Chapter 15

De Novo Transcriptome Assembly in Polyploid Species

Juan J. Gutierrez-Gonzalez and David F. Garvin

Abstract

In the absence of a reference genome, the ultimate goal of a de novo transcriptome assembly is to accurately and comprehensively reconstruct the set of messenger RNA transcripts represented in the sample. Non-reference assembly of the transcriptome of polyploid species poses a particular challenge because of the presence of homeologs that are difficult to disentangle at the sequence level. This is especially true for hexaploid oats, which have three highly similar subgenomes, two of which are thought to be nearly identical. Under these circumstances, most software packages and established pipelines encounter difficulties in rendering an accurate transcriptome because they are typically developed, refined, and tested for diploid organisms. We present a protocol for transcriptome assembly in oats that can be extended both to other polyploids and species with highly duplicated genomes.

Key words Oats, Polyploid, De novo transcriptome assembly, RNAseq, Transcript assembly, Homeolog, Homeoallele

1 Introduction

Understanding the assemblage of genes that are expressed in a particular tissue at a certain time—a transcriptome—is of utmost importance for plant researchers because transcriptomes not only provide information on the genes that are active at a certain stage but also on their expression levels. In the event that a reference genome is not available, a de novo transcriptome assembly has to be constructed. Because of the absence of a reference genome sequence that can guide the process, de novo assemblies are more challenging to achieve in an accurate manner. Polyploids pose an extra challenge for de novo transcriptome assembly because of the presence of subgenomes that are frequently very similar. In hexaploid oats this is particularly relevant due to the high homology among the three constituent subgenomes [1, 2]. In a previous study, sequences for homeologous genes coding for the main steps of the vitamin E biosynthesis pathway were identified [1]. The results drew attention to the high similarity among homeologs, with those deriving from two

Sebastian Gasparis (ed.), *Oat: Methods and Protocols*, Methods in Molecular Biology, vol. 1536,
DOI 10.1007/978-1-4939-6682-0_15, © Springer Science+Business Media LLC 2017

of the three oat subgenomes being nearly identical. Indeed, the nucleotide sequence identity between the two more similar homeologous transcripts for the different steps of the vitamin E pathway ranged from 98.6 to 99.8%, and from 98.0 to 100% at the amino acid level. Moreover, the identities between the two more similar homeologs compared to the more divergent one ranged from 95.5 to 98.2% and from 95.1 to 99.6%, at the nucleotide sequence and amino acid levels, respectively.

This high degree of similarity makes homeologous oat transcripts difficult both to differentiate from each other, and to identify sequencing errors. Two main sources of errors can affect a de novo transcriptome assembly in oats and in other polyploids with highly similar subgenomes. The first is homeolog collapse, which occurs when multiple homeologs are assembled into a single hybrid transcript. The second is SNP shuffling, and occurs when homeolog-specific SNPs are assembled into a homeolog to which it does not belong. This commonly results in a transcript number lower (homeolog collapse) or higher (SNP shuffling) than the three expected transcript isoforms assembled for the same gene, at least for single copy genes that derive from each subgenome. Thus, to increase the chances for a transcriptome with polyploid characteristics to be accurately assembled, it is essential to have (1) high quality starting materials, including high-quality nonfragmented RNA and, subsequently, long high-quality sequencing reads; (2) adequate read coverage; and (3) appropriate software packages and an analysis pipeline that can discriminate homeolog nucleotide differences and sequencing errors. Here, we have developed and tested a pipeline for the de novo assembly of oat transcriptomes. The protocol uses open source software and can be extended to assemble transcriptomes of other polyploids.

