An opinionated guide to the proper care and feeding of your transcriptome

Matthew D. MacManes[1],

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

∗ E-mail: macmanes@gmail.com

⊕ Twitter: @macmanes

Abstract

Characterizing transcriptomes in both model and 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. Each step may be accomplished in one of several different ways, using different software packages, each producing different results. This analytical complexity begs the question – *Which method(s) are optimal?* Using reference and non-reference based evaluative methods, I propose a set of guidelines that aim to standardize and facilitate the process of transcriptome assembly. These recommendations include the generation of between 20 million and 40 million sequencing reads from single individual where possible, error correction of reads, gentle quality trimming, assembly filtering using `Transrate` and/or gene expression, annotation using `dammit`, and appropriate reporting. These recommendations have been extensively benchmarked and applied to publicly available transcriptomes, resulting in improvements in both content and contiguity. To facilitate the implementation of the proposed standardized methods, I have released a set of version controlled open-sourced code, The `Oyster River Protocol for Transcriptome Assembly`, available at http://oyster-river-protocol.rtfd.org/.

Introduction

For all biology, modern sequencing technologies has provided for an unprecedented opportunity to gain a deep understanding of genome level processes that underlie a very wide

array of natural phenomenon, from intracellular metabolic processes to global patterns of population variability. Transcriptome sequencing has been influential, particularly in functional genomics, 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. Unlike studies of adaptation based on one or a small number of candidate genes (e.g. (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 (*e.g.,* Robinson et al. (2010); Love et al. (2014)) across experimental treatments has resulted in increased resolution for studies aimed at understanding changes in gene expression.

As a direct result of their widespread popularity, a diverse toolset for the assembly and analysis of transcriptome exists. Notable amongst the wide array of tools include several for quality visualization - `FastQC` (available here) and `SolexaQA` (Cox et al., 2010), read trimming (e.g. `Skewer` (Jiang et al., 2014), `Trimmomatic` (Bolger et al., 2014) and `Cutadapt` (Martin, 2011)), read normalization (`khmer` (Pell et al., 2012)), error correction (Le et al., 2013), assembly (`Trinity` (Haas et al., 2013), `SOAPdenovoTrans` (Xie et al., 2014)), and assembly verification (`Transrate` (Smith-Unna et al., 2015)), `BUSCO` (Simão et al., 2015), and `RSEM-eval` (Li et al., 2014)). The ease with which these tools may be used to produce transcriptome assemblies belies the true complexity underlying the overall process. Indeed, the subtle (and not so subtle) methodological challenges associated with transcriptome reconstruction may result in highly variable assembly quality. Amongst the most challenging include isoform reconstruction and simultaneous assembly of low- and high-coverage transcripts (Modrek et al., 2001; Johnson et al., 2003), which together make accurate transcriptome assembly technically challenging. As in child rearing, production of a respectable transcriptome sequence requires a large investment in time and resources. At every step in development, care must be taken correct, but not overcorrect. Here, I propose a set of guidelines for the care and feeding that will result in the production of an accurate, useful, and well-adjusted transcriptome.

In particular, I focus my efforts on the early- and mid-development of the transcrip-

tome – which, unfortunately are steps often neglected or abused – and reporting. Particularly flagrant are abuses related to the selection and quality control of input data, and lack of appropriate post-assembly quality evaluation. Here, I aim to define a set of evidence based analyses and methods aimed at improving transcriptome assembly, which in turn has significant effects on all downstream analyses.

Methods

To demonstrate the merits of my recommendations, a large number of assemblies were produced using a variety of methods. For all assemblies performed, Illumina sequencing adapters were removed from both ends of the sequencing reads, as were nucleotides with quality Phred ≤ 2, using the program `Trimmomatic` version 0.32 (Bolger et al., 2014). The reads were assembled using Trinity release 2.1.1 (Haas et al., 2013) using default settings. `Trinity` was used as the default assembler as it has been previously reported to be best in class (Li et al., 2014; Smith-Unna et al., 2015). Assemblies were characterized using `Transrate` version 1.0.1 (Smith-Unna et al., 2015). Using this software, I generated three kinds of metrics: contig metrics; mapping metrics which used as input the same reads that were fed into the assembler for each assembly; and comparative metrics which used as input the *Mus musculus* version 75 transcriptome. In addition to the metrics provided by `Transrate`, I evaluated completeness of each assembly by use of `BUSCO`, a software package that searches for highly conserved, near-universal, single copy orthologs.

To understand the influence of read depth on assembly quality, I produced subsets of size 1,2,5,10,20,40,60,80,100 million paired end reads of two publicly available paired-end datasets - A *Mus* dataset -SRR797058 described in Macfarlan et al. (2012) and a human dataset - SRR1659968. The subsampling procedure was accomplished via the software package `seqtk` (`https://github.com/lh3/seqtk`). For the evaluation of the effects of sequence polymorphism on assembly quality, I use reads from BioProject PRJNA157895 described in Macmanes and Lacey (2012), a *Ctenomys* dataset which consists of 10 read files from the hypothalami of 10 different individuals. This dataset was assembled two ways. First, the reads from all 10 individuals were jointly assembled in one large assembly [CODE]. This assembly was compared to the assembly of a single individual [CODE]. Assemblies were generated and evaluated as per above.

To evaluate the effects of error correction, I used the subsampled read datasets, which were subsequently error corrected using the following software packages: `SEECER` version 0.1.3 (Le et al., 2013), `Lighter` version 1.0.7 (Song et al., 2014), `SGA` version 0.10.13 (Simpson and Durbin, 2012), `bfc` version r177 (Li, 2015), `RCorrector` (Song and Florea, 2015),and `BLESS` version 0.24 (Heo et al., 2014). In correction algorithms (`SGA`, `BLESS`, `bfc`) that allowed for the use of larger *kmer* lengths, I elected to error correct with a small ($k = 31$) and a long ($k = 55$) *kmer*, while for the other software (`RCorrector`, `SEECER` and `Lighter`) that does not allow for longer *kmer* values, I set $k = 31$. `bfc` requires interleaved reads, which was accomplished using khmer version 2.0 (Brown et al., 2015, 2012; McDonald and Brown, 2013). Code for performing these steps is available [here].

