The Oyster River Protocol: a multi-assembler and kmer approach for de novo transcriptome assembly

Matthew D. MacManes

Department of Molecular, Cellular and Biomedical Sciences, University of New Hampshire, Durham, NH, USA

ABSTRACT

Characterizing transcriptomes in non-model organisms has resulted in a massive increase in our understanding of biological phenomena. This boon, largely made possible via high-throughput sequencing, means that studies of functional, evolutionary, and population genomics are now being done by hundreds or even thousands of labs around the world. For many, these studies begin with a de novo transcriptome assembly, which is a technically complicated process involving several discrete steps. The Oyster River Protocol (ORP), described here, implements a standardized and benchmarked set of bioinformatic processes, resulting in an assembly with enhanced qualities over other standard assembly methods. Specifically, ORP produced assemblies have higher Detonate and TransRate scores and mapping rates, which is largely a product of the fact that it leverages a multi-assembler and kmer assembly process, thereby bypassing the shortcomings of any one approach. These improvements are important, as previously unassembled transcripts are included in ORP assemblies, resulting in a significant enhancement of the power of downstream analysis. Further, as part of this study, I show that assembly quality is unrelated with the number of reads generated, above 30 million reads. Code Availability: The version controlled open-source code is available at https://github.com/macmanes-lab/Oyster_River_Protocol. Instructions for software installation and use, and other details are available at http://oyster-river-protocol.rtfd.org/.

Subjects Bioinformatics, Genomics
Keywords Transcriptome, Assembly, Bioinformatics

Submitted 22 August 2017
Accepted 21 July 2018
Published 3 August 2018

Corresponding author
Matthew D. MacManes,
macmanes@gmail.com

Academic editor
Richard Emes

Additional Information and Declarations can be found on page 14

DOI **10.7717/peerj.5428**

INTRODUCTION

For all biology, modern sequencing technologies have provided for an unprecedented opportunity to gain a deep understanding of genome level processes that underlie a very wide array of natural phenomena, from intracellular metabolic processes to global patterns of population variability. Transcriptome sequencing has been influential (*Mortazavi et al., 2008*; *Wang, Gerstein & Snyder, 2009*), particularly in functional genomics (*Lappalainen et al., 2013*; *Cahoy et al., 2008*), and has resulted in discoveries not possible even just a few years ago. This in large part is due to the scale at which these studies may be conducted (*Li et al., 2017*; *Tan et al., 2017*). Unlike studies of adaptation based on one or a small number of candidate genes (*Fitzpatrick et al., 2005*;

Panhuis, 2006), modern studies may assay the entire suite of expressed transcripts—the transcriptome—simultaneously. In addition to issues of scale, as a direct result of enhanced dynamic range, newer sequencing studies have increased ability to simultaneously reconstruct and quantitate lowly- and highly-expressed transcripts (*Wolf, 2013*; *Vijay et al., 2013*). Lastly, improved methods for the detection of differences in gene expression (*Robinson, McCarthy & Smyth, 2010*; *Love, Huber & Anders, 2014*) across experimental treatments have resulted in increased resolution for studies aimed at understanding changes in gene expression.

As a direct result of their widespread popularity, a diverse tool set for the assembly of transcriptome exists, with each potentially reconstructing transcripts others fail to reconstruct. Amongst the earliest of specialized de novo transcriptome assemblers were the packages `Trans-ABySS` (*Robertson et al., 2010*), `Oases` (*Schulz et al., 2012*), and `SOAPdenovoTrans` (*Xie et al., 2014*), which were fundamentally based on the popular de Bruijn graph-based genome assemblers `ABySS` (*Simpson et al., 2009*), `Velvet` (*Zerbino & Birney, 2008*), and `SOAP` (*Li et al., 2008*), respectively. These early efforts gave rise to a series of more specialized de novo transcriptome assemblers, namely `Trinity` (*Haas et al., 2013*), and `IDBA-Tran` (*Peng et al., 2013*). While the de Bruijn graph approach remains powerful, newly developed software explores novel parts of the algorithmic landscape, offering substantial benefits, assuming novel methods reconstruct different fractions of the transcriptome. `BinPacker` (*Liu et al., 2016*), for instance, abandons the de Bruijn graph approach to model the assembly problem after the classical bin packing problem, while `Shannon` (*Kannan et al., 2016*) uses information theory, rather than a set of software engineer-decided heuristics. These newer assemblers, by implementing fundamentally different assembly algorithms, may reconstruct fractions of the transcriptome that other assemblers fail to accurately assemble.

In addition to the variety of tools available for the de novo assembly of transcripts, several tools are available for pre-processing of reads via read trimming (e.g., `Skewer`; *Jiang et al., 2014*, `Trimmomatic`; *Bolger, Lohse & Usadel, 2014*, `Cutadapt`; *Martin, 2011*), read normalization (`khmer`; *Pell et al., 2012*), and read error correction (`SEECER`; *Le et al., 2013*, `RCorrector`; *Song & Florea, 2015*, `Reptile`; *Yang, Dorman & Aluru, 2010*). Similarly, benchmarking tools that evaluate the quality of assembled transcriptomes including `TransRate` (*Smith-Unna et al., 2016*), `BUSCO` (Benchmarking Universal Single-Copy Orthologs; *Simão et al., 2015*), and `Detonate` (*Li et al., 2014*) have been developed. Despite the development of these evaluative tools, this manuscript describes the first systematic effort coupling them with the development of a de novo transcriptome assembly pipeline.

The ease with which these tools may be used to produce and characterize transcriptome assemblies belies the true complexity underlying the overall process (*Ungaro et al., 2017*; *Wang & Gribskov, 2017*; *Moreton, Izquierdo & Emes, 2015*; *Yang & Smith, 2013*). Indeed, the subtle (and not so subtle) methodological challenges associated with transcriptome reconstruction may result in highly variable assembly quality. In particular, while most tools run using default settings, these defaults may be sensible only for one specific (often unspecified) use case or data type. Because parameter optimization is both

dataset-dependent and factorial in nature, an exhaustive optimization particularly of entire pipelines, is never possible. Given this, the production of a de novo transcriptome assembly requires a large investment in time and resources, with each step requiring careful consideration. Here, I propose an evidence-based protocol for assembly that results in the production of high quality transcriptome assemblies, across a variety of commonplace experimental conditions or taxonomic groups.

This manuscript describes the development of The Oyster River Protocol (ORP)[1] for transcriptome assembly. It explicitly considers and attempts to address many of the shortcomings described in *Vijay et al. (2013)*, by leveraging a multi-kmer and multi-assembler strategy. This innovation is critical, as all assembly solutions treat the sequence read data in ways that bias transcript recovery. Specifically, with the development of assembly software comes the use of a set of heuristics that are necessary given the scope of the assembly problem itself. Given each software development team carries with it a unique set of ideas related to these heuristics while implementing various assembly algorithms, individual assemblers exhibit unique assembly behavior. By leveraging a multi-assembler approach, the strengths of one assembler may complement the weaknesses of another. In addition to biases related to assembly heuristics, it is well known that assembly kmer-length has important effects on transcript reconstruction, with shorter kmers more efficiently reconstructing lower-abundance transcripts relative to more highly abundant transcripts. Given this, assembling with multiple different kmer lengths, then merging the resultant assemblies may effectively reduce this type of bias. Recognizing these issue, I hypothesize that an assembly that results from the combination of multiple different assemblers and lengths of assembly-kmers will be better than each individual assembly, across a variety of metrics.

