On Using Longer RNA-seq Reads to Improve Transcript Prediction Accuracy*

Anna Kuosmanen[1], Ahmed Sobih[1], Romeo Rizzi[2], Veli Mäkinen[1] and Alexandru I. Tomescu[1]
[1]*Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland*
[2]*Department of Computer Science, University of Verona, Italy*
{*aekuosma, vmakinen, tomescu*}*@cs.helsinki.fi, ahmedsobeeh1985@gmail.com, romeo.rizzi@univr.it*

Keywords: RNA-seq, Long Reads, Transcript Prediction, Network Flow, Splicing Graph, Minimum Path Cover

Abstract: Over the past decade, sequencing read length has increased from tens to hundreds and then to thousands of bases. Current cDNA synthesis methods prevent RNA-seq reads from being long enough to entirely capture all the RNA transcripts, but long reads can still provide connectivity information on chains of multiple exons that are included in transcripts. We demonstrate that exploiting full connectivity information leads to significantly higher prediction accuracy, as measured by the F-score. For this purpose we implemented the solution to the Minimum Path Cover with Subpath Constraints problem introduced in (Rizzi et al., 2014), which is an extension of the classical Minimum Path Cover problem and was shown solvable by min-cost flows. We show that, under hypothetical conditions of perfect sequencing, our approach is able to use long reads more effectively than two state-of-the-art tools, StringTie and FlipFlop. Even in this setting the problem is not trivial, and errors in the underlying flow graph introduced by sequencing and alignment errors complicate the problem further. As such our work also demonstrates the need for a development of a good spliced read aligner for long reads. Our proof-of-concept implementation, as well as the simulated data and the validation scripts, are available at `http://www.cs.helsinki.fi/en/gsa/traphlor`.

INTRODUCTION

With the advent of third-generation PacBio and Oxford nanopore sequencers, the sequencing read length has increased from a few hundred to many thousand bases. These long reads have caused a breakthrough with genome assembly, but they have not yet been widely adopted in use for transcriptome analysis. However, it is very likely that there will be a shift from short RNA-seq reads to long RNA-seq reads in the near future as well.

The optimal case for long read RNA-seq would naturally be the sequencing of full-length transcripts. However, while the sequencing technologies might allow for this, current cDNA synthesis methods unfortunately do not. In experiments it has been shown that full-length reads are less likely to be observed with transcripts longer than 2.5 kb (Sharon et al., 2013). But even with non-full-length reads we can gain valuable information about the connectivity of non-neighboring exons from long reads.

The idea of using long reads in transcript prediction pre-dates the development of RNA-seq. (Florea et al., 2005) proposed a method where expressed sequence tag (EST) sequences were used to score candidate transcripts, which were gained by enumerating over all the paths in the splicing graph, to measure their suitability for gene annotation. In the era of RNA-seq, several papers have similarly formulated the assembly problem as that of finding the minimum number of RNA transcripts such that every read is contained in at least one of them (Rizzi et al., 2014; Bao et al., 2013). Such a parsimony criterion is also found in methods dealing with shorts reads alone (Trapnell et al., 2010; Song and Florea, 2013).

While the polynomial time algorithm given in (Bao et al., 2013) for this problem was not complete, (Rizzi et al., 2014) proved that this problem is indeed polynomially solvable by network flows. In this paper we implemented this network flow approach as a proof-of-concept software. While it is instinctively obvious that increasing read length should increase the transcript prediction accuracy, this topic has not yet been explored in an experimental setting. Our experiments show that in general this statement about

*To appear in *Proc. BIOINFORMATICS 2016*, `http://bioinformatics.biostec.org/`

higher prediction accuracy holds when long reads are properly modeled, but if the model does not take long reads into account, the sensitivity of the prediction can decrease as read length increases.

Even under hypothetical conditions of perfect sequencing, the transcript prediction problem is not trivial, and errors introduced to the underlying graph by sequencing and alignment errors complicate the problem further. Our work demonstrates that correctly aligned long reads combined with a model taking into account the long reads can raise transcript prediction accuracy to a new level, and highlights the need for the development of a good spliced long read aligner.

