant sequences, the benchmark transcriptome was obtained at both nucleotide and protein levels.

TABLE 1 Comparison of transcriptome assembly performance among different methods[a]

	Genome-guided (same reference)			Genome-guided (different reference)			*De novo*[b]			
	Cufflinks	StringTie	Scallop	Cufflinks	StringTie	Scallop	IDBA-Tran	SOAPdenovo-Trans	Trinity	rnaSPAdes
[*A. thaliana* **No0** (CS6805): 18,875 (100, 1, 1)][c]										
				Reference: Col0						
# contigs[d]	19,288	21,027	21,397	21,178	20,264	18,817	22,768	29,773	23,476	27,664
Accuracy*	0.62	0.65	0.61	0.18	0.22	0.22	0.25	0.30	0.40	0.28
C/I	3.07	2.92	2.45	0.39	0.54	0.56	0.58	0.60	1.06	0.57
[*A. thaliana* **Col0** (TAIR9): 15,508 (79.03, 1.29, 8)][c]										
				Reference: No0						
# contigs[d]	15,768	16,908	18,055	17,441	16,470	17,179	20,449	21,371	19,409	31,494
Accuracy*	0.38	0.44	0.46	0.14	0.17	0.19	0.20	0.25	0.36	0.19
C/I	1.20	1.43	1.42	0.30	0.39	0.45	0.42	0.52	0.92	0.32
[**Soybean** (GCF_000004515.4): 18,215 (93.75, 1.07, 7)][c]										
# contigs[d]	18,823	20,887	19,355				33,243	52,700	24,346	23,686
Accuracy*	0.48	0.46	0.48				0.13	0.08	0.25	0.24
C/I	1.77	1.44	1.67				0.22	0.12	0.53	0.52
[**Rice** (GCF_001433935): 11,294 (97.97, 1.02, 3)][c]										
# contigs[d]	10,200	9,344	11,436				13,151	18,000	10,508	13,182
Accuracy*	0.39	0.40	0.48				0.16	0.17	0.30	0.28
C/I	1.42	1.74	1.80				0.36	0.30	0.93	0.69

Table continued on following page

TABLE 1 Comparison of transcriptome assembly performance among different methodsa (Continued)

	Genome-guided (same reference)			Genome-guided (different reference)			*De novo*[b]			
	Cufflinks	StringTie	Scallop	Cufflinks	StringTie	Scallop	IDBA-Tran	SOAPdenovo-Trans	Trinity	rnaSPAdes
[*Z. mays* **B73** (GCF_000005005): 17,108 (74.08, 1.5, 20)][c]										
				Reference: Mo17						
# contigs[d]	14,512	15,585	16,592	17,347	20,887	19,119	24,603	27,403	22,327	23,764
Accuracy*	0.26	**0.32**	0.26	0.08	0.09	0.10	0.11	0.08	**0.17**	0.11
C/I	0.79	**1.04**	0.71	0.18	0.18	0.21	0.20	0.13	**0.35**	0.20
[*Z. mays* **Mo17** (GCA_003185045.1): 17,479 (96.91, 1.04, 6)][c]										
				Reference: B73						
# contigs[d]	18,163	24,388	21,572	18,543	21,944	19,257	24,916	26,257	21,537	21,469
Accuracy*	**0.29**	0.24	0.26	0.08	0.08	0.08	0.13	0.09	**0.18**	0.16
C/I	**0.80**	0.50	0.60	0.18	0.15	0.17	0.24	0.16	**0.37**	0.33

[a] The best Accuracy* and C/I among all assemblers are shown in red. The scores in blue are the best among the *de novo* assemblers.

[b] The default *k*-mer sizes were used for the *de novo* assemblers.

[c] After each species name, the accession numbers of the reference genomic sequences used are shown in parentheses. The assembly of *A. thaliana* No0 (55) was downloaded from the 1001 genomes project (56). The assembly of *A. thaliana* Col0 was from the version 9 of the TAIR reference genome (57) and version 3 of the AtRTD transcriptome data set (58). The number after the colon is the total number of transcripts included in each benchmark dataset. The numbers in parentheses are % single-isoform gene, the average number of isoforms/gene, and the maximum number of isoforms/gene, in this order.

[d] The numbers of contigs are based on those unique at the protein sequence level.

Existence of isoforms in transcriptomes can impact the assembly performance. As shown in Table 1, a significant variation in the number of isoforms was incorporated among the six benchmark datasets. The *Z. mays* B73 dataset has the highest level of isoform complexity. It contains more than 35% of the genes with two or more isoforms and the maximum number of isoforms in a gene is 20. In contrast, the majority of the genes (93%) in the dataset based on another strain of maize, Mo17, have only one isoform (no alternative splicing). The *A. thaliana* No0 dataset has no multiple-isoform genes as the No0 reference transcriptome does not include isoform information, and hence each gene is represented by a single transcript. Although these datasets may not represent the actual distribution of isoforms in these plant genomes, they are useful for testing the impact of isoforms in transcriptome assembly. In addition to incorporating isoforms, simulated benchmark datasets can be generated incorporating different levels of ploidy. More details of these simulation protocols are found in (7, 60).