The effects of `khmer` digital normalization (Pell et al., 2012) were characterized by generating three 20 million, 40 million, and 80 million read subsets of the larger *Mus* dataset. Digital normalization was performed using a median kmer abundance threshold of 30. The resulting datasets were assembled using `Trinity`, and evaluated using `BUSCO` and `Transrate`. Code for performing these steps is available in the `diginorm` target of the [Makefile].

Post-assembly processing was evaluated using several assembly datasets of various sizes, generated above. Each assembly was evaluated using `Transrate`. `Transrate` produces a score based on contig and mapping metrics, as well as a more optimal assembly where poorly supported contigs (putative assembly artifacts) are removed. Both the original and `Transrate` optimal assembly are evaluated using `BUSCO`, to help better understand if filtration results in the loss of non-artifactual transcripts. In addition to `Transrate` filtration, an additional, or alternative filtration step is performed using estimates of gene expression (TPM=transcripts per million). TPM is estimated by two different software packages that implement two distinct methods - `Salmon` (Patro et al., 2015) and `Kallisto` (Bray et al., 2015). Transcripts whose expression is estimated to be greater than a given threshold, typically TPM=1 or TPM=0.5 are retained. As above, the filtered assemblies are evaluated using `BUSCO`, to help better understand if filtration results in the loss of non-artifactual transcripts. Code for performing these steps is available in the `QC` target of the makefile available [here].

Recommendations

0.1 Input Data

Summary Statement: Sequence 1 or more tissues from 1 individual to a depth of between 20 million and 40 million 100bp or longer paired-end reads.

When planning to construct a transcriptome, the first question to ponder is the type and quantity of data required. While this will be somewhat determined by the specific goals of the study and availability of tissues, there are some general guiding principals. As of 2014, Illumina continues to offer the most flexibility in terms of throughout, analytical tractability, and cost (GLENN, 2011). It is worth noting however, that long-read (e.g. PacBio) transcriptome sequencing is just beginning to emerge as an alternative (Au et al., 2013), particularly for researchers interested in understanding isoform complexity. Though currently lacking the throughput for accurate quantitation of gene expression, long read technologies, much like they have done for *de novo* genome assembly, seem likely to replace short-read-based *de novo* transcriptome assembly at some point in the future.

For the typical transcriptome study, one should plan to generate a reference based on 1 or more tissue types, with each tissue adding unique tissue-specific transcripts and isoforms. Though increasing the amount of sequence data collected does increase the accuracy and completeness of the assembly (Figure 1, 3) albeit marginally, a balance between cost and quality exists. For the datasets examined here (mammal tissues), sequencing more than between 20M and 40M paired-end reads is associated with the discovery of very few additional transcripts, and only minor improvement in other assembly metrics. Read length should be at least 100bp, with longer reads likely aiding in isoform reconstruction and contiguity (Garber et al., 2011).

Because sequence polymorphism increases the complexity of the *de bruijn* graph (Iqbal et al., 2012; Studholme, 2010), and therefore may negatively effect the assembly itself, the reference transcriptome should be generated from reads corresponding to as homogeneous a sample as possible. For outbred, non-model organisms, this usually means generating reads from a single individual. When more then one individual is required to

meet other requirements (e.g. number of reads or experimental treatment conditions), keeping the number of individuals to a minimum is paramount. For instance, when performing an experiment where a distinct set of genes may be expressed in different treatments (or sexes), the recommendation is to sequence one individual from each treatment class.

To illustrate this effect, I examined the effects of assembling reads from 10 individuals jointly, versus assembling a representative individual. This individual was selected based on having the highest number of reads. The individual assembly of 38 million paired end read took approximately 23 hours and 20Gb of RAM, while the joint assembly took five days and 150Gb of RAM. Per Table 1, the joint assembly used more than eight times more reads, and is more than four times larger than the assembly of a single individual. Despite the additional read data, the `Transrate` score is markedly decreased, although the `BUSCO` statistics are slightly better. The large joint assembly suffers from major structural problems that are unfixable via the proposed filtering procedures. Specifically, read-mapping data suggests that 28.7% of the contigs in the joint assembly could be merged, versus 15% in the single assembly. This structural problem is likely the result of sequence polymorphism and may cause significant issues for many common downstream processes.

Table 1

Name	Num. Reads	Num. Contigs	Assembly Size	Score	BUSCO
Single Ind.	38M	205812	131.6Mb	0.3064	C:81%,D:41%,M:9%
10 Ind.	269M	913295	440.2Mb	0.22011	C:88%,D:51%,M:5%

Table 1. A comparison of the raw assemblies resulting from a single individuals versus the joint assembly of 10 individuals. The individual assembly of 38 million reads resulted in an assembly of size 131.6 million bases, a `Transrate` score of 0.3064. 81% of BUSCOs were found to be complete, with 9% missing from the dataset. The joint assembly of 10 individuals, consisting of 269 million paired-end reads resulted in an assembly of size 440.2 million bases and a `Transrate` score of 0.22011.