METHODS

Our algorithm consists of two main parts: creating a splicing graph and solving the assembly problem on top of this graph.

From the read alignments, we construct a splicing graph (Heber et al., 2002), where nodes are exons, and arcs are exons consecutive in some read alignment. Since a splicing graph is constructed from read alignments, it is also acyclic. The long read alignments give some paths of the graph (referred to as subpath constraints) which need to be covered entirely by a collection of paths (the assembled transcripts).

Our assembly objective is to have a minimum number of paths covering all nodes and all subpath constraints, and among such collections of paths, to have one that minimizes a certain cost, as we will explain below. This problem was formulated as the "Minimum Weight Minimum Path Cover with Subpath Constraints (MW-MPC-SC)" problem by (Rizzi et al., 2014), and proved solvable by minimum-cost network flows. See Fig. 1 for a simple example. We implemented a slightly modified version of the solution from (Rizzi et al., 2014), which we briefly describe next.

Recall from (Rizzi et al., 2014; Mäkinen et al., 2015) that a minimum collection of paths covering every node (a minimum path cover) which also minimizes the sum of the weights of the arcs used by its paths can be solved by network flows as follows. Subdivide every node v into two nodes v_{in} and v_{out} connected by an arc (v_{in}, v_{out}), with demand 1. All in-neighbors of v become in-neighbors of v_{in}, and all out-neighbors of v become out-neighbors of v_{out}. Add a global source s connected to every node from where a path is allowed to start. Also add a high weight on these arcs, which will force the solution to have the minimum number of paths. Likewise, add a global sink and arcs from every node in which a path is allowed to end. Then compute a minimum-cost flow on this acyclic flow network, and arbitrarily decompose it into paths, which will form an optimal solution.

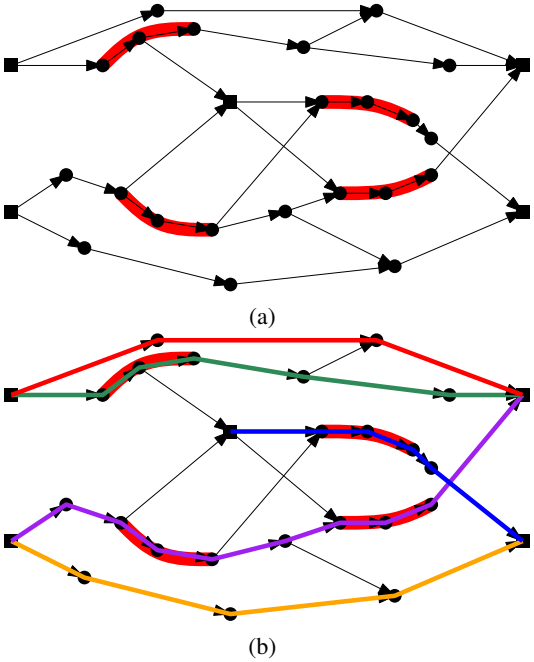
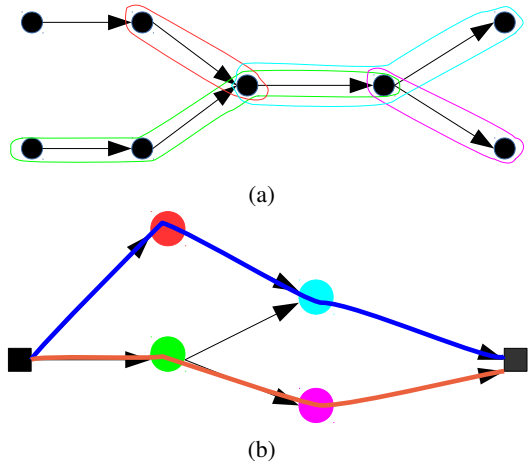


Figure 1: Fig. 1(a): an example of a splicing graph, in which three subpath constraints are drawn in red. The square nodes are the ones where the solutions paths are allowed to start or to end. Fig. 1(b): the minimum number of paths covering all nodes and subpath constraints.