PERFORMANCE COMPARISON AMONG TRANSCRIPTOME ASSEMBLERS

In this section, we demonstrate how the performance among transcriptome assemblers can be compared using the simulated benchmark datasets prepared in the previous section. Before running transcriptome assemblers, the simulated reads need to be preprocessed. We used the following settings:

- Quality filtering using Erne-filter 2.0 (61) with minimum mean Phred quality 20, ‘ultra-sensitive’ flag, and paired-end mode
- Read normalization using Khmer (62) with *k*-mer size of 32, an expected coverage of 50X, and paired-end mode

We compared the transcriptome assembly performance among three genome-guided (Cufflinks, StringTie, and Scallop), four *de novo* (IDBA-Tran, SOAPdenovo-Trans, Trinity, and rnaSPAdes), and three ensemble (EvidentialGene, Concatenation, and the consensus approach) assemblers. For this analysis, performance metrics were calculated at the level of protein sequences. The longest ORF was identified by ORFfinder from each contig, and the translated ORF sequences were compared against the translated benchmark transcriptome. A contig was considered correctly assembled only if its coded protein sequence was identical to one of the translated benchmark transcripts.

Genome-guided approach

We used HISAT2 for aligning simulated short reads to their reference genomes before using the three genome-guided assemblers. To examine the effect of the reference genome for *A. thaliana* and *Z. mays* in addition to aligning each read set against the reference genome from which the simulated RNA-seq was performed, it was also aligned against the genome of the different strain of the same species. These results are shown as “same reference” and “different reference” in Table 1, respectively. The simplest test is the one with the *A. thaliana* No0 dataset, which does not include multiple isoforms for any gene, assembled using the same No0 genome as the reference. Surprisingly, no genome-guided methods had an accuracy greater than 65%, with more than 25% of assembled contigs being incorrect (C/I ≤ 3). With more realistic isoform complexity, no method achieved an accuracy greater than 50%. With both of the *Z. may* datasets, more than half of assembled contigs were incorrect (C/I ≤ 1). When these genome-guided methods were used with different references, although they are still from the same species, assembly performance deteriorated significantly: < 22% for the *A. thaliana* datasets and < 10% for the *Z. mays* datasets. For both *Z. mays* datasets, only 1 in 6 contigs were found to be correctly assembled (C/I ≤ 0.2). It is notable that both *Z. mays* datasets generated lower quality assemblies compared to other datasets. A relatively lower quality of the *Z. mays* genomes may have contributed to the significantly poor performance of these assemblers with these datasets.

The overlap between correctly and incorrectly assembled contigs among the assemblies generated by the three genome-guided assemblers is illustrated in Figure 1 (63). While each assembler generated a unique set of correct as well as incorrect contigs, ~70% or more of correctly assembled contigs were generated by all three assemblers. The exception was for the *Z. mays* B73 (37%) dataset. In contrast, the majority of incorrectly assembled contigs (62-87%) were uniquely generated by each assembler, and a very small number of contigs were incorrectly assembled by all three methods.

De novo approach

Each of the four *de novo* assemblers was run with the default parameters. As shown in Table 1, for all benchmark datasets, all *de novo* assemblers generated more contigs compared to genome-guided methods. However, their low accuracy (< 0.31) and C/I scores (< 0.63) indicate that the majority of contigs were incorrectly assembled. Trinity, followed by rnaSPAdes, performed better than other *de novo* assemblers for all datasets. Interestingly, while the *de novo* assemblers did not perform better than the genome-guided methods used with the same references, the performance of the *de novo* assemblers was better than the genome-guided methods when they were used with different references. Similar to the genome-guided assembly, the largest numbers (≥ 30% except 17% for the *Z. mays* B73 dataset) of the correctly assembled contigs were found in the group of contigs shared by all four *de novo* assemblers (Figure 2). Incorrectly assembled contigs were also found to be most likely assembled by individual assemblers uniquely and not shared with other assemblies.

Combining *de novo* assemblies generated using different *k*-mers

Since the optimum *k*-mer size for each transcript assembly varies, different sets of correctly assembled contigs are expected even when the same *de novo* method is used with different *k*-mer sizes. Therefore, by combining the results from multiple *k*-mers, we expect to find more contigs correctly assembled by

