BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 10 2013, pages 1250–1259
doi:10.1093/bioinformatics/btt127

Sequence analysis

Advance Access publication March 14, 2013

BRANCH: boosting RNA-Seq assemblies with partial or related genomic sequences

Ergude Bao[1], Tao Jiang[1] and Thomas Girke[2,*]

[1]Department of Computer Science and Engineering and [2]Department of Botany and Plant Sciences, University of California, Riverside, CA 92521, USA

Associate Editor: Ivo Hofacker

ABSTRACT

Motivation: *De novo* transcriptome assemblies of RNA-Seq data are important for genomics applications of unsequenced organisms. Owing to the complexity and often incomplete representation of transcripts in sequencing libraries, the assembly of high-quality transcriptomes can be challenging. However, with the rapidly growing number of sequenced genomes, it is now feasible to improve RNA-Seq assemblies by guiding them with genomic sequences.

Results: This study introduces BRANCH, an algorithm designed for improving *de novo* transcriptome assemblies by using genomic information that can be partial or complete genome sequences from the same or a related organism. Its input includes assembled RNA reads (transfrags), genomic sequences (e.g. contigs) and the RNA reads themselves. It uses a customized version of BLAT to align the transfrags and RNA reads to the genomic sequences. After identifying exons from the alignments, it defines a directed acyclic graph and maps the transfrags to paths on the graph. It then joins and extends the transfrags by applying an algorithm that solves a combinatorial optimization problem, called the Minimum weight Minimum Path Cover with given Paths. In performance tests on real data from *Caenorhabditis elegans* and *Saccharomyces cerevisiae*, assisted by genomic contigs from the same species, BRANCH improved the sensitivity and precision of transfrags generated by Velvet/Oases or Trinity by 5.1–56.7% and 0.3–10.5%, respectively. These improvements added 3.8–74.1% complete transcripts and 8.3–3.8% proteins to the initial assembly. Similar improvements were achieved when guiding the BRANCH processing of a transcriptome assembly from a more complex organism (mouse) with genomic sequences from a related species (rat).

Availability: The BRANCH software can be downloaded for free from this site: http://manuals.bioinformatics.ucr.edu/home/branch.

Contact: thomas.girke@ucr.edu

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

Received on October 22, 2012; revised on February 27, 2013; accepted on March 8, 2013

1 INTRODUCTION

Recent advances in next generation sequencing (NGS) technologies have transformed *de novo* sequencing of genomes and transcriptomes into routine tasks that are not only feasible for large sequencing centers anymore but also for individual research groups (Martin and Wang, 2011; Zerbino and Birney, 2008). The main factors driving this development are reduced NGS run costs, and improvements of sequence volume and read length. Although it is now relatively straightforward to obtain a draft shotgun sequence of a new genome, fragmented into thousands of contigs and scaffolds, the finishing and annotation steps of a complete genome sequence are still very time consuming tasks. Transcriptome sequencing can be often a much more targeted alternative, especially when working with large genomes, or projects where the expressed gene sequences are the main features of interest to gain insight into the functionally relevant regions (e.g. proteins) encoded in a genome (Martin and Wang, 2011). For an unsequenced organism, this includes the sequencing and assembly of RNA samples where most of the genes of interest are expressed, followed by functional annotation routines of the assembled transcripts using sequence similarity searches against protein reference databases. Subsequently, one can perform RNA-Seq gene expression profiling experiments where the assembled transcripts serve as reference in the read mapping step. Challenges related to transcriptome sequencing and assembly include the following: (i) abundance differences of RNA sequences make it difficult to obtain RNA samples representing most of the expressed genes of an organism and (ii) secondary structures as well as instability of RNA molecules can result in uneven read coverage of the underlying RNA sequences. As a result, transcriptome assemblies will usually only represent a subpopulation of genes in a genome, and the assembled RNA sequences are often fragmented or incomplete with respect to their full length.

This study proposes a new method, named BRANCH, for improving the completeness of *de novo* transcriptome assemblies by making use of partial or complete genomic sequence information from the same or closely related species. It involves the initial *de novo* assembly of the RNA-Seq reads to transfrags and DNA reads to genomic contigs using existing NGS assembly software for both types of data. For instance, the genomic reads can be assembled with Velvet (Zerbino and Birney, 2008), ABySS (Simpson *et al.*, 2009), ALLPATHS (Butler *et al.*, 2008), SOAPdenovo (Li *et al.*, 2010) or IDBA (Peng *et al.*, 2010), whereas the RNA reads can be assembled with *de novo* transcriptome assemblers like Velvet/Oases (Schulz *et al.*, 2012; Zerbino and Birney, 2008), Trinity (Grabherr *et al.*, 2011), Trans-ABySS (Robertson *et al.*, 2010), SOAPdenovo-Trans (Li *et al.*, 2010) or T-IDBA (Peng *et al.*, 2011). In a downstream transcriptome assembly enhancement step, the genomic contig information is used to identify novel exons, extend incomplete

*To whom correspondence should be addressed.

transfrags and join fragmented ones using the BRANCH algorithm introduced in this study. This hybrid approach of guiding transcriptome assemblies with preliminary genomic sequencing information is a practical and cost-effective possibility, as one can sequence nowadays a genomic sample of a 1 GB genome of interest at 20–50 coverage with the read output from only 1–2 flow cell lanes of a modern NGS instrument. Technically, the collection and sequencing of a genomic sample is also very straightforward, and stability issues or abundance variations of sequences are less a concern with genomic DNA than RNA. Alternatively, the genome contigs can be substituted by an existing genome sequence from a related species with high enough DNA sequence identity (usually $>90-95\%$) to the RNA-Seq sample. This option eliminates the need for generating the genomic contig dataset.

The genomic sequences provide an additional backbone of evidence for improving *de novo* transcriptome assemblies by minimizing their typical errors and limitations, such as incomplete transfrags (e.g. missing exons), fragmented transfrags, chimeric transfrags, and so forth owing to low read coverage and base calling errors. When aligning the transfrags and RNA-Seq reads against given genomic contigs, one can extend and correct many of these fragmented or incomplete transfrags. For instance, two transfrags aligned next to each other on the same contig can be joined if a sufficient number of RNA reads can be aligned to support this merger. Similarly, a transfrag can be extended if the RNA read coverage along the corresponding region of the genomic sequences indicates a truncated transcript sequence. Because genomic contigs also contain errors, it is important to allow in this process only those transfrag modifications that are supported by high-quality alignments.

BRANCH contains features that intersect in parts with reference-based splice variant assembly tools (sometimes referred to as *ab initio* assemblers; Feng *et al*., 2010; Guttman *et al*., 2010; Li *et al*., 2011; Trapnell *et al*., 2010), such as the identification of splice variants from RNA sequence alignments against a reference. What makes BRANCH distinct from these tools is that it is designed to maximize the number and completeness of exons contained in preassembled transfrags guided by partial or complete genome sequences from the same or a closely related organism. It does this even for sequence regions with low RNA read coverage. This functionality is novel and relevant for *de novo* transcriptome assembly projects of unsequenced or only partially sequenced genomes because the additional exonic sequence information will contribute to the functional annotatability of the coding regions of RNA sequences in downstream protein similarity searches.

