Improving transcriptome reconstruction by connecting genomic contigs through RNA-seq read alignments

Raúl Wijfjes

Abstract

Current sequencing projects mainly produce draft genomes. Due to assembly errors and fragmentation of contigs, these genomes are not well-suited as a reference for genome-guided transcriptome assembly method. In this study, I present the Glue algorithm which assembles genomes by making use of both genomic and transcriptome datasets. Based on alignments of transcriptome data to genomic contigs, Glue merges contigs likely to contain fragments of the same gene into a single sequence. Tests on *A. thaliana* datasets indicated that limitations of currently available RNA-seq alignment tools prevent Glue from performing optimally. Despite these limitations however, transcripts reconstructed by using a genome-guided transcriptome assembly method contained a higher percentage of full-length transcripts after using Glue on the draft genomic assembly which was used as a reference. This result shows that applying Glue to a draft genome makes the assembly more suited as a reference for transcriptome assembly methods.

Introduction

In recent years, gene discovery methodology has been revolutionized by RNA-seq, a high-throughput sequencing method which generates a large amount of transcriptome data at single-base resolution and over a large range of expression levels[1]. Applications of RNA-seq datasets include annotating a newly sequenced genome[2] or measuring gene expression between organisms under different conditions[3]. For these kind of studies, it is crucial to obtain a high quality transcriptome. Transcripts can be reconstructed out of transcriptome reads via two different approaches. Reference-based assembly methods leverage alignments of RNA-seq reads to a reference genome produced by a splice-aware read aligner such as Tophat2[4] to reconstruct full transcripts. *De novo* methods directly assemble transcripts out of RNA-seq reads without using a reference genome.

Reconstructing transcripts via the latter approach is very challenging, due to the large amount of alternative splicing in higher eukaryotes[5]. Consider a gene which has eight exons and five isoforms (Figure 1a). As short transcriptome reads of 100-150 bp usually do not cover more than two exons, a RNA-seq dataset can often only link two neighbouring exons to one another (Figure 1b). As some of the exons are shared between the isoforms, it is very difficult to reproduce the correct isoforms *de novo*. Therefore, genome-guided transcriptome assembly methods are generally more accurate than *de novo* methods.

However, the results of reference-based assembly methods are very dependent on the quality of the used reference genome. Nowadays, most sequencing projects produce draft assemblies which never reach a finished state[6]. These assemblies consist out of a large number of short contigs. As a result, single genes may be split between multiple contigs. To illustrate the magnitude of this problem, 26% of the gene families of an annotated genome assembly of chimpanzee had a different number of genes compared to the initial draft assembly[7]. In about 71% of these families, this difference was caused by genes being split between different contigs in the draft assembly, which led to them being seen as two separate genes. Clearly, draft genomes are not well-suited to be used as a reference for genome-guided transcriptome assembly methods and the transcriptomes produced from these genomes can skew the results of gene expression studies.

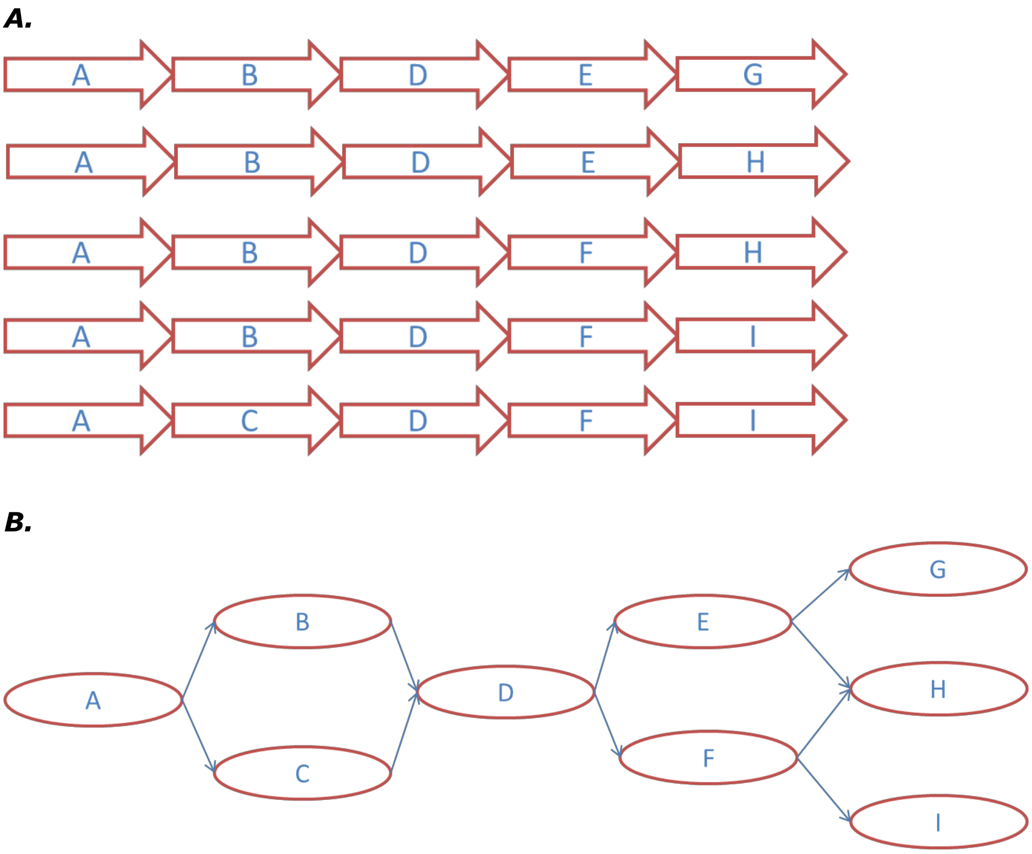

Figure 1. Alternative splicing makes de novo transcriptome assembly challenging. *(A) Five isoforms of a gene having eight different exons, depicted by the letters A-I. This example was described in an earlier paper[8]. (B) The graph depicting the isoforms described in A. Edges are drawn between exons which are linked through RNA-seq reads. Several isoforms can be reconstructed from this graph. However, it is unknown which of these isoforms are real by looking at the graph on its own. For example, all isoforms including CDE are false.*

Here, I present a novel proof-of-concept algorithm which attempts to produce a genomic assembly which is better suited for genome-guided transcriptome assembly methods than the draft genomes produced by *de novo* genomic assemblers. This algorithm, named Glue, makes use of RNA-seq reads mapped to a genomic assembly to join contigs which contain parts of the same gene into a single sequence. The idea of scaffolding genomic contigs by using RNA-seq data is not new[9,10]. However, previous methods joined two contigs based on paired-end RNA-seq reads of which one mate mapped to the first contig and the other mate to the second contig. The novelty of Glue lies in the fact that it is the only algorithm to my knowledge which scaffolds contigs by using RNA-seq reads which map across exon-intron boundaries. If the intronic sequence could not be properly assembled, these reads may map partly to two different contigs, which can be joined into a single sequence as a result. Furthermore, it does not only identify gaps between contigs during the scaffolding process, but also tries to fill them by making use of the connections between the contigs.

