Quantitative Biology 2017, 5(4): 280–290
DOI 10.1007/s40484-017-0109-2

REVIEW

Transcriptome assembly strategies for precision medicine

Lu Wang[1], Lipi Acharya[2], Changxin Bai[1] and Dongxiao Zhu[1,*]

[1] Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
[2] Dow AgroSciences, Indianapolis, IN 46268, USA
* Correspondence: dzhu@wayne.edu

Received March 17, 2017; Revised April 17, 2017; Accepted April 19, 2017

***Background*: Precision medicine approach holds great promise to tailored diagnosis, treatment and prevention. Individuals can be vastly different in their genomic information and genetic mechanisms hence having unique transcriptomic signatures. The development of precision medicine has demanded moving beyond DNA sequencing (DNA-Seq) to much more pointed RNA-sequencing (RNA-Seq) [*Cell*, 2017, 168: 584–599].**
***Results*: Here we conduct a brief survey on the recent methodology development of transcriptome assembly approach using RNA-Seq.**
***Conclusions*: Since transcriptomes in human disease are highly complex, dynamic and diverse, transcriptome assembly is playing an increasingly important role in precision medicine research to dissect the molecular mechanisms of the human diseases.**

Keywords: precision medicine; transcriptome assembly; RNA-Seq; *de novo*; *De Bruijn*

INTRODUCTION

Precision medicine is an emerging area in healthcare that aims to provide personalized diagnosis, treatment, and prevention by taking an individual's genetic information, environment, and lifestyle into account [1]. The genetic information and genomic mechanisms are vastly different among individuals, which cause considerable variations in gene expression. The latter can be quantified by utilizing next-generation RNA sequencing methodologies such as RNA-sequencing (RNA-Seq). RNA-Seq can provide valuable information that may lead to improved clinical diagnosis, prognosis and treatment plan [2].

In precision medicine, personalized prevention and treatment are considered independently [3]. An efficient way to implement personalized prevention is to obtain the genetic structure and functionality along with the variation of gene expression captured by RNA-Seq. For example, personalized prevention is becoming possible by analyzing the transcriptomic data in individual oncology patient to determine the unique transcriptomic signatures of certain cancers [4]. With these signatures, potential patients who are yet to be diagnosed for cancer can be identified and personalized prevention can be designed for them ahead of time.

Personalized treatments with tailored therapies have been developed to treat single genetic alterations in malignant tumors [4]. In precision medicine, treatments are designed for specific targets and are altered pathologically with cellular signaling components requiring an insight into the cellular mechanisms [4]. Transcriptome assembly approaches enable an in-depth understanding of the cellular mechanisms and cellular processes [5] and thus plays a key role in both personalized prevention and treatment. The development of deep DNA-Seq and RNA-Seq technologies provide an unprecedented opportunity to consider and incorporate individual variability for optimizing personalized prevention and treatment of human diseases [6].

Initial precision medicine research using deep sequencing technology more relate to using DNA-Seq to dissect individual genetic variation for each patient sample. These variations range from small-scale single nucleotide polymorphisms (SNP's) to large-scale genome aberrations. The exploitative increasing of The Cancer Genome Atlas (TCGA) data has stimulated extensive research in

extracting and exploiting genomic features for patient stratification, diagnosis and prognosis [7,8]. More recent research has seen a critical need to move beyond DNA-Seq to much more pointed RNA-Seq technology [9]. Different from genome, transcriptome is more diverse, dynamic and complex. The alternative splicing and gene editing mechanisms further compromise the performance of genome alignment approach employed in DNA-Seq [10]. Thus, the choice of a more effective approach to dissect transcriptome has been nailing down to transcriptome assembly.

In this review article, we survey a number of recently developed transcriptome assembly strategies including *de novo* transcriptome assembly, reference-based transcriptome assembly, and *de novo* and reference-based combined strategies [11]. We also discuss the *De Bruijn* graphs for representing the structure of transcripts, which have been utilized to select a set of transcripts with maximum support from RNA-Seq reads. Further, we describe a generative probability model for transcriptome assembly followed by a few assessments metrics for assembly quality.

TRANSCRIPTOME ASSEMBLY STRATEGIES

Reconstructing full length transcripts from RNA-Seq reads has many challenges. Similar to genome assembly from short reads, transcriptome assembly needs to piece short reads together. Some softwares, such as Velvet [12], ABySS [13] and Euler [14], have been developed to tackle the *de novo* genome assembly problems. However, these tools cannot be used for transcriptome assembly directly due to several considerations as follows.

The sequencing depth of transcripts can vary dramatically from different levels of gene expression while the DNA sequencing depth is expected to be the same in all genes [10,15]. For a specific gene, the transcript variants may share exons that make it very hard to solve the ambiguous problems [11]. If the reference genome is provided, some approaches start by clustering overlapping reads from each gene where a graph needs to be reconstructed.

On the other hand, if the reference genome is not available, *De Bruijn* graphs are used to represent a set of transcripts. Ideally, each *De Bruijn* graph may correspond to an expressed gene. Isoforms can be achieved by traversing each *De Bruijn* graph, in which some isoforms are false positive transcripts. Once the graph representing the structure of transcriptome in an organism is constructed, the most important step is to select a set of transcripts that are most likely represented by a data set of reads. In the following sections, we describe the transcriptome assembly strategies using different models and select methods in order to extract a set of transcripts.

De novo transcriptome assembly

Many organisms do not have the reference genome due to various reasons such as biological diversity as well as time and cost factors. The *de novo* assembly methods are useful when there is no reference genome available or when there are some alterations, mutations and genome rearrangements within the reference genome. Without a reference genome, the *de novo* transcriptome assembly strategy for next-generation sequencing has been developed to assemble transcripts directly using the reads [16]. *De novo* assemblers take advantage of the redundancy of reads and merge them into complete transcripts using the sequence overlap-based *De Bruijn* graph instead of mapping reads onto the reference genome. Figure 1 presents an overview of the *de novo* transcriptome assembly strategy.