In addition to developing an enhanced pipeline, the work suggests an exhaustive way of characterizing assemblies while making available a set of fully-benchmarked reference assemblies that may be used by other researchers in developing new assembly algorithms and pipelines. Although many other researchers have published comparisons of assembly methods, up until now these have been limited to single datasets assembled a few different ways (*Marchant et al., 2016*; *Finseth & Harrison, 2014*), thereby failing to provide more general insights.

[1] Named the Oyster River Protocol because the ideas, and some of the code, was developed while overlooking the Oyster River, located in Durham, New Hampshire. NB, the naming assembly of protocols after bodies of water was, to the best of my knowledge, first done by C. Titus Brown (The Eel Pond Protocol: http://khmer-protocols.readthedocs.io/en/latest/mrnaseq/index.html), and may have subconsciously influenced me in naming this protocol.

METHODS

Datasets

In an effort at benchmarking the assembly and merging protocols, I downloaded a set of publicly available RNAseq datasets (Table 1) that had been produced on the Illumina sequencing platform. These datasets were chosen to represent a variety of taxonomic groups, so as to demonstrate the broad utility of the developed methods. Because datasets were selected randomly with respect to sequencing center and read number, they are likely to represent the typical quality of Illumina data circa 2014–2017.

Software

The ORP can be installed on the Linux platform, and does not require superuser privileges, assuming `Linuxbrew` (*Jackman & Birol, 2016*) is installed. The software is implemented

Table 1 Lists the datasets used in this study.

Type	Accession	Species	Number of reads (M)	Read length (bp)
Animalia	ERR489297	*Anopheles gambiae*	206	100
Animalia	DRR030368	*Echinococcus multilocularis*	73	100
Animalia	ERR1016675	*Heterorhabditis indica*	51	100
Animalia	SRR2086412	*Mus musculus*	54	100
Animalia	DRR036858	*Mus musculus*	114	100
Animalia	DRR046632	*Oncorhynchus mykiss*	82	76
Animalia	SRR1789336	*Oryctolagus cuniculus*	31	100
Animalia	SRR2016923	*Phyllodoce medipapillata*	86	100
Animalia	ERR1674585	*Schistosoma mansoni*	39	100
Plant	DRR082659	*Aeginetia indica*	69	90
Plant	DRR053698	*Cephalotus follicularis*	126	90
Plant	DRR069093	*Hevea brasiliensis*	103	100
Plant	SRR3499127	*Nicotiana tabacum*	30	150
Plant	DRR031870	*Vigna angularis*	60	100
Protozoa	ERR058009	*Entamoeba histolytica*	68	100

Note:
All datasets are publicly available for download by accession number at the European Nucleotide Archive or NCBI Short Read Archive.

as a stand-alone makefile which coordinates all steps described below. All scripts are available at https://github.com/macmanes-lab/Oyster_River_Protocol, and run on the Linux platform. The software is version controlled and openly-licensed to promote sharing and reuse. A guide for users is available at http://oyster-river-protocol.rtfd.io.

Pre-assembly procedures

For all assemblies performed, Illumina sequencing adapters were removed from both ends of the sequencing reads, as were nucleotides with quality Phred $\leq$2, using the program `Trimmomatic` version 0.36 (*Bolger, Lohse & Usadel, 2014*), following the recommendations from *MacManes (2014)*. After trimming, reads were error corrected using the software `RCorrector` version 1.0.2 (*Song & Florea, 2015*), following recommendations from *MacManes & Eisen (2013)*. The code for running this step of the ORPs is available at https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/oyster.mk#L145. The trimmed and error corrected reads were then subjected to de novo assembly.

Assembly

I assembled each trimmed and error corrected dataset using three different de novo transcriptome assemblers and three different kmer lengths, producing four unique assemblies. First, I assembled the reads using `Trinity` release 2.4.0 (*Haas et al., 2013*), and default settings (k = 25), without read normalization. The decision to forgo normalization is based on previous work (*MacManes, 2015*) showing slightly worse

performance of normalized datasets. Next, the `SPAdes` RNAseq assembler (version 3.10) (*Chikhi & Medvedev, 2014*) was used, in two distinct runs, using kmer sizes 55 and 75. Lastly, reads were assembled using the assembler `Shannon` version 0.0.2 (*Kannan et al., 2016*), using a kmer length of 75. These assemblers were chosen based on the fact that they (1) use an open-science development model, whereby end-users may contribute code, (2) are all actively maintained and are undergoing continuous development, and (3) occupy different parts of the algorithmic landscape.

This assembly process resulted in the production of four distinct assemblies. The code for running this step of the ORPs is available at https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/oyster.mk#L148.

Assembly merging via OrthoFuse

To merge the four assemblies produced as part of the ORP, I developed new software that effectively merges transcriptome assemblies. Described in brief, `OrthoFuse` begins by concatenating all assemblies together, then forms groups of transcripts by running a version of `OrthoFinder` (*Emms & Kelly, 2015*) packaged with the ORP, modified to accept nucleotide sequences from the merged assembly. These groupings represent groups of homologous transcripts. While isoform reconstruction using short-read data is notoriously poor, by increasing the inflation parameter by default to $I = 4$, it attempts to prevent the collapsing of transcript isoforms into single groups. After `Orthofinder` has completed, a modified version of `TransRate` version 1.0.3 (*Smith-Unna et al., 2016*) which is packaged with the ORP, is run on the merged assembly, after which the best (= highest contig score) transcript is selected from each group and placed in a new assembly file to represent the entire group. The resultant file, which contains the highest scoring contig for each orthogroup, may be used for all downstream analyses. `OrthoFuse` is run automatically as part of the ORP, and additionally is available as a stand alone script, https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/orthofuser.mk.

Assembly evaluation

All assemblies were evaluated using `ORP-TransRate`, `Detonate` version 1.11 (*Li et al., 2014*), `shmlast` version 1.2 (*Scott, 2017*), and `BUSCO` version 3.0.2 (*Simão et al., 2015*). `TransRate` evaluates transcriptome assembly contiguity by producing a score based on length-based and mapping metrics, while `Detonate` conducts an orthogonal analysis, producing a score that is maximized by an assembly that is representative of input sequence read data. `BUSCO` evaluates assembly content by searching the assemblies for conserved single copy orthologs found in all Eukaryotes. I report default `BUSCO` metrics as described in *Simão et al. (2015)*. Specifically, "complete orthologs," are defined as query transcripts that are within two standard deviations of the length of the `BUSCO` group mean, while contigs falling short of this metric are listed as "fragmented." `Shmlast` implements the conditional reciprocal best hits test (*Aubry et al., 2014*), conducted in this case against the Swiss-Prot protein database (downloaded October, 2017) using an *e*-value of $1E$-10.

In addition to the generation of metrics to evaluation the quality of transcriptome assemblies, I generated a distance matrix of assemblies for each dataset using the `sourmash` package (*Titus Brown & Irber, 2016*), in an attempt at characterizing the algorithmic landscape of assemblers. Specifically, each assembly was characterized using the `compute` function using 5,000 independent sketches. The distance between assemblies was calculated using the compare function and a kmer length of 51. These distance matrices were visualized using the `isoMDS` function of the MASS package (https://CRAN.R-project.org/package=MASS).

Statistics

All statistical analyses were conducted in R version 3.4.0 (*R Core Development Team, 2011*). Violin plots were constructed using the beanplot (*Kampstra, 2008*) and the beeswarm R packages (https://CRAN.R-project.org/package=beeswarm). Expression distributions were plotted using the ggridges package (https://CRAN.R-project.org/package=ggridges).

