BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 12 2014, pages 1660–1666
doi:10.1093/bioinformatics/btu077

Sequence analysis

Advance Access publication February 13, 2014

SOAPdenovo-Trans: *de novo* transcriptome assembly with short RNA-Seq reads

Yinlong Xie[1,2,3,†], Gengxiong Wu[2,†], Jingbo Tang[2,4,†], Ruibang Luo[2,3,5,†], Jordan Patterson[6], Shanlin Liu[2], Weihua Huang[2], Guangzhu He[2], Shengchang Gu[2,7], Shengkang Li[2], Xin Zhou[2], Tak-Wah Lam[3], Yingrui Li[5], Xun Xu[2], Gane Ka-Shu Wong[2,6,8,*] and Jun Wang[2,9,10,11,*]

[1]School of Bioscience and Bioengineering, South China University of Technology 510006, Guangzhou, China, [2]BGI-Shenzhen, Shenzhen 518083, China, [3]HKU-BGI Bioinformatics Algorithms and Core Technology Research Laboratory and Department of Computer Science, University of Hong Kong, Pokfulam, Hong Kong, [4]Institute of Biomedical Engineering, XiangYa School of Medicine, Central South University, Changsha 410008, China, [5]BGI-tech, BGI-Shenzhen, Shenzhen 518083, China, [6]Department of Medicine, University of Alberta, Edmonton, AB T6G 2E1, Canada, [7]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, [8]Department of Biological Sciences, University of Alberta, Edmonton, AB T6G 2E9, Canada, [9]The Novo Nordisk Foundation Center for Basic Metabolic Research, University of Copenhagen, Copenhagen DK-2200, Denmark, [10]Department of Biology, University of Copenhagen, Copenhagen DK-2200, Denmark and [11]Princess Al Jawhara Center of Excellence in the Research of Hereditary Disorders, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Associate Editor: Inanc Birol

ABSTRACT

Motivation: Transcriptome sequencing has long been the favored method for quickly and inexpensively obtaining a large number of gene sequences from an organism with no reference genome. Owing to the rapid increase in throughputs and decrease in costs of next- generation sequencing, RNA-Seq in particular has become the method of choice. However, the very short reads (e.g. 2 × 90 bp paired ends) from next generation sequencing makes *de novo* assembly to recover complete or full-length transcript sequences an algorithmic challenge.

Results: Here, we present SOAPdenovo-Trans, a *de novo* transcriptome assembler designed specifically for RNA-Seq. We evaluated its performance on transcriptome datasets from rice and mouse. Using as our benchmarks the known transcripts from these well-annotated genomes (sequenced a decade ago), we assessed how SOAPdenovo-Trans and two other popular transcriptome assemblers handled such practical issues as alternative splicing and variable expression levels. Our conclusion is that SOAPdenovo-Trans provides higher contiguity, lower redundancy and faster execution.

Availability and implementation: Source code and user manual are available at http://sourceforge.net/projects/soapdenovotrans/.

Contact: xieyl@genomics.cn or bgi-soap@googlegroups.com

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 18, 2013; revised on January 15, 2014; accepted on January 31, 2014

1 INTRODUCTION

Transcript sequences and gene expression levels can now be efficiently obtained using RNA-Seq on next-generation sequencing technologies, providing increased throughputs and decreased costs. Applications for RNA-Seq include discriminating expression levels of allelic variants and detecting gene fusions (Maher *et al.*, 2009). To carry out these types of analyses requires an assembler that can reconstruct the transcripts from very short reads (e.g. 2 × 90 bp paired ends). Assemblers such as Cufflinks (Trapnell *et al.*, 2010), Scripture (Guttman *et al.*, 2010) and ERANGE (Mortazavi *et al.*, 2008) recover transcript sequences by aligning the reads to a reference genome. However, reference genomes are not always available, especially if the genome is unusually large and/or polyploid, which is often the case for plants. In these situations, *de novo* assembly is required. The challenge is not only to recover full-length transcripts but also to identify alternative splice forms in the presence of variable gene expression levels.

Historically, the first *de novo* assemblers for next-generation sequencing, like Velvet (Zerbino and Birney, 2008), ABySS (Simpson *et al.*, 2009) and SOAPdenovo (Li *et al.*, 2010), were developed for genomes. These programs were intended to recover sequences for genomes of a known (estimated) size with a defined number of chromosomes. In contrast, transcriptome assemblers must recover an unknown number of RNA sequences, typically on the order of tens of thousands. Further, transcript sequences are only a few (k)ilobases in length, as compared with chromosomes, which can be hundreds of (M)egabases in length. Adding to the complexity is that gene expression levels vary by many orders of magnitude, so that for any non-zero sequencing error rate the most highly expressed genes will always harbor many discrepant bases, making it impossible to

*To whom correspondence should be addressed.