Basic *de novo*

A *De Bruijn* graph [17] does not require storing all the reads and overlapping information in the memory. Within a graph, a node is expressed by a sequence of special k nucleotides or k-mers. The nodes are connected by edges with $k-1$ overlapping information. Ideally, thousands of *De Bruijn* graphs are constructed from all the reads. Each path in the *De Bruijn* graph represents a possible transcript. Usually, for each raw read of length l, a sliding window is used to break the read into $(l-k+1)$ overlapping k-mers. The graphs construction process uses a constant-time hash table to search each k-mer, where each k-mer can be stored at once.

Erroneous data may cause an inappropriate branch in the process of graph construction. Error removal process plays a vital role in transcriptome assembly, which may generate putative transcripts. The uniform coverage is used to remove errors produced in the sequencing process or SNP's. Each next generation sequencing platform has one error model to indicate the probability of erroneous nucleotides produced by the sequencer.

Three main error structures are: (i) tips caused by the errors at the edges of reads, (ii) bubbles caused by the internal read errors and (iii) erroneous connections proposed in Velvet [12]. Tips can be recognized based on two criteria: the length and the number of branches. Bubbles can be removed using tour bus algorithm in the *De Bruijn* graph. Erroneous connections can be removed using a basic coverage cut-off. However, these errors removing strategies are typically used in the genome assembly. Some subtle modifications need to be made to utilize the error structures in the transcriptome assembly.

Transcriptome assembly problem has its own computa-

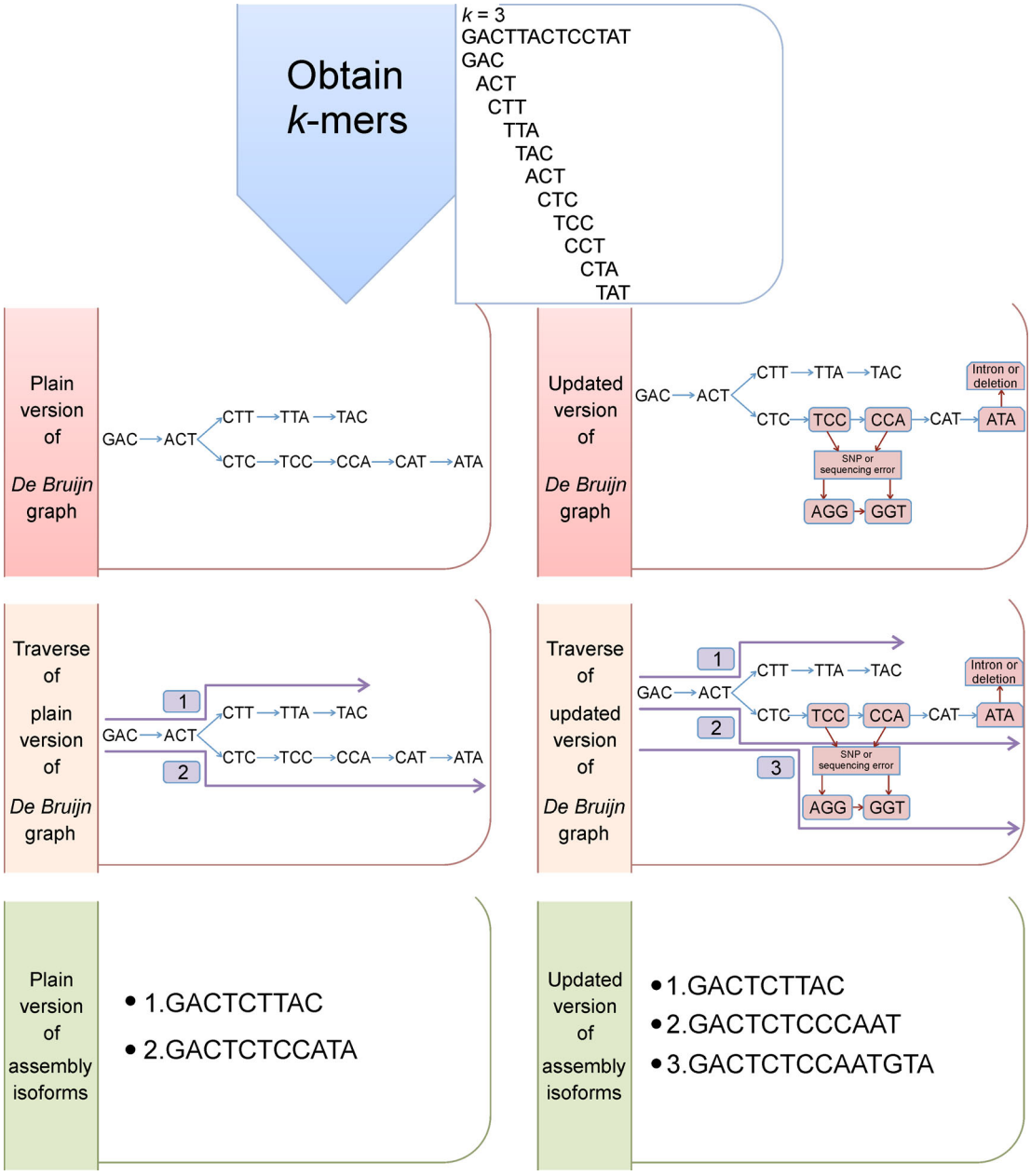

Figure 1. ***De novo*** **transcriptome assembly strategy.** Given a read, we generate all length *k* substrings (*k*-mers), and with all *k*-mers, we generate two versions of *De Bruijn* graphs, which are plain and updated versions without or with intron, deletion, single-nucleotide polymorphism (SNP) or sequencing error. Each node in the *De Bruijn* graphs is expressed by a sequence of special *k* nucleotides or *k*-mers, where *k* = 3 in this example. Traversing the two versions of *De Bruijn* graphs, we finally obtain the assembly isoforms.

tional challenges different from the genome assembly. For example, the number of erroneous reads derived from a highly-expressed transcript may be more than the number of right ones derived from a low expressed transcript which makes the error removal process difficult. An efficient way to represent the *De Bruijn* graph is to allow sequences to distribute over a cluster as proposed in Ref. [13].