Yet, inaccuracies which are present in the contigs produced by a *de novo* genomic assembler could skew the genomic assemblies produced by Glue. Most genomic assemblers make use of a de Bruijn graph structure during the assembly process, extracting contigs from the graph by using heuristics. As a result of these heuristics, the resulting set of contigs could contain errors. I tested whether using a the de Bruijn graph representation of the genomic dataset as input to Glue instead of a set of contigs helps to prevent these errors. A de Bruijn graph retains all uncertainties of an assembly without taking ambiguous choices. Therefore, a de Bruijn graph should be a more objective representation of a genome assembly than a set of contigs. A downside of using a de Bruijn graph is that the sequences contained in the graph are more fragmented than a set of contigs. However, this problem can be resolved by connecting these sequences with Glue. The feasibility of using a de Bruijn graph was proven by the TAG[11] algorithm, which maps metatranscriptomic reads to a de Bruijn graph representation of a metagenome and traverses the graph to reconstruct the transcripts.

Glue was evaluated by checking whether the set of transcripts found after performing a genome-guided assembly with the contigs produced by Glue as a reference contains a higher percentage of complete transcripts compared to the set of transcript reconstructed after using a draft assembly produced by a *de novo* genomic assembler as a reference. Application of Glue on *A. thaliana* datasets revealed that it does not perform optimally due to the fact that currently available RNA-seq tools are not optimized for mapping RNA-seq reads partly to one genomic contig and partly to another. As a result, the use of a de Bruijn graph representation of a genomic dataset as input for Glue is limited. Despite these limitations, applying Glue to a set of genomic contigs leads to a larger percentage of full-length transcripts being found by genome-guided transcriptome assembly methods than before the application. Therefore, it can be concluded that joining contigs based on RNA-seq reads mapping across exon-intron boundaries is a useful method of making a draft genome assembly which is more suitable as a reference for genome-guided assembly methods.

Methods

Extracting genomic sequences from a de Bruijn graph

In a de Bruijn graph representation of a genomic dataset, nodes are subsequences of the reads of length *k*, known as *k*-mers, which are connected by edges if they share a *k - 1* suffix-prefix overlap (Figure 2a). Simple paths in the graph are paths in which all internal nodes have an in- and outdegree of one, the start node has an out-degree of one and the end node has an in-degree of one. These simple paths can be collapsed into a single node depicting a sequence which is called a unitig. A compacted de Bruijn graph can be built from these unitigs in which nodes are no longer *k*-mers but unitigs, while the edges are still based on *k - 1* overlaps (Figure 2b).

To map RNA-seq reads to a de Bruijn graph, it is required to extract the unitigs of the graph as existing RNA-seq mapping tools only work with linear representations of a genome, such as contigs. Two algorithms are used to obtain the genomic unitigs. First, all of the *k*-mers of the genomic dataset are counted using KMC2 (version 2.2.0[12]), which was chosen due to its high speed and frugal memory use. Next, all *k*-mers with a frequency below a set threshold (default: 4) are removed, as it is assumed that they correspond to sequencing errors. Unitigs are produced from the remaining *k*-mers by using BCALM (GitHub commit 59a346c[13]), a tool which can produce unitigs in low memory without explicitly drawing a de Bruijn graph. A custom Python script (BCALM_to_FASTA.py) converts the unitigs produced by BCALM into FASTA format, which makes them suitable to be used as input for RNA-seq mapping tools.

Aligning RNA-seq reads to genomic contigs

RNA-seq reads are aligned to the genomic contigs by using GSNAP (version 2014-12-29[14]) with parameters -m 5 -N 1 -E 0. This particular aligner was chosen, due to the fact that it is the only alignment tool available to my knowledge which is able to produce so called split alignments of a RNA-seq read to pairs of contigs within reasonable time (Figure 3). With these alignments, RNA-seq reads map partly to one contig and partly to another contig. Split alignments result from RNA-seq reads mapping across exon-intron boundaries of which the intronic region could not be properly assembled, because it contained for instance repeats. It is important that these alignments can be performed, as they will be used later on by the Glue algorithm to merge contigs into a single sequence. STAR[15] and Segemehl[16] were considered as alternative mappers, but were dropped after showing excessive running times when applied to the datasets included in this study.