2 METHODS

2.1 Overview of the algorithm

BRANCH consists of two major components: *Exon Detection* and *Transfrag Extension*. The *Exon Detection* component aligns the RNA reads against the preassembled *de novo* transfrags, and then it aligns both the transfrags and the remaining reads (that failed to align) against preassembled genomic contigs or a closely related genome using a modified version of the BLAT alignment program (see the discussion later in the text; Kent, 2002). Subsequently, it identifies exons and splice junctions in the read pileups against the contigs. Pileup regions meeting certain minimum length and read coverage requirements are considered exons, and low coverage regions between them are introns if they are spanned by gapped alignments and splice junction signals. In addition to the exons contained in the initial transfrags, this step identifies novel candidate exons that are often missed in *de novo* transcriptome assemblies, mainly owing to uneven RNA read coverage. Guided by the additional DNA sequence information, BRANCH is designed to resolve those low coverage regions very efficiently. The *Transfrag Extension* component builds a weighted directed acyclic graph where the nodes represent the detected exons and the edges splice junctions while recording the paths through the graph corresponding to each transfrag. The weight of an edge is determined by the read density supporting the connectivity between the nodes. It then extends the recorded paths (i.e. transfrags) by finding the minimum number of paths with the minimum total weight that cover all recorded paths as well as the remaining nodes (i.e. the novel exons), resulting in extended transfrags.

The following describes the BRANCH algorithm in more details. Section 2.2 introduces the BLAT-based alignment method, and Sections 2.3 and 2.4 describe BRANCH's exon detection and transfrag extension algorithms, respectively. Some illustrations of the algorithms are given in Figures 1 and 2.

2.2 Alignment steps

An important preprocessor for our method is an alignment tool that can accurately align short RNA reads as well as much longer transfrags against genomic contigs while inserting gaps at exon–intron junctions. Several alignment tools are available for mapping short RNA reads with gaps and limited numbers of mismatches against genome sequences. These include TopHat (Langmead *et al*., 2009; Trapnell *et al*., 2009), GMAP (Wu and Watanabe, 2005), SpliceMap (Au *et al*., 2010) and MapSplice (Wang *et al*., 2010). For aligning longer transfrag sequences, software tools designed for generating long gapped alignments, such as BLAT, are more suitable than short read aligners. Hence, the current implementation of BRANCH uses a modified version of BLAT that

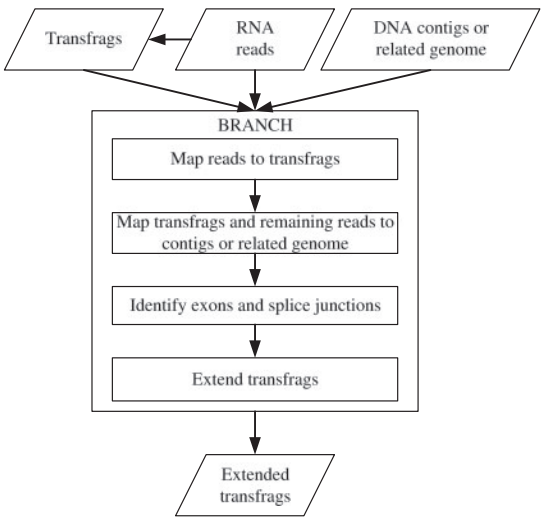

Fig. 1. Input, processing steps and output of BRANCH. RNA reads are assembled with existing assembly software to *de novo* transfrags. BRANCH maps the RNA reads to the transfrags, and the transfrags and the remaining RNA reads to the genomic sequences. The latter are usually custom assembled contigs or gene sequences from a related organism. Guided by the resulting read pileups, BRANCH identifies existing and novel exons and splice junctions and uses this information to extend the initial transfrags