Two issues need to be considered to distribute *De Bruijn* graph over a cluster: storing the *k*-mers information and location of a special *k*-mer. The forward and reverse complement of each *k*-mer is represented by a number of values {0, 1, 2, 3} assigning to bases {A, C, G, T}. The adjacency information for each *k*-mer is stored in 8 bits, which represent every possible base extension and their directions. Some branches caused by sequencing

errors need to be removed before merging vertices into contigs based on a threshold length. The pair-end information can be used to make sure that contigs can be linked together.

De novo extension: single *k*-mer assemblers

Trinity [16] is the first software developed for *de novo* transcriptome assembly using RNA-Seq data with single *k* value to recover full-length isoforms efficiently in mouse, yeast and butterfly. Trinity constructs *De Bruijn* graph from large size RNA-Seq data and divides them into a number of *De Bruijn* graphs, where each graph represents a set of transcripts of a single gene. Trinity was specially designed for removing erroneous edges to ensure the optimized support for the reads.

Three main steps are included to process large amounts of RNA-Seq reads. In the first step, a greedy algorithm is used to obtain contigs. In the second step, contigs are used to build individual *De Bruijn* graphs. In the last step, some trim edges are removed and some paths are compacted to output one linear sequence representing each transcript.

At the first stage of assembly, a dictionary of *k*-mers is constructed from all the reads associated with their frequencies followed by removing the *k*-mers with frequency less than 5% compared to the highest frequency ones in the dictionary. Then a seed *k*-mer is selected as the cornerstone sequence to assemble a contig. It extends the sequence in the dictionary until no more *k*-mers exist. At the second stage, the contigs obtained at the first stage are clustered based on $k-1$ overlapping bases or the number of reads spanning the junction across contigs. Each component is represented using a *De Bruijn* graph. At the last stage, merging consecutive nodes in linear path is implemented for the graph simplification. Some edges, likely sequencing error, are pruned in the final graph. Dynamic programming is used to traverse these paths in the graph.

Oases is specially proposed for transcriptome assembly problem and developed from the Velvet [18]. The contigs construction in Oases follows the same process used in Velvet and differs in the contig correction process. Besides a slightly modified error correction algorithm, Oases uses a local edge removal strategy, which removes an outgoing edge with coverage representing less than 10% of the total coverage of all outgoing edges. Finally, a coverage cut-off is used to filter contigs in the assembly process. Once the contigs are constructed, scaffold is constructed from these contigs to estimate the distance between contigs. These contigs are labeled using the length of each contig. Two short contigs can only be concatenated without gap by a direct connection. One short contig and long contig can only be connected by a direct graph.

Similar to the contigs correction stage, some filter technologies are applied to the scaffolds based on the static and dynamic coverage thresholds. In this way, some contigs supported by reads with a lower threshold are deleted. Long contigs are first clustered into connected components called *loic* and short contigs are added to one of the long contigs in the cluster, where long distance connections are removed. Transcripts are extracted in the *loic* with the following four categories: chains, bubbles, forks, and complex *loic* using a dynamic programming search algorithm. The preliminary contigs obtained from different *k* value in the first stage are fed into the remaining part of the transcriptome assembly pipeline. The final assembly results are based on the combination of contigs with different *k* values capturing different expression levels.

SOAPdenevo-Trans [19] incorporates the following two strategies to improve the performance of transcriptome assembly: the robust heuristic graph-traversal method using Oases and the error-removal model using Trinity. SOAPdenevo-Trans also uses a method to simplify the scaffolding graphs to improve more accurate results. The sequencing errors are removed in which a weak depth cutoff is applied followed by the Trinity error-removal method to handle the remaining errors.

By utilizing both single-end and paired-end reads information, contigs are generated with linkages. The insert sizes from the paired-end reads are used to estimate the distance between linkages and the short contigs ($\leqslant$100 bp) are removed. This is followed by linearizing the contigs to scaffold with the strategy that, for example, three contigs can be linearized if any two of them are linked and the linkages do not conflict with each other.

Contigs are clustered into sub-graphs according to their linkages. Each sub-graph consists of a set of transcripts. The sub-graph can be fragmented due to alternative splicing events, which makes the sub-graph have a branch. A traversing graph method used in Oases is applied to the sub-graph to generate possible transcripts from linear or fork path.

In many cases, the sub-graph can be uniquely converted into one transcript. Three different kinds of transcripts are defined as linear, fork, and bubble. These three kinds of transcripts are easily identifiable using the degrees of nodes. Each path is associated with a score. For the most complex path, only the top scoring transcripts are retained.

The majority of the gaps inside the scaffolds are composed of repeats that are masked during the scaffold construction. To disassemble the repeats and fill in the gaps, paired-end information is used to retrieve the pair of reads where one read is well-aligned on the contigs and another read is located in the gap region followed by a local assembly for the collected reads.

Long reads will help optimize the contig size and long insert-sized libraries are essential for clustering and construction of long scaffolds. In theory, without long insert-sized libraries, repeats extending beyond the paired-end insert sizes are unable to be resolved and assembled. For the gap closure, considered as the last step in assembly, sufficient sequencing depth of each insert-sized library is correlative to the effectiveness of filling the corresponding sized gaps.

SOAPdenovo [20] and Trans-ABySS were reported to be successfully applied to the transcriptome assembly in Refs. [21] and [22]. For the methods used in Velvet and ABySS, the special parameter *k* needs to be pre-defined.

