Summer 7-6-2022

ConSemblEX: A Consensus-Based Transcriptome Assembly Approach that Extends ConSemble and Improves Transcriptome Assembly

Richard Mwaba
University of Nebraska-Lincoln, rmwaba2@huskers.unl.edu

CONSEMBLEX: A CONSENSUS-BASED APPROACH THAT EXTENDS CONSEMBLE AND IMPROVES TRANSCRIPTOME ASSEMBLY

by

Richard Mwaba

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Jitender Deogun and Professor Etsuko Moriyama

Lincoln, Nebraska

July, 2022

CONSEMBLEX: A CONSENSUS-BASED APPROACH THAT EXTENDS CONSEMBLE AND IMPROVES TRANSCRIPTOME ASSEMBLY

Richard Mwaba, M.S.

University of Nebraska, 2022

Adviser: Jitender Deogun

An accurate transcriptome is essential to understanding biological systems enabling omics analyses such as gene expression, gene discovery, and gene-regulatory network construction. However, assembling an accurate transcriptome is challenging, especially for organisms without adequate reference genomes or transcriptomes. While several methods for transcriptome assembly with different approaches exist, it is still difficult to establish the most accurate methods. This thesis explores the different transcriptome assembly methods and compares their performances using simulated benchmark transcriptomes with varying complexity. We also introduce ConSemblEX to improve a consensus-based ensemble transcriptome assembler, ConSemble, in three main areas: we provide the ability to use any number of assemblers, provide a variety of consensus assembly outputs, and provide information about the effect of each assembler in the final assembly. Using five assembly methods both in the *de novo* and genome-guided approaches, we showed how ConSemblEX can be used to explore various strategies for consensus assembly, such as ConSemblEX-4+, to find the optimum assembly. Compared to the original ConSemble, ConSemblEX improved the *de novo* assembly performance, increasing the *precision* by 14% and F_1 by 5%, and significantly reducing the *FP* by 49%. In the genome-guided assembly, ConSemblEX had an identical performance to the original ConSemble. We showed that ConSemblEX provides tools to explore how different methods perform and behave depending

on the datasets. With the ConSemblEX-select assembly, we further demonstrated that we can improve consensus-based assembly more by choosing optimum overlap sets among different methods. Such information provides the foundation to develop machine learning algorithms in the future to further improve transcriptome assembly performance.

DEDICATION

To Richard Snr. and Aphia.

ACKNOWLEDGMENTS

I want to express my appreciation and sincere gratitude to Professor Jitender Deogun and Professor Etsuko Moriyama for their incredible guidance during the course of my experiments and writing of this thesis project.

I want to thank Professor Juan Cui for being on my thesis committee and providing further guidance. I also acknowledge my colleagues, Sojan Shrestha, Bailee Egan, and Simreen Kaur, for their timely feedback, shared ideas, and overall support throughout my program. A special thank you goes to my partner Charity Mwansa for her amazing support, mum and dad for all their sacrifices, and my siblings and friends for their love.

Most importantly, I would like to thank Dr. Charles Wood, Dr. Catherine Chunda and the AMTRIP fellowship for believing in me and giving me a chance to pursue my dreams through the financial support for my program.

Above all, I thank God for all His goodness.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

It has long been the quest of science to understand living things, the relations that exist between them, and how they coexist. This is not limited to a higher level of understanding, but also accounts for the molecular level, which is their genetic makeup located in the cells. Many unique characteristics of organisms retain a trace to a set of genes responsible for their development. Therefore, studying those genes would help scientists have a deeper understanding of an organism as a whole. One of the critical elements to this probe is the deoxyribonucleic acid (DNA), which is the molecule that contains hereditary information in all living things. Variations in DNA can affect the physical appearance of an organism, how it develops, functions, or responds to its environment. DNA stores information as a code comprising chemical bases, adenine (A), cytosine (C), thymine (T), and guanine (G) [3]. The order of these bases (a nucleotide sequence) determines the unique information available to develop and maintain an organism. Specific sequences form genes whose complete set in an organism is the genome [3].

Cells utilize the genetic information to produce proteins needed in a multistep process called gene expression. The process involves two primary steps, transcrip-

tion and translation (Fig. 1.1). In transcription, the information in DNA is copied into a new ribonucleic acid (RNA) molecule, messenger RNA (mRNA, also called a transcript), making it available for protein synthesis. A collection of all the available transcripts resulting from transcription is called a transcriptome [3]. Translation decodes the information contained in the transcript into a particular sequence of amino acids to synthesize a protein [3].

Depending on gene expression patterns, transcriptomes vary in the composition of transcripts among different cell types and stages of development within an organism. An accurate picture of gene expression can be gathered by sequencing all the transcripts from target cells using high-throughput next-generation sequencing (NGS) platforms such as Illumina. NGS technologies produce hundreds to millions of short reads. These short reads need to be assembled into a transcriptome using various assembly methods. Achieving an accurate transcriptome across all transcripts is a hard task. In this regard, different transcriptome assembly methods have been developed to address the challenges.

1.2 Motivation

Transcriptome assembly is central to understanding organisms at the molecular level, enabling the discovery of the expression levels and their relationships. However, the computational assembly of accurate transcriptomes remains a considerable challenge, especially for non-model organisms. Variances such as the quality of RNA-sequencing reads, presence of isoforms (alternatively spliced transcripts generated from the same gene), polyploidy, and other biological properties could lead to incorrect and incomplete assemblies [2]. Availability of a good reference transcriptome or genome can also affect the quality of the assembly in the case of genome-guided methods.

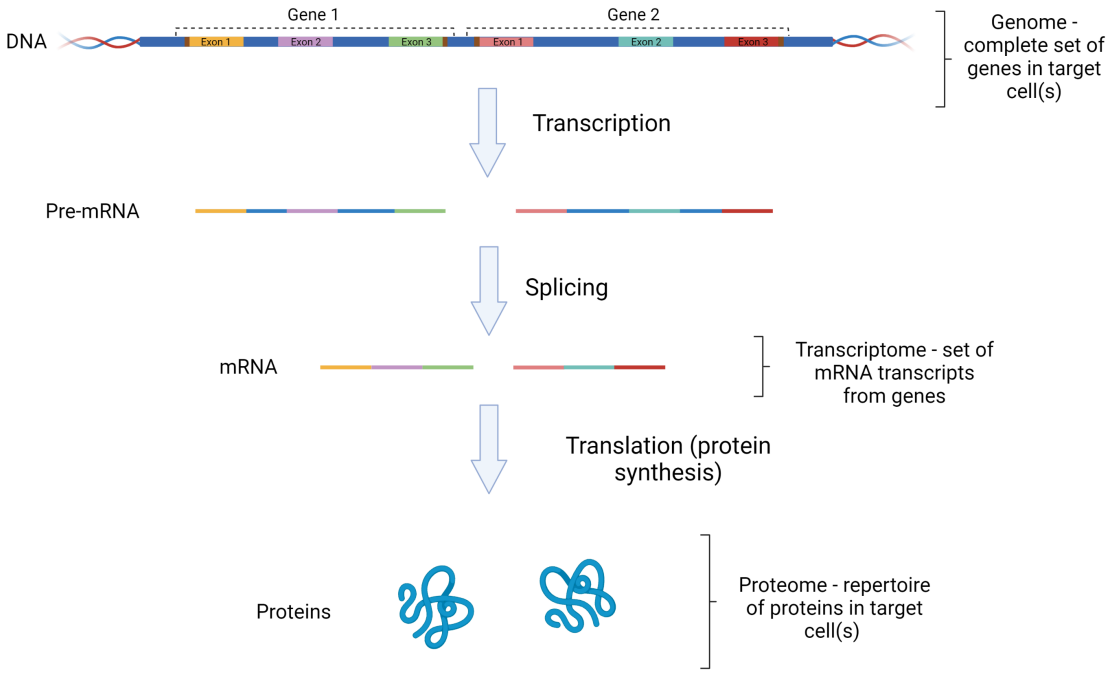


Figure 1.1: Gene expression overview. Genes have exons - the transcribed regions that mainly include protein coding regions, and introns - the non-transcribed, non-coding regions. A complete set of all genes in target cells, tissue or an entire organism forms the genome. During expression, the genetic information in DNA is transcribed into precursor mRNAs (pre-mRNA). The splicing process removes introns (blue regions between exons) from the pre-mRNA producing a mature mRNA. The set of all available mature mRNA molecules from a genome in a cell is referred to as a transcriptome. The mRNA molecules code the particular amino acid sequences and are translated into proteins. A proteome of a cell is a repertoire of proteins coded on the genome of the organism and translated from a transcriptome of a cell.

Consensus-based ensemble assemblers such as ConSemble [46] have been developed to address many of the shortfalls of individual assemblers. ConSemble merges results from four specific assemblers and finds consensus contigs as representative contigs. Although it has been reported to perform better than individual assemblers, further improvement of the assembly performance is possible. Three areas have been identified for further improvement.

Firstly, individual assemblers used can be expanded from those initially provided.

In ConSemble, SOAPDenovo-Trans [49], rnaSPAdes [4], Trinity [18], and IDBA-tran [37] are used for *de novo* assembly while Cufflinks [45], Bayesembler [33], Scallop [40], and StringTie2 [27] are used in genome-guided assembly. With the various existing assemblers and ever-evolving approaches, it would be ideal to allow for inclusion of more assemblers.

Secondly, the main output of ConSemble is a 3+ consensus (comprising all three-way overlaps and the four-way overlap). The four-way and a 2+ consensus (composed of all two-way overlaps and 3+) were also explored. Inasmuch as these available consensus assemblies are the most compelling in the study, there is a possibility of other combinations producing as good as or even better results. Further, the quality of the 3+ is likely to be affected as the number of assemblers increases.

Lastly, generating only the final ConSemble result (ConSemble3+), does not provide the information regarding the effect of each assembler on the result. For example, among the assemblies produced by three methods, one method could only provide 5% correctly assembled contigs, significantly affecting the final result. In such a case, an overlap with other methods would be a better option.

1.3 Main Contributions

The thesis addresses the three points described in Section 1.2 and establishes a foundation for a more generalized and improved ConSemble approach. We introduce ConSemblEX, which expands on the current ConSemble approach by enabling the use of more than four assemblers. To demonstrate this capability, a recently published *de novo* method, BayesDenovo [41], is incorporated. This expansion provides flexibility in choosing assemblers and an opportunity to integrate newer assemblers in the future. As the number of assemblers increases, possible combinations also

increase, prompting the need to analyze many more combinations. ConSemblEX allows all combinations to be easily analyzed facilitating the examination of the effect of increasing the number of assemblers on the transcriptome assembly accuracy and selection of the optimum strategy for the consensus assembly. Lastly, the effect of each assembler used on the final consensus assembly is assessed to ensure the best assemblers are picked.

Collectively, ConSemblEX provides a tool to produce more accurate and complete transcriptome assemblies.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 provides the background on transcriptome assembly and related works using different approaches. Chapter 3 overviews the methods used in the original ConSemble and introduces the modifications, ConSemblEX, along with the approach used to evaluate the assembled transcriptomes. Chapter 4 presents and discusses results from two separate experiments on three simulated benchmark datasets, and compares the performance of ConSemblEX to the existing state of the art transcriptome assembly methods. Finally, Chapter 5 presents the conclusion of this thesis and the future work.

Chapter 2

Background and Related Work

In multicellular organisms, every cell contains the same set of genes (genome). However, different cells show different patterns of gene expression. Thus, by collecting and comparing transcriptomes (the entire set of mRNA transcripts) of different types of cells or tissues, researchers can understand what constitutes a specific cell type and how changes in transcriptional activity may reflect on the functions and development of the cells and organisms [34]. To collect a transcriptome, we need to gain access to ideally all the transcripts of specific cells or tissues at a specific time. This is accomplished mainly by employing RNA-sequencing (RNA-seq) [47], using NGS methods. RNA-seq experiments typically comprise isolating mRNAs from cell or tissue samples, fragmenting them into shorter segments, converting them to cDNAs, preparing the sequencing library, and sequencing it to obtain millions of short fragments called reads (Fig. 2.1)]. The reads are generally 30-500 bp (base pairs) based on the sequencing technology applied. Illumina HiSeq 2500[1], one of the most used NGS platforms, for example, produces reads up to 250 bp long. Following sequencing, the obtained reads are either aligned to a reference genome in genome-guided assemblers or assembled *de novo* to produce a genome-scale transcription map consisting of the transcriptional structure and expression level for each gene [28]. This is the basis for transcriptome

[1] `https://www.illumina.com/systems/sequencing-platforms/hiseq-2500.html`

assembly.

2.1 Transcriptome Assembly Approaches

Transcriptome assembly enables the study of genetic variations in organisms at the transcript level. Genome-guided assemblers utilize information from the reference genome to produce transcriptome assemblies [34, 32]. The quality of genome-guided assembly can be primarily affected by the quality of the reference genome and its closeness to the target transcriptome. In *de novo* assembly, assemblers overlap the millions of reads into contiguous sequences (contigs) representing transcripts. Although this is often a challenging process and often less accurate compared to genome-guided assemblers, *de novo* transcritpome assemblers are important especially for non-model organisms that lack a fully sequenced genome. When a suitable reference genome is unavailable, only *de novo* assemblers can be used.

Despite the difference in the underlying algorithms, many individual assemblers can correctly assemble the majority of the core set of transcripts [46, 2]. Ensemble assemblers leverage this by combining multiple assemblies to retain contigs that are likely to be correctly assembled. While many transcriptome assemblers have been developed, transcriptome assembly remains a non-trivial task, attracting many studies to make it more efficient and improve assembly correctness and overall assembly completeness.

2.1.1 Genome-guided assembly

As illustrated at the bottom of Figure 2.1, genome-guided assemblers build upon an existing reference genome, as well as RNA-seq reads obtained, to produce a transcriptome assembly for the target organism. Genome-guided assemblers generally perform

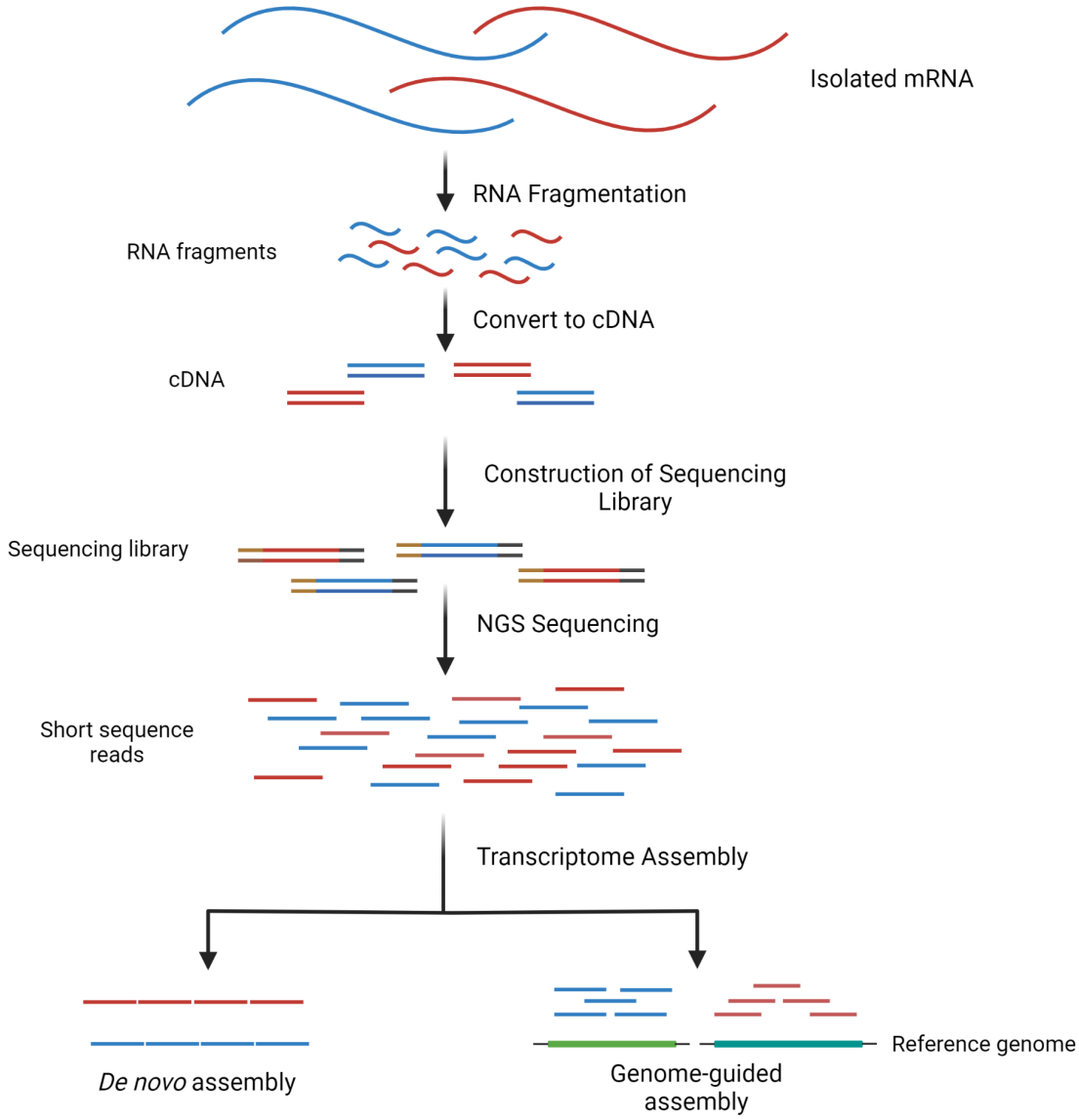


Figure 2.1: RNA-seq overview. mRNAs are first isolated from cell or tissue samples. Next, fragmentation and cDNA synthesis is performed. Conversion of mRNAs to cDNAs is a conventional approach as most sequencing technologies require DNA libraries. Construction of the sequencing library follows, including ligation of sequence adaptors (brown) to cDNA strands to preserve strand information. Sequencing using NGS platforms such as Illumina is employed using the prepared library producing reads 75-100bp long. Finally, the transcriptome is assembled using computational methods, either *de novo* or genome-guided assembly approach.

the assembly in three steps: (i) The RNA-seq reads are aligned to the reference genome using a splice-aware aligner such as TopHat2 [26], HISAT2 [25], STAR [11], or SpliceMap [1][2]. Splice-aware aligners identify exons at slice-junctions in the reference genome and ensure that no reads are aligned to an intron [2]; ii) Aligned reads from each locus are then clustered to construct a splice graph representing all possible isoforms for a gene; (iii) Finally, the representative splice graph is traversed resolving individual isoforms [2]. Graph construction and traversal algorithms and the choice of aligners differ among assemblers, which is fundamental to the quality of the assembly results. Further, the quality of the reference is also vital for accurate assembly [2]. In a situation where the reference genome is divergent from the read sequences, such as the human reference genome against the read sequences from another primate genome, depending on the level of divergence between the human and primate genomes, the resulting assembly would comprise highly inaccurate transcripts. Even in situations where the reference-genome is only slightly different, such as different strains of the same species or different versions, the accuracy of the assembly would be affected [46]. We show this in the results in Section 4.2.

One of the genome-guided assemblers is Scallop [40]. Scallop resolves weighted paths of the splice graph and finds the minimum set of paths to determine the assembled transcripts. With some modification to the general approach, Cufflinks [45] constructs a directed overlap graph of fragment alignments on each independent gene locus, which is then transitively reduced to extract a minimum path cover (i.e., a minimum-size set of paths covering all the nodes) of the graph. The reduced graph is then used to derive a weighted bipartite graph whose maximum matching represents the assembly. Bayesembler [33] approaches assembly probabilistically through the derivation of a Bayesian model of the RNA sequencing process. It establishes candidate transcripts from a set of splice graphs and uses Bayesian inference to resolve the

most likely combination of candidates. The model maintains consistent means of combining prior knowledge about the inadequacy in the number of expressed transcripts with information from the reads and their abundance. StringTie2 [27] determines the expression levels of transcripts while assembling isoforms of every gene by finding the maximum flows in flow networks constructed for each of the heaviest paths in the splice graph. StringTie2 extends the original StringTie [38] by implementing a novel control flow algorithm to reconstruct transcripts, and a more refined method to merge paired reads into fragments from the initial stage of the assembly. It also implements more efficient data structures to provide the capacity to handle long-read sequences [27]. Despite StringTie2 focuses on assembly using long-read sequences, reportedly it mostly generates more accurate transcriptome assemblies than StringTie even when using short-read sequences.

2.1.2 *De novo* assembly

De novo assembly is a reference-free-based strategy that leverages motifs of RNA-seq reads to find overlaps assembled into longer contigs. It can be applied to organisms that do not have a well-annotated genome and in many cases where no sequence information exists. *De novo* assembly is also used to complement genome-guided assemblers because some contigs may only be assembled in *de novo* assemblers [46]. Many *de novo* assemblers construct de Bruijn graphs from RNA-seq reads and identify contigs as optimal paths (longest paths or paths greater than selected threshold) within the graphs (see Figure 2.2) [39, 24, 8, 37, 18]. In de Bruijn graphs, a node is defined by a substring of a read sequence of fixed length k (k-mer), with k substantially shorter than the read length. The nodes are then adjacent if they overlap by a $k-1$ nucleotide substring and the read dataset supports this link. The choice of the k-mer has a significant impact on the assembly result. Small k-mer values produce

large amounts of short contigs, consequently covering the less abundant transcripts and producing highly fragmented contigs [23, 9, 12]. They also cannot deal with repetitive sequences. While large k-mer values tend to produce a more contiguous assembly consisting of high coverage contigs and more splice variants, the assembly contains longer but fewer contigs leading to lower transcript representation [9, 12]. For this reason, using a single k-mer in *de novo* assembly can often result in incomplete assembly of transcriptomes leading to loss of relevant information. Choosing whether to use a single k-mer or multiple k-mers is among the nuances in several *de novo* assemblers.

