To appear in IEEE/ACM Transactions on Computational Biology and Bioinformatics

Explaining a Weighted DAG with Few Paths for Solving Genome-Guided Multi-assembly

Alexandru I. Tomescu, Travis Gagie, Alexandru Popa, Romeo Rizzi, Anna Kuosmanen, Veli Mäkinen

Abstract—RNA-Seq technology offers new high-throughput ways for transcript identification and quantification based on short reads, and has recently attracted great interest. This is achieved by constructing a weighted DAG whose vertices stand for exons, and whose arcs stand for split alignments of the RNA-Seq reads to the exons. The task consists in finding a number of paths, together with their expression levels, which optimally explain the weights of the graph under various fitting functions, such as least sum of squared residuals. In (Tomescu *et al. BMC Bioinformatics*, 2013) we studied this *genome-guided multi-assembly* problem when the number of allowed solution paths was linear in the number of arcs.

In this paper we further refine this problem by asking for a bounded number k of solution paths, which is the setting of most practical interest. We formulate this problem in very broad terms, and show that for many choices of the fitting function it becomes NP-hard. Nevertheless, we identify a natural graph parameter of a DAG G, which we call *arc-width* and denote $\langle G \rangle$, and give a dynamic programming algorithm running in time $O(W^k \langle G \rangle^k (\langle G \rangle + k)n)$, where n is the number of vertices and W is the maximum weight of G. This implies that the problem is fixed-parameter tractable (FPT) in the parameters W, $\langle G \rangle$ and k. We also show that the arc-width of DAGs constructed from simulated and real RNA-Seq reads is small in practice. Finally, we study the approximability of this problem, and, in particular, give a fully polynomial-time approximation scheme (FPTAS) for the case when the fitting function penalizes the maximum ratio between the weights of the arcs and their predicted coverage.

Index Terms—RNA-sequencing, transcript prediction, splicing graph, NP-hardness, dynamic programming, fixed-parameter tractability, digraph-width measure, approximation algorithm.

1 Introduction

1.1 Background

In this paper we tackle a biological multi-assembly problem [2] motivated by the recent RNA-Seq technology [3], [4], [5]: reconstruct as accurately as possible the RNA transcripts of a gene, given only a set of short RNA reads sequenced from them. The transcripts are concatenations of exons, the difficulty of the problem arising from the fact that they can have some identical exons.

Even though some *de novo* tools try to assemble the transcripts only from the RNA-Seq reads [6], most tools use reference information. This second setting consists of two non-trivial steps. The first is the spliced alignment of the RNA-Seq reads to the reference genome, as solved by [7], [8]. The second problem, which is the one we tackle in this paper, is separating the coverage obtained in the first step into individual transcripts.

This paper is an extended version of [1].

- *A.I. Tomescu, T. Gagie, A. Kuosmanen and V. Mäkinen are with the Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Finland. E-mails: {tomescu,gagie,aekuosma,vmakinen,}@cs.helsinki.fi*
- *A. Popa is with the School of Science and Technology, Nazarbayev University, Astana, Kazakhstan. Email: alexandru.popa@nu.edu.kz*
- *R. Rizzi is with the Department of Computer Science, University of Verona, Italy. E-mail: romeo.rizzi@univr.it*

This *genome-guided* multi-assembly problem has attracted great interest from the community, resulting in tools such as Cufflinks [9], IsoInfer/IsoLasso [10], [11], SLIDE [12], CLIIQ [13], Scripture [14], iReckon [15], TRIP [16], NSMAP [17], Montebello [18], FlipFlop [19]. These methods rely on a graph model, the most common one being a *splicing graph* [20]. Its vertices represent contiguous stretches of DNA uninterrupted by spliced reads (called *pseudo-exons*), while its arcs are derived from overlaps, or from spliced read alignments. Since it arises from alignments to a reference genome, the splicing graph is directed and acyclic (a DAG); the orientation of the arcs is according to the starting positions of the pseudo-exons inside the genome. Every vertex v has an associated observed average coverage, computed as the total length of the read fragments aligned to the pseudo-exon v, divided by the pseudo-exon length. Similarly, every arc (u, v) has an associated coverage, which is the total number of reads splice-aligned to the junction between pseudo-exons u and v. Throughout this paper we denote by n and m the number of vertices and arcs, respectively, of the input DAG.