RESULTS AND DISCUSSION

A total of 15 RNAseq datasets, ranging in size from (30–206 M paired end reads) were assembled using the ORP and with `Trinity`. Each assembly was evaluated using the software `BUSCO`, `shmlast`, `Detonate`, and `TransRate`. From these, several metrics were chosen to represent the quality of the produced assemblies. Of note, all the assemblies produced as part of this work are available at DOI 10.5281/zenodo.1320141. A file containing the evaluative metrics is available at https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/manuscript/orp.csv, while the distance matrices are available within the folder https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/manuscript/. R code used to conduct analyses and make figures is found at https://github.com/macmanes-lab/Oyster_River_Protocol/blob/master/manuscript/R-analysis.Rmd.

Assembled transcriptomes

The `Trinity` assembly of trimmed and error corrected reads generally completed on a standard Linux server using 24 cores, in less than 24 h. RAM requirement is estimated to be close to 0.5 Gb per million paired-end reads. The assemblies on average contained 176 k transcripts (range 19–643 k) and 97 Mb (range 14 MB–198 Mb). Other quality metrics will be discussed below, specifically in relation to the ORP produced assemblies.

Oyster River Protocol assemblies generally completed on a standard Linux server using 24 cores in 3 days. Typically `Trinity` was the longest running assembler, with the individual `SPAdes` assemblies being the shortest. RAM requirement is estimated to be 1.5–2 Gb per million paired-end reads, with `SPAdes` requiring the most. The assemblies on average contained 153 k transcripts (range 23–625 k) and 64 Mb (range 8 MB–181 Mb).

MinHash sketch signatures (*Ondov et al., 2016*) of each assemblies of a given dataset were calculated using `sourmash` (*Titus Brown & Irber, 2016*), and a MDS plot was

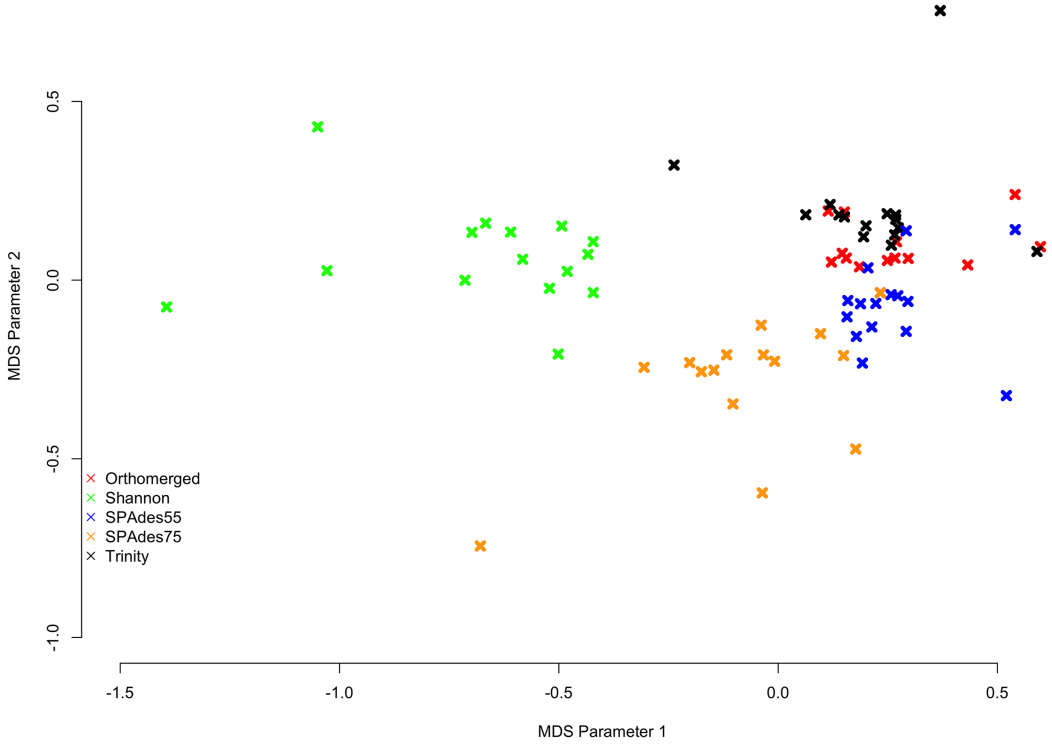

Figure 1 MDS plot describing the similarity within and between assemblers. Colored *x*'s mark individual assemblies, with red marks corresponding to the ORP assemblies, green marks corresponding to the Shannon assemblies, blue marks corresponding to the SPAdes55 assemblies, orange marks corresponding to the SPAdes75 assemblies, and the black marks corresponding to the Trinity assemblies. In general assemblies produced by a given assembler tend to cluster together.

Full-size DOI: 10.7717/peerj.5428/fig-1

generated (Fig. 1) from their distances. Interestingly, each assembler tends to produce a specific signature which is relatively consistent between the fifteen datasets. `Shannon` differentiates itself from the other assemblers on the first (*x*) MDS axis, while the other assemblers (`SPAdes` and `Trinity`) are separated on the second (*y*) MDS axis.

Assembly structure

The structural integrity of each assembly was evaluated using the `TransRate` and `Detonate` software packages. As many downstream applications depend critically on accurate read mapping, assembly quality is correlated with increased mapping rates. The split violin plot presented in Fig. 2A visually represents the mapping rates of each assembly, with lines connecting the mapping rates of datasets assembled with `Trinity` and with the ORP, respectively. The average mapping rate of the `Trinity` assembled datasets was 87% (sd = 8%), while the average mapping rates of the ORP assembled datasets was 93% (sd = 4%). This test is statistically significant (one-sided Wilcoxon rank sum test, $p = 2E\text{-}2$). Mapping rates of the other assemblies are less than that of the ORP assembly, but in most cases, greater than that of the Trinity assembly. This aspect of assembly quality is critical. Specifically mapping rates measure how representative the assembly is of the reads. If I assume that the vast majority of generated reads come