Trinity [18] uses a single k-mer to fully reconstruct a large fraction of transcripts with low base error rates [20] while applying de Bruijn graphs. It performs an exhaustive enumeration to find paths in the graph and is regarded as one of the best single k-mer assemblers currently available [23, 31]. SOAPdenovo-trans [49] also uses a single k-mer with a slightly modified de Bruijn graph algorithm, which lowers redundancy, and has a lower computation time than Trinity. IDBA-tran [37] adopts the idea of multiple k-mers to capture information from transcripts with both high and low expression using a slightly modified de Bruijn graph. Instead of building a de Bruijn graph and finding contigs for each k-mer value, an accumulated graph is built iteratively where the output from one iteration is treated as input in the next iteration until it reaches the maximum size of k-mer. The result of these intermediate graphs is a compound de Bruijn graph that spans all the k-mer values. rnaSPades [4] is another single k-mer method with a slightly modified de Bruijn graph where chimeric and erroneous edges are removed from the graph during assembly. Bridger [7] employs a new *de novo* approach that uses a rigorous mathematical model, called the minimum path cover, to construct splice graphs used to build compatibility graphs for transcriptome reconstruction. Splice graphs are directed acyclic graphs, whose nodes

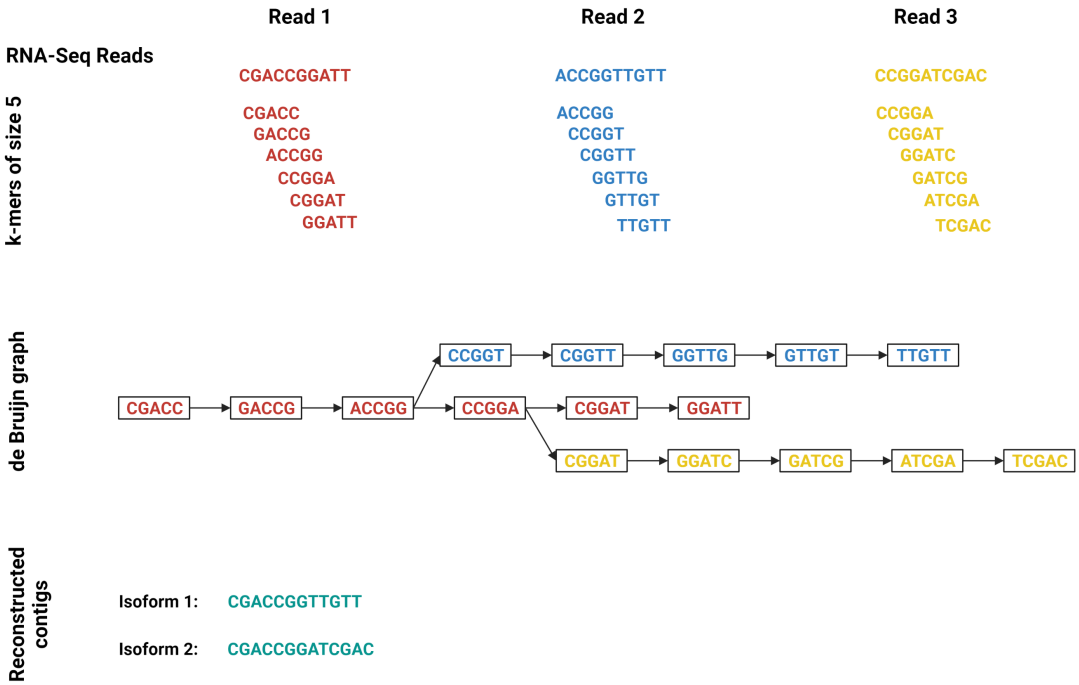


Figure 2.2: Example of de Bruijn graph. Substrings of length k (k-mers, here 5-mers) are found from the three reads shown at the top. The nodes (rectangles) of the de Bruijn graph are defined by k-mers. Adjacent nodes have overlapping prefix and/or suffix of $k-1$ nucleotides and are also supported by the read dataset. The longest paths or those longer than a specified threshold in the graph are extracted and contigs are obtained from those paths. In this example, two paths of length ≥ 10 (assuming a threshold of 10) are extracted leading to two final contigs. They are most likely isoforms derived from the same gene.

correspond to exons and edges represent splicing junctions, where splicing events occur [21]. Bridger typically uses a single k-mer but has a counterpart, Bridger-M, utilizing multiple k-mer values and merging assemblies from the different k-mers. A more recent assembler, BayesDenovo [41], integrates the Bayesian framework and a read-guided strategy to construct splice graphs from de Bruijn graphs. Unlike many other assemblers, BayesDenovo uses a non-deterministic approach that identifies contigs probabilistically.

2.1.3 Ensemble approach

Individual assemblers encounter often challenges covering the lowly expressed transcripts, which leads to less accurate and incomplete assemblies. As much as *de novo* and genome-guided assemblers face these challenges and produce varying assembly results, they often assemble the majority of core transcripts correctly [46]. Therefore, it is reasonable to consider that merging results from multiple assemblers would deliver a more complete and accurate assembly. Ensemble assemblers attempt to address the limitations of individual assemblers while preserving the correctly assembled transcripts [2, 46]. Typically, ensemble assemblers cluster contigs from multiple assemblers and choose a representative contig from each cluster to form the final [2, 46]. However, there is no guarantee that the representative contig retained from each cluster is the correct assembly of the transcript. While different ensemble assemblers have distinct approaches for clustering and selection of representative contigs, their final assemblies are in general of relatively higher quality compared to individual assemblers.

EvidentialGene [16, 17] and the "Concatenation" method [6] merge multiple *de novo* assemblies, use CD-HIT [14] to cluster the contigs, and select representative contigs for the final assembly set. TransBorrow [50], on the other hand, uses genome-guided assemblies. It combines results from different assemblers, builds splice graphs from mapped reads, and extracts paired subpaths from the splice graphs. Reliable subsequences from the assemblers and the paired subpaths are then mapped to the splice graphs to form subpaths that guide the final assembly. Recently, Voshall et al. developed ConSemble [46], an ensemble transcriptome assembly approach that combines results from four transcriptome assemblers (either *de novo* or genome-guided methods). ConSemble performs *de novo* transcriptome assembly using four assem-

blers (Trinity, SOAPdenovo-trans, IDBA-tran, and rnaSPAdes) with multiple k-mers for each method. For each assembler, contig sets are merged and clustered based on the coded protein sequences. It then proceeds to find contig sequences overlapped among three or more assemblers at the protein level producing the final contig sets. The procedure is the same for genome-guided assembly, apart from the different set of four assemblers (Cufflinks, Bayesembler, Scallop, and StringTie2).

Compared to the other essemble approaches, ConSemble has a superior performance in the *de novo* assembly and performs considerably better than TransBorrow in the genome-guided assembly [2, 46]. It recovers more correctly assembled transcripts with a high precision, hence reducing the number of misassembled transcripts. EvidentialGene over-assembles, and therefore recovers significantly large numbers of false positive contigs resulting in less accurate overall assemblies. In datasets where isoforms are present, all the ensemble methods performed well identifying isoforms for genes with five or less isoforms while ConSemble and Concantenation perform better for genes with five or more isoforms [2, 46]. It is also important to note that ensemble methods are limited by the individual assembly methods they use.

2.2 Transcriptome Assembly Performance Evaluation

Given the complexity of transcriptome assembly, it is imperative that the assembly performance and accuracy are quantitatively evaluated to fully understand the quality of the assembled transcriptomes. Different evaluation methods have been proposed. There are two approaches: reference-free and reference-based. Reference-free methods are useful in the absence of a benchmark reference dataset. They employ statistical metrics calculated solely on the assembled transcriptome. Some incorporate also the mapping efficiency of RNA-seq reads. Reference-based methods utilize benchmark

datasets to provide different metrics based on the comparison between the assembled transcriptome to the benchmark.

2.2.1 Reference-free metrics

The most commonly applied reference-free metrics are various descriptive statistics of the assembly output. These include:

1. The number of assembled contigs: This is the simplest metric. While fewer numbers of contigs do not necessarily indicate a better assembly, too large a number of assembled contigs can be indicative of the presence of either or both of fragmented contigs or false positives.

2. Median contig length: This is another simple metric. Although it alone cannot be used for peformance analysis, it can provide useful descriptive information regarding the assembled contigs.

3. N50 (Nx): It describes the longest contig such that all contigs of at least that length compose at least 50% (x%) of the bases in the assembly [30, 35]. It was originally developed to evaluate genome assembly where longer contigs are generally considered to be better. It is motivated by the idea that a more significant number of identified overlaps among input reads will have more reads assembled into contigs resulting in a better assembly. However, it is apparent that the trivial concatenation of all input reads would maximize the N50 metric, leading to low-quality assemblies.

Despite the above metrics being informative regarding the nature of assemblies, they are often misleading especially for transcriptome assessment. For example, the number and the length of contigs are not always good indicators of transcriptome

assembly accuracy and quality since the length and abundance of transcripts vary depending on the genes and the level of their expression [22, 36].

Alternative methods such as RSEM-EVAL provided by DETONATE (*DE novo* TranscriptOme rNa-seq Assembly with or without the Truth Evaluation) [30] and TransRate scores [43] leverage the biological properties of transcriptomes to address the inadequacies of the aforementioned descriptive metrics. They combine multiple factors, including the compactness of an assembly and how the RNA-seq data support the assembly.

RSEM-EVAL is a probabilistic model-based score that depends solely on the assembly and the corresponding RNA-seq reads [30]. The RSEM-EVAL score of an assembly (A) is defined as the log joint probability of the A and the reads D used to construct A:

$$score_{RSEM\text{-}EVAL}(A) = \log P(A, D) \tag{2.1}$$

Therefore, a better and more complete assembly maximizes the RSEM-EVAL score.

TransRate [43] provides a detailed assessment of the assembly by scoring the quality of each contig (TransRate contig score) and establishes an overall TransRate assembly score that is a statistical summary of the contig scores. The TransRate contig score is the product of the following scores:

1. $s(C_{nuc})$: It measures the extent to which the nucleotides in the mapped reads are the same as those in the assembled contig. A better assembly is one whose nucleotides accurately represent the nucleotides of the true transcript, therefore, maximizing this score.

2. $s(C_{cov})$: It measures the proportion of nucleotides in the contig with no sup-

porting read data. It penalizes contigs with no coverage and negatively affects the overall contig score.

3. $s(C_{ord})$: It measures the extent to which the order of the nucleotides in the contig matches the order in the mapped reads. Incorrectly mapped reads negatively affect the contig score and inform about partially assembled transcripts.

4. $s(C_{seg})$: This score measures the probability that the coverage depth of the transcript represents a single transcript and not a chimeric assembly.

The TransRate assembly score aims to provide insight into the accuracy and completeness of any given assembly. It is evaluated as the geometric mean of the mean TransRate contig score and the proportion of reads that map to the assembly.

2.2.2 Reference-based metrics

If a gold standard reference dataset is available, reference-based metrics can be calculated using either actual biological data, simulated data, or both, depending on the availability of such datasets. In the case of biological data, the metrics are determined through the alignment of RNA-seq reads or contigs to the reference assembly.

DETONATE implements REF-EVAL. It estimates the true assembly of contigs or scaffolds based on alignments of RNA-seq reads to reference transcripts [30]. Using these true contigs or scaffolds, REF-EVAL provides assembly precision, recall, and F_1 scores at contig (scaffold) or nucleotide level. Precision is the fraction of contigs (scaffolds) or nucleotides that correctly map to the reference sequences (true contigs or scaffolds). Recall is the fraction of the reference sequences (at either contig or nucleotide level) that are correctly recovered by the assembly. F_1 score is the harmonic mean of precision and recall (see Equations 2.3 - 2.5). REF-EVAL further provides the k-mer compression (KC) score, which is a combination of the weighted k-mer

recall (WKR) and inverse compression rate (ICR):

$$score_K C = WKR - ICR \tag{2.2}$$

The KC score measures the degree to which the assembly compresses the RNA-seq data.

The quality and completeness of the assembly can also be evaluated based on the number of genes or proteins found in the assembly relative to a reference dataset. rnaQuast [5] achieves this by aligning an assembly to an annotated reference genome, calculating various alignment and gene-database metrics such as the total number of genes and isoforms, isoform and exon length distribution, average number of exons per gene, numbers of aligned, unaligned, or misassembled transcripts. BUSCO (Benchmarking Universal Single-Copy Orthologs) [42] also assesses assembly completeness against lineage-specific protein sets. The BUSCO sets are searched in the assembly at protein level and the results are summarized into the number of genes in four categories: complete (C), duplicated (D), fragmented (F), and missing (M) [42].

If simulated datasets are available, as we have applied in this study, there is an advantage of knowing the ground truth in benchmark datasets. A set of known transcripts (benchmark) is used to simulate RNA-seq reads. The read sets are used to assemble contigs by using an assembly method to be evaluated. Finally, the assembled contig sets are compared with the benchmark transcripts. The comparison usually seeks to find contigs that are perfectly identical (100%) to sequences in the benchmark. Alternatively, a less stringent comparison can be performed using a lower threshold ($< 100\%$) to identify not completely but nearly identical contigs [46]. Results of the comparison are categorized as (i) true postives (TP) – correctly assembled

contigs with matches in the benchmark, (ii) false positives (FP) – misassembled contigs with no matches in the benchmark, or (iii) false negatives (FN) – benchmark transcripts without corresponding assembled contigs. While true negatives (TN) can be determined if the benchmark datasets include negative sequences, from which read sequences are not generated, as in the case of this study, often TN is not available. The numbers of contigs in each categories are denoted as *TP*, *FP*, and *FN*, respectively. The comparisons can be performed either at nucleotide or protein level. In either case, the following three performance metrics can be calculated:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{2.5}$$

Precision provides information about the proportion of correctly assembled contigs relative to the assembly. A higher value of *precision*, therefore, indicates that there are more correctly assembled contigs than incorrect ones in the assembly set. *Recall*, on the other hand, informs about correctly assembled contigs relative to the actual transcripts in the benchmark. A higher *recall* shows that the assembly recovers a good number of benchmark transcripts correctly achieving a more complete assembly. Finally, F_1 is the harmonic mean of *recall* and *precision*. It is a combined metric that

balances the trade-off between *precision* and *recall*.

Chapter 3

Materials and Methods

3.1 Reference Genomes and Transcriptomes Used

Reference genomes and transcriptomes used in this study were obtained from the original ConSemble publication [46]. This was done to ensure a direct comparison to the published results. The data is freely available at `http://bioinfolab.unl.edu/emlab/consemble/`. The four reference genomes used are as follows: the *Arabidopsis thaliana* accession Nossen (No-0) genome originally assembled by Gan et al. [15], the *A. thaliana* accession Columbia (Col-0) genome from TAIR (version 9) [44], the human HG38 (Human) reference genome (GCF_000001405.39), and the human HX1 reference genome[1].

3.2 Simulated Benchmark Transcriptomes and Read Sets

The simulated benchmark transcriptomes and read sets were also obtained from the ConSemble [46] publication. In summary, for each reference benchmark transcriptome (Col-0-Ref, No-0-Ref, or Human-Ref), approximately 250 millions of 76bp read pairs were generated using the Flux Simulator v1.2.1 [19]. The varied numbers of isoforms among the datasets accounted for the different levels of complexity often found in

[1] `http://hx1.wglab.org`

real biological data. The No-0-Ref dataset is the simplest and does not include any isoforms per gene. The Col-0-Ref and Human-Ref datasets contain up to nine and forteen isoforms, respectively [46].

3.3 Read Processing

After the simulation, generated reads were quality filtered using Erne-filter 2.0 [13] to remove low-quality bases and filter out contaminated bases at an average quality of q20 with the "ultra-sensitive" flag. This resulted in 494 million, 493 million, and 491 million reads for No-0, Col-0, and Human datasets, respectively, discarding about 3 million reads from each dataset. The filtered reads were subsequently normalized using the "normalization-by-median" method of the Khmer software package [10], with a k-mer length of 32 bp, memory size of 32GB, and 50$\times$ expected coverage to remove redundant reads and low quality reads prior to the assembly process. Both the quality filter and normalization were performed in the paired-end mode.

3.4 *De Novo* Assembly

For four assemblers Trinity, SOAPdenovo-trans, IDBA-tran, and rnaSPAdes, the assemblies were obtained from ConSemble [46]. Each assembly was performed using the default k-mer length as well as multiple lengths as follows;

- Trinity: 19 - 31 $k$mer lengths with increments of 4,
- rnaSPAdes: 17 - 71 k-mer lengths with increments of 4,
- SOAPdenovo-trans: 15 - 75 k-mer lengths with increments of 4, and
- IDBA-tran: 20 - 60 k-mer lengths with increments of 10.

In addition, a new *de novo* assembly was performed using BayesDenovo v1.0 with a single default k-mer length (25) as well as multiple k-mer lengths ranging from 21 to 31 with increments of 2, to best cover the allowable range of values (20 to 32).

3.5 Genome-guided Assembly

The four genome-guided methods, Cufflinks, Bayesembler, Scallop, and StringTie2, were obtained from ConSemble [46].The reads were mapped to the reference genomes using Tophat2 v2.0.14 [26]. An additional assembly was performed using StringTie v1.3 and included in the analysis to determine the performance including five assemblers. This also allowed us to evaluate the effect of using multiple versions of the same assembler on the assembly quality. Following the experimental design used in Voshall et al. [46], each read set was assembled using two different reference genomes as shown in Table 3.1. This was done to evaluate the effect of the similarity of the reference genome to the read set mapped against.

Table 3.1: Experimental design of the genome-guided assembly using two types of references

Read set	**Assembly type**	**Reference genome**
No-0	same	No-0
No-0	different	Col-0
Col-0	same	Col-0
Col-0	different	No-0
Human	same	Human (HG38)
Human	different	HX1

3.6 Ensemble Assembly

The ConSemble assemblies were obtained from the ConSemble publication [46]. For *de novo* assembly, Trinity, SOAPdenovo-trans, IDBA-tran, and rnaSPAdes were used.

The assemblies were pooled from different $k-$mer lengths as described in Section 3.4. For genome-guided assembly, Bayesembler, Cufflinks, Scallop, and StringTie2 assemblies from the same and different references were used.

3.7 ConSemblEX Implementation

ConSemblEX is implemented in Python 3 using an object-oriented programming approach (Figure 3.1). It has three main classes, Assembly, Analysis, and Read Processing, which house its modular functionality and are abstracted from the user. ConSemblEX is accessed using the command shell interface through specific commands detailed in the documentation. The code and documentation is available freely at the ConSemblEX GitHub repository[2].

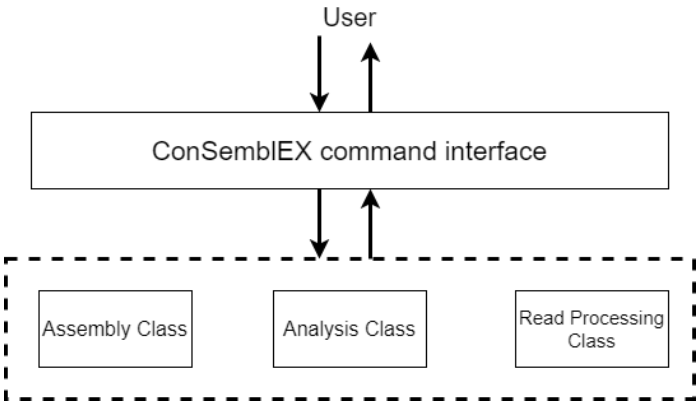


Figure 3.1: ConSemblEX architecture.

In principle, ConSemblEX is an extended version of ConSemble originally developed by Vorshall et al. [46]. In ConSemblEX, any number of methods in any combination, as opposed to using only previously chosen four of either *de novo* or genome-guided methods, can be used. To demonstrate this, five of both *de novo* and genome-guided assemblers were used in this study. Trinity, SOAPdenovo-trans,

[2] `https://github.com/bioinfo-emlab-unl/consemblex`

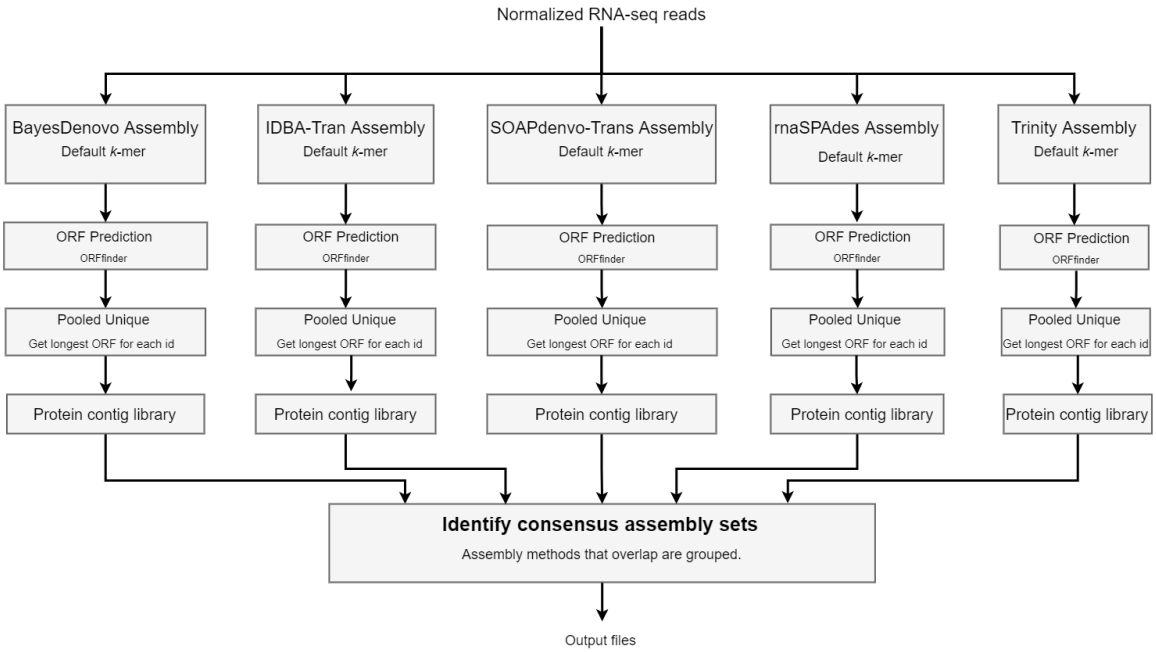


Figure 3.2: The ConSemblEX pipeline for *de novo* default assembly.

IDBA-tran, rnaSPAdes, and BayesDenovo were used in the *de novo* assembly, while Cufflinks, Bayesembler, Scallop, StringTie, and StringTie2 were used in the genome-guided assembly.

For the *de novo* assembly, two approaches were explored. In the first pipeline shown in Figure 3.2, all five assemblers were run with the default k-mer lengths. In the second pipeline shown in Figure 3.3, each of the five *de novo* assembly was performed using varying k-mer lengths as described in Section 3.4, and the contigs produced using the different k-mer were pooled into one contig set for each method. As shown in Figure 3.4, the genome-guided assembly was perform in two ways, using the reference genome same as or different from the read dataset source (see Table 3.1).

Each of the five contig sets is passed to ORFfinder [48] to search for open reading frames (ORFs). The longest ORF found from each contig is translated into the protein sequence. This converts each contig set (nucleotide sequences) to a "pro-

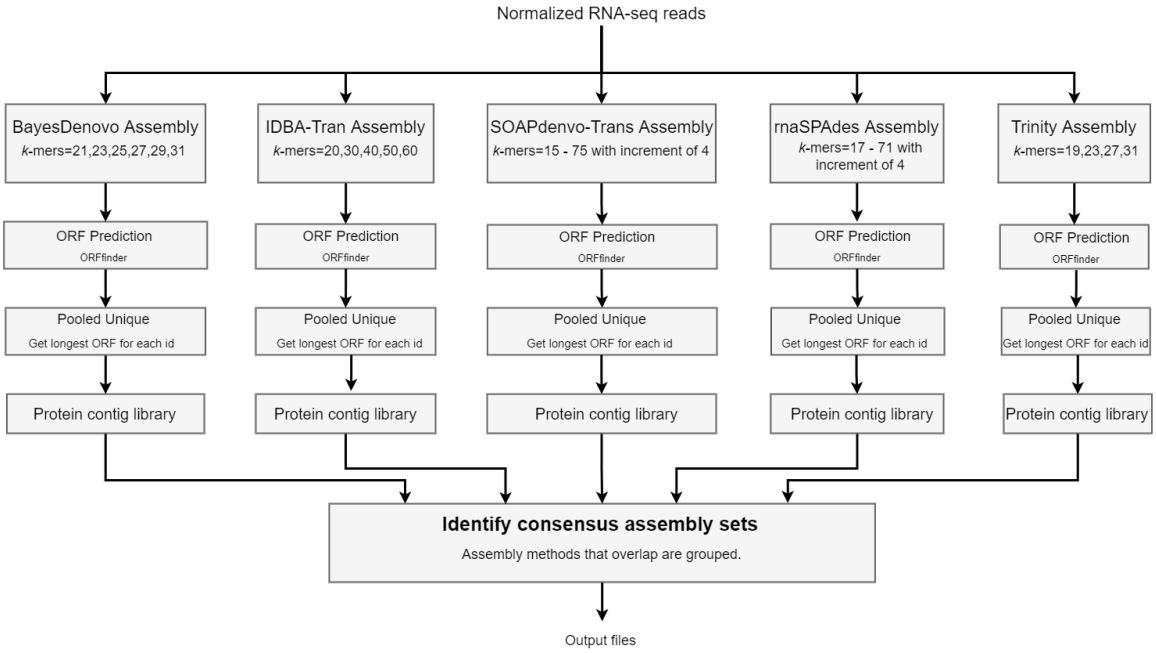


Figure 3.3: The ConSemblEX pipeline for *de novo* pooled assembly.