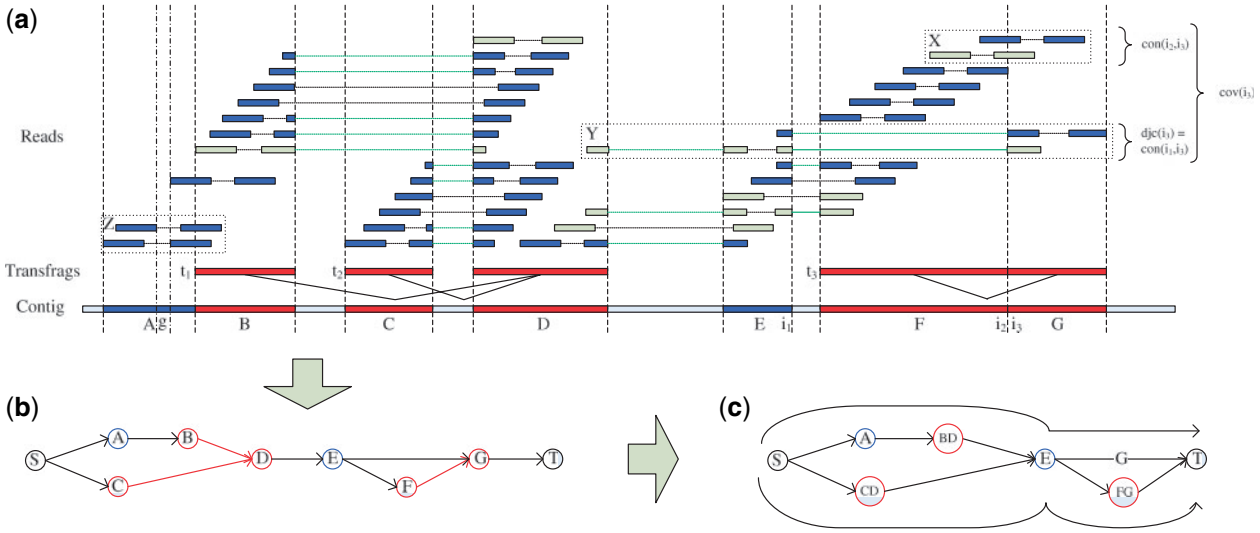

Fig. 2. Illustration of important features of BRANCH algorithm. (**a**) A sample pileup is shown containing PE RNA reads, preassembled transfrags (t_1 to t_3) and one contig with exons A to G. Reads of good quality are indicated in blue and low quality reads in grey. PE pairs and spliced read fragments are connected with thin black and green lines, respectively. The red bars (B, C, D, F and G) in the contig are the exons identified by BRANCH based on the alignment of the preassembled transfrags against the contig. The blue bars (A and E) in the contig are two additional exons identified by BRANCH based on spliced and PE reads aligning with both contig regions covered by transfrags and those not covered by transfrags. Those exons (here A and E) are often missed by *de novo* transcriptome assemblers owing to insufficient read coverage and/or sequence errors. The sequencing gap g in exon A could be closed with PE reads in rectangle Z, because their insert size obtained from the alignment against the contig, agrees with the expected insert size of the library. Another situation where BRANCH improves transcriptome assemblies is given on the right side of the diagram. Here the exon region FG, corresponding to transfrag t_3, is subdivided by an internal splice site i_1/i_2 into two exons. This is supported by a minimum number of splice junction reads with gaps (rectangle Y) spanning contig positions i_1 and i_3. The coverage $\mathrm{cov}(i_3)$ is the number of junction reads overlapping with base position i_3, here reads in rectangles X and Y; the downstream junction coverage $\mathrm{djc}(i_3)$ is the number of junction reads overlapping with base positions i_1 and i_3 in rectangle Y where $i_1 + 1 < i_3$; the connectivity $\mathrm{con}(i_2, i_3)$ between positions i_2 and i_3 is the number of reads overlapping with bases i_2 and i_3 in rectangle X where $i_2 < i_3$. (**b**) A junction graph has been constructed from the alignment. In this graph, exons are nodes and edges are connections among them that are weighted based on the read support from the spliced alignments. Source and sink nodes are added at the beginning (S) and the end (T) of the graph, respectively. The paths corresponding to the *de novo* transfrags are marked in red: $B \to D$, $C \to D$ and $F \to G$. (**c**) The TE Algorithm collapses these paths to path nodes BD, CD and FG. The resulting Minimum weight Minimum Path Cover with given Paths (MMPCP) in the original graph (b) includes the paths indicated by round arrows: $S \to A \to BD \to E \to G \to T$ and $S \to CD \to E \to FG \to T$. Each of them corresponds to an extended transfrag

we have optimized to align both types of RNA sequences with acceptable run time, sensitivity and error tolerance against genomic contigs. These changes to the BLAT executable are similar to those introduced by Grant *et al.* (2011), but they have been customized for our specific needs of aligning long and short sequences. They include early filtration of candidate alignments to minimize execution time, disk space and support for handling paired-end (PE) read data. In addition, the boundaries of identified introns are screened for the presence of canonical (GT-AG) and non-canonical (e.g. GC-AG, AT-AC) splice sites. This information is used to optimize the exon–intron junctions obtained from the alignment results.

2.3 Exon detection algorithm

The *Exon Detection* (ED) Algorithm identifies exons and splice junctions. It uses the modified BLAT software described in the previous section to first align the RNA reads (single or PE) against the transfrags and then the transfrags as well as all the remaining reads (not mapped in previous step) against the contigs or a related genome reference. The latter read pool contains RNA reads derived from exon sequences missing in the transfrag sequences, whereas others may have failed to align owing to base calling errors. After aligning the transfrags and reads to the contigs, the ED Algorithm identifies exons and splice junctions guided by the coverage information obtained from the alignment result. Regions with a minimum RNA read coverage b and a minimum width a are considered exons. Both a and b are user definable parameters. In future upgrades of BRANCH, these thresholds will be optimized for the user dynamically to minimize false positive exon predictions owing to contaminations with unspliced pre-mRNAs and other sources of noise in the data. After identifying candidate exons, the algorithm locates splice junctions between them based on the gap positions in the tranfrag sequences and/or RNA junction reads aligned against the contig sequences. Alternative splice sites within exons are identified in areas where a minimum number of junction reads share the same gap that spans across one or more exon regions. Figure 2a illustrates these steps with an example. The outcome of the ED Algorithm is additional exonic sequences not contained in the initial transfrag sequences. This includes extensions of incomplete exons and the identification of novel exons (complete or partial) along with their connections. The detailed steps of the ED Algorithm and its pseudo code are given below.

Step 1 is the alignment of the RNA reads, transfrags and contigs as described earlier in the text.

Step 2 identifies an exon region based on the alignment, where we denote the coverage of a contig position i by $\mathrm{cov}(i)$. In Figure 2a, the coverage of junction base i_3 is the number of junction reads overlapping with it in rectangles X and Y. The reads in rectangle X align over their full length against the transfrag t_3 and the contig, whereas the reads in

rectangle Y align completely only against the contig. Both read sets overlap with position i_3. The algorithm computes the coverage for each contig base and identifies any contig region, with start and end positions $[l, r]$, as an exon range, if the width of the contig region satisfies $r - l + 1 \geq a$ and the average coverage of the contig region satisfies $\sum_{l \leq i \leq r} \frac{\text{cov}(i)}{(r-l+1)} \geq b$, where a and b are the minimum width and the minimum coverage requirements, respectively. In certain cases, the newly identified exon regions may be fragmented in areas with very low or no RNA read coverage. Suppose a novel exon range $[l, r]$ contains sufficient read coverage in subranges $[l, i]$ and $[j, r]$ $(i < j)$, but subrange (i, j) has zero coverage. In such a case, two partial exons $[l, i]$ and $[j, r]$ will be identified instead of the complete exon $[l, r]$. Such gaps can be closed, if there is a sufficient number of PE reads spanning $[l, i]$ and $[j, r]$, and the mapping distances of the read pairs agree with the approximate insert length of the library. An example of such a case is given in Figure 2a, where the coverage gap g divides exon A into two parts, but it can be closed with the PE read support shown in rectangle Z. To minimize the risk of incorporating introns, this type of gap closures are only performed if the mapping distances of the read pairs agree with the approximate insert length of the RNA-Seq library. Alternatively, the user can specify this parameter.

Step 3 identifies alternative splice junction sites within exons. Here, we define the *upstream* and the *downstream junction coverage*. The upstream junction coverage at contig position i, denoted as ujc(i), is the number of reads having bases at positions j and $j+1$ aligned at contig positions i and $k > i+1$, respectively. Similarly, the downstream junction coverage at contig position i, denoted as djc(i), is the number of reads having bases at positions $j-1$ and j aligned at contig positions $k < i-1$ and i, respectively. For example, the downstream junction coverage at base i_3, djc(i_3), is the number of junction reads in rectangle Y of Figure 2a covering i_3. The aligned junction reads overlap with bases i_1 and i_3, where $i_1 + 1 < i_3$. The algorithm records the upstream and downstream junction coverages for each contig base, and then splits such a region $[l, r]$ at contig positions i and $i+1 (l \leq i < i+1 \leq r)$, if the upstream junction coverage at i satisfies ujc(i) $\geq c$ or the downstream junction coverage at $i+1$ satisfies djc($i+1$) $\geq c$, where c is the minimum upstream/downstream junction coverage requirement to split exon regions.