[†]The authors wish it to be known that, in their opinion, the first four authors should be regarded as Joint First Authors.

define an absolute threshold for the number of sequencing errors allowed per assembly. Another issue is that most contemporary *de novo* transcriptome assemblers, like Trans-ABySS (Robertson *et al.*, 2010), Multiple-k (Surget-Groba and Montoya-Burgos, 2010), Rnnotator (Martin *et al.*, 2010), Oases (Schulz *et al.*, 2012) and Trinity (Grabherr *et al.*, 2011), use the *de Bruijn* graph (DBG) schema for computational and memory efficiency, which means that alternative splice forms transcribed from the same locus will be combined into a single complicated *de Bruijn* sub-graph. This then needs to be addressed.

In recent years, some important changes have been introduced to improve transcriptome assembly. Oases enumerated all possible transcripts with the simplifying concept of assembly sub-graphs and then used a robust heuristic algorithm to traverse these graphs. Trinity introduced a new error removal model to account for variations in gene expression levels and then used a dynamic programming procedure to traverse their graphs. However, there is a lot of room for improvement, e.g. Oases produces more redundant transcripts, possibly due to it lacking an effective error-removal model (Lu *et al.*, 2013), and Trinity produces fewer full-length transcripts, possibly due to it not using paired-end data for scaffold construction.

Here we present a *de novo* RNA-Seq assembler, SOAPdenovo-Trans, which builds on these previous innovations to overcome a few remaining issues. SOAPdenovo-Trans incorporates the error-removal model from Trinity and the robust heuristic graph traversal method from Oases. In addition, we use a strict transitive reduction method to simplify the scaffolding graphs, and provide more accurate results. To assess the impact of these changes, we evaluated all three assemblers on rice and mouse, which have established transcriptome data linked to genome annotations produced over the last decade. The results here demonstrated that SOAPdenovo-Trans provides higher contiguity, lower redundancy and faster execution.

2 METHODS

SOAPdenovo-Trans is a DBG-based assembler for transcriptome data, derived from the SOAPdenovo2 (Luo *et al.*, 2012) genome assembler, which has an effective scaffolding module that—with some modifications—is also suitable for transcriptome assembly. However, SOAPdenovo2 was designed for genomes with uniform sequencing depth. Thus, its error-removal model is not applicable to RNA-Seq data. It also does not allow for alternative splicing. Adopting and improving on concepts from Trinity and Oases resolved these issues

The SOAPdenovo-Trans algorithm (schema in Fig. 1A) consists of two main steps: (i) contig assembly and (ii) transcript assembly, described below.

2.1 Contig assembly

DBG construction is done as per SOAPdenovo2, but sequencing errors are removed in two ways: globally (as in the genome version of SOAPdenovo2) and locally (which is an addition specific to SOAPdenovo-Trans). For global error removal, low-frequency *k*-mers, edges, arcs (direct linkage between contigs in the DBG) and tips are removed, and bubbles are pinched. This is done in SOAPdenovo2 under the assumption that most are the result of sequencing errors. However, for the most highly expressed genes in a transcriptome, sequencing errors often generate *k*-mers that exceed any reasonable global error removal threshold. These cannot be corrected by global error removal. In contrast, for the most lowly expressed genes, such low-frequency *k*-mers can legitimately arise in the absence of sequencing errors; hence, in the global error removal, we only applied a weak depth cutoff (by default ≤2) so that these genes are not mistakenly removed from the graph. We then used Trinity's error-removal method to handle the remaining sequencing errors. It defines a percentage threshold for filtration (≤5% of the total or maximal depth of the adjacent graph element, which can refer to *k*-mers, arcs or edges), not a constant threshold, and is better suited for highly expressed genes. Finally, we used the same method as SOAPdenovo2 to generate contigs.