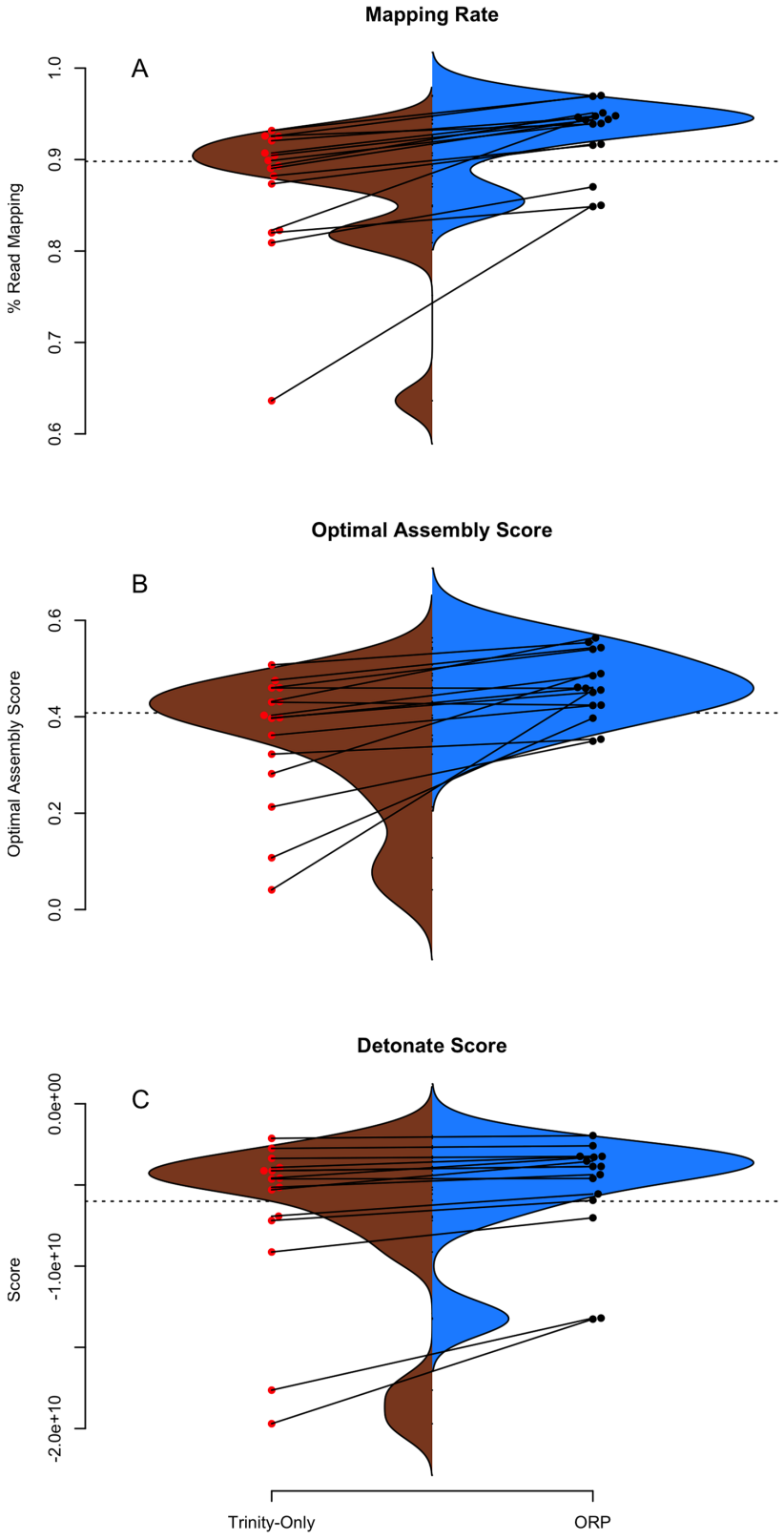
Mapping Rate
A
% Read Mapping
1.0
0.9
0.8
0.7
0.6
Optimal Assembly Score
B
Optimal Assembly Score
0.6
0.4
0.2
0.0
Detonate Score
C
Score
0.0e+00
-1.0e+10
-2.0e+10
Trinity-Only
ORP

Figure 2 TransRate and Detonate generated statistics. (A–C) Split violin plots depict the relationship between Trinity assemblies (brown color) and ORP produced assemblies (blue color). Red and black dots indicate the value of a given metric for each assembly. Lines connecting the red and black dots connect datasets assembled via the two methods. Full-size DOI: 10.7717/peerj.5428/fig-2

from the biological sample under study, when reads fail to map, that fraction of the biology is lost from all downstream analysis and inference. This study demonstrates that across a wide variety of taxa, assembling RNAseq reads with any single assembler alone may result in a decrease in mapping rate and in turn, the lost ability to draw conclusions from that fraction of the sample.

Figure 2B describes the distribution of `TransRate` assembly scores, which is a synthetic metric taking into account the quality of read mapping and coverage-based statistics. The `Trinity` assemblies had an average optimal score of 0.35 (sd = 0.14), while the ORP assembled datasets had an average score of 0.46 (sd = 0.07). This test is statistically significant (one-sided Wilcoxon rank sum test, p-value = 1.8E-2). Optimal scores of the other assemblies are less than that of the ORP assembly, but in most cases, greater than that of the `Trinity` assembly. Figure 2C describes the distribution of `Detonate` scores. The `Trinity` assemblies had an average score of −6.9E9 (sd = 5.2E9), while the ORP assembled datasets had an average score of −5.3E9 (sd = 3.5E9). This test not is statistically significant, though in all cases, relative to all other assemblies, scores of the ORP assemblies are improved (become less negative), indicating that the ORP produced assemblies of higher quality.

In addition to reporting synthetic metrics related to assembly structure, `TransRate` reports individual metrics related to specific elements of assembly quality. One such metric estimates the rate of chimerism, a phenomenon which is known to be problematic in de novo assembly (*Ungaro et al., 2017*; *Singhal, 2013*). Rates of chimerism are relatively constant between all assemblers, ranging from 10% for the `Shannon` assembly, to 12% for the `SPAdes75` assembly. The chimerism rate for the ORP assemblies averaged 10.5% (± 4.7%). While the new method would ideally improve this metric by exclusively selecting non-chimeric transcripts, this does not seem to be the case, and may be related to the inherent shortcomings of short-read transcriptome assembly.

Of note, consistent with all short-read assemblers (*Ungaro et al., 2017*), the ORP assemblies may not accurately reflect the true isoform complexity. Specifically, because of the way that single representative transcripts are chosen from a cluster of related sequences, some transcriptional complexity may be lost. Consider the cluster containing contigs {AB, A, B} where AB is a false-chimera, selecting a single representative transcript with the best score could yield either A or B, thereby excluding an important transcript in the final output. I believe this type of transcript loss is not common, based on how contigs are scored (Table 1; Fig. 3; *Smith-Unna et al., 2016*), though strict demonstration of this is not possible, given the lack of high-quality reference genomes for the majority of the datasets. More generally, mapping rates, `Detonate` and `TransRate` score improvements suggest that this type of loss is not widespread.

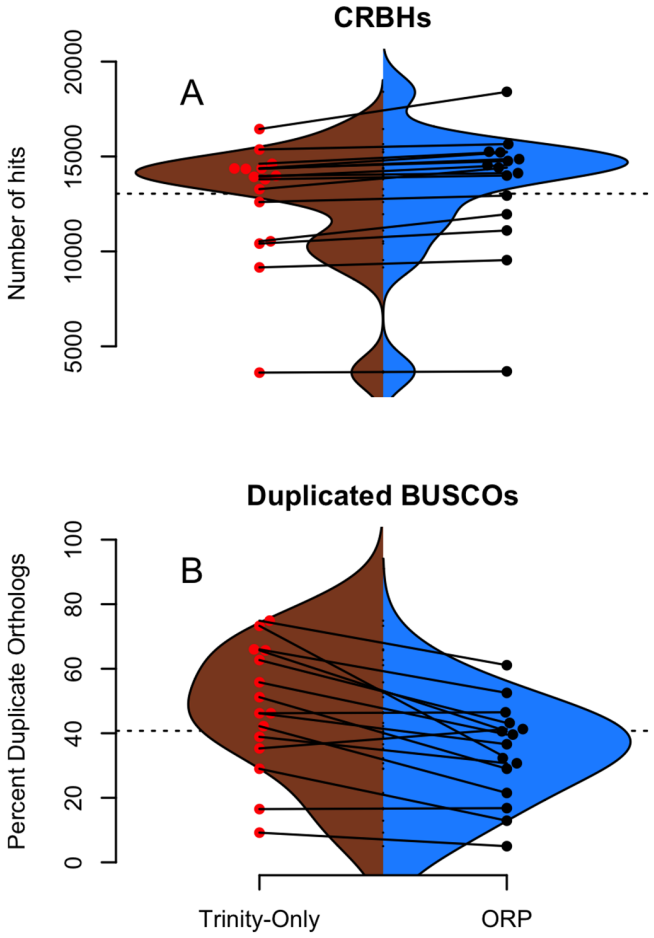

Figure 3 Shmlast and BUSCO generated statistics. (A and B) Split violin plots depict the relationship between Trinity assemblies (brown color) and ORP produced assemblies (blue color). Red and black dots indicate the value of a given metric for each assembly. Lines connecting the red and black dots connect datasets assembled via the two methods. Full-size DOI: 10.7717/peerj.5428/fig-3

Assembly content

The genic content of assemblies was measured using the software package `Shmlast`, which implements the conditional reciprocal blast test against the Swiss-prot database. Presented in Table 2 and in Fig. 3A, ORP assemblies recovered on average 13,364 (sd = 3,391) blast hits, while all other assemblies recovered fewer (minimum `Shannon`, mean = 10,299). In every case across all assemblers, the ORP assembler retained more reciprocal blast hits, though only the comparison between the ORP assembly and `Shannon` was significant (one-sided Wilcoxon rank sum test, $p = 4E\text{-}3$). Notably, in all cases, each assembler was both missing transcripts contained in other assemblies, and contributed unique transcripts to the final merged assembly (Table 2), highlighting the utility of using multiple assemblers.

