SCIENCE CHINA
Life Sciences

SPECIAL TOPIC
• RESEARCH PAPER •

February 2013 Vol.56 No.2: 143–155
doi: 10.1007/s11427-013-4442-z

Comparative study of *de novo* assembly and genome-guided assembly strategies for transcriptome reconstruction based on RNA-Seq

LU BingXin[1,2], ZENG ZhenBing[2*] & SHI TieLiu[1*]

[1] *Center for Bioinformatics and Computational Biology, Shanghai Key Laboratory of Regulatory Biology, Institute of Biomedical Sciences and School of Life Sciences, East China Normal University, Shanghai 200241, China;*
[2] *Software Engineering Institute, School of Software Engineering, East China Normal University, Shanghai 200062, China*

Received August 2, 2012; accepted October 20, 2012

Transcriptome reconstruction is an important application of RNA-Seq, providing critical information for further analysis of transcriptome. Although RNA-Seq offers the potential to identify the whole picture of transcriptome, it still presents special challenges. To handle these difficulties and reconstruct transcriptome as completely as possible, current computational approaches mainly employ two strategies: *de novo* assembly and genome-guided assembly. In order to find the similarities and differences between them, we firstly chose five representative assemblers belonging to the two classes respectively, and then investigated and compared their algorithm features in theory and real performances in practice. We found that all the methods can be reduced to graph reduction problems, yet they have different conceptual and practical implementations, thus each assembly method has its specific advantages and disadvantages, performing worse than others in certain aspects while outperforming others in anther aspects at the same time. Finally we merged assemblies of the five assemblers and obtained a much better assembly. Additionally we evaluated an assembler using genome-guided *de novo* assembly approach, and achieved good performance. Based on these results, we suggest that to obtain a comprehensive set of recovered transcripts, it is better to use a combination of *de novo* assembly and genome-guided assembly.

transcriptome reconstruction, RNA-Seq, *de novo* assembly, genome-guided assembly

Citation: Lu B X, Zeng Z B, Shi T L. Comparative study of *de novo* assembly and genome-guided assembly strategies for transcriptome reconstruction based on RNA-Seq. Sci China Life Sci, 2013, 56: 143−155, doi: 10.1007/s11427-013-4442-z

Transcriptome refers to all the transcripts and their corresponding quantity in cells, under specific development stages or physiological conditions [1]. Transcriptome reconstruction is the process of identifying transcripts and isoforms (splice variants of a gene) which are expressed in a specific sample [2]. According to Martin et al., the assembly of transcriptome aims to obtain a collection of fragments, each one representing a full-length transcript as accurate as possible, from short reads [3]. As a fundamental step for further study of transcriptome [4], assembly can be utilized to discover novel transcripts and previously unknown genes, compute gene expression level and so on. Even for those well annotated transcriptome, assembly can still be used to improve genome annotation [5]. In recent years, RNA-seq, an increasingly widespread next-generation deep-sequencing approach to transcriptome profiling, provides the promise of a much more comprehensive study of transcriptome in a cost effective way during considerably less time [1]. RNA-seq can detect transcripts not corresponding to existing genomic sequence and genes with low expression levels. Most importantly, it owns the capability to reconstruct a

*Corresponding author (email: zbzeng@sei.ecnu.edu.cn; tieliushi01@gmail.com) complete picture of the transcriptome across diverse conditions in theory [2].

Some genome assemblers have already succeeded in assembling transcriptome from RNA-seq data [6,7]. However, different from genome assembly, there are some specific challenges for transcriptome assembly [2,8,9]. The first issue is a wide range of gene expression levels, which leads to non-uniform sequence coverage. Lowly expressed genes may be only partially covered by a few reads and hard to be recovered to full length. The second critical problem is to handle pervasive alternative spliced isoforms. One gene may have several isoforms, and reads are too short to tell which isoform they are from. Thirdly, homologous and repeated sequences, similar isoform sequences derived from the same gene may cause ambiguities in assembly. Fourthly, it is problematic to discriminate exons and introns which may originate from incompletely spliced precursor RNAs. Moreover, assembly methods are limited by current sequencing library protocols. For instance, samples that are strand-specific require assembly method to take into account the strand orientation to tease apart overlapping transcripts.

Up to now several transcriptome assemblers have been developed to handle these challenges. Depending on the usage of a reference genome, these assembly methods fall into two major categories: genome-guided assembly and *de novo* (or genome-independent) assembly [2,4,10,11]. Genome-guided transcriptome assembly strategy refers to first aligning sequencing reads to a reference genome and then assembling overlapping alignments into transcripts. In contrast, *de novo* transcriptome assembly method directly reconstructs overlapping reads into transcripts by utilizing the redundancy of sequencing reads themselves. For organisms without reference genome, only *de novo* assembly is possible, but for those with reference genome, both approaches are workable, although it is generally preferable to employ genome-guided assembly. Moreover, it is very likely that *de novo* assembly can effectively complement the results of genome-guided assembly in the presence of reference genome sequences.

To understand comprehensively the differences and connections between the two kinds of transcriptome reconstruction methods when being applied to organisms with reference genome, we selected five representative assemblers: cufflinks [12,13] and scripture [14] from genome-guided category; trans-ABySS [8], trinity[9] and oases [15] from *de novo* category, and then we investigated their common and different points from various aspects. Firstly the similar and different algorithm features of each assembler in either category were studied. Subsequently based on several common metrics, real performances of the five assemblers were compared to get an idea of their pros and cons in practice. According to the results, each assembler had its specific superiority in certain aspects on some datasets, and furthermore either assembly strategy demonstrated unique features on the same data. Finally, we suggest that it may be better to combine genome-guided assembly and *de novo* assembly together to get a more accurate and complete transcriptome assembly. To validate this assumption, we tested two possible combination approaches: one is to merge the results of each assembly, and the other is to use a genome-guided *de novo* assembler. The final combined output from the first method significantly improved separate assembly, and the second method also provides a good result despite it seems to need modifications to achieve better performance. Moreover, the second strategy provides insights into finding an efficient hybrid assembly method.

1 Materials and methods

1.1 Data used

To find out algorithm features of each assembler, related literatures and source codes were studied. To test the real performances of these assembly methods, three Illumina RNA-Seq datasets from three species—human, mouse and *Schizosaccharomyces pombe* were selected respectively. These species all have relatively complete gene annotation, so they are convenient for assessment of transcriptome reconstruction methods. Additionally the reconstruction of *S. pombe* transcriptome is challenging for very short introns and dense transcripts. The *S. pombe* genome pombe_09052011.fasta was downloaded from the Sanger Institute. All the other reference genome sequences and annotation files were obtained from Ensembl, with versions GRCh37.64 for human, NCBIM37.64 for mouse, Schizosaccharomyces_pombe.EF2.13.gtf for *S. pombe*. The reference annotation file of human was further filtered to include only chromosome 1–22, MT, X, Y. Similarly, the reference annotation file of mouse was filtered to contain just chromosome 1–19, MT, X, Y.

In the following, the three datasets are simply refered to as brain, esc, spombe respectively. The brain dataset accessed by GSE30222, is from several regions of the brain of 23 adult donors with 22 Caucasian and one unknown. The esc dataset accessed by GSE20851, is from mouse embryonic stem cells and initially used in the evaluation of scripture. The spombe dataset accessed by SRP005611, is firstly used in the test of trinity, and only data for late stationary phase of *S. pombe* was downloaded for analysis. Detailed information of them is shown in Table S1. Sequencing coverage for each dataset was estimated by Lander/Waterman equation [16] for computing coverage, with the haploid genome length replaced by approximate transcriptome size of each species (human ~266 Mbp, mouse ~175 Mbp, *S. Pombe* ~13 Mbp). According to the results, these datasets have gradually increasing coverage (brain ~22×, esc ~50×, spombe ~499×).