Step 4 determines which exons are joined based on their *connectivity* in the alignment result. The connectivity between the last base of an exon and the first base of a downstream exon at contig positions i and $j > i$, denoted as con(i, j), is the number of reads having bases at positions k and $k+1$ aligned at contig positions i and j. In Figure 2a, the connectivity between i_2 and i_3 is the number of reads in rectangle X with matching bases at positions i_2 and i_3. The algorithm computes the connectivity for each pair of exons and identifies two exons at positions $[l_1, r_1]$ and $[l_2, r_2] (r_1 < l_2)$ as a junction, if the connectivity of the pair of boundary bases at r_1 and l_2 satisfies con(r_1, l_2) $\geq d$, where d is the minimum connectivity requirement to connect two exons.

Algorithm 1 Exon detection: ED(R, T, C)

1: Align reads R to *de novo* transfrags T with BLAT and then align T and the unaligned reads R_n to contigs C

2: Record the coverage for each base at contig position i, and identify each region $[l, r]$ in a contig, where $r - l + 1 \geq a$ and $\sum_{l \leq i \leq r} \frac{cov(i)}{(r-l+1)} \geq b$

3: Record the upstream and downstream junction coverages for each base at contig position i, split a region $[l, r]$ at bases i and $i+1 (l \leq i < i+1 \leq r)$, if ujc($i$) $\geq c$ or djc($i+1$) $\geq c$ and identify the resulting regions as exons

4: Record the connectivity for each pair of bases at contig positions i and $j > i$ and identify the splice junction of each exon pair $[l_1, r_1]$ and $[l_2, r_2] (r_1 < l_2)$, if con(r_1, l_2) $\geq d$

2.4 Transfrag extension algorithm

The *Transfrag Extension* (TE) Algorithm extends and often joins *de novo* transfrags based on the additional exon sequences and splice junctions identified in the previous *Exon Detection* step. For this, it identifies the connections best supported by the data and then joins the corresponding sequence fragments accordingly. The final output is extended transfrag sequences, as well as novel transfrags. For example, if the connectivity data obtained in the previous step indicate that a newly identified exon $\in$ is connected with an existing exon ϵ', and ϵ' appears in two separate transfrags t and t', then the algorithm has to decide if ϵ is connected with t and/or t'. A similar, but not identical, problem is solved by the Cufflinks algorithm for identifying transcript variants in RNA-Seq data (Trapnell *et al*., 2010). Thus, our algorithm adopts certain components of this method, whereas others are specific to BRANCH's main application addressing the transfrag extension problem.

2.4.1 Mathematical formulations

DEFINITION 1. *A junction graph is a directed acyclic graph, where each node represents an exon and each edge represents a splice junction.*

Based on the exons and splice junctions identified by the ED Algorithm, BRANCH builds a *junction graph* G where each node v represents an exon ϵ and the connecting edges are splice junctions among exons. Two nodes v and v' are connected by an edge $e(v, v')$ if their corresponding exons ϵ and ϵ' are junction exons. Similar to the approach chosen by Trapnell *et al*. (2010), the graph is weighted based on the *percent-spliced-in* value introduced by Wang *et al*. (2008). The latter expresses the density of the RNA reads supporting a transcript relative to the density of all the RNA reads mapping to the corresponding genomic region of the transcript. The percent-spliced-in value for any exon ϵ (and thus node v in the junction graph) is defined by:

$$\psi_\epsilon = \frac{\text{number of compatible reads overlapping with exon } \in}{\text{number of reads overlappling with exon } \in \times \text{length of exon } \in}. \quad (1)$$

In the above formula, the *overlap* and *compatibility* of an aligned RNA read γ and an exon ϵ are defined as follows. Read γ and exon ϵ overlap if and only if their start coordinates $l(\gamma)$ and $l(\epsilon)$ and end coordinates $r(\gamma)$ and $r(\epsilon)$ in the reference genome satisfy $l(\gamma) \leq l(\epsilon)$ and $r(\gamma) \geq l(\epsilon)$, or $l(\epsilon) \leq l(\gamma)$ and $r(\epsilon) \geq l(\gamma)$. Overlapped read γ and exon ϵ are compatible if and only if any gap $[i(\gamma), j(\gamma)]$ in the alignment of γ does not overlap with the exon ϵ. The value for any exon pair ϵ and ϵ' [and thus edge $e(v, v')$] is defined as the absolute difference of their weights with amplification:

$$w(\in, \in') = -log(1 - |\psi_\epsilon - \psi'_\epsilon|) \quad (2)$$

The smaller w is, the more likely that the ϵ and ϵ' are from the same transcript.

Clearly, each given transfrag corresponds to a path in G. These are called *given paths*. As we are interested in extending the transfrags by possibly merging them and adding more novel exons, we formulate the transfrag extension problem in BRANCH as a combinatorial optimization problem called the Minimum weight Minimum Path Cover with given Paths (MMPCP) problem. An MMPCP is a smallest set of paths with the minimum weight in the junction graph G that contains

all the given paths P as subpaths and cover all the nodes of V. Here, we seek the smallest number of paths because we would like to maximize the length of each extended transfrag. The minimum total weight requirement guarantees that any two exons ϵ and ϵ' in each extended transfrag are from the same true transcript.

2.4.2 Outline of the TE algorithm Our idea to find an MMPCP is to build a new junction graph G' from G by (i) converting each given path $p \in P$ to a node $v(p)$ and (ii) maintaining the connection between any two nodes v and $v' \notin p$ through a subpath of p by introducing an edge $e(v, v')$. The new node $v(p)$ will be referred to as path node and the new edge $e(v, v')$ as path edge. To keep the two graphs equivalent, the total weight of a given path will be added to each in-edge of the corresponding path node, and the path edges will be weighted using the total weights of the corresponding subpaths. This conversion is illustrated in Figure 2b and c. Then, we invoke a Combinatorial Optimization (CO) Algorithm for solving the Minimum weight Minimum Path Cover (MMPC) problem in the new graph G' (see Supplementary Materials). If P' is the resulting MMPC for G' from the CO Algorithm, the paths in P', or the transfrags they represent, may not be fully extended. To address this, we can iterate the above process for solving the MMPCP problem by recording P' as new given paths and extending them recursively, until they cannot be extended anymore. The TE Algorithm is more formally outlined in the following pseudocode. The final output of the TE Algorithm consists of transfrags that have been extended with exonic sequences from the Exon Detection step, as well as some novel transfrags.

Algorithm 2 Transfrag extension: TE(G, P)

```
Assign weights to the edges of G using Equation (2)
for each given path p ∈ P do
  Convert p to path node v(p) and add the total weight of p to each
  in-edge of v(p)
  for any pair of nodes v and v'∉p do
    if there is a path p'∉P from v to q ∈ subpath(p) and then to v' then
      Introduce a path edge e(v, v')
      Weight e(v, v')
    end if
  end for
  Delete p from G
end for
{G is converted to G'}
P' ← CO(G')
if P' = P then
  return the resultant MMPCP P'
else
  return TE(G', P')
end if
```

2.5 Implementation and performance

BRANCH has been implemented in C++ with the LEMON library (Dezso *et al.*, 2011) for Linux operating systems. The modified BLAT executable is distributed along with BRANCH. The expected input includes RNA reads (single or PE), assembled transfrags and genomic contigs or gene sequences from a closely related species. Most of BRANCH's execution time is spent on the initial alignment with BLAT ranging from 0.1–0.5 h per million reads. The subsequent steps are more memory than CPU intensive for storing the genomic contigs (0.1 GB RAM per million nucleotides). Both the execution time and memory usage of BRANCH are approximately linear in the number of RNA-Seq reads and size of the genomic contigs, respectively.