How to choose a optimized *k* value is still a challenge in the single *k*-mer assemblers. The *k* value impacts the performance of transcriptome assembly in several ways, such as the gene length and the coverage of each gene. Generally speaking, the bigger *k* value performs better on transcripts with the high-expression levels while smaller *k* value performs better on transcripts with low-expression levels [23]. Multiple-*k* strategy has been applied in Trans-ABySS [24] and Oases [18] to assemble transcripts to handle both high and low expression levels with different *k* values. Multiple-*k* strategy performs better with the variable transcripts expression and multiple transcript isoforms.

Reference-based transcriptome assembly strategy

As the reference genome is available for model organisms, transcripts can be inferred from the accumulation of read alignment to the genome, which is referred to as the reference-based transcriptome assembly strategy.

Overlap graph is used in the reference-based transcriptome assembly based on the pairwise alignment information between reads. In the graph, nodes represent reads and an edge between two nodes indicates the overlapping sequence between two reads. By aligning reads to the reference genome, an order of reads along transcripts information can be used to construct the overlap graph per locus. Reads contained within other reads are needed to be ignored. The overlap graph should be treated as a directed acyclic graph (DAG) that can be traversed by the deep first search (DFS) algorithm to extract all paths that can express all the reads where each path represents a transcript. Figure 2 provides a simple example to show the idea of the reference-based transcriptome assembly strategy.

Cufflinks [25] uses the overlap graphs to indicate putative transcripts from the Tophat [26] alignment results. The mechanism of extracting potential transcripts utilizes partitioning algorithm to select the minimum number of transcripts which can express all the reads data in order to reduce the false positive rate.

If the genome annotation file is provided, exons and introns information can be queried for each gene [27]. A splicing graph is a DAG, where nodes represent exons and an edge represents intron connection of two adjacent exons. Transcripts can be enumerated by the combination of different exons. Since the RNA-Seq read is too short to cover a whole exon, a splicing read connecting two exons is used to determine the edge. Once the splicing graph is constructed, the next step is to select a subset of likely transcripts. Some methods including greedy methods and Expectation Maximization are used in the transcripts selection.

De novo and reference-based combined strategy

So far, most existing *de novo* assemblers use the *De Bruijn* graph to represent the structure of transcripts which is traversed by some optimized algorithm. Single *k*-mer method is used in the transcriptome assembly, which is extended from the genome assembly. The latter performs better on uniform transcript expression data. For most eukaryotes, the expression levels of transcripts change dramatically from several fold changes to thousands of fold changes, which makes multiple *k* value strategy succeed in those organisms. If the exaction of reference genome is provided, reference-based assembly method should have some criteria to instruct the assembly process.

A method called *Bridge* which combines the *de novo* assembler Trinity with the reference-based assembler Cufflinks has been proposed [28] to tackle the transcriptome assembly problem. Bridge builds splicing graphs for all genes from the reference genome. For a special gene, exons are represented by nodes and the splicing junctions are represented by edges. The splicing junctions work mainly to extract each transcript.

A rigorous model, named minimum path cover (MPC), recovers the minimal set of transcripts from the splicing graph. All the reads are partitioned into *k*-mers from which a hash table is constructed by recording each *k*-mer and its corresponding abundance. The contigs are generated from *k*-mers by the same strategy used in Trinity. These contigs are treated as the trunk of a splicing graph. Each *k*-mer needs to be checked in the trunk to see if there exists a *k*-mer with an alternative extension that has not been used. In this way, contigs can be extended as long as possible.

The contigs can be added into the trunk if the following three criteria are met, i.e., the length should be larger than 80, the contig is not the same as the one in the trunk, and at least two reads support the contig. The splicing graph is grown by finding bifurcation *k*-mers and repeating until no bifurcation *k*-mer exist. In a splicing graph, nodes and edges represent exons and splicing junctions, respectively