1.2 Software used and parameter settings in assembly process

Almost all the five transcriptome assemblers are updated from time to time, so some software used in the comparison study was not the latest version. Here assemblers used were: cufflinks v2.0.0, scripture VPaperR3 (scripture-beta2.jar), ABySS [6,17] v1.3.3, trans-ABySS v1.3.2, trinityrnaseq_ r2012-04-27, velvet [7] v1.2.03, oases v0.2.06. To avoid merging pair end files, velvet v1.2.07 was used for dataset esc and spombe. Other related software used in the assembly was: bowtie [18] version 0.12.7, samtools [19] version 0.1.18.0, tophat [20] v2.0.0, blat [21] v34×12. All the assemblers were run on a 512G NUMA server with Intel Xeon CPU of 2.67 GHz as well as 48 cores, and the operating system installed was CentOS Linux release 6.0. To ensure the good quality of sequencing reads, fastx_toolkit (http://hannonlab.cshl.edu/fastx_toolkit/index.html) was applied to remove reads with a Phred quality score less than 20 over the 80% nt on brain dataset, and sickle (https:// github.com/ucdavis-bioinformatics/sickle) was utilized to filter the other two pair-end datasets with option '-q 2' in order to keep the integrity of pair end reads. Besides, fastQC (http://www. bioinformatics.babraham.ac.uk/pro- jects/ fastqc/) was used to check the quality of the three datasets both before and after filtering.

In the assembly, most parameters for the software were set as default. Some critical parameters are listed in Table S2. Only the first step of trans-ABySS (FEM) was employed. Since oases and trans-ABySS use multiple *k*-mers to achieve both higher sensitivity and specificity, 8 *k*-mers were carefully chosen according to their manuals and read length on each dataset. However, to get optimal results, some parameters should be tested many times, especially *k* value. In this analysis, some parameters may not be the best ones, but they can still give a somewhat fair comparison between these assemblers. Cufflinks have two different assembly modes: with or without the help of reference annotation. Both modes were run with the same other parameters in the experiments, and cufflinks assembly using annotation is represented as cufflinks (RABT). Cufflinks (RABT) employed full annotation files, but the filtered annotation files were used in all downstream analysis. For spombe dataset, it was suggested that blat was better than tophat when aligning reads to genome [9], but from the comparison of mapping results (Table S3), blat had no apparent advantages. Thus to avoid problems of format converting, spliced alignment from tophat, rather than blat, was taken as input to subsequent genome-guided assembly.

1.3 Assessment metrics of assembly results

Transfrag (transcribed sequence fragment) here refers to assembly output of an assembler, while transcript locus denotes a locus consisting of a set of transcripts which does not overlap with genomic locations of another set in any other locus. The overall analysis process is shown in Figure 1.

Since there are no standard metrics to assess transcriptome assembly, several simple standards were extracted from existing indexes used in related literatures. Firstly transcriptome assemblers can be assessed from five aspects: accuracy, completeness, contiguity, chimerism and variant resolution [4], so we used the five criteria to get an overall assessment. Similar to the definitions in Ronnator [3], here accuracy is defined by the percentage of transfrags that share at least 95% identity with the reference genome by at least 80% of the transfrag length; completeness denotes the percentage of known transcripts covered by transfrags to at least 80% of the transcript length; contiguity is the percentage of known transcripts covered by a single transfrag over at least 80% of the transcript length; chimerism is identified as the fraction of transfrags that overlap with more than

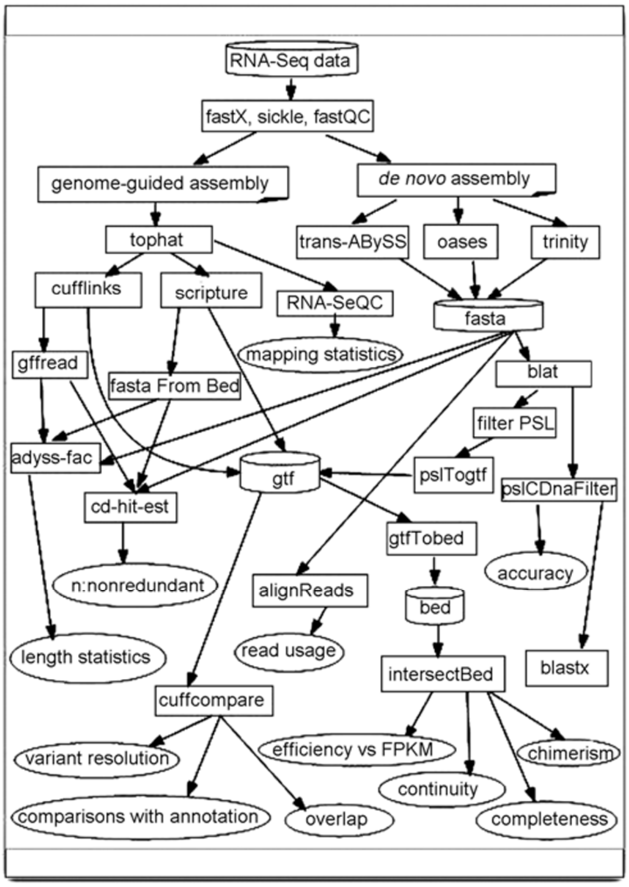

Figure 1 Flow diagram showing the analysis process in the comparative study. RNA-seq data was first filtered and then input to genome-guided assemblers and *de novo* assemblers. Generally, the original assembly results are fasta files for *de novo* assemblers, and gtf files for genome-guided assemblers. These files were further processed by various programs and scripts to calculate metrics used in the study. Shape cylinder represents data file. Shape rectangle represents program or script used for analysis. Shape rectangle with an angle represents transcriptome reconstruction strategy. Shape eclipse represents the final metric.

50% of two or more annotated transcripts. Lastly variant resolution is simply represented by average number of isoforms per annotated gene.

Genome-guided assembly methods are based on spliced alignment, so the output transfrags are largely identical to corresponding sequences in the reference genome, reducing the necessity of mapping transfrags to genome again. To validate transfrags from three *de novo* assemblers, original fasta files were mapped to reference gnome by blat with default parameters, except for dataset spombe which was imposed with restraint max intron length equaling to 900. To get the accuracy of transfrags reconstructed by *de novo* assemblers, all mapped transfrags were filtered by pslCDnaFilter requiring at least 95% sequence identity across at least 80% of the transfrag length. The final percentage of accuracy was then calculated by dividing the number of remained transfrags by the number of all recovered transfrags. The transfrags unaligned and failing to pass the filter were then blasted against non-redundant protein sequences (nr) database with *e*-value 1×10^{-5} to check their fidelity. Completeness and contiguity were calculated by dividing the total number of transcripts instead of only those expressed, because cufflinks (RABT) can reconstruct transcripts that are even not expressed in the sample. The computation of completeness, contiguity and chimerism was aided by intersectBed from Bedtools [22], with option '-split -wo' (additional option '-f 0.5 -s' for the computation for chimerism) and a little customization of output format to get all overlap intervals between transfrags and annotated transcripts. The output files were then parsed by custom perl scripts to calculate exact length of each overlap. Variant resolution was obtained by parsing refmap file output by cuffcompare to get the number of unique transcripts corresponding to each known gene.