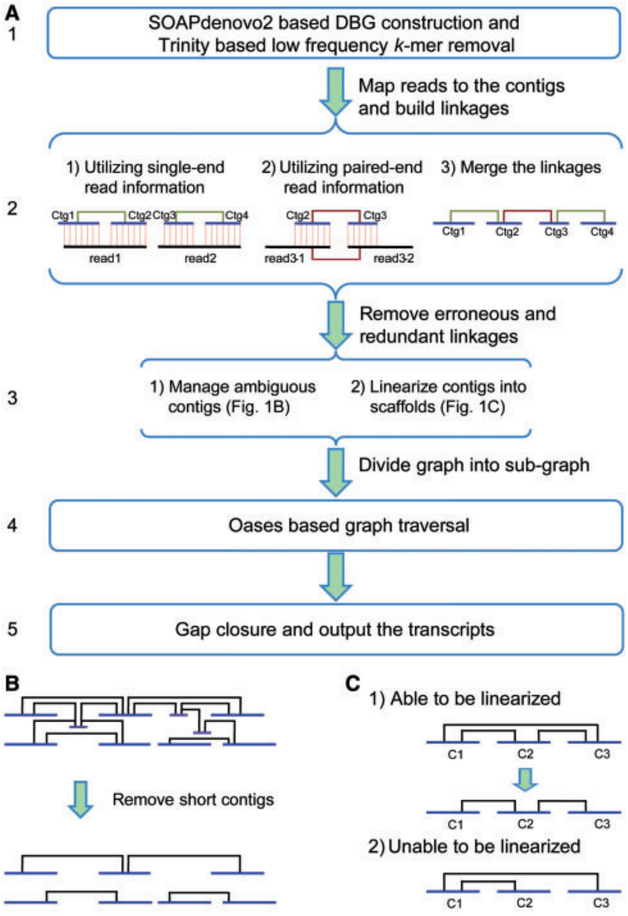
A
1
SOAPdenovo2 based DBG construction and Trinity based low frequency k-mer removal
Map reads to the contigs and build linkages
2
1) Utilizing single-end read information
2) Utilizing paired-end read information
3) Merge the linkages
Ctg1
Ctg2
Ctg3
Ctg4
read1
read2
Ctg2
Ctg3
read3-1
read3-2
Ctg1
Ctg2
Ctg3
Ctg4
Remove erroneous and redundant linkages
3
1) Manage ambiguous contigs (Fig. 1B)
2) Linearize contigs into scaffolds (Fig. 1C)
Divide graph into sub-graph
4
Oases based graph traversal
5
Gap closure and output the transcripts
B
Remove short contigs
C
1) Able to be linearized
C1
C2
C3
C1
C2
C3
2) Unable to be linearized
C1
C2
C3

Fig. 1. Overview of SOAPdenovo-Trans algorithm (**A**1) Contig assembly: DBG are constructed from reads; sequencing errors are removed; and contigs are then constructed. (A2–A5) Transcript assembly: single- and paired-end reads are mapped to the assembled contigs to construct scaffold graphs. Transcripts are generated by traversing through reliable paths for each graph. (**B**) Management of ambiguous contigs. (**C**) Linearizing contigs into scaffolds

2.2 Transcript assembly

2.2.1 Scaffold construction Reads are mapped back onto the contigs to build linkages, as per SOAPdenovo2, except that SOAPdenovo-Trans uses both single-end reads and paired-end reads, while SOAPdenovo2 uses only paired-end reads. This is important because transcripts are much shorter than chromosomes, so it is essential to use the information that may only be found in single-end reads. The number of reads is then used to assign weights to these linkages, and insert sizes from the paired-ends are used to estimate the distances between linkages.

2.2.2 Graph simplification Contigs that are identified as being ambiguous, with multiple successive linkages or of exceptionally high depth (~two times the mean depth), were masked for scaffold building in the genome version of SOAPdenovo2. This, however, is inappropriate for transcriptome assembly because of alternative splicing and variable gene expression levels. Alternative splicing establishes multiple successive linkages from a unique contig. The data representation of this appears analogous to ambiguities in whole genome assembly. Variable gene expression levels make it impossible to define a contig as repetitive using a single depth constant. One of the methods by which SOAPdenovo-Trans copes with these issues is by unconditional removal of short contigs (default ≤100 bp). This removes not only sequencing errors but also short ambiguous contigs caused by repeats, which in turn obviates the need for the scaffolding module to resolve complicated ambiguities. As a result, this increases its ability to identify alternative splicing events (Fig. 1B). Conversely, unconditional removal of short contigs results in the creation of many small gaps, but this is corrected in the final phase of our algorithm by a gap-filling module described in Section 2.2.4.

Linearization of contigs to scaffolds also differs in genome and transcriptome assembly. For genomes, after introducing paired-end reads with multiple tiers of insert sizes, a starting contig may have multiple successive contigs at different distances from the starting contig. The expectation is for these contigs to be linearly integrated into a single scaffold; however, for transcriptomes, conflicts may legitimately arise because multiple alternative splice forms share the same starting contig. To simplify the graphs properly, we used a more stringent linearization method in SOAPdenovo-Trans (Fig. 1C): For example, three contigs, *c1*, *c2* and *c3*, can be linearized if(i) there exists explicit linkage between '*c1* and *c2*', '*c2* and *c3*' and '*c1* and *c3*' and (ii) the distances between *c1*, *c2* and *c3* inferred from linkages do not conflict with each other.

2.2.3 Graph traversal Contigs were clustered into sub-graphs according to their linkage. Each sub-graph consists of a set of transcripts (alternative splice forms) that share common exons. SOAPdenovo-Trans traverses these sub-graphs using the algorithm from Oases to generate possible transcripts from linear, fork and bubble paths. For the most complex paths, only the top scoring transcripts are retained.