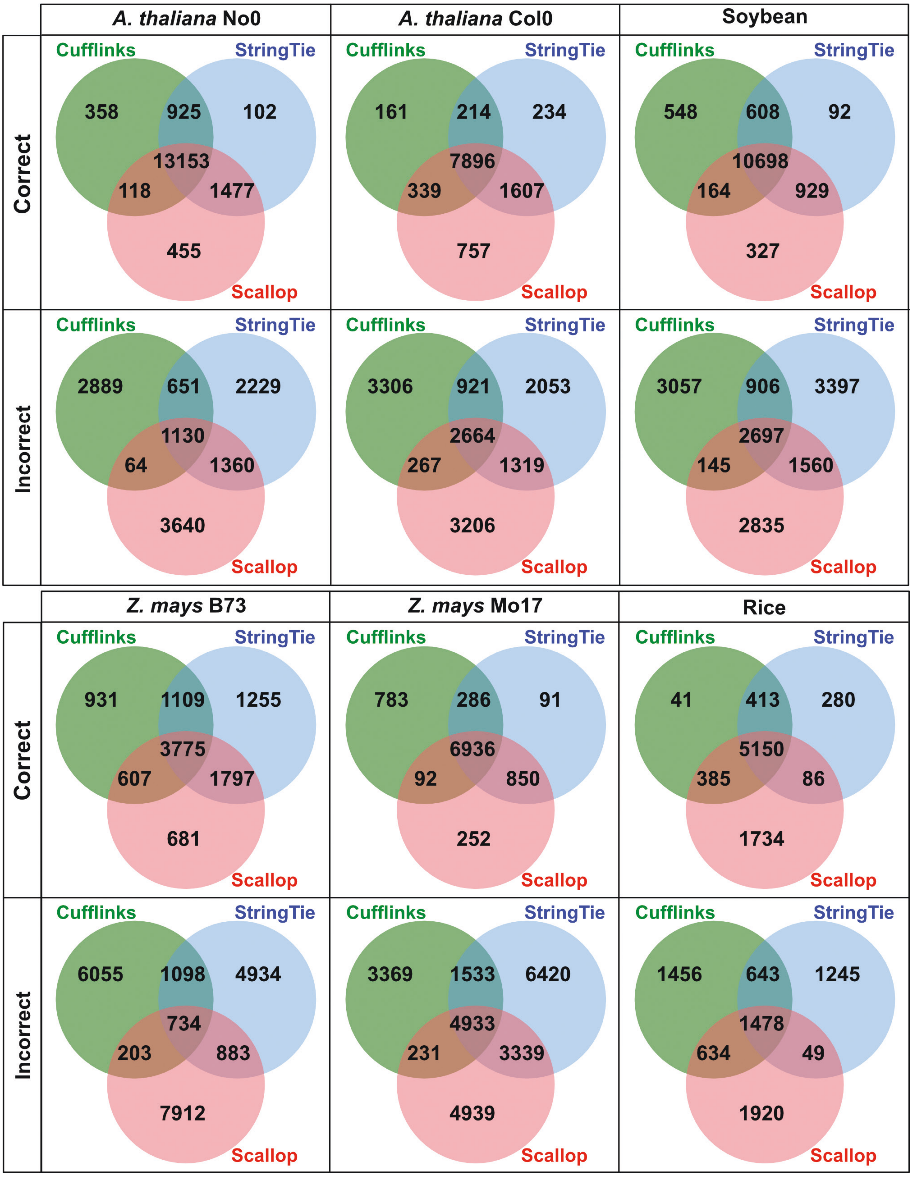

Figure 1. Numbers of correctly and incorrectly assembled contigs shared among the three genome-guided assemblers. Each genome-guided assembly was performed using the reference and the RNA-seq data from the same genome. Venn diagrams were generated using jvenn (63).

de novo assemblers. To illustrate this idea, we used multiple *k*-mer sizes for each of the four *de novo* assemblers and generated a “pooled assembly” by combining their results (the union set). The four pooled assemblies are compared in Figure 3. Compared to Figure 2, the proportion of correctly