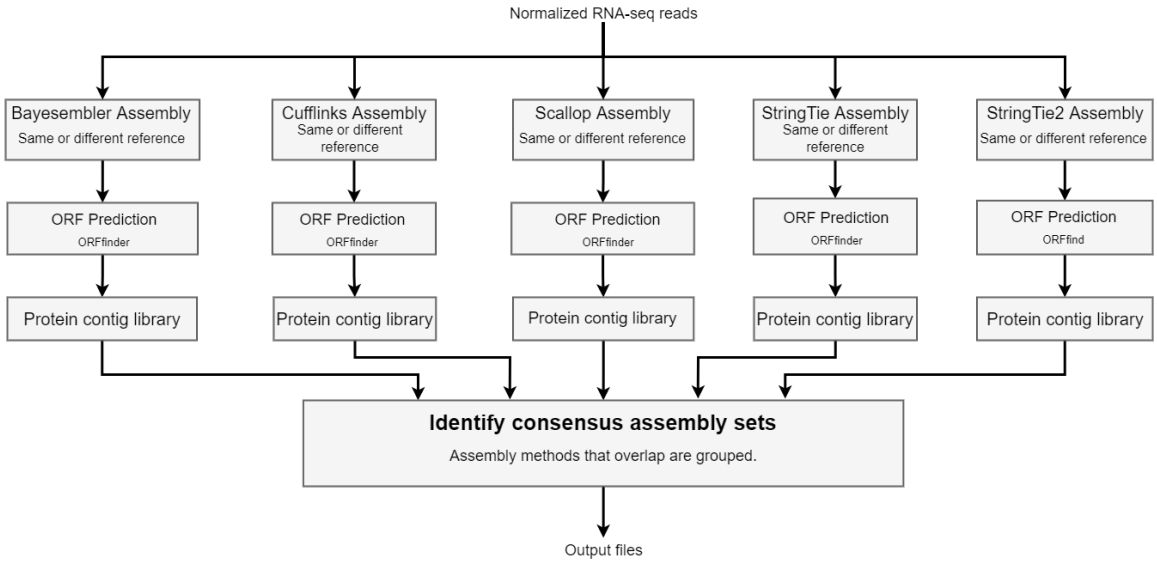


Figure 3.4: The ConSemblEX pipeline for genome-guided assembly.

tein contig library" (including only unique protein sequences), from which the final consensus assembly is performed. The basis of the consensus assembly is to identify contigs that share protein sequences among the contig libraries generated from different assemblers. Hence, from the five protein contig libraries, contigs are clustered if their protein sequences are identical (100%). As shown in Tables 3.2 and 3.3, using these clusters, contigs are grouped into 31 discrete sets based on the overlapping assembly methods. The 31 distinct sets are further sub-grouped according to their cardinality (overlap size), forming the intermediate output of ConSemblEX assembly.

Higher levels of ConSemblEX assembly can be obtained based on the cardinality of the overlapping assembly set. Following the notation used in ConSemble, when exactly N ($1 \leq N \leq 5$) overlaps are required, all sets with the cardinality equal to N are selected and the assembly is referred to as ConSemblEX-N (i.e., ConSemblEX-2, ConSemblEX-3, ...). Depending on the assembly approach in question, *de novo* or genome-guided, the assembly is referred to as ConSemblEX-Nd (i.e., ConSemblEX-2d, ConSemblEX-3d, ...) or ConSemblEX-Ng (i.e., ConSemblEX-2g, ConSemblEX-3g, ...), respectively. When N or more overlaps are required, assembly is called ConSemblEX-N+d (i.e., ConSemblEX-2+d, ConSemblEX-3+d, ...) for *de novo* assembly and ConSemblEX-N+g (i.e., ConSemblEX-2+g, ConSemblEX-3+g,...) for genome-guided assembly. Instead of trying to provide possibly the single best assembly output, ConSemblEX provides a collection of outputs that includes all possible exclusive sets of overlaps (intersections) among the assembly methods, as well as the union sets of the exclusive overlaps. All these contig sets are available at both nucleotide and protein levels. This provides the maximum flexibility to choose the best assembly for particular needs.

As mentioned earlier, ConSemblEX is modularized, which means actual assembly is independent of the analysis. Therefore, a new assembly method can be included

Table 3.2: Discrete sets of *de novo* assembly contigs

Overlaps	**Distinct assembly sets**
1-way	BayesDenovo
	IDBA-tran
	SOAPdenovo-trans
	rnaSPAdes
	Trinity
2-way	BayesDenovo ∩ IDBA-tran
	BayesDenovo ∩ SOAPdenovo-trans
	BayesDenovo ∩ rnaSPAdes
	BayesDenovo ∩ Trinity
	IDBA-tran ∩ SOAPdenovo-trans
	IDBA-tran ∩ rnaSPAdes
	IDBA-tran ∩ Trinity
	SOAPdenovo-trans ∩ rnaSPAdes
	SOAPdenovo-trans ∩ Trinity
	rnaSPAdes ∩ Trinity
3-way	BayesDenovo ∩ IDBA-tran ∩ SOAPdenovo-trans
	BayesDenovo ∩ IDBA-tran ∩ rnaSPAdes
	BayesDenovo ∩ IDBA-tran ∩ Trinity
	BayesDenovo ∩ SOAPdenovo-trans ∩ rnaSPAdes
	BayesDenovo ∩ SOAPdenovo-trans ∩ Trinity
	BayesDenovo ∩ rnaSPAdes ∩ Trinity
	IDBA-tran ∩ SOAPdenovo-trans ∩ rnaSPAdes
	IDBA-tran ∩ SOAPdenovo-trans ∩ Trinity
	IDBA-tran ∩ rnaSPAdes ∩ Trinity
	SOAP ∩ rnaSPAdes ∩ Trinity
4-way	BayesDenovo ∩ IDBA-tran ∩ SOAPdenovo-trans ∩ rnaSPAdes
	BayesDenovo ∩ IDBA-tran ∩ SOAPdenovo-trans ∩ Trinity
	BayesDenovo ∩ IDBA-tran ∩ rnaSPAdes ∩ Trinity
	BayesDenovo ∩ SOAPdenovo-trans ∩ rnaSPAdes ∩ Trinity
	IDBA-tran ∩ SOAPdenovo-trans ∩ rnaSPAdes ∩ Trinity
5-way	BayesDenovo ∩ IDBA-tran ∩ SOAPdenovo-trans ∩ rnaSPAdes ∩ Trinity

directly in the analysis by simply including the assembly output, regardless of the method and where it was performed. ConSemblEX will perform the analysis based on the new number of methods and will output the necessary results without need for any modification.

Table 3.3: Discrete sets of genome-guided assembly contigs

Overlap	**Distinct assembly sets**
1-way	Bayesembler
	Cufflinks
	Scallop
	StringTie
	StringTie2
2-way	Bayesembler ∩ Cufflinks
	Bayesembler ∩ Scallop
	Bayesembler ∩ StringTie
	Bayesembler ∩ StringTie2
	Cufflinks ∩ Scallop
	Cufflinks ∩ StringTie
	Cufflinks ∩ StringTie2
	Scallop ∩ StringTie
	Scallop ∩ StringTie2
	StringTie ∩ StringTie2
3-way	Bayesembler ∩ Cufflinks ∩ Scallop
	Bayesembler ∩ Cufflinks ∩ StringTie
	Bayesembler ∩ Cufflinks ∩ StringTie2
	Bayesembler ∩ Scallop ∩ StringTie
	Bayesembler ∩ Scallop ∩ StringTie2
	Bayesembler ∩ StringTie ∩ StringTie2
	Cufflinks ∩ Scallop ∩ StringTie
	Cufflinks ∩ Scallop ∩ StringTie2
	Cufflinks ∩ StringTie ∩ StringTie2
	Scallop ∩ StringTie ∩ StringTie2
4-way	Bayesembler ∩ Cufflinks ∩ Scallop ∩ StringTie
	Bayesembler ∩ Cufflinks ∩ Scallop ∩ StringTie2
	Bayesembler ∩ Cufflinks ∩ StringTie ∩ StringTie2
	Bayesembler ∩ Scallop ∩ StringTie ∩ StringTie2
	Cufflinks ∩ Scallop ∩ StringTie ∩ StringTie2
5-way	Bayesembler ∩ Cufflinks ∩ Scallop ∩ StringTie ∩ StringTie2

3.8 Benchmarking and Assembly Performance Analysis

All the final assemblies were compared to the corresponding benchmark dataset (Col-0, No-0, or Human) to determine the number of correctly and incorrectly assembled contigs in each assembly. Contigs were evaluated at the protein level and at the 100%

identity to the translated transcripts in the benchmark dataset. Contigs identical to the benchmark transcripts were considered true positives (TP). Contigs that were not fully identical to any transcripts in the benchmark were considered false positives (FP). Finally, transcripts in the benchmark that were not identical to any contigs in the assembly sets were considered false negatives (FN). As described in Section 2.2.2, in this study, true negative was not determined.

The overall performance was measured using *precision*, *recall*, and F_1 *score* as shown in Equations 2.3, 2.4, and 2.5.

3.9 Figures and Plotting

The intersection graphs were generated using the UpSetPlot Python3 API (version 0.6.0) [29]. The performance plots were generated using Matplotlib Python3 API[3] and Microsoft Excel Software. The biological figures were created using BioRender[4]. The pipeline diagrams and ConSemblEX architecture were created using draw.io[5].

[3] `https://matplotlib.org/stable/index.html`
[4] `https://app.biorender.com/`
[5] `https://app.diagrams.net/`

Chapter 4

Results and Discussion

4.1 Performance of *De Novo* Assembly Methods

4.1.1 Individual assembly methods

We began by comparing the performance of individual *de novo* assembly methods on the three datasets, No-0, Col-0, and Human, against the corresponding benchmark datasets No-0-Ref, Col-0-Ref, and Human-Ref. As shown in Table 4.1, when the default k-mers were used, all methods but BayesDenovo assembled a significantly larger number of contigs (shown as "Total") than the expected number (shown as "Actual") in all the datasets. BayesDenovo assembled fewer numbers of contigs than expected by values ranging from 1.24% to 20% for all the datasets, while the rest of the methods assembled much more contigs than expected, ranging from 25% (Trinity) to 113% (rnaSPAdes) more. Despite mostly large numbers of assembled contigs, the numbers of correctly assembled contigs (TP) were low in all assembly methods. BayesDenovo was the most precise method with the average *precision* of 0.54 (0.61, 0.51, and 0.49, for No-0, Col-0, and Human, respectively). Trinity had the best *recall* (0.64, 0.60, and 0.50, for No-0, Col-0, and Human, respectively) and F_1 (0.57, 0.53, and 0.45, for No-0, Col-0 and Human, respectively) for all datasets. All assembly methods performed relatively worse in the Col-0 and Human datasets as expected,

due to the presence of isoforms in these datasets. An interesting observation among the individual assemblers was how close the performance between Trinity and Bayesdenovo was in all the datasets in terms of F_1 score. However, Bayesdenovo had better *precision* but lower *recall* than Trinity in all the datasets. This disparity resulted from Bayesdenovo generally recovering fewer incorrect contigs than Trinity, which always recovered more correct contigs than Bayesdenovo. Therefore, Trinity recovered the most complete assemblies relative to the benchmark (even with higher FP), while Bayesdenovo generated the most accurate assemblies. Overall, Trinity provided the best assemblies with slightly better F_1 scores, closely followed by Bayesdenovo. However, this was at the cost of an increased number of false positives generated by Trinity.

Table 4.1: Performance of individual *de novo* assemblers using default k-mers[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	BayesDenovo	18875	16332	10022	6310	8853	**0.61**	0.53	**0.57**
	IDBA-trans	18875	22802	8344	14458	10531	0.37	0.44	0.40
	SOAPdenovo-Trans	18875	29876	11119	18757	7756	0.37	0.59	0.46
	rnaSPAdes	18875	40333	9206	31127	9669	0.23	0.49	0.31
	Trinity	18875	23523	12059	11464	6816	0.51	**0.64**	**0.57**
Col-0	BayesDenovo	15508	15316	7876	7440	7632	**0.51**	0.51	0.51
	IDBA-trans	15508	20430	6021	14409	9487	0.29	0.39	0.34
	SOAPdenovo-Trans	15508	21371	7281	14090	8227	0.34	0.47	0.39
	rnaSPAdes	15508	31494	7556	23938	7952	0.24	0.49	0.32
	Trinity	15508	19417	9255	10162	6253	0.48	**0.60**	**0.53**
Human	BayesDenovo	17669	14139	6914	7225	10755	**0.49**	0.39	0.43
	IDBA-trans	17669	20960	6154	14806	11515	0.29	0.35	0.32
	SOAPdenovo-Trans	17669	22005	5933	16072	11736	0.27	0.34	0.30
	rnaSPAdes	17669	21244	7637	13607	10032	0.36	0.43	0.39
	Trinity	17669	21279	8765	12514	8904	0.41	**0.50**	**0.45**

[a]The **boldfaced** numbers and green highlights show the best score among all the methods for each dataset.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of unique contigs assembled by the assembly method.

To recover more contigs, assemblies using multiple k-mer lengths were pooled for each assembly method. As shown in Table 4.2, in these pooled assemblies, all the methods assembled a large number of contigs ranging from 116% (BayesDenovo) to

1,269% (rnaSPades) for all datasets. This was because the numbers of incorrectly assembled contigs (FP) significantly increased for all datasets affecting the *precision*. For BayesDenovo, the *precision* reduces by close to 50%, from an average of 0.54 in the default assembly to 0.29 in the pooled assembly. Despite this negative effect on *precision*, the *recall* values increased in all datasets. The highest *recall* value among all the assembly methods increased from 0.64 in the default assembly by Trinity to 0.75 in the pooled assembly by rnaSPAdes (both for No-0). Overall, BayesDenovo exhibited a better balance between *precision* and *recall*, producing the best F_1.

Table 4.2: Performance of individual *de novo* assemblers using multiple k-mers[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	BayesDenovo	18875	40207	11945	28262	6930	**0.30**	0.63	**0.40**
	IDBA-trans	18875	106631	13799	92832	5076	0.13	0.73	0.22
	SOAPdenovo-Trans	18875	209403	13615	195788	5260	0.07	0.72	0.12
	rnaSPAdes	18875	258550	14172	244378	4703	0.05	**0.75**	0.10
	Trinity	18875	84687	12783	71904	6092	0.15	0.68	0.25
Col-0	BayesDenovo	15508	33562	9273	24289	6235	**0.28**	0.60	**0.38**
	IDBA-trans	15508	60312	10318	49994	5190	0.17	**0.67**	0.27
	SOAPdenovo-Trans	15508	158677	9324	149353	6184	0.06	0.60	0.11
	rnaSPAdes	15508	177276	9441	167835	6067	0.05	0.61	0.10
	Trinity	15508	77211	10104	67107	5404	0.13	0.65	0.22
Human	BayesDenovo	17669	28722	8179	20543	9490	**0.28**	0.46	**0.35**
	IDBA-trans	17669	52368	9301	43067	8368	0.18	0.53	0.27
	SOAPdenovo-Trans	17669	124507	9318	115189	8351	0.07	0.53	0.13
	rnaSPAdes	17669	249643	10216	239427	7453	0.04	**0.58**	0.08
	Trinity	17669	40726	9603	31123	8066	0.24	0.54	0.33

[a]The **boldfaced** numbers and green highlights show the best score among all the methods for each dataset.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of unique contigs assembled by the assembly method.

The results show that *de novo* assemblers are bound to behave differently depending on the dataset in question. Further, we will always encounter the trade-off between *precision* and *recall* depending on our approach. Therefore, it is apparent that using a single *de novo* assembly method will always be insufficient. ConSemble leverages the commonalities among various assembly methods to construct a consensus-based ensemble assembly. Despite how different assembly methods can be, their assembled

contigs have overlaps.

4.1.2 ConSemblEX assembly

Among the results of individual assembly (Table 4.1), the newly included method, BayesDenovo, performed consistently well in all the datasets. Therefore, inclusion of this method in the ConSemble pipeline is expected to improve the assembly performance. Using the current ConSemble pipeline, this cannot be achieved very easily. Thus, as described in Section 3.7 we introduced ConSemblEX, an extension of ConSemble, which allows us to include any number of assemblies in the consensus assembly pipeline and extract and examine all combinations of methods easily.

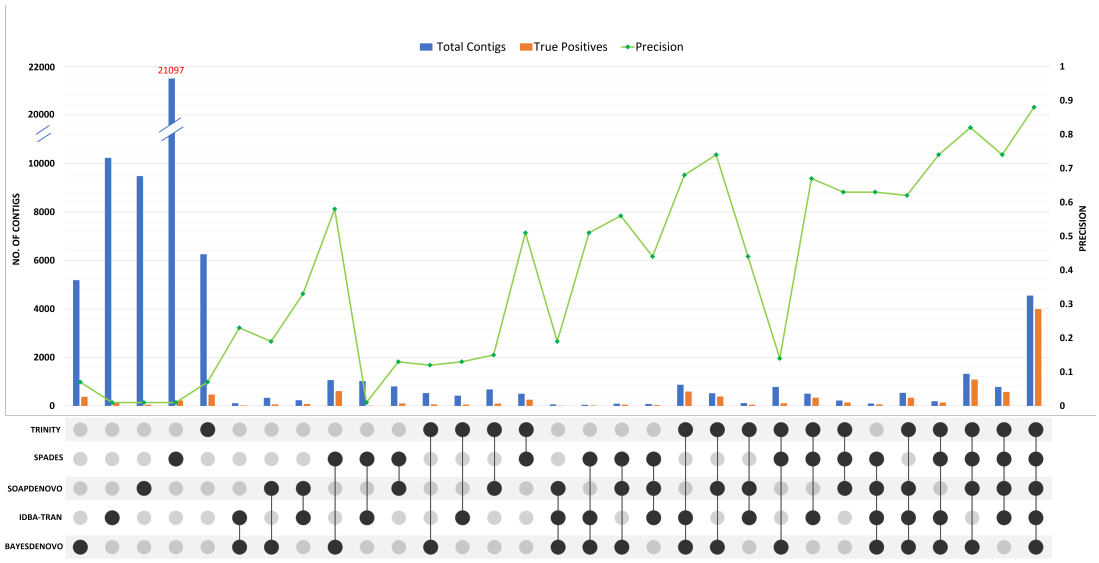
The principle behind the consensus assembly is to combine different assemblies and determine overlapping contigs among them. These overlapping contigs are more inclined toward being correctly assembled. In Figure 4.1 as well as Figures B.1 and B.2, how the correctly assembled contigs are overlapped among the assembly methods is illustrated. There are 26 distinct overlap sets each with a cardinality between 2 and 5 (Tables 4.3 as well as in Tables A.1 and A.2). *TP* increases proportionally with the cardinality of the overlap sets across the datasets. In the unique sets of individual assemblers, only a few contigs were correctly assembled out of the many contigs produced, resulting in a large *FP* (averages of 10,203 and 74,029, for default and pooled pipelines, respectively) and lower *precision*. When the 2-way overlaps were considered, the total number of contigs in the overlaps reduced significantly, reducing *FP* (averages of 878 and 5473, for default and pooled pipelines, respectively) and increasing the *precision*. The 3-way and 4-way overlaps consistently showed lower *FP*, besides a few anomalies, and had better *precision* than the 2-way and 1-way (unique) overlaps. When all the assembly methods were considered in the overlap (5-way intersection), 70% - 96% of the contigs were correctly assembled, leaving only

small proportions of contigs incorrectly assembled. The 5-way overlap may therefore seem like the best assembly because of its high precision. However, the assembled contigs include only $40 \sim 60\%$ of the benchmark transcripts (shown as *recall*) leading to not very high F_1 scores ($0.51 \sim 0.74$). Therefore, it provides an important assembly set that can be incorporated to obtain further improved assemblies.

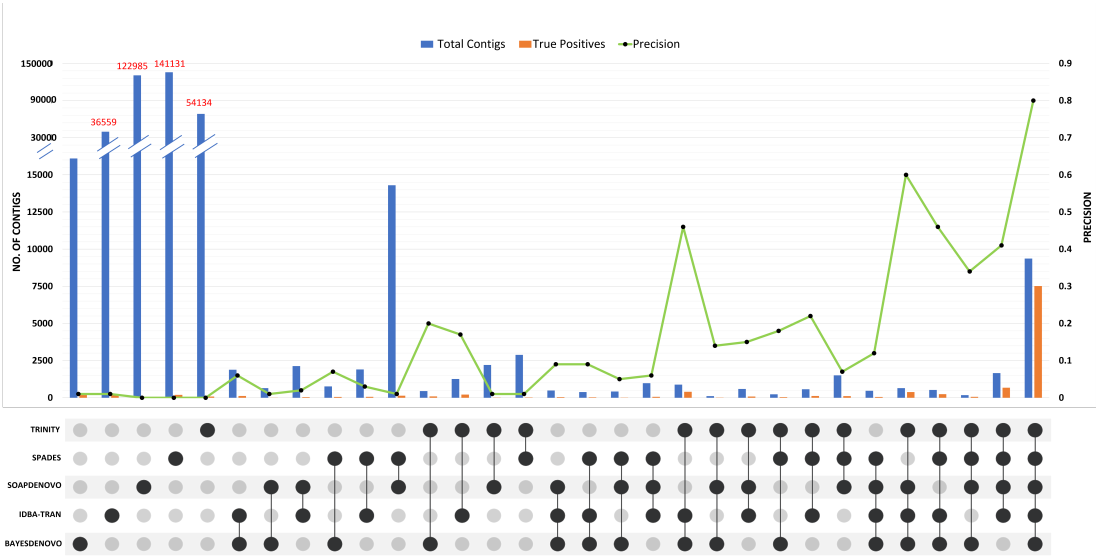
When the unions of various overlaps are considered, the trade-off between *precision* and *recall* became clearer. In the union of all assembled contigs, "1-way+" in Table 4.4 (also in Tables A.3 and A.4), the *recall* increased to the average of 0.72 (0.78, 0.74, and 0.64, respectively, for No-0, Col-0, and Human), indicating a good amount of benchmark transcripts recovered correctly. However, more false positives were introduced in the assembly reducing the *precision* significantly. The 2-way+ assembly set significantly reduced the number of false positives from an average of 450,000 in the 1-way+ assembly set to an average of 38,000 across the datasets, increasing the *precision* and hence F_1 significantly. The 3-way+ and 4-way+ assembly sets followed the same pattern and achieved the average *precision* and *recall* at 0.70 (4-way+) and 0.63 (3-way+), respectively, and F_1 at 0.63 (4-way+) accross the datasets. Higher overlaps show higher *precision* while lower overlaps tend to have higher *recall*.