The biological multi-assembly and quantification problem translates to covering the graph with paths under different cost models, such as least sum of squared residuals (IsoInfer/IsoLasso, SLIDE), or least sum of absolute values of the residuals (CLIIQ). Many of the above mentioned tools work by exhaustively enumerating all possible (combinations of) paths, un-

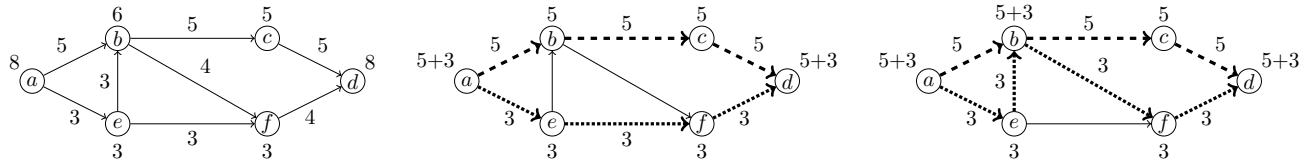


Fig. 1. An example for $k = 2$, and fitting function $f(x) = x^2$. In Fig. 1(a), a splicing directed acyclic graph; its vertices and arcs are labeled with their observed average coverage. In Fig. 1(b), the optimal two paths for Problem 2-UTEO, with expression levels 5 and 3, respectively; vertices and arcs are labeled with their predicted coverage from these two paths. The cost of this solution is $1+1$, from vertex b, and from arc (f, d), respectively. In the case of Problem 2-UTEC, we have to add 3^2+4^2 to the cost of this non-optimal solution, from uncovered arcs (e, b), (b, f). In Fig. 1(c), the optimal 2 paths for Problem 2-UTEC, with expression levels 5 and 3, respectively; vertices and arcs are labeled with their predicted coverage from these two paths. The cost of this solution is $2^2 + 1 + 1 + 3^2$, from vertex b, and arcs (b, f), (f, d), (e, f), respectively.

der some restrictions, and then estimating their fitting with an Integer Linear Program, Quadratic Program, or a QP + LASSO regression. Cufflinks computes a minimum weight minimum path cover, and only in a second step estimates the expression levels of the paths.

1.2 Previous work

In [21] we introduced a general framework, encompassing many of the previous cost models; according to the survey [22], it can be classified as *de novo* genome-guided, since it does not use gene annotation information. Let f be a fitting function penalizing the absolute difference between the observed coverage of a vertex or an arc, and the sum of the expression levels of the paths using that vertex or arc (we call this sum the *predicted coverage* of that vertex or arc). The genome-guided multi-assembly problem can be simply stated as finding (an unlimited number of) paths with associated expression levels which minimize the sum of the penalties of all residuals for each vertex and arc. Formally, we have the following problem:[1]

Problem UTEC (Unannotated transcript expression—cover)**.** *Given a DAG $G = (V, E)$, a weight function $w : V \cup E \to \mathbb{R}_+$, and a fitting function $f : \mathbb{R}_+ \to \mathbb{R}_+$, find a tuple $\mathcal{P}$ of paths from the sources of G to the sinks of G, with an estimated expression level $e(P)$ for each path $P \in \mathcal{P}$, which minimize*

$$\sum_{v \in V} f\Big(\Big|w(v) - \sum_{P \in \mathcal{P} \text{ s.t. } v \in P} e(P)\Big|\Big) + \sum_{(u,v) \in E} f\Big(\Big|w(u,v) - \sum_{P \in \mathcal{P} \text{ s.t. } (u,v) \in P} e(P)\Big|\Big).$$

1. To be precise, Problem UTEC was stated as receiving in input a different fitting function for every vertex and arc; this is important in practice, since the fitting function can depend, for example, also on the exon length, or on the variance of its coverage. Nevertheless, for simplicity we state here all the problems with a unique fitting function. All results and algorithms apply to this more general case as well.

For example, if $f(x) = x^2$, then we have a least sum of squared residuals model similar to the ones in IsoInfer/IsoLasso and SLIDE, and if $f(x) = x$ we have a model as in CLIIQ (see Fig. 1 for an example). For any convex fitting function, Problem UTEC can be solved in polynomial-time by a reduction to a minimum-cost flow problem with convex costs [21]. The reduction works by finding the optimal flow, and then splitting this flow into at most $|E|$ paths.

However, in practice we are interested in parsimoniously explaining the given DAG with few paths, since a small fraction of the graph may be erroneous. This can be due to various biological events such as template switching, self-priming, intron retention, or due to technical errors related to reading or alignment [23], [5], [24], [25]. Notice that splitting any flow into the minimum number of paths is an NP-hard problem [26]. For this reason, in [21] we employed a heuristic from [26] for splitting the flow by repeatedly choosing and removing the path carrying the maximum amount of flow (i.e., the path of maximum *bottleneck*).