2.2.4 Gap filling/correction As noted in Graph Simplification, many small gaps were introduced by masking contigs ≤100 bp before scaffold construction. To compensate for this, we used the DBG- based gap-filling method from SOAPdenovo2. Paired-end information was used to cluster semi-unmapped reads into the gap regions, and then these reads were locally assembled into a consensus. In instances where multiple consensus sequences were assembled, we selected the sequence that had a length most consistent with the gap size.

2.3 Benchmark to genome

For our first benchmark test dataset, we used rice transcriptome data from *Oryza sativa 9311* (panicle at booting stage). Paired-end sequences were generated on an Illumina GA platform (Zhang *et al.*, 2010) with 200 bp insert sizes and 75 bp read lengths. For our analysis, we used a large (L) and small (S) dataset. The L dataset contained 39.9 M reads totaling 5.98 Gbp of sequence, which was obtained from the following:

http://www.ncbi.nlm.nih.gov/sra/SRX017631

http://www.ncbi.nlm.nih.gov/sra/SRX017632

http://www.ncbi.nlm.nih.gov/sra/SRX017633

http://www.ncbi.nlm.nih.gov/sra/SRX017630

The S dataset was down-sampled from the L dataset (using the first file, SRX017631) and contained 9.8 M reads totaling 1.47 bp.

The second benchmark test dataset was mouse transcriptome data from *Mus musculus* (dendritic cells). Paired-end sequences were generated on an Illumina GAII platform (Grabherr *et al.*, 2011) with 300 bp insert sizes and 76 bp read lengths. Here, the L dataset contained 36.1 M reads totaling 5.49 Gbp (after quality filtering, see the filtering steps in the supplement), and was obtained from the following:

http://www.ncbi.nlm.nih.gov/sra/SRX062280

The S dataset was down-sampled from the L dataset (extracting one of every three reads from the L dataset) and contained 12.0 M reads totaling 1.83 Gbp.

As Trinity only supports 25-mers, all assemblers were run with *k*-mer = 25, to make the comparisons 'fair'. SOAPdenovo-Trans (version 0.99) was run with the parameters: '-i 20 -q 5 -Q 2 -H 200 -e 20 -S 48 –r -F -L 100 -c 2 -t 5'. Oases (version 0.2.8) with Velvet (version 1.2.10) was run using minimum-length-of-output-transcripts set to 100. Trinity (version r2013-08-14) was run with minimum-assembled-contig-length-to-report set to 100. The reference genomes and curated annotations were downloaded from the following two Web sites.

Rice: MSU Rice Genome Annotation Project Release 7 at ftp://ftp.plantbiology.msu.edu/pub/data/Eukaryotic_Projects/o_sativa/annotation_dbs/pseudomolecules/version_7.0/

Mouse: Mus_musculus.NCBIM37.64 at ftp://ftp.ensembl.org/pub/release-64/fasta/mus_musculus/

Note that for rice, our transcriptome data came from the *indica* subspecies, but our reference genome came from the *japonica* subspecies. We chose the *japonica* genome as a reference because these annotations are more extensively (manually) curated than their *indica* counterparts. Ideally, we should have used *japonica* transcriptome data, but we used *indica* transcriptome data instead because there is little *japonica* data from the Illumina platform that is freely available. The use of these different subspecies is not totally unreasonable because they differ on average by only a fraction of a percent (Yu *et al.*, 2005). We do, however, note that there are local regions of higher variability that will prevent some *indica* transcripts from aligning to the *japonica* genome.

All of the transcript-to-genome alignments were done in BLAT (Kent, 2002) using a 95% identity cutoff. We required that 95% of the transcript length be accounted for in one consistent alignment before we deemed the transcript to be correctly assembled. When that alignment criterion was not met, we searched for 'chimeric' assemblies that would account for 95% of the transcript length with multiple alignments that occurred in different orientations, on different chromosomes, or in distal regions of the same chromosome. When a transcript aligned to multiple genome loci, we selected the locus with the longest alignment. We did not determine the 'best' alignment when different genome loci gave the same aligned length because this occurred in < 1% of the assemblies. When multiple transcripts aligned to the same genome locus and we needed a single representative for our analysis, we selected the largest of these (putative) alternative splice forms.

3 RESULTS

To compare the performance of SOAPdenovo-Trans, Trinity and Oases, we assembled two sets of paired-end Illumina data, (L)arge and (S)mall, for rice and mouse. As both genomes were sequenced a decade ago, the annotation has been extensively curated, making these appropriate benchmarks to assess the assembly software. We chose to assess both plant and animal transcriptomes because most other studies only assessed animals (or even simpler organisms like yeast), and we wanted to be sure that our assembler could handle the difficulties created by plant data. Plants have larger gene families and more transposable elements (TEs); some of these TEs are also highly expressed. SOAPdenovo-Trans was designed for use in the 1000 plants (1 KP) initiative www.onekp.com, and thus it was essential to manage these difficulties.