Since there is no single metric that could determine which assembler is superior, then other various metrics were also computed to compare the five assemblers more comprehensively, including computing resources usage, transfrags number and length, sequencing read usage, comparisons with reference annotations. Computing resources usage was obtained by time command in linux, with elapsed wall clock time and max resident memory being extracted. Transfrags length statistics were calculated by abyss-fac.pl from ABySS package. For oases, the final length of each transfrag in original fasta file was actually L+k–1, where L is the length of the transfrag and k=27. However, for the convenience of comparison, all the length statistics were obtained directly from the original fasta files of *de novo* assemblers, as 26 is relatively smaller and causes little effect on final results. Cd-hit-est [23] was used to explore redundant transfrags (with parameters -c 1). However, to avoid losing information, all transfrags were used for further analysis. To detect the percentage of read usage, which is the fraction of sequencing reads really used in assembly, sequencing reads input to assembly were aligned back to transfrags by bowtie, using alignReads.pl from trinity package. In addition, read usage for velvet-oases was acquired directly from log files to check the results of bowtie. At the same time, RNA-SeQC [24] v1.1.7 was utilized to analyze the spliced mapping results which were input to genome-guided assemblers, to explore the impact of mapping on genome-guided assembly. The initial psl file output by blat was filtered by filterPSL.pl from augustus [25] package to retain only the unique best alignment which was then converted to gtf format by custom perl script. Scripture output was converted to gtf format by its task 'toGFF' too. Then all the gtf files were compared with reference annotation by cuffcompare. Several important standards were extracted from the cuffcompare outputs.

Furthermore, relationship between transcripts reconstruction results and expression level of known transcripts was analyzed on three datasets, since gene expression level has a significant impact on the performance of assemblers. Actually a robust transcriptome reconstructing method should recover transcripts of diverse expression levels. Thus it is meaningful to explore the connection between the efficiency of transcriptome reconstruction by each assembler and the abundance of transcripts. FPKM of annotated transcripts was calculated by cufflinks with option '-G', filtered by FPKM>0.0005 and then divided into 20 quantiles. Subsequently median value for coverage of a known transcript (as fraction of the reference transcript length) by a single transfrag is calculated in each quantile. Meanwhile, in order to get the correlation between transfrags output by different assemblers, overlap of recovered full-length annotated transcripts and novel transcripts, which were obtained from parsing tmap files output by cuffcompare, among five assemblers were also examined, with cufflinks (RABT) being excluded. Here full-length annotated transcripts are those matching intron chain completely with a transfrag, and novel transcripts are those sharing at least one splice junction with a transfrag.

2 Results

2.1 Comparisons of algorithm features

All the transciptome assembly algorithms used in the five assemblers are finally deduced to graph reduction problems. Because it is hard to find optimal graph reductions in assembly [26,27], real assemblers often rely on heuristic algorithms and approximation algorithms to get approximate solutions, such as removing redundancy, correcting errors, discarding uncertainty, reducing complexity. At the same time, pragmatic engineering techniques are utilized to resolve difficulties in real conditions, like random errors and systematic biases in sequencing reads, as well as physical limitations of computers when handling large volumes of data. Similar to genome assemblers, it seems that the success of a transciptome reconstruction assembler also largely depends on its heuristics [28]. Different from genome assembly, transcriptome assembly has to construct a graph, instead of a linear consensus sequence, to cope with alternative splicing [29]. Generally, splicing graph can be used to represent the complex structures of alternative splicing transcripts[30,31]. Splicing graph can be built with or without the presence of reference genome, and therefore can be applied in both genome-guided and *de novo* transcript assembly. However, in practice each assembler has its own representation of graphs, resulting in specific graph simplification and traversing methods.

Cufflinks [32] solves the assembly problem by finding a maximum matching in a weighted bipartite graph derived from a partial order (directed acyclic graph) on overlapping compatible spliced alignments, which permits polynomial time complexity. It finds only the minimal transcripts sufficient to explain all the splicing events in the sequencing data, which will then be used for abundance quantification. By assembling fragments in each locus separately it uses few computing resources. It can also utilize reference annotation to improve assembly, which is abbreviated as RABT (reference annotation based transcript assembly).

By contrast, scripture is designed to reconstruct all transcripts expressed at significant levels in the sequenced sample, so it will find all possible transcripts useful for annotation. It transforms assembly into a statistical segmentation problem by searching significant paths instead of significant exons in a connectivity graph constructed from spliced alignments. Then it builds a transcript graph to extract isoforms. It assembles each chromosome separately which can be parallelized and in turn reduce resource usage significantly.

Trans-ABySS actually is a bunch of scripts post-processing the results of parallel *de novo* genome assembler ABySS which assembles sequencing reads several times by de Bruijn graph using different *k*-mers. Trans-ABySS merges all the contigs output by ABySS into a non-redundant set of transfrags. It mainly analyzes junction contigs of 2(*k*–1) bp, composed of two (*k*–1) overlaps, to get spliced isoforms. Because ABySS is a distributed assembler aimed to address memory limitations of large data, it consumes much less computing resources than other *de novo* assemblers. Besides assembly, trans-ABySS can also predict polyadenylation sites, identify gene fusion, and compute a gene-level expression metric.

Conversely, trinity [33] is a 3-module assembler specifically designed for transcriptome reconstruction, composed of inchworm, chrysalis and butterfly. Firstly inchworm assembles reads in a greed way and usually results in a set of full length contigs for major isoform as well as unique portions of minor spliced variants. Chrysalis then clusters output from inchworm into connected components which are likely to represent alternative splice forms and closely related paralogs, and builds de Bruijn transcript graphs for each component. Finally butterfly processes each generated graph and enumerates full length alternatively splice isoforms and transcripts from paralogous genes. This modularity offers a flexible way to extend trinity. It has been suggested that some parts of trinity can be replaced by other more efficient modules [33]. For example, part of inchworm has already been substituted by jellyfish, a faster method to count *k*-mer abundance in parallel.

Lastly, oases combines the multiple *k*-mer strategy in trans-ABySS with a similar toplogical analysis like trinity, trying to deal with a broad spectrum of expression levels and alternative isoforms. It is built upon velvet and post-processes preliminary contigs output from a single *k*-mer assembly of velvet and further continues to construct a transcript de Bruijn graph followed by special topological analysis to extract isoforms. It runs with multiple k-mers and merges all the transfrags from different *k*-mers by oases-M in the end. Its dynamic error removal methods largely contribute to its robustness.

Although all the five assemblers have different details of implementations, they still share a core set of features. These features can be subdivided into the following aspects:

(i) models to deal with features of reads
(ii) graph construction
(iii) graph reduction and transcripts extraction
(iv) ways to resolve specific challenges
(v) support for parallel computing

Details about the similarities and differences between the five assemblers are shown in Table S4.

In principle, genome-guided assembly has several inherent advantages over *de novo* assembly [2,4,30]. In the first place, spliced mapping can partition very similar reads or reads from paralogous regions, based on their different genomic loci, into individual sets which can then be reconstructed seperately, reducing required computing resources to a large extent. Secondly, sequencing errors, which will increase the complexity of assembly a lot, can also be resolved by mapping reads to genome. In addition, proximal aligned reads can be grouped into exons, and splice sites can help to extend exon boundaries. Moreover, genome-guided assembly is quite sensitive to lowly expressed transcripts since they can be mapped to reference genome. However, *de novo* assembly still has its merits. Given sufficient read coverage, *de novo* assembly can still efficiently assemble a complete transcriptome. Furthermore, limitations imposed by spliced alignment, like errors and artifacts, may negatively influence the following genome-guided assembly and lead to false positive isoforms, while *de novo* assembly methods do not have this issue. Being independent of reference genome instead helps *de novo* assembly discover new transcripts not in the reference due to missing genes [34], structual variants or other reasons, identify transcripts with long introns, and detect special events like trans-splicing, chromosomal rearrangements and so on. These differences in algorithm features of each assembly category as well as the five assemblers themselves may lead to dissimilar performances of the five assemblersin reality [4].