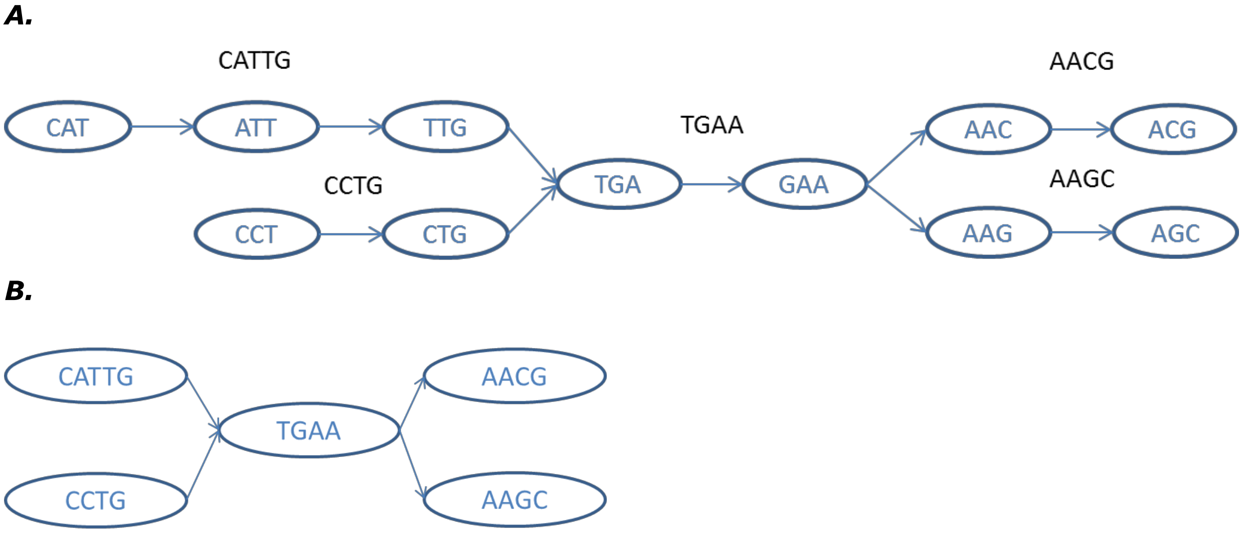

Figure 2. Examples of a De Bruijn graph and a compacted de Bruijn graph *(A) A de Bruijn graph constructed from short read data using a value of k = 3. Nodes are connected by edges if they share a suffix-prefix overlap of k -1 nucleotides. Simple paths can be collapsed into unitigs, which are shown in black. (B) The compressed de Bruijn graph created from A.*

Identifying collapsed repeats

Collapsed repeats pose a problem to the Glue algorithm. Consider an example where Glue merges three contigs into a single sequence, with the middle contig corresponding to a collapsed repeat. Because Glue lacks any information about the copy number of the collapsed repeat in the genome, one cannot accurately determine how many copies of the collapsed repeat should be included in the sequence. Therefore, Glue excludes collapsed repeats from the merging procedure.

To identify contigs which correspond to collapsed repeats, the Myers' A-statistic of all contigs is computed. This metric is an approximation of the log-odds ratio between the probability of the contig being unique and the probability of the contig corresponding to a collapsed repeat[17]. To calculate the A-statistics, the genomic reads are remapped to the contigs using Bowtie2 (version 2.1.0[18]) with parameters -I 0 -X 500. After mapping, a module of SGA (version 0.10.13[19]), named sga-astat.py, is used with default parameters to produce a file containing the A-statistic of each contig.

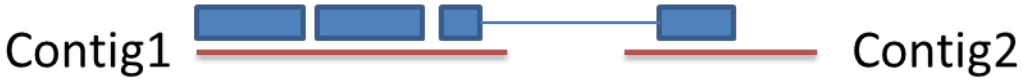

Figure 3. Example of a split alignment of a RNA-seq read to a pair of contigs. *With a split alignment, one part of a read maps partly to one contig and the other part maps to another contig. Blue rectangles correspond to aligned RNA-seq reads.*

The Glue algorithm

Glue is implemented in Python 3.4 and takes three different files as input: a FASTA file containing genomic contigs, a SAM file containing alignments of RNA-seq reads to these contigs produced by GSNAP and a file produced by sga-astat.py which contains the Myers' A-statistics of each contig (Figure 4). All input files can be produced as described above. Alternatively, the FASTA file containing genome contigs can be generated by using a *de novo* genomic assembly method. The output of Glue is a FASTA file containing an updated version of the genomic contigs, in which all contigs containing parts of the same transcripts are merged into single sequences as much as possible.

First, Glue leverages the split alignments contained in the provided SAM file to identify pairs of contigs which potentially contain parts of the same transcript. As an example of how such an alignment appears in a SAM file produced by GSNAP, consider a read of 100 bp of which 51 bp maps to contig 1 and 49 bp maps to contig 2. This alignment will be represented by two records. The first record shows 51 bp of the read matching contig 1, while the second record shows 49 bp of the read mapping to contig 2. In both records, the part which does not align to the contig is hard-clipped. Therefore, finding split alignments in the SAM file is a matter of finding all pairs of alignment lines which belong to the same read and contain hard clipping in their cigar string (H). The alignments are saved in the form of a Python dictionary containing tuples of two contigs as keys and the amount of reads which are split aligned to these contigs as values.

Next, Glue uses the Python dictionary produced from the SAM file to build scaffolds out of the contigs. While a scaffold usually refers to contigs connected using mate-pair genomic reads, split alignments of RNA-seq reads are used for scaffolding in this case. Scaffolds are saved in the form of adjacency graphs (Figure 5a). First, Glue identifies all unique contigs by using the file containing the A-statistics for each contig which was provided as input. These contigs form the nodes of the scaffold graphs. After identifying these contigs, the Python dictionary is used to connect all pairs of unique contigs which contain a split alignment of a RNA-seq read by an edge. These edges represent the gaps which are present between the single-copy contigs. They are undirected to account for reverse complements of contigs. The end result is a collection of disjoint and bidirected graphs, which will be dubbed transcript graphs from now on.