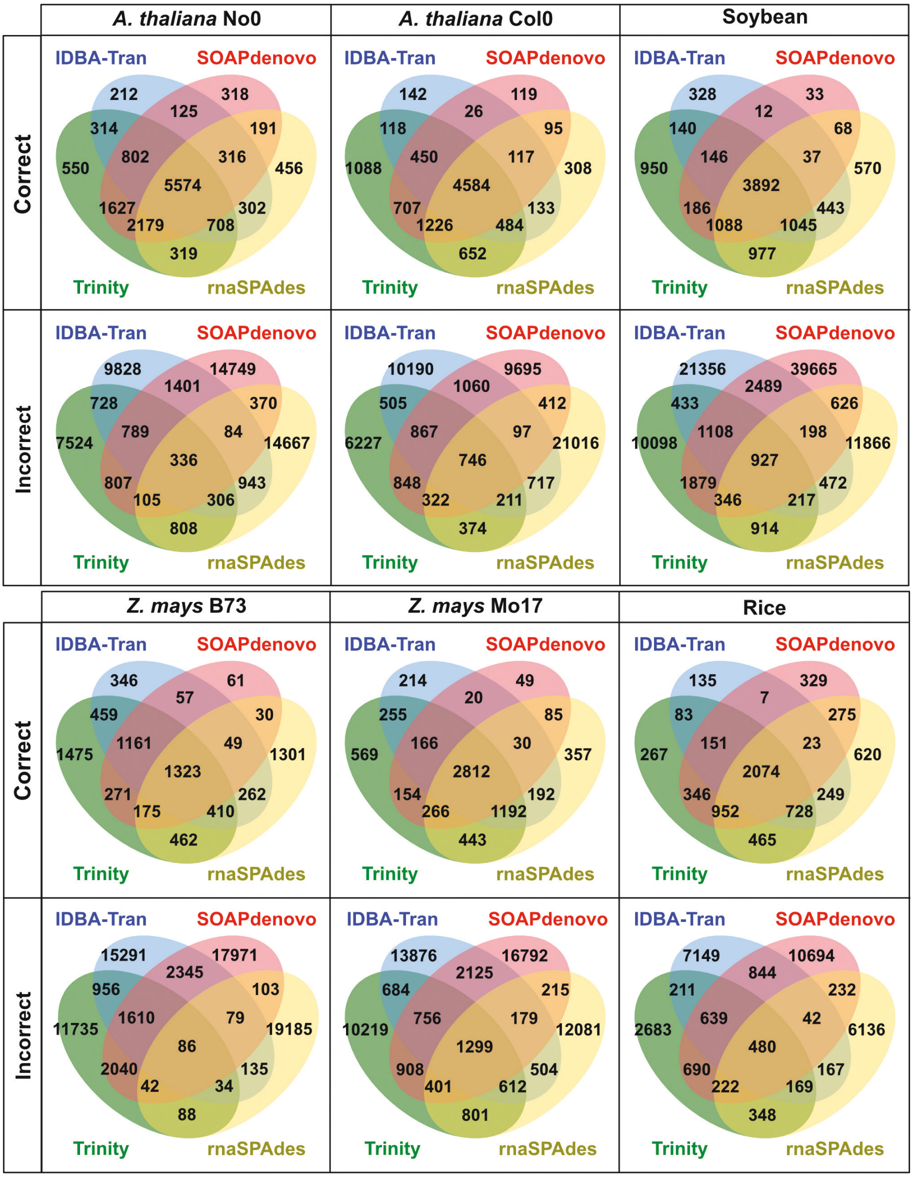

Figure 2. Numbers of correctly and incorrectly assembled contigs shared among the four *de novo* assemblers used with the default settings. Venn diagrams were generated using jvenn (63).

assembled contigs shared by all four pooled assemblies increased significantly (≥ 55% except 36% for the *Z. mays* B73 dataset). Furthermore, only a very small proportion (≤ 10%) of the incorrectly assembled contigs were shared by two or more pooled assemblies.

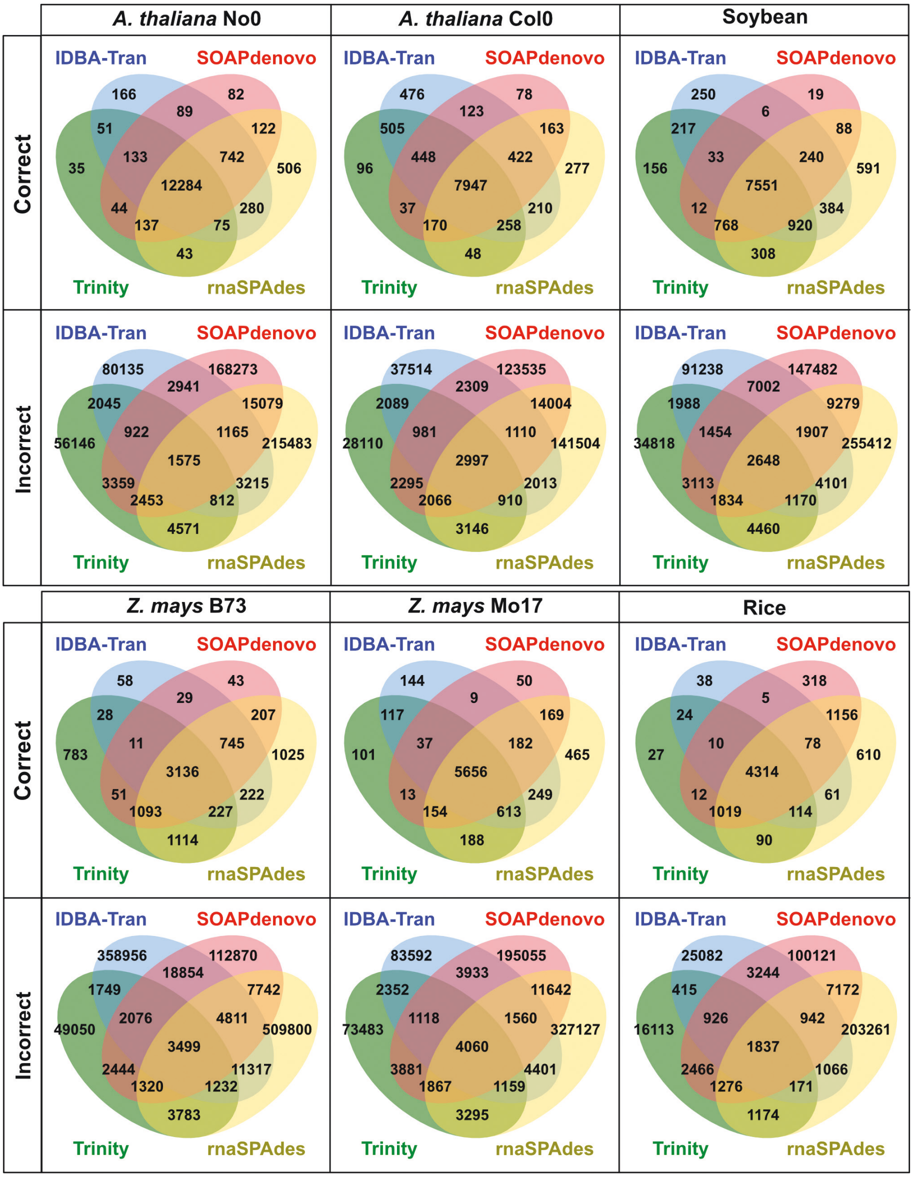

Figure 3. Numbers of correctly and incorrectly assembled contigs shared among the four pooled *de novo* assemblies. The following *k*-mers are used: for IDBA-Trans, *k*=20~60 with increment of 10; for SOAPdenovo-Trans and rnaSPAdes, *k*=19~71 with increment of 4; and for Trinity, *k*=15~31 with increment of 4. Venn diagrams were generated using jvenn (63).