4.1.3 Selecting the best ConSemblEX output assemblies

While any new assembly method can be added to ConSemblEX, it would be necessary to understand how each assembly method affects the overall consensus assembly. Table 4.3, shows details of the pooled consensus assembly for the Col-0 dataset, highlighting how each assembly method and overlaps among them contribute to the overall assembly. The detailed results for the No-0 and the Human datasets are found in Tables A.1 and A.2, respectively. Among the assembler unique sets, SOAPdenovo-Trans and rnaSPAdes produced large numbers of total unique contigs despite only a small



(a) Col-0 default k-mer



(b) Col-0 pooled k-mers

Figure 4.1: Distribution of the assembled contig overlaps for the Col-0 dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five *de novo* assemblers. Assembly was performed using the default k-mer values (a) or multiple k-mer values (b). Overlaps among the five methods are indicated with connected closed circles. Each overlapping set is exclusive to each other.

Table 4.3: *De novo* assembly overlaps using multiple k-mers for the Col-0 dataset[a]

BayesDenovo	IDBA-tran	SOAPdenovo	rnaSPAdes	Trinity	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					15508	16106	180	15926	15328	0.0112	0.0116	0.0114
	✘				15508	36559	239	36320	15269	0.0065	0.0154	0.0092
		✘			15508	122985	67	122918	15441	0.0005	0.0043	0.0010
			✘		15508	141131	197	140934	15311	0.0014	0.0127	0.0025
				✘	15508	54134	86	54048	15422	0.0016	0.0055	0.0025
All unique (1-way)					**15508**	**370915**	**769**	**370146**	**14739**	**0.0021**	**0.0496**	**0.0040**
✘	✘				15508	1884	117	1767	15391	0.0621	0.0075	0.0135
✘		✘			15508	648	7	641	15501	0.0108	0.0005	0.0009
✘			✘		15508	762	53	709	15455	0.0696	0.0034	0.0065
✘				✘	15508	451	91	360	15417	0.2018	0.0059	0.0114
	✘	✘			15508	2132	49	2083	15459	0.0230	0.0032	0.0056
	✘		✘		15508	1905	66	1839	15442	0.0346	0.0043	0.0076
	✘			✘	15508	1259	210	1049	15298	0.1668	0.0135	0.0250
		✘	✘		15508	14296	143	14153	15365	0.0100	0.0092	0.0096
		✘		✘	15508	2202	19	2183	15489	0.0086	0.0012	0.0021
			✘	✘	15508	2881	30	2851	15478	0.0104	0.0019	0.0033
All 2-way overlaps					**15508**	**28420**	**785**	**27635**	**14723**	**0.0276**	**0.0506**	**0.0357**
✘	✘	✘			15508	491	43	448	15465	0.0876	0.0028	0.0054
✘	✘		✘		15508	385	35	350	15473	0.0909	0.0023	0.0044
✘	✘			✘	15508	883	402	481	15106	0.4553	0.0259	0.0491
✘		✘	✘		15508	418	20	398	15488	0.0478	0.0013	0.0025
✘		✘		✘	15508	115	16	99	15492	0.1391	0.0010	0.0020
✘			✘	✘	15508	234	43	191	15465	0.1838	0.0028	0.0055
	✘	✘	✘		15508	977	62	915	15446	0.0635	0.0040	0.0075
	✘	✘		✘	15508	593	86	507	15422	0.1450	0.0055	0.0107
	✘		✘	✘	15508	576	125	451	15383	0.2170	0.0081	0.0155
		✘	✘	✘	15508	1509	108	1401	15400	0.0716	0.0070	0.0127
All 3-way overlaps					**15508**	**6181**	**940**	**5241**	**14568**	**0.1521**	**0.0606**	**0.0867**
✘	✘	✘	✘		15508	469	57	412	15451	0.1215	0.0037	0.0071
✘	✘	✘		✘	15508	644	386	258	15122	0.5994	0.0249	0.0478
✘	✘		✘	✘	15508	525	241	284	15267	0.4590	0.0155	0.0301
✘		✘	✘	✘	15508	178	61	117	15447	0.3427	0.0039	0.0078
	✘	✘	✘	✘	15508	1661	679	982	14829	0.4088	0.0438	0.0791
All 4-way overlaps					**15508**	**3477**	**1424**	**2053**	**14084**	**0.4095**	**0.0918**	**0.1500**
✘	✘	✘	✘	✘	**15508**	**9369**	**7521**	**1848**	**7987**	**0.8028**	**0.4850**	**0.6047**

[a]The highlights show the overlaps included in the ConSemblEX-select-d assembly. The **boldfaced** numbers show the best scores among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

fraction being correctly assembled. The total number of contigs found in their 2-way intersection set was much higher than in any other 2-way intersection due to its very high *FP*. The rest of the 2-way intersections produced ∼90% fewer contigs than the SOAPdenovo-Trans and rnaSPAdes intersection. However, all the 2-way intersections had low *precision*, mostly ≤ 0.2 in all the datasets. When the 3-way and 4-way intersections were considered, only a few 3-way intersections still had *precision* ≥ 0.2, while four out of five 4-way intersections had *precision* ≥ 0.2. This indicated that not all overlap sets could contribute positively to the consensus assembly. Therefore, a better approach would be needed to ensure that only those contributing positively to the consensus assembly were included in the "final" ConSemblEX set. Four assembly sets were constructed using four *precision* values (0.1, 0.2, 0.3, and 0.4) as

Table 4.4: *De novo* assembly using various union sets among overlapping contig sets for the Col-0 dataset[a]

Assembly sets[b]	Actual[c]	Total[d]	*TP*	*FP*	*FN*	Precision	Recall	F_1
1-way+	15508	418362	11439	406923	4069	0.0273	**0.7376**	0.0527
2-way+	15508	47447	10670	36777	4838	0.2249	0.6880	0.3390
3-way+	15508	19027	9885	9142	5623	0.5195	0.6374	0.5725
4-way+	15508	12846	8945	3901	6563	**0.6963**	0.5768	**0.6310**
1-way ∪ 2-way	15508	399335	1554	397781	13954	0.0039	0.1002	0.0075
2-way ∪ 3-way	15508	34601	1725	32876	13783	0.0499	0.1112	0.0688
3-way ∪ 4-way	15508	9658	2364	7294	13144	0.2448	0.1524	0.1879
1-way ∪ 4-way	15508	374392	2193	372199	13315	0.0059	0.1414	0.0112
2-way ∪ 4-way	15508	31897	2209	29688	13299	0.0693	0.1424	0.0932
1-way ∪ 5-way	15508	380284	8290	371994	7218	0.0218	0.5346	0.0419
2-way ∪ 5-way	15508	37789	8306	29483	7202	0.2198	0.5356	0.3117
3-way ∪ 5-way	15508	15550	8461	7089	7047	**0.5441**	**0.5456**	**0.5449**
1-way ∪ 2-way ∪ 3-way	15508	405516	2494	403022	13014	0.0062	0.1608	0.0118
1-way ∪ 2-way ∪ 4-way	15508	402812	2978	399834	12530	0.0074	0.1920	0.0142
1-way ∪ 3-way ∪ 4-way	15508	380573	3133	377440	12375	0.0082	0.2020	0.0158
1-way ∪ 2-way ∪ 5-way	15508	408704	9075	399629	6433	0.0222	0.5852	0.0428
1-way ∪ 3-way ∪ 5-way	15508	386465	9230	377235	6278	0.0239	0.5952	0.0459
1-way ∪ 4-way ∪ 5-way	15508	383761	9714	374047	5794	0.0253	0.6264	0.0487
2-way ∪ 3-way ∪ 5-way	15508	43970	9246	34724	6262	0.2103	0.5962	0.3109
2-way ∪ 4-way ∪ 5-way	15508	41266	9730	31536	5778	**0.2358**	**0.6274**	**0.3428**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.
[b]1-way to 4-way overlaps are shown in Table 4.3 such as "All 2-way overlaps".
[c]Total number of transcripts in the benchmark transcriptomes.
[d]Total number of contigs assembled by assembly method.

the inclusion threshold. As shown in Figure 4.2, at the *precision* threshold of 0.2, a better balance between precision and recall was observed. Therefore, the precision threshold of 0.2 was chosen to select the overlap sets to be included. This assembly is called ConSemblEX-select-d. Eight intersections sets (highlighted in cyan in Table 4.3) of varying cardinality make up the ConSemblEX-select-d, for the Col-0 dataset. Eight and nine sets were also chosen for the No-0 and Human dataset, as indicated in Tables A.1 and A.2. We also examined performance of ConSemblEX-3+d, ConSemblEX-4+d, and ConSemblEX-5d, and compared to the performance of ConSemblEX-select-d assembly.

As shown in Table 4.4 (also in Tables A.3 and A.4), the 5-way overlap contributed the most to any assembly set it was a part of. All the unions that included the 5-way

overlap set had either a good *precision* or a good *recall*. The most notable was the "3-way ∪ 5-way", which recorded 0.54 in all the metrics for the Col-0 dataset. The "2-way ∪ 4-way ∪ 5-way" had a high *recall* of 0.63. However, we did not include these assembly sets in the further analysis later as they did not show a significant improvement over their core 5-way overlap set.

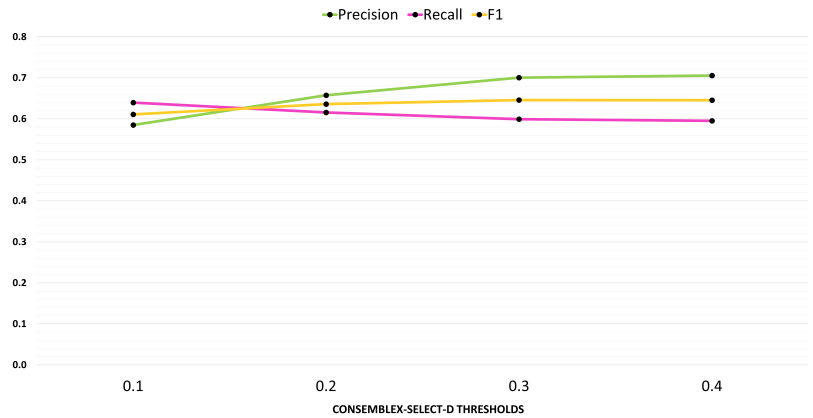
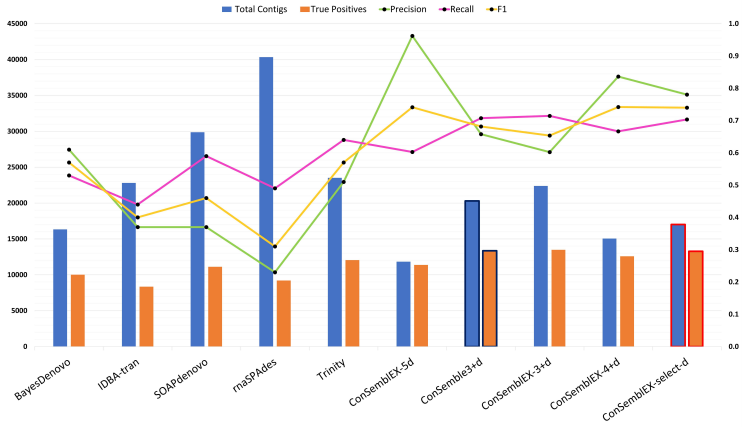


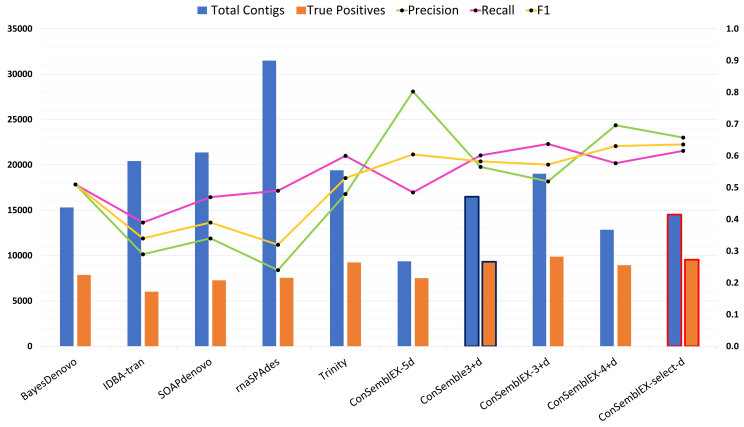
Figure 4.2: Selection of the *precision* thresholds for the *de novo* ConSemblEX-select assembly for the Col-0 dataset. Four thresholds were used to explore the selection of assembly overlap sets for ConSemblEX-select-d based on performance metrics.

4.1.4 Performance of ConSemblEX compared to other *de novo* assembly methods

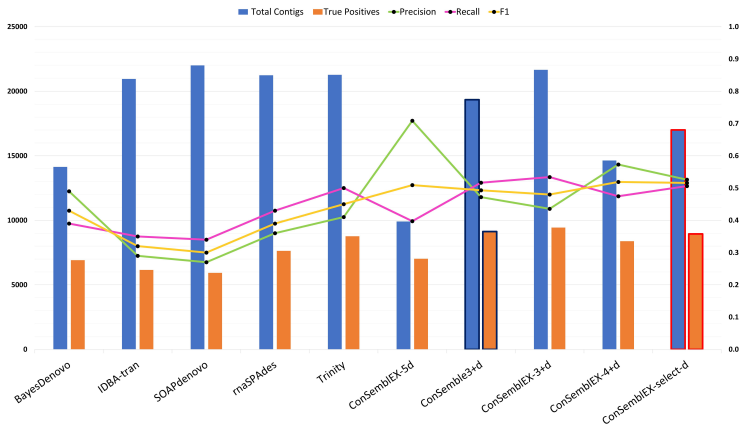
The assembly performance of ConSemblEX-3+d, ConSemblEX-4+d, ConSemblEX-5d, as well as ConSemblEX-select-d were compared against ConSemble and all individual assemblers. The results are summarized in Figure 4.3 and Table A.5. The results were highly dependent on the complexity of the datasets for all the assembly methods.



(a) No-0 dataset



(b) Col-0 dataset



(c) Human dataset

Figure 4.3: Performance comparison among individual *de novo* assemblers, ConSemble3+d, and ConSemblEX assemblies. Numbers of all assembled contigs and correctly assembled contigs are shown in blue and orange bars, respectively. ConSemble and ConSemblEX assemblies were run using multiple k-mers, while the individual assembly methods were run using default k-mer lengths.

In the No-0 dataset, which does not include isoforms, ConSemble3+d recorded *precision* of 0.66 and recall of 0.71 resulting in F_1 of 0.68. It assembled 20,297 contigs in total (11% more than expected), 13,352 of them were correctly assembled, and 6,945 were incorrectly assembled. ConSemblEX-5d had the highest *precision* (0.96), which is expected since the core contigs are more likely to be shared among all assembly methods. On the other hand, it only recovered 60% of transcripts in the benchmark resulting in the highest F_1 of 0.74 (shared with ConSemblEX-4+d and ConSemblEX-select-d). Even with the large disparity between *precision* and *recall*, ConSemblEX-5d outperformed all the individual methods. It also had better *precision* and F_1 than ConSemble3+d but a lower *recall*. ConSemblEX-4+d assembled slightly fewer correct contigs than ConSembleEX-3+d and ConSemble3+d, but more than any of the individual methods. While it assembled 15,050 contigs in total, fewer than the transcripts in the benchmark by 6,292 (33%), 12,583 were correctly assembled, leaving only 2,467 contigs incorrectly assembled. With these numbers, ConSemblEX-4+d recorded the second-highest *precision* (0.84) and the shared highest F_1 (0.74) among all *de novo* assemblers. In contrast, ConSemble3+d and ConSemblEX-3+d reported the highest *recall* (0.71) among all *de novo* assemblers. However, ConSemble3+d recorded a higher *precision* (0.66) and hence, a higher F_1 (0.68) compared to ConSemblEX-3+d (*precision*: 0.60 and F_1: 0.65). The ConSemblEX-select-d had the overall best performance for the No-0 dataset showing a great balance between *precision* (0.78) and *recall* (0.70) with an F_1 score of 0.74. It assembled 13,273 contigs correctly, only 213 lower than ConSemblEX-3+d, the consensus set with the highest *TP*.

All the methods assembled fewer contigs correctly in the more complex Col-0 and Human datasets where isoforms were present. ConSemble3+d displayed another similar performance to ConSemblEX-3+d in both Col-0 (*precision*: 0.57 and F_1: 0.58)

and Human (*precision*: 0.47 and F_1: 0.49), besides correctly assembling 300–500 less contigs than ConSemblEX-3+d. ConSemblEX-5d consistently recorded the highest *precision* (0.80 and 0.71 for Col-0 and Human, respectively) among all methods, and had lower *recall* (0.49 and 0.40 for Col-0 and Human, respectively). ConSembEX-4+d and ConSemblEX-select-d had the best F_1 (0.64 and 0.52 for Col-0 and Human, respectively) among all methods. However, both consensus sets still assembled fewer contigs than expected. ConSemblEX-3+d again reported the highest *recall* (0.64 and 0.53 for Col-0 and Human, respectively) in both datasets with 18–23% more than expected assembled contigs. However, it showed lower *precision* (0.52 and 0.44 for Col-0 and Human, respectively) and F_1 (0.57 and 0.48 for Col-0 and Human, respectively). ConSemblEX-select-d again showed the best balance between *precision* (0.66 and 0.53 for Col-0 and Human, respectively) and *recall* (0.62 and 0.51 for Col-0 and Human, respectively) and, hence, the best performance.

Despite the lower performance in datasets containing isoforms, ConSemblEX-4+d and the ConSemblEX-select-d showed superior performance compared to the original ConSemble. Although ConSemblEX-3+d showed an identical or a slightly lower performance compared to ConSemble3+d, ConSemblEX still produced more correctly assembled contigs and reduced the number of incorrectly assembled contigs. These observations prove that the addition of BayesDenovo improved the overall transcriptome assembly performance.

4.2 Performance of Genome-guided Assembly Methods

4.2.1 Individual genome-guided assembly methods

As described in Voshall et al. [46], the performance of genome-guided assemblers mostly depends on the reference genome used. The assembly performs well if the

reference genome is from the same organism from which the RNA-seq was performed. Using a different reference genome, even if slightly (such as from a different strain of the same species), could lead to poor assembly performance. Such effects of using different reference genomes were confirmed in this study. As shown in Table 4.5, when the same reference genome was used, all the methods performed relatively well across all the datasets. In the simpler No-0 dataset, all but Bayesembler recorded high *recall* ranging from 0.77 to 0.83 (StringTie) and *precision* ranging from 0.71 to 0.75. Considering the No-0 dataset did not include any isoforms and the same reference was used, the methods could have achieved better *precision* and *recall*. Bayesembler also stood out in terms of the total number of contigs recovered with 20% fewer contigs than expected (shown as "Actual"). The rest of the methods recovered an average of 9% more contigs than expected, ranging between 2.2% (Cufflinks) and 13.4% (Scallop).

Table 4.5: Performance of individual genome-guided assemblers using the same reference genome[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	Bayesembler	18875	15172	11201	3971	7674	0.74	0.59	0.66
	Cufflinks	18875	19288	14531	4757	4344	**0.75**	0.77	0.76
	Scallop	18875	21397	15184	6213	3691	0.71	0.80	0.75
	StringTie	18875	21026	15634	5392	3241	0.74	**0.83**	**0.78**
	StringTie2	18875	21196	15137	6059	3738	0.71	0.80	0.76
Col-0	Bayesembler	15508	15143	9158	5985	6350	**0.60**	0.59	0.60
	Cufflinks	15508	15768	8560	7208	6948	0.54	0.55	0.55
	Scallop	15508	18055	10534	7521	4974	0.58	**0.68**	**0.63**
	StringTie	15508	16908	9891	7017	5617	0.58	0.64	0.61
	StringTie2	15508	17722	10034	7688	5474	0.57	0.65	0.60
Human	Bayesembler	17669	13919	7524	6395	10145	**0.54**	0.43	**0.48**
	Cufflinks	17669	14923	7280	7643	10389	0.49	0.41	0.45
	Scallop	17669	26857	8642	18215	9027	0.32	**0.49**	0.39
	StringTie	17669	16311	8094	8217	9575	0.50	0.46	**0.48**
	StringTie2	17669	15850	7388	8462	10281	0.47	0.42	0.44

[a]The **boldfaced** numbers and green highlights show the best score among all the methods for each dataset.

[b]Total number of transcripts in the benchmark transcriptomes.

[c]Total number of contigs assembled by assembly method.

In the Col-0 and Human datasets, all methods had a lower performance with *precision* ranging from 0.54 to 0.60 and from 0.32 to 0.54, respectively. Bayesembler had the highest *precision* in both datasets (0.60 and 0.54 for Col-0 and Human, respectively), making it the most precise, despite consistently producing the least number of contigs (20% fewer than expected). Scallop recorded the highest *recall* (0.68 and 0.49 for Col-0 and Human, respectively) and recovered the most contigs (16.4% and 52% more than expected for Col-0 and Human, respectively). When a different reference genome was used, as shown in Table 4.6, all the assembly methods performed substantially worse, reporting $precision \leq 0.37$, $recall \leq 0.37$, and $F_1 = 0.34$ in all datasets. Scallop barely produced any correct contigs (66/16705), resulting in all the metrics being below 0.01. It is also interesting to note that the older version of StringTie slightly outperformed the newer version, StringTie2, when the same reference genome was used. StringTie had better *precision* and *recall* and hence better F_1 across all datasets. The performance was very close when the different reference genome was used. Regardless, StringTie2 only had a slight edge in the more complex datasets (Col-0 and Human), in which it recovered a slightly higher TP than StringTie. This difference in performance further highlights how unpredictable and challenging transcriptome assembly can be.