Figure 1

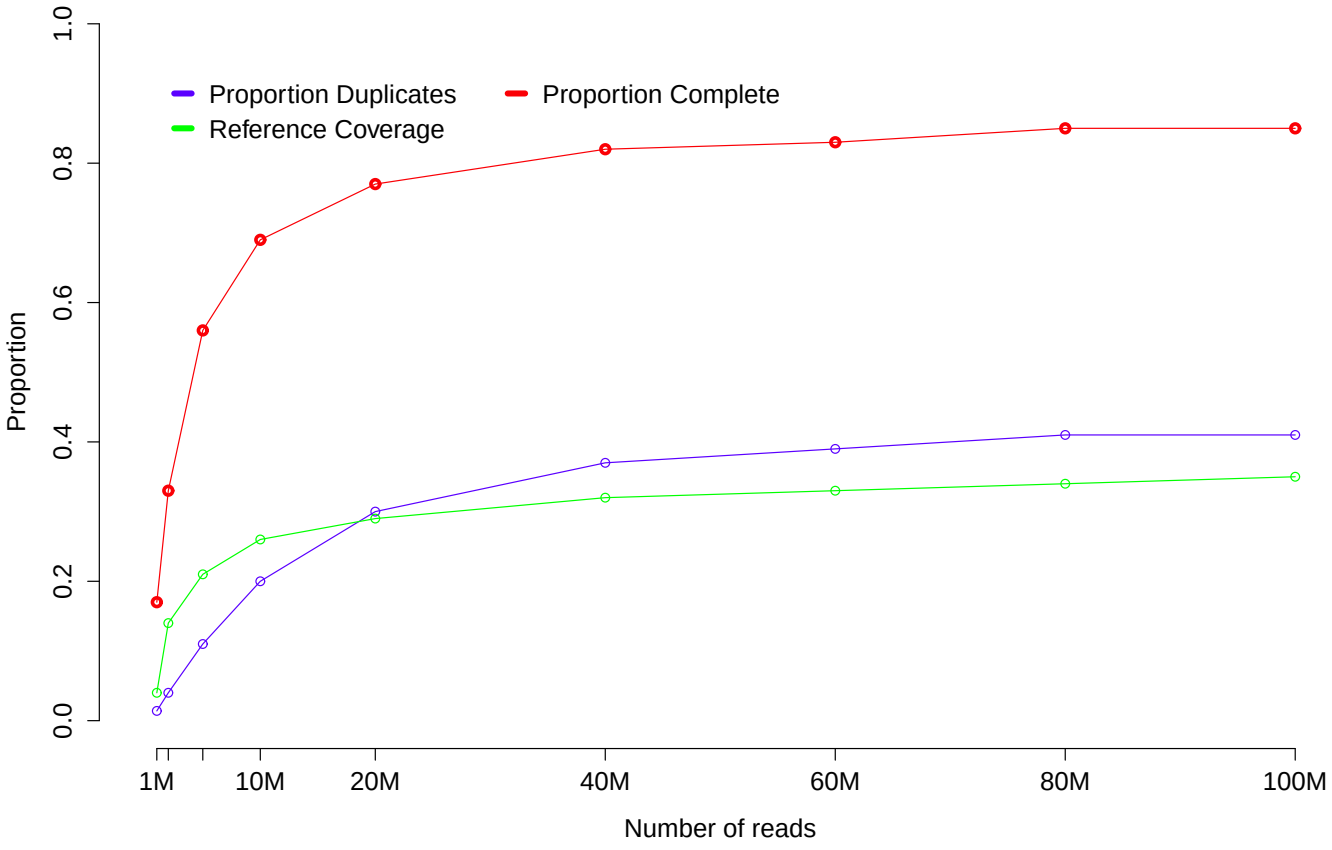

Figure 1. Assembly of multiple subsetted datasets suggests that sequencing beyond 20-40 million paired end reads does not results in further sequence discovery. Proportion complete indicates the proportion of BUSCOs that were found to be full length. Proportion duplicates are those BUSCOs that were found multiple times in the assembly dataset. Reference coverage is a `Transrate` generated metric indicating the proportion of the reference *Mus* transcriptome found in the *de novo* assembly. Higher numbers for reference coverage and proportion complete indicate a more complete assembly.

0.2 Quality Control of Sequence Read Data

Summary Statement:Visualize your read data. Error correct reads using `bfc` for low to moderately sized datasets and `RCorrector` for higher coverage datasets. Remove adapters, and employ gentle quality filtering using PHRED $\leq$ 2 as a threshold.

Before assembly, it is critical that appropriate quality control steps are implemented. It is often helpful to generate some metrics of read quality on the raw data. Several software packages are available – I am fond of `SolexaQA` (Cox et al., 2010) and `FastQC`. Immediately upon download of the read dataset from the sequence provider, metrics of read quality, generated by either of these two software packages, should be generated. Of note – a copy of the raw reads should be compressed and archived, preferably on a physically separated device for long term archival storage. For this, I have successfully used Amazon S3 cloud storage, though many options exist.

Immediately after visualizing the raw data, error correction of the sequencing reads should be done (MacManes and Eisen, 2013). A very large number of read correction software packages exist, and several of them are benchmarked here using the *Mus* (Figure 2, and Tables S1-S11) and *Homo* datasets (Tables S12-S21). In all evaluated datasets, the error correction `bfc` was the best when correcting less than approximately 20M paired-end reads. When correcting more, the software `RCorrector` provided the optimal correction. The effects of error correction on assembly were evaluated using `BUSCO` and `Transrate`. While error correction did not result in significant improvements in `BUSCO` metrics, the transrate scores were substantially improved (Figure 3). These scores were largely improved by the fact that assemblies using error corrected reads had fewer low-covered based and contigs, and a slightly higher mapping rate.

The error corrected reads are then subjected to vigorous adapter sequence removal, typically using `Trimmomatic` (Bolger et al., 2014) or `Skewer` (Jiang et al., 2014). With adapter sequence removal may be a quality trimming step. Here, substantial caution is required, as aggressive trimming has detrimental effects on assembly quality (MacManes, 2014). Specifically, I recommend trimming at Phred=2, a threshold associated with removal of only the lowest quality bases. After adapter removal and quality trimming, the

previously error corrected reads are now ready for *de novo* transcriptome assembly.