One possible workaround for reporting few solution paths appeared for example in IsoLasso [11] and in FlipFlop [19]. These methods add a regularization term $\lambda \sum_{P \in \mathcal{P}} e(P)$ to an objective function similar to the one in Problem UTEC, for some opportune λ. The experiments in [11] and [19] show that the optimal solution according to this objective function also prefers few solution paths. To be more precise, the method in [19] is also based on a reduction to a minimum-cost flow problem, by appropriately adding this regularization term as cost in the flow network. The resulting flow is split into paths by the same heuristic from [26], [21]. The number of paths produced in this manner is low, but it is not proven that the resulting flow is in fact decomposed into the minimum number of paths (recall that the problem of minimally splitting a flow is in general NP-hard [26]).

Another workaround was proposed in [1], where we generalized Problem UTEC to ask for a given

number k of paths:

Problem k**-UTEC** (k-Unannotated transcript expression—cover)**.** *Given a DAG $G = (V, E)$, a weight function $w : V \cup E \to \mathbb{R}_+$, a fitting function $f : \mathbb{R}_+ \to \mathbb{R}_+$, and an integer k, find a tuple $\mathcal{P}$ of k paths from the sources of G to the sinks of G, with an estimated expression level $e(P)$ for each path $P \in \mathcal{P}$, which minimize*

$$\sum_{v \in V} f\Big(w(v) - \sum_{P \in \mathcal{P} \text{ s.t. } v \in P} e(P)\Big) + \sum_{(u,v) \in E} f\Big(w(u,v) - \sum_{P \in \mathcal{P} \text{ s.t. } (u,v) \in P} e(P)\Big).$$

In [1] we also introduced the following variant, in which the paths should explain only the weights of the vertices and arcs appearing on them, as opposed to the entire graph; thus, the vertices and arcs not appearing on a predicted path are seen as outliers.

Problem k**-UTEO** (k-Unannotated transcript expression–outlier)**.** *Given a DAG $G = (V, E)$, a weight function $w : V \cup E \to \mathbb{R}_+$, a fitting function $f : \mathbb{R}_+ \to \mathbb{R}_+$, and an integer k, find a tuple $\mathcal{P}$ of k paths from the sources of G to the sinks of G, with an estimated expression level $e(P)$ for each path $P \in \mathcal{P}$, which minimize*

$$\sum_{P \in \mathcal{P}} \sum_{v \in P} f\Big(w(v) - \sum_{Q \in \mathcal{P} \text{ s.t. } v \in Q} e(Q)\Big) + \sum_{P \in \mathcal{P}} \sum_{(u,v) \in P} f\Big(w(u,v) - \sum_{Q \in \mathcal{P} \text{ s.t. } (u,v) \in Q} e(Q)\Big).$$

The main result from [1] is that both the k-UTEC and k-UTEO problems are NP-hard. However, if the possible expression levels of the solution paths are assumed to belong to a known set of positive integers $\{1, 2, \ldots, W\}$, then they are solvable by dynamic programming in time $O(W^k n^k (n^2 + \Delta^k))$ and space $O(n^k)$, where Δ is the maximum in-degree. The idea of this algorithm is, for every k-tuple of possible expression levels, to compute the optimal k-tuple of paths having the given expression levels, and ending in every k-tuple of vertices.

Observe that applying k-UTEC for all possible values of k solves Problem UTEC. In particular, if Problem UTEC has an optimal solution with small k, one could be able to find it fast using an algorithm for Problem k-UTEC, yet one could not give a proof of the optimality. For practical purposes, iterating the algorithm for small values of k may still be a good way to select a proper value of k.

Moreover, the solution we will present in this paper for a more general multi-assembly problem immediately allows k to be chosen as in [11] and [19], by adding the regularization term $\lambda \sum_{P \in \mathcal{P}} e(P)$ to the objective functions. Another common way to select the "best" parameter k in analogous problems is to use the *minimum description length* (MDL) principle [27].

1.3 Contribution

In this paper we investigate the limits of a dynamic programming-like approach as in [1]. Accordingly, we generalize Problems k-UTEC and k-UTEO by allowing:

- the fitting function to take as parameters both the observed coverage and the predicted coverage, not only their absolute difference;
- the objective function to be any p-norm $\|\cdot\|_p$ of the vector of penalties, for any $p \in \mathbb{R}_+$.

On the one hand, we show that for any fitting function from two superclasses of positive definite functions, and for any p-norm, we can still solve the corresponding problem by dynamic programming, and give a faster algorithm than in [1]. On the other hand, we show that for any such fitting function, the corresponding problem remains NP-hard. We also give some approximation results, and in particular, present a fully polynomial-time approximation scheme (FPTAS) for the fitting function penalizing the ratio between observed and predicted coverage.