3 EVALUATION

3.1 Test results with simulated data

3.1.1 Background The performance of BRANCH was tested with real and simulated data. The main objective of these experiments was to assess the efficiency of BRANCH for improving the representation of full-length transcripts in *de novo* transcriptome assemblies, but also its splice variant resolution, error tolerance and robustness with respect to variable degrees of incomplete representation of transcript and genomic sequences. Although tests on real data provide more reliable results for the performance of an algorithm, simulated data were included here because they allow a more systematic evaluation of a wide variety of data properties than this would be possible with real data only. To mimic in these tests real data as much as possible and minimize bias toward any method, all sequences were randomly sampled from a real genome, meaning they were only partially synthetic. The results on real datasets are given in the next section. In the tests with simulated data, we varied the number of RNA reads, the average length of the contigs, the relative genome coverage by the contigs and the base call error rates in both the RNA reads and the contigs. Benchmarking BRANCH's main utility—the enhancement of RNA-Seq assemblies guided by genomic sequences—against other tools is currently not easily possible owing to the lack of software designed for this purpose. However, a very informative performance measure is to determine how well BRANCH can improve *de novo* assembled transfrags with respect to their full-length and gene coverage in a genome. For this, we compared the final results generated by BRANCH with the initial *de novo* transfrags that we generated in the tests on simulated data with the Velvet/Oases and Trinity transcriptome assemblers. Velvet/Oases and Trinity were chosen here among other software options (e.g. Trans-ABySS, SOAPdenovo-Trans) because of their good sensitivity and precision performance (Zhao *et al.*, 2011).

3.1.2 DataSets and Tests The simulated test datasets were randomly sampled from the genome and transcriptome sequences of *Caenorhabditis elegans* provided by Ensembl's FTP site. From the genome sequence, we sampled three types of contig sets and from the transcriptome two types of RNA-Seq sets as follows: (i) contigs of variable length of 1, 10, 50 and 100 kb; (ii) contigs with variable coverage of the *C.elegans* genome of 40, 60, 80 and 100%; (iii) contigs with variable sequence error rates of 0, 1, 2 and 3% by substituting bases at random positions; (iv) different numbers (10, 30, 50 and 70 million) of PE RNA reads of 2×100 bp length and 200–300 bp insert length while maintaining an abundance distribution among the reference transcripts that is typical for RNA-Seq samples (see Supplementary Table S-1); and (v) RNA reads with variable error rates of 0, 1, 2 and 3%. The simulated RNA-Seq sets were assembled to transfrags using Velvet/Oases with its parameter optimization script and Trinity with its default parameter settings.

To be consistent with recent studies on *de novo* RNA assemblies, we define in our tests *sensitivity* and *precision* in a similar manner. *Sensitivity* is the number of reference transcripts, which could be aligned, here with BLAT, to a transfrag with ≥95% identity over ≥80% of the transcript's length and ≥95% of the transfrag's length (Martin and Wang, 2011). Additionally, test

results with variable length coverage values are given in Section 3.2.3 and Figure 4. *Precision* is defined as the percentage of transfrags, which could be aligned to a reference transcript with ≥95% identity over ≥95% of the transfrag's length, but without a minimum length coverage requirement for the transcript (Zhao *et al.*, 2011; Schulz *et al.*, 2012; Robertson *et al.*, 2010). Moreover, we compare among the different assembly methods the following performance parameters: numbers of covered transcripts, complete transcripts and completely represented exonic regions of genes. For the latter two, we also require ≥95% identity and ≥95% length coverage of the reference and the transfrag.

3.1.3 Results Figure 3 and Supplementary Tables S-2 to S-6 give the test results for the simulated data sets for variable contig lengths, contig sequence error rates, contig coverages, numbers of RNA reads and RNA read base call error rates, respectively. All other parameters are constant settings, which are specified in the legends. Compared with the input transfrags generated by Velvet/Oases and Trinity, BRANCH post-processing improves their sensitivity and precision substantially by 2.3–19.9% and 1.7–15.7%, respectively. The relative sensitivity improvements by BRANCH for both assemblers are ~2-fold higher when the coverage of the genome by contigs is raised from 40% to 100% (Fig. 3c and Supplementary Table S-4), whereas increasing sequence error rates from 0 to 3% in the contigs have a less pronounced impact by reducing the relative sensitivity improvements in the most extreme cases by 20–34% (Fig. 3b and Supplementary Table S-3). Because BRANCH also identifies novel transfrags, the initial number of transfrags increases as expected (in these tests by 6.7–72.6%). The 487–5394 transfrag extension events recorded in the BRANCH results lead to 0.2–9.0% more completely assembled transcripts increasing the number of completely assembled exonic regions of genes by 0.1–7.6%. The latter improvements are less pronounced owing to the more stringent full-length criteria applied in these cases. Most importantly, the transfrags processed by BRANCH have a 6.0–18.5% higher coverage of the total number of exons annotated in the *C.elegans* genome than the initial transfrag sets. These results indicate that BRANCH improves the chosen quality parameters of transcriptome assemblies relatively effectively over the range of test variables evaluated in these experiments.

3.2 Test results with real data

3.2.1 Experimental design The performance of BRANCH on real data was tested with published Illumina NGS samples available in NCBI's Sequence Read Archive. To generate meaningful test results, it was important to choose here NGS data meeting today's standards for efficient RNA-Seq transcriptome assemblies with respect to read length (>50 bp) and PE read information. BRANCH's performance on the two main types of genomic guide sequences was evaluated by including in one set of tests custom genomic contigs assembled from NGS reads of the same organism as the RNA reads, and in another case existing genome sequence from a closely related organism (Table 1). The influence of the completeness of the genomic sequence information on the performance of BRANCH was tested by comparing the results guided by assembled contigs with those from complete gene sequences.

