BRANCH: boosting RNA-Seq assemblies with partial or related genomic sequences
Ergude Bao 1, Tao Jiang 1 and Thomas Girke 2*
1Department of Computer Science and Engineering, 2Department of Botany and Plant Sciences, University of California, Riverside, California 92521

Associate Editor: Prof. Ivo Hofacker

ABSTRACT
Motivation: De novo transcriptome assemblies of RNA-Seq data are important for genomics applications of unsequenced organisms. Due 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 utilizing 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 (MMPCP). In performance tests on real data from C. elegans and S. 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 8.3-33.8% 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

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 (Zerbino and Birney, 2008; Martin and Wang, 2011). The main factors driving this development are reduced NGS run costs, and improvements of sequence volume and read length. While 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: (1) abundance differences of RNA sequences make it difficult to obtain RNA samples representing most of the expressed genes of an organism and (2) 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), while the RNA reads can be assembled with de novo transcriptome assemblers like Velvet/Oases (Zerbino and Birney, 2008; Schulz et al., 2012), 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 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 since one

* to whom correspondence should be addressed
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 data set.

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, etc. due 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; Trapnell et al., 2010; Guttman et al., 2010; Feng et al., 2010; Li et al., 2011), 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 below; 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 due 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 (DAG) 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 (Trapnell et al., 2009; Langmead 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 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 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 paired-end) against the transfragts, and then the transfragts 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 transfragts sequences, while others may have failed to align due to base calling errors. After aligning the transfragts 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 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 due 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 transfragts 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 are additional exonic sequences not contained in the initial transfragts 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, transfragts and contigs as described above.

**Step 2** identifies an exon region based on the alignment, where we denote the coverage of a contig position \( i \) by \( \text{cov}(i) \). 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 \( \sum_{i \leq j \leq r} \frac{\text{cov}(i)}{r - l + 1} \geq b \), 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
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 R0 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 ≥ a and \( \sum_{i \in C} \text{cov}(i) \geq b \)
3. Record the upstream and downstream junction coverage for each base at contig position i, split a region [l, r] at bases i and i + 1 (1 ≤ i < r), if ujc(i) ≥ c or djc(i + 1) ≥ 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 [l1, r1] and [l2, r2], if con(l1, r2) ≥ d

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 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 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 t3, djc(t3), is the number of junction reads in rectangle Y of Figure 2a covering t3. The aligned junction reads overlap with bases t1 and t3, where t1 + 1 < t3. The algorithm records the upstream and downstream junction coverages for each contig base, and then splits such a region [l, r] at positions i and i + 1 (1 ≤ i < r), if the upstream junction coverage at i satisfies ujc(i) ≥ c or the downstream junction coverage at i + 1 satisfies djc(i + 1) ≥ 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 t2 and t3 is the number of reads in rectangle X with matching bases at positions t2 and t3. The algorithm computes the connectivity for each pair of exons and identifies two exons at positions [l1, r1] and [l2, r2], if con(l1, r2) ≥ d, where d is the minimum connectivity requirement to connect two exons.

2.4 Transfrag Extension Algorithm

The Transfrag Extension (TE) Algorithm extends and often joins de novo transfrags based on the additional exon sequences andsplice 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 \( \epsilon \) is connected with an existing exon \( \epsilon' \), then the algorithm to decide if \( \epsilon \) is connected with \( \epsilon' \) and/or \( \epsilon'' \). 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, while 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 DAG, 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 \( \epsilon \) and the connecting edges are splice junctions among exons. Two nodes \( v \) and \( v' \) are connected by an edge \( e(\epsilon, \epsilon') \) if their corresponding exons \( \epsilon \) and \( \epsilon' \) 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 \( \epsilon \) and (thus node \( v \) in the junction graph) is defined by:

\[
\psi_v = \frac{\text{number of compatible reads overlapping with exon } \epsilon}{\text{number of reads overlapping with exon } \epsilon \times \text{length of exon } \epsilon}.
\]