Formally, we propose the following problem.[2]

Problem (f, k, p)**-GGMA** (Genome-guided multi-assembly)**.** *Given a DAG $G = (V, E = \{a_1, \ldots, a_m\})$, a weight function $w : E \to \mathbb{R}_+$, a fitting function $f : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$, and an integer k, find a tuple $\mathcal{P}$ of k paths from the sources of G to the sinks of G, with an estimated expression level $e(P)$ for each path $P \in \mathcal{P}$, which minimize*

$$\|err\|_p,$$

where err denotes the m-dimensional vector having $f\Big(w(a_i), \sum_{P \in \mathcal{P} \text{ s.t. } a_i \in P} e(P)\Big)$ as i-th component (for $i \in \{1, \ldots, m\}$), and $\|err\|_p$ notes its p-norm.

Observe that given a fitting function $f : \mathbb{R}_+ \to \mathbb{R}_+$ for Problem k-UTEC or k-UTEO, Problem k-UTEC is the same as Problem $(f_c, k, 1)$-GGMA, where

$$f_c(x, y) = f(|x - y|),$$

and Problem k-UTEO is the same as Problem $(f_o, k, 1)$-GGMA, for

$$f_o(x, y) = \begin{cases} 0, & \text{if } y = 0, \\ f(x - y), & \text{otherwise.} \end{cases}$$

Problem GGMA also leads to other natural problem variants. For example, if we take

$$f_{e,\delta}(x, y) = \begin{cases} 0, & \text{if } |x - y| \leqslant \delta \\ 1, & \text{otherwise,} \end{cases}$$

2. For the sake of clarity, we make the simplifying assumption that the input DAG is only arc-weighted. This is no restriction, since a weighted vertex v can be replaced by two new vertices v_1 and v_2, connected by an arc of weight $w(v)$, such that the in-neighbors of v_1 are the same as the in-neighbors of v, and the out-neighbors of v_2 are the same as the in-neighbors of v. The resulting DAG still has $O(n)$ vertices and $O(n + m)$ arcs.

for a $\delta \geqslant 0$ given in input, then Problem $(f_{e,\delta}, k, 1)$-GGMA asks for the k paths and their expression levels which maximize the number of arcs whose predicted coverage is within δ to their observed coverage.

If we take

$$f_r(x, y) = \max\left\{\frac{x}{y}, \frac{y}{x}\right\},$$

then Problem (f_r, k, ∞)-GGMA asks for the k paths and their expression levels such that the maximum, over all arcs, of the worst ratio between the observed coverage and the predicted coverage is minimized (we interpret a division by 0 as $+\infty$, so that we need to cover all arcs of the DAG).

Our first result concerning Problem GGMA, presented in Sec. 2, is to show that it remains NP-hard for any p-norm, and for any fitting function from two superclasses of positive definite functions, generalizing f_c, f_o defined above.

Moreover, in Sec. 3 we extend our dynamic programming approach from [1] for the entire family of problems (f, k, p)-GGMA. We do this by identifying a natural graph parameter of a DAG G, which we call the *arc-width* of G, denoted $\langle G\rangle$; it is defined as the minimum number of paths needed to cover all the arcs of G.

We give a dynamic programming-based algorithm working in time $O(W^k\langle G\rangle^k(\langle G\rangle + k)n)$. In particular, this improves our algorithm in [1], since $\langle G\rangle \leqslant |E(G)| \leqslant n\Delta$. However, observe that arc-width should be much smaller in practice, since the splicing DAGs arise from a few RNA transcripts, plus some erroneous arcs. These are due for example to reading or alignment errors, or intron retention. In fact, in Sec. 4 we compute the arc-width for graphs constructed from simulated and real reads from genes of human chromosome 2, and show it is generally much lower than the number of vertices, thus making this algorithm significantly faster than our previous solution in [1].

Recall that a fixed-parameter tractable (FPT) algorithm in a parameter t is an algorithm running in time $O(h(t)p(n))$, where $p(n)$ is a polynomial in the input size n, and $h(t)$ is an arbitrary function of t, but not depending on n. Given a fixed k-tuple of expression levels for the solution paths, our algorithm runs in time $O(\langle G\rangle^k(\langle G\rangle + k)n)$, thus we can say that this algorithm is FPT in $\langle G\rangle + k$ (by taking, e.g., $h(t) = t^t$.)