We first assessed the computational demands of the three software programs with regard to peak memory and time (Table 1). For both measurements, SOAPdenovo-Trans was more than competitive with the other two programs.

Alignment of the assembled transcripts to the annotated genomes (Table 2) showed that SOAPdenovo-Trans produced the fewest transcripts, by more than factor of 2 in the most extreme cases, even after removing assemblies that were shorter than 300 bp. However, the number of annotated genome loci recovered was consistent among the three algorithms, differing only by <7%. One might naively attribute the differences in transcript numbers to alternative splice forms, but we would advise caution. There could be, for example, non-overlapping fragments of the same isoform or redundant copies of the same isoform.

The following analyses are focused only on those transcripts that aligned to genome loci with annotated genes. We used the terms series-A and series-B to denote the sets of transcripts that included or excluded putative alternative splice forms, respectively. Series-A includes all assembled transcripts, while series-B is a strict subset that includes only the largest assembled transcript for any given gene.

To properly assess the differences between assemblers, it is important to first understand how the rice and mouse assemblies differed from each other. Despite the fact that the rice and mouse datasets have similar amounts of raw input data, i.e. S and L datasets (S: rice versus mouse: 1.47 versus 1.83 Gbp; L: 5.98

Table 1. Computational requirements

Method	Rice				Mouse			
	Small dataset		Large dataset		Small dataset		Large dataset	
	Peak memory (GB)	Time (h)	Peak memory (GB)	Time (h)	Peak memory (GB)	Time (h)	Peak memory (GB)	Time (h)
SOAP*denovo*-Trans	10.7	0.2	29.3	0.8	10.5	0.3	16.7	1.0
Trinity	11	4.3	30	10.4	11	4.5	17	8.9
Oases	9.9	0.4	44.2	3.6	9.1	0.5	29.8	2.1

Note: All assemblies were processed with 10 threads, on a computer with two Quad-core Intel 2.8 GHz CPUs and 70 GB of memory, running CentOS 5.

Table 2. Classification of assembled transcripts

	Rice						Mouse					
	Small dataset			Large dataset			Small dataset			Large dataset		
	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases
All sizes	61 425	107 403	64 490	99 398	170 880	127 815	48 224	96 551	42 933	86 961	174 992	80 454
>300 bp	25 800	37 548	36 097	38 789	64 934	75 135	16 286	29 900	27 598	25 037	46 939	51 356
Correct	23 682	31 764	30 001	34 718	52 943	61 865	15 959	28 239	26 005	24 318	43 598	47 582
Correct (%)	91.8	84.6	83.1	89.5	81.5	82.3	98.0	94.4	94.2	97.1	92.9	92.7
Chimeric	526	2021	2185	1020	4736	4309	170	1101	757	439	2510	1967
Chimeric (%)	2.0	5.4	6.1	2.6	7.3	5.7	1.0	3.7	2.7	1.8	5.3	3.8
Series-A (includes AS)	21 630	28 799	27 666	28 074	43 694	53 994	13 068	22 205	21 645	16 868	29 689	36 309
Series-A (non-TE)	20 685	27 341	26 442	26 802	41 414	51 611	–	–	–	–	–	–
Series-A (%) (non-TE)	95.6	94.9	95.6	95.5	94.8	95.6	–	–	–	–	–	–
Series-B (excludes AS)	14 797	14 790	13 738	17 906	17 772	17 092	9486	9743	9205	10 511	10 777	10 268
Series-B (non-TE)	14 224	14 199	13 200	17 106	16 917	16 288	–	–	–	–	–	–
Series-B (%) (non-TE)	96.1	96.0	96.1	95.5	95.2	95.3	–	–	–	–	–	–

Note: Our analyses generated a successive reduction in the number of assemblies. First, we restricted our analyses to assemblies larger than 300 bp. BLAT alignments to the reference genomes were done at 95% sequence identity. Assemblies were deemed to be correct when ≥95% of their lengths could be accounted for in one consistent alignment. If not, assemblies were deemed to be 'chimeric' when 95% of their lengths could be accounted for in two or more alignments with different orientations, on different chromosomes or on distal regions of the same chromosome. We then confined our analysis to assemblies that overlapped with annotated genes. Because multiple assemblies could align to the same genome locus, we generated two datasets: series-A and -B, which included or excluded putative alternative splicing forms, respectively. In choosing among the isoforms, whether for series-B or the genome annotations, we always used the longest available sequence. In the case of the rice transcriptome, about 30.3% of the 55 986 annotated genome loci were known to be TEs, but our data showed that this was not a confounding issue. We indicate here the percentage of the assembled transcripts that were not known to be TEs.

versus 5.49 Gbp), the rice assemblies contained more genes than the mouse assemblies, ranging from 49–70% using the series-B gene counts. This higher number of genes was not due to differences in transposable element (TE) abundance for rice because >95% of the expressed rice genes were non-TEs. Given that many more rice genes had to be recovered from the same amount of sequence data, the read depths per gene were lower; as a result, the rice assemblies were not as high quality as the mouse assemblies. Furthermore, we expected that, given no extensive assembly errors (i.e. ones so extreme that they could not even be defined as chimeric), all but a very small percentage of the assembled transcripts should align to the genome. This was the case for mouse, but not for rice, where close to 10% failed to align because of subspecies differences, i.e. the use of *indica* transcriptomes and *japonica* genome annotations. We could eliminate most of the alignment failures by aligning the transcripts to combined genomes of both subspecies; however, to avoid the complications of having two genome annotations, we used only the alignments to the *japonica* genome.