2.2 Comparisons of real performance

2.2.1 Overall assessment of assembly results

The results of assessing five assemblers on three datasets by five overall metrics are shown in Table 1. For dataset esc and spombe, trans-ABySS had the highest accuracy followed by trinity and then oases. But for dataset brain, trans-ABySS was the least accurate one. For dataset esc the accuracy of oases was much lower than the other two *de novo* assemblers, while trinity performed poorest on dataset spombe in term of accuracy. The properties of each dataset may have a significant impact on each *de novo* assembler. For example, trans-ABySS may be incapable of handling single end samples efficiently, and oases may be less capable of dealing with non-strand-specific reads in dataset esc. It is worth noting that both oases and trans-ABySS produced transfrags with polyA tails of at least 35 bp while trinity did not. For dataset brain, trans-ABySS had 17694 transfrags with such polyA tails and oases had 4152. For dataset esc, oases had 1035, and trans-ABySS had 209. For dataset spombe, oases had 72, and trans-ABySS had 3. These long polyA tails may cause less transfrags mapped to reference genome and contribute partly to low accuracy of trans-ABySS on brain dataset and oases on esc dataset. Of the unaligned transfrags for each *de novo* assembler, oases had 2479 blastx hits on dataset esc, 1310 on dataset spombe, while trinity had 2118 hits on dataset esc, 1900 on dataset spombe, and trans-ABySS had 8020 hits on dataset esc, 96 on dataset spombe. Thus these transfrags not mapped to reference genome may still be bona fide transcripts.

For shallow sequencing dataset brain, the completeness and contiguity of genome-guided assemblers were obviously much higher than *de novo* assemblers. However, with the increase of sequencing depth, the completeness and contiguity of *de novo* assemblers increased much quicker than genome-guided assemblers, even up to values higher than genome-guided assemblers for ultra-deep sequencing spombe dataset. Among *de novo* assemblers, trinity had the largest completeness and contiguity, followed by oases and trans-ABySS, except that on dataset spombe where trans-ABySS had slightly larger completeness than oases. While among genome-guided assemblers, cufflinks was better than scripture. Cufflinks (RABT) had best completeness and contiguity among all assemblers, but still not exactly as 100%, which may be partly owing to artifacts introduced by additional annotation information.

With respect to isoform resolution, except for dataset spombe, different assemblers detected various isoforms per gene. These isoforms may be truly expressed transcripts or just assembly artifacts, which need to be further validated. Normally *de novo* assemblers found more average number of isoforms per gene for dataset brain and esc, except that

Table 1 Overall assessment of transcriptome assemblers from five aspects (accuracy, completeness, contiguity, chimerism and variant resolution)

Dataset	Assembler	Accuracy (%)	Completeness (%)	Contiguity (%)	Chimerism (%)	Variant resolution
brain	oases	86.9	23.14	7.57	0.20	4.1
	trinity	94.1	23.45	8.59	0.84	4.0
	trans-ABySS	80.7	18.70	3.08	0.38	4.9
	cufflinks(RABT)		99.53	99.53	1.78	3.0
	cufflinks		26.45	14.21	5.78	2.7
	scripture		21.70	11.39	2.78	3.0
	cuffmerge		31.34	19.63	1.52	2.9
	inchworm	96.9	17.10	4.88	0.55	4.9
esc	oases	70.4	35.03	24.47	0.20	2.3
	trinity	93.4	40.43	25.18	0.78	2.1
	trans-ABySS	97.9	31.87	19.15	0.98	2.5
	cufflinks(RABT)		99.61	99.61	2.66	2.4
	cufflinks		36.34	31.30	14.12	1.4
	scripture		32.18	25.39	3.67	1.6
	cuffmerge		44.89	39.86	2.15	1.6
	inchworm	95.7	37.69	20.23	0.24	3.2
spombe	oases	97.1	84.47	79.76	0.07	1.0
	trinity	88.4	90.93	80.82	0.92	1.0
	trans-ABySS	99.4	85.03	78.44	0.52	1.0
	cufflinks(RABT)		98.82	98.82	1.56	1.0
	cufflinks		73.78	72.07	22.09	1.0
	scripture		64.90	62.26	10.24	1.0
	cuffmerge		92.14	91.52	11.16	1.0
	inchworm	98.4	68.98	36.21	0.03	1.0

cufflinks (RABT) obtained more mean number of isoforms per gene for dataset esc. Trans-ABySS generated most average isoforms per gene among *de novo* assemblers, followed by oases, and trinity found the least. Scripture yielded more isoforms per gene than cufflinks, which is a natural result of their opposite design ideas on parsimony.

Genome-guided assemblers produced more fused transcripts, and this may be due to improperly merging adjacent genes on the same strand sometimes. It seems that they yielded more fused transcripts as read coverage increases, and achieved highest fusion in data from *S. pombe* which has the deepest sequencing coverage. Higher gene density of *S. pombe* may also be a possible reason for extremely high chimerism on dataset spombe. In contrast *de novo* assemblers had chimerism less than 1 across all the three datasets, alleviating the fusion problem effectively. Oases, in particular, generated fewest chimeric transcripts, even less than 0.25 of the huge number of transfrags it produced on each dataset.

2.2.2 *Other various metrics of assembly results*

(i) Computing resources usage. Generally, *de novo* assemblers consume much more time and memory than genome-guided assembly [2], which is shown in Figure 2. Being consist with previous results [35], oases consumed most memory but has a higher speed. Without adapting to a computing grid and using just default parameters, trinity used quite a long time to finish assembly. It also took up large memory, sometimes even larger than oases. The usage of disk by oases and trinity was also substantial in the experiments and became a bottle block to use the programs on limited space. By contrast, trans-ABySS finished assembly with less memory and disk space in a not long period of time. Both cufflinks and scripture used similar amount of memory and time, much less than *de novo* assemblers. For oases, the choice of *k*-mer is essential, and using a too small *k*-mer lead to a very large graph and in turn much more memory and time were needed to process the graph. Time for trinity to assembly brain dataset was quite long partly because some butterfly commands failed for the first time and had to rerun for about 24 h. Time for trans-ABySS to assembly esc dataset were long partly due to the failure of *k*-mer 62 which cost about 14 h for the second run. In other words, it took trinity and trans-ABySS quite a long time to handle complex data. Probably tuning of some parameters can reduce resource usage to some extent. According to Robert et al. [36], 2010-06-08 version of trinity has largely improved runtime performance by optimizing some critical components. Nonetheless, *de novo* assemblers is much more computing resource demanding.

(ii) Transfrag number and length statistics. As for the statistics of transfrags number and length (Table S5), *de novo* assemblers obtained much more transfrags than genome-guided assemblers, which may contribute to alignments reducing plenty of artifacts. Typically, oases and trans-ABySS reconstructed several folds of fragments than other assemblers, largely due to the merge of multiple assemblies. Scripture detected more transfrags than cufflinks, which reveals again that cufflinks is more conservative.