2 Materials

The protocol starts with high-quality RNA extracted from the desired tissue, using standard RNA plant extraction protocols (*see* **Note 1**). An amount between 500 and 1000 ng of total RNA is recommended. For a simple transcriptome assembly, where differential gene expression analysis is not part of the study, replicates are not required. Care must be taken when selecting the plant tissue from which to extract RNA, depending on the planned use of the transcriptome (*see* **Note 2**). For instance, one may want to select several different tissues and/or developmental stages to increase the likelihood of capturing a wide range of expressed genes. Once RNA from the desired tissue(s) is extracted, it is usually sent to a genomics facility for further processing. At this facility, sample cDNA libraries will be constructed and subsequently run on a high throughput sequencing platform. For this protocol we assume that library preparation and sequencing are going to take place in such a sequencing facility.

2.1 RNA Extraction

1. TRIzol reagent (Invitrogen, Carlsbad, CA).
2. RNeasy plant columns (Qiagen, Valencia, CA).
3. RNase-free DNase I (New England BioLabs, Ipswich, MA).

2.2 Library Preparation

Library preparation and sequencing are usually carried out in the core sequencing facility and thus, except for some guiding tips, it will not be described in detail here. For a successful assembly with this protocol, the insert size has to be between 500 and 650 bp long, so that a significant proportion of the paired 300 bp long reads have an overlapping fragment (*see* Fig. 1). To size select, first shear sample single stranded RNA, or its double stranded complementary DNA, with a sonicator (Covaris S220 or similar). If RNA was used, synthesize double stranded cDNA from sheared RNA fragments and carry the samples through the remainder of the Illumina® TruSeq® library preparation procedure: adapter ligation and amplification. Use a Caliper® LabChipXT® fractionation system or similar for interval size selection (*see* **Note 3**).

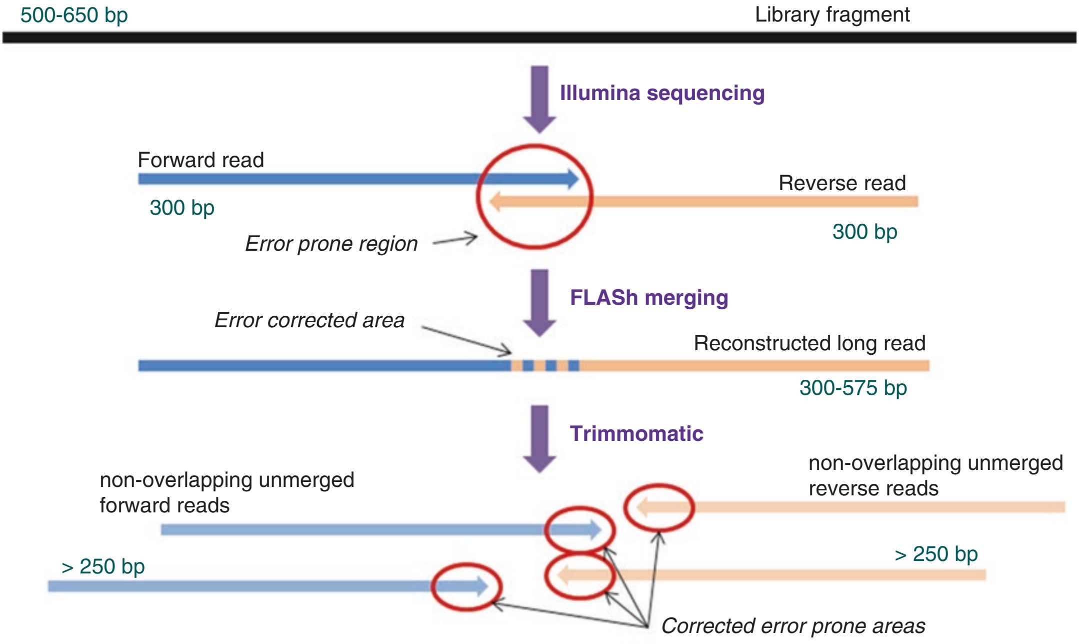

Fig. 1 *Schematic representation of read processing steps*. First, library fragments are sequenced on the Illumina MiSeq instrument to render 300 bp long paired-end reads. Forward reads are in *blue*; reverse in *orange*. The quality within Illumina reads typically decreases toward the end (*red circles*), making these regions error-prone. Second, reads whose 3′-ends overlap are merged into a single longer read by FLASh, which can also correct base errors in the overlapping error-prone region. Third, the remaining unmerged paired-end reads are cleaned of errors and other contaminants with the help of Trimmomatic