Comparisons of the assembled and annotated transcript can, at least in principle, be complicated if the sequences represent different isoforms created from different combinations of exons. Under those circumstances, the concept of ‘full length’ cannot even be defined by the ratio of lengths. However, in practice, the overlap between the assembled and annotated transcript is almost always perfect (Fig. 2). Hence, the two sequences almost always represent the same isoform. This allowed us to simplify our calculations for deriving the next plot (Fig. 3), which presents the cumulant for the assembled transcript lengths versus the assembled-to-annotation length ratios. What this is meant to show is the extent to which full-length transcripts are recovered, for any definition of completeness, without having to choose an arbitrary threshold like 95% of 100%. The use of total length on the *y*-axis is meant to de-emphasize the fact that there are many small assemblies that, even in aggregate, do not amount to much. The ideal is a step function with a rapid increase at ratios near 100%, and SOAPdenovo-Trans came closer to this than did Trinity or Oases. Based on the ‘shoulder’ in the curve, the data indicated that SOAPdenovo-Trans using only 1.83 Gbp of mouse data outperformed Trinity when it used 5.49 Gbp of mouse data. Note also that the increase begins before ratios of 100%, meaning that in many instances the assembled transcript was longer than the annotated transcript, which is not unexpected because untranslated region (UTR) sequences tend to be poorly annotated.

To put a solid number on how many genes or isoforms were recovered, we chose an arbitrary threshold, 100 or 95% of the expected length in Table 3. Here, we only counted the isoforms that had been recorded in the genome annotations. While it is possible that the transcriptome data contained isoforms that had not previously been discovered, it is equally possible that these ‘putative alternative splice forms’ were assembly errors. The only way to avoid a misleading isoform count is to record only what had previously been annotated. Rather surprisingly, we found that Trinity and Oases did not recover more isoforms than

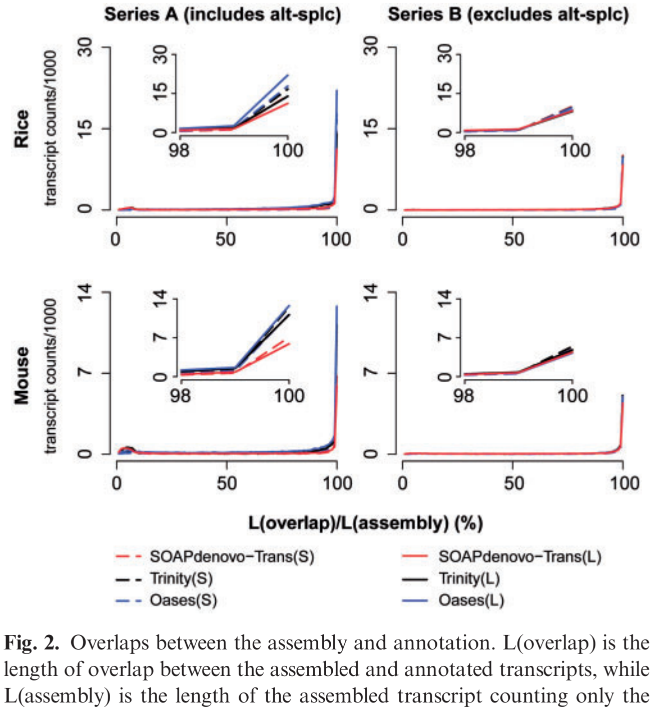

Fig. 2. Overlaps between the assembly and annotation. L(overlap) is the length of overlap between the assembled and annotated transcripts, while L(assembly) is the length of the assembled transcript counting only the portion that successfully aligned to the genome. Here, we show the distribution in the number of assembled transcripts as a function of the overlap-to-assembly lengths