Regarding `BUSCO` scores, `Trinity` assemblies contained on average 86% (sd = 21%) of the full-length orthologs as defined by the `BUSCO` developers, while the ORP assembled datasets contained on average 86% (sd = 13%) of the full length transcripts. Other

Table 2 **Describes the number of genes contained in the assemblies, with the row labeled concatenated representing the combined average (± standard deviation) number of genes contained in all assemblies of a given dataset.**

Assembly	Genes	Delta	Unique
Concatenated	14,674 ± 3,590		
SPAdes55		−1,739 ± 758	570 ± 266
SPAdes75		−2,711 ± 2,047	301 ± 195
Shannon		−4,375 ± 3,508	302 ± 241
Trinity		−1,952 ± 803	520 ± 301

Note:
The other rows contain information about each assembly. The column labeled delta contains the average number (± standard deviation) of genes missing, relative to the concatenated number. The unique column contains the average number of genes (± standard deviation) unique to that assembly.

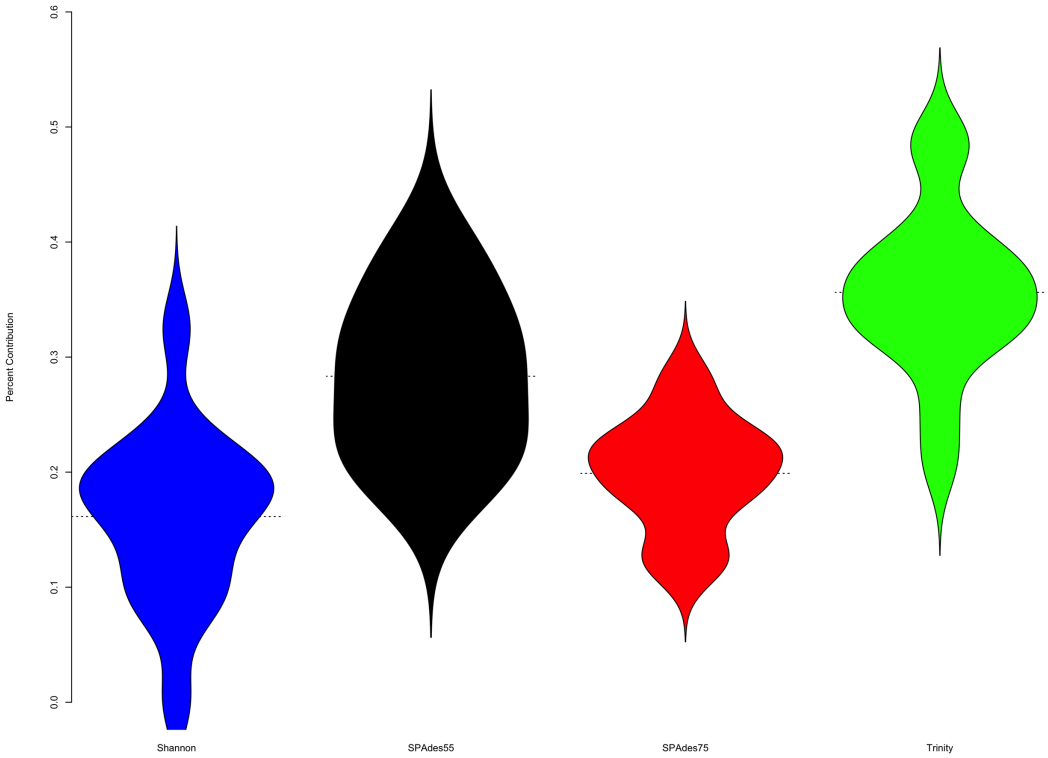

Figure 4 **Plot describes the percent contribution of each assembler to the final ORP assembly.** The proportion of the final transcripts contained in the merged assembly that are a product of each assembler is shown. Violin plots illustrate that Shannon contributes on average the fewest number of transcripts (<20% of transcripts) to the final merged assembly, while Trinity contributes on average the most. Small dashed lines on each side of the plot mark the median of the distribution.
Full-size DOI: 10.7717/peerj.5428/fig-4

assemblers contained fewer full-length orthologs. The `Trinity` and ORP assemblies were missing, on average 4.5% (sd = 8.7%) of orthologs. The `Trinity` assembled datasets contained 9.5% (sd = 17%) of fragmented transcripts while the ORP assemblies each contained on average 9.4% (sd = 9%) of fragmented orthologs. The other assemblers in all cases contained more fragmentation. The rate of transcript duplication, depicted in Fig. 3B is 47% (sd = 20%) for `Trinity` assemblies, and 34% (sd = 15%) for ORP

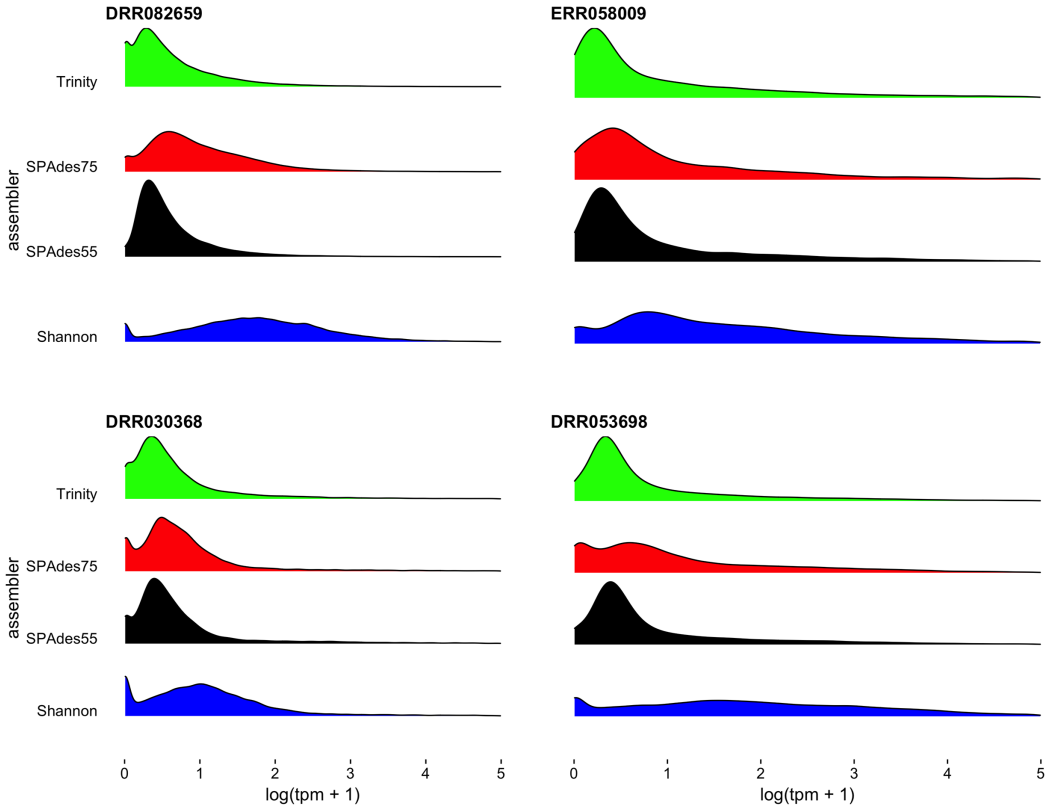

Figure 5 Distribution of gene expression for each assembler. Distribution of gene expression (log(TPM+1)), broken down by individual assembly, for four representative datasets are shown. As predicted, the use of a higher kmer value with the SPAdes assembler resulted in biasing reconstruction toward more highly expressed transcripts. Interestingly, Shannon uniquely exhibits a bias towards the reconstruction of high-expression transcripts (or away from low-abundance transcripts).