Finally, in Sec. 3.3 we study the approximability of Problem GGMA. Our strategy is to discretize the weights in $\{1, \ldots, W\}$ according to an arithmetic or geometric progression. We give some approximation results for Problems (f_c, k, ∞)-GGMA and (f_o, k, ∞)-GGMA (where we use an arithmetic progression of ratio εW), while for Problem (f_r, k, p)-GGMA (where we use a geometric progression of ratio $1 + \varepsilon$) we obtain an FPTAS, that is, an algorithm which is given an $\varepsilon > 0$, and returns in time polynomial in both the input size and in $1/\varepsilon$ a solution of cost within factor $(1 + \varepsilon)^{\pm 1}$ to the optimal one.

2 NP-Hardness of Problem GGMA

We first consider a family of fitting functions for which Problem GGMA is NP-hard even in the strong sense. This family is made up of the functions $f : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ satisfying the following property:

Property 1. *For any $x, y \in \mathbb{R}_+$ it holds that*

- $f(x, y) \geqslant 0$,
- $f(x, x) = f(y, y)$, *and*
- $f(x, x) \leqslant f(x, y)$, *with equality if and only if $y = x$.*

Observe that functions $f(x, y) = (x - y)^2, f_{e,0}, f_r$ discussed in the previous section satisfy Property 1. More generally, Property 1 holds for any positive definite function (a function satisfying the separation and the coincidence axioms of a metric).

Theorem 1. *Problem (f, k, p)-GGMA is NP-hard in the strong sense for any function f satisfying Property 1, and any $p \in \mathbb{R}_+$.*

Proof: We follow the proof of [26, Proposition 2] for splitting a flow into a given number of paths, underlining the differences in what follows. We reduce from 3-PARTITION. In this problem, we are given a set $A = \{a_1, \ldots, a_{3q}\}$ with $3q$ positive integers, such that:

- $B/4 < a_i < B/2$, for all $i \in \{1, \ldots, 3q\}$, and
- $\sum_{i=1}^{3q} a_i = qB$.

We are asked whether there exists a partition of A into q disjoint sets, such that the sum of the integers in each of these sets is B.

Given an instance (A, B) to 3-PARTITION, we construct (see also Fig. 2) the DAG $G_{A,B}$ having:

- $V(G_{A,B}) = \{x_1, \ldots, x_{3q}, y_1, y_2, z_1, \ldots, z_q\}$,
- for every $i \in \{1, \ldots, 3q\}$, an arc (x_i, y_1) with coverage a_i,
- an arc (y_1, y_2) with coverage $w(y_1, y_2) = qB$,
- for every $i \in \{1, \ldots, q\}$, an arc (y_2, z_i) with coverage $w(y_2, z_i) = B$.

We prove that there exists a partition of A into q sets of size B if and only if Problem GGMA admits on $G_{A,B}$ a solution made up of $3q$ paths of cost $\|null\_err\|_p$, where $null\_err$ is the vector corresponding to the case where for each arc, its predicted coverage equals its observed coverage, that is,

$$null\_err = (f(a_1, a_1), \ldots, f(a_{3q}, a_{3q}), f(qB, qB), f(B, B), \ldots, f(B, B)).$$

For the forward implication, let $A_1, \ldots, A_q$ be a partition of A into q sets of size B. To obtain a solution to Problem GGMA with cost $\|null\_err\|_p$, for every $A_i = \{a_{i_1}, a_{i_2}, a_{i_3}\}$ we add to the solution the three paths (x_{i_1}, y_1, y_2, z_i), (x_{i_2}, y_1, y_2, z_i), (x_{i_3}, y_1, y_2, z_i), with expression levels a_{i_1}, a_{i_2}, a_{i_3}, respectively. These three

paths completely cover the arcs (x_{i_1}, y_1), (x_{i_2}, y_1), and (x_{i_3}, y_1), respectively, and they are the only paths to do so, since $A_1, \ldots, A_q$ is a partition of A. This results in the costs $f(a_{i_1}, a_{i_1})$, $f(a_{i_2}, a_{i_2})$, $f(a_{i_3}, a_{i_3})$, respectively, in the vector $null\_err$. Moreover, since $a_{i_1} + a_{i_2} + a_{i_3} = B$, then these three paths together completely cover the arc (y_2, z_i). This corresponds to a component equal to $f(B, B)$ in the vector $null\_err$ corresponding to the arc (y_2, z_i). Since $\sum_{i=1}^{3q} a_i = qB$, we have that also the arc (y_1, y_2) is completely covered, which gives the component $f(qB, qB)$ of the vector $null\_err$. Thus the error vector of this solution equals $null\_err$.