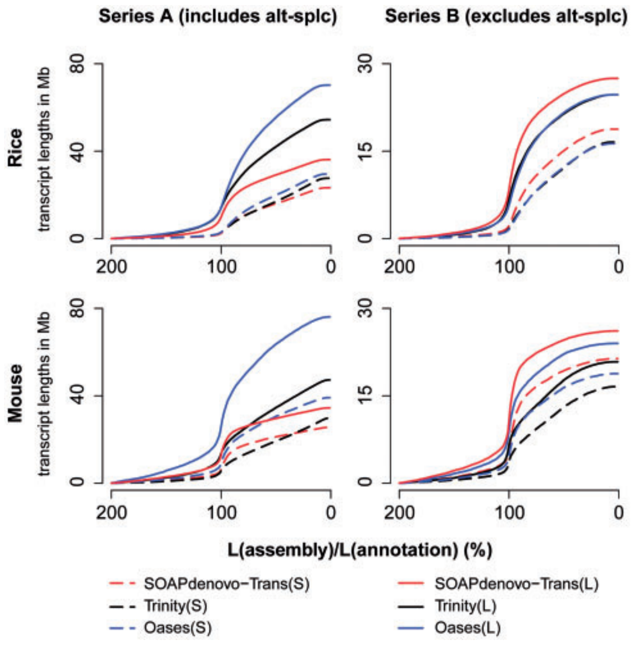

Fig. 3. Cumulants of assembled transcript lengths. In contrast to Figure 2, where we showed a distribution, here we plot a cumulant. L(assembly) is the length of the assembled transcript, counting only the portion that aligned to the genome, while L(annotation) is the length of the annotated transcript. Notice that the assembly-to-annotation lengths are plotted in reverse, from large to small. The ideal result is a step function with a sharp rise at 100%, but it begins to increase prior to 100% because the assembled transcripts contain UTRs that were not present in the annotated transcripts

Table 3. Evaluations based on number of ‘full-length’ annotations recovered

	Rice						Mouse					
	Small dataset			Large dataset			Small dataset			Large dataset		
	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases	SOAP*de novo*-trans	Trinity	Oases
Coverage = 100%												
Genes	386	472	355	1589	1657	1043	2897	3071	2984	4303	4449	4 192
Isoforms	405	524	382	1769	1914	1175	3505	3939	3922	5572	6193	6 298
Coverage ≥ 95%												
Genes	1904	1780	1469	5103	4434	3440	6000	5090	5563	7963	6674	7 211
Isoforms	2300	2 229	1849	6237	5633	4353	9043	7619	8975	12 663	10 784	13 114

Note: The alignment criterion is at least 95% sequence identity covering the entire (or ≥95%) annotation, and containing at most 5% insertions or deletions.

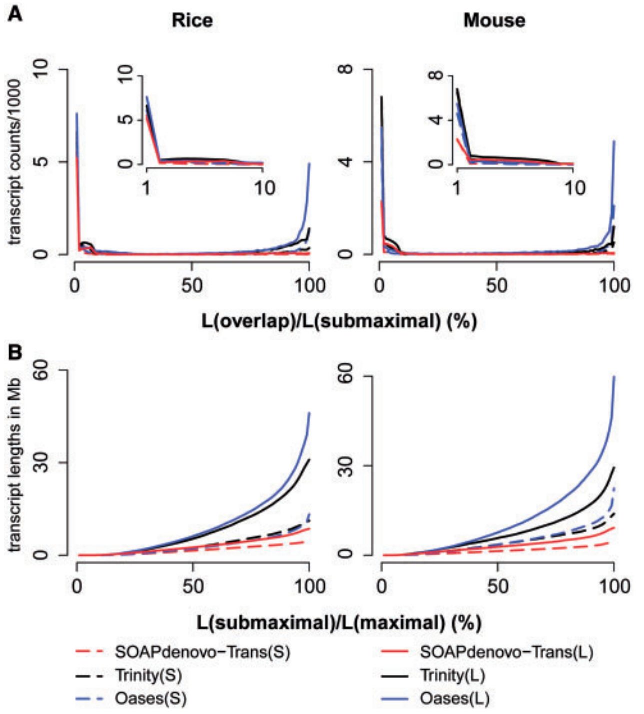

Fig. 4. Analysis of alternative splice forms. Given a set of assembled transcripts aligning to the same genome locus, L(submaximal) is the length of any transcript other than the largest, while L(maximal) is the length of the largest transcript. L(overlap) is the length of the overlap between the two. As in Figures 2 and 3, we show a distribution for the number of transcripts and then a cumulant for the transcript lengths

SOAPdenovo-Trans, even though they produced many more assemblies.

To investigate why the assemblers, especially Oases, generated so many putative alternative splice forms, we did a comparison of the submaximal transcripts (i.e. all but the largest of the many transcripts that aligned to a particular genome locus) to the maximal transcript (Fig. 4A). In many cases, we saw virtually no overlap between the submaximal and maximal transcripts, indicating that the assemblers produced non-overlapping fragments of the same isoform. In many other cases, the overlap to submaximal ratio was equal to one, which meant no new exons were recovered, unlike what is typically seen with genuine instances of alternative splicing. We noticed that the assemblers often produced multiple artifactual transcripts as a result of minor substitution errors in the raw input data. All had about the same length, in contrast to the common form of alternative splicing where exons are added or subtracted, which would result in 10–20% changes in the transcript lengths (e.g. 1 out of 10 exons in an animal gene or 1 out of 5 exons in a plant gene). We tested for artifacts of this type by plotting the cumulant for the transcript lengths as a function of submaximal-to-maximal lengths (Fig. 4B). The sharp increase as the ratios approach one showed that all the assemblers created artifacts of this type, but SOAPdenovo-Trans was the least offensive of the tested software.