(Rizzi et al., 2014) observed that this reduction can be modified to solve the MW-MPC-SC problem as well. The idea is to model every subpath constraint with first node u and last node w by adding an arc (u_{out}, w_{in}) with cost equalling the sum of the costs of its arcs and demand 1. Also, the demands on its nodes have to be set back to 0. The main difficulty is to deal with overlapping subpath constraints, as these new arcs may increase the optimum number of paths needed to satisfy all the constraints. See Fig. 2 for more details. (Rizzi et al., 2014) proved that the optimal solutions are preserved if constraints sharing a longest suffix-prefix overlap are merged iteratively.

Even though this strategy preserved the minimum number of solution paths, we observed experimentally that this iterative greedy and local merging does not produce the best results. Therefore, we merge subpath constraints in a more globally informed manner, similarly to (Ntafos and Hakimi, 1979). We create another flow network with the subpath constraints as nodes. An arc is added between two nodes if the constraints they represent share a suffix-prefix overlap, and the weights on the arcs are set based on the difference of coverage between their endpoints (see



(a)

(b)

Figure 3: Fig. 3(a): an example of a splicing graph with four subpath constraints with suffix-prefix overlaps. Fig. 1(b): the flow network created from the overlapping constraints and the minimum number of paths covering all the nodes. The square nodes are the global source and the global sink of the flow network.

Fig. 3 for a simple example). A high weight is set, as above, on each arc exiting from the global source of the network, to prefer solutions with the minimum number of paths. The resulting minimum-cost flow (solved with the LEMON library (Lemon, 2014)) is then split into paths. Finally, the constraints represented by the nodes belonging to a same path are merged. At the end of this process the resulting constraints have no suffix-prefix overlap.

RESULTS

We compared our method's performance against two state-of-the-art transcript assemblers, StringTie (Pertea et al., 2015) and FlipFlop (Bernard et al., 2014). All three of the tools use a network flow-based approach.

Since StringTie was shown to have superior performance (Pertea et al., 2015) over short read assemblers Cufflinks (Trapnell et al., 2010), Isolasso (Li et al., 2011), Scripture (Guttman et al., 2010) and Traph (Tomescu et al., 2013), we did not include any of them in this comparison.

For real data there is no ground truth to compare against, thus we used simulated data. Flux Simulator (Griebel et al., 2012), which simulates both the library preparation and sequencing, does not to our knowledge support simulating long reads, and we used the RNASeqReadSimulator (Li, 2012), which only simulates sequencing, instead. For the simulation we used all human transcripts (GRCh37/hg19) that were at least one kilobase long, as provided by the UCSC Genome Browser (Karolchik et al., 2014).

First we sampled weights for the transcripts from log-normal distribution ($\mu = -4, \sigma = 1$) to simulate expression levels. With these parameters, the simulated expression levels of the transcripts vary by three orders of magnitude. While this is significantly lower than the six orders of magnitude that have been observed in expression levels in cells (Holland, 2002), it is more informative for our purposes, as our experiments showed that with high variance, very high read coverage is required to find evidence for more than one transcript per gene locus.

Reads were then sampled from the transcripts based on the simulated weights, with the starting positions of the reads on the transcript following uniform distribution. To exclude the effect the increasing coverage has on transcript predictions, we opted to keep the coverage constant by decreasing the number of simulated reads as read length was increased. We used a total of eight data sets: 60 million 400 bp reads, 30 million 800 bp reads, 20 million 1200 bp reads, 15 million 1600 bp reads, 12 million 2000 bp reads, 10 million 2400 bp reads, 8.6 million 2800 bp reads and 7.5 million 3200 bp reads. In the case that the transcript was shorter than the chosen read length, the whole transcript was added to the data set.