2.3 Sequencing

For this protocol we will use Illumina sequencing by synthesis (SBS) chemistry on an Illumina MiSeq® system, which integrates cluster generation, amplification, sequencing, and data analysis into a single instrument (*see* **Note 4**). Libraries have to be sequenced with a 300 bp paired-end run, using V3 chemistry to allow the highest output (up to 15 Gb). Sheared cDNA input can be as little as 10–100 ng.

We believe that for oat transcriptomes, Illumina MiSeq® long paired reads are currently able to yield the best assemblies (*see* **Note 5**). An added value of Illumina reads is that they can also be used later in downstream gene expression quantification analysis. However, sequencing is a rapid-evolving area of research and other promising technologies are on the rise that could also be used in a short future (*see* **Note 6**). This protocol assumes that sequencing is performed in an external genomics facility.

2.4 Computer Resources

This protocol was designed and tested with a 64bit Linux-compatible environment in mind (*see* **Note 7**). Some familiarity with the Unix/Linux command-line user interface would be useful but is not required. A significant amount of memory (200–300 Gb) dedicated over a period extending several days (3–7) and considerable hard drive storage capacity (up to 50–100 Gb) may be required. Nevertheless, the resources considered necessary vary and depend on factors such the number of reads obtained and the diversity of transcripts. This protocol assumes that there is a high performance computing system in your research center with high speed networks, high performance storage, multicore processors, and large amounts of memory to support compute and memory intensive programs.

2.5 Software Packages

All software packages used in this protocol can be obtained free of charge for academic use. This protocol uses the following software packages: FastQC for read quality control; FLASh for paired-end read merging; Trimmomatic for read trimming; and Velvet and Oases for read assembly. They will be described more in detail in the next sections.

2.5.1 Quality Control (QC)

1. FastQC (www.bioinformatics.babraham.ac.uk/projects/fastqc/). The quality of input reads is a major factor determining the accuracy of a de novo transcriptome assembly (*see* **Note 8**). Low quality reads are prone to errors and thus are an obstacle for a precise assembly, especially in polyploid genomes because assemblers might not correctly discriminate homeoallelic SNPs from sequencing errors [3]. Thus, read quality checking should be performed once reads are delivered from the sequencing facility. Overall, Illumina technology produces low-error reads; however, the quality of base calling decreases toward the ends, especially with longer reads. For a 300 bp read, the length used

in this protocol, quality typically decreases greatly in the last 25–50 bases. In diploid organisms, this issue is ameliorated by increased read depth, and thus read errors are corrected to a great extent by assemblers. In polyploid species with very similar subgenomes, base calling accuracy is of paramount importance because assemblers have difficulties in discriminating between subgenome-specific SNPs and read miscalls [3].

2.5.2 Read Elongation

1. FLASh (https://sourceforge.net/projects/flashpage/files/).
   FLASh is a very fast and accurate software tool to merge paired-end reads from next-generation sequencing experiments [4]. FLASh is designed to merge pairs of reads when the original DNA fragments are shorter than twice the length of reads. If a mismatch is present in the overlapping fragment, it can be corrected based on the quality of the base(s) in question. The resulting merged longer reads can significantly improve transcriptome assemblies.

2.5.3 Read Trimming

1. Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic).
   Trimmomatic is a fast, multithreaded command line quality control tool that can be used to trim and crop Illumina (fastq) poor quality data, as well as to remove adapters and other Illumina-specific sequences [5]. These adapters can pose a significant problem to accurate assemblies as they can be a source for foreign base introgressions.