Two datasets were chosen from diverse multicellular eukaryotic organisms (*C.elegans* and mouse) to account for splice variants and variable degrees of sequence complexity, and a third one was from a unicellular eukaryotic organism (*Saccharomyces cerevisiae*) with a densely organized genome and rare alternative splicing. To evaluate the impact of directional information in the RNA reads, we used in two cases non-strand-specific RNA-Seq samples and in another case a strand-specific sample. The RNA reads from all sample sets were assembled with Velvet/Oases and Trinity (Grabherr *et al.*, 2011; Zhao *et al.*, 2011). In case of Trinity, the default parameter settings recommended by its developers were used. DNA reads were assembled to contigs with Velvet using the VelvetOptimiser tool for parameter optimization. To also compare against an alignment-based splice variant assembler, we included Cufflinks, which is a fundamentally different method compared with the above *de novo* assemblers. Cufflinks was only included in the test case with the known genome sequences as guide reference because it was the only situation where the minimal input data types, required for this method, were available. The splice junction information was obtained by aligning the RNA reads with Tophat (version 2) against the genomic sequences. Both Tophat and Cufflinks were run with their default parameter settings.

The results obtained from the different tests were used to compute similar quality parameters (Table 1) as in the previous section assessing among other properties the full-length and

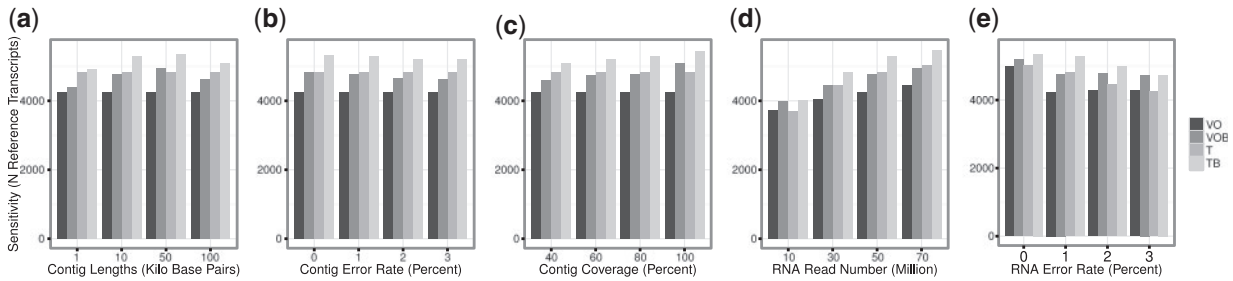

Fig. 3. Sensitivity tests on simulated data. Sensitivity measures of Velvet/Oases (VO), Velvet/Oases with BRANCH post-processing (VOB), Trinity (T) and Trinity with BRANCH post-processing (TB) are plotted for variable (**a**) contig lengths, (**b**) sequence error rates in contigs, (**c**) relative coverages of the reference genome by contigs, (**d**) number of RNA reads and (**e**) base call error rates in RNA reads. The invariable parameter settings include 10 kb contig length, 1% sequence errors in contigs, 80% contig coverage, 50 million PE RNA reads, and 1% base call error rate in RNA reads

Table 1. Performance on real data

Method	Sensitivity	Precision	N Transfrags	N Complete transcripts	N Complete genes	N Covered transcripts	N Exons	N Extensions	N Proteins
(a) RNA-Seq Assembly of C.elegans (BRANCH Guided by Genomic Sequences from C.elegans)									
Velvet/Oases (VO)	5015	32.8%	55 083	3248	2986	3844	96 078	—	3839
Velvet/Oases + BRANCH (VOB)[1]	5332	43.3%	62 201	3446	3159	4187	107 467	5726	4683
Velvet/Oases + BRANCHg (VOBg)[2]	6602	42.3%	56 369	4419	3973	4825	107 876	7696	4811
Trinity (T)	5048	39.4%	32 083	3708	3416	4152	116 128	—	4866
Trinity + BRANCH (TB)[1]	5303	43.3%	51 997	3848	3539	4360	121 484	5320	5706
Trinity + BRANCHg (TBg)[2]	6309	42.3%	49 197	4500	4080	4852	122 345	6877	5892
Cufflinksg (Cg) [2]	5147	47.7%	14 685	3300	3073	3436	114 029	—	2997
(b) Strand-Specific RNA-Seq Assembly of S.cerevisiae (BRANCH Guided by Genomic Sequences from S.cerevisiae)									
Velvet/Oases (VO)	282	38.6%	75 053	54	54	132	1211	—	926
Velvet/Oases + BRANCH (VOB)[1]	442	39.8%	80 831	94	94	212	1875	9514	1239
Trinity (T)	315	41.0%	11 451	146	146	201	4375	—	1957
Trinity + BRANCH (TB)[1]	412	41.3%	13 394	206	206	261	4498	2322	2119
(c) RNA-Seq Assembly of Mouse (BRANCH Guided by Genomic Sequences from Rat)									
Velvet/Oases (VO)	7103	23.4%	447 689	2922	2331	5230	123 070	—	12 260
Velvet/Oases + BRANCH (VOB)[3]	7417	24.5%	518 360	3073	2478	5595	123 939	3325	12 747
Trinity (T)	4593	25.4%	143 757	1971	1621	3177	100 453	—	8676
Trinity + BRANCH (TB)[3]	4916	27.2%	187 478	2128	1776	3501	101 964	1295	9217

Assembly results of RNA-Seq data from (a) *C.elegans*, (b) *S.cerevisiae* and (c) *M.musculus* are given for the transcriptome *de novo* assemblers Velvet/Oases and Trinity. The splice variant assembler Cufflinks was included in one case where its required input was available. The resulting transfrags were post-processed with BRANCH (e.g. referred to as Trinty + BRANCH) using under (a) custom assembled genome contigs[1] or known gene sequences[2] from *C.elegans*, and under (c) the gene sequences from the rat genome[3]. The latter evaluates BRANCH's performance for a case where a closely related guide genome sequence is available. The sample from *S.cerevisiae* (b) uses custom assembled contigs along with strand-specific RNA-Seq data from the same organism. The other two cases contained non-strand specific RNA samples. The acronyms introduced in the first column serve as sample labels in Figures 3–5. The performance criteria considered in the remaining columns are described in Sections 3.1.2 and 3.2.1.

splice variant resolution of the transfrags. To also evaluate the functional annotatability of the assembled transcripts before and after processing them with BRANCH, they were used as queries in BLASTX searches (E-value cutoff 10^{-9}) against the protein databases of the corresponding organisms. The obtained results were queried for nearly complete protein matches requiring $\geq$ 95% identity on the protein sequence level.

3.2.2 Datasets The first NGS sample set is from *C.elegans*. Its genomic read set contained 57 million 2×54–76 bp long PE reads (accessions: SRR066623, SRR066625; Weber *et al.*, 2010) and its RNA-Seq set contained 72 million 2×100 bp PE reads (accession: SRR316929; Hillier *et al.*, 2009). The second sample set is from mouse (*Mus musculus*) with 34 million 2×76 bp PE RNA-Seq reads (accessions: SRR290901, SRR290902; unpublished). The gene sequences from rat (*Rattus norvegicus*) were used in this case as genomic guide sequence to test BRANCH's performance for a situation where a related genome sequence is available. The third sample set is from *S.cerevisiae* with 4 million 2×76 bp long PE genomic reads (accessions: SRR527545, SRR527546; unpublished), and 10 million 2×76 bp strand-specific PE RNA-Seq reads (accession: SRR059177; Levin *et al.*, 2010).