Figure 2

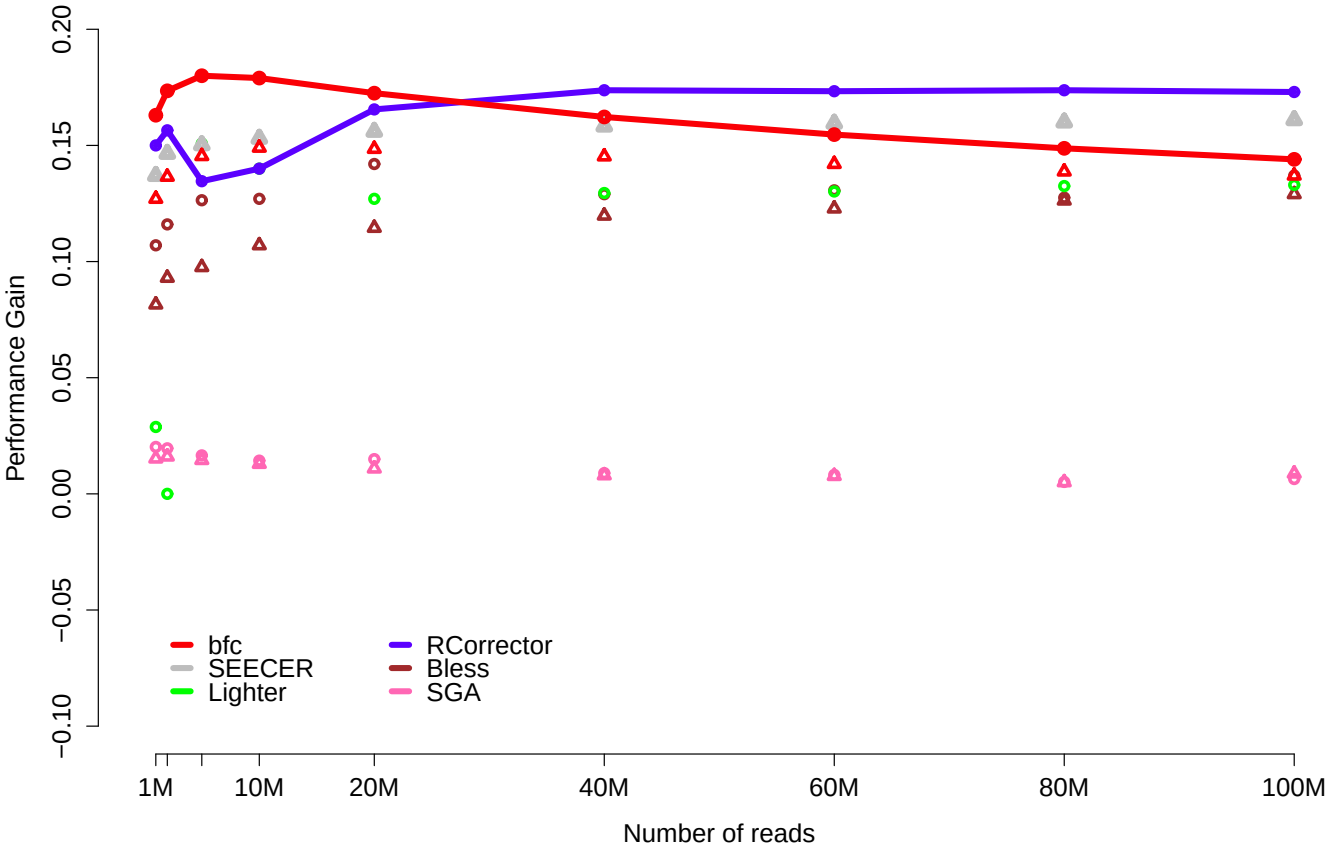

Figure 2. Error correction of reads results in a performance gain (defined as: (perfect error corrected reads - perfect raw reads) + reads made better - reads made worse). Perfect reads are reads that map to the reference without mismatch. Better and worse reads are those that map with fewer or more mismatches. Low coverage datasets are best corrected with `bfc`, which higher coverage datasets are optimally corrected with `RCorrector`. The best performing corrections improve the quality of more than 15% of reads.

Figure 3

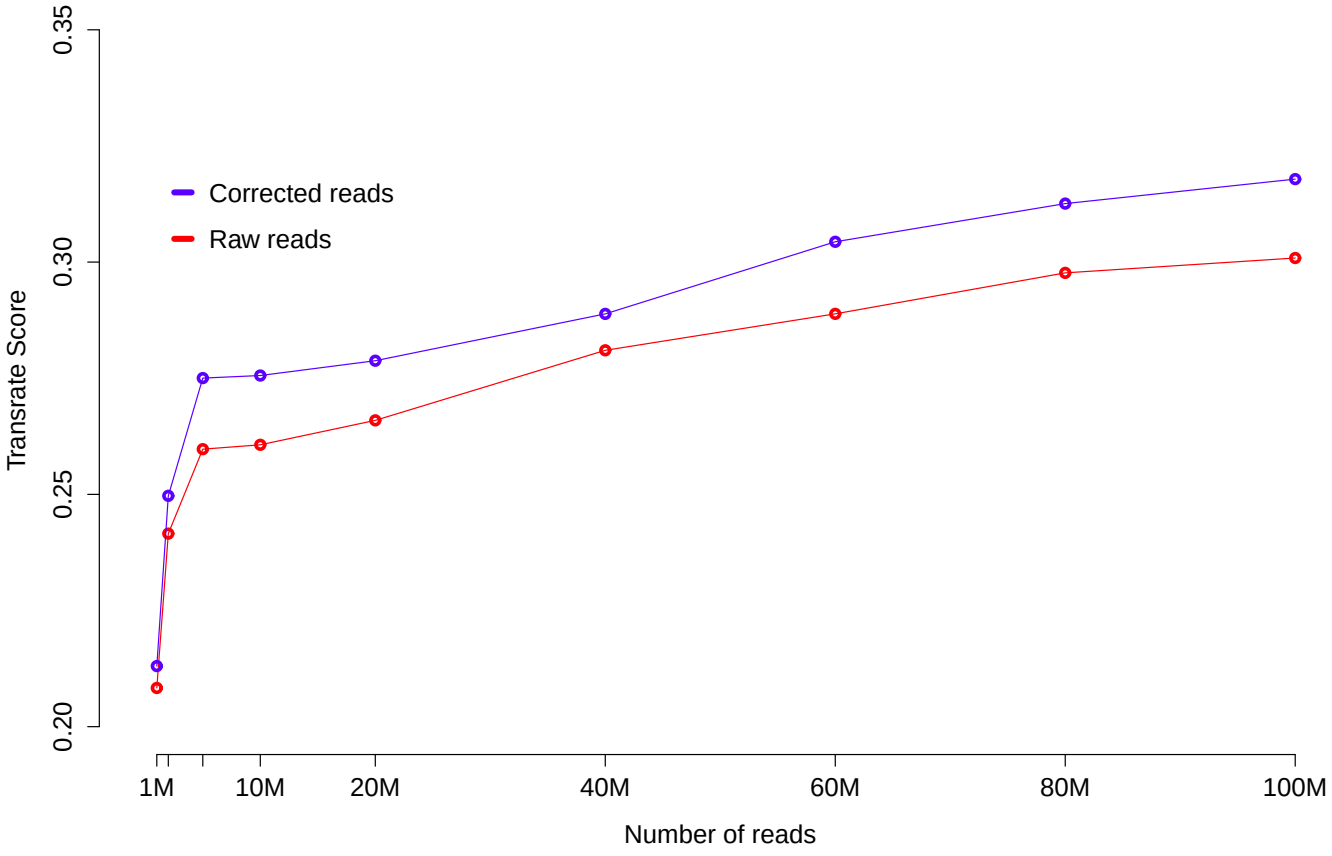

Figure 3. Error correction (with the best performing correction software, described in Figure 2), results in a consistent increase in the `Transrate` score, which indicates a higher quality assembly across all coverage depths.