2.5.4 Assembly of Reads

1. Velvet (http://www.ebi.ac.uk/~zerbino/velvet/).
2. Oases (http://www.ebi.ac.uk/~zerbino/oases/).
   Velvet is a de novo genomic assembler specially designed for short read sequencing technologies [6]. Oases [7] was released as an extension of Velvet to address the particular issues affecting transcriptome assemblies, including (1) transcript sequencing depth potentially varying several orders of magnitude due to differences in gene expression, and (2) the presence of transcript isoforms due to alternative splicing. The Velvet/Oases package is capable of using a combination of long, short, single, and paired-end reads to render high-quality transcriptome assemblies (*see* **Note 9**). For this protocol, Velvet/Oases has to be compiled with specific non-default instructions (*see* **Note 10**).

3 Methods

3.1 RNA Extraction and Library Preparation

1. Grind selected plant tissue using mortar and pestle in the presence of liquid nitrogen.
2. Extract RNA following well-established protocols. We recommend using the TRIzol method, followed by a purification

step with RNeasy plant columns, and digestion with RNase-free DNase I to eliminate traces of DNA. More details can be found in [3].

3. Send high-quality RNA to a core facility for library construction and sequencing. For this protocol we assume that the sequencing facility delivers the resultant sequencing data in two files containing paired end reads: `MiSeq_R1_pair.fastq` and `MiSeq_R2_pair.fastq`, for forward and reverse pairs, respectively.

3.2 Read QC

1. Create a working directory (`MiSeq_reads`) where reads will be copied into.

```
% mkdir MiSeq_reads
% cd MiSeq_reads
```

2. Download and install FastQC (www.bioinformatics.babraham.ac.uk/projects/fastqc/).

3. Run it on the file reads.

```
% /path-to-fastqc-directory/fastqc MiSeq_R1_
pair.fastq MiSeq_R2_pair.fastq
```

The symbol % represents the command line prompt. Substitute `/path-to-fastqc-directory/` with the absolute path to where `fastqc` has been installed. The QC step will provide some basic statistics and other important information. For instance, this will permit an assessment of the presence of sequence contaminants, such as Illumina adapters. Reads should come free of such contaminants from the sequencing facility. Without doubt, apart from the presence of contaminants, the most important piece of information to obtain after running `fastqc` is the per base sequence quality, which will give us a graphic representation of the average base quality along the read and the point at which quality drops below a value that will compromise accurate assembly. This information will be used to modify parameters in downstream software.

3.3 Trimmomatic (Optional)

Should contaminants be detected in the reads, they will have to be removed before the next step, using specialized software (*see* Subheading 3.5). Reads should also go through this optional step if their average quality is low. Because read quality is a key component for an accurate assembly, it is not advisable to proceed with the protocol if read quality is too low. In this case, rerunning the sequencing step is generally recommended.

3.4 Merging Paired Reads with FLASh

1. Download and install FLASh (https://sourceforge.net/projects/flashpage/files/).

2. Run the following command:

```
% /path-to-flash-directory/flash \
MiSeq_R1_pair.fastq MiSeq_R2_pair.fastq \
-m 25 -M 250 -x 0.1 -o flash.out 2>&1 | tee flash.
   log
```

The first two files are the forward and reverse read input files. The backslash "\" character means that we are continuing the command on the next line, and it is used in scripting languages to improve readability. "m" (min-overlap) is the minimum required overlap length between two reads. Adjust this parameter according to the average insert size and read quality toward the end. Lower values increase the number of extended reads but at the risk of also increasing incorrectly merged pairs. We do not recommend m to be set below 15–20. "M" (max-overlap) is the maximum overlap length expected in 90% read pairs. "x" (max-mismatch-density) is the maximum allowed ratio between the number of mismatched base pairs and the overlap length. Adjust this parameter according to the average read quality toward the ends of the reads. Increasing x increases the number of correctly merged read pairs, but at the expense of increasing the number of incorrectly merged pairs. The string after "o" (flash.out) indicates the prefix for the output files. The "2>&1" operator redirects standard error to the standard output stream (screen). Here, it is piped with "|" to divert a copy of the standard output stream to the intermediate file flash.log.