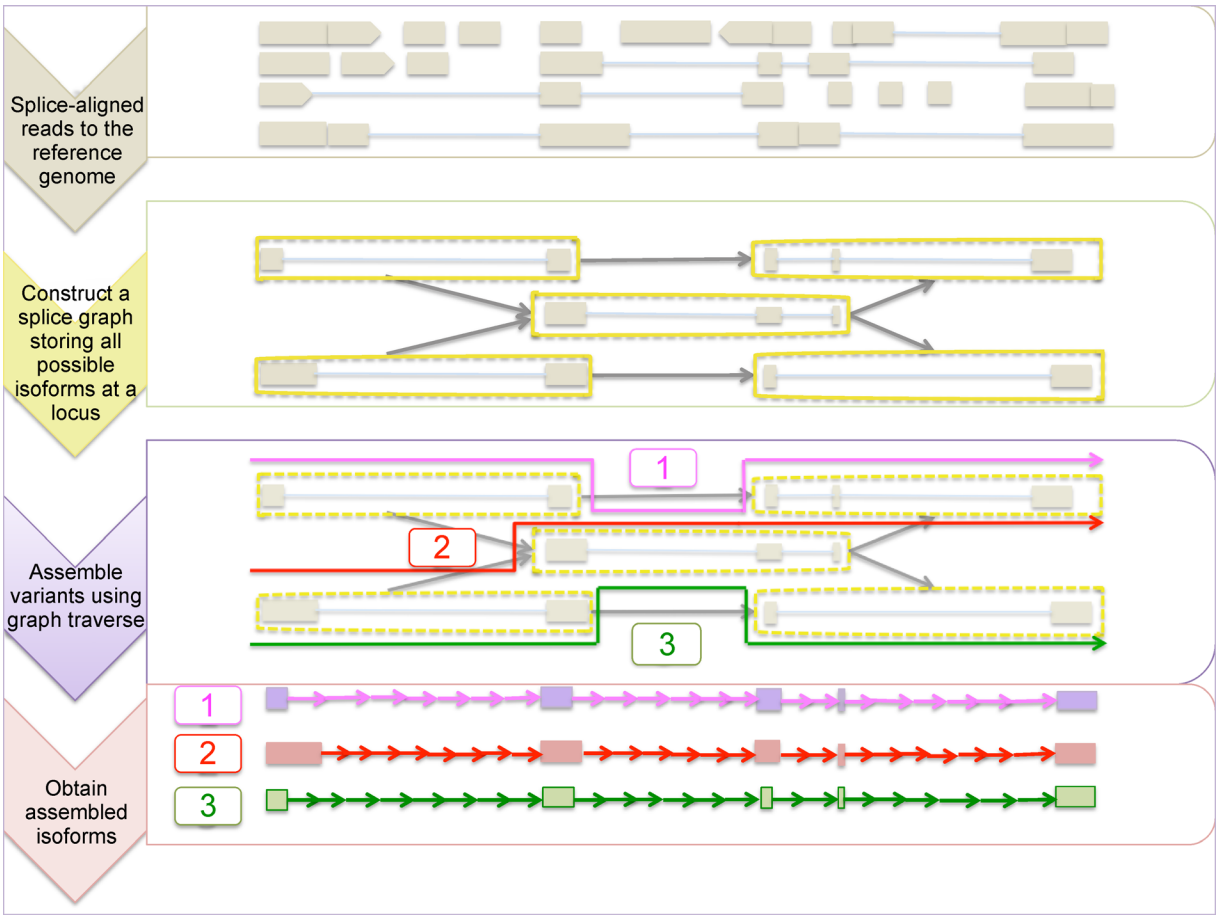

Figure 2. Reference-based transcriptome assembly strategy. We first splice-align reads to the reference genome in order to construct a splice graph to show all possible isoforms at a locus. Then we traverse the constructed graph to obtain the isoforms at the end.

(Figure 3A). One transcript is one path in the splicing graph, but not every path in the splicing graph is necessarily one real spliced transcript. The problem can be treated as that we want to obtain a set of paths that could cover all edges. Thus, we first construct an auxiliary graph (Figure 3B), and then apply MPC model to the new graph. Two consecutive edges in splicing graph are compatible if they could originate from a same spliced isoform. Based on this, a directed graph M, called compatibility graph (Figure 3C) is constructed as following: each edge of splicing graph is assigned as one node and a directed edge (x, y) is placed between nodes x and y if they are compatible. Finding a minimum path cover is equivalent to finding a maximum anti-chain, i.e., a set of mutually incompatible nodes in graph M. Then a maximum anti-chain can be reduced to find a maximum matching of a certain bipartite graph, called the reachability graph (Figure 3D). Given a min-cost maximum cardinality match M, any node without an incident edge in M is a member of an anti-chain. Each member of this anti-chain could be extended to a path using M, which is further extended if it does not correspond to a full-length transcript.

Generative probability model

Transcript assembly problem can be considered as the problem of extracting the optimized paths form a graph to support all the RNA-Seq reads. RNA-Seq reads data can be treated as drawings from different transcripts underlying some stochastic models. A probabilistic approach [29] has been proposed to assemble transcriptome from a set of RNA sequencing reads. Based on this method, the expressed transcripts levels can be inferred by applying Gibbs sampling to the model. Firstly, the splicing graphs are constructed using the alignment results from TopHat2 [30]. Secondly, in each splicing graph, the candidate transcripts are constructed by continuously traversing until a preset value has been reached. In this process, some edges are discarded because of the lowest coverage on them. At last, a generative model is used on the candidate transcripts to determine the combination of