4.2.2 ConSemblEX assembly

Distribution of assembly overlaps are shown in Figure 4.4 (also in Figures B.3 and B.4). The 26 distinct overlap sets are summarized in Table 4.7 (also in Tables A.6 and A.7). When the same reference was used, the number of correctly assembled contigs (TP) increased with the number of overlapping assembly methods in an overlap set. All unique sets from individual methods were among overlap sets with the lowest *precision* (average of 0.05) and FP (average of 2884). The *precision* increased

Table 4.6: Performance of individual genome-guided assemblers using a different reference genome[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	Bayesembler	18875	15531	5142	10389	13733	**0.33**	0.27	0.30
	Cufflinks	18875	19938	6510	13428	12365	**0.33**	0.34	**0.34**
	Scallop	18875	22298	6908	15390	11967	0.31	**0.37**	**0.34**
	StringTie	18875	21772	7069	14703	11806	0.32	**0.37**	0.35
	StringTie2	18875	21768	6858	14910	12017	0.32	0.36	**0.34**
Col-0	Bayesembler	15508	15330	4810	10520	10698	**0.31**	0.31	**0.31**
	Cufflinks	15508	16664	4321	12343	11187	0.26	0.28	0.27
	Scallop	15508	16705	66	16639	15442	**0.004**	**0.00**	**0.00**
	StringTie	15508	17791	5074	12717	10434	0.29	0.33	0.3
	StringTie2	15508	18332	5199	13133	10309	0.28	**0.34**	**0.31**
Human	Bayesembler	17669	14610	5413	9197	12256	**0.37**	0.31	**0.34**
	Cufflinks	17669	16258	5296	10962	12373	0.33	0.30	0.31
	Scallop	17669	18778	6132	12646	11537	0.33	**0.35**	**0.34**
	StringTie	17669	18339	5851	12488	11818	0.32	0.33	0.32
	StringTie2	17669	20203	6217	13986	11452	0.31	**0.35**	0.33

[a]The **boldfaced** numbers and green highlights show the best score among all the methods for each dataset. The **boldfaced** numbers and orange highlights show the worst score across the methods for all datasets.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

significantly to an average of 0.21 in the 2-way overlap sets, reducing the *FP* (average of 270) significantly as a result. The 3-way and 4-way overlaps showed further reduction in *FP* and a higher *precision* than the 2-way overlaps and unique sets. The 5-way overlap again had high *precision* (0.95, 0.80, and 0.74, for the No-0, Col-0, and Human datasets, respectively) and a lower *recall* (0.50, 0.40, and 0.26, for the No-0, Col-0, and Human datasets, respectively). Because of its low *recall* the 5-way overlap was not considered the best assembly but remained an important part of the chosen output assembly sets. When the different reference was used, all the unique sets from individual methods had an even lower *precision* (average of 0.02). In the 2-way overlaps, the *precision* increased to an average of 0.24. However, some overlap sets within the 2-way overlaps of the Col-0 dataset only assembled as few as 2 contigs, hence inflating the *precision*. The 3-way and 4-way overlaps showed a similar pattern to the 2-way overlaps, with a further increase in *precision*, and the Col-0 dataset hav-

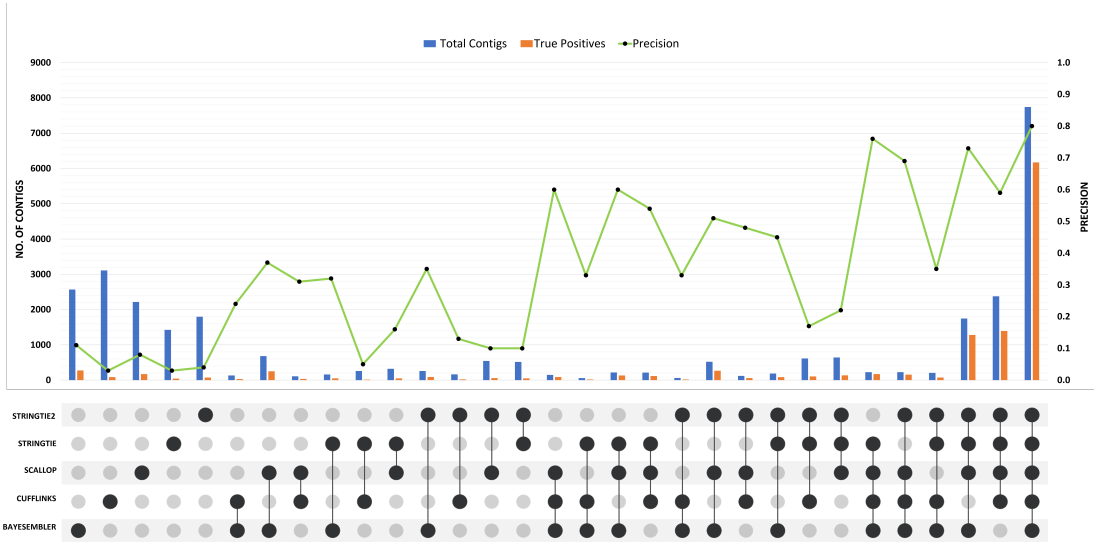
ing a number of overlap sets assembling as few as 3 contigs. The 5-way overlap had a higher *precision* (0.49, 0.0.62, and 0.54, for the No-0, Col-0, and Human datasets, respectively) than all the other overlap sets, but considerably lower than the same reference assembly.

In the unions of overlaps, as shown in Table 4.8, (also in Tables A.8, and A.9), "1-way+" union that includes all assembled contigs recovered a considerable amount of benchmark transcripts indicated by an average *recall* of 0.73 (0.88, 0.75, and 0.57 for No-0, Col-0, and Human, respectively). However, this is at the cost of lower *precision* resulting from the many *FP* introduced by the individual overlap sets. Better assemblies were achieved in the "2-way+" and "3-way+" with increased *precision* and *recall*, and hence a better F_1 score. The pattern extended to the "4-way+" that finally recorded the best *precision* (0.91, 0.74, and 0.68 for No-0, Col-0, and Human, respectively) and *recall* (0.74, 0.60, and 0.40 for No-0, Col-0, and Human, respectively) among all union assembly sets.

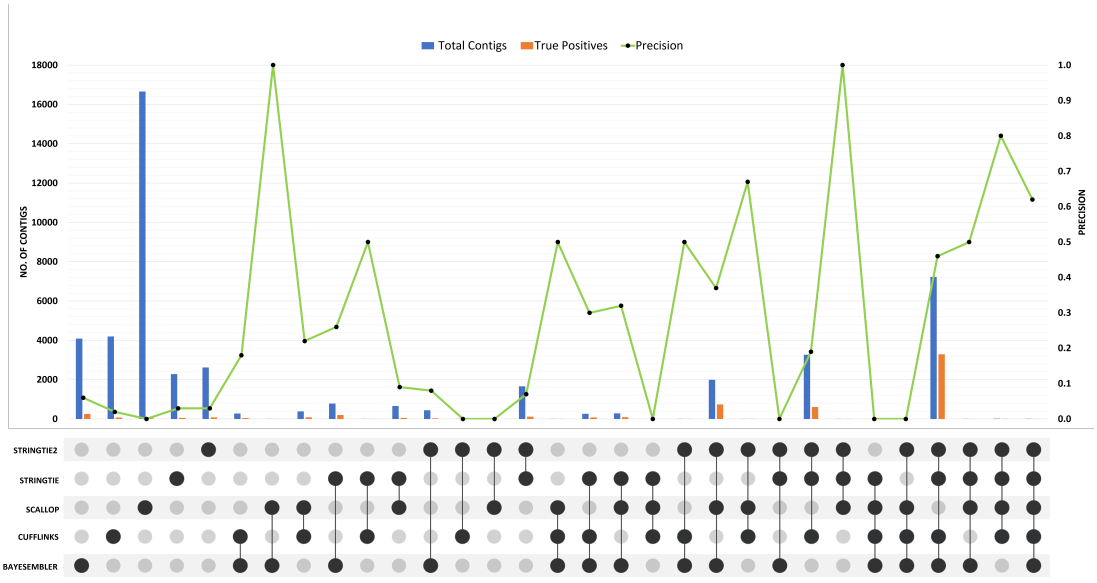
The observations in the unions of overlaps were consistent in the assemblies using a different reference; however, the overall performance was much worse. Therefore, we only focused on the better performing "3-way+" and "4-way+" for further analysis.

4.2.3 Selecting the best ConSemblEX output assembly

As indicated earlier in Section 4.2.2, the overlaps and unions collectively show an improvement in the accuracy of the assembly. However, not every overlap may be contributing to better assemblies. Table 4.7 (as well as Tables A.6 and A.7) details the performance of various assembly overlaps when the same reference genome is used. In the Col-0 dataset, all the unique sets of individual assemblers had similar performance with very low *precision* in both the same and different reference assemblies. The



(a) Col-0 same reference genome



(b) Col-0 different reference genome

Figure 4.4: Distribution of the assembled contig overlaps for the Col-0 dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five genome-guided assemblers. Each assembly was performed using the same reference genome (a) and different reference genome (b). Overlaps among the five methods are indicated with the connected closed circles. Each overlap set is exclusive to each other.

Table 4.7: Genome-guided assembly overlaps using the same reference genome for the Col-0 dataset[a]

Bayesembler	Cufflinks	Scallop	StringTie	StringTie2	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					15508	2567	273	2294	15235	0.1063	0.0176	0.0302
	✘				15508	3109	87	3022	15421	0.0280	0.0056	0.0093
		✘			15508	2217	171	2046	15337	0.0771	0.0110	0.0193
			✘		15508	1423	43	1380	15465	0.0302	0.0028	0.0051
				✘	15508	1795	73	1722	15435	0.0407	0.0047	0.0084
All unique (1-way)					**15508**	**11111**	**647**	**10464**	**14861**	**0.0582**	**0.0417**	**0.0486**
✘	✘				15508	133	32	101	15476	0.2406	0.0021	0.0041
✘		✘			15508	681	249	432	15259	0.3656	0.0161	0.0308
✘			✘		15508	159	51	108	15457	0.3208	0.0033	0.0065
✘				✘	15508	257	89	168	15419	0.3463	0.0057	0.0113
	✘	✘			15508	108	33	75	15475	0.3056	0.0021	0.0042
	✘		✘		15508	256	14	242	15494	0.0547	0.0009	0.0018
	✘			✘	15508	164	21	143	15487	0.1280	0.0014	0.0027
		✘	✘		15508	320	51	269	15457	0.1594	0.0033	0.0064
		✘		✘	15508	544	57	487	15451	0.1048	0.0037	0.0071
			✘	✘	15508	517	51	466	15457	0.0986	0.0033	0.0064
All 2-way overlaps					**15508**	**3139**	**648**	**2491**	**14860**	**0.2064**	**0.0418**	**0.0695**
✘	✘	✘			15508	149	90	59	15418	0.6040	0.0058	0.0115
✘	✘		✘		15508	58	19	39	15489	0.3276	0.0012	0.0024
✘	✘			✘	15508	61	20	41	15488	0.3279	0.0013	0.0026
✘		✘	✘		15508	220	132	88	15376	0.6000	0.0085	0.0168
✘		✘		✘	15508	521	267	254	15241	0.5125	0.0172	0.0333
✘			✘	✘	15508	188	84	104	15424	0.4468	0.0054	0.0107
	✘	✘	✘		15508	214	116	98	15392	0.5421	0.0075	0.0148
	✘	✘		✘	15508	122	58	64	15450	0.4754	0.0037	0.0074
	✘		✘	✘	15508	613	105	508	15403	0.1713	0.0068	0.0130
		✘	✘	✘	15508	640	138	502	15370	0.2156	0.0089	0.0171
All 3-way overlaps					**15508**	**2786**	**1029**	**1757**	**14479**	**0.3693**	**0.0664**	**0.1125**
✘	✘	✘	✘		15508	227	173	54	15335	0.7621	0.0112	0.0220
✘	✘	✘		✘	15508	227	157	70	15351	0.6916	0.0101	0.0200
✘	✘		✘	✘	15508	208	72	136	15436	0.3462	0.0046	0.0092
✘		✘	✘	✘	15508	1746	1279	467	14229	0.7325	0.0825	0.1483
	✘	✘	✘	✘	15508	2378	1392	986	14116	0.5854	0.0898	0.1557
All 4-way overlaps					**15508**	**4786**	**3073**	**1713**	**12435**	**0.6421**	**0.1982**	**0.3028**
✘	✘	✘	✘	✘	15508	**7741**	**6171**	**1570**	**9337**	**0.7972**	**0.3979**	**0.5309**

[a]The highlights show the overlaps included in the ConSemblEX-select-g assembly. The **boldfaced** numbers show the best score among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

differing overlap performances became more visible in the 2-way overlaps, where 40% of overlaps had *precision* ≥ 0.3 while the rest had ≤ 0.15, indicating that not all overlaps could be effective. The 3-way and 4-way overlaps showed the same pattern with an even more significant number of them reporting *precision* ≥ 0.3. As we did for *de novo* assemblies, four assembly sets were constructed using *precision* as the inclusion thresholds in both the same and different reference assemblies. The *precision* of 0.3 was chosen for the threshold for the same reference assembly because, as shown in Figure 4.5 it showed a better balance between *precision* and *recall* and hence the better F_1. In the different reference assembly, the *precision* of 0.2 was chosen as the threshold to cover more overlaps as most high scoring overlaps only

Table 4.8: Genome-guided assembly using various union sets among overlapping contig sets and using the same reference genome for the Col-0 dataset[a]

Assembly sets[b]	Actual[c]	Total[d]	*TP*	*FP*	*FN*	Precision	Recall	F_1
1-way+	15508	29563	11568	17995	3940	0.3913	**0.7459**	0.5133
2-way+	15508	18452	10921	7531	4587	0.5919	0.7042	0.6432
3-way+	15508	15313	10273	5040	5235	0.6709	0.6624	**0.6666**
4-way+	15508	12527	9244	3283	6264	**0.7379**	0.5961	0.6595
1-way ∪ 2-way	15508	14250	1295	12955	14213	0.0909	0.0835	0.0870
2-way ∪ 3-way	15508	5925	1677	4248	13831	0.2830	0.1081	0.1565
3-way ∪ 4-way	15508	7572	4102	3470	11406	0.5417	0.2645	0.3555
1-way ∪ 4-way	15508	15897	3720	12177	11788	0.2340	0.2399	0.2369
2-way ∪ 4-way	15508	7925	3721	4204	11787	0.4695	0.2399	0.3176
1-way ∪ 5-way	15508	18852	6818	12034	8690	0.3617	0.4396	0.3969
2-way ∪ 5-way	15508	10880	6819	4061	8689	0.6267	0.4397	0.5168
3-way ∪ 5-way	15508	10527	7200	3327	8308	**0.6840**	**0.4643**	**0.5531**
1-way ∪ 2-way ∪ 3-way	15508	17036	2324	14712	13184	0.1364	0.1499	0.1428
1-way ∪ 2-way ∪ 4-way	15508	19036	4368	14668	11140	0.2295	0.2817	0.2529
1-way ∪ 3-way ∪ 4-way	15508	18683	4749	13934	10759	0.2542	0.3062	0.2778
1-way ∪ 2-way ∪ 5-way	15508	21991	7466	14525	8042	0.3395	0.4814	0.3982
1-way ∪ 3-way ∪ 5-way	15508	21638	7847	13791	7661	0.3626	0.5060	0.4225
1-way ∪ 4-way ∪ 5-way	15508	23638	9891	13747	5617	0.4184	0.6378	0.5053
2-way ∪ 3-way ∪ 5-way	15508	13666	7848	5818	7660	0.5743	0.5061	0.5380
2-way ∪ 4-way ∪ 5-way	15508	15666	9892	5774	5616	**0.6314**	**0.6379**	**0.6346**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.
[b]1-way to 4-way overlaps are shown in Table 4.7 such as "All 2-way overlaps".
[c]Total number of transcripts in the benchmark transcriptomes.
[d]Total number of contigs assembled by assembly method.

assembled a few contigs. The overlaps that are included in ConSemblEX-select-g assembly for the Col-0 dataset using the same reference are shown in Table 4.7 with cyan highlights. Those chosen in the No-0 and Human datasets are shown in Tables A.6 and A.7, respectively. As shown in Table 4.8 (also in Tables A.8 and A.9), all the unions that included the 5-way overlap had a good *precision* or a good *recall*.

4.2.4 Performance of ConSemblEX compared to other genome-guided assembly methods

Figures 4.6 and 4.7 summarize the performance comparisons (see Tables A.10 and A.11 for details) among ConSemble, ConSemblEX, and all individual assembly methods using the same and different reference genomes. In the simpler No-0 dataset,

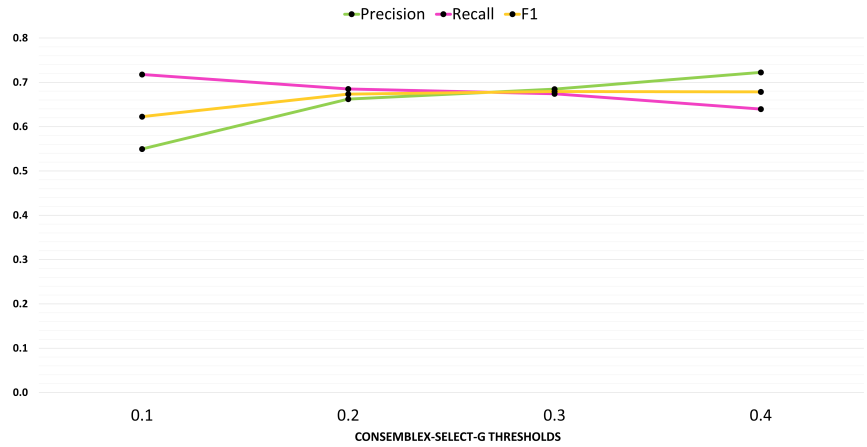
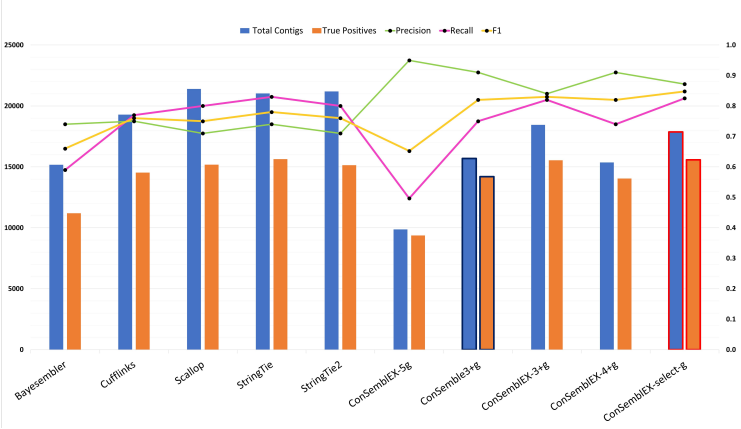


Figure 4.5: Selection of the *precision* thresholds for genome-guided ConSemblEX-select assembly using the same reference for the Col-0 dataset. Four *precision* thresholds were used to explore ConSemblEX-select-g based on their performance metrics.

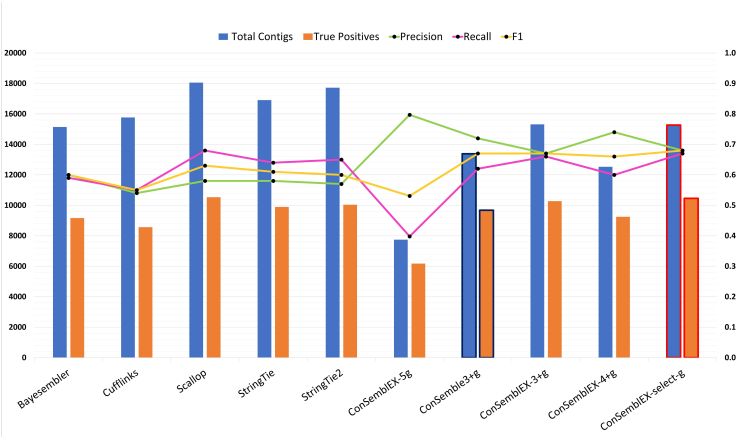
while using the same reference genome, ConSemble3+g recorded a high *precision* (0.91) and *recall* (0.75) resulting in high F_1 (0.82). ConSemblEX-5g recorded the highest *precision* (0.95) but a lower *recall* (0.50) than ConSemble3+g, indicating that only half of the benchmark transcripts were recovered. ConSemblEX-4+g had virtually the same performance. ConSemblEX-3+g had a slightly lower performance (*precision*: 0.84, *recall*: 0.82 and F_1: 0.83) compared to ConSemblEX-4+g and ConSemble3+g, however, it recovered ∼1300 more contigs correctly than both assembly sets. Its total number of assembled contigs was also very close (only 424 fewer) than the expected number in the benchmark (18,875 transcripts). Despite recovering the bigger number of correctly assembled contigs amongst the consensus-based assembly sets, the ConSemblEX-select-g recorded a slightly lower *precision* (0.87) than both ConSemblEX-4+g and ConSemble3+g, but a higher *recall*, and hence slightly higher F_1 (0.85). ConSemblEX-select-g also recovered ∼1376 correctly assembled contigs than ConSemble3+g and ConSemblEX-4+g.

Similar to what we observed in individual assembly methods, the performance of

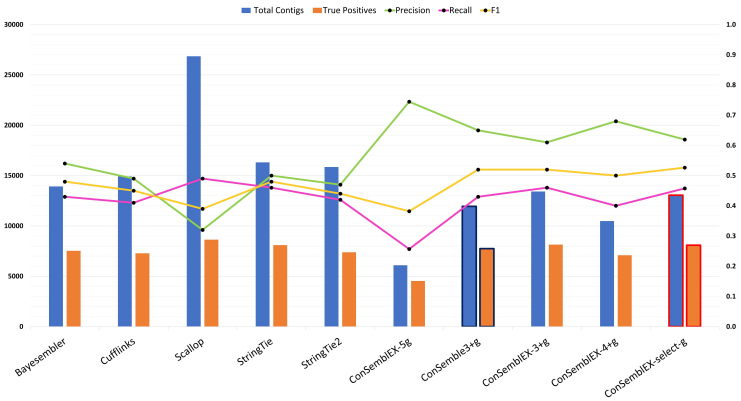
ConSemblEX was lower in the more complex Col-0 and Human datasets. ConSemblEX-4+g and ConSemble3+g performances were again close, with a max of 0.3 separating their metrics. ConSemblEX-5g recorded the highest *precision* (0.80 and 0.74 for Col-0 and Human, respectively) and the lowest *recall* (0.40 and 0.26 for Col-0 and Human, respectively) among all methods. ConSeblEX-3+g recovered ∼1000 more correct contigs than ConSemblEX-4+g and ConSemble3+g leading to a better *recall* (0.66 and 0.46 for Col-0 and Human, respectively) than both. Finally, ConSemblEX-select-g recorded a slightly lower *precision* than ConSemblEX-4+g and ConSemble3+g but assembled ∼1000 more contigs correctly and had higher *recall* (0.67).



(a) No-0 dataset

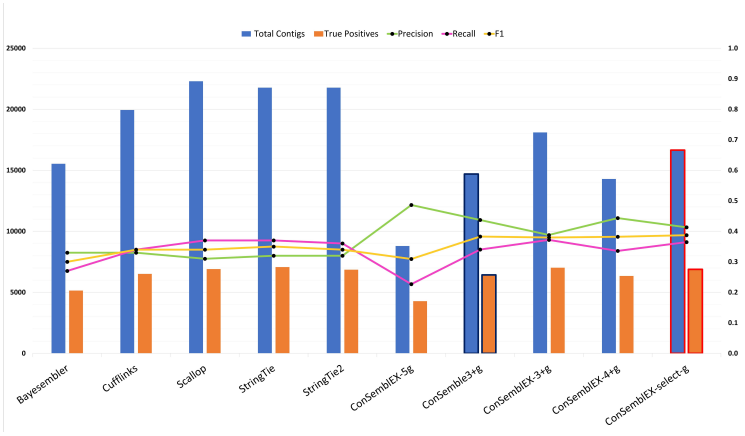


(b) Col-0 dataset

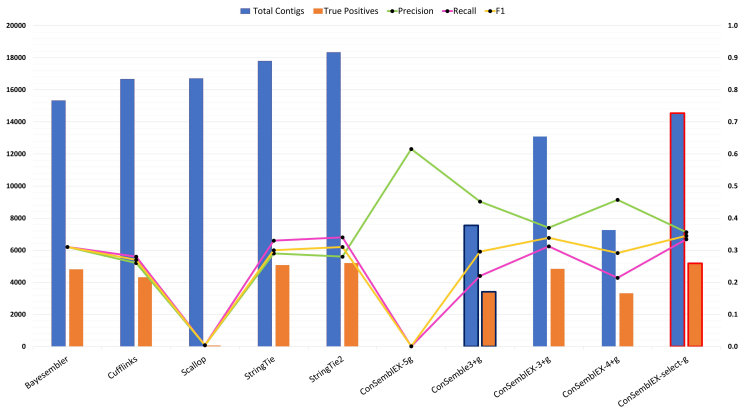


(c) Human dataset

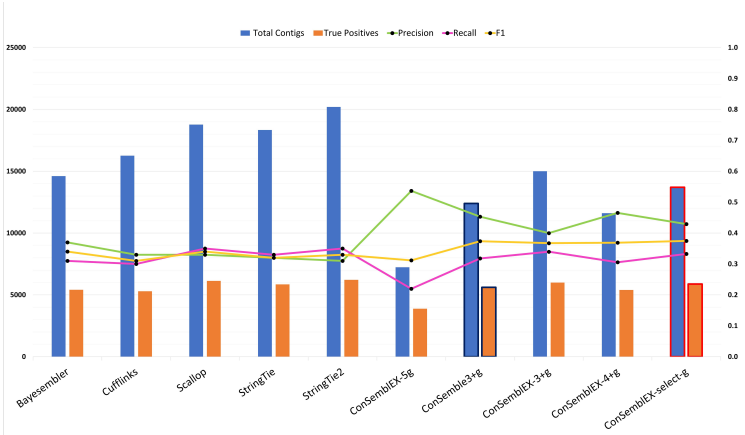
Figure 4.6: Performance comparisons among individual genome-guided assemblers, ConSemble3+g, and ConSemblEX assemblies using the same reference genome. Numbers of total assembled contigs and incorrectly assembled contigs are shown in blue and orange bars, respectively.