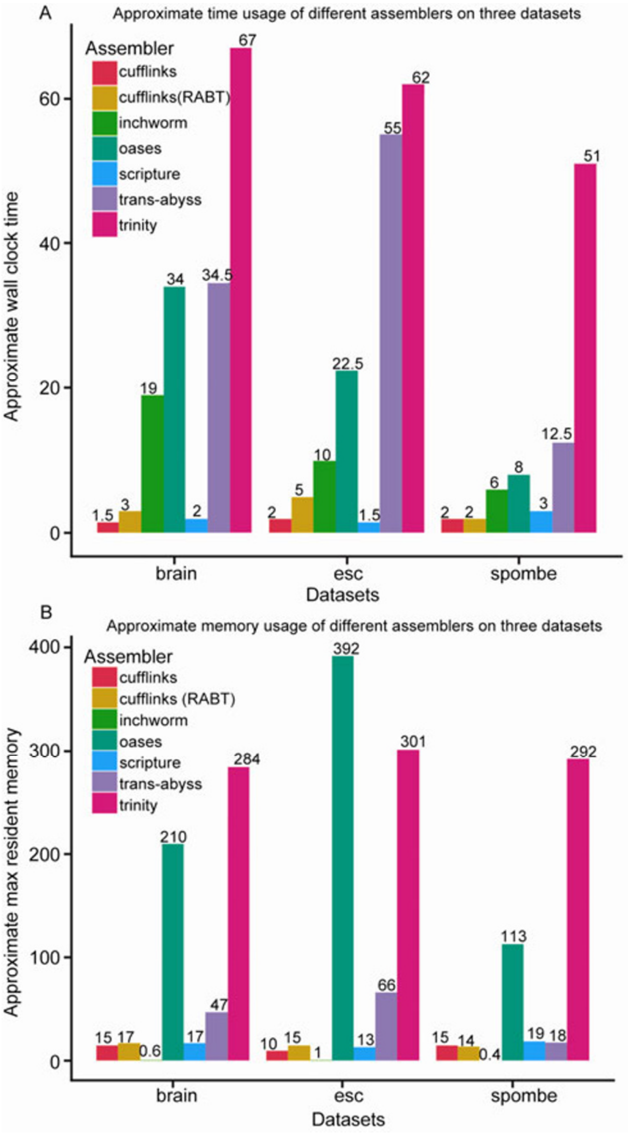

Figure 2 Resource usage of each transcriptome assembler on three datasets. A, Runtime usage of each transcriptome assembler on three datasets. The time unit of numbers on the top of each bar is hour. B, Max memory usage of each transcriptome assembler on three datasets. The memory unit of numbers on the top of each bar is gigabyte.

On the other hand, genome-guided assemblers produced much longer fragments, as illustrated in Figure 3. The length distribution of transfrags from cufflinks (RABT) is very similar to that of known transcripts, so it can be used as a benchmark. It is not difficult to see that cufflinks recovered least short transcripts across all datasets, partly because split alignments cannot detect very short exons. Scripture had the same issue, but it reconstructed more short transfrags than cufflinks. Moreover, scripture recovered far

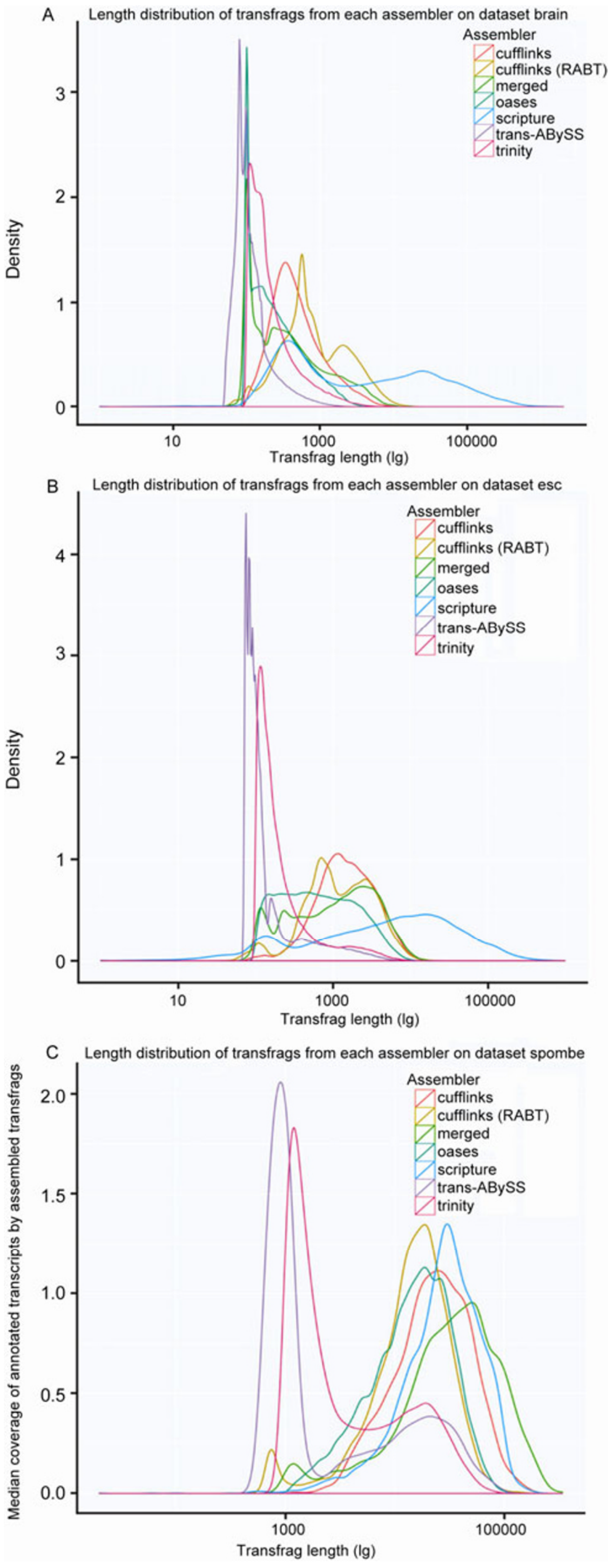

Figure 3 Length distribution of transfrags from each assembler on three datasets. Transfrag length is transformed in lg scale, but tick labels for the *x*-axis show the original data values. 'Merged' represents result of cuffmerge.

more transfrags which are longer than known transcripts. Apparently most transfrags from *de novo* assemblers were very short. Typically trans-ABySS had lots of fragments below 100 bp as a result of using transfrags of length 2(*k*–1) for analyzing alternative splicing variants. Trans-ABySS also got more fragmented transfrags for dataset brain in the absence of pair ends. Although *de novo* assemblers, like oases, may sometimes produce extremely long transfrags, they still yielded lots of short fragments, thus mean length and N50 of transfrags from them were still lower than genome-guided assemblers. However, the gap gradually decreases as the sequencing coverage goes up. Notably the length distribution of transfrags from oases tends to be like those of genome-guided assemblers, especially on dataset spombe. In all, oases yielded longest fragments among *de novo* assemblers, followed by trans-ABySS and then trinity.

In addition, the output of oases and scripture were much more redundant. For oases the possible reason is very likely due to the lack of efficient filter steps after merging multiple *k*-mers, although it indeed tries to remove redundant transfrags (identical or included in others) in oase-M. For scripture, it may be owing to particularly long transfrags which contain many short ones entirely.

(iii) Sequencing read usage. With respect to read usage, oases obviously employed much less reads to reconstruct transcripts compared with trinity and trans-ABySS, which utilized similarly larger amount of reads (Figure S1). It seems that as sequencing depth grows, the percent of read usage is also up. Table S6 further demonstrates read usage of oases, which is closely related to the settings of *k*-mer length and coverage cutoff. The value of *k*-mer which is too high or too low both caused the decrease of reads usage for dataset spombe. Also lower coverage cutoff may lead to more reads being employed. Compared to the excellent mapping results (Table S3), *de novo* assemblers lose advantage in efficiently using sequencing reads than genome-guided assemblers which identify transcripts based on spliced alignment. The insufficient usage of reads by de novo assemblers, especially oases, offers room to further postprocess unassembled reads.