3.2.3 Assemblies assisted with custom genome contigs The performance test results for the *C.elegans* data are given in Table 1a. In comparison with the initial transfrags assembled by Velvet/Oases, BRANCH shows a 6.3 and 10.5% improved sensitivity and precision performance, respectively, when guided by the 88 175 genome contigs assembled for this experiment representing 90.4% of its genome. To evaluate the sensitivity performance over a wider threshold range of transcript length coverage values, Figure 4 compares among the different assembly methods the number of reference transcripts from *C.elegans* that aligned with the transfrags over increasing minimum overlap values from 10% to 90%. BRANCH exhibits here a consistent improvement compared with the other methods over the full range of overlap thresholds. When comparing the sensitivity performance among the different methods for variable expression levels (see Fig. 5), BRANCH shows the greatest improvements for weaker expressed transcripts. This is in agreement with its design feature for improving the assembly of transfrags with low read coverages.

With respect to the other performance parameters recorded in Table 1, BRANCH also increases the number of complete transcripts, complete genes, covered transcripts and exons annotated in the *C.elegans* genome by 6.1, 5.8, 8.9 and 11.9%, respectively. When using the transfrags in translated BLASTX searches against the *C.elegans* protein database, the BRANCH results show a remarkable increase (22.0%) of the number of complete protein sequences encoded in the assembled transcript set. In Table 1, the number of nearly complete protein sequences is usually larger than the number of complete transcripts because the latter also contain untranslated 5′ and 3′ regions that make the full-length cut-off criteria ($\geq$95% sequence identity over $\geq$95% of the reference length) more stringent for transcripts than for proteins. BRANCH processing extends 5726 transfrags from the initial assembly, and it identifies many novel transfrags

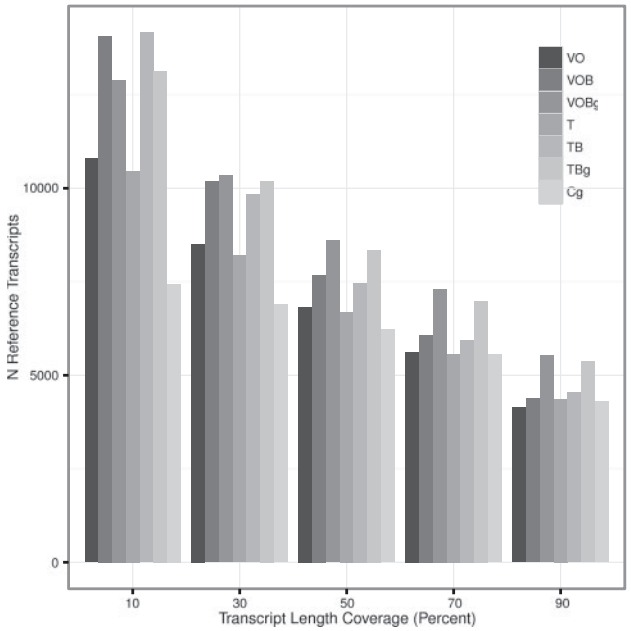

Fig. 4. Transcript length coverage. The number of reference transcripts of the *C.elegans* dataset is plotted that aligned with the transfrags over increasing overlap thresholds from ≥10% to ≥90%. The acronyms assigned to the different methods in the legend are defined in the first column of Table 1

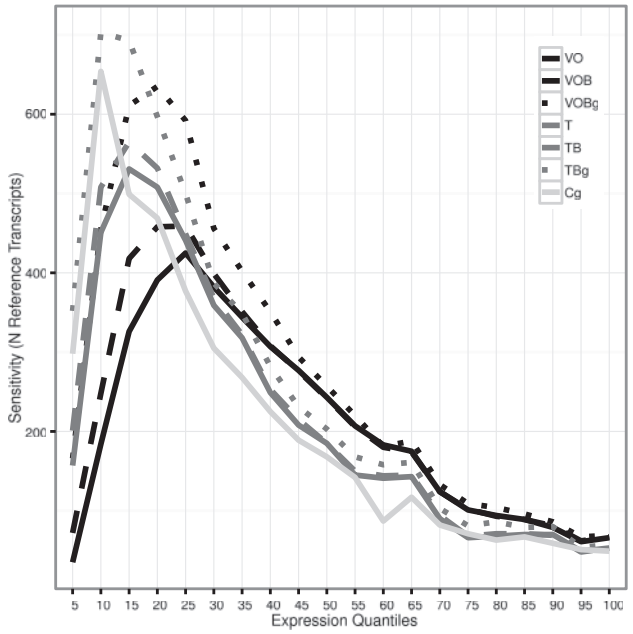

Fig. 5. Sensitivity performance for variable expression quantiles of *C.elegans* data. The number of assembled transcripts (*x*-axis) are plotted across different expression levels (*y*-axis). The acronyms assigned to the different methods in the legend are defined in the first column of Table 1

resulting in a 12.9% increase of the total transfrag pool. When the same tests are performed using Trinity instead of Velvet/Oases as RNA-Seq *de novo* assembler, then BRANCH shows a 5.1% and 3.9% improved sensitivity and precision, and it increases the number of complete transcripts, complete genes, covered transcripts, exons and nearly complete proteins by 3.8, 3.6, 5.0, 4.6 and 17.3%, respectively. Overall, these improvements are the result of 5320 transfrag extension events generated by BRANCH. As expected, when the gene sequences instead of assembled contigs are provided, the transfrags processed by BRANCH show additional improvements compared with the ones obtained from Velvet/Oases. With a total of 7696 transfrag extension events obtained by BRANCH, the sensitivity improves by 31.6, the precision by 9.5 and the number of complete and covered transcripts by 36.1 and 25.5%, respectively. In addition, the number of complete proteins increases by 25.3%. For Trinity, the same tests result in similar improvements by BRANCH: the sensitivity, precision, number of complete transcripts, number of covered transcripts and the number of complete proteins improve by 25.0, 2.9, 21.4, 16.9 and 21.1%, respectively.

In the test case with known reference genes, BRANCH also outperforms Cufflinks in sensitivity by 22.6–28.3% but shows a slightly lower precision (5.4%). For the remaining test parameters, BRANCH's performance is consistently superior over Cufflinks'. In addition, Cufflinks is unable to produce even nearly as good results (data not shown) in the tests with the other types of genomic guide sequences, BRANCH has been specifically designed for, including custom contigs and genomic sequences from related genomes. The main reasons for these performance difference are as follows. First, Cufflinks has been designed for a different use case, which is the prediction of splice variants for completed genomes, preferentially with well-defined gene/exon boundary annotations. Second, owing to its transfrag input, BRANCH's performance greatly depends on the quality of the upstream *de novo* assemblies. If those were of poor quality, then Cufflinks' ranking in this comparison could change. Third, BRANCH has been optimized to extend transfrags in sequences regions with very low RNA read coverage (Fig. 5). Despite those utility differences, the performance results presented here demonstrate that it is currently not possible to replace BRANCH's main functionality with a reference-based splice variant assembly tool, even when an 'idealized' reference gene set is available as in the Cufflinks result given in Table 1a.