(a) No-0 dataset



(b) Col-0 dataset



(c) Human dataset

Figure 4.7: Performance comparisons among individual genome-guided assemblers, ConSemble3+g, and ConSemblEX assemblies using a different reference genome. Numbers of total assembled contigs and incorrectly assembled contigs are shown in blue and orange bars, respectively.

As reported before, using a different reference decreased the assembly performance significantly in all the datasets (see Figure 4.7). All the methods had low $precision \leq 0.49$, $recall \leq 0.37$ resulting in low $F_1 \leq 0.39$. From these observations, it is clear that using the *de novo* approach would be a better option in cases where a good reference does not exist. Further, it would be interesting to investigate whether the assembled contigs, when using a different reference, can be used with the *de novo* approach to improve the performance of the assembly.

The consensus-based assembly methods collectively outperformed the individual assembly methods, highlighting the advantage of using a consensus assembly method even in the genome-guided assembly. When the realistic output, ConSemblEX-4+g, was considered, the performance of ConSemblEX was identical to or very close to that of ConSemble. Nonetheless, ConSemblEX-select-g recovered more correctly assembled contigs, which reduced the *FP* overall.

4.3 Discussion

Assembling an accurate and complete transcriptome is a challenging task. Many assembly methods, be it *de novo* or genome-guided, only assemble part of the transcriptome correctly. In *de novo* assembly, we observed that most of the individual methods generated far more contigs than expected and yet only accurately assembled $\leq 60\%$ of the transcripts. In genome-guided assembly, even when a good reference was available, the assembly methods only managed to recover $\leq 80\%$of the transcripts. The performance was even worse when a different reference was used in the assembly. Regardless of the triviality of the difference (e.g., versions of the human genome), the assembly was significantly impacted. The genome-guided assembly methods with different reference genomes only recovered $\leq 37\%$ of the benchmark

transcripts. In addition, the assembly results were heavily dependent on the complexity of the dataset, which can cause inconsistency in the assemblies produced by various methods.

Ensemble methods can overcome some of the limitations of *de novo* and genome-guided methods by selecting consensus contigs from various individual assemblers. The likelihood of accurately assembled contigs significantly increases when recovered in more than one assembler. ConSemble uses consensus information among individual assembly methods (both *de novo* and genome-guided) to produce more accurate consensus assemblies that outperform individual assemblers. Nonetheless, it has some limitations highlighted in Section 1.2. ConSemblEX applies the core principles of ConSemble assembly, and modifies a few things to address its limitations.

In the *de novo* approach, ConSemblEX incorporated a new assembly method, BayesDenovo, increasing the number of assemblers. Besides the union overlaps for the final assembly sets, ConSemblEX provided ConSemblEX-select-d by selecting overlap sets with *precision* ≥ 0.2. ConSemblEX-select-d delivered a better performance with higher *precision* (+9%) and F_1 (+5%) than ConSemble3+d across all datasets tested. The closeness in *recall* indicated that the higher *precision* and F_1 shown with ConSemblEX-select-d were mainly due to lower *FP* leading also to smaller numbers of total contigs. In fact, while ConSemblEX-select-d and ConSemble3+d only had a difference of $\sim$150 in their *TP* across all datasets, ConSemblEX-select-d had a lower *FP*, an average of 30%, than ConSemble3+d across all datasets. While using the *precision* threshold to choose assembly sets for ConSemblEX-select-d worked well with simulated data, it can not be done where no benchmark transcriptome exists. In this case, simpler union sets such as ConSemblEX-3+d, ConSemblEX-4+d, and ConSemblEX-5d, can be considered. Compared to ConSemble3+d, ConSemblEX-4+d also recorded higher *precision* (+14%) and F_1 (+5%) but a lower *recall* (-3%).

ConSemblEX-4+d also assembled fewer *TP* than ConSemble3+d (on average, 630 fewer). Furthermore, ConSemblEX-4+d significantly reduced *FP* compared to ConSemble3+d, (from ∼8100 to ∼4200, equivalent to 49% reduction). We observed that the addition of BayesDenovo improved the overall assembly. Therefore, ConSemblEX generally performed better than ConSemble.

In the genome-guided approach, the older version of StringTie was added to the ConSemblEX assembly pipeline in addition to the four methods initially used in ConSemble. ConSemblEX-select-g was defined differently for the same-reference and different-reference assemblies. Regardless of ConSemblEX-select-g reporting a lower *precision* (-4%) than ConSemble3+g, it assembled 830 more contigs correctly and hence a higher *recall* (+5%) and slightly higher F_1 (+1.7%) across all datasets. The ConSemblEX assemblies (ConSemblEX-3+g, ConSemblEX-4+g, ConSemblEX-5g) and ConSemble had similar performances across all datasets overall, with only minor (1%) differences between *precision* and *recall* in the No-0 and Human datasets. With the different reference genomes assemblies, both ConSemblEX and ConSemble performed badly even with the carefully constructed ConSemblEX-select-g. In the Col-0 dataset, we saw that ConSemblEX-5g could not assemble a significant number of contigs, with only 13 contigs assembled. Thus, when a good reference does not exist, the *de novo* pipeline would be the better option, as we observed that it outperformed the genome-guided pipeline. Nonetheless, even when appropriate reference genomes are not available, combining the best *de novo* and genome-guided assemblies would be interesting and such approach needs to be investigated whether the contigs assembled by both methods could lead to a more accurate assembly.

Although ConSemblEX-select provides a better assembly, defining the metrics to select overlap sets is the major challenge. Reference-free evaluation tools such as TransRate [43] would be ideal for calculating statistics of various overlaps and using

them to determine the performance of overlaps. However, unlike the trivial assessment using *precision*, the tools would introduce new patterns that need more understanding to define a better overlap. Furthermore, various tools define assembly performance differently. Therefore, a dependency on the selected evaluation tool is likely to exist, leading to potential inconsistency in the assembly results. Using different evaluation tools together and applying a machine learning model to understand the varying patterns would help overcome the possible dependency problem.

Chapter 5

Conclusions and Future Work

We presented ConSemblEX, a consensus-based transcriptome assembly approach that is an extension of ConSemble [46]. It expands on the number of individual assembly methods used in the consensus assembly, provides details about the distribution of the assembled contigs, and provides alternative assemblies as outputs. We tested the assembly performance of ConSemblEX using five assembly methods, each of *de novo* and genome-guided, across three datasets of varying complexity. In the *de novo* pipeline, ConSemblEX-4+d performs better than ConSemble3+d increasing the *precision* by 14% and F_1 by 5%, despite the 3% lower *recall*. It also significantly reduces the *FP* by 49% resulting in a more accurate assembly. ConSemblEX also provides a ConSemblEX-select-d, which only consists of the better-performing overlap sets. Compared to Consemble3+d, ConSemblEX-select-d increases the *precision* by 9% and F_1 by 5% while maintaining the high *recall*. It also reduces the *FP* by 30%. Therefore, we showed that the extra overlap performance information provided by ConSemblEX could be used to construct an even more accurate assembly. In the genome-guided pipeline, ConSemblEX performs identically to ConSemble using the same and different reference genomes. Comparing ConSemblEX-select-g and ConSemble, precision is reduced by 4%, while recall and F_1 are increased by 5% and 1.7%, respectively.

In the future, we can define how overlap sets are selected to construct the better assembly by applying machine learning model to ConSemblEX. We plan to incorporate, for example, TransRate and DETONATE [30] evaluation tools to select the best-performing overlaps sets. TransRate provides reference-free evaluation metrics and would be more useful in the *de novo* pipeline. DETONATE provides both reference-free and reference-based metrics and can be used in both *de novo* and genome-guided pipelines. With TransRate and DETONATE, we do not need a benchmark transcriptome to evaluate the performance of the overlaps and final assembly sets. We also plan to investigate whether we can use the assembled contigs when the appropriate reference genome is not available for genome-guided assembly by analyzing them together with *de novo* assembly sets.

Appendix A

Table A.1: *De novo* assembly overlaps using multiple k-mers for the No-0 dataset[a]

BayesDenovo	IDBA-Trans	SOAPdenovo	rnaSPAdes	Trinity	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					18875	20941	18	20923	18857	0.0009	0.0010	0.0009
	✘				18875	78613	157	78456	18718	0.0020	0.0083	0.0032
		✘			18875	167709	74	167635	18801	0.0004	0.0039	0.0008
			✘		18875	215181	426	214755	18449	0.0020	0.0226	0.0036
				✘	18875	55227	30	55197	18845	0.0005	0.0016	0.0008
All unique (1-way)					**18875**	**537671**	**705**	**536966**	**18170**	**0.0013**	**0.0374**	**0.0025**
✘	✘				18875	1698	8	1690	18867	0.0047	0.0004	0.0008
✘		✘			18875	665	8	657	18867	0.0120	0.0004	0.0008
✘			✘		18875	831	83	748	18792	0.0999	0.0044	0.0084
✘				✘	18875	962	5	957	18870	0.0052	0.0003	0.0005
	✘	✘			18875	2795	81	2714	18794	0.0290	0.0043	0.0075
	✘		✘		18875	3165	223	2942	18652	0.0705	0.0118	0.0202
	✘			✘	18875	1651	32	1619	18843	0.0194	0.0017	0.0031
		✘	✘		18875	14753	100	14653	18775	0.0068	0.0053	0.0059
		✘		✘	18875	3121	32	3089	18843	0.0103	0.0017	0.0029
			✘	✘	18875	4238	25	4213	18850	0.0059	0.0013	0.0022
All 2-way overlaps					**18875**	**33879**	**597**	**33282**	**18278**	**0.0176**	**0.0316**	**0.0226**
✘	✘	✘			18875	232	8	224	18867	0.0345	0.0004	0.0008
✘	✘		✘		18875	325	57	268	18818	0.1754	0.0030	0.0059
✘	✘			✘	18875	445	18	427	18857	0.0404	0.0010	0.0019
✘		✘	✘		18875	441	22	419	18853	0.0499	0.0012	0.0023
✘		✘		✘	18875	279	12	267	18863	0.0430	0.0006	0.0013
✘			✘	✘	18875	374	17	357	18858	0.0455	0.0009	0.0018
	✘	✘	✘		18875	1635	633	1002	18242	0.3872	0.0335	0.0617
	✘	✘		✘	18875	787	37	750	18838	0.0470	0.0020	0.0038
	✘		✘	✘	18875	640	42	598	18833	0.0656	0.0022	0.0043
		✘	✘	✘	18875	2185	57	2128	18818	0.0261	0.0030	0.0054
All 3-way overlaps					**18875**	**7343**	**903**	**6440**	**17972**	**0.1230**	**0.0478**	**0.0689**
✘	✘	✘	✘		18875	272	107	165	18768	0.3934	0.0057	0.0112
✘	✘	✘		✘	18875	268	96	172	18779	0.3582	0.0051	0.0100
✘	✘		✘	✘	18875	246	32	214	18843	0.1301	0.0017	0.0033
✘		✘	✘	✘	18875	405	80	325	18795	0.1975	0.0042	0.0083
	✘	✘	✘	✘	18875	2036	894	1142	17981	0.4391	0.0474	0.0855
All 4-way overlaps					**18875**	**3227**	**1209**	**2018**	**17666**	**0.3747**	**0.0641**	**0.1094**
✘	**✘**	**✘**	**✘**	**✘**	**18875**	**11823**	**11374**	**449**	**7501**	**0.9620**	**0.6026**	**0.7410**

[a]The highlights show the overlaps included in the ConSemblEX-select-d assembly. The **boldfaced** numbers show the best score among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

Table A.2: *De novo* assembly overlaps using multiple k-mers for the Human dataset[a]

BayesDenovo	IDBA-Trans	SOAPdenovo	rnaSPAdes	Trinity	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					17669	11186	122	11064	17547	0.0109	0.0069	0.0085
	✘				17669	26486	132	26354	17537	0.0050	0.0075	0.0060
		✘			17669	90739	160	90579	17509	0.0018	0.0091	0.0030
			✘		17669	213288	420	212868	17249	0.0020	0.0238	0.0036
				✘	17669	16211	171	16040	17498	0.0105	0.0097	0.0101
All unique (1-way)					**17669**	**357910**	**1005**	**356905**	**16664**	**0.0028**	**0.0569**	**0.0054**
✘	✘				17669	1212	42	1170	17627	0.0347	0.0024	0.0044
✘		✘			17669	600	21	579	17648	0.0350	0.0012	0.0023
✘			✘		17669	726	88	638	17581	0.1212	0.0050	0.0096
✘				✘	17669	426	42	384	17627	0.0986	0.0024	0.0046
	✘	✘			17669	2232	17	2215	17652	0.0076	0.0010	0.0017
	✘		✘		17669	2595	89	2506	17580	0.0343	0.0050	0.0088
	✘			✘	17669	1137	132	1005	17537	0.1161	0.0075	0.0140
		✘	✘		17669	10719	336	10383	17333	0.0313	0.0190	0.0237
		✘		✘	17669	1655	24	1631	17645	0.0145	0.0014	0.0025
			✘	✘	17669	2963	156	2807	17513	0.0526	0.0088	0.0151
All 2-way overlaps					**17669**	**24265**	**947**	**23318**	**16722**	**0.0390**	**0.0536**	**0.0452**
✘	✘	✘			17669	488	7	481	17662	0.0143	0.0004	0.0008
✘	✘		✘		17669	497	26	471	17643	0.0523	0.0015	0.0029
✘	✘			✘	17669	510	113	397	17556	0.2216	0.0064	0.0124
✘		✘	✘		17669	330	47	283	17622	0.1424	0.0027	0.0052
✘		✘		✘	17669	141	17	124	17652	0.1206	0.0010	0.0019
✘			✘	✘	17669	345	99	246	17570	0.2870	0.0056	0.0110
	✘	✘	✘		17669	1307	204	1103	17465	0.1561	0.0115	0.0215
	✘	✘		✘	17669	693	52	641	17617	0.0750	0.0029	0.0057
	✘		✘	✘	17669	903	218	685	17451	0.2414	0.0123	0.0235
		✘	✘	✘	17669	1807	268	1539	17401	0.1483	0.0152	0.0275
All 3-way overlaps					**17669**	**7021**	**1051**	**5970**	**16618**	**0.1497**	**0.0595**	**0.0851**
✘	✘	✘	✘		17669	703	75	628	17594	0.1067	0.0042	0.0082
✘	✘	✘		✘	17669	476	121	355	17548	0.2542	0.0068	0.0133
✘	✘		✘	✘	17669	843	221	622	17448	0.2622	0.0125	0.0239
✘		✘	✘	✘	17669	331	117	214	17552	0.3535	0.0066	0.0130
	✘	✘	✘	✘	17669	2378	831	1547	16838	0.3495	0.0470	0.0829
All 4-way overlaps					**17669**	**4731**	**1365**	**3366**	**16304**	**0.2885**	**0.0773**	**0.1219**
✘	✘	✘	✘	✘	**17669**	**9908**	**7021**	**2887**	**10648**	**0.7086**	**0.3974**	**0.5092**

[a]The highlights show the overlaps included in the ConSemblEX-select-d assembly. The **boldfaced** numbers show the best score among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

Table A.3: *De novo* assembly using various union sets among overlapping contig sets and multiple k-mers for the No-0 dataset[a]

Assembly sets[b]	Actual[c]	Total[d]	*TP*	*FP*	*FN*	Precision	Recall	F_1
1-way+	18875	593943	14788	579155	4087	0.0249	**0.7835**	0.0483
2-way+	18875	56272	14083	42189	4792	0.2503	0.7461	0.3748
3-way+	18875	22393	13486	8907	5389	0.6022	0.7145	0.6536
4-way+	18875	15050	12583	2467	6292	**0.8361**	0.6666	**0.7418**
1-way ∪ 2-way	18875	571550	1302	570248	17573	0.0023	0.0690	0.0044
2-way ∪ 3-way	18875	41222	1500	39722	17375	0.0364	0.0795	0.0499
3-way ∪ 4-way	18875	10570	2112	8458	16763	0.1998	0.1119	0.1435
1-way ∪ 4-way	18875	540898	1914	538984	16961	0.0035	0.1014	0.0068
2-way ∪ 4-way	18875	37106	1806	35300	17069	0.0487	0.0957	0.0645
1-way ∪ 5-way	18875	549494	12079	537415	6796	0.0220	0.6399	0.0425
2-way ∪ 5-way	18875	45702	11971	33731	6904	0.2619	0.6342	0.3708
3-way ∪ 5-way	18875	19166	12277	6889	6598	**0.6406**	**0.6504**	**0.6455**
1-way ∪ 2-way ∪ 3-way	18875	578893	2205	576688	16670	0.0038	0.1168	0.0074
1-way ∪ 2-way ∪ 4-way	18875	574777	2511	572266	16364	0.0044	0.1330	0.0085
1-way ∪ 3-way ∪ 4-way	18875	548241	2817	545424	16058	0.0051	0.1492	0.0099
1-way ∪ 2-way ∪ 5-way	18875	583373	12676	570697	6199	0.0217	0.6716	0.0421
1-way ∪ 3-way ∪ 5-way	18875	556837	12982	543855	5893	0.0233	0.6878	0.0451
1-way ∪ 4-way ∪ 5-way	18875	552721	13288	539433	5587	0.0240	**0.7040**	0.0465
2-way ∪ 3-way ∪ 5-way	18875	53045	12874	40171	6001	0.2427	0.6821	0.3580
2-way ∪ 4-way ∪ 5-way	18875	348929	13180	35749	5695	**0.2694**	0.6983	**0.3888**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.

[b]1-way to 4-way overlaps are shown in Table A.1 such as "All 2-way overlap".

[c]Total number of transcripts in the benchmark transcriptomes.

[d]Total number of contigs assembled by assembly method.

Table A.4: *De novo* assembly using various union sets among overlapping contig sets and multiple k-mers for the Human dataset[a]

Assembly sets[b]	Actual[c]	Total[d]	*TP*	*FP*	*FN*	Precision	Recall	F_1
1-way+	17669	403835	11389	392446	6280	0.0282	**0.6446**	0.0540
2-way+	45925	17669	10384	35541	7285	0.2261	0.5877	0.3266
3-way+	17669	21660	9437	12223	8232	0.4357	0.5341	0.4799
4-way+	17669	14639	8386	6253	9283	**0.5729**	0.4746	**0.5191**
1-way ∪ 2-way	17669	382175	1952	380223	15717	0.0051	0.1105	0.0098
2-way ∪ 3-way	17669	31286	1998	29288	15671	0.0639	0.1131	0.0816
3-way ∪ 4-way	17669	11752	2416	9336	15253	0.2056	0.1367	0.1642
1-way ∪ 4-way	17669	362641	2370	360271	15299	0.0065	0.1341	0.0125
2-way ∪ 4-way	17669	28996	2312	26684	15357	0.0797	0.1309	0.0991
1-way ∪ 5-way	17669	367818	8026	359792	9643	0.0218	0.4542	0.0416
2-way ∪ 5-way	17669	34173	7968	26205	9701	0.2332	0.4510	0.3074
3-way ∪ 5-way	17669	16929	8072	8857	9597	**0.4768**	**0.4568**	**0.4666**
1-way ∪ 2-way ∪ 3-way	17669	389196	3003	386193	14666	0.0077	0.1700	0.0148
1-way ∪ 2-way ∪ 4-way	17669	386906	3317	383589	14352	0.0086	0.1877	0.0164
1-way ∪ 3-way ∪ 4-way	17669	369662	3421	366241	14248	0.0093	0.1936	0.0177
1-way ∪ 2-way ∪ 5-way	17669	392083	8973	383110	8696	0.0229	0.5078	0.0438
1-way ∪ 3-way ∪ 5-way	17669	374839	9077	365762	8592	0.0242	0.5137	0.0463
1-way ∪ 4-way ∪ 5-way	17669	372549	9391	363158	8278	0.0252	**0.5315**	0.0481
2-way ∪ 3-way ∪ 5-way	17669	41194	9019	32175	8650	0.2189	0.5104	0.3064
2-way ∪ 4-way ∪ 5-way	17669	38904	9333	29571	8336	**0.2399**	0.5282	**0.3299**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.

[b]1-way to 4-way overlaps are shown in Table A.2 such as "All 2-way overlaps".

[c]Total number of transcripts in the benchmark transcriptomes.

[d]Total number of contigs assembled by assembly method.

Table A.5: Performance of ConSemblEX compared to ConSemble and individual *de novo* assemblers[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	BayesDenovo	18875	16332	10022	6310	8853	0.61	0.53	0.57
	IDBA-tran	18875	22802	8344	14458	10531	0.37	0.44	0.40
	SOAPdenovo-trans	18875	29876	11119	18757	7756	0.37	0.59	0.46
	rnaSPAdes	18875	40333	9206	31127	9669	0.23	0.49	0.31
	Trinity	18875	23523	12059	11464	6816	0.51	0.64	0.57
	ConSemblEX-5d	18875	11823	11374	449	7501	**0.96**	0.60	**0.74**
	ConSemble3+d	18875	20297	13352	6945	5523	0.66	**0.71**	0.68
	ConSemblEX-3+d	18875	22393	13486	8907	5389	0.60	**0.71**	0.65
	ConSemblEX-4+d	18875	15050	12583	2467	6292	0.84	0.67	**0.74**
	ConSemblEX-select-d	18875	17010	13273	3737	5602	0.78	0.70	**0.74**
Col-0	BayesDenovo	15508	15316	7876	7440	7632	0.51	0.51	0.51
	IDBA-tran	15508	20430	6021	14409	9487	0.29	0.39	0.34
	SOAPdenovo-trans	15508	21371	7281	14090	8227	0.34	0.47	0.39
	rnaSPAdes	15508	31494	7556	23938	7952	0.24	0.49	0.32
	Trinity	15508	19417	9255	10162	6253	0.48	0.6	0.53
	ConSemblEX-5d	15508	9369	7521	1848	7987	**0.80**	0.49	0.60
	ConSemble3+d	15508	16500	9326	7174	6182	0.57	0.60	0.58
	ConSemblEX-3+d	15508	19027	9885	9142	5623	0.52	**0.64**	0.58
	ConSemblEX-4+d	15508	12846	8945	3901	6563	0.70	0.58	0.63
	ConSemblEX-select-d	15508	14521	9549	4972	5959	0.66	0.62	**0.64**
Human	BayesDenovo	17669	14139	6914	7225	10755	0.49	0.39	0.43
	IDBA-tran	17669	20960	6154	14806	11515	0.29	0.35	0.32
	SOAPdenovo-trans	17669	22005	5933	16072	11736	0.27	0.34	0.3
	rnaSPAdes	17669	21244	7637	13607	10032	0.36	0.43	0.39
	Trinity	17669	21279	8765	12514	8904	0.41	0.50	0.45
	ConSemblEX-5d	17669	9908	7021	2887	10648	**0.71**	0.40	0.51
	ConSemble3+d	17669	19349	9128	10221	8541	0.47	0.52	0.49
	ConSemblEX-3+d	17669	21660	9437	12223	8232	0.44	**0.53**	0.48
	ConSemblEX-4+d	17669	14639	8386	6253	9283	0.57	0.47	0.51
	ConSemblEX-select-d	17669	17001	8945	8056	8724	0.53	0.51	**0.52**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each dataset.