Analysis of *k*-mers used in assembled contigs

The *k*-mers of a contig that are not present in the benchmark transcriptome are considered to be false *k*-mers. When false *k*-mers are used for the de Bruijn graph construction in *de novo* assemblers, it generates incorrect contigs. To understand why the *Z. mays* B73 dataset generated poor assemblies regardless of the methods, we analyzed *k*-mers found in contigs assembled by the four *de novo* assemblers (Table 2). Compared to the assemblies generated from the Rice dataset, those generated from the *Z. mays* B73 dataset were represented by significantly lower numbers of true *k*-mers (the *k*-mers that are found in the benchmark transcriptome). In any of the *Z. mays* B73 assemblies generated by the four methods, fewer than 50% of assembled contigs contained *k*-mers 90% or more of which were true (those found in the benchmark data). It appears that large numbers of false *k*-mers were included in the de Bruijn graph construction for the maize transcriptomes leading to the poor *de novo* assembly performance for this dataset.

Ensemble approach

We finally compared the assembly performance of all individual methods with the three ensemble approaches, EvidentialGene, Concatenation, and the aforementioned consensus approach (60). Both EvidentialGene and Concatenation over-assembled and accumulated incorrectly assembled contigs as shown in their significantly higher Recall compared to Precision (Figure 4). It indicates that these methods recover many transcripts correctly at the expense of having a disproportionally large number of incorrectly assembled contigs. The F-measure (the combined score of Recall and Precision) scored lower for EvidentialGene and Concatenation compared to individual *de novo* assemblies for most of the datasets. It should be noted, however, that although many contigs retained by these ensemble methods are identified to be incorrect, they are still reported as highly similar (> 98%) to the benchmark transcripts (60). The consensus approach consistently performed better than all the *de novo* assemblers for all datasets and achieved a performance similar to the genome-guided assemblers without requiring good reference genomes.

TABLE 2 **The *k*-mer analysis for the *de novo* assemblies using the *Z. mays* B73 and Rice datasets**[a]

	IDBA-Tran	SOAPdenovo-Trans	Trinity	rnaSPAdes
[Rice]				
% true *k*-mers[b]	96.09	97.56	98.89	55.27
% contigs with >90% true *k*-mers[c]	95.96	92.86	97.68	58.91
[*Z. mays* B73]				
% true *k*-mers[b]	26.18	48.7	53.57	27.72
% contigs with >90% true *k*-mers[c]	15.23	47.6	38.7	21.81

[a] All results are based on pooled assembly.

[b] The proportion (%) of the *k*-mers (*k*=31) found in the contigs that were also found in the benchmark transcripts (true *k*-mers).

[c] The proportion (%) of the contigs where 90% or more of the *k*-mer found were true *k*-mers.