0.3 Coverage normalization

Summary Statement: Normalize your data, only if you have to.

Depending on the volume of input data, the availability of a high-memory workstation, and the rapidity with which the assembly is needed, coverage normalization may be employed. This process, which, using a streaming algorithm and measurement of the median kmer abundance of each read, aims to erode areas of high coverage while leaving untouched, reads spanning lower coverage areas. Normalization may be accomplished in the software package `khmer` (Pell et al., 2012), or within `Trinity` using a computational algorithm based on `khmer`. In our tests, normalization did dramatically reduce RAM requirements and runtime, though it also decreased the number of complete BUSCO's found by 4%, and the transrate score from 0.266 to 0.251. Given this, our recommendation is to employ digital normalization when the assembly is otherwise impossible, or when results are urgently needed, but that it should not be used by default for the production of transcriptome assemblies.

0.4 Assembly

Summary Statement: Assemble your data using Trinity, then remove poorly supported contigs.

For non-model organisms lacking reference genomic resources, the error correction, adapter and quality trimming reads should be assembled *de novo* into transcripts. Currently, the assembly package `Trinity` (Haas et al., 2013) is thought to currently be the most accurate (Li et al., 2014), and therefore is recommended over other assemblers. While attempting a merged assembly with multiple assemblers may *ultimately* result in the highest quality assembly, options for merging assemblies are currently limited, and therefore is not recommended.

Trinity's underlying algorithm have been pre-optimized to recover large numbers of alternative isoforms, including many that are minimally supported by read data. As a result, in many cases, the raw assembly will require filtration to remove these assembly

artifacts. Reference dependent and independent evaluative tools (*e.g.,* `Transrate`, `BUSCO`) allow for evidence-based post-assembly filtration. Typically, an initial quality-evaluation and filtration step is implemented using Transrate. This process assigns a score to the assembly, and creates an alternative assembly by removing contigs based on read-mapping metrics. This filtration step may result in the removal of a large proportion (as much as 67%) of the transcripts. Reference-based metrics are generated before and after this filtration step to ensure that filtration has not been too aggressive - that a significant number of known transcripts have not been removed. After Transrate filtration, or alternative to it, it is often helpful to employ a filtration step based on TPM. Because underlying assumptions of gene expression estimation software vary, which may results in variation in the actual estimates, gene expression is typically estimated using two different packages, `Salmon` and `Kallisto`. Transcripts whose expression is less than either 1 or 0.5 are removed. Again, reference-based metrics are generated to ensure that a significant number of known transcripts are not removed.

The results of filtration on several datasets of varying size are presented in Table 2. The reads used in the 1M,5M,10M,20M subset assemblies were corrected with `bfc`, while the reads for the larger assemblies were corrected with `RCorrector`. Each dataset was trimmed to a quality of Phred <2, and assembled with `Trinity`. The raw assembly was filtered by `Transrate` and by gene expression. `BUSCO` evaluation was performed before and after these filtration steps. In general, for low coverage datasets (less than 20 million reads), filtering based on expression, using TPM=1 as a threshold performs well, with `Transrate` filtering being too aggressive. With higher coverage data (more than 60 million reads) `Transrate` filtering may be optimal, as may gene expression filtering using a threshold of TPM=0.5.

Name	Subset	BUSCO	Num. Contigs	Assembly Size	Transrate Score
Raw	1M	C:17%[D:1.4%],F:10%,M:72%	29719	9.8Mb	0.21274
Transrate_Filt	1M	C:16%[D:1.2%],F:8.2%,M:75%	9860	9.1Mb	0.31918
TPM_Filt	1M	C:17%[D:1.4%],F:10%,M:72%	29503	9.8Mb	0.21683
Raw	5M	C:56%[D:13%],F:9.8%,M:33%	52611	35.3Mb	0.27401
Transrate_Filt	5M	C:52%[D:11%],F:9.2%,M:38%	21383	31.1Mb	0.39856
TPM_Filt	5M	C:56%[D:12%],F:10%,M:33%	51476	33.9Mb	0.28302
Raw	10M	C:69%[D:23%],F:7.0%,M:23%	72688	52.8Mb	0.27558
Transrate_Filt	10M	C:64%[D:20%],F:6.3%,M:29%	28249	44.7Mb	0.4092
TPM_Filt	10M	C:69%[D:21%],F:7.0%,M:23%	69561	49.2Mb	0.2881
Raw	20M	C:78%[D:32%],F:4.5%,M:17%	108072	76.2Mb	0.27888
Transrate_Filt	20M	C:70%[D:27%],F:4.7%,M:24%	45169	62.1Mb	0.39389
TPM_Filt	20M	C:77%[D:29%],F:4.8%,M:17%	97519	66.8Mb	0.29878
Raw	40M	C:82%[D:38%],F:3.7%, M:14%	163561	107Mb	0.2859
Transrate_Filt	40M	C:74%[D:32%],F:4%, M:21%	91367	85.5Mb	0.3796
TPM_Filt	40M	C:82%[D:32%],F:3.6%, M:14%	117819	83.3Mb	0.3037
Raw	60M	C:84%[D:40%],F:3.2%,M:12%	204040	127Mb	0.29616
Transrate_Filt	60M	C:78%[D:35%],F:3.2%,M:18%	166503	107Mb	0.37018
TPM_Filt	60M	C:82%[D:31%],F:3.3%,M:13%	109485	86.9Mb	0.30128
Raw	80M	C:85%[D:40%],F:3.2%,M:11%	237401	146Mb	0.30139
Transrate_Filt	80M	C:85%[D:39%],F:3.2%,M:11%	222900	132Mb	0.37997
TPM_Filt	80M	C:82%[D:32%],F:2.9%,M:14%	96968	88.5Mb	0.29261
Raw	100M	C:85%[D:41%],F:3.0%,M:11%	264751	159Mb	0.30567
Transrate_Filt	100M	C:85%[D:40%],F:3.1%,M:11%	247413	143.8Mb	0.39242
TPM_Filt	100M	C:83%[D:32%],F:2.7%,M:14%	86993	88.4Mb	0.2828

Table 2. Post-assembly filtration. Using assemblies from the 1M,5M,10M,20M,40M,60M,80M,100M read subsets, I evaluated the effects of `Transrate` and TPM filtration using a threshold of TPM=1. Both `Transrate` and TPM filtering reduced the number of contigs and assembly size, though the magnitudes were dependent on the depth of sequencing. `BUSCO` scores were either decreased in some cases, or stable in others, representing the differential effects of filtering on different sized assemblies. In general, for low coverage datasets (less than 20 million reads), filtering based on expression, using TPM=1 as a threshold performs well, with `Transrate` filtering being too aggressive. With higher coverage data (more than 60 million reads) `Transrate` filtering may perform better, as mat expression filtering with a lower threshold.