FLASh will produce three main output files with reads, along with several statistic and error files: flash.out.extendedFrags.fastq: file containing the merged reads; flash.out.notCombined_1.fastq: file with forward reads of the pairs that were not merged; and flash.out.notCombined_2.fastq: file with reverse reads of the pairs that were not merged.

3.5 Trimmomatic

1. Download and install Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic).
2. Run the following command on the non-merged Illumina paired reads:

```
% java -jar $TRIMMOMATIC/trimmomatic.jar PE
   -threads 4 -phred33 \
-trimlog trimmomatic.log \
flash.out.notCombined_1.fastq  flash.out.not-
   Combined_2.fastq \
flash.out.notCombined_1.PE.fastq \
flash.out.notCombined_1.SE.fastq \
flash.out.notCombined_2.PE.fastq \
flash.out.notCombined_2.SE.fastq \
ILLUMINACLIP:$TRIMMOMATIC/adapters/TruSeq2-
   PE.fa:2:30:10:2:true \
LEADING:10  TRAILING:10  SLIDINGWINDOW:4:15
   MINLEN:250
```

This will do the following:

- Remove adapters and other Illumina-specific sequences (`ILLUMINACLIP:$TRIMMOMATIC/adapters/TruSeq2-PE.fa:2:30:10:2:true`). This command will clip reads according to the adapters and other sequences in `TruSeq2-PE.fa`, maximum mismatch count (`2`), palindrome clip threshold (`30`), simple clip threshold (`10`), minimum adapter length (`2`) to allow shorter fragments to be removed, and keep both reads (`true`) in the cases in which forward and reverse contain the same sequence information, albeit in reverse complement. Most likely you will not have to modify this part.
- Remove leading low quality or N bases (below quality 10) (`LEADING:10`).
- Remove trailing low quality or N bases (below quality 10) (`TRAILING:10`).
- Scan the read with a 4-base wide sliding window, cutting when the average quality per base drops below 15 (`SLIDINGWINDOW:4:15`).
- Eliminate reads below the 250 bases long (`MINLEN:250`).

You may want to modify the numbers in `LEADING`, `TRAILING`, and `SLIDINGWINDOW` depending on the quality of your reads. Additionally, you may want to adjust `MINLEN` by using a higher or lower threshold if the quality of reads is high or low, respectively.

Two input files with the forward (`flash.out.notCombined_1.fastq`) and reverse (`flash.out.notCombined_2.fastq`) reads to clean are needed. They contain the reads not merged by FLASh in the previous step. Trimmomatic will produce five output files: `trimmomatic.log` is the log file, which includes information such as the read name, the surviving sequence length, the number of bases trimmed from the start, and the number of bases trimmed from the end; `flash.out.notCombined_1.PE.fastq` and `flash.out.notCombined_2.PE.fastq` are the forward and reverse paired read files that survived the trimming process as pairs. Finally, `flash.out.notCombined_1.SE.fastq` and `flash.out.notCombined_2.SE.fastq` are the unpaired read files whose matching pair did not pass the trimming process, and thus are single read files now.

3.6 Read Assembly

1. Download and install Velvet (https://www.ebi.ac.uk/~zerbino/velvet/) and Oases (http://www.ebi.ac.uk/~zerbino/oases/). The Velvet/Oases package is comprised of three independent modules that have to be run in a sequential order: *velveth*, *velvetg*, and *oases*. Because of the large size of the intermediate files created in the process, we strongly

recommend testing Velvet/Oases in a scratch directory and copy the informative files (typically just `transcripts.fa`) to a more permanent directory when you are satisfied with the outcome. For large datasets some recommendations can be followed to speed up the process (*see* **Note 11**).