For the backward implication, consider a solution with $3q$ paths for Problem GGMA of total cost $\|null\_err\|_p$. From the fact that $\|\cdot\|_p$ is a p-norm, and function f satisfies Property 1, in any solution with $3q$ paths with cost $\|null\_err\|_p$, the predicted coverage of each arc must equal its observed coverage.

Moreover, observe that each vertex x_i is contained in exactly one path. Indeed, by the above observation, the predicted coverage of the arc (y_1, y_2) is precisely qB. This implies that the sum of the expression levels of all $3q$ paths is qB; consequently, each of the $3q$ arcs from the vertices $x_1, \ldots, x_{3q}$ to y_1 must be covered by at most one and thus exactly one of the $3q$ paths.

For every $i \in \{1, \ldots, q\}$, let Q_i denote the set of paths in this optimal solution covering vertex z_i. From the above observations, the sum of their expression levels is B, and their expression levels belong to A. Since $B/4 < a < B/2$, for all $a \in A$, then each Q_i contains exactly three paths. This implies that for any $1 \leqslant i < j \leqslant q$, $Q_i \cap Q_j = \emptyset$. Thus, by associating with each $i \in \{1, \ldots, q\}$ the subset of A that corresponds to the first arc of the three paths of Q_i, we obtain a partition of A into q sets, each of size B. □

In RNA-seq experiments, the expression levels of the observed coverages can be orders of magnitude apart. The above reduction can be easily modified to construct such an instance of the splicing graph, as follows. For an input (A, B) to the 3-PARTITION problem, we can introduce in $G_{A,B}$ from the above proof two other vertices x_0 and z_0 and the arc (x_0, z_0), with observed coverage a_0, where a_0 is orders of magnitude higher than the elements of A. Then by following the proof verbatim, it holds that there exists a partition of A into q sets of size B if and only if Problem GGMA admits on $G_{A,B}$ a solution made up of $3q + 1$ paths of cost $\|null\_err\|_p$, where $null\_err$ is the vector corresponding to the case where for each arc, its predicted coverage equals its observed coverage. Indeed, in the forward implication, we also need to add to the solution the path (x_0, z_0) with expression level a_0, and the proof of the reverse implication does not depend on the new vertices x_0 and z_0. Various other such transformations can be made to $G_{A,B}$ so that it resembles as much as possible real splicing graphs.

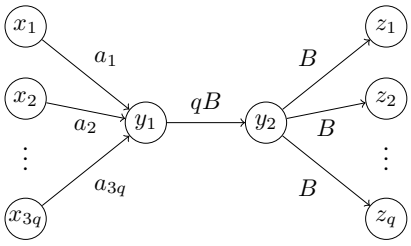


Fig. 2. A reduction of 3-PARTITION to Problem GGMA.

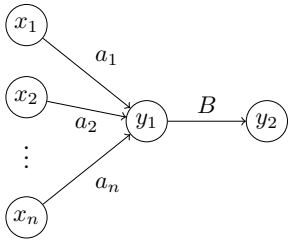


Fig. 3. A reduction of SUBSET SUM to Problem GGMA.

Corollary 1. *Problem k-UTEC with fitting functions $f(x) = |x|$, or $f(x) = x^2$, is NP-hard in the strong sense.*

Proof: Function f_c defined starting from f, as in Sec. 1.3, satisfies Property 1. □

Corollary 2. *Problem $(f_{e,\delta}, k, 1)$-GGMA, where $\delta \geqslant 0$ is a parameter received in input, and Problem (f_r, k, ∞)-GGMA are NP-hard in the strong sense.*

Proof: If we set $\delta = 0$, then function $f_{e,0}$ satisfies Property 1. Likewise, also function f_r satisfies Property 1. □

Our next property captures fitting functions for Problem k-UTEO, in the sense that arcs without predicted coverage should not count in the objective function.

Property 2. *For any $x, y \in \mathbb{R}_+$ it holds that*

- *$f(x, y) \geqslant 0$, with equality if and only if $y \in \{x, 0\}$.*

We cannot prove NP-hardness in the strong sense anymore for fitting functions satisfying Property 2, as our reduction is now from the problem SUBSET SUM, which is NP-hard only in the weak sense [28].

Theorem 2. *Problem (f, k, p)-GGMA is NP-hard in the weak sense, for any function f satisfying Property 2, and any $p \in \mathbb{R}_+$.*

Proof: In the SUBSET SUM problem we are given a set $A = \{a_1, a_2, \ldots, a_n\}$ of positive integers, together with positive integers B and k, and we are asked whether there exists a subset $A' \subseteq A$ such that $|A'| \leqslant k$ and $\sum_{a \in A'} a = B$.