(iv) Comparison with reference annotation. Table 2 summarizes the comparison of transfrags from each assembler with annotated transcript model. In total, *de novo* assemblers detected far more transcripts and loci than genomeguided assemblers when compared with the reference annotation files, particularly oases and trans-ABySS. Thus *de novo* assemblers generally achieved more transcripts per locus, which is in line with their high variant resolution shown in Table 1. The exceptionally large numbers of transcripts per locus for oases and trans-ABySS on dataset spombe are perhaps owing to multiple similar transcripts at individual locus from allelic variation or undetected sequencing errors, which is hard to discriminate without a reference genome. However, trinity discovered relatively more loci and thus much less transcripts per locus,

Table 2 Results of comparing transfrags output by five assemblers with reference annotations on three datasets[a)]

	Number of mRNA	Number of loci	Number of transcripts per locus	Base sn (%)	Base sp (%)	Number of full length (=) transfrags	Number of partial (c) transfrags	Number of novel (j) transfrags
human	166625	49201						
oases	701113	357179	~2.0	31.4	40.3	4437	171274	13625
trinity	406653	310028	~1.3	35.5	45.7	5564	106694	18718
trans-ABySS	618800	248551	~2.5	28.4	45.5	3503	378909	7704
cufflinks (RABT)	233316	107737	~2.2	99.3	78.1	158186	232	8646
cufflinks	115018	105663	~1.1	37.3	58.4	5322	32083	11449
cuffmerge	287218	237105	~1.2	41.1	41.3	7341	41951	44498
inchworm	604933	421837	~1.4	25.2	34.4	2793	182727	279653
scripture	112758	74385	~1.5	25.9	56.2	6463	28175	33815
mouse	93525	36292						
oases	283997	77942	~3.6	24.7	39.1	3927	47997	14277
trinity	318049	253777	~1.3	25.2	25.2	4720	25124	9350
trans-ABySS	270505	94620	~2.9	28.8	38.1	4274	85231	6246
cufflinks (RABT)	111177	48292	~2.3	99.5	82.0	89703	187	6567
cufflinks	41146	31895	~1.3	43.9	59.8	8247	5508	9831
scripture	73149	33347	~2.2	35.7	64.5	8292	10467	31405
cuffmerge	157565	94120	~1.7	48.6	32.8	8662	7566	45044
inchworm	1107272	442826	~2.5	24.1	21.6	2618	338492	2545
spombe	6907	6210						
oases	33220	4105	~8.1	41.5	50.5	3467	5842	1693
trinity	24078	14645	~1.6	92.7	76.2	4485	3969	1481
trans-ABySS	32748	5913	~5.5	64.6	55.1	1956	8836	1026
cufflinks (RABT)	7528	6076	~1.2	98.9	96.9	6866	119	235
cufflinks	3779	3502	~1.1	70.3	83.0	1572	174	839
scripture	3844	2759	~1.4	44.9	60.3	743	136	1493
cuffmerge	6405	4572	~1.4	90.6	56.7	913	97	2363
inchworm	372728	12852	~29.0	59.4	73.6	1781	348584	382

a) Full length transfrag refers to a transfrag with complete match of intron chain as a reference transcript, corresponding to classcode =; partial transfrag refers to a transfrag contained in a reference transcript, corresponding to classcode c; novel transfrag refers to potentially novel isoform sharing at least one splice junction with a reference transcript, corresponding to classcode j.

even less than scripture for dataset brain and esc. Trinity also yielded surprisingly large number of transcripts and loci for dataset esc, possibly a result of lacking strand-specific information. Similarly, scripture detected more transcripts and loci than cufflinks for dataset esc. Additionally scripture always found more transcripts per locus than cufflinks across all three datasets, which is also in concordance with variant resolution metric. But the power of RABT helped cufflinks find more transcripts per locus.

The performance of cufflinks (RABT) also stood out regarding base sensitivity and specificity. In general, genome-guided assemblers had higher specificity, which is probably linked to the fact that *de novo* assemblers generated too many transfrags beyond reference annotation. However, the sensitivity of assemblers varies a lot, and it appears that each assembler performed better on certain datasets. For instance, cufflinks had highest sensitivity for brain and esc dataset, but trinity had particularly higher sensitivity than cufflinks for dataset spombe. This can be partly explained by special measures trinity taken to deal with gene-dense genomes. Moreover, all *de novo* assemblers had higher sensitivity than scripture for dataset brain, but less sensitivity than genome-guided assemblers on dataset esc. This may be owing to that esc dataset is not strand-specific, causing problems to distinguish orientation of transcripts for *de novo* assemblers. Probably due to the same reason, scripture instead achieved higher sensitivity for dataset esc while performing worse on the other two datasets. Generally trans-ABySS performed well on all datasets, while oases had the lowest sensitivity on dataset esc and spombe.

Overall, genome-guided assemblers found more full-length transfrags than *de novo* assemblers, with the exception of spombe dataset. One possible reason is that spombe dataset has pretty high sequence coverage. As expected, cufflinks (RABT) produced much more full-length transfrags and less partial transfrags than other assemblers. Scripture obtained more full-length transfrags than cufflinks for dataset brain and esc, but less than half of the number of full-length transfrags detected by cufflinks for spombe datastet, suggesting problems in dealing with complexity of *S. pombe* genome. However, the number of known transcripts covered entirely by a single transfrag from scripture (brain, 2435; esc, 3809; spombe, 1574) was much smaller than that of full-length transfrags from scripture reported by cuffcompare. This reveals that many transfrags from scripture do not exactly overlap with annotated transcripts on initial and terminal exons, although they have identical intron chains (or complete exon-intron borders) as known transcripts. Among the three *de novo* assemblers, trinity identified most full-length transfrags and least partial transfrags, whereas trans-ABySS recovered most partial transfrags, partially because ABySS tends to report fragmented contigs on which trans-ABySS is based on. As for novel transfrags, scripture recovered most except on dataset spombe, while cufflinks and cufflinks (RABT) detected much less. In short, among *de novo* assemblers, oases identified more novel transfrags and trans-ABySS found least.

2.3 Relationships between transcripts reconstruction and expression level

Figure 4 shows the correlation between transcriptome reconstruction efficiency by each assembler and transcript expression level clearly. Without exception, cufflinks (RABT) reconstructed almost all the transcripts completely across all FPKM quantiles on three datasets, while other assemblers only reached this performance on ultra-deep sequencing dataset spombe at higher expression quantiles. Except for cufflinks (RABT), the median coverage of annotated transcripts by assembled transfrags output by all assemblers rises when transcript expression level increases, suggesting that these assemblers reconstructed more and more complete transcripts. However, these assemblers vary in the extent of rising. For example, scripture performed worst at low expression level on dataset brain, but was only worse than cufflinks when reaching high levels. Besides, trans-ABySS had slight disadvantages over scripture at low quantiles for dataset esc, but caught up with scripture at higher expression quantiles. Nevertheless, for dataset spombe these assemblers quickly acquired similar results at higher levels, although they had large differences at very low expression levels, especially scripture which showed much lower median coverage of transcripts than other assemblers.