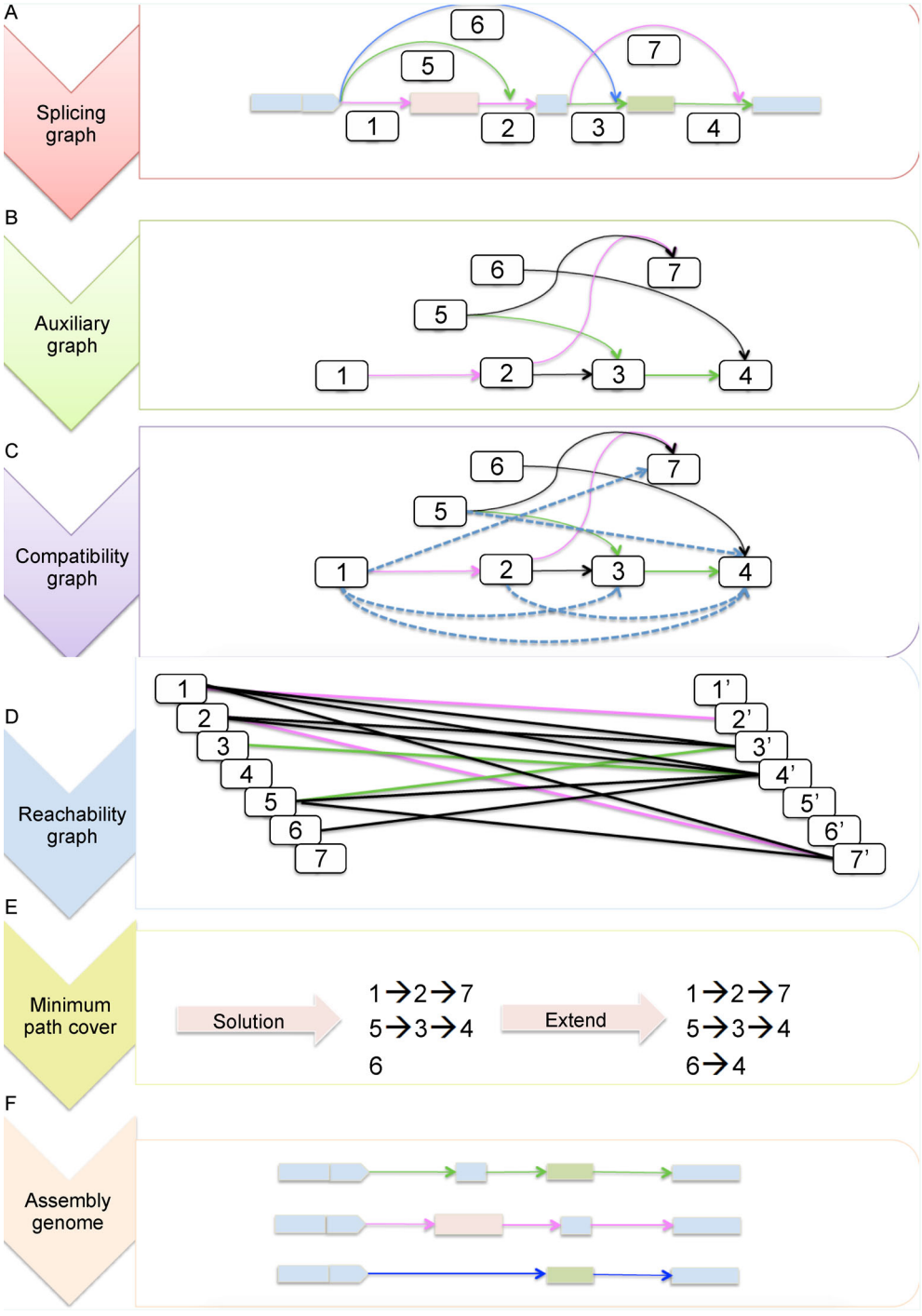

Figure 3. Work flow of the bridge algorithm.

transcripts and their corresponding abundance levels. The corresponding graphic model is showing in Figure 4.

The process of generating a set of reads $\boldsymbol{F}$ from a set of candidate transcripts $\boldsymbol{S}$ is shown as follows: Firstly, the abundance $\boldsymbol{e}$ for transcripts in $\boldsymbol{S}$ are sampled from a prior distribution. Let e_j denote the abundance of the jth candidate transcript:

$$e_j \begin{cases} = 0 \ if \ z_j = 0 \\ > 0 \ if \ z_j = 1 \end{cases}, \tag{1}$$

where z_j is known as indicator variable, which equals to 1 means the abundance of the jth candidate transcript is larger than 0, otherwise not. Thus, we have the sum of indicators denoted as b_z:

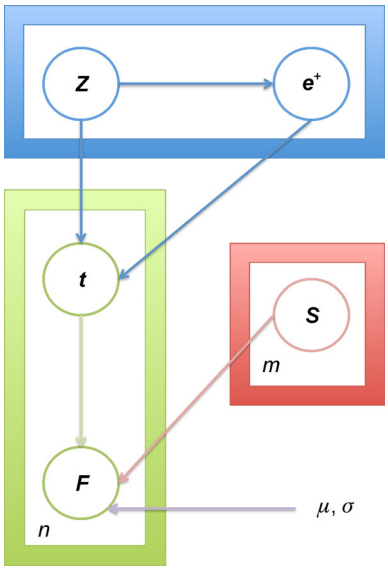

Figure 4. A generative model of the RNA-sequencing process with the notation. $\boldsymbol{Z}$ is used to model the transcripts that have non-zero abundance; $\boldsymbol{e}^+$ is the relative abundance; and $\boldsymbol{F}$ is the joint distribution over n reads conditionally parameterized by μ, σ and dependent on a set $\boldsymbol{S}$ of m candidate transcripts and transcript index $\boldsymbol{t}$.

$$b_z = \sum_{j=1}^{m} z_j, \tag{2}$$

where m is the number of candidate transcripts in $\boldsymbol{S}$.

Then a candidate index $P(t \in \boldsymbol{e})$ is followed by sampling a read from the distribution $P(f|t)$. By assuming the reads are generated independently, the joint distribution over a set of reads $\boldsymbol{F}$ can be written as:

$$P(\boldsymbol{F}, \boldsymbol{t}, \boldsymbol{e}) = P(\boldsymbol{e}) \prod_{i=1}^{n} P(f_i|t_i) P(t_i|\boldsymbol{e}). \tag{3}$$

The candidate transcript abundances need to be predefined by a prior distribution. Let $\boldsymbol{Z}$ models the transcripts are expressed by using a Bernoulli distribution with parameter π such that:

$$P(\boldsymbol{Z}|\pi) = \pi^{b_Z} (1-\pi)^{m-b_Z} K_{Z_0}, \tag{4}$$

where

$$K_{Z_0} = \frac{1}{1-(1-\pi)^m}.$$

$\boldsymbol{e}^+$ represents the relative abundances that are distributed as Dirichlet distribution with density function:

$$P(\boldsymbol{e}^+|Z, \gamma) = \frac{\Gamma(\gamma) b_z}{\Gamma \gamma^{b_z}} \prod_{k=1}^{b_z} (\boldsymbol{e}_k^+)^{\gamma-1}, \tag{5}$$

where Γ is the gamma function.

The relative abundance $\boldsymbol{e}$ can be represented as:

$$P(\boldsymbol{e}|\pi, \gamma) = P(\boldsymbol{e}^+, \boldsymbol{Z}|\pi, \gamma) P(\boldsymbol{Z}|\pi). \tag{6}$$

So, the joint distribution over a set of n reads $\boldsymbol{F}$, transcripts index $\boldsymbol{t}$ and transcript abundances $\boldsymbol{e}$ are conditioned on the candidate transcripts $\boldsymbol{S}$ and some other hyper-parameters as follows:

$$P(\boldsymbol{F}, \boldsymbol{t}, \boldsymbol{e}|\boldsymbol{S}, \mu, \sigma, \pi, \lambda) = P(\boldsymbol{e}|\pi, \lambda \Pi_{i=1}^{n}) P(f_i|t_i, S, \mu, \sigma) P(t_i|\boldsymbol{e}). \tag{7}$$

In order to get the posterior distribution over the expressed transcripts and the corresponding abundances along with the assignments of reads to them, the Gibbs sampling is used to solve the problem. Firstly, expressed candidates are sampled conditionally on the assignment of reads. Secondly, abundance values are sampled conditionally on the expressions and the assignment of reads. Eventually, the distributions of reads to candidates are sampled conditionally on the expressed transcripts, which are corresponding values of abundances, and associated with the contingent probabilities of the observation of the reads given candidate transcripts. The fraction of a candidate transcript used in the sampling process is used as the candidate confidence.