0.5 Annotation, post-assembly quality verification, & reporting

Summary Statement: Verify the quality of your assembly using content based metrics. Annotate using `dammit` Report `Transrate` score, `BUSCO` statistics, number of unique transcripts, etc. Do not report meaningless statistics such as N50.

Annotation is a critically important step in transcriptome assembly. Much like other steps, numerous options exist. Though the research requirements may drive the annotation process, I propose that a core set of annotations be provided with all *de novo* transcriptome assembly projects. The process through which these core annotations are accomplished is coordinated by the software package `dammit`. This software takes as input a fasta file and outputs a standard gff3 containing annotations. After annotation, but before downstream use, it is important to assess the quality of a transcriptome. Many authors have attempted to use typical genome assembly quality metrics for this purpose. In particular, N50 and other length-based summary statistic are often reported (e.g. (Hiz et al., 2014; Shinzato et al., 2014; Liang et al., 2013)). However, in addition to being a poor proxy for quality in genome assembly (Bradnam et al., 2013), N50 in the context of a transcriptome assembly carries very little information because the optimal contig length is not known (Li et al., 2014) - real transcripts vary greatly in length, ranging from tens of nucleotides to tens of thousands of nucleotides. Reportable metrics should be chosen based on their relevance for assembly optimization given the biological question at hand. In most cases, this means maximizing the number of transcripts that can be confidently attributed to the organism, while minimizing the number of technical artifacts related to the process of sequencing, quality control, and assembly. For many researchers, this means evaluation with both `BUSCO` and `Transrate`. The statistics found in Table 1 should be presented for all assemblies, with additional information supplementing these core vital statistics as needed.

Testing the Oyster River Protocol

To evaluate the Oyster River Protocol for Transcriptome Assembly, I selected three publicly available Illumina RNAseq datasets and their corresponding assembled tran-

scriptomes. These three assemblies included the Nile Tilapia, *Oreochromis niloticus* (Zhang et al. (2013), SRR797490), an unpublished study of the Mediterranean black widow, *Latrodectus tredecimguttatus* (SRR954929), and lastly a work on *Delia antiqua* (Guo et al. (2015), SRR916227). I analyzed the original transcriptomes using both `BUSCO` and `Transrate`, then followed the protocol as described here. Code for data analysis of the *Oreochromis* is available here. The other samples were processed in an identical fashion. The application of the Oyster River Protocol on these datasets resulted universally in a substantial (as much as 22%) improvement in the completeness of assemblies. Given a major goal of these types of studies includes reconstruction all expressed genes, this improvement may have substantial improvement on downstream work. The `Transrate` score was dramatically improved as well, particularly in the *Oreochromis* and *Delia* assemblies. This improvement speaks to the improvement of the structure of the assembly.

The filtering process through which these more optimal assemblies were is key. Evaluating both the `BUSCO` and `Transrate` scores before and after, allows for an objective way to decide if filtering has been too restrictive or not. Indeed, for the *Latrodectus* assembly, both `Transrate` and TPM filtering reduced the `BUSCO` score, while substantially increasing the `Transrate` score. Depending on the goals of the experiment, it may be determined that the structural integrity of the assembly outweighs improved content. In contrast to how post-assembly filtering is typically done, this method allow for the researcher to make an informed decision about these processes.

Table 3

Name	Number Reads	Number Contigs	Assembly Size (Mb)	Transrate Score	BUSCO Score
Oreochromis	25.2M	79198/140035/100376/116038/88456	32.0/75.1/69.5/58.6/57.7	0.1103/0.2173/0.4778/0.2595/0.4479	C:39%,M:46%/C:58%M:28%/C:57%,M:30%/C:57%,M:30%/C:56%,M:31%
Latrodectus	27.6M	10259/36394/30932/27973/NA	10.6/13.5/13.1/10.9/NA	0.43673/0.2795/0.4968/0.338/NA	C:48%,M:38%/C:58%,M:28%/C:46%,M:39%/C:46%,M:41%/NA
Delia	25.8M	29451/49099/38614/46145/32689	12.4/19.3/18.8/17.9/15.8	0.393/0.2036/0.4572/0.2305/0.4341	C:40%,M:48%/C:62%,M:21%/C:61%,M:23%/C:61%,M:23%/C:61%,M:25%

Table 3. The results of the application of the Oyster River Protocol to three available transcriptomes. Within each column, the 5 metrics, separated by forward slashes are: 1. The original assembly 2. The raw `Trinity` assembly 3. The `Transrate` filtered assembly 4. The TPM=1 filtered assembly, and 5. The Transrate filtered assembly that has been further filtered by expression. In all cases the assembly content, as evaluated by the `BUSCO` score is dramatically improved over the original assembly. These content-improved assemblies have acceptable `Transrate` scores, which in 2 of 3 cases are vastly superior to the scores of the original assembly.

Conclusions

With the rapid adoption of high-throughput sequencing, studies of functional, evolutionary and population genomics are now being done by hundreds or even thousands of labs around the world. These studies typically begin with a *de novo* transcriptome assembly. Assembly may be accomplished in one of several different ways, using different software packages, with each method producing different results. This complexity begs the question – *Which method(s) are optimal?* Using reference and non-reference based evaluative methods, I have proposed a set of guidelines `The Oyster River Protocol for Transcriptome Assembly` that aim to standardize and facilitate the process of transcriptome assembly. These recommendations include limiting assembly to between 20 million and 40 million sequencing reads from single individual where possible, error correction of reads, gently quality trimming, assembly filtering using `Transrate` or gene expression, annotation using `dammit`, and appropriate reporting. The processes result in a high quality transcriptome assembly appropriate for downstream usage. Assemblies generated in the process of developing this protocol are available here.