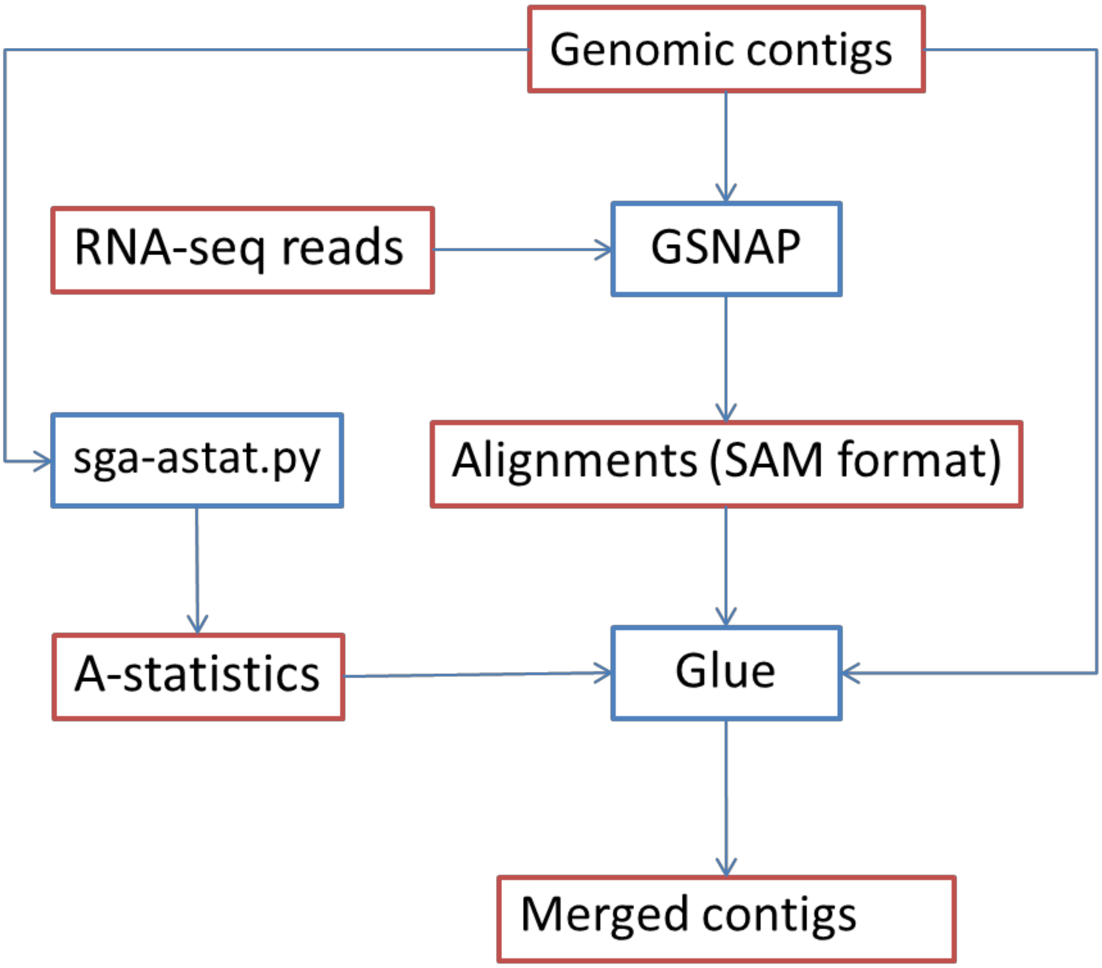

Figure 4. Overview of the input and output of the Glue algorithm. *Blue boxes correspond to computational algorithms and red boxes to files.*

After producing the scaffold graphs, Glue attempts to fill all gaps which are represented by the edges. Gaps are expected to consist mainly out of intronic sequences, but can also contain for example repeated parts of exons which could not be properly assembled. In order to determine these gap sequences, all contigs which may be part of this gap need to be identified. To this end, an adjacency graph is produced which contains both the single-copy and the repeated contigs as nodes (Figure 5b). Similar to de Bruijn graphs, directed edges are drawn between nodes based on suffix-prefix overlaps. The length of the overlaps depends on the overlap used by the *de novo* genomic assembly method to produce the contigs (e.g. *k-1* for assembly methods based on a de Bruijn graph structure). The produced graph is formally known as a string graph. As the contigs in the scaffold graphs are a subset of the contigs found in the string graph, the latter graph can be used to fill the gaps present in the scaffolds graphs.

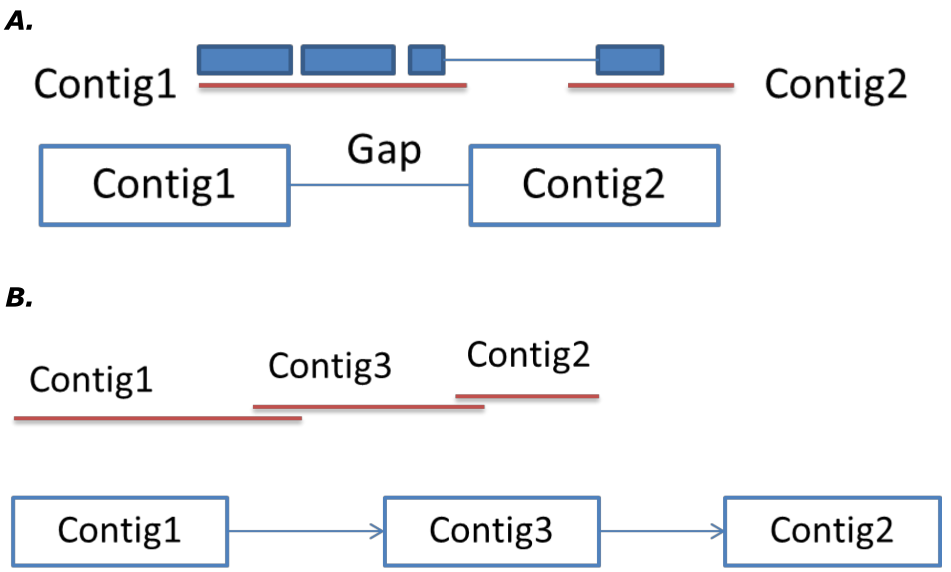

Figure 5. Examples of a scaffold graph and a string graph. *(A) A scaffold graph. If a RNA-seq read aligns partly to one contig and partly to another one, this pair of contig is connected by an undirected edge. In the end, a collection of graphs are produced, each graph representing an ordered set of genomic regions containing exons. These graphs are dubbed here as scaffold graphs. (B) A string graph. String graphs contain all single-copy contigs and repeated contigs as nodes. Nodes are connected by directed edges if the contigs share a suffix-prefix overlap.*

To fill the gaps in the scaffold graphs, Glue first locates the terminal nodes of each scaffold graph and searches for all paths connecting them. The longest path, which contains all of the contigs, is used in the following gap filling step. Consider a path of contigs in a scaffold graph which is represented as C_1, C_2, C_3, C_4. Starting from the first neighbouring pair of contigs, in this case C_1 and C_2, Glue determines whether there is a single path in the compressed de Bruijn graph which connects the pair. If a single path can be found, C_1 and C_2 are replaced by the string which is spelled out by this path in the string graph. If no path can be found, Glue investigates whether C_1 and C_2 share a suffix-prefix overlap of a certain minimum length (by default 25). If such an overlap exists, the contig sequences are merged based on this overlap. If it does not, the gap between the contigs cannot be resolved and the sequence between C_1 and C_2 will be replaced by 100 ambiguity (N) symbols which symbolize a gap of unknown length and composition. In the case that multiple paths are found between the contigs in the string graph, the sequence between C_1 and C_2 will be replaced by ambiguity symbols as well. In the end, Glue produces one genomic sequence per scaffold graph.