This method can assemble more transcripts with higher precision and accurate transcript abundance levels compared to other models. While some other transcriptome assembly methods need to tune the parameters for their models and produce a final set of transcripts assembly, the generative probability model omits the parameter tuning process and provides confidence estimation for transcripts which can be used to prioritize potentially novel transcripts for validation.

Resources of computational tools

In this section, we provide a list of software implementations developed for transcriptome assembly. Table 1 summarizes the basic information of the software packages for each computational tool.

ASSESSMENT METRICS FOR TRANSCRIPTOME ASSEMBLY

Many transcriptome assembly methods including genome-guided (e.g., reference-based) and reference-free (e.g., *de novo*) have been developed to analyze the transcripts from RNA-Seq. However, evaluating the assembly performance, especially when the ground truth is not available, is still a challenge. The correct transcriptome assembly should be consistent with some statistical characteristics of the data generation process

Table 1. Summary of software for transcriptome assembly.

Software	Strategy	Link	Year of publication
Trans-ABySS [24]	*De novo*	https://github.com/bcgsc/transabyss	2010
Rnnotator [31]	*De novo*	https://sites.google.com/a/lbl.gov/rnnotator/	2010
Oases [18]	*De novo*	https://github.com/dzerbino/oases	2012
Trinity [32]	*De novo*	https://github.com/trinityrnaseq/trinityrnaseq/wiki	2013
SOAPdenovo-Trans [19]	*De novo*	https://github.com/aquaskyline/SOAPdenovo-Trans	2014
Bridger [28]	*De novo*	https://github.com/fmaguire/Bridger_Assembler	2015
Cufflinks [33]	Reference-based	http://cole-trapnell-lab.github.io/cufflinks/	2010
Scripture [34]	Reference-based	http://software.broadinstitute.org/software/scripture/	2010
TransComb [35]	Reference-based	https://sourceforge.net/projects/transcriptomeassembly/files/	2016
Bayesembler [29]	Probability model	https://github.com/bioinformatics-centre/bayesembler	2014

[36]. Some existing standard metrics, such as contigs count, coverage and N50 evaluating the whole-genome assembly performance, have been applied to transcriptome assembly validations [37,38]. For example, N50 is the weighted median of the length of contigs in transcriptome assembly. The sequencing depth is similar to the length of contigs, but non-uniform expression may have less information than transcriptome assembly [39].

Reference-based transcriptome assembly methods are based on the alignment results from existing reference genome, so the output transcripts of transcriptome assembly results are largely identical to the reference genome. Precision and recall are two statistical indicators expressing the fraction of transcriptome assembly results that are matched to the reference transcripts.

However, the reference sequences are not available in many cases which makes the evaluation tasks more difficult. The aforementioned novel generative probability model, which depends on the transcriptome assembly and the RNA-Seq reads, has been proposed to evaluate the reference-free transcriptome assembly performance. The compactness of the transcriptome assembly and the supported RNA-Seq reads into an evaluation score are combined in the novel generative probability model [40]. The evaluation score can be used to select the best assembler, filter unnecessary contigs, and optimize an assembler's parameters. Computing resource is another metric to assess the transcriptome assembly performance. Usually, reference-free methods use more time and memory than genome-guided methods [41].

CONCLUSION

With the development of RNA-Seq technologies, precision medicine is becoming attractive and practical that holds greater promise for future success. However, the excessive running time and memory requirement of the transcriptome assembly algorithms for big RNA-Seq data is still a daunting challenge. Speedy transcriptome assembly in tens of thousands to millions of patient samples without compromise of accuracy is expected to greatly advance precision medicine in the next decades.

ACKNOWLEDGEMENTS

This paper is based upon work supported by the National Science Foundation under Grand Nos. 1637312 and 1451316.

COMPLIANCE WITH ETHICS GUIDELINES

The authors Lu Wang, Lipi Acharya, Changxin Bai and Dongxiao Zhu declare that they have no conflict of interests.

This article is a review article and does not contain any studies with human or animal subjects performed by any of the authors.

REFERENCES

1. Buguliskis, J. S. (2015) Could rna-seq become the workhorse of precision medicine? Genet. Eng. Biotech. N. 35, 8–9 https://doi.org/10.1089/gen.35.05.03
2. Chen, R. and Snyder, M. (2013) Promise of personalized omics to precision medicine. Wiley Interdiscip. Rev. Syst. Biol. Med., 5, 73–82
3. Collins, F. S. and Varmus, H. (2015) A new initiative on precision medicine. N. Engl. J. Med., 372, 793–795
4. Klauschen, F., Andreeff, M., Keilholz, U., Dietel, M. and Stenzinger, A. (2014) The combinatorial complexity of cancer precision medicine. Oncoscience, 1, 504–509
5. Çakır, Ö., Turgut-Kara, N., Arı, Ş. and Zhang, B. (2015) *De novo* transcriptome assembly and comparative analysis elucidate complicated mechanism regulating *Astragalus chrysochlorus* response to selenium stimuli. PLoS One, 10, e0135677
6. Nayak, L., Ray, I., De, R. K. (2016) Precision medicine with electronic medical records: from the patients and for the patients, Ann. Transl. Med. 4 (Suppl 1), S61
7. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. and Fotiadis, D. I. (2015) Machine learning applications in cancer prognosis and prediction. Comput. Struct. Biotechnol. J., 13, 8–17