We considered two cases: the ideal conditions of perfect read alignments (created by converting the simulated reads into alignments with BEDTools (Quinlan and Hall, 2010)) and reads aligned with an alignment software (in this case, GMAP (Wu and Watanabe, 2005)). For the latter case, we did not introduce any sequencing errors into the simulation of the reads.

For the experiments we used Dell PowerEdge M610 with 32 GB of RAM and 2 Intel Xeon E5540 2.53GHz CPU:s. Default parameters were used for all the assemblers. When processing the GMAP alignments, FlipFlop attempted to allocate over 50 GB of RAM and as such failed to run on the machines available to us.

For validation, we followed the example of (Li et al., 2011). All predicted transcripts are matched against all annotated transcripts used in creating the data sets, and two transcripts consisting of more than one exon are considered to match if (1) they include the same set of exons and (2) all internal boundary candidates are identical (beginning of first exon and end of last exon do not need to match). Single exon transcripts are considered to match if the overlapping area occupies at least 50% of the length of each transcript. Contrary to Li's *et al.* approach where multiple predicted transcripts could match a single annotated

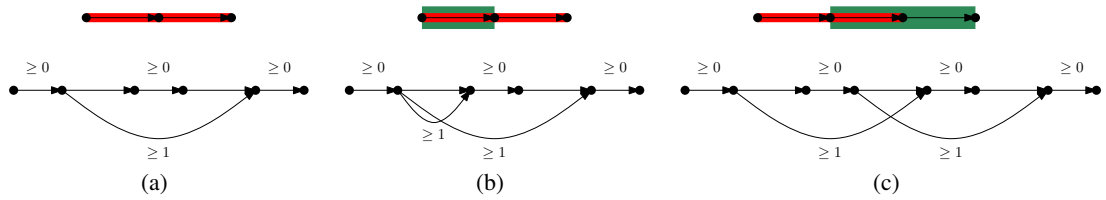


Figure 2: Fig. 2(a): each node v in a subpath constraint was subdivided into the arc (v_{in}, v_{out}) with demand 1. A subpath constraint (red) is modeled as an arc from the *out* copy of its first node to the *in* copy of its last node. The demands of its nodes are set back to 0, and the demand of the new arc is set to 1. Fig. 2(b): a subpath constraint (green) is fully included in another subpath constraint (red). Fig. 2(c): a subpath constraint (red) has a suffix-prefix overlap with another subpath constraint (green). In both of these cases, modelling the subpath constraints as described in Fig. 2(a) increases the minimum number of solution paths.

transcript, we chose a criteria that only one predicted transcript can match a single annotated transcript.

We define *sensitivity* as the number of matched transcripts divided by the number of annotated transcripts and *precision* as the number of matched transcripts divided by the number of predicted transcripts. *F-score*, the standard measure of performance, is the harmonic mean of sensitivity and precision.

In the development of our method we favored high sensitivity over high precision, and as can be seen in Figure 4(a) and Figure 5(a), our method has significantly higher sensitivity than StringTie and FlipFlop as read length increases. However, high sensitivity comes at the cost of lowered precision, especially when alignment errors are introduced into the data (as can be seen in Figure 5(b)). With perfect mappings, it can be seen in Figure 4(c) that using the standard measure of performance, f-score, our method performs more accurately than the competitors when read length increases above 400 bp.