2. Run *velveth* with the following command:

```
cd scratch_directory
% /path-to-velvet-directory/velveth Miseq_
   assembly 149 \
-long -fastq /MiSeq_reads/flash.out.extended-
   Frags.fastq \
-shortPaired -separate -fastq /MiSeq_reads/
   flash.out.notCombined_1.PE.fastq \ /MiSeq_
   reads/flash.out.notCombined_2.PE.fastq \
-short -fastq /MiSeq_reads/flash.out.notCom-
   bined_1.SE.fastq \
/MiSeq_reads/flash.out.notCombined_2.SE.fastq
```

Where `Miseq_assembly` is the name of the folder we want Velvet/Oases to store the results. Probably, the most important parameter to specify is the *k-mer* value (`149` in this case). We recommend higher (125–149) than typical values suggested for diploid species (*see* **Note 9**). We also recommend testing several *k-mers*. The next entries are self explanatory. They describe the type of reads (`-long`, `-short`, `-shortPaired`) and the files that contain them (for more detailed parameter description check Velvet manual, http://www.ebi.ac.uk/~zerbino/velvet/Manual.pdf).

3. Run *velvetg* with the following command:

```
% /path-to-velvet-directory/velvetg Miseq_
   assembly \
-read_trkg yes -min_pair_count 2 -cov_cutoff
   auto -ins_length 650 \
-exp_cov 5 -max_gap_count 1 -min_contig_lgth
   300 -conserveLong yes
```

Change the `ins_length` according to your paired-end library average insert size (*see* **Note 3**). Also, modify the expected *k-mer* coverage (`exp_cov`) accordingly. The rest of the parameters are probably best left untouched (for more detailed parameter description check Velvet manual).

4. Run *oases* with the following command:

```
% /path-to-oases-directory/oases Miseq_assem-
   bly \
-min_trans_lgth 300
```

where `min_trans_lgth` is the desired minimum transcript length. After a successful run (*see* **Note 12**) several output files are created, of which `transcripts.fa` is the most useful for use as it contains the assembled homeologs. `transcripts.fa` is a `fasta` file with transcript isoforms organized in *Loci*, where individual transcript isoforms within a *Locus* might represent homeologs but also splice variants. The `fasta` header for each transcript appears as:

```
>Locus_x_Transcript_y/Y_Confidence_z_Length_L
```

where "`x`" is the Locus unique identification number; "`y`" is the transcript isoform number among the "`Y`" isoforms in this `Locus`; "`z`" is a number between $0 \leq z \leq 1$ (ideally we want `z` to be as close as 1 as possible); and "`L`" is the length of that particular transcript isoform. For oats, in the ideal situation where three putative homeologs are assembled, we should see something like this:

```
>Locus_x_Transcript_1/3_Confidence_z_Length_L
>Locus_x_Transcript_2/3_Confidence_z_Length_L
>Locus_x_Transcript_3/3_Confidence_z_Length_L
```

Of course, the values of `z` and `L` could be different among the three isoforms (homeologs).

4 Notes

1. RNA will be extracted using standard plant RNA extraction protocols. RNA purity and integrity is crucial for a successful read synthesis. We recommend using the TRIzol® method (Invitrogen, Carlsbad, CA), followed by purification with RNeasy plant columns (Qiagen, Valencia, CA). Digestion with RNase-free DNase I (New England BioLabs, Ipswich, MA) to eliminate DNA traces is highly recommended. Prior to library construction, determine RNA quality and integrity with RNA6000 Nano Assay on the Agilent 2100 Bioanalyzer™ (Agilent Technologies Inc, Santa Clara, CA). A RIN value above 7 is desirable.
2. In order to increase transcript diversity, samples from several biological replicates and/or developmental stages may be collected and pooled. If the ultimate purpose for the transcriptome is to be used as a foundation for expression analysis, biological replicates are a must. Long paired-end reads are also highly recommended, and used in this protocol, because they are more likely to map to a single homeolog. If the final purpose of the transcriptome is expression studies, very strict or even zero-tolerance parameters are desired when aligning reads to the assembled transcriptome for read counting.
3. The more stable cDNA is preferred over unstable, single stranded RNA as starting material for shearing. The desired target fragment size of 500–650 bp reflects the length after subtracting off the Illumina adapter/barcode sequence of 120 bp. Thus, the fragment to select in the fractionation system must be in the range of 620–770. For a correct shearing and library representation, the starting RNA or cDNA material should be at least 500 ng; however, because of the subsequent