For an instance (A, B, k) of the SUBSET SUM problem, we construct, very similarly to the proof of Thm. 1 and as depicted in Fig. 3, the directed acyclic graph $G_{A,B}$ having:

- $V(G_{A,B}) = \{x_1, \ldots, x_n, y_1, y_2\}$,

- for every $i \in \{1, \ldots, n\}$, we add an arc (x_i, y_1) to $G_{A,B}$, with observed coverage a_i,
- we add an arc (y_1, y_2) with observed coverage $w(y_1, y_2) = B$ to $G_{A,B}$.

We show that the SUBSET SUM problem admits a solution to an input (A, B, k) if and only if Problem GGMA admits on $G_{A,B}$ a solution made up of at most $\ell \leqslant k$ paths of cost 0.

For the forward implication, assume that $B = \{a_{i_1}, \ldots, a_{i_\ell}\}$, $\ell \leqslant k$, is a solution to the SUBSET SUM problem on (A, B, k), thus that $a_{i_1} + \cdots + a_{i_\ell} = B$. It is immediately seen that the tuple of ℓ paths (x_{i_j}, y_1, y_2), each being assigned expression level a_{i_j}, for all $j \in \{1, \ldots, \ell\}$ is a solution with cost 0.

For the backward implication, let $\mathcal{P} = (P_1, \ldots, P_\ell)$ be a tuple of ℓ paths from sources to the unique sink y_2, $\ell \leqslant k$, of cost 0. Let $\{x_{i_1}, \ldots, x_{i_{\ell'}}\}$, $\ell' \leqslant \ell$, be the subset of $\{x_1, \ldots, x_n\}$ whose elements are contained in some path in $\mathcal{P}$. It readily follows that $B = \sum_{1 \leqslant j \leqslant \ell'} a_{i_j}$, since these paths use the arc (y_1, y_2) and the observed coverage of this arc is B. This leads to the desired subset of A of size B. □

Corollary 3. *Problem k-UTEO with fitting functions $f(x) = |x|$, or $f(x) = x^2$, is NP-hard.*

Proof: Problem k-UTEO with fitting function $f(x) = |x|$ is the same as Problem $(f_o, k, 1)$-GGMA, where

$$f_o(x, y) = \begin{cases} 0, & \text{if } y = 0, \\ |x - y|, & \text{otherwise.} \end{cases}$$

The claim follows since this function $f_o(\cdot, \cdot)$ satisfies Property 2. Analogously for $f(x) = x^2$. □

3 ALGORITHMS

3.1 The arc-width of a DAG

We start by introducing the graph parameter which will guide the dynamic programming algorithm.

Definition 1. *Given a DAG G, the* arc-width *of G, denoted $\langle G \rangle$, is the minimum number of directed paths that cover all arcs of G.*

For an example, see Fig. 4(a). By Dilworth's theorem [29], $\langle G \rangle$ also equals the size of the maximum set of arcs such that there is no directed path between any two of them. Moreover, by the constructive proof of [30], it can be computed in time $O(n^{5/2})$ by an application of a maximum matching algorithm [31].

We next introduce the notion of *rank* of a vertex in a DAG G, with the purpose of transforming G into an equivalent DAG $\widetilde{G}$ such that all arcs are between vertices of consecutive ranks. From this it will follow that we can base our DP algorithm by considering only k-tuples of vertices of the same rank. Moreover, it will hold that at most $\langle G \rangle$ vertices have the same rank in $\widetilde{G}$.

Definition 2. *The* rank *of a vertex x in a DAG G, denoted $rank(x)$, is the length of a longest directed path from x to a sink of G.*

See Fig. 4 for an example. The rank of every vertex of a DAG can be computed in time $O(m)$, by doing a topological sort of the vertices of G, and then assigning rank 0 to the sinks, and

$$rank(x) = 1 + \max_{y \in N^+(x)} rank(y),$$

to all other vertices x processed in inverse topological order, where $N^+(x)$ denotes the out-neighborhood of x.

Given a DAG G, let $\widetilde{G}$ denote the DAG obtained from G as follows. For every arc (u, v) such that $rank(u) > rank(v) + 1$, subdivide (u, v) into as many arcs as there are ranks between $rank(u)$ and $rank(v)$, see Fig. 4(b). Stated formally, remove arc (u, v), and add new vertices $z_1, \ldots, z_{rank(u)-rank(v)-1}$, and arcs $(u, z_1), (z_1, z_2), \ldots, (z_{rank(u)-rank(v)-1}, v)$. Observe that the endpoints of every arc of $\widetilde{G}$ now have consecutive ranks.