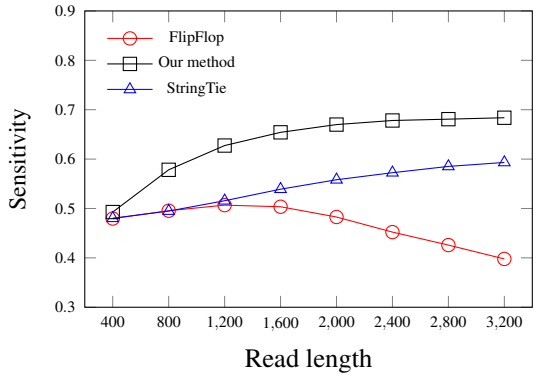
CONCLUSIONS AND FUTURE WORK

In this article we demonstrated the utility of long RNA-seq reads in transcript prediction. We implemented the solution to the "Minimum Weight Minimum Path Cover with Subpath Constraints" problem by (Rizzi et al., 2014), which models long reads as subpath constraints, that is, sequences of exons that have to be fully contained in one of the solution paths. We showed with simulated data that, with proper models, increasing read length can improve transcript prediction accuracy significantly (as measured by the F-score). We also showed that our proof-of-concept software is able to use long reads more effectively than the competitors StringTie and FlipFlop, while also remaining on par with modest read lengths. It should be noted though that while StringTie and FlipFlop can use long reads, they were designed for short reads.

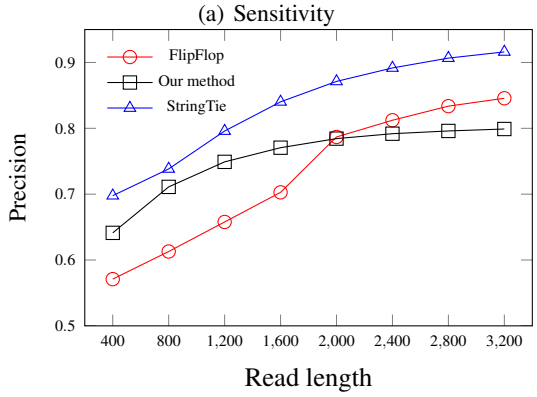
As our problem formulation requires all the subpath constraints to be contained in one of the solution paths, the model is sensitive to high levels of noise, e.g. alignment errors near splice sites. Our initial experiments showed that even with modest indel rates (1%) finding the splice sites reliably was too challenging for any publicly available high-throughput tool we are aware of. This problem could be tackled with an exon chaining approach (Gelfand et al., 1996); however this is solvable only in quadratic time, and scaling it to high-throughput sequencing read alignment would require massive parallelism. Finding effective methods for spliced long read alignment has been the subject of several PhD theses (Kopylova, 2013; Vyverman, 2014), and our work further demonstrates the need for finding such a method.

We are, however, confident that long RNA-seq reads can in the end be utilized even without a perfect spliced read aligner: Our method only requires the connectivity information, that is, a chain of exons forming each long read. The exons can be predicted reliably with short RNA-seq reads, and therefore a hybrid approach appears amenable. This suggests to study a relaxed version of the spliced alignment problem, where the correctness of the chain of exons is optimized rather than the global alignment score. This is a subject of another manuscript.

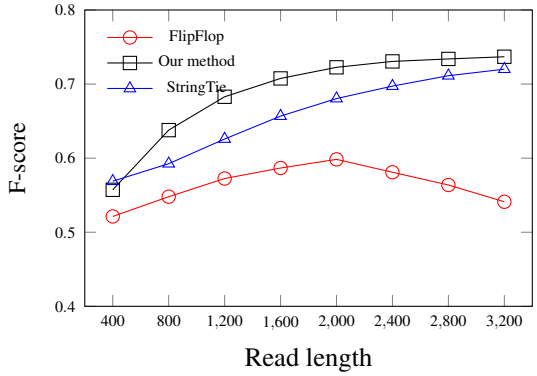
Last, note that currently our approach produces only transcript sequences, but there exist a multitude of tools for quantifying transcript abundances (Li and Dewey, 2011) and differential expression analysis between samples (Robinson et al., 2010; Anders and Huber, 2010; Glaus et al., 2012) that can be applied to our tool's output. Adding the quantification step to the tool is also one possible direction for future work.