In the above formula, the overlap and compatibility of an aligned RNA read \( \gamma \) and an exon \( \epsilon \) are defined as follows. Read \( \gamma \) and exon \( \epsilon \) overlap if and only if their start coordinates overlap, and the reference genome satisfy \( l(\gamma) \leq l(\epsilon) \) and \( r(\gamma) \geq l(\epsilon) \), \( l(\epsilon) \leq r(\gamma) \), and \( r(\epsilon) \geq l(\gamma) \). Overlapped read \( \gamma \) and exon \( \epsilon \) are compatible if and only if any gap \( i(\epsilon), j(\epsilon) \) in the alignment of \( \gamma \) does not overlap with the exon \( \epsilon \).

The value for any pair \( \epsilon \) and \( \epsilon' \) (and thus edge \( e(\epsilon, \epsilon') \)) is defined as the absolute difference of their weights with amplification:

\[
w(\epsilon, \epsilon') = -\log(1 - |\psi_v - \psi_{v'}|)\]

The smaller \( w \) is, the more likely that the \( \epsilon \) and \( \epsilon' \) are from the same transcript.

Clearly, each given transfrag corresponds to a path in \( G \). These are called given paths. Since 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 \( \epsilon \) and \( \epsilon' \) 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 converting each given path \( p \in P \) to a node \( v(p) \) and (2) maintaining the connection between any two nodes \( v \) and \( v' \) if by a subpath of \( p \) by introducing an edge \( e(v, v') \). The newly node \( v(p) \) will be added 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 2c.

Then we invoke a Combinatorial Optimization (CO) Algorithm for solving the Minimum weight Minimum Path Cover (MMPCP) problem in the new graph \( G' \) (see Supplementary Materials). If \( P' \) is the resulting MMPCP 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 finding the MMPCP problem as new given paths and extend 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.

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 paired-end), 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 hours 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. While 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 data sets 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 due 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 Data Sets and Tests. The simulated test data sets were randomly sampled from the genome and transcriptome sequences of C. 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: (1) contigs of variable length of 1, 10, 50 and 100 kb; (2) contigs with variable coverage of the C. elegans genome of 40, 60, 80 and 100%; (3) contigs with variable sequence error rates of 0, 1, 2 and 3% by substituting bases at random positions; (4) different numbers (10, 30, 50 and 70 million) of paired-end RNA reads of 2x 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 Table ??); and (5) 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 Tables ?? to ?? 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 to 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 about two fold higher when the coverage of the genome by contigs is raised from 40% to 100% (Figure 3c and Table ??), whereas increasing sequence error rates from 0-3% in the contigs have a less pronounced impact by reducing the relative sensitivity improvements in the most extreme cases by 20-34% (Figure 3b and Table ??). Because BRANCH also identifies novel transfrags, the initial number of transfrags increases as expected (in these tests by 6.7-72.6%). The 487-5,394 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...

---

**Algorithm 2 Transfrag Extension: TE(G, P)***

Assign weights to the edges of G using Equation (2) for each given path \( p \in P \) 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 \( v' \) that do not belong to \( p \) do

if there is a path \( p' \) from \( v \) to \( q \) in \( P \) 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 \( G' \)

if \( P' = \text{CO}(G') \) then return the resultant MMPCP \( P' \)

else return \( \text{TE}(G', P') \)