[b]Total number of transcripts in the benchmark transcriptomes.

[c]Total number of contigs assembled by assembly method.

Table A.6: Genome-guided assembly overlaps using the same reference genome for the No-0 dataset[a]

Bayesembler	Cufflinks	Scallop	StringTie	StringTie2	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					18875	2487	32	2455	18843	0.0129	0.0017	0.0030
	✘				18875	2842	172	2670	18703	0.0605	0.0091	0.0158
		✘			18875	2900	289	2611	18586	0.0997	0.0153	0.0265
			✘		18875	1501	20	1481	18855	0.0133	0.0011	0.0020
				✘	18875	1867	34	1833	18841	0.0182	0.0018	0.0033
All unique					**18875**	**11597**	**547**	**11050**	**18328**	**0.0472**	**0.0290**	**0.0359**
✘	✘				18875	78	28	50	18847	0.3590	0.0015	0.0030
✘		✘			18875	419	52	367	18823	0.1241	0.0028	0.0054
✘			✘		18875	40	3	37	18872	0.0750	0.0002	0.0003
✘				✘	18875	91	5	86	18870	0.0549	0.0003	0.0005
	✘	✘			18875	39	17	22	18858	0.4359	0.0009	0.0018
	✘		✘		18875	311	135	176	18740	0.4341	0.0072	0.0141
	✘			✘	18875	254	107	147	18768	0.4213	0.0057	0.0112
		✘	✘		18875	382	75	307	18800	0.1963	0.0040	0.0078
		✘		✘	18875	615	67	548	18808	0.1089	0.0035	0.0069
			✘	✘	18875	718	56	662	18819	0.0780	0.0030	0.0057
All 2-way overlaps					**18875**	**2947**	**545**	**2402**	**18330**	**0.1849**	**0.0289**	**0.0499**
✘	✘	✘			18875	15	14	1	18861	0.9333	0.0007	0.0015
✘	✘		✘		18875	23	6	17	18869	0.2609	0.0003	0.0006
✘	✘			✘	18875	74	50	24	18825	0.6757	0.0026	0.0053
✘		✘	✘		18875	98	63	35	18812	0.6429	0.0033	0.0066
✘		✘		✘	18875	159	47	112	18828	0.2956	0.0025	0.0049
✘			✘	✘	18875	70	23	47	18852	.3286	0.0012	0.0024
	✘	✘	✘		18875	479	421	58	18454	0.8789	0.0223	0.0435
	✘	✘		✘	18875	78	43	35	18832	0.5513	0.0023	0.0045
	✘		✘	✘	18875	1011	628	383	18247	0.6212	0.0333	0.0632
		✘	✘	✘	18875	1081	201	880	18674	0.1859	0.0106	0.0201
All 3-way overlaps					**18875**	**3088**	**1496**	**1592**	**17379**	**0.4845**	**0.0793**	**0.1362**
✘	✘	✘	✘		18875	185	171	14	18704	0.9243	0.0091	0.0179
✘	✘	✘		✘	18875	51	44	7	18831	0.8627	0.0023	0.0046
✘	✘		✘	✘	18875	230	152	78	18723	0.6609	0.0081	0.0159
✘		✘	✘	✘	18875	1279	1137	142	17738	0.8890	0.0602	0.1128
	✘	✘	✘	✘	18875	3745	3169	576	15706	0.8462	0.1679	0.2802
All 4-way overlaps					**18875**	**5490**	**4673**	**817**	**14202**	**0.8512**	**0.2476**	**0.3836**
✘	✘	✘	✘	✘	**18875**	**9873**	**9374**	**499**	**9501**	**0.9495**	**0.4966**	**0.6521**

[a]The highlights show the overlaps included in the ConSemblEX-select-g assembly. The **boldfaced** numbers show the best score among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

Table A.7: Genome-guided assembly overlaps using the same reference genome for the Human dataset[a]

Bayesembler	Cufflinks	Scallop	StringTie	StringTie2	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
✘					17669	2172	133	2039	17536	0.0612	0.0075	0.0134
	✘				17669	3437	289	3148	17380	0.0841	0.0164	0.0274
		✘			17669	12267	277	11990	17392	0.0226	0.0157	0.0185
			✘		17669	1951	89	1862	17580	0.0456	0.0050	0.0091
				✘	17669	3078	379	2699	17290	0.1231	0.0214	0.0365
All unique					**17669**	**22905**	**1167**	**21738**	**16502**	**0.0509**	**0.0660**	**0.0575**
✘	✘				17669	157	28	129	17641	0.1783	0.0016	0.0031
✘		✘			17669	627	182	445	17487	0.2903	0.0103	0.0199
✘			✘		17669	142	30	112	17639	0.2113	0.0017	0.0034
✘				✘	17669	174	37	137	17632	0.2126	0.0021	0.0041
	✘	✘			17669	291	49	242	17620	0.1684	0.0028	0.0055
	✘		✘		17669	303	61	242	17608	0.2013	0.0035	0.0068
	✘			✘	17669	183	45	138	17624	0.2459	0.0025	0.0050
		✘	✘		17669	528	86	442	17583	0.1629	0.0049	0.0095
		✘		✘	17669	739	244	495	17425	0.3302	0.0138	0.0265
			✘	✘	17669	910	93	817	17576	0.1022	0.0053	0.0100
All 2-way overlaps					**17669**	**4054**	**855**	**3199**	**16814**	**0.2109**	**0.0484**	**0.0787**
✘	✘	✘			17669	189	71	118	17598	0.3757	0.0040	0.0080
✘	✘		✘		17669	108	28	80	17641	0.2593	0.0016	0.0032
✘	✘			✘	17669	56	20	36	17649	0.3571	0.0011	0.0023
✘		✘	✘		17669	477	223	254	17446	0.4675	0.0126	0.0246
✘		✘		✘	17669	385	175	210	17494	0.4545	0.0099	0.0194
✘			✘	✘	17669	146	51	95	17618	0.3493	0.0029	0.0057
	✘	✘	✘		17669	408	174	234	17495	0.4265	0.0098	0.0193
	✘	✘		✘	17669	73	19	54	17650	0.2603	0.0011	0.0021
	✘		✘	✘	17669	456	123	333	17546	0.2697	0.0070	0.0136
		✘	✘	✘	17669	645	167	478	17502	0.2589	0.0095	0.0182
All 3-way overlaps					**17669**	**2943**	**1051**	**1892**	**16618**	**0.3571**	**0.0595**	**0.1020**
✘	✘	✘	✘		17669	1476	1055	421	16614	0.7148	0.0597	0.1102
✘	✘	✘		✘	17669	244	121	123	17548	0.4959	0.0068	0.0135
✘	✘		✘	✘	17669	253	115	138	17554	0.4545	0.0065	0.0128
✘		✘	✘	✘	17669	1219	717	502	16952	0.5882	0.0406	0.0759
	✘	✘	✘	✘	17669	1195	544	651	17125	0.4552	0.0308	0.0577
All 4-way overlaps					**17669**	**4387**	**2552**	**1835**	**15117**	**0.5817**	**0.1444**	**0.2314**
✘	✘	✘	✘	✘	**17669**	**6094**	**4538**	**1556**	**13131**	**0.7447**	**0.2568**	**0.3819**

[a]The highlights show the overlaps included in the ConSemblEX-select-g assembly. The **boldfaced** numbers show the best score among all the overlaps.
[b]Total number of transcripts in the benchmark transcriptomes.
[c]Total number of contigs assembled by assembly method.

Table A.8: Genome-guided assembly using the same reference genome using various union sets among overlapping contig sets for the No-0 dataset[a]

Assembly sets[a]	**Actual**[c]	**Total**[d]	***TP***	***FP***	***FN***	**Precision**	**Recall**	F_1
1-way+	18875	32995	16635	16360	2240	0.5042	**0.8813**	0.6414
2-way+	18875	21398	16088	5310	2787	0.7518	0.8523	0.7989
3-way+	18875	18451	15543	2908	3332	0.8424	0.8235	**0.8328**
4-way+	18875	15363	14047	1316	4828	**0.9143**	0.7442	0.8206
1-way ∪ 2-way	18875	14544	1092	13452	17783	0.0751	0.0579	0.0654
2-way ∪ 3-way	18875	6035	2041	3994	16834	0.3382	0.1081	0.1639
3-way ∪ 4-way	18875	8578	6169	2409	12706	0.7192	0.3268	0.4494
1-way ∪ 4-way	18875	17087	5220	11867	13655	0.3055	0.2766	0.2903
2-way ∪ 4-way	18875	8437	5218	3219	13657	0.6185	0.2765	0.3821
1-way ∪ 5-way	18875	21470	9921	11549	8954	0.4621	0.5256	0.4918
2-way ∪ 5-way	18875	12820	9919	2901	8956	0.7737	0.5255	0.6259
3-way ∪ 5-way	18875	12961	10870	2091	8005	**0.8387**	**0.5759**	**0.6829**
1-way ∪ 2-way ∪ 3-way	18875	17632	2588	15044	16287	0.1468	0.1371	0.1418
1-way ∪ 2-way ∪ 4-way	18875	20034	5765	14269	13110	0.2878	0.3054	0.2963
1-way ∪ 3-way ∪ 4-way	18875	20175	6716	13459	12159	0.3329	0.3558	0.3440
1-way ∪ 2-way ∪ 5-way	18875	24417	10466	13951	8409	0.4286	0.5545	0.4835
1-way ∪ 3-way ∪ 5-way	18875	24558	11417	13141	7458	0.4649	0.6049	0.5257
1-way ∪ 4-way ∪ 5-way	18875	26960	14594	12366	4281	0.5413	**0.7732**	0.6368
2-way ∪ 3-way ∪ 5-way	18875	15908	11415	4493	7460	0.7176	0.6048	0.6564
2-way ∪ 4-way ∪ 5-way	18875	18310	14592	3718	4283	**0.7969**	0.7731	**0.7848**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.

[b]1-way to 4-way overlaps are shown in Table A.6 such as "All 2-way overlaps".

[c]Total number of transcripts in the benchmark transcriptomes.

[d]Total number of contigs assembled by assembly method.

Table A.9: Genome-guided assembly using the same reference genome using various union sets among overlapping contig sets for the Human datasets

Assembly sets[b]	**Actual**[c]	**Total**[d]	***TP***	***FP***	***FN***	**Precision**	**Recall**	F_1
1-way+	17669	40383	10163	30220	7506	0.2517	0.5752	0.3501
2-way+	17669	17478	8996	8482	8673	0.5147	0.5091	0.5119
3-way+	17669	13424	8141	5283	9528	0.6065	**0.4608**	**0.5237**
4-way+	17669	10481	7090	3391	10579	**0.6765**	0.4013	0.5037
1-way ∪ 2-way	17669	26959	2022	24937	15647	0.0750	0.1144	0.0906
2-way ∪ 3-way	17669	6997	1906	5091	15763	0.2724	0.1079	0.1545
3-way ∪ 4-way	17669	7330	3603	3727	14066	0.4915	0.2039	0.2883
1-way ∪ 4-way	17669	27292	3719	23573	13950	0.1363	0.2105	0.1654
2-way ∪ 4-way	17669	8441	3407	5034	14262	0.4036	0.1928	0.2610
1-way ∪ 5-way	17669	28999	5705	23294	11964	0.1967	**0.3229**	0.2445
2-way ∪ 5-way	17669	10148	5393	4755	12276	0.5314	0.3052	0.3877
3-way ∪ 5-way	17669	9037	5589	3448	12080	**0.6185**	0.3163	**0.4186**
1-way ∪ 2-way ∪ 3-way	17669	29902	3073	26829	14596	0.1028	0.1739	0.1292
1-way ∪ 2-way ∪ 4-way	17669	31346	4574	26772	13095	0.1459	0.2589	0.1866
1-way ∪ 3-way ∪ 4-way	17669	30235	4770	25465	12899	0.1578	0.2700	0.1991
1-way ∪ 2-way ∪ 5-way	17669	33053	6560	26493	11109	0.1985	0.3713	0.2587
1-way ∪ 3-way ∪ 5-way	17669	31942	6756	25186	10913	0.2115	0.3824	0.2724
1-way ∪ 4-way ∪ 5-way	17669	33386	8257	25129	9412	0.2473	**0.4673**	0.3235
2-way ∪ 3-way ∪ 5-way	17669	13091	6444	6647	11225	0.4922	0.3647	0.4190
2-way ∪ 4-way ∪ 5-way	17669	14535	7945	6590	9724	**0.5466**	0.4497	**0.4934**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each union.

[b]1-way to 4-way overlaps are shown in Table A.7 such as "All 2-way overlaps".

[c]Total number of transcripts in the benchmark transcriptomes.

[d]Total number of contigs assembled by assembly method.

Table A.10: Performance of ConSemblEX compared to individual genome-guided assemblers using the same reference genome[a]

Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	Bayesembler	18875	15172	11201	3971	7674	0.74	0.59	0.66
	Cufflinks	18875	19288	14531	4757	4344	0.75	0.77	0.76
	Scallop	18875	21397	15184	6213	3691	0.71	0.80	0.75
	StringTie	18875	21026	15634	5392	3241	0.74	**0.83**	0.78
	StringTie2	18875	21196	15137	6059	3738	0.71	0.80	0.76
	COnSemblEX-5g	18875	9873	9374	499	9501	**0.95**	0.50	0.65
	ConSemble3+g	18875	15688	14200	1488	4675	0.91	0.75	0.82
	ConSemblEX-3+g	18875	18451	15543	2908	3332	0.84	0.82	0.83
	ConSemblEX-4+g	18875	15363	14047	1316	4828	0.91	0.74	0.82
	ConSemblEX-select-g	18875	17870	15576	2294	3299	0.87	**0.83**	**0.85**
Col-0	Bayesembler	15508	15143	9158	5985	6350	0.60	0.59	0.60
	Cufflinks	15508	15768	8560	7208	6948	0.54	0.55	0.55
	Scallop	15508	18055	10534	7521	4974	0.58	**0.68**	0.63
	StringTie	15508	16908	9891	7017	5617	0.58	0.64	0.61
	StringTie2	15508	17722	10034	7688	5474	0.57	0.65	0.60
	ConSemblEX-5g	15508	7741	6171	1570	9337	**0.80**	0.40	0.53
	ConSemble3+g	15508	13380	9679	3701	5829	0.72	0.62	0.67
	ConSemblEX-3+g	15508	15313	10273	5040	5235	0.67	0.66	0.67
	ConSemblEX-4+g	15508	12527	9244	3283	6264	0.74	0.60	0.66
	ConSemblEX-select-g	15508	15265	10452	4813	5056	0.68	0.67	**0.68**
Human	Bayesembler	17669	13919	7524	6395	10145	0.54	0.43	0.48
	Cufflinks	17669	14923	7280	7643	10389	0.49	0.41	0.45
	Scallop	17669	26857	8642	18215	9027	0.32	**0.49**	0.39
	StringTie	17669	16311	8094	8217	9575	0.50	0.46	0.48
	StringTie2	17669	15850	7388	8462	10281	0.47	0.42	0.44
	ConSemblEX-5g	17669	6094	4538	1556	13131	**0.74**	0.26	0.38
	ConSemble3+g	17669	11945	7744	4201	9925	0.65	0.43	0.52
	ConSemblEX-3+g	17669	13424	8141	5283	9528	0.61	0.46	0.52
	ConSemblEX-4+g	17669	10481	7090	3391	10579	0.68	0.40	0.50
	ConSemblEX-select-g	17669	13055	8085	4970	9584	0.62	0.46	**0.53**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each dataset.

[b]Total number of transcripts in the benchmark transcriptomes.

[c]Total number of contigs assembled by assembly method.

Table A.11: Performance of ConSemblEX compared to individual genome-guided assemblers using a different reference genome[a]

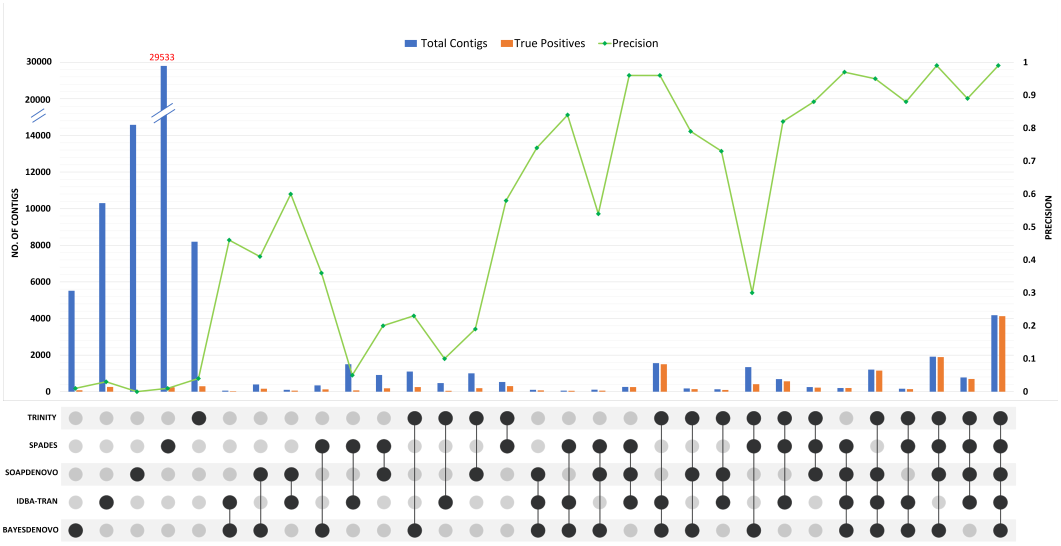
Dataset	Assembly Method	Actual[b]	Total[c]	*TP*	*FP*	*FN*	Precision	Recall	F_1
No-0	Bayesembler	18875	15531	5142	10389	13733	0.33	0.27	0.3
	Cufflinks	18875	19938	6510	13428	12365	0.33	0.34	0.34
	Scallop	18875	22298	6908	15390	11967	0.31	**0.37**	0.34
	StringTie	18875	21772	7069	14703	11806	0.32	**0.37**	0.35
	StringTie2	18875	21768	6858	14910	12017	0.32	0.36	0.34
	ConSemblEX-5g	18875	8797	4279	4518	14596	**0.49**	0.23	0.31
	ConSemblEX-3+g	18875	18101	7020	11081	11855	0.39	**0.37**	0.38
	ConSemblEX-4+g	18875	14291	6336	7955	12539	0.44	0.34	0.38
	ConSemblEX-select-g	18875	16649	6879	9770	11996	0.41	0.36	**0.39**
Col-0	Bayesembler	15508	15330	4810	10520	10698	0.31	0.31	0.31
	Cufflinks	15508	16664	4321	12343	11187	0.26	0.28	0.27
	Scallop	15508	16705	66	16639	15442	0.0	0.0	0.0
	StringTie	15508	17791	5074	12717	10434	0.29	0.33	0.3
	StringTie2	15508	18332	5199	13133	10309	0.28	**0.34**	0.31
	ConSemblEX-5g	15508	13	8	5	15500	**0.62**	0.00	0.00
	ConSemblEX-3+g	15508	13083	4840	8243	10668	0.37	0.31	**0.34**
	ConSemblEX-4+g	15508	7256	3317	3939	12191	0.46	0.21	0.29
	ConSemblEX-select-g	15508	17161	5266	11895	10242	0.31	**0.34**	0.32
Human	Bayesembler	17669	14610	5413	9197	12256	0.37	0.31	0.34
	Cufflinks	17669	16258	5296	10962	12373	0.33	0.3	0.31
	Scallop	17669	18778	6132	12646	11537	0.33	**0.35**	0.34
	StringTie	17669	18339	5851	12488	11818	0.32	0.33	0.32
	StringTie2	17669	20203	6217	13986	11452	0.31	**0.35**	0.33
	ConSemblEX-5g	17669	7234	3881	3353	13788	**0.54**	0.22	0.31
	ConSemblEX-3+g	17669	15006	6000	9006	11669	0.40	0.34	0.37
	ConSemblEX-4+g	17669	11607	5399	6208	12270	0.47	0.31	0.37
	ConSemblEX-select-g	17669	13705	5879	7826	11790	0.43	0.33	**0.38**

[a]The **boldfaced** numbers with green highlights show the best score among all the overlaps for each dataset.

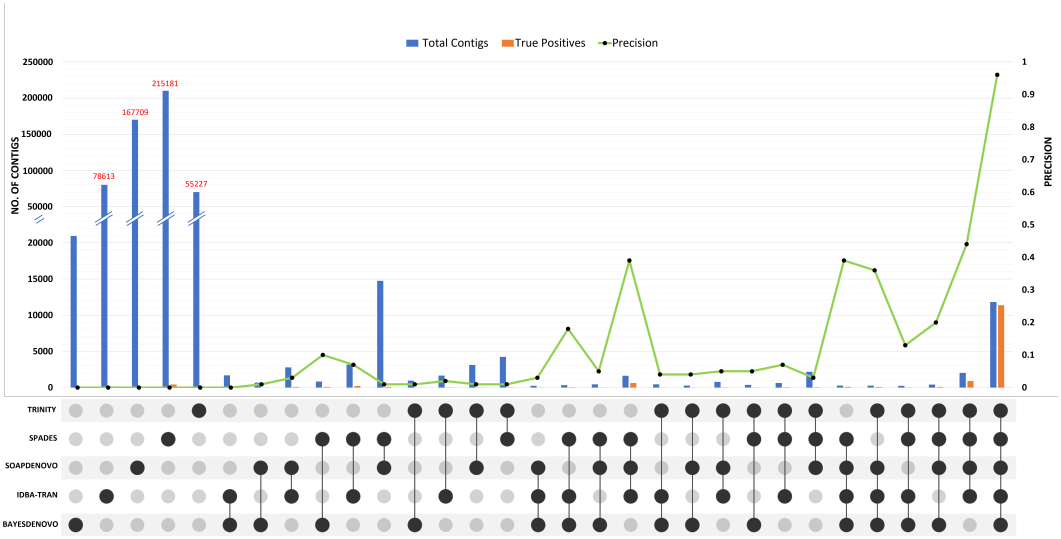
[b]Total number of transcripts in the benchmark transcriptomes.

[c]Total number of contigs assembled by assembly method.