size selection step an amount of 10 μg or higher is recommended (also *see* **Note 8**).

4. Illumina sequencing by synthesis (SBS) chemistry is the most widely adopted next-generation sequencing technology because of the low sequencing errors and high-throughput. Illumina MiSeq® system using V3 chemistry is capable of generating 20–25 M reads on average and more than 12 Gb (up to 15 Gb) of data for each 2 × 300 base pair run. The MiSeq also has the ability to multiplex several samples per run. Currently, Illumina sequencing delivers the highest yield of error-free data. The typical error rate for an entire Illumina Miseq 300 bp long read may vary between 0.5 and 2 % and depends on library type, run quality, and a host of other factors.

5. In our experience, combining different technologies, such as Roche-454 and Illumina, to perform what is known as a *hybrid assembly* is not recommended for polyploid species, where the similarity among subgenomes can confound assemblers. This is because different technologies have distinct error patterns. Assembler algorithms are typically designed with a particular type of sequencing technology in mind and to deal with specific error patterns.

6. Single molecule real-time (SMRT®) sequencing (PACBIO®, Pacific Biosciences of California, Inc) is a promising technique that can develop reads as long as 10 kb on average. However, its single pass accuracy of about 85 % is probably still too high to discriminate oat homeolog single nucleotide variants from sequencing errors. Should SMRT sequencing improve accuracy in the next years, it will represent an interesting alternative to Illumina technology for transcriptome assemblies and it will be worth considering.

7. Although the protocol might work on other computer systems, particularly on a 32-bit machine, no other system has been tested by us. Running a 32-bit machine might find memory to be a limiting factor.

8. Two key factors affecting the quality of an assembly are the quality of sequence reads and the level of read duplication. Read errors can confound assemblers and are especially problematic in polyploid organisms. Read duplication is highly dependent on the quality of the input cDNA and library preparation. To minimize these errors and increase efficiency of sequencing and rare gene discovery, we recommend (1) conducting library preparation from enough RNA/cDNA starting material and perform some type of library normalization so there is no over-representation of certain abundant transcripts; and (2) avoid a large variance in the fragment size selected such that smaller fragments become over-represented. Normalization

methodologies are desirable in situations where the actual RNA counts are unimportant (de novo transcriptome), but not appropriate when transcript quantification is an objective (RNAseq).

9. There are two major classes of assembling algorithms: the *de Bruijn graphs* and the overlap-layout-consensus [8]. Velvet/Oases is an example of the first type. For assemblers that use *de Bruijn graphs* methods, the quality of a de novo assembly is highly dependent on the selected *k-mer* value. The optimum value depends on the sequencing depth, the read error rate, and the complexity of the genome/transcriptome to be assembled [9]. For transcriptomes, where coverage is not uniform, the dilemma is choosing between a higher *k-mer* length that will result in a more contiguous assembly of highly expressed transcripts; or a lower *k-mer*, which theoretically will favor weakly expressed transcripts but will lead to the assembly of numerous and highly fragmented transcripts [7]. For polyploids, we do not recommend performing a *merged assembly*, that is, running velvet with different values of *k* and merge the results of all assemblies into a single nonredundant assembly. According to our benchmarks, merging tends to produce highly fragmented and mixed assemblies. Neither do we recommend the use of overlap-layout-consensus assemblers for transcriptomes in polyploid species, as they have the tendency to mix fragments from different homeologs.