end if

---
a fundamentally different method compared to the above assembled to contigs with Velvet using the VelvetOptimiser tool settings recommended by its developers were used. DNA reads were obtained by aligning the RNA reads with Tophat (Version 2) against this method, were available. The splice junction information was only situation where the minimal input data types, required for known genome sequences as guide reference, because it was the assemblers. Cufflinks was only included in the test case with the 2011; Zhao et al. organisms (C. elegans on the performance of BRANCH, was tested by comparing the genome sequence from a closely related organism (Table 1). The tests custom genomic contigs assembled from NGS reads of the 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 data sets 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 (S. cerevisiae) with a densely organized genome and rare alternative splicing. To evaluate the sensitivity of reference transcripts from 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 (SRA). 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 (>50bp) and paired-end 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 data sets 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 (S. 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 to 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 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 10−5) against the protein databases of the corresponding organisms. The obtained results were queried for nearly complete protein matches requiring ≥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 2x 55-76 bp long paired-end reads (accessions: SRR066623, SRR066625; Weber et al. 2010) and its RNA-Seq set contained 72 million 2x 100 bp paired-end reads (accession: SRR316929; Hillier et al. 2009). The second sample set is from mouse (Mus musculus) with 34 million 2x 76 bp paired-end 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 2x 76 bp long paired-end genomic reads (accessions: SRR527545, SRR527546; unpublished), and 10 million 2x 76 bp strand-specific paired-end RNA-Seq reads (accession: SRR59177; 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 to 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 reference transcripts from 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. 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 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 10−5) against the protein databases of the corresponding organisms. The obtained results were queried for nearly complete protein matches requiring ≥95% identity on the protein sequence level.
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 unannotated 5' and 3' regions that make the full-length cutoff criteria (≥95% sequence identity over ≥95% of the reference length) more stringent for transcripts than for proteins. BRANCH processing extends 5,726 transfrags from the initial assembly and it identifies many novel transfrags resulting in a 12.9% increase of the total transfrag pool. When these 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 5,320 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 to the ones obtained from Velvet/Oases. With a total of 7,696 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%, 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, due 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 (Figure 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 results 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 1,360 custom assembled guide contigs representing 92.4% of this genome. With