(a) Sensitivity

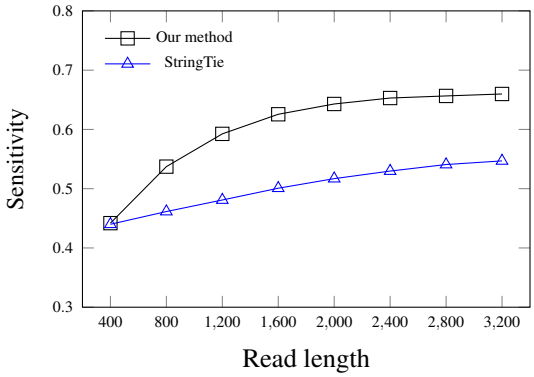


(b) Precision

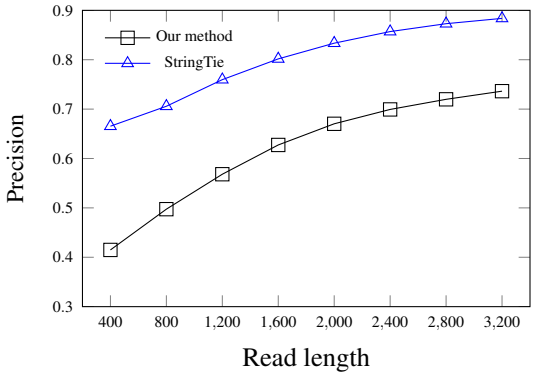


(c) F-score

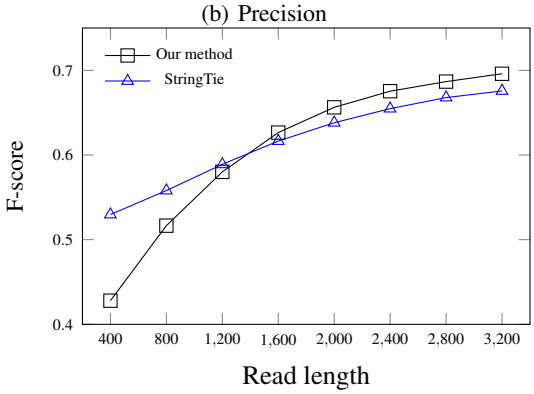
Figure 4: Sensitivity, precision and F-score with perfect alignments.



(a) Sensitivity



(b) Precision



(c) F-score

Figure 5: Sensitivity, precision and F-score using alignments from GMAP.

ACKNOWLEDGEMENTS

We would like to thank all the anonymous reviewers for their constructive feedback.

Funding: This work was partially supported by the Academy of Finland [284598 to A.K., A.S. and V.M., 274977 to A.I.T.].

REFERENCES

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biol*, 11(10):R106.

Bao, E., Jiang, T., and Girke, T. (2013). BRANCH: boosting RNA-Seq assemblies with partial or related genomic sequences. *Bioinformatics*, 29(10):1250–1259.

Bernard, E., Jacob, L., Mairal, J., and Vert, J.-P. (2014). Efficient RNA isoform identification and quantification

from RNA-Seq data with network flows. *Bioinformatics*, 30(17):2447–2455.

Florea, L., Di Francesco, V., Miller, J., Turner, R., Yao, A., Harris, M., Walenz, B., Mobarry, C., Merkulov, G. V., Charlab, R., Dew, I., Deng, Z., Istrail, S., Li, P., and Sutton, G. (2005). Gene and alternative splicing annotation with AIR. *Genome Res*, 15(1):54–66.

Gelfand, M. S., Mironov, A. A., and Pevzner, P. A. (1996). Gene recognition via spliced sequence alignment. *Proc. Natl Acad Sci U S A*, 93(17):9061–6.

Glaus, P., Honkela, A., and Rattray, M. (2012). Identifying differentially expressed transcripts from RNA-seq data with biological variation. *Bioinformatics*, 28(13):1721–1728.

Griebel, T., Zacher, B., Ribeca, P., Raineri, E., Lacroix, V., Guigó, R., and Sammeth, M. (2012). Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res*, 40(20):10073–10083.