4 DISCUSSION

Sequence assembly using real-world datasets has always required many subtle algorithmic changes to produce the best results, and it is clear that no single algorithm has a ‘magic bullet’ that solves all of the problems. We developed SOAPdenovo-Trans by combining novel concepts introduced by Trinity and Oases with concepts developed for the genome version of SOAPdenovo2. On top of this, we added modifications of our own, suitable for transcriptome studies. As demonstrated here, we believe we have produced an algorithm that substantially improves on the currently available tools for transcriptome assembly. Given the complexity of these analyses, however, SOAPdenovo-Trans is unlikely to be the final word in transcriptome assembly.

In particular, we tested one of the reference-based assemblers, Cufflinks, and found that it provided even better results than SOAPdenovo-Trans. These results suggest that, perhaps, there is information in these datasets that, with additional algorithm modifications, can be recovered. For example, a multiple *k*-mers strategy may improve transcriptome assembly. Current multiple *k*-mers assembly strategies generally fall into one of the two categories: (i) after using different values for *k*-mer assembly, separately, the resultant assemblies are merged into one final set. This might result in a more complete transcript set, but it may also introduce redundancy; (ii) iterate different *k*-mer DBG assemblies during contig construction. This strategy could potentially make the best use of reads and paired-end information, but whether it is worth developing such an algorithm depends in part on the ongoing developments in sequencing technology. There is an expectation of improvements in read lengths in the future. If so, it would necessarily alter the types of issues faced by transcriptome analysis.

Finally, SOAPdenovo-Trans, unlike Trinity and Oases, does not yet perform strand-specific assembly, and this is planned for a future development to further improve this algorithm.

Funding: The State Key Development Program for Basic Research of China-973 Program (NO.2011CB809202); Shenzhen Key Laboratory of Transomics Biotechnologies (CXB201108250096A); China National GeneBank; and the 1000 plants (1KP) initiative, which is led by G.K.S.W. and funded by the Alberta Ministry of Enterprise and Advanced Education, Alberta Innovates Technology Futures (AITF) Innovates Centre of Research Excellence: (iCORE), and Musea Ventures.

Conflict of Interest: none declared.

REFERENCES

Grabherr,M.G. *et al.* (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.*, **29**, 644–652.

Guttman,M. *et al.* (2010) *Ab initio* reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat. Biotechnol.*, **28**, 503–510.

Kent,W.J. (2002) BLAT–the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Li,R. *et al.* (2010) *De novo* assembly of human genomes with massively parallel short read sequencing. *Genome Res.*, **20**, 265–272.

Lu,B. *et al.* (2013) Comparative study of *de novo* assembly and genome-guided assembly strategies for transcriptome reconstruction based on RNA-Seq. *Sci. China Life Sci.*, **56**, 143–155.

Luo,R. *et al.* (2012) SOAPdenovo2: an empirically improved memory-efficient short-read *de novo* assembler. *Gigascience*, **1**, 18.

Maher,C.A. *et al.* (2009) Transcriptome sequencing to detect gene fusions in cancer. *Nature*, **458**, 97–101.

Martin,J. *et al.* (2010) Rnnotator: an automated *de novo* transcriptome assembly pipeline from stranded RNA-Seq reads. *BMC Genomics*, **11**, 663.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, **5**, 621–628.

Robertson,G. *et al.* (2010) *De novo* assembly and analysis of RNA-seq data. *Nat. Methods*, **7**, 909–912.

Schulz,M.H. *et al.* (2012) Oases: robust *de novo* RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics*, **28**, 1086–1092.

Simpson,J.T. *et al.* (2009) ABySS: a parallel assembler for short read sequence data. *Genome Res.*, **19**, 1117–1123.

Surget-Groba,Y. and Montoya-Burgos,J.I. (2010) Optimization of *de novo* transcriptome assembly from next-generation sequencing data. *Genome Res.*, **20**, 1432–1440.

Trapnell,C. *et al.* (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**, 511–515.

Yu,J. *et al.* (2005) The Genomes of Oryza sativa: a history of duplications. *PLoS Biol.*, **3**, e38.

Zerbino,D.R. and Birney,E. (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res.*, **18**, 821–829.

Zhang,G. *et al.* (2010) Deep RNA sequencing at single base-pair resolution reveals high complexity of the rice transcriptome. *Genome Res.*, **20**, 646–654.