<table>
<thead>
<tr>
<th>Method</th>
<th>Sensitivity</th>
<th>Precision</th>
<th>N Transfrags</th>
<th>N Complete transcripts</th>
<th>N Complete genes</th>
<th>N Covered transcripts</th>
<th>N Exons</th>
<th>N Extensions</th>
<th>N Proteins</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) RNA-Seq Assembly of <em>C. elegans</em> (BRANCH Guided by Genomic Sequences from <em>C. elegans</em>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velvet/Oases (VO)</td>
<td>5.015</td>
<td>32.8%</td>
<td>55,083</td>
<td>3,248</td>
<td>2,986</td>
<td>3,844</td>
<td>96,078</td>
<td>-</td>
<td>3,839</td>
</tr>
<tr>
<td>Velvet/Oases+BRANCH (VOB)</td>
<td>5.332</td>
<td>43.3%</td>
<td>62,201</td>
<td>3,446</td>
<td>3,159</td>
<td>4,187</td>
<td>107,467</td>
<td>5,726</td>
<td>4,683</td>
</tr>
<tr>
<td>Velvet/Oases+BRANCHg (VOBg)</td>
<td>6.602</td>
<td>42.3%</td>
<td>56,369</td>
<td>3,441</td>
<td>3,173</td>
<td>4,283</td>
<td>107,876</td>
<td>7,696</td>
<td>4,811</td>
</tr>
<tr>
<td>Trinity (T)</td>
<td>5.048</td>
<td>39.4%</td>
<td>32,083</td>
<td>3,708</td>
<td>3,416</td>
<td>4,152</td>
<td>116,128</td>
<td>-</td>
<td>4,866</td>
</tr>
<tr>
<td>Trinity+BRANCH (TB)</td>
<td>5.303</td>
<td>43.3%</td>
<td>51,997</td>
<td>3,848</td>
<td>3,539</td>
<td>4,360</td>
<td>121,484</td>
<td>5,320</td>
<td>5,706</td>
</tr>
<tr>
<td>Trinity+BRANCHg (TBg)</td>
<td>6.309</td>
<td>42.3%</td>
<td>49,197</td>
<td>4,080</td>
<td>3,850</td>
<td>4,852</td>
<td>122,345</td>
<td>6,877</td>
<td>5,892</td>
</tr>
<tr>
<td>Cufflinksg (Cg)</td>
<td>5.147</td>
<td>47.7%</td>
<td>14,685</td>
<td>3,300</td>
<td>3,073</td>
<td>3,436</td>
<td>114,029</td>
<td>-</td>
<td>2,997</td>
</tr>
<tr>
<td>(b) Strand-Specific RNA-Seq Assembly of <em>S. cerevisiae</em> (BRANCH Guided by Genomic Sequences from <em>S. cerevisiae</em>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velvet/Oases (VO)</td>
<td>282</td>
<td>38.6%</td>
<td>75,053</td>
<td>54</td>
<td>54</td>
<td>132</td>
<td>1,211</td>
<td>-</td>
<td>926</td>
</tr>
<tr>
<td>Velvet/Oases+BRANCH (VOB)</td>
<td>442</td>
<td>39.8%</td>
<td>80,831</td>
<td>94</td>
<td>94</td>
<td>212</td>
<td>1,875</td>
<td>9,514</td>
<td>1,239</td>
</tr>
<tr>
<td>Trinity (T)</td>
<td>315</td>
<td>41.0%</td>
<td>11,451</td>
<td>146</td>
<td>146</td>
<td>201</td>
<td>4,375</td>
<td>-</td>
<td>1,957</td>
</tr>
<tr>
<td>Trinity+BRANCH (TB)</td>
<td>412</td>
<td>41.3%</td>
<td>13,394</td>
<td>206</td>
<td>206</td>
<td>261</td>
<td>4,498</td>
<td>2,322</td>
<td>2,119</td>
</tr>
<tr>
<td>(c) RNA-Seq Assembly of Mouse (BRANCH Guided by Genomic Sequences from Rat)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Velvet/Oases (VO)</td>
<td>7,103</td>
<td>23.4%</td>
<td>447,689</td>
<td>2,922</td>
<td>2,331</td>
<td>5,230</td>
<td>123,070</td>
<td>-</td>
<td>12,260</td>
</tr>
<tr>
<td>Velvet/Oases+BRANCH (VOB)</td>
<td>7,417</td>
<td>24.5%</td>
<td>518,360</td>
<td>3,073</td>
<td>2,478</td>
<td>5,595</td>
<td>123,939</td>
<td>3,325</td>
<td>12,747</td>
</tr>
<tr>
<td>Trinity (T)</td>
<td>4,593</td>
<td>25.4%</td>
<td>143,757</td>
<td>3,300</td>
<td>3,073</td>
<td>3,436</td>
<td>114,029</td>
<td>-</td>
<td>8,677</td>
</tr>
<tr>
<td>Trinity+BRANCH (TB)</td>
<td>4,916</td>
<td>27.2%</td>
<td>187,478</td>
<td>3,176</td>
<td>3,073</td>
<td>3,430</td>
<td>101,964</td>
<td>1,295</td>
<td>9,217</td>
</tr>
</tbody>
</table>

Table 1. Performance on real data. 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 Trinity+BRANCH) using under (a) custom assembled genome contigs or known gene sequences or from *C. elegans*, and under (c) the gene sequences from the rat genome. The latter evaluates BRANCH's performance for a case where a closely related guide genome sequence is available. The sample from *S. cerevisiae* 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.
Fig. 4: Transcript length coverage. The number of reference transcripts of the *C. elegans* data set is plotted that aligned with the transfrags over increasing overlap thresholds from \( \geq 10\% \) to \( \geq 90\% \). The acronyms assigned to the different methods in the legend are defined in the first column of Table 1.

this data set, 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 data set, 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 to the other organisms chosen in Table 1, mainly due 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 data set 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 since 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 in order 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 above 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.

ACKNOWLEDGEMENT

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

Funding: This work was supported by 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.].

REFERENCES