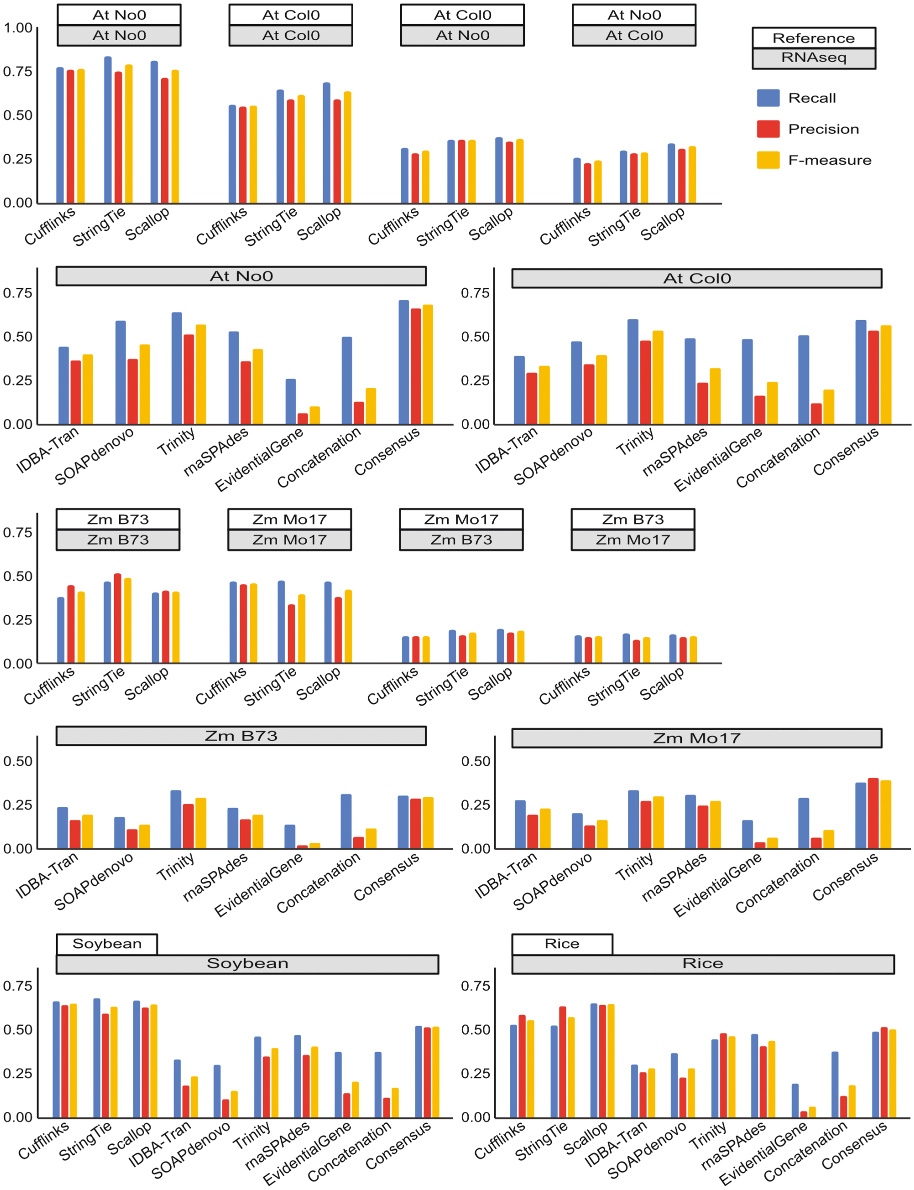

Figure 4. Comparison of transcriptome assembly performance among different methods. The simulated RNA-seq data (gray boxes) and the reference genome (for genome-guided methods; white boxes) used are shown at the top of each bar chart. The default *k*-mers were used for the *de novo* methods. At: *A. thaliana*, Zm: *Z. mays*.

CONCLUSION

In this chapter, we show how availability of a high-quality reference genome affects the transcriptome assembly performance by the genome-guided approach. When such reference genomes are not available, as in the case for non-model organisms, *de novo* assemblers can achieve good performance. However, challenges due to isoform complexity, polyploidy, and optimal parameter selection remain. The most significant parameter in de Bruijn graph-based *de novo* assembly methods is the *k*-mer size. Ensemble approaches take advantage of pooling the *de novo* assemblies based on different methods as well as multiple *k*-mers to increase the number of correct contigs without accumulating incorrect contigs. Among the three ensemble methods compared here, the consensus approach performed the best for all benchmark plant datasets tested. Finally, we note the importance of the simulated benchmarked datasets for assessment and improvement of the performance of transcriptome assembly.

Acknowledgement: We thank Dr. Jitender Deogun (University of Nebraska-Lincoln) for comments during the study. We also thank Kushagra Kapil (University of Nebraska-Lincoln) for generating the database and visualization of assembler performance. Part of this study was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative. This work has been supported by National Science Foundation EPSCoR RII Track-1: Center for Root and Rhizobiome Innovation (CRRI) Award OIA-1557417 to ENM.

Conflict of interest: The authors declare no potential conflicts of interest with respect to research, authorship, and/or publication of this article.

Copyright and Permission Statement: The authors confirm that the materials included in this chapter do not violate copyright laws. Where relevant, appropriate permissions have been obtained from the original copyright holder(s), and all original sources have been appropriately acknowledged or referenced.

REFERENCES

1. Ozsolak F, Milos PM. RNA sequencing: advances, challenges and opportunities. Nat Rev Genet. 2011;12(2):87–98. https://doi.org/10.1038/nrg2934
2. Martin JA, Wang Z. Next-generation transcriptome assembly. Nat Rev Genet. 2011;12(10):671–82. https://doi.org/10.1038/nrg3068
3. Stark R, Grzelak M, Hadfield J. RNA sequencing: the teenage years. Nat Rev Genet. 2019;20(11):631–56. https://doi.org/10.1038/s41576-019-0150-2
4. Vijay N, Poelstra JW, Künstner A, Wolf JB. Challenges and strategies in transcriptome assembly and differential gene expression quantification. A comprehensive *in silico* assessment of RNA-seq experiments. Mol Ecol. 2013;22(3):620–34. https://doi.org/10.1111/mec.12014
5. Hsieh PH, Oyang YJ, Chen CY. Effect of *de novo* transcriptome assembly on transcript quantification. Sci Rep. 2019;9(1):8304. https://doi.org/10.1038/s41598-019-44499-3
6. Gutierrez-Gonzalez JJ, Garvin DF. De novo transcriptome assembly in polyploid species. Methods Mol Biol. 2017;1536:209–21. https://doi.org/10.1007/978-1-4939-6682-0_15