Acknowledgments

This work was significantly impacted by numerous discussions with C. Titus Brown, Richard Smith-Unna, Camille Scott, and many others. More generally, the work and it's presentation has been influenced by supporters of the Open Access and Science movements.

References

Au, K. F., Sebastiano, V., Afshar, P. T., Durruthy, J. D., Lee, L., Williams, B. A., van Bakel, H., Schadt, E. E., Reijo-Pera, R. A., Underwood, J. G., and Wong, W. H. (2013). Characterization of the human ESC transcriptome by hybrid sequencing. *PNAS*, 110(50):201320101–30.

Bolger, A. M., Lohse, M., and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):btu170–2120.

Bradnam, K. R., Fass, J. N., Alexandrov, A., Baranay, P., Bechner, M., Birol, I., Boisvert,

394 S., Chapman, J. A., Chapuis, G., Chikhi, R., Chitsaz, H., Chitsaz, H., Chou, W.-C.,
395 Corbeil, J., Del Fabbro, C., Docking, T. R., Durbin, R., Earl, D., Emrich, S., Fedotov,
396 P., Fonseca, N. A., Ganapathy, G., Gibbs, R. A., Gnerre, S., Gnerre, S., Godzaridis, E.,
397 Goldstein, S., Haimel, M., Hall, G., Haussler, D., Hiatt, J. B., Ho, I. Y., Hiatt, J. B.,
398 Hunt, M., Howard, J., Hunt, M., Jackman, S. D., Jaffe, D. B., Jaffe, D. B., Jiang, H.,
399 Jarvis, E. D., Kazakov, S., Kazakov, S., Kersey, P. J., Kersey, P. J., Kitzman, J. O.,
400 Kitzman, J. O., Koren, S., Knight, J. R., Koren, S., Lam, T.-W., Lavenier, D., Lavenier,
401 D., Laviolette, F., Li, Y., Laviolette, F., Li, Z., Li, Y., Liu, B., Liu, B., Liu, Y., Liu,
402 Y., Maccallum, I., Luo, R., Maccallum, I., MacManes, M. D., Maillet, N., Melnikov,
403 S., Naquin, D., Ning, Z., Ning, Z., Otto, T. D., Otto, T. D., Paten, B., Paulo, O. S.,
404 Paulo, O. S., Phillippy, A. M., Pina-Martins, F., Place, M., Place, M., Przybylski, D.,
405 Przybylski, D., Qin, X., Qu, C., Ribeiro, F. J., Richards, S., Richards, S., Rokhsar,
406 D. S., Ruby, J. G., Scalabrin, S., Schatz, M. C., Schwartz, D. C., Sergushichev, A.,
407 Sergushichev, A., Sharpe, T., Sharpe, T., Shaw, T. I., Shaw, T. I., Shendure, J., Shi,
408 Y., Simpson, J. T., Song, H., Song, H., Tsarev, F., Tsarev, F., Vezzi, F., Vicedomini,
409 R., Wang, J., Vieira, B. M., Worley, K. C., Wang, J., Worley, K. C., Yin, S., Yiu,
410 S.-M., Yin, S., Yuan, J., Yiu, S.-M., Yuan, J., Zhang, G., Zhang, H., Zhou, S., and
411 Korf, I. F. (2013). Assemblathon 2: evaluating *de novo* methods of genome assembly
412 in three vertebrate species. *GigaScience*, 2(1):10.

413 Bray, N., Pimentel, H., Melsted, P., and Pachter, L. (2015). Near-optimal RNA-Seq
414 quantification. *arXiv.org*.

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

Brown, C. T., Howe, A., Zhang, Q., Pyrkosz, A. B., and Brom, T. H. (2012). A Reference-Free Algorithm for Computational Normalization of Shotgun Sequencing Data. *arXiv.org*.

Cox, M. P., Peterson, D. A., and Biggs, P. J. (2010). SolexaQA: At-a-glance quality assessment of Illumina second-generation sequencing data. *BMC Bioinformatics*, 11(1):485.

Fitzpatrick, M., Ben-Shahar, Y., Smid, H., Vet, L., Vet, L., Robinson, G. E., Sokolowski, M., and Sokolowski, M. (2005). Candidate genes for behavioural ecology. *Trends In Ecology & Evolution*, 20(2):96–104.

Garber, M., Grabherr, M. G., Guttman, M., and Trapnell, C. (2011). Computational methods for transcriptome annotation and quantification using RNA-seq. *Nature Methods*, 8(6):469–477.

GLENN, T. C. (2011). Field guide to next-generation DNA sequencers. *Molecular Ecology Resources*, 11(5):759–769.

Guo, Q., Hao, Y.-J., Li, Y., Zhang, Y.-J., Ren, S., Si, F.-L., and Chen, B. (2015). Gene cloning, characterization and expression and enzymatic activities related to trehalose metabolism during diapause of the onion maggot Delia antiqua (Diptera: Anthomyiidae). *Gene*, 565(1):106–115.

Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., Couger, M. B., Eccles, D., Li, B., Lieber, M., MacManes, M. D., Ott, M., Orvis, J., Pochet, N., Strozzi, F., Weeks, N., Westerman, R., William, T., Dewey, C. N., Henschel, R., Leduc, R. D., Friedman, N., and 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.

Heo, Y., Wu, X.-L., Chen, D., Ma, J., and Hwu, W.-M. (2014). BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads. *Bioinformatics*, 30(10):1354–1362.

Hiz, M. C., Canher, B., Niron, H., and Turet, M. (2014). Transcriptome analysis of salt tolerant common bean (Phaseolus vulgaris L.) under saline conditions. *PloS one*, 9(3):e92598.

456 Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). *De novo* assembly
457 and genotyping of variants using colored *de Bruijn* graphs. *Nature Publishing Group*,
458 44(2):226–232.

459 Jiang, H., Lei, R., Ding, S.-W., and Zhu, S. (2014). Skewer: a fast and accurate
460 adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics*,
461 15(1):182.