3.2.4 Assemblies with strand-specific rna-seq data and custom genome contigs Table 1b gives the results for *S.cerevisiae* where strand-specific RNA-Seq data was used along with 1360 custom assembled guide contigs representing 92.4% of this genome. With this dataset, BRANCH improves the sensitivity and precision of the transfrags generated by Velvet/Oases by 56.7 and 1.2%, and those from Trinity by 30.8 and 0.3%, respectively. At the same time, the numbers of complete transcripts, complete genes, covered transcripts and proteins annotated in the *S.cerevisiae* genome increase with BRANCH relative to the input data from both *de novo* assemblers by 41.1–74.1%, 41.1–74.1%, 29.9–60.6% and 8.3–33.8%, respectively. In general, the improvements achieved by BRANCH are more pronounced for the Velvet/Oases input because Trinity performs better on this dataset, leaving less room for improvements. Nonetheless, the results for both *de novo* assemblers demonstrate that BRANCH post-processing can lead to considerable

improvements of transfrags generated from strand-specific RNA-Seq data. This is even the case for a unicellular eukaryotic organism like *S.cerevisiae* where the risk of assembling chimeric transfrags is elevated compared with the other organisms chosen in Table 1, mainly owing to the much higher gene density and frequency of overlapping genes in its genome. Because chimeric events negatively impact the precision performance—a metric BRANCH improves—their frequency is likely to be lower in the transfrags post-processed by BRANCH than the ones from the upstream *de novo* assemblers. It is important to point out here that the current version of BRANCH does not detect or correct chimeric transfrags generated by the *de novo* assemblers. However, future improvements to our software will include such a feature.

3.2.5 Assemblies assisted with a related genome The sequencing and assembly of a genomic guide sequence can be avoided if a genome from a closely related organism is available, which is an important use case of BRANCH. Table 1c gives the test results for such a situation where the genes from rat served as guide sequence for improving the assembly of RNA-Seq data from mouse. In this dataset, the sensitivity and precision improves with BRANCH post-processing for Velvet/Oases by 4.4 and 1.1%, and for Trinity by 7.0 and 1.8%, respectively. The other test parameters also show noticeable improvements. The numbers of complete transcripts, complete genes, covered transcripts and proteins annotated in the mouse genome increase by 5.2–8.0%, 6.3–9.6%, 7.0–10.2% and 4.0–6.2%, respectively. Overall, the improvements with a closely related genome are slightly less pronounced than with guide contigs from the same organism. This is expected, as heterologous sequences represent a more challenging situation where it is important to perform the read and transfrag mapping against the related genome sequences with stringent enough mapping parameters to minimize the formation of false positive extension and fusion events of transfrags. When relaxing these parameters, one can increase the number of extension events, but often this will result in a decreased precision.

In summary, the aforementioned test results demonstrate BRANCH's efficiency in improving the representation of full-length transcripts in *de novo* assemblies by taking advantage of genomic guide sequence information from the same or a closely related organism.

4 CONCLUSIONS AND FUTURE WORK

This study introduces BRANCH as an efficient reference assisted post-processing method for enhancing *de novo* transcriptome assemblies. It can be used in combination with most *de novo* transcriptome assembly software tools. The assembly improvements are achieved with help from partial or complete genomic sequence information. They can be obtained by sequencing and assembling a genomic DNA sample in addition to the RNA samples required for a transcriptome assembly project. This approach is practical because it requires only preliminary genome assembly results in form of contigs. Nowadays, the latter can be generated with very reasonable cost and time investments. In case the genome sequence of a closely related organism is available, one can skip the genome assembly step and use the related gene sequences instead. This type of reference assisted assembly approach provides many attractive opportunities for improving *de novo* NGS assemblies in the future by making use of the rapidly growing number of reference genome information available to us.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the core facilities at the Institute for Integrative Genome Biology (IIGB) at UC Riverside.

Funding: Grants from the USDA National Institute for Food and Agriculture [NIFA-2010-65106-20675 to T.G.] and the National Science Foundation [ABI-0957099 to T.G., IOB-0420152 to T.G., MCB-1021969 to T.G., IIS-0711129 to T.J.].

Conflict of Interest: none declared.

REFERENCES

Au,K. *et al.* (2010) Detection of splice junctions from paired-end RNA-seq data by splicemap. *Nucleic Acids Res.*, **38**, 4570–4578.

Butler,J. *et al.* (2008) Allpaths: de novo assembly of whole-genome shotgun microreads. *Genome Res.*, **18**, 810–820.

Dezso,B. *et al.* (2011) Lemon—an open source C++ graph template library. *Electron. Notes Theor. Comput. Sci.*, **264**, 23–45.

Feng,J. *et al.* (2010) Inference of isoforms from short sequence reads. In: *Research in Computational Molecular Biology*. Springer, pp. 138–157.

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

Grant,G. *et al.* (2011) Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, **27**, 2518–2528.

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

Hillier,L.W. *et al.* (2009) Massively parallel sequencing of the polyadenylated transcriptome of C. elegans. *Genome Res.*, **19**, 657–666.

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

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Levin,J.Z. *et al.* (2010) Comprehensive comparative analysis of strand-specific RNA sequencing methods. *Nat. Methods*, **7**, 709–715.

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

Li,W. *et al.* (2011) IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. In: *Research in Computational Molecular Biology*. Springer, pp. 168–188.

Martin,J. and Wang,Z. (2011) Next-generation transcriptome assembly. *Nat. Rev. Genet.*, **12**, 671–682.

Peng,Y. *et al.* (2010) IDBA–a practical iterative de Bruijn graph de novo assembler. In: *Research in Computational Molecular Biology*. Springer, pp. 426–440.

Peng,Y. *et al.* (2011) T-IDBA: a de novo iterative de Bruijn graph assembler for transcriptome. In: *Research in Computational Molecular Biology*. Springer, pp. 337–338.

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

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

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

Trapnell,C. *et al.* (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, **25**, 1105–1111.

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

Wang,E. *et al.* (2008) Alternative isoform regulation in human tissue transcriptomes. *Nature*, **456**, 470–476.

Wang,K. *et al.* (2010) MapSplice: accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Res.*, **38**, e178.

Weber,K.P. *et al.* (2010) Whole genome sequencing highlights genetic changes associated with laboratory domestication of C. elegans. *PLoS One*, **5**, e13922.

Wu,T. and Watanabe,C. (2005) GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, **21**, 1859.

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

Zhao,Q. *et al.* (2011) Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study. *BMC Bioinformatics*, **12(Suppl. 14)**, S2.