7. Voshall A, Moriyama EN. Next-generation transcriptome assembly and analysis: Impact of ploidy. Methods. 2020;176:14–24. https://doi.org/10.1016/j.ymeth.2019.06.001
8. Voshall A, Moriyama EN. Chapter 2, Next-generation transcriptome assembly: strategies and performance analysis. In: Adburakhmonov IY, editor. Bioinformatics in the Era of Post Genomics and Big Data. London, UK: IntechOpen; 2018. https://doi.org/10.5772/intechopen.73497
9. Gongora-Castillo E, Buell CR. Bioinformatics challenges in *de novo* transcriptome assembly using short read sequences in the absence of a reference genome sequence. Nat Prod Rep. 2013;30(4):490–500. https://doi.org/10.1039/c3np20099j
10. Huang X, Chen XG, Armbruster PA. Comparative performance of transcriptome assembly methods for non-model organisms. BMC Genomics. 2016;17:523. https://doi.org/10.1186/s12864-016-2923-8
11. Mahmood K, Orabi J, Kristensen PS, Sarup P, Jorgensen LN, Jahoor A. De novo transcriptome assembly, functional annotation, and expression profiling of rye (*Secale cereale* L.) hybrids inoculated with ergot (*Claviceps purpurea*). Sci Rep. 2020;10(1):13475. https://doi.org/10.1038/s41598-020-70406-2
12. Florea LD, Salzberg SL. Genome-guided transcriptome assembly in the age of next-generation sequencing. IEEE/ACM Trans Comput Biol Bioinform. 2013;10(5):1234–40. https://doi.org/10.1109/TCBB.2013.140
13. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol. 2013;14(4):R36. https://doi.org/10.1186/gb-2013-14-4-r36
14. Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. Nat Biotechnol. 2019;37(8):907–15. https://doi.org/10.1038/s41587-019-0201-4
15. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013;29(1):15–21. https://doi.org/10.1093/bioinformatics/bts635
16. Maretty L, Sibbesen JA, Krogh A. Bayesian transcriptome assembly. Genome Biol. 2014;15(10):501. https://doi.org/10.1186/s13059-014-0501-4
17. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol. 2010;28(5):511–5. https://doi.org/10.1038/nbt.1621
18. Pertea M, Pertea GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. Nat Biotechnol. 2015;33(3):290–5. https://doi.org/10.1038/nbt.3122
19. Shao M, Kingsford C. Accurate assembly of transcripts through phase-preserving graph decomposition. Nat Biotechnol. 2017;35(12):1167–9. https://doi.org/10.1038/nbt.4020
20. Oshlack A, Robinson MD, Young MD. From RNA-seq reads to differential expression results. Genome Biol. 2010;11(12):220. https://doi.org/10.1186/gb-2010-11-12-220
21. Compeau PEC, Pevzner PA, Tesler G. How to apply de Bruijn graphs to genome assembly. Nat Biotechnol. 2011;29(11):987–91. https://doi.org/10.1038/nbt.2023
22. 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. Nat Biotechnol. 2011;29(7):644–52. https://doi.org/10.1038/nbt.1883
23. Haas BJ, Papanicolaou A, Yassour M, Grabherr M, Blood PD, Bowden J, et al. *De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nat Protoc. 2013;8(8):1494–512. https://doi.org/10.1038/nprot.2013.084
24. 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(13):i326–34. https://doi.org/10.1093/bioinformatics/btt219
25. Xie Y, Wu G, Tang J, Luo R, Patterson J, Liu S, et al. SOAPdenovo-Trans: *de novo* transcriptome assembly with short RNA-Seq reads. Bioinformatics. 2014;30(12):1660–6. https://doi.org/10.1093/bioinformatics/btu077
26. Bushmanova E, Antipov D, Lapidus A, Prjibelski AD. rnaSPAdes: a *de novo* transcriptome assembler and its application to RNA-Seq data. Gigascience. 2019;8(9). https://doi.org/10.1093/gigascience/giz100

27. Durai DA, Schulz MH. Informed kmer selection for *de novo* transcriptome assembly. Bioinformatics. 2016;32(11):1670–7. https://doi.org/10.1093/bioinformatics/btw217
28. Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, et al. SOAPdenovo2: an empirically improved memory-efficient short-read *de novo* assembler. Gigascience. 2012;1(1):18. https://doi.org/10.1186/2047-217X-1-18
29. Bankevich A, Nurk S, Antipov D, Gurevich AA, Dvorkin M, Kulikov AS, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. J Comput Biol. 2012;19(5):455–77. https://doi.org/10.1089/cmb.2012.0021
30. Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S. TransRate: reference-free quality assessment of de novo transcriptome assemblies. Genome Res. 2016;26(8):1134–44. https://doi.org/10.1101/gr.196469.115
31. Gilbert DG. Genes of the pig, *Sus scrofa*, reconstructed with EvidentialGene. PeerJ. 2019;7:e6374. https://doi.org/10.7717/peerj.6374
32. Cerveau N, Jackson DJ. Combining independent *de novo* assemblies optimizes the coding transcriptome for nonconventional model eukaryotic organisms. BMC Bioinformatics. 2016;17(1):525. https://doi.org/10.1186/s12859-016-1406-x
33. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28(23):3150–2. https://doi.org/10.1093/bioinformatics/bts565
34. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10. https://doi.org/10.1016/S0022-2836(05)80360-2
35. Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402. https://doi.org/10.1093/nar/25.17.3389
36. Yu T, Mu Z, Fang Z, Liu X, Gao X, Liu J. TransBorrow: genome-guided transcriptome assembly by borrowing assemblies from different assemblers. Genome Res. 2020;30(8):1181–90. https://doi.org/10.1101/gr.257766.119
37. Bushmanova E, Antipov D, Lapidus A, Suvorov V, Prjibelski AD. rnaQUAST: a quality assessment tool for *de novo* transcriptome assemblies. Bioinformatics. 2016;32(14):2210–2. https://doi.org/10.1093/bioinformatics/btw218
38. O'Neil ST, Emrich SJ. Assessing *de novo* transcriptome assembly metrics for consistency and utility. BMC Genomics. 2013;14:465. https://doi.org/10.1186/1471-2164-14-465
39. Li B, Fillmore N, Bai Y, Collins M, Thomson JA, Stewart R, et al. Evaluation of *de novo* transcriptome assemblies from RNA-Seq data. Genome Biol. 2014;15(12):553. https://doi.org/10.1186/s13059-014-0553-5
40. Seppey M, Manni M, Zdobnov EM. BUSCO: assessing genome assembly and annotation completeness. Methods Mol Biol. 2019;1962:227–45. https://doi.org/10.1007/978-1-4939-9173-0_14
41. Kriventseva EV, Kuznetsov D, Tegenfeldt F, Manni M, Dias R, Simao FA, et al. OrthoDB v10: sampling the diversity of animal, plant, fungal, protist, bacterial and viral genomes for evolutionary and functional annotations of orthologs. Nucleic Acids Res. 2019;47(D1):D807-D11. https://doi.org/10.1093/nar/gky1053
42. Holzer M, Marz M. *De novo* transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers. Gigascience. 2019;8(5). https://doi.org/10.1093/gigascience/giz039
43. Zhao M, Liu D, Qu H. Systematic review of next-generation sequencing simulators: computational tools, features and perspectives. Brief Funct Genomics. 2017;16(3):121–8. https://doi.org/10.1093/bfgp/elw012
44. Li B, Dewey CN. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. BMC Bioinformatics. 2011;12:323. https://doi.org/10.1186/1471-2105-12-323
45. Benidt S, Nettleton D. SimSeq: a nonparametric approach to simulation of RNA-sequence datasets. Bioinformatics. 2015;31(13):2131–40. https://doi.org/10.1093/bioinformatics/btv124
46. Assefa AT, Vandesompele J, Thas O. SPsimSeq: semi-parametric simulation of bulk and single-cell RNA-sequencing data. Bioinformatics. 2020;36(10):3276–8. https://doi.org/10.1093/bioinformatics/btaa105
47. Gerard D. Data-based RNA-seq simulations by binomial thinning. BMC Bioinformatics. 2020;21(1):206. https://doi.org/10.1186/s12859-020-3450-9