10. By default, Velvet is compiled with `MAXKMERLENTH` of 31 bp. Thus, hash-lengths (*k-mers*) are limited to 31 bp. For the Velvet/Oases package to be able to run with the parameter values specified in this protocol, it has to be compiled with the following instructions, so that both long *k-mers* and long sequences can be used:

```
make 'CATEGORIES=7' 'MAXKMERLENGTH=149' 'LONGSEQUENCES=1'
    'BIGASSEMBLY=1'
```

11. To speed up the assembly process a couple of approaches may be taken. First, you may use multithreading, which allows a program to make use of multiple CPU cores on the same machine. To turn on multithreading in Velvet use the option make `'OPENMP=1'` at compilation. OpenMP allows a program to make use of multiple CPU cores on the same machine. You might have to set the environment variables `'OMP NUM THREADS'` and `'OMP THREAD LIMIT'`. See the Velvet documentation for more information. Second, you can significantly speed up Velvet by working with a binary version of sequences. For this, simply add the `-create_binary` parameter to the *velveth* command:

```
% /path-to-velvet-directory/velveth Miseq_
    assembly 149 -create_binary (…)
```

12. There is not a standard reliable method to validate the results of a de novo transcriptome assembly. Nevertheless, at least three approaches can be taken to assess the quality of a transcriptome without the need for a reference genome. First, software packages such as TransRate [10] can be used for reference-free quality assessment of de novo transcriptome assemblies. Second, quality can be assessed by aligning the de novo transcriptome to a well-curated set of reference ESTs present in public databases (GenBank, EMBL, or DDBJ). The last but the least precise method is by aligning the transcriptome to a well-conserved set of genes from close relatives.

Acknowledgments

Mention of trade names or commercial products in this article is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the U.S. Department of Agriculture. This work was carried out in part using computing resources at the University of Minnesota Supercomputing Institute. The authors thank the Biomedical Genomics Center, University of Minnesota for the testing of different protocols for library preparation and sequencing reactions.

References

1. Gutierrez-Gonzalez JJ, Garvin DF (2016) Subgenome-specific assembly of vitamin E biosynthesis genes and expression patterns during seed development provide insight into the evolution of oat genome. Plant Biotechnol J. doi:10.1111/pbi.12571
2. Gutierrez-Gonzalez JJ, Garvin DF (2011) Reference genome-directed resolution of homologous and homeologous relationships within and between different oat linkage maps. Plant Genome 4:178–190
3. Gutierrez-Gonzalez JJ, Zheng JT, Garvin DF (2013) Analysis and annotation of the hexaploid oat seed transcriptome. BMC Genomics 14:471
4. Magoc T, Salzberg S (2011) FLASh: fast length adjustment of short reads to improve genome assemblies. Bioinformatics 27(21):2957–2963
5. Bolger AM, Lohse M, Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 30(15):2114–2120
6. Zerbino DR, Birney E (2008) Velvet: algorithms for the novo short read assembly using de Bruijn graphs. Genome Res 18:821–829
7. Schulz MH, Zerbino DR, Vingron M, Birney E (2012) Oases: robust *de novo* RNAseq assembly across the dynamic range of expression levels. Bioinformatics 28:1086–1092
8. Li Z, Chen Y, Mu D, Yuan J, Shi Y, Zhang H, Gan J, Li N, Hu X, Liu B, Yang B, Fan W (2012) Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-Bruijn-graph. Brief Funct Genomics 11(1):25–37
9. Surget-Groba Y, Montoya-Burgos JI (2010) Optimization of de novo transcriptome assembly from next-generation sequencing data. Genome Res 20:1432–1440
10. Smith-Unna R, Boursnell C, Patro R, Hibberd JM, Kelly S (2015) TransRate: reference free quality assessment of *de-novo* transcriptome assemblies. Genome Res 26(8):1134–1144