Appendix B

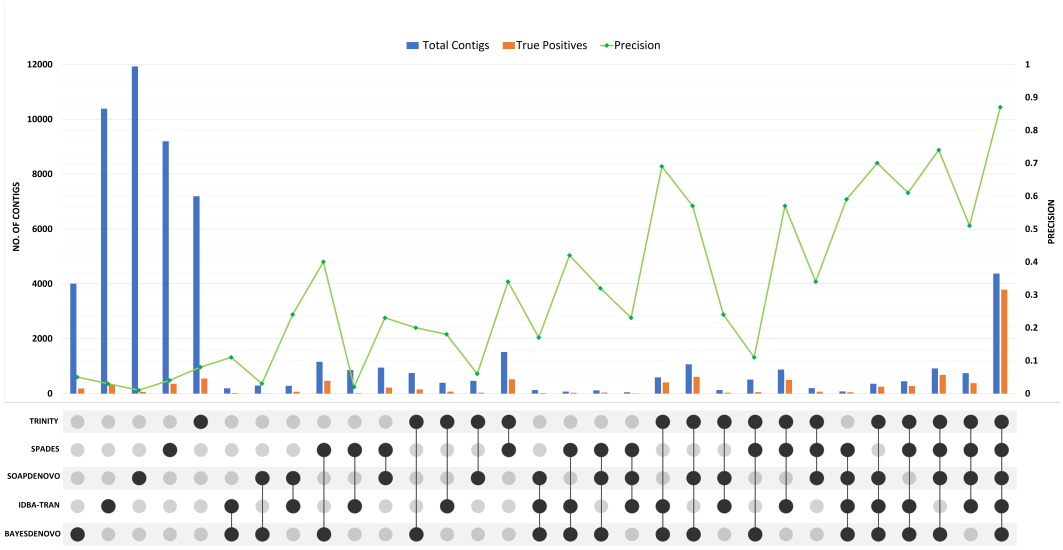


(a) No-0 default k-mer

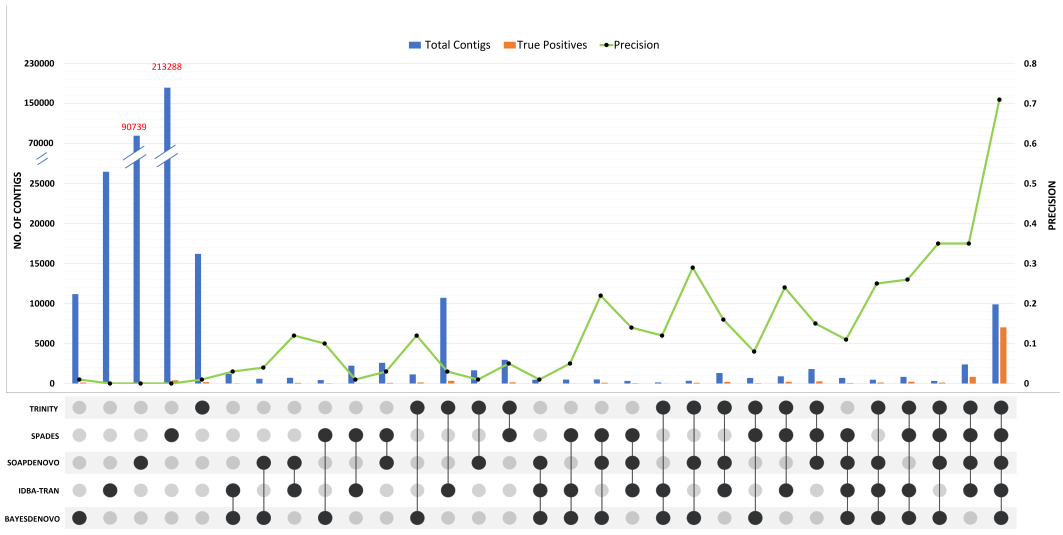


(b) No-0 pooled k-mers

Figure B.1: Distribution of the assembled contig overlaps for the No-0 dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five *de novo* assemblers. Assembly was performed using the default k-mer values (a) or multiple k-mer values (b). Overlaps among the five methods are indicated with connected closed circles. Each overlap set is exclusive to each other.

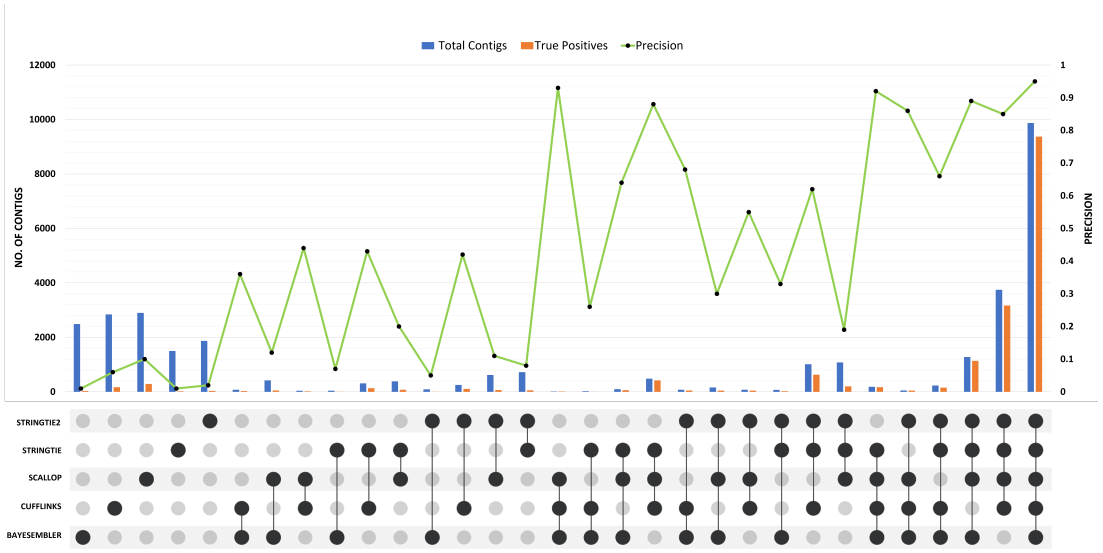


(a) Human default k-mer

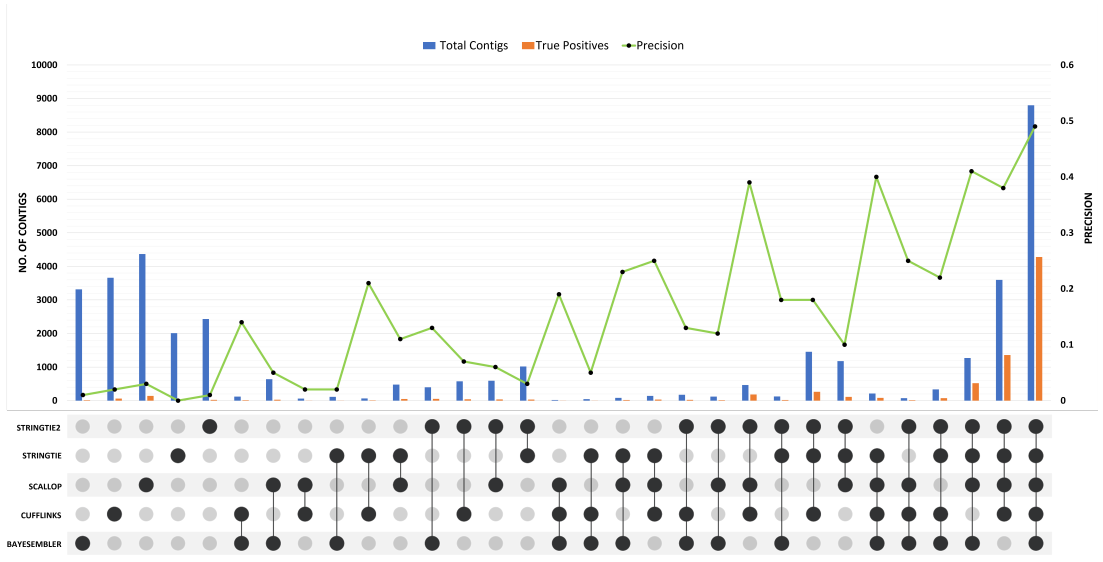


(b) Human pooled k-mers

Figure B.2: Distribution of the assembled contig overlaps for the Human dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five *de novo* assemblers. Assembly was performed using the default k-mer values (a) or multiple k-mer values (b). Overlaps among the five methods are indicated with connected closed circles. Each overlap set is exclusive to each other.

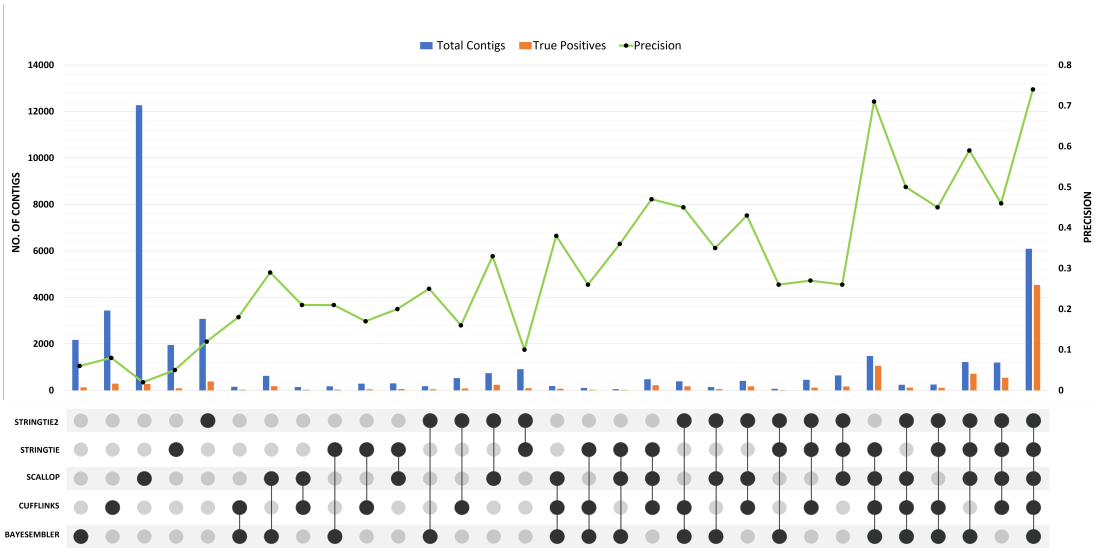


(a) No-0 same reference-genome

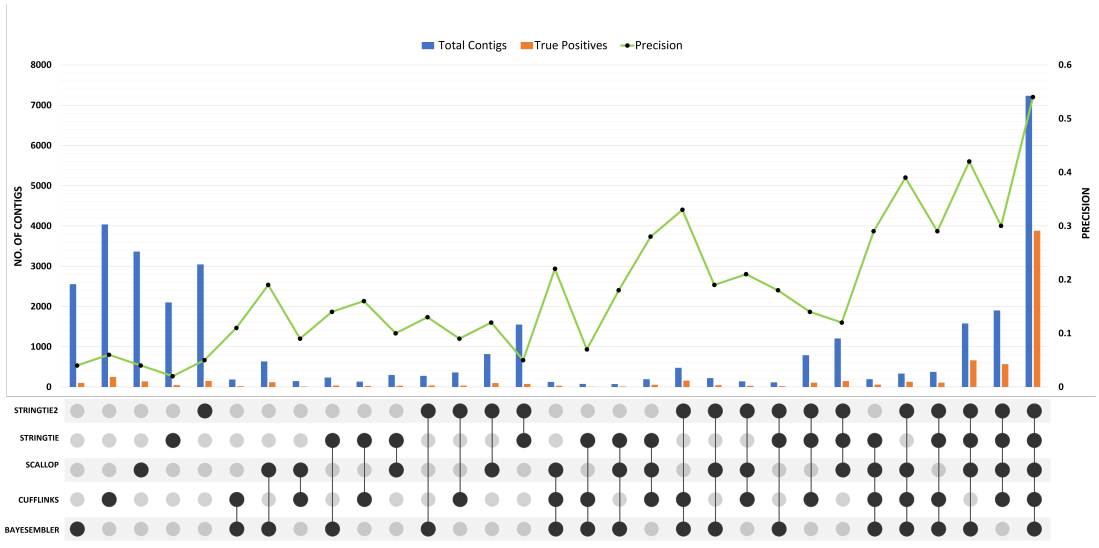


(b) No-0 different reference-genome

Figure B.3: Distribution of the assembled contig overlaps for the No-0 dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five genome-guided assemblers for the Col-0 dataset. Each assembly was performed using the same reference-genome (a) and different reference-genome (b). Overlaps among the five methods are indicated with the connected closed circles. Each overlap set is exclusive to each other.



(a) Human same reference



(b) Human different reference

Figure B.4: Distribution of the assembled contig overlaps for the Human dataset. The numbers of total assembled contigs (blue) and correctly assembled contigs (orange) are shown for each overlapping assembly set among five genome-guided assemblers for the Col-0 dataset. Each assembly was performed using the same reference-genome (a) and different reference-genome (b). Overlaps among the five methods are indicated with the connected closed circles. Each overlap set is exclusive to each other.

Bibliography

[1] K. F. Au, H. Jiang, L. Lin, Y. Xing, and W. H. Wong. Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic Acids Research*, 38(14):4570–4578, Aug. 2010.

[2] S. Behera, A. Voshall, and E. N. Moriyama. Plant Transcriptome Assembly: Review and Benchmarking. In N. Helder I., editor, *Bioinformatics*. Exon Publications, Brisbane (AU), 2021.

[3] T. A. Brown. *Genomes*. Wiley-Liss, Oxford, 2nd edition, 2002.

[4] E. Bushmanova, D. Antipov, A. Lapidus, and A. D. Prjibelski. rnaSPAdes: a de novo transcriptome assembler and its application to RNA-Seq data. *GigaScience*, 8(9), Sept. 2019.

[5] E. Bushmanova, D. Antipov, A. Lapidus, V. Suvorov, and A. D. Prjibelski. rnaQUAST: a quality assessment tool for de novo transcriptome assemblies. *Bioinformatics*, 32(14):2210–2212, July 2016.

[6] N. Cerveau and D. J. Jackson. Combining independent de novo assemblies optimizes the coding transcriptome for nonconventional model eukaryotic organisms. *BMC Bioinformatics*, 17:525, Dec. 2016.

[7] Z. Chang, G. Li, J. Liu, Y. Zhang, C. Ashby, D. Liu, C. L. Cramer, and X. Huang. Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. *Genome Biology*, 16(1):30, Feb. 2015.

[8] R. Chikhi, A. Limasset, S. Jackman, J. Simpson, and P. Medvedev. On the representation of de Bruijn graphs. Technical Report arXiv:1401.5383, arXiv, Oct. 2014.

[9] R. Chikhi and P. Medvedev. Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1):31–37, Jan. 2014.

[10] M. R. Crusoe, H. F. Alameldin, S. Awad, E. Boucher, A. Caldwell, R. Cartwright, A. Charbonneau, B. Constantinides, G. Edvenson, S. Fay, J. Fenton, T. Fenzl, J. Fish, L. Garcia-Gutierrez, P. Garland, J. Gluck, I. González, S. Guermond, J. Guo, A. Gupta, J. R. Herr, A. Howe, A. Hyer, A. Härpfer, L. Irber, R. Kidd, D. Lin, J. Lippi, T. Mansour, P. McA'Nulty, E. McDonald, J. Mizzi, K. D. Murray, J. R. Nahum, K. Nanlohy, A. J. Nederbragt, H. Ortiz-Zuazaga, J. Ory, J. Pell, C. Pepe-Ranney, Z. N. Russ, E. Schwarz, C. Scott, J. Seaman, S. Sievert, J. Simpson, C. T. Skennerton, J. Spencer, R. Srinivasan, D. Standage, J. A. Stapleton, S. R. Steinman, J. Stein, B. Taylor, W. Trimble, H. L. Wiencko, M. Wright, B. Wyss, Q. Zhang, E. Zyme, and C. T. Brown. The khmer software package: enabling efficient nucleotide sequence analysis. Technical Report 4:900, F1000Research, Sept. 2015.

[11] A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, Jan. 2013.

[12] D. A. Durai and M. H. Schulz. Informed kmer selection for de novo transcriptome assembly. *Bioinformatics*, 32(11):1670–1677, June 2016.

[13] C. D. Fabbro, S. Scalabrin, M. Morgante, and F. M. Giorgi. An Extensive Evaluation of Read Trimming Effects on Illumina NGS Data Analysis. *PLOS ONE*, 8(12):e85024, Dec. 2013.

[14] L. Fu, B. Niu, Z. Zhu, S. Wu, and W. Li. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, 28(23):3150–3152, Dec. 2012.

[15] X. Gan, O. Stegle, J. Behr, J. G. Steffen, P. Drewe, K. L. Hildebrand, R. Lyngsoe, S. J. Schultheiss, E. J. Osborne, V. T. Sreedharan, A. Kahles, R. Bohnert, G. Jean, P. Derwent, P. Kersey, E. J. Belfield, N. P. Harberd, E. Kemen, C. Toomajian, P. X. Kover, R. M. Clark, G. Rätsch, and R. Mott. Multiple reference genomes and transcriptomes for Arabidopsis thaliana. *Nature*, 477(7365):419–423, Sept. 2011.

[16] D. G. Gilbert. Genes of the pig, Sus scrofa, reconstructed with EvidentialGene. *PeerJ*, 7:e6374, Feb. 2019.

[17] D. G. Gilbert. Longest protein, longest transcript or most expression, for accurate gene reconstruction of transcriptomes? Technical report, bioRxiv, Nov. 2019.

[18] M. G. Grabherr, B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, I. Amit, X. Adiconis, L. Fan, R. Raychowdhury, Q. Zeng, Z. Chen, E. Mauceli, N. Hacohen, A. Gnirke, N. Rhind, F. di Palma, B. W. Birren, C. Nusbaum, K. Lindblad-Toh, N. Friedman, and A. Regev. Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nat Biotechnol*, 29(7):644–652, May 2011.

[19] T. Griebel, B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, R. Guigó, and M. Sammeth. Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research*, 40(20):10073–10083, Nov. 2012.

[20] B. He, S. Zhao, Y. Chen, Q. Cao, C. Wei, X. Cheng, and Y. Zhang. Optimal assembly strategies of transcriptome related to ploidies of eukaryotic organisms. *BMC Genomics*, 16(1):65, Feb. 2015.

[21] S. Heber, M. Alekseyev, S.-H. Sze, H. Tang, and P. A. Pevzner. Splicing graphs and EST assembly problem. *Bioinformatics*, 18(suppl_1):S181–S188, July 2002.

[22] L. A. Honaas, E. K. Wafula, N. J. Wickett, J. P. Der, Y. Zhang, P. P. Edger, N. S. Altman, J. C. Pires, J. H. Leebens-Mack, and C. W. dePamphilis. Selecting Superior De Novo Transcriptome Assemblies: Lessons Learned by Leveraging the Best Plant Genome. *PLOS ONE*, 11(1):e0146062, Jan. 2016.

[23] M. Hölzer and M. Marz. De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers. *GigaScience*, 8(5), May 2019.

[24] R. M. Idury and M. S. Waterman. A new algorithm for DNA sequence assembly. *J Comput Biol*, 2(2):291–306, 1995.

[25] D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg. Graph-Based Genome Alignment and Genotyping with HISAT2 and HISAT-genotype. *Nat Biotechnol*, 37(8):907–915, Aug. 2019.

[26] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*, 14(4):R36, 2013.

[27] S. Kovaka, A. V. Zimin, G. M. Pertea, R. Razaghi, S. L. Salzberg, and M. Pertea. Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biol*, 20(1):278, Dec. 2019.

[28] K. R. Kukurba and S. B. Montgomery. RNA Sequencing and Analysis. *Cold Spring Harb Protoc*, 2015(11):951–969, Apr. 2015.

[29] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992, Dec. 2014.

[30] B. Li, N. Fillmore, Y. Bai, M. Collins, J. A. Thomson, R. Stewart, and C. N. Dewey. Evaluation of de novo transcriptome assemblies from RNA-Seq data. *Genome Biology*, 15(12):553, Dec. 2014.

[31] S. Madritsch, A. Burg, and E. M. Sehr. Comparing de novo transcriptome assembly tools in di- and autotetraploid non-model plant species. *BMC Bioinformatics*, 22(1):146, Mar. 2021.

[32] S. Mao, L. Pachter, D. Tse, and S. Kannan. RefShannon: A genome-guided transcriptome assembler using sparse flow decomposition. *PLOS ONE*, 15(6):e0232946, June 2020.

[33] L. Maretty, J. A. Sibbesen, and A. Krogh. Bayesian transcriptome assembly. *Genome Biology*, 15(10):501, Oct. 2014.

[34] J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nat Rev Genet*, 12(10):671–682, Oct. 2011.

[35] J. R. Miller, S. Koren, and G. Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327, June 2010.

[36] S. T. O'Neil and S. J. Emrich. Assessing De Novo transcriptome assembly metrics for consistency and utility. *BMC Genomics*, 14(1):465, July 2013.

[37] Y. Peng, H. C. M. Leung, S.-M. Yiu, M.-J. Lv, X.-G. Zhu, and F. Y. L. Chin. IDBA-tran: a more robust de novo de Bruijn graph assembler for transcriptomes with uneven expression levels. *Bioinformatics*, 29(13):i326–334, July 2013.

[38] M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. T. Mendell, and S. L. Salzberg. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat Biotechnol*, 33(3):290–295, Mar. 2015.

[39] P. A. Pevzner. 1-Tuple DNA sequencing: computer analysis. *J Biomol Struct Dyn*, 7(1):63–73, Aug. 1989.

[40] M. Shao and C. Kingsford. Accurate assembly of transcripts through phase-preserving graph decomposition. *Nat Biotechnol*, 35(12):1167–1169, Dec. 2017.

[41] X. Shi, X. Wang, A. F. Neuwald, L. Halakivi-Clarke, R. Clarke, and J. Xuan. A Bayesian approach for accurate de novo transcriptome assembly. *Sci Rep*, 11:17663, Sept. 2021.

[42] F. A. Simão, R. M. Waterhouse, P. Ioannidis, E. V. Kriventseva, and E. M. Zdobnov. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics*, 31(19):3210–3212, Oct. 2015.

[43] R. Smith-Unna, C. Boursnell, R. Patro, J. M. Hibberd, and S. Kelly. TransRate: reference-free quality assessment of de novo transcriptome assemblies. *Genome Res.*, 26(8):1134–1144, Aug. 2016.

[44] D. Swarbreck, C. Wilks, P. Lamesch, T. Z. Berardini, M. Garcia-Hernandez, H. Foerster, D. Li, T. Meyer, R. Muller, L. Ploetz, A. Radenbaugh, S. Singh,

V. Swing, C. Tissier, P. Zhang, and E. Huala. The Arabidopsis Information Resource (TAIR): gene structure and function annotation. *Nucleic Acids Res*, 36(Database issue):D1009–1014, Jan. 2008.

[45] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter. Transcript assembly and abundance estimation from RNA-Seq reveals thousands of new transcripts and switching among isoforms. *Nat Biotechnol*, 28(5):511–515, May 2010.

[46] A. Voshall, S. Behera, X. Li, X.-H. Yu, K. Kapil, J. S. Deogun, J. Shanklin, E. B. Cahoon, and E. N. Moriyama. A consensus-based ensemble approach to improve transcriptome assembly. *BMC Bioinformatics*, 22(1):513, Oct. 2021.

[47] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63, Jan. 2009.

[48] D. L. Wheeler, D. M. Church, S. Federhen, A. E. Lash, T. L. Madden, J. U. Pontius, G. D. Schuler, L. M. Schriml, E. Sequeira, T. A. Tatusova, and L. Wagner. Database resources of the National Center for Biotechnology. *Nucleic Acids Research*, 31(1):28–33, Jan. 2003.

[49] Y. Xie, G. Wu, J. Tang, R. Luo, J. Patterson, S. Liu, W. Huang, G. He, S. Gu, S. Li, X. Zhou, T.-W. Lam, Y. Li, X. Xu, G. K.-S. Wong, and J. Wang. SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics*, 30(12):1660–1666, June 2014.

[50] T. Yu, Z. Mu, Z. Fang, X. Liu, X. Gao, and J. Liu. TransBorrow: genome-guided transcriptome assembly by borrowing assemblies from different assemblers. *Genome Res*, 30(8):1181–1190, Aug. 2020.