As a final step, Glue takes all sequences produced from the scaffold graphs and removes those which have a length below a certain threshold (default = 500 bp). This cut-off is used to prevent short genomic sequences which are likely to contain only partial transcripts from appearing in the final output.

Used datasets

To evaluate the performance of Glue and the feasibility of using a de Bruijn graph representation of a genomic dataset as input, two publicly available *A. thaliana* Col-0 datasets were used. In this study, only the reads associated with chromosome 1 were used, to facilitate fast running times of the used computational algorithms. The following datasets were used:

- ***A. thaliana* Col-0 genomic read dataset (SRX883065).** This dataset consists out of 100 bp paired-end Illumina genomic reads having an insert size of 200-500 bp. The total length of the dataset is 3038 Mbp. As the *A.thaliana* reference genome (TAIR10 assembly) has a golden path length of 119.15 Mb, the expected coverage of this dataset is 25x. To obtain all reads associated with chromosome 1, I mapped the full datasets to chromosome 1 of the reference genome by using BWA-mem (version 0.7.5a-r405[20]) with default settings, followed by removing all reads which did not show any hit.
- ***A. thaliana* Col-0 RNA-seq read dataset (SRX314612).** This dataset contains 101 bp single-end Illumina RNA-seq reads. It has a total length of 2508 Mbp. To obtain the reads associated with chromosome 1, I mapped the full dataset to chromosome 1 of the reference genome by using GSNAP[14] (parameters: -N 1 -m 2 -w 4000), followed by removing all reads which did not have any hit.

Reference genome and annotation

The reference genome and reference annotation of *A. thaliana* were both downloaded from the ftp server of TAIR using the following links:

- Reference genome:
  ftp://ftp.arabidopsis.org/home/tair/Sequences/whole_chromosomes
- Annotation:
  ftp://ftp.arabidopsis.org/home/tair/home/tair/Genes/TAIR10_genome_release/TAIR10_gff3/TAIR10_GFF3_genes.gff

Evaluation

To test whether using Glue on a set of genomic contigs leads to a higher quality transcriptome, I applied it to the genomic dataset and RNA-seq dataset of chromosome 1 of *A. thaliana* described above. First, contigs were produced from the genomic reads by using SOAPdenovo2 (version 2.04[21]) with a *k*-value of 33 (see Results for the motivation behind this choice). The RNA-seq reads were mapped to the produced contigs using GSNAP[14] after which Glue with default settings was used to merge contigs based on the resulting alignments. Transcripts were reconstructed by using Cufflinks (version 2.2.1[22]), after remapping the RNA-seq reads to the remaining merged contigs with Tophat2 (version 2.0.14[4]) with the parameter --max-intron-length set to 4000. The resulting set of transcripts was aligned to the *A. thaliana* reference genome to assess its quality.

A similar procedure was used to evaluate whether using unitigs of a de Bruijn graph instead of contigs produced by genomic assemblers as input to Glue led to a more accurate transcriptome in the end. Unitigs were produced from the genomic dataset of *A. thaliana* using a *k*-value of 33, after which transcripts were assembled as described above, providing the unitigs as input to Glue instead of the SOAPdenovo2 contigs produced earlier.

In addition, transcripts were produced and evaluated by using scaffolds produced by SOAPdenovo2 as a reference. These scaffolds are generated by making use of paired-end information. They were used to determine whether it could be more effective to scaffold contigs using paired-end data, before merging them using RNA-seq reads.

Furthermore, to provide a golden standard, a transcriptome was produced by using the TAIR10 assembly of *A. thaliana* as a reference. Lastly, the *de novo* transcriptome assembler Trinity (version 2.0.6[23]) was run with default settings to test the assumption that using a reference genome during transcriptome assembly improves the quality of the resulting set of transcripts.

As the main goal of Glue is to prevent transcripts from being fragmented during the transcriptome assembly process, I determined the percentage of complete transcripts which was found in the set of transcriptomes produced by each method. A transcript is defined as complete if it aligns with 95% identity to the reference genome*,* 90% of the length of the transcript aligns to a region which is annotated as mRNA and if the alignment covers 90% of the total length of the annotation. If an alignment fulfills the first two requirements but does not cover 90% of the annotation, the corresponding transcript is defined as a partial transcript. I distinguish between these two categories to reveal which fraction of the transcriptome produced by a certain method corresponds to full-length transcripts and which fragment to only parts of transcripts.

Moreover, to assess sensitivity, I calculated the fraction of expressed transcripts which were found by each method. The total amount of expressed transcripts which were contained in the RNA-seq dataset was determined by first aligning the dataset to the reference genome with Tophat2[4]. Next, transcripts were reconstructed and quantified by using Cufflinks[22]. Finally, the total amount of expressed transcripts was computed by counting all transcripts corresponding to a reference annotation which had a FPKM higher than or equal to 1. These transcripts were assumed to have come to expression under the condition during which the RNA-seq dataset was extracted. After determining the amount of expressed transcripts (4940), the sensitivity of each method was measured in terms of recall. Recall is defined as the number of complete transcripts found by a method divided by the total number of expressed transcripts.

To produce the three metrics described above for each method, reconstructed transcripts were aligned to the *A. thaliana* reference genome by using GMAP[24] with default settings, producing SAM output. The fillmd command of SAMtools (version 0.1.19[25]) was used in combination with a custom Python script (filter_md_bam_file.py) to remove alignments which had less than 95% identity. After filtering, the alignment files were intersected with the reference annotation of *A. thaliana* using the intersect command of the bedtools2 (version 2.17.0[26]) suite. Specifically, the parameters -wo -F 0.9 -s were used to produce a bed file containing all partial and complete transcripts and the parameters -wo -f 0.9 -r -s to produce a bed file containing only complete transcripts. The output bed files were parsed by the grep (parameters: -P "\tmRNA\t" and -P "Chr1\t") and wc (parameters: -l) unix command line tools to compute the percentage of complete transcripts, the percentage of partial + complete transcripts and the recall of a set of transcripts.