Full-size DOI: 10.7717/peerj.5428/fig-5

assemblies. This result is statistically significant (One sided Wilcoxon rank sum test, p-value = 0.02). Of note, all other assemblers produce less transcript duplication than does the ORP assembly, but none of these differences arise to the level of statistical significance.

While the majority of the `BUSCO` metrics were unchanged, the number of orthologs recovered in duplicate (>1 copy), was decreased when using the ORP. This difference is important, given that the relative frequency of transcript duplication may have important implications for downstream abundance estimation, with less duplication potentially resulting in more accurate estimation. Although gene expression quantitation software (*Patro et al., 2017*; *Bray et al., 2016*) probabilistically assigns reads to transcripts in an attempt at mitigating this issue, a primary solution related to decreasing artificial transcript duplication could offer significant advantages.

Assembler contributions

To understand the relative contribution of each assembler to the final merged assembly produced by the ORP, I counted the number of transcripts in the final merged assembly

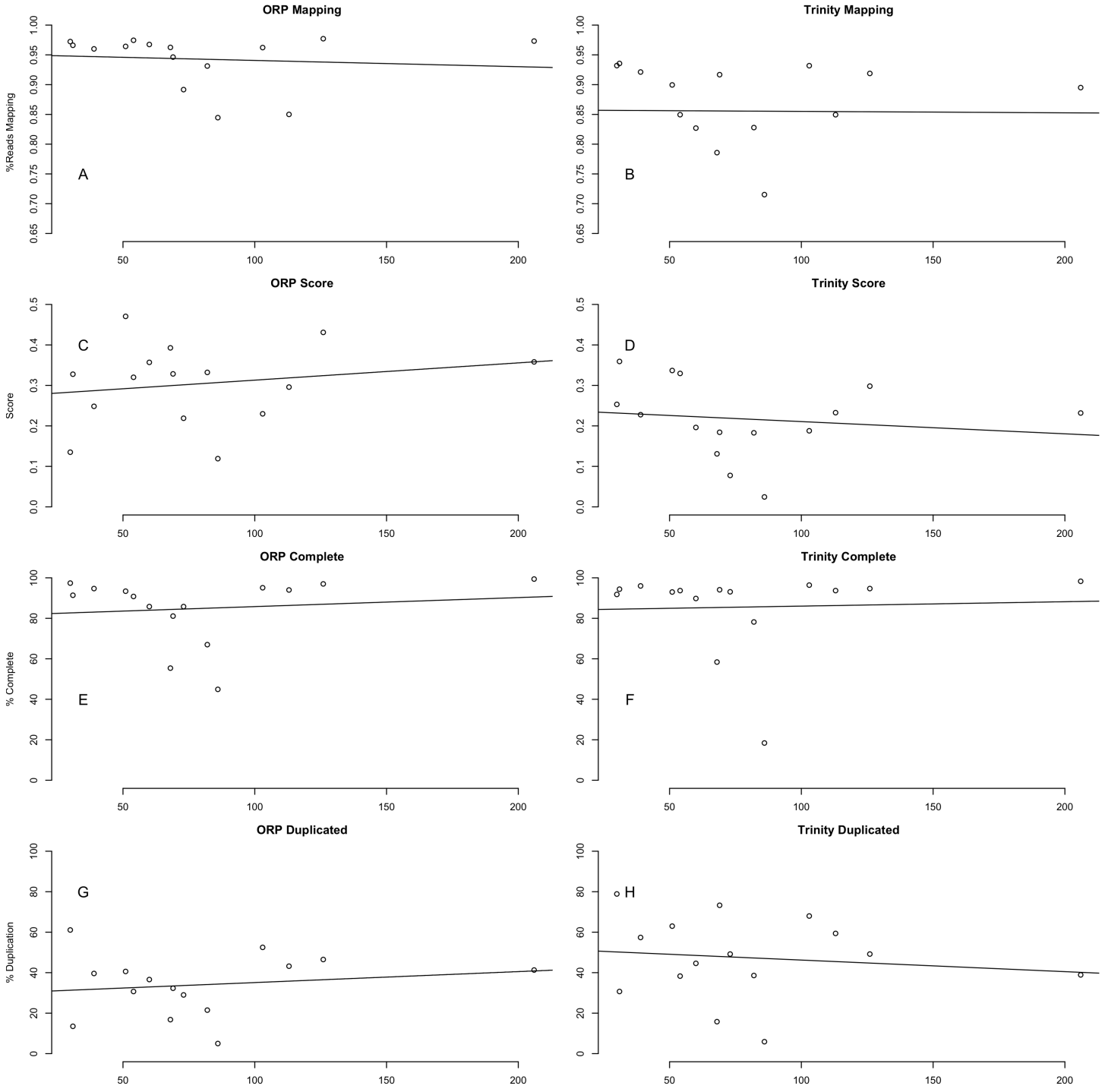

Figure 6 No relationship between metrics and dataset size. The relationship between a subset of assembly metrics and the number of read pairs are shown and is not significant. (A) ORP mapping; (B) Trinity mapping; (C) ORP score; (D) Trinity score; (E) ORP complete; (F) Trinity complete; (G) ORP duplicated; (H) Trinity duplicated. In all cases the *x*-axis is millions of paired-end reads. Full-size DOI: 10.7717/peerj.5428/fig-6

that originated from a given assembler (Fig. 4). On average, 36% of transcripts in the merged assembly were produced by the `Trinity` assembler. A total of 16% were produced by `Shannon`. `SPAdes` run with a kmer value of length = 55 produced 28% of transcripts, while `SPAdes` run with a kmer value of length = 75 produced 20% of transcripts.

To further understand the potential biases intrinsic to each assembler, I plotted the distribution of gene expression estimates for each merged assembly, broken down by the assembler of origin (Fig. 5, depicting four randomly selected representative assemblies). As is evident, most transcripts are lowly expressed, with `SPAdes` and `Trinity` both doing a sufficient job in reconstructing these transcripts. Of note, the `SPAdes` assemblies using kmer-length = 75 is biased, as expected, toward more highly expressed transcripts relative to kmer-length 55 assemblies. `Shannon` demonstrates a unique profile, consisting of, almost exclusively high-expression transcripts, showing a previously undescribed bias against low-abundance transcripts. These differences may reflect a set of assembler-specific heuristics which translate into differential recovery of distinct fractions of the transcript community. Figure 5 and Table 2 describe the outcomes of these processes in terms of transcript recovery. Taken together, these expression profiles suggest a mechanism by which the ORP outperforms single-assembler assemblies. While there is substantial overlap in transcript recovery, each assembler recovers unique transcripts (Table 2; Fig. 5) based on expression (and potentially other properties), which when merged together into a final assembly, increases the completeness.