462 Johnson, J. M., Castle, J., Garrett-Engele, P., Kan, Z., Loerch, P. M., Armour, C. D.,
463 Santos, R., Schadt, E. E., and Stoughton, Roland Shoemaker, D. D. (2003). Genome-
464 wide survey of human alternative pre-mRNA splicing with exon junction microarrays.
465 *Science*, 302(5653):2141–2144.

466 Le, H. S., Schulz, M. H., McCauley, B. M., Hinman, V. F., and Bar-Joseph, Z. (2013).
467 Probabilistic error correction for RNA sequencing. *Nucleic Acids Research*, 41(10):1–11.

468 Li, B., Fillmore, N., Bai, Y., Collins, M., Thomson, J. A., Stewart, R., and Dewey, C.
469 (2014). Evaluation of *de novo* transcriptome assemblies from RNA-Seq data. Technical
470 report.

471 Li, H. (2015). Correcting Illumina sequencing errors for human data. *arXiv.org*.

472 Liang, C., Liu, X., Yiu, S.-M., and Lim, B. L. (2013). *De novo* assembly and character-
473 ization of Camelina sativa transcriptome by paired-end sequencing. *BMC Genomics*,
474 14(1):146.

475 Love, M. I., Huber, W., and anders, S. (2014). Moderated estimation of fold change and
476 dispersion for RNA-Seq data with DESeq2. *biorxiv.org*.

477 Macfarlan, T. S., Gifford, W. D., Driscoll, S., Lettieri, K., Rowe, H. M., Bonanomi, D.,
478 Firth, A., Singer, O., Trono, D., and Pfaff, S. L. (2012). Embryonic stem cell potency
479 fluctuates with endogenous retrovirus activity. *Nature*, 487(7405):57–63.

480 MacManes, M. D. (2014). On the optimal trimming of high-throughput mRNA sequence
481 data. *Frontiers in Genetics*, 5.

482 MacManes, M. D. and Eisen, M. B. (2013). Improving transcriptome assembly through
483 error correction of high-throughput sequence reads. *PeerJ*, 1:e113.

484 Macmanes, M. D. and Lacey, E. A. (2012). The Social Brain: Transcriptome Assembly
485 and Characterization of the Hippocampus from a Social Subterranean Rodent, the
486 Colonial Tuco-Tuco (*Ctenomys sociabilis*). *PloS one*, 7(9):e45524.

487 Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing
488 reads. *EMBnet.journal*, 17(1):pp. 10–12.

489 McDonald, E. and Brown, C. T. (2013). khmer: Working with Big Data in Bioinformatics.
490 *arXiv.org*.

491 Modrek, B., Resch, A., Grasso, C., and Lee, C. (2001). Genome-wide detection of al-
492 ternative splicing in expressed sequences of human genes. *Nucleic Acids Research*,
493 29(13):2850–2859.

494 Panhuis, T. M. (2006). Molecular evolution and population genetic analysis of candidate
495 female reproductive genes in *Drosophila*. *Genetics*, 173(4):2039–2047.

496 Patro, R., Duggal, G., and Kingsford, C. (2015). Accurate, fast, and model-aware tran-
497 script expression quantification with Salmon. *biorxiv.org*, pages 1–35.

498 Pell, J., Hintze, A., Canino-Koning, R., Howe, A., Tiedje, J. M., and Brown, C. T.
499 (2012). Scaling metagenome sequence assembly with probabilistic *de Bruijn* graphs.
500 *Proceedings of the National Academy of Sciences*, 109(33):13272–13277.

501 Robinson, M. D., McCarthy, D. J., McCarthy, D. J., and Smyth, G. K. (2010). edgeR:
502 a Bioconductor package for differential expression analysis of digital gene expression
503 data. *Bioinformatics*, 26(1):139–140.

504 Shinzato, C., Inoue, M., and Kusakabe, M. (2014). A snapshot of a coral "holobiont":
505 a transcriptome assembly of the scleractinian coral, porites, captures a wide variety of
506 genes from both the host and symbiotic zooxanthellae. *PloS one*, 9(1):e85182.

507 Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V., and Zdobnov, E. M.
508 (2015). BUSCO: assessing genome assembly and annotation completeness with single-
509 copy orthologs. *Bioinformatics*, 31(19):3210–3212.

510 Simpson, J. T. and Durbin, R. (2012). Efficient *de novo* assembly of large genomes using
511 compressed data structures. *Genome Research*, 22(3):549–556.

512 Smith-Unna, R. D., Boursnell, C., Patro, R., Hibberd, J. M., and Kelly, S. (2015). Tran-
513 sRate: reference free quality assessment of de-novo transcriptome assemblies. Technical
514 report.

515 Song, L. and Florea, L. (2015). Rcorrector: efficient and accurate error correction for
516 Illumina RNA-seq reads. *GigaScience*, 4(1):48.

517 Song, L., Florea, L., and Langmead, B. (2014). Lighter: fast and memory-efficient se-
518 quencing error correction without counting. *Genome Biology*, 15(11):509.

519 Studholme, D. J. (2010). *De novo* assembly of short sequence reads. *Briefings In Bioin-*
520 *formatics*, 11(5):457–472.

521 Vijay, N., Poelstra, J. W., Künstner, A., Wolf, J. B. W., and Wolf, J. B. W. (2013).
522 Challenges and strategies in transcriptome assembly and differential gene expression
523 quantification. A comprehensive *in silico* assessment of RNA-seq experiments. *Molec-*
524 *ular Ecology*, 22(3):620–634.

525 Wolf, J. B. W. (2013). Principles of transcriptome analysis and gene expression quantifi-
526 cation: an RNA-seq tutorial. *Molecular Ecology Resources*, 13(4):559–572.

527 Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., Huang, W., He, G., Gu, S.,
528 Li, S., Zhou, X., Li, Y., Xu, X., Wong, G. K.-S., and Wang, J. (2014). SOAPdenovo-
529 Trans: *de novo* transcriptome assembly with short RNA-Seq reads. *Bioinformatics*,
530 30(12):1660–1666.

531 Zhang, R., Zhang, L.-l., Ye, X., Tian, Y.-y., Sun, C.-f., Lu, M.-x., and Bai, J.-j. (2013).
532 Transcriptome profiling and digital gene expression analysis of Nile tilapia (Oreochromis
533 niloticus) infected by Streptococcus agalactiae. *Molecular biology reports*, 40(10):5657–
534 5668.