Results

Using Glue results into a higher number of reconstructed full-length transcripts

When comparing the transcriptomes resulting from all tested methods (Table 1), it can be seen that after applying Glue to the genomic contigs produced by SOAPdenovo2, the transcriptome produced by using these contigs as a reference in Cufflinks contained a higher percentage of complete transcripts (+6.2%). While slightly less annotated transcripts were discovered as a result, this loss is minimal (-0.7% recall). To determine whether the higher percentage of complete transcripts was not primarily caused by the 500 bp length cut-off, Glue was run without using any cut-off at all. While less pronounced (+3.9%), the improvement remained. These results show that merging contigs based on RNA-seq information leads to a draft genomic assembly which is more suitable for reference-guided transcriptome assembly methods, even without using a length cut-off.

The percentage of complete transcripts resulting from the genomic contigs connected by Glue is similar to the percentage achieved by using the scaffolds produced by SOAPdenovo2 as a reference. In contrast, the percentage of correctly reconstructed partial and complete transcripts together is slightly worse when using the scaffolds (-1.9 %), showing that scaffolding using paired-end information may lead to incorrectly including fragments from different transcripts in the same scaffold. In terms of recall, the scaffolds perform better than the contigs connected by Glue (+3.9%). These results suggest that it may be worth it to scaffold contigs after using Glue by using paired-end information, to maximize sensitivity. Additionally, Trinity performed the worst in all produced evaluation metrics, showing the added value of using a genomic reference during the transcriptome assembly process.

Limitations of GSNAP hamper the performance of Glue

For both sets of transcripts which were produced after using Glue on either contigs produced by SOAPdenovo2 or unitigs, a large difference can be observed between the percentage of reproduced partial + complete transcripts and the percentage of reconstructed complete transcripts (>35%). This result implicates that significantly more contigs containing transcript fragments could be merged into a single sequence by Glue than the amount done at the moment. The limited performance of Glue could be due to the fact that GSNAP may not be optimal for producing split alignments of RNA-seq reads to a fragmented draft genome assembly.

To test this hypothesis, I assessed how well GSNAP could map RNA-seq reads to the unitigs compared to the set of contigs. As described earlier in this paper, sequences stored in a de Bruijn graph are generally more fragmented than a set of contigs produced by a *de novo* genomic assembler, which is clearly illustrated by the difference in NG50 (Table 2). As both sets have the same level of genome coverage, the set of unitigs can essentially be seen as a more fragmented version of the set of contigs. If GSNAP would not be optimized for producing split alignments, it should map a lower percentage of RNA-seq reads to the unitigs than to the contigs produced by SOAPdenovo2, as a larger amount of RNA-seq read require a split alignment in a more fragmented assembly. This turned out to be indeed the case (Table 2).

The problems GSNAP has with split alignments are caused by its algorithmic design. For a split alignment of a RNA-seq read to a pair of genomic sequences, GSNAP requires anchors of at least 31 bp or more in each of the sequences. These anchors are harder to find in the more fragmented unitig assembly than in the SOAPdenovo2 assembly. Moreover, GSNAP is unable to map RNA-seq reads which span across more than two sequences. These reads are more likely to occur in a more fragmented assembly. The design limitations of GSNAP ultimately hamper the performance of Glue, as it depends on split alignments to merge contigs.

Table 1. Comparing the quality of the transcriptomes obtained using different methods of transcriptome assembly

Method	**Number of transcripts reported**	**Partial[#] + complete[$] transcripts (%)***	**Complete transcripts (%)***	**Recall[^] (%)**
SOAPdenovo2 contigs + Cufflinks	8109	72.8	32.6	53.5
SOAPdenovo2 contigs + Glue + Cufflinks	6711	75.6	38.8	52.7
SOAPdenovo2 contigs + Glue (no length cut-off) + Cufflinks	7114	74.4	36.5	52.5
SOAPdenovo2 scaffolds + Cufflinks	7656	73.7	38.7	60.0
Unitigs + Glue + Cufflinks	7262	76.5	29.5	44.0
Reference genome + Cufflinks	6091	77.2	64.0	78.9
Trinity	8612	48.9	19.0	33.2

#Partial transcript: Alignment to reference has 95% identity and 90% of the transcript aligns to a region annotated as mRNA

$Complete transcript: Alignment to reference has 95% identity, 90% of the transcript aligns to an annotated as mRNA, 90% of the length of the annotation is covered

^Recall: Amount of complete transcripts found / Total amount of expressed transcripts

**Percentages are calculated by dividing the amount of partial and complete transcripts (or the amount of complete transcripts) by the total amount of transcripts reported.*

Unitigs could potentially improve the performance of Glue

As a result of the limitations of GSNAP described above, Glue is able to merge less pairs of unitigs into a single sequence than pairs of SOAPdenovo2 contigs. While Glue is able to join 313 pairs of SOAPdenovo2 contigs based on split alignments of RNA-seq reads, only 279 pairs of unitigs could be merged by this method (Table 3). Naturally, the percentage of complete transcripts found in the transcriptome produced by using the unitigs is significantly worse (-9.3%) than the one produced by using the SOAPdenovo2 contigs (Table 1).

However, the set of transcripts resulting from the unitigs contains a slightly higher percentage of partial + complete transcripts compared to the set resulting from the contigs produced by SOAPdenovo2, nearly reaching the level achieved by using the *A. thaliana* reference genome. Moreover, a much larger fraction of the gaps in the scaffold graphs could be filled unambiguously when using unitigs instead of contigs (Table 3). This result shows that using a de Bruijn graph representation of a genomic assembly instead of a set of contigs could potentially improve the performance of Glue, if RNA-seq reads could be properly split aligned to the unitigs.