Generally speaking, *de novo* assemblers are more likely to be limited by low read depth than genome-guided assemblers, lacking enough information to reconstruct full length transcripts. But when sequencing coverage goes up higher, *de novo* assemblers can perform quite well, even better than genome-guided assemblers [9,15]. Hence it is not surprising to see that *de novo* assemblers had comparable performances with genome-guided assemblers throughout the spectrum of expression quantiles on all three datasets. Particularly trinity surpassed cufflinks except at higher expression levels on dataset brain. Oases had a similar performance as trinity at higher quantiles, but it performed poorly at lower levels

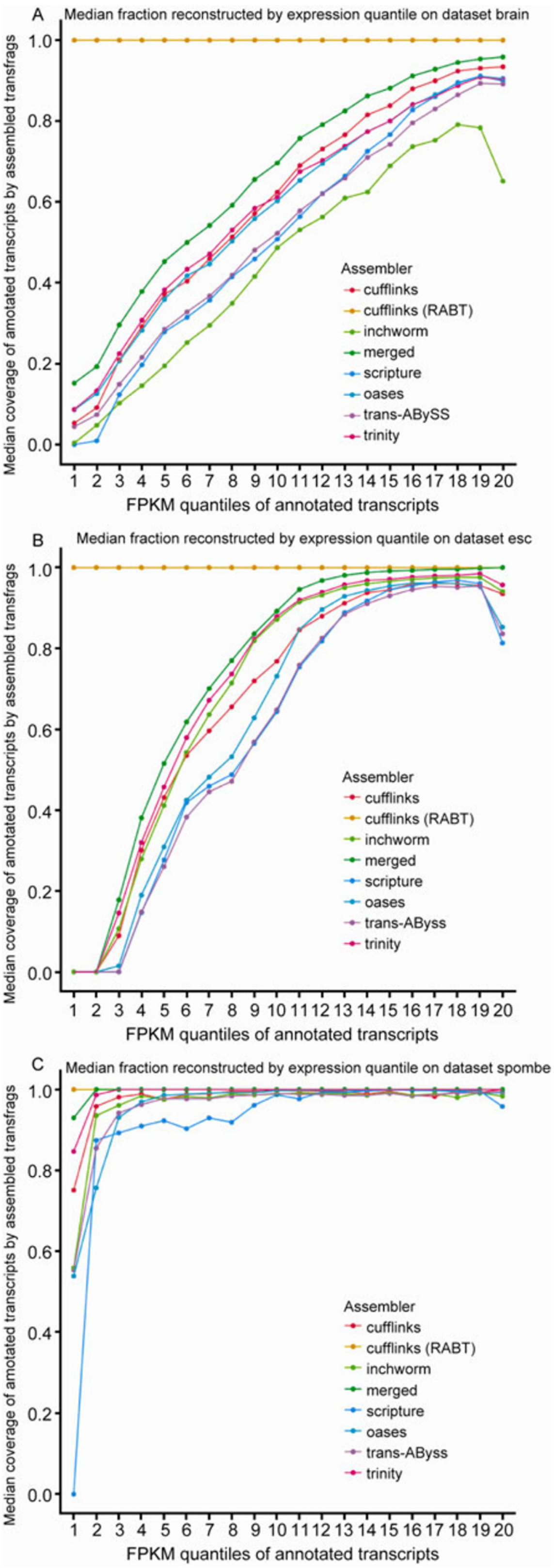

Figure 4 Correlation between transcript reconstruction efficiency and transcript expression level on each dataset. Point in each curve represents median coverage of annotated transcripts by a single assembled transfrag from corresponding assembler, according to expression quantiles (in 5% increments) in dataset brain (A), esc (B) and spombe (C). 'Merged' represents result of cuffmerge.

and had a sharper drop at the highest quantile. Trans-ABySS appears to perform worst overall among *de novo* assemblers, but it still exceeded scripture sometimes. It is also intriguing to note that most assemblers had a more or less decrease at the highest expression level, typically on dataset esc, which is probably caused by increasing sequencing errors in highly abundant reads [15]. Among them, scripture always had the biggest drop.

2.4 Overlap between reconstructed transfrags

It has been shown that each assembler can detect specific transcripts not discovered by others [15]. According to the overlap between full length annotated transcripts and novel transcripts (Figure S2) respectively, all the five assemblers indeed reconstructed considerable amount of unique full length and novel transcripts while sharing substantial number of common ones (full length: brain, 757/11836; esc, 422/13531; spombe, 85/4886; novel: brain, 2088/10888; esc, 668/10073; spombe, 57/1635), but they still detected significant amount of unique transcripts.

It seems that each assembler found more unique full length transcripts and novel transcripts on some datasets and less on another. For example, trinity found most unique full-length transcripts on spombe dataset, but much less on dataset brain and esc. Scripture detected most unique full-length transcripts on dataset brain and esc but least on dataset spombe. Oases found more unique novel transcripts than other *de novo* assemblers for dataset esc, but least on dataset spombe. Cufflinks found most unique novel transcripts on dataset brain and esc, but much less on dataset spombe. Conversely trans-ABySS found least unique novel transcripts on dataset brain and esc, but much more on dataset spombe. This suggests the necessity of integrating different assemblers together to have a more comprehensive assembly.

3 Discussion

Based on the results of comparing the five assemblers from several metrics on three different Illumina RNA-Seq datasets, we can easily observe that each transcriptome assembler has its own strengths which are not implemented by others, although cufflinks and trinity seem to have steadily better performance across all the three datasets. Cufflinks (RABT) had nearly perfect results via utilizing the reference annotation. Scripture can reconstruct more long transfrags and transfrags with same intron chains as annotated transcripts. *De novo* assemblers can identify more transfrags and isoforms per locus, and detect short transfrags ignored by genome-guided assemblers, among which oases, in particular, can recover much more long transfrags as sequencing coverage increases. Trans-ABySS consumed much less computing resources than other *de novo* assemblers, and still had a pretty good performance, like finding most splice variants per gene. Furthermore, with the increase of sequencing coverage, *de novo* assemblers can work better than genome-guided assemblers in some aspects.

Broadly speaking, both genome-guided assembly and *de novo* assembly have their advantages and drawbacks, performing worse or missing some information in certain aspects while having a good performance in other conditions. Therefore, it is not easy to choose a single best assembly method once and for all. The wise choice of transcriptome assemblers is always dependent on concrete context. Generally *de novo* transcriptome assemblers are much more computationally expensive than genome-guided assemblers, and are used when reference genome is not available. But when the reference genome is accessible, they can also be utilized. Furthermore, it might be better to combine genome-guided assembly and *de novo* assembly to get a more comprehensive transcriptome. This combined approach might take advantage of superiorities of either strategy and complement the shortness of each other [2]. To validate the assumption, we merged assemblies of the five assemblers (excluding cufflinks (RABT)) by cuffmerge, and then evaluated the results with some of previous metrics. The merged assembly significantly improved individual assembly from each assembler, from overall assessment to length statistics, comparison with annotation and sensitivity to expression level (Figures 3 and 4; Tables 1 and 2; Table S5).

In theory, it is also feasible to integrate these different algorithms to get a more ideal assembly. It is said that many parts of scripture can be used in *de novo* assembly [14], and RABT of cufflinks can be used with other assemblers too [12]. This flexibility in the design of assembly algorithms provides possibility to adapt some components of an assembler to fit in different situations. However it requires further efforts to design an efficient hybrid method. Currently as a represent of genome-guided *de novo* assembler, inchworm integrates the two transcriptome reconstruct strategies cleverly, hence we tested inchworm03132011 (http://inchworm.sourceforge.net/) to see its performance in terms of previous metrics. The results (Figures 2 and 4; Tables 1 and 2; Table S5) showed that inchworm had a relatively good performance. Inchworm took much less time than *de novo* assemblers. It also consumed much less memory, at most 1 g, due to partitioning aligned reads to multiple small subsets and doing assembly separately. However, it got more transfrags with shorter length. Its performance in comparison with annotation is also not well, but it detected more isoforms for each gene and had low chimerism. Since it leverages alignment, its accuracy was pretty high, whereas it had low completeness and continuity. Inchworm transfrags had lowest median coverage of known transcripts with respect to expression quantiles on dataset brain, but showed better performance on dataset esc and spombe. In a word, inchworm had a desired performance, despite that it appears to still require optimization. Most importantly, inchworm provides an ingenious way to combine genome-guided assembly and *de novo* assembly.