Quality is independent of read depth

This study included read datasets of a variety of sizes. Because of this, I was interested in understanding if the number of reads used in assembly was strongly related to the quality of the resultant assembly. Conclusively, this study demonstrates that between 30 million paired-end reads and 200 million paired-end reads, no strong patterns in quality are evident (Fig. 6). This finding is in line with previous work (*MacManes, 2015*), suggesting that assembly metrics plateau at between 20 and 40 M read pairs, with sequencing beyond this level resulting in minimal gain in performance.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

The author received no funding for this work.

Competing Interests

The author declares that they have no competing interests.

Author Contributions

- Matthew D. MacManes conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, authored or reviewed drafts of the paper, approved the final draft.

Data Availability

The following information was supplied regarding data availability:

GitHub: https://github.com/macmanes-lab/Oyster_River_Protocol

Zenodo: DOI 10.5281/zenodo.1320141.

REFERENCES

Aubry S, Kelly S, Kümpers BMC, Smith-Unna RD, Hibberd JM. 2014. Deep evolutionary comparison of gene expression identifies parallel recruitment of trans-factors in two independent origins of c4 photosynthesis. *PLOS Genetics* **10(6)**:e1004365 DOI 10.1371/journal.pgen.1004365.

Bolger AM, Lohse M, Usadel B. 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30(15)**:2114–2120 DOI 10.1093/bioinformatics/btu170.

Bray NL, Pimentel H, Melsted P, Pachter L. 2016. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology* **34(5)**:525–527 DOI 10.1038/nbt.3519.

Cahoy JD, Emery B, Kaushal A, Foo LC, Zamanian JL, Christopherson KS, Xing Y, Lubischer JL, Krieg PA, Krupenko SA, Thompson WJ, Barres BA. 2008. A transcriptome database for astrocytes, neurons, and oligodendrocytes: a new resource for understanding brain development and function. *Journal of Neuroscience* **28(1)**:264–278 DOI 10.1523/jneurosci.4178-07.2008.

Chikhi R, Medvedev P. 2014. Informed and automated k-mer size selection for genome assembly. *Bioinformatics* **30(1)**:31–37 DOI 10.1093/bioinformatics/btt310.

Emms DM, Kelly S. 2015. OrthoFinder: solving fundamental biases in whole genome comparisons dramatically improves orthogroup inference accuracy. *Genome Biology* **16(1)**:157 DOI 10.1186/s13059-015-0721-2.

Finseth FR, Harrison RG. 2014. A comparison of next generation sequencing technologies for transcriptome assembly and utility for RNA-seq in a non-model bird. *PLOS ONE* **9(10)**:e108550 DOI 10.1371/journal.pone.0108550.

Fitzpatrick M, Ben-Shahar Y, Vet L, Smid H, Robinson GE, Sokolowski M. 2005. Candidate genes for behavioural ecology. *Trends In Ecology & Evolution* **20(2)**:96–104 DOI 10.1016/j.tree.2004.11.017.

Haas BJ, Papanicolaou A, Yassour M, Grabherr M, Blood PD, Bowden J, Couger MB, Eccles D, Li B, Lieber M, MacManes MD, Ott M, Orvis J, Pochet N, Strozzi F, Weeks N, Westerman R, William T, Dewey CN, Henschel R, Leduc RD, Friedman N, Regev A. 2013. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature Protocols* **8(8)**:1494–1512 DOI 10.1038/nprot.2013.084.

Jackman S, Birol I. 2016. Linuxbrew and Homebrew for cross-platform package management [version 1; not peer reviewed]. *F1000Research* **5(ISCB Comm J)**:1795 DOI 10.7490/f1000research.1112681.1.

Jiang H, Lei R, Ding S-W, Zhu S. 2014. Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics* **15(1)**:182 DOI 10.1186/1471-2105-15-182.

Kampstra P. 2008. Beanplot: a boxplot alternative for visual comparison of distributions. *Journal of Statistical Software* **28(1)**:1–9 DOI 10.18637/jss.v028.c01.

Kannan S, Hui J, Mazooji K, Pachter L, Tse D. 2016. Shannon: an information-optimal de novo RNA-seq assembler. *bioRxiv preprint*.

Lappalainen T, Sammeth M, Friedländer MR, 't Hoen PAC, Monlong J, Rivas MA, Gonzàlez-Porta M, Kurbatova N, Griebel T, Ferreira PG, Barann M, Wieland T, Greger L,

Van Iterson M, Almlöf J, Ribeca P, Pulyakhina I, Esser D, Giger T, Tikhonov A, Sultan M, Bertier G, MacArthur DG, Lek M, Lizano E, Buermans HPJ, Padioleau I, Schwarzmayr T, Karlberg O, Ongen H, Kilpinen H, Beltran S, Gut M, Kahlem K, Amstislavskiy V, Stegle O, Pirinen M, Montgomery SB, Donnelly P, McCarthy MI, Flicek P, Strom TM, Geuvadis Consortium, Lehrach H, Schreiber S, Sudbrak R, Carracedo A, Antonarakis SE, Häsler R, Syvänen A-C, Van Ommen G-J, Brazma A, Meitinger T, Rosenstiel P, Guigo R, Gut IG, Estivill X, Dermitzakis ET. 2013. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature* **501(7468)**:506–511 DOI 10.1038/nature12531.

Le HS, Schulz MH, McCauley BM, Hinman VF, Bar-Joseph Z. 2013. Probabilistic error correction for RNA sequencing. *Nucleic Acids Research* **41(10)**:e109 DOI 10.1093/nar/gkt215.

Li B, Fillmore N, Bai Y, Collins M, Thomson JA, Stewart R, Dewey CN. 2014. Evaluation of de novo transcriptome assemblies from RNA-seq data. *Genome Biology* **15(12)**:553 DOI 10.1186/s13059-014-0553-5.

Li X, Kim Y, Tsang EK, Davis JR, Damani FN, Chiang C, Hess GT, Zappala Z, Strober BJ, Scott AJ, Li A, Ganna A, Bassik MC, Merker JD, GTEx Consortium, Hall IM, Battle A, Montgomery SB. 2017. The impact of rare variation on gene expression across tissues. *Nature* **550(7675)**:239–243 DOI 10.1038/nature24267.

Li R, Li Y, Kristiansen K, Wang J. 2008. SOAP: short oligonucleotide alignment program. *Bioinformatics* **24(5)**:713–714 DOI 10.1093/bioinformatics/btn025.

Liu J, Li G, Chang Z, Yu T, Liu B, McMullen R, Chen P, Huang X. 2016. BinPacker: packing-based de novo transcriptome assembly from RNA-seq data. *PLOS Computational Biology* **12(2)**:e1004772 DOI 10.1371/journal.pcbi.1004772.

Love MI, Huber W, Anders S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology* **15(12)**:550 DOI 10.1186/s13059-014-0550-8.

MacManes MD. 2014. On the optimal trimming of high-throughput mRNA sequence data. *Frontiers in Genetics* **5**:13 DOI 10.3389/fgene.2014.00013.

MacManes MD. 2015. Establishing evidenced-based best practice for the de novo assembly and evaluation of transcriptomes from non-model organisms. *bioRxiv preprint* DOI 10.1101/035642.

MacManes MD, Eisen MB. 2013. Improving transcriptome assembly through error correction of high-throughput sequence reads. *PeerJ* **1**:e113 DOI 10.7717/peerj.113.

Marchant A, Mougel F, Mendonça V, Quartier M, Jacquin-Joly E, Da Rosa JA, Petit E, Harry M. 2016. Comparing de novo and reference-based transcriptome assembly strategies by applying them to the blood-sucking bug *Rhodnius prolixus*. *Insect Biochemistry and Molecular Biology* **69**:25–33 DOI 10.1016/j.ibmb.2015.05.009.