Table 2. Comparing unitigs to SOAPdenovo contigs

	Unitigs (k = 33)	SOAPdenovo contigs (k = 33)
Genome coverage# (%)	99.98	99.98
NG50$	3499	4876
RNA-seq reads mapped by GSNAP (%)	77.6	90.6

#Genome coverage: Percentage of bases in the reference genome which are covered by at least one contig or unitig. This metric was produced by mapping all sequences against the reference genome of A. thaliana using BWA-mem[20] with default settings. The produced SAM file was converted into pileup format using the mpileup command of SAMtools[25], after which the resulting pileup file was parsed by a custom Python script (check_genomic_coverage.py) to obtain the genome coverage.
$Computed by QUAST (version 2.3[27])

Table 3. Performance of Glue regarding SOAPdenovo2 contigs and unitigs

Assembly	**Merged pairs**	**Gaps filled unambiguously#**	**Gaps filled by inserting 100 N's**
SOAPdenovo2 contigs	313	15	298
Unitigs	279	100	179

#Gaps can be filled unambiguously by either using the string graph or by merging contigs based on suffix-prefix overlap

Effect of k size on gap filling step of Glue

To assess the effect of the used value of *k* on the performance of the gap filling step of Glue, I produced unitigs with different values of *k* and determined the amount of paths found between unitigs connected by RNA-seq reads in the compressed de Bruijn graph (Figure 6). At lower values of *k*, the amount of unitigs connected by zero paths decreases while the amount of unitigs connected by multiple paths increases. The opposite effect can be seen when increasing *k*. These observations are caused by the fact that the string graph produced be Glue is more tangled at lower k values, while being more fragmented at high k-values. To have Glue unambiguously fill gaps, unitigs should be connected by a single path as much as possible. By this metric, a k-value of 27 seems to be the most appropriate. However, the value of k should not be too small as well, as RNA-seq reads map more poorly to shorter unitigs. Therefore, a higher k-value of 33 was chosen in the end for the evaluation procedure described above, which still produced a reasonable number of unitigs connected by a single path. For a fair comparison, the same value of k was used for constructing the contigs by SOAPdenovo2.

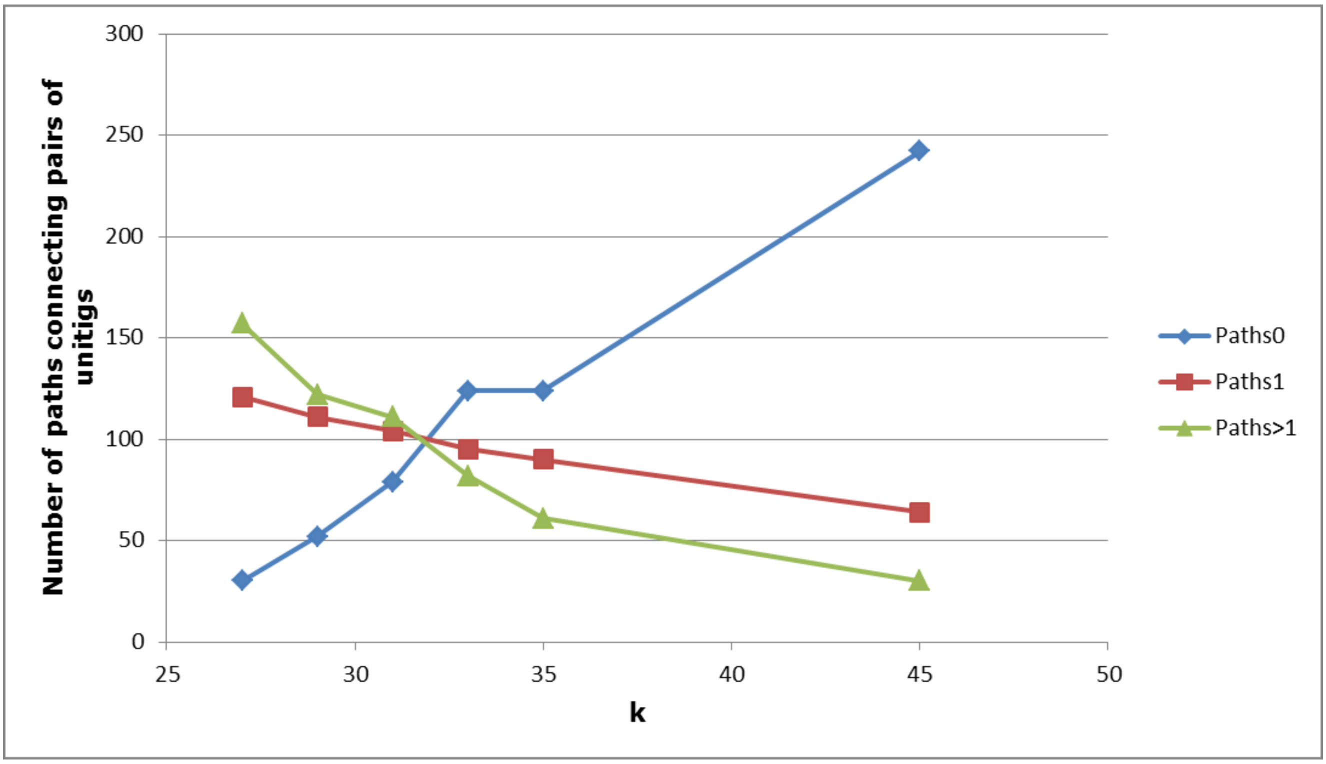

Figure 6. The effect of the chosen value of k for the amount of paths found between pairs of unitigs connected by RNA-seq reads. *As the k-mer size increases from 27 to 35, the number of pairs found which are connected by a single path or by multiple paths decreases while the amount of unitigs linked by zero paths increases. This trend continues at a larger k-value of 45.*

Discussion

In this paper, I presented a proof-of-concept algorithm called Glue which for the first time to my knowledge, uses alignments of RNA-seq reads mapping across exon-intron boundaries to merge contigs of a draft genome into a single sequence. After applying Glue to genomic contigs produced by SOAPdenovo2 from an *A. thaliana* dataset, the transcriptome reconstructed from these contigs by using a genome-guided assembly method contained a higher percentage of complete transcripts compared to before the merging. However, the percentage was below that of the set of transcripts obtained using the TAIR10 reference genome, so there is still plenty of room for improvement. Particularly, including mate-pair information of a genomic read dataset could greatly improve the performance of Glue. For instance, if genomic contigs in a string graph are connected by multiple paths, the information could serve as a constraint for determining which of these paths is correct. As a result, Glue would be able to fill more gaps unambiguously. Furthermore, mate-pair information could be used to determine the copy number of contigs corresponding to collapsed repeats, enabling Glue to include these contigs in the merging procedure.

While Glue was used here to prevent transcripts from being fragmented between different contigs, its applications can extend beyond this purpose. For instance, it could be used to scaffold genomic contigs after which a tool such as GapFiller[28] can fill the gaps captured in the scaffolds. Moreover, Glue could serve as a way of improving the contiguity of metagenomes, which are very fragmented by nature. However, one should be careful to not misinterpret the N's inserted inside scaffolds as being a gap of 100 nucleotides, as it can have any length in truth.