Actually, there have been two commonly used combing strategies: assemble-then-align and align-then-assemble [4]. Assemble-then-align means using *de novo* assembly at first, followed by extending transfrags through aligning the assembly result along with unassembled reads to the reference genome, whereas align-then-assemble refers to firstly reconstructing transcripts by aligning reads to the reference genome, and subsequently handling the small fraction of unmapped reads by *de novo* assembly. But combination approaches may vary a lot in practice depending on concrete conditions. For instance, cufflinks and velvet have been used together to unveil some missing expressed genes in human genome, via performing genome-guided assembly first and then using *de novo* assembly to explore novel gene outside the reference genome [34]. Whatever ways to combine genome-guided assembly and *de novo* assembly, it seems that integrating multiple assemblers in each category can achieve a better assembly.

In summary, no transcriptome assembler in either genome-guided assembly category or *de novo* assembly category is the best choice for every condition, and each assembler is able to capture unique transcripts not detected by others. Although tuning parameters may get better results for some assemblers, it is likely that integrating assemblers from both categories together can offer a more accurate picture of transcriptome and can further improve genome annotation and other downstream analysis. Actually, all these assemblers are designed flexibly, revealing the possibility of combing some of them together. Furthermore, some combination approaches have been successfully applied in practice to achieve a more complete transcriptome assembly. However, further efforts are needed to explore an optimal pipeline of hybrid assembly using both genome-guided assembly and *de novo* assembly. Nevertheless, adopting an integrative approach to assemble transcriptome seems to bring many benefits for further study of transciptome.

We are very grateful to Chen Geng for insightful suggestions. We thank Zhao Chen for tackling annoying problems with the server and useful advices, Shi CaiPing for helpful discussion at the beginning of the research, Xue RuiChao and Shen ZhongShao for kind help on some scripts. This work was supported by the National Basic Research Program of China (2010CB945401), the National Natural Science Foundation of China (31240038, 31171264, 31071162, 31000590) and the Science and Technology Commission of Shanghai Municipality (11DZ2260300).

1 Wang Z, Gerstein M, Snyder M. RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet, 2009, 10: 57–63

2 Garber M, Grabherr M G, Guttman M, et al. Computational methods for transcriptome annotation and quantification using RNA-seq. Nat Methods, 2011, 8: 469–477

3 Martin J, Bruno V M, Fang Z, et al. Rnnotator: an automated *de novo* transcriptome assembly pipeline from stranded RNA-Seq reads. BMC Genomics, 2010, 11: 663

4 Martin J A, Wang Z. Next-generation transcriptome assembly. Nat Rev Genet, 2011, 12: 671–682

5 Trapnell C, Roberts A, Goff L, et al. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat Protoc, 2012, 7: 562–578

6 Birol I, Jackman S D, Nielsen C B, et al. *De novo* transcriptome assembly with ABySS. Bioinformatics, 2009, 25: 2872–2877

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

8 Robertson G, Schein J, Chiu R, et al. *De novo* assembly and analysis of RNA-seq data. Nat Methods, 2010, 7: 909–912

9 Grabherr M G, Haas B J, Yassour M, et al. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol, 2011, 29: 644–652

10 Haas B J, Zody M C. Advancing RNA-Seq analysis. Nat Biotechnol, 2010, 28: 421–423

11 Chen G, Wang C, Shi T. Overview of available methods for diverse RNA-Seq data analyses. Sci China Life Sci, 2011, 54: 1121–1128

12 Roberts A, Pimentel H, Trapnell C, et al. Identification of novel transcripts in annotated genomes using RNA-Seq. Bioinformatics, 2011, 27: 2325–2329

13 Trapnell C, Williams B A, Pertea G, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol, 2010, 28: 511–515

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

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

16 Lander E S, Waterman M S. Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics, 1988, 2: 231–239

17 Simpson J T, Wong K, Jackman S D, et al. ABySS: a parallel assembler for short read sequence data. Genome Res, 2009, 19: 1117–1123

18 Langmead B, Trapnell C, Pop M, et al. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol, 2009, 10: R25

19 Li H, Handsaker B, Wysoker A, et al. The sequence alignment/map format and SAMtools. Bioinformatics, 2009, 25: 2078–2079

20 Trapnell C, Pachter L, Salzberg S L. TopHat: discovering splice junctions with RNA-Seq. Bioinformatics, 2009, 25: 1105–1111

21 Kent W J. BLAT—the BLAST-like alignment tool. Genome Res, 2002, 12: 656–664

22 Quinlan A R, Hall I M. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics, 2010, 26: 841–842

23 Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, 2006, 22: 1658–1659

24 DeLuca D S, Levin J Z, Sivachenko A, et al. RNA-SeQC: RNA-seq metrics for quality control and process optimization. Bioinformatics, 2012, 28: 1530–1532

25 Stanke M, Tzvetkova A, Morgenstern B. AUGUSTUS at EGASP: using EST, protein and genomic alignments for improved gene prediction in the human genome. Genome Biol, 2006, 7(Suppl1): S11. 1–8

26 Medvedev P, Georgiou K, Myers G, et al. Computability of Models for Sequence Assembly, in Algorithms in Bioinformatics. Berlin Heidelberg: Springer, 2007. 289–301

27 Nagarajan N, Pop M. Parametric complexity of sequence assembly: theory and applications to next generation sequencing. J Comput Biol, 2009, 16: 897–908

28 Miller J R, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics, 2010, 95: 315–327

29 Lee C. Generating consensus sequences from partial order multiple sequence alignment graphs. Bioinformatics, 2003, 19: 999–1008

30 Schulz M H. Data structures and algorithms for analysis of alternative splicing with RNA-seq data. Dissertation for doctoral degree. Berlin: Free University of Berlin, 2010

31 Xing Y, Resch A, Lee C. The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures. Genome Res, 2004, 14: 426–441

32 Trapnell B C. Transcript assembly and abundance estimation with high-throughput RNA sequencing. Dissertation for doctoral degree. College Park: University of Maryland, 2010

33 Iyer M K, Chinnaiyan A M. RNA-Seq unleashed. Nat Biotechnol, 2011, 29: 599–600

34 Chen G, Li R Y, Shi L M, et al. Revealing the missing expressed genes beyond the human reference genome by RNA-Seq. BMC Genomics, 2011, 12: 590

35 Zhao Q Y, Wang Y, Kong Y M, et al. Optimizing *de novo* transcriptome assembly from short-read RNA-Seq data: a comparative study. BMC Bioinformatics, 2011, 12: S2

36 Henschel R, Lieber M, Wu L S, et al. Trinity RNA-Seq assembler performance optimization. In: Proceedings of the 1st Conference of the Extreme Science and Engineering Discovery Environment: Bridging from the eXtreme to the Campus and Beyond, Chicago, Illinois, USA, 2012. 1–8

Supporting Information

Figure S1 Read usage for three *de novo* transcriptome assemblers on three datasets.

Figure S2 Overlap between reconstucted full-length annotated transcripts and novel transcripts on three datasets. S1 refers to trinity, S2 refers to oases, S3 refers to trans-ABySS, S4 refers to cufflinks, S5 refers to scripture. A and B, On brain; C and D, On esc; E and F, On spombe.

Table S1 Details of three datasets

Table S2 Critical parameters of some assemblers on three datasets

Table S3 Mapping statistics of spliced alignment on three datasets

Table S4 Similarities and differences of algorithm features for five assemblers

Table S5 Number and length statistics of transfrags output by transcriptome assemblers on three datasets

Table S6 Read usage for velvet-oases on dataset spombe and esc

The supporting information is available online at life.scichina.com and www.springerlink.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.