8. Vural, S., Wang, X. and Guda, C. (2016) Classification of breast cancer patients using somatic mutation profiles and machine learning approaches. BMC Syst. Biol., 10, 62
9. Hyman, D. M., Taylor, B. S. and Baselga, J. (2017) Implementing genome-driven oncology. Cell, 168, 584–599
10. Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., Szcześniak, M. W., Gaffney, D. J., Elo, L. L., Zhang, X., *et al.* (2016) A survey of best practices for RNA-seq data analysis. Genome Biol., 17, 13
11. Martin, J. A. and Wang, Z. (2011) Next-generation transcriptome assembly. Nat. Rev. Genet., 12, 671–682
12. Zerbino, D. R. and Birney, E. (2008) Velvet: algorithms for *de novo* short read assembly using *de Bruijn* graphs. Genome Res., 18, 821–829
13. Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. and Birol, I. (2009) ABySS: a parallel assembler for short read sequence data. Genome Res., 19, 1117–1123
14. Pevzner, P. A., Tang, H. and Waterman, M. S. (2001) An Eulerian path approach to DNA fragment assembly. Proc. Natl. Acad. Sci. USA, 98, 9748–9753
15. Fumagalli, M. (2013) Assessing the effect of sequencing depth and sample size in population genetics inferences. PLoS One, 8, e79667
16. Grabherr, M. G., Haas, B. J., Yassour, M., Levin, J. Z., Thompson, D. A., Amit, I., Adiconis, X., Fan, L., Raychowdhury, R., Zeng, Q., *et al.* (2011) Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat. Biotechnol., 29, 644–652
17. Bruijn, N. (1946) A Combinatorial Problem. In Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen. Series A, 49, 758–764
18. Schulz, M. H., Zerbino, D. R., Vingron, M. and Birney, E. (2012) Oases: robust *de novo* RNA-seq assembly across the dynamic range of expression levels. Bioinformatics, 28, 1086–1092
19. Xie, Y., Wu, G., Tang, J., Luo, R., Patterson, J., Liu, S., Huang, W., He, G., Gu, S., Li, S., *et al.* (2014) SOAPdenovo-Trans: *de novo* transcriptome assembly with short RNA-Seq reads. Bioinformatics, 30, 1660–1666
20. Li, R., Yu, C., Li, Y., Lam, T.-W., Yiu, S.-M., Kristiansen, K. and Wang, J. (2009) SOAP2: an improved ultrafast tool for short read alignment. Bioinformatics, 25, 1966–1967
21. Shi, C.-Y., Yang, H., Wei, C.-L., Yu, O., Zhang, Z.-Z., Jiang, C.-J., Sun, J., Li, Y.-Y., Chen, Q., Xia, T., *et al.* (2011) Deep sequencing of the *Camellia sinensis* transcriptome revealed candidate genes for major metabolic pathways of tea-specific compounds. BMC Genomics, 12, 131
22. Garg, R., Patel, R. K., Tyagi, A. K. and Jain, M. (2011) *De novo* assembly of chickpea transcriptome using short reads for gene discovery and marker identification. DNA Res., 18, 53–63
23. Zhao, Q.-Y., Wang, Y., Kong, Y.-M., Luo, D., Li, X. and Hao, P. (2011) Optimizing *de novo* transcriptome assembly from short-read RNA-Seq data: a comparative study. BMC Bioinformatics, 12, S2
24. Robertson, G., Schein, J., Chiu, R., Corbett, R., Field, M., Jackman, S. D., Mungall, K., Lee, S., Okada, H. M., Qian, J. Q., *et al.* (2010) *De novo* assembly and analysis of RNA-seq data. Nat. Methods, 7, 909–912
25. Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., Pimentel, H., Salzberg, S. L., Rinn, J. L. and Pachter, L. (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nat. Protoc., 7, 562–578
26. Trapnell, C., Pachter, L. and Salzberg, S. L. (2009) TopHat: discovering splice junctions with RNA-Seq. Bioinformatics, 25, 1105–1111
27. Yandell, M. and Ence, D. (2012) A beginner's guide to eukaryotic genome annotation. Nat. Rev. Genet., 13, 329–342
28. Chang, Z., Li, G., Liu, J., Zhang, Y., Ashby, C., Liu, D., Cramer, C. L. and Huang, X. (2015) Bridger: a new framework for *de novo* transcriptome assembly using RNA-seq data. Genome Biol., 16, 30
29. Maretty, L., Sibbesen, J. A. and Krogh, A. (2014) Bayesian transcriptome assembly. Genome Biol., 15, 501
30. Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R. and Salzberg, S. L. (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biol., 14, R36
31. Martin, J., Bruno, V. M., Fang, Z., Meng, X., Blow, M., Zhang, T., Sherlock, G., Snyder, M. and Wang, Z. (2010) Rnnotator: an automated *de novo* transcriptome assembly pipeline from stranded RNA-Seq reads. BMC Genomics, 11, 663
32. Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., Couger, M. B., Eccles, D., Li, B., Lieber, M., *et al.* (2013) *De novo* transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nat. Protoc., 8, 1494–1512
33. Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J. and Pachter, L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat. Biotechnol., 28, 511–515
34. Guttman, M., Garber, M., Levin, J. Z., Donaghey, J., Robinson, J., Adiconis, X., Fan, L., Koziol, M. J., Gnirke, A., Nusbaum, C., *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
35. Liu, J., Yu, T., Jiang, T. and Li, G. (2016) TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs. Genome Biol., 17, 213
36. Myers, E. W. (1995) Toward simplifying and accurately formulating fragment assembly. J. Comput. Biol., 2, 275–290
37. Kumar, S. and Blaxter, M. L. (2010) Comparing *de novo* assemblers for 454 transcriptome data. BMC Genomics, 11, 571
38. Zeng, V., Villanueva, K. E., Ewen-Campen, B. S., Alwes, F., Browne, W. E. and Extavour, C. G. (2011) *De novo* assembly and characterization of a maternal and developmental transcriptome for the emerging model crustacean *Parhyale hawaiensis*. BMC Genomics, 12, 581
39. Zhu, J., He, F., Wang, J. and Yu, J. (2008) Modeling transcriptome based on transcript-sampling data. PLoS One, 3, e1659

40. Li, B., Fillmore, N., Bai, Y., Collins, M., Thomson, J. A., Stewart, R. and Dewey, C. N. (2014) Evaluation of *de novo* transcriptome assemblies from RNA-Seq data. Genome Biol., 15, 553

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