In this study, GSNAP was used to produce split alignments of RNA-seq reads to a set of genomic sequences, which is a far from ideal method of generating these alignments. Due to this limitation, the benefits of using a de Bruijn graph as input instead of a set of contigs produced by a genomic assembler for Glue are marginal as of now. Yet, the idea of using a de Bruijn graph as a reference should not be discarded yet. GSNAP is meant to align short RNA-seq reads to a set of linear contigs and therefore is not able to make use of the fact that you know the connections between unitigs in a de Bruijn graph. While several other tools exist which do make use of the de Bruijn graph structure during read alignment, they are unsuitable for mapping eukaryotic reads which map across splice sites.

Yet, several of the concepts introduced by these tools could serve as the basis for a new alignment algorithm which could accomplish this feat. For instance, reads could be aligned to a compressed de Bruijn graph by using unitig overlaps to seed the alignment[11,29] or by exact matching of read k-mers to the graph[30]. To map exon-spanning reads of which only a small part of the read aligns to one of the exons, a similar idea as employed in HISAT[31] can be used. Namely, the long part of the read can be mapped first to the graph, followed by searching for the correct location of the remaining small part within nearby regions, instead of probing the whole graph. Finally, one could identify the different paths which represent possible alignments of a RNA-seq read to a de Bruijn graph by using Bandage[32]. By exploring the concepts introduced by the algorithms mentioned above, the problem of mapping spliced reads to a de Bruijn graph representation of a draft genome could potentially be solved in the future. This would open up the way for using a de Bruijn graph representation of a genomic assembly as a reference to improve transcriptome assembly.

Despite the limitations of GSNAP, Glue still managed to make a draft genome assembly better suited for genome-guided transcriptome methods. This result shows that joining genomic contigs based on RNA-seq reads mapping across exon-intron boundaries is a helpful method to improve transcriptome reconstruction. Therefore, I expect that Glue is a useful tool for aiding studies which require a high quality transcriptome, such as gene expression.

References

1. Wang, Z., Gerstein, M. & Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.* **10,** 57–63 (2009).
2. Tu, Q., Cameron, R. A., Worley, K. C., Gibbs, R. a & Davidson, E. H. Gene structure in the sea urchin Strongylocentrotus purpuratus based on transcriptome analysis. *Genome Res.* **22,** 2079–87 (2012).
3. Ding, Y. *et al.* Four distinct types of dehydration stress memory genes in Arabidopsis thaliana. *BMC Plant Biol.* **13,** 229 (2013).
4. Kim, D. *et al.* TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.* **14,** R36 (2013).
5. Wang, E. T. *et al.* Alternative isoform regulation in human tissue transcriptomes. *Nature* **456,** 470–476 (2008).
6. Reddy, T. B. K. *et al.* The Genomes OnLine Database (GOLD) v.5: a metadata management system based on a four level (meta)genome project classification. *Nucleic Acids Res.* **1,** 1–8 (2014).
7. Denton, J. F. *et al.* Extensive Error in the Number of Genes Inferred from Draft Genome Assemblies. *PLoS Comput. Biol.* **10,** e1003998 (2014).

8. Peng, Y. *et al.* IDBA-tran: A more robust *de novo* de Bruijn graph assembler for transcriptomes with uneven expression levels. *Bioinformatics* **29,** 326–334 (2013).
9. Mortazavi, A. *et al.* Scaffolding a Caenorhabditis nematode genome with RNA-seq. *Genome Res.* **20,** 1740–1747 (2010).
10. Zhang, S. V., Zhuo, L. & Hahn, M. W. AGOUTI: improving genome assembly and an-notation using transcriptome data. *BioRxiv Prepr.* (2015).
11. Ye, Y. & Tang, H. Utilizing de Bruijn graph of metagenome assembly for metatranscriptome analysis. *Bioinformatics* 1–8 (2015).
12. Deorowicz, S., Kokot, M., Grabowski, S. & Debudaj-Grabysz, a. KMC 2: fast and resource-frugal k-mer counting. *Bioinformatics* **31,** 1569–1576 (2015).
13. Chikhi, R., Limasset, A., Jackman, S., Simpson, J. T. & Medvedev, P. On the representation of de bruijn graphs. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)* **8394 LNBI,** 35–55 (2014).
14. Wu, T. D. & Nacu, S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics* **26,** 873–81 (2010).
15. Dobin, A. *et al.* STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).
16. Hoffmann, S. *et al.* Fast mapping of short sequences with mismatches, insertions and deletions using index structures. *PLoS Comput. Biol.* **5,** e1000502 (2009).
17. Myers, E. W. The fragment assembly string graph. *Bioinformatics* **21,** ii79–ii85 (2005).
18. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9,** 357–9 (2012).
19. Simpson, J. T. & Durbin, R. Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* **22,** 549–556 (2012).
20. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv Prepr.* **00,** 3 (2013).
21. Luo, R. *et al.* SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience* **1,** 18 (2012).
22. Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* **28,** 511–515 (2010).
23. Grabherr, M. G. *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biotechnol.* **29,** 644–652 (2011).
24. Wu, T. D. & Watanabe, C. K. GMAP: A genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics* **21,** 1859–1875 (2005).
25. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).
26. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26,** 841–842 (2010).
27. Gurevich, A., Saveliev, V., Vyahhi, N. & Tesler, G. QUAST: Quality assessment tool for genome assemblies. *Bioinformatics* **29,** 1072–1075 (2013).
28. Boetzer, M. & Pirovano, W. Toward almost closed genomes with GapFiller. *Genome Biol.* **13,** R56 (2012).
29. Limasset, A. & Peterlongo, P. Read Mapping on de Bruijn graph. *Prepr. ArXiv* (2015). at <http://arxiv.org/abs/1505.04911>
30. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal RNA-Seq quantification. *Prepr. ArXiv* (2015).
31. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods* **12,** 357–360 (2015).
32. Wick, R. R., Schultz, M. B., Zobel, J. & Holt, K. E. Bandage: interactive visualisation of de novo genome assemblies. *Bioinformatics* btv383 (2015). doi:10.1093/bioinformatics/btv383