Guttman, M., Garber, M., Levin, J. Z., Donaghey, J., Robinson, J., Adiconis, X., Fan, L., Koziol, M. J., Gnirke, A., Nusbaum, C., Rinn, J. L., Lander, E. S., and Regev, A. (2010). Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nat Biotechnol*, 28(5):503–510.

Heber, S., Alekseyev, M., Sze, S.-H., Tang, H., and Pevzner, P. A. (2002). Splicing graphs and EST assembly problem. *Bioinformatics*, 18 Suppl 1:S181–S188.

Holland, M. J. (2002). Transcript abundance in yeast varies over six orders of magnitude. *J Biol Chem*, 277(17):14363–14366.

Karolchik, D., Barber, G. P., Casper, J., Clawson, H., Cline, M. S., Diekhans, M., Dreszer, T. R., Fujita, P. A., Guruvadoo, L., Haeussler, M., Harte, R. A., Heitner, S., Hinrichs, A. S., Learned, K., Lee, B. T., Li, C. H., Raney, B. J., Rhead, B., Rosenbloom, K. R., Sloan, C. A., Speir, M. L., Zweig, A. S., Haussler, D., Kuhn, R. M., and Kent, W. J. (2014). The UCSC Genome Browser database: 2014 update. *Nucleic Acids Res*, 42(Database issue):D764–D770.

Kopylova, E. (2013). *New algorithmic and bioinformatic approaches for the analysis of data from high throughput sequencing*. PhD thesis, Université des Sciences et Technologie de Lille-Lille I.

Lemon (2014). **L**ibrary for **E**fficient **M**odeling and **O**ptimization in **N**etworks. http://lemon.cs.elte.hu/.

Li, B. and Dewey, C. N. (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, 12:323.

Li, W. (2012). http://alumni.cs.ucr.edu/~liw/rnaseqreadsimulator.html.

Li, W., Feng, J., and Jiang, T. (2011). IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *J Comput Biol*, 18(11):1693–1707.

Mäkinen, V., Belazzougui, D., Cunial, F., and Tomescu, A. I. (May 2015). *Genome-Scale Algorithm Design—Biological Sequence Analysis in the Era of High-Throughput Sequencing*. Cambridge University Press. URL `www.genome-scale.info`.

Ntafos, S. C. and Hakimi, S. L. (1979). On path cover problems in digraphs and applications to program testing. *IEEE Transactions on Software Engineering*, SE-5(5):520–529.

Pertea, M., Pertea, G. M., Antonescu, C. M., Chang, T.-C., Mendell, J. T., and Salzberg, S. L. (2015). StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat Biotechnol*, 33(3):290–295.

Quinlan, A. R. and Hall, I. M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842.

Rizzi, R., Tomescu, A. I., and Mäkinen, V. (2014). On the complexity of Minimum Path Cover with Subpath Constraints for multi-assembly. *BMC Bioinformatics*, 15(S-9):S5.

Robinson, M. D., McCarthy, D. J., and Smyth, G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140.

Sharon, D., Tilgner, H., Grubert, F., and Snyder, M. (2013). A single-molecule long-read survey of the human transcriptome. *Nat Biotechnol*, 31(11):1009–1014.

Song, L. and Florea, L. (2013). CLASS: constrained transcript assembly of RNA-seq reads. *BMC Bioinformatics*, 14 Suppl 5:S14.

Tomescu, A. I., Kuosmanen, A., Rizzi, R., and Mäkinen, V. (2013). A novel min-cost flow method for estimating transcript expression with RNA-Seq. *BMC Bioinformatics*, 14 Suppl 5:S15.

Trapnell, C., Williams, B. A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M. J., Salzberg, S. L., Wold, B. J., and Pachter, L. (2010). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol*, 28(5):511–515.

Vyverman, M. (2014). *ALFALFA: Fast and Accurate Mapping of Long Next Generation Sequencing Reads*. PhD thesis, Ghent University.

Wu, T. D. and Watanabe, C. K. (2005). GMAP: a genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, 21(9):1859–1875.