Martin M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17(1)**:10 DOI 10.14806/ej.17.1.200.

Moreton J, Izquierdo A, Emes RD. 2015. Assembly, assessment, and availability of de novo generated eukaryotic transcriptomes. *Frontiers in Genetics* **6(217)**:361 DOI 10.3389/fgene.2015.00361.

Mortazavi A, Williams BA, Mccue K, Schaeffer L, Wold B. 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods* **5(7)**:621–628 DOI 10.1038/nmeth.1226.

Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. 2016. Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biology* **17(1)**:132 DOI 10.1186/s13059-016-0997-x.

Panhuis TM. 2006. Molecular evolution and population genetic analysis of candidate female reproductive genes in *Drosophila*. *Genetics* **173(4)**:2039–2047 DOI 10.1534/genetics.105.053611.

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. 2017. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods* **14(4)**:417–419 DOI 10.1038/nmeth.4197.

Pell J, Hintze A, Canino-Koning R, Howe A, Tiedje JM, Brown CT. 2012. Scaling metagenome sequence assembly with probabilistic *de Bruijn* graphs. *Proceedings of the National Academy of Sciences of the United States of America* **109(33)**:13272–13277 DOI 10.1073/pnas.1121464109.

Peng Y, Leung HCM, Yiu S-M, Lv M-J, Zhu X-G, Chin FYL. 2013. IDBA-tran: a more robust de novo *de Bruijn* graph assembler for transcriptomes with uneven expression levels. *Bioinformatics* **29(13)**:i326–i334 DOI 10.1093/bioinformatics/btt219.

R Core Development Team. 2011. *R: A Language and Environment for Statistical Computing*. Vienna: The R Foundation for Statistical Computing. *Available at http://www.R-project.org/*.

Robertson G, Schein J, Chiu R, Corbett R, Field M, Jackman SD, Mungall K, Lee S, Okada HM, Qian JQ, Griffith M, Raymond A, Thiessen N, Cezard T, Butterfield YS, Newsome R, Chan SK, She R, Varhol R, Kamoh B, Prabhu A-L, Tam A, Zhao Y, Moore RA, Hirst M, Marra MA, Jones SJM, Hoodless PA, Birol I. 2010. De novo assembly and analysis of RNA-seq data. *Nature Methods* **7(11)**:909–912 DOI 10.1038/nmeth.1517.

Robinson MD, McCarthy DJ, Smyth GK. 2010. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26(1)**:139–140 DOI 10.1093/bioinformatics/btp616.

Schulz MH, Zerbino DR, Vingron M, Birney E. 2012. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* **28(8)**:1086–1092 DOI 10.1093/bioinformatics/bts094.

Scott C. 2017. shmlast: an improved implementation of conditional reciprocal best hits with LAST and Python. *Journal of Open Source Software* **2(9)**:142 DOI 10.21105/joss.00142.

Simão FA, Waterhouse RM, Ioannidis P, Kriventseva EV, Zdobnov EM. 2015. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* **31(19)**:3210–3212 DOI 10.1093/bioinformatics/btv351.

Simpson JT, Wong K, Jackman SD, Schein JE, Jones SJM, Birol I. 2009. ABySS: a parallel assembler for short read sequence data. *Genome Research* **19(6)**:1117–1123 DOI 10.1101/gr.089532.108.

Singhal S. 2013. De novo transcriptomic analyses for non-model organisms: an evaluation of methods across a multi-species data set. *Molecular Ecology Resources* **13(3)**:403–416 DOI 10.1111/1755-0998.12077.

Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S. 2016. TransRate: reference-free quality assessment of de novo transcriptome assemblies. *Genome Research* **26(8)**:1134–1144 DOI 10.1101/gr.196469.115.

Song L, Florea L. 2015. Rcorrector: efficient and accurate error correction for Illumina RNA-seq reads. *GigaScience* **4(1)**:1–8 DOI 10.1186/s13742-015-0089-y.

Tan MH, Li Q, Shanmugam R, Piskol R, Kohler J, Young AN, Liu KI, Zhang R, Ramaswami G, Ariyoshi K, Gupte A, Keegan LP, George CX, Ramu A, Huang N, Pollina EA, Leeman DS, Rustighi A, Goh YPS, GTEx Consortium; Laboratory, Data Analysis and Coordinating Center (LDACC)—Analysis Working Group; Statistical Methods groups—Analysis Working Group; Enhancing GTEx (eGTEx) groups; NIH Common Fund; NIH/NCI; NIH/NHGRI; NIH/NIMH; NIH/NIDA; Biospecimen Collection Source Site—NDRI;

Biospecimen Collection Source Site—RPCI; Biospecimen Core Resource—VARI; Brain Bank Repository—University of Miami Brain Endowment Bank; Leidos Biomedical—Project Management; ELSI Study; Genome Browser Data Integration and Visualization—EBI; Genome Browser Data Integration and Visualization—UCSC Genomics Institute, University of California Santa Cruz, Chawla A, Del Sal G, Peltz G, Brunet A, Conrad DF, Samuel CE, O'Connell MA, Walkley CR, Nishikura K, Li JB. 2017. Dynamic landscape and regulation of RNA editing in mammals. *Nature* **550(7675)**:249–254 DOI 10.1038/nature24041.

Titus Brown C, Irber L. 2016. sourmash: a library for MinHash sketching of DNA. *Journal of Open Source Software* **1(5)**:27 DOI 10.21105/joss.00027.

Ungaro A, Pech N, Martin J-F, McCairns RJS, Mévy J-P, Chappaz R, Gilles A. 2017. Challenges and advances for transcriptome assembly in non-model species. *PLOS ONE* **12(9)**:e0185020 DOI 10.1371/journal.pone.0185020.

Vijay N, Poelstra JW, Künstner A, Wolf JBW. 2013. Challenges and strategies in transcriptome assembly and differential gene expression quantification. A comprehensive in silico assessment of RNA-seq experiments. *Molecular Ecology* **22(3)**:620–634 DOI 10.1111/mec.12014.

Wang Z, Gerstein M, Snyder M. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics* **10(1)**:57–63 DOI 10.1038/nrg2484.

Wang S, Gribskov M. 2017. Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis. *Bioinformatics* **33(3)**:327–333 DOI 10.1093/bioinformatics/btw625.

Wolf JBW. 2013. Principles of transcriptome analysis and gene expression quantification: an RNA-seq tutorial. *Molecular Ecology Resources* **13(4)**:559–572 DOI 10.1111/1755-0998.12109.

Xie Y, Wu G, Tang J, Luo R, Patterson J, Liu S, Huang W, He G, Gu S, Li S, Zhou X, Lam TW, Li Y, Xu X, Wong GK-S, Wang J. 2014. SOAP de novo-trans: de novo transcriptome assembly with short RNA-seq reads. *Bioinformatics* **30(12)**:1660–1666 DOI 10.1093/bioinformatics/btu077.

Yang X, Dorman KS, Aluru S. 2010. Reptile: representative tiling for short read error correction. *Bioinformatics* **26(20)**:2526–2533 DOI 10.1093/bioinformatics/btq468.

Yang Y, Smith SA. 2013. Optimizing de novo assembly of short-read RNA-seq data for phylogenomics. *BMC Genomics* **14(1)**:328 DOI 10.1186/1471-2164-14-328.

Zerbino DR, Birney E. 2008. Velvet: algorithms for de novo short read assembly using *de Bruijn* graphs. *Genome Research* **18(5)**:821–829 DOI 10.1101/gr.074492.107.