48. Love MI, Hogenesch JB, Irizarry RA. Modeling of RNA-seq fragment sequence bias reduces systematic errors in transcript abundance estimation. Nat Biotechnol. 2016;34(12):1287–91. https://doi.org/10.1038/nbt.3682
49. Frazee AC, Jaffe AE, Langmead B, Leek JT. Polyester: simulating RNA-seq datasets with differential transcript expression. Bioinformatics. 2015;31(17):2778–84. https://doi.org/10.1093/bioinformatics/btv272
50. Griebel T, Zacher B, Ribeca P, Raineri E, Lacroix V, Guigo R, et al. Modelling and simulating generic RNA-Seq experiments with the flux simulator. Nucleic Acids Res. 2012;40(20):10073–83. https://doi.org/10.1093/nar/gks666
51. Ono Y, Asai K, Hamada M. PBSIM: PacBio reads simulator--toward accurate genome assembly. Bioinformatics. 2013;29(1):119–21. https://doi.org/10.1093/bioinformatics/bts649
52. Lau B, Mohiyuddin M, Mu JC, Fang LT, Bani Asadi N, Dallett C, et al. LongISLND: *in silico* sequencing of lengthy and noisy datatypes. Bioinformatics. 2016;32(24):3829–32. https://doi.org/10.1093/bioinformatics/btw602
53. Wick RR. Badread: simulation of error-prone long reads. Journal of Open Source Software. 2019;4(36):2. https://doi.org/10.21105/joss.01316
54. Hafezqorani S, Yang C, Lo T, Nip KM, Warren RL, Birol I. Trans-NanoSim characterizes and simulates nanopore RNA-sequencing data. Gigascience. 2020;9(6). https://doi.org/10.1093/gigascience/giaa061
55. Gan X, Stegle O, Behr J, Steffen JG, Drewe P, Hildebrand KL, et al. Multiple reference genomes and transcriptomes for *Arabidopsis thaliana*. Nature. 2011;477(7365):419–23. https://doi.org/10.1038/nature10414
56. The 1001 Genomes Consortium. 1,135 genomes reveal the global pattern of polymorphism in *Arabidopsis thaliana*. Cell. 2016;166(2):481–91. https://doi.org/10.1016/j.cell.2016.05.063
57. Reiser L, Subramaniam S, Li D, Huala E. Using the *Arabidopsis* Information Resource (TAIR) to find information about *Arabidopsis* genes. Curr Protoc Bioinformatics. 2017;60:1 11 1–1 45. https://doi.org/10.1002/cpbi.36
58. Zhang R, Calixto CP, Tzioutziou NA, James AB, Simpson CG, Guo W, et al. AtRTD - a comprehensive reference transcript dataset resource for accurate quantification of transcript-specific expression in *Arabidopsis thaliana*. New Phytol. 2015;208(1):96–101. https://doi.org/10.1111/nph.13545
59. Wheeler DL, Church DM, Federhen S, Lash AE, Madden TL, Pontius JU, et al. Database resources of the National Center for Biotechnology. Nucleic Acids Res. 2003;31(1):28–33. https://doi.org/10.1093/nar/gkg033
60. Voshall A, Behera S, Li X, Yu X-H, Kapil K, Deogun JS, et al. A consensus-based ensemble approach to improve *de novo* transcriptome assembly. bioRxiv. 2020;2020.06.08.139964. https://doi.org/10.1101/2020.06.08.139964
61. 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(12):e85024. https://doi.org/10.1371/journal.pone.0085024
62. Crusoe MR, Alameldin HF, Awad S, Boucher E, Caldwell A, Cartwright R, et al. The khmer software package: enabling efficient nucleotide sequence analysis. F1000Res. 2015;4:900. https://doi.org/10.12688/f1000research.6924.1
63. Bardou P, Mariette J, Escudie F, Djemiel C, Klopp C. jvenn: an interactive Venn diagram viewer. BMC Bioinformatics. 2014;15:293. https://doi.org/10.1186/1471-2105-15-293