The following lemma places a bound on the number of vertices of each rank in $\widetilde{G}$ in terms of $\langle G \rangle$.

Lemma 1. *Let $s_r \geqslant 0$ be the number of sources of rank smaller than r in a DAG G. If G does not have isolated vertices, then for every $r \geqslant 0$ there are at most $\langle G \rangle - s_r$ vertices of rank r in $\widetilde{G}$.*

Proof: First observe that there can be no directed path between two vertices of the same rank, by the definition of rank.

Assume, for a contradiction, that there exists a set X of vertices of $\widetilde{G}$ having the same rank r, with $|X| \geqslant \langle G \rangle - s_r + 1$. Note that each of the s_r sources of rank strictly smaller than r has to be covered by a distinct path not passing through X. By the definition of arc-width, it follows that the vertices of X can be covered by at most $\langle G \rangle - s_r$ paths.

Since G has no isolated vertices, then $\widetilde{G}$ has no isolated vertices, and thus every vertex x in X has at least one in-coming or out-going arc a_x. The arc a_x can be covered only by a directed path passing through x. Moreover, there can be no directed path passing through two vertices in X; therefore, we have that each of the at least $\langle G \rangle - s_r + 1$ vertices of X has to be covered by a distinct path, a contradiction. □

Observe that the set of values that the $rank$ function takes on $\widetilde{G}$ equals the set of values that the $rank$ function takes on G. Moreover, the cardinality of this set of values is at most n, the number of vertices of G.

3.2 The dynamic programming algorithm

We will consider fitting functions satisfying Properties 1 or 2; accordingly, since we consider the p-norm, the maximum expression level of a path in an optimal

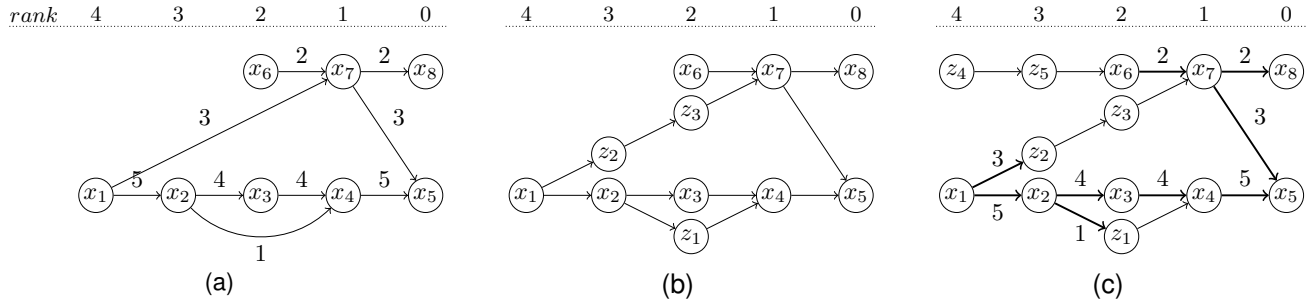


Fig. 4. In Fig. 4(a), an arc-weighted DAG G of arc-width $\langle G\rangle = 4$; the vertices of G are drawn from right to left, in increasing order on their ranks, indicated in the top of the figures. In Fig. 4(b), the transformed DAG $\widetilde{G}$ (we omit the arc weights from this figure). In Fig. 4(c), the DAG obtained by further transforming $\widetilde{G}$ as explained in the proof of Thm. 3; the arcs without weights belong to the set Y of arcs not contributing to the objective function.

solution is at most the maximum weight of an arc; denote this maximum value by W. Supposing that the expression levels are integers, then we can enumerate all k-tuples of expression levels in $\{1,\ldots,\lceil W\rceil\}$, and for each such tuple, find by dynamic programming the optimal k paths having these expression levels.

In practical terms, having these two steps separate means that we can employ any local search heuristic for finding the optimal expression levels. This search will be guided by the cost of the objective function returned by the dynamic programming; the search can be done at any chosen granularity of the expression levels, possibly including *a priori* information about the true expression levels. For example, in [1] we used a genetic algorithm.

In Sec. 3.3 we will show that we can discretize the weights in $\{1,\ldots,\lceil W\rceil\}$ and obtain an FPTAS for Problem (f_r,k,p)-GGMA

Theorem 3. *Given a DAG G and a k-tuple $(w_1,\ldots,w_k)$ of expression levels, we can find in time $O(\langle G\rangle^k(\langle G\rangle+k)n)$ the optimal paths for Problem (f,k,p)-GGMA having these expression levels.*

Proof: We further generalize the problem by considering also a set Z of arcs which are excluded from the evaluation of the objective function, i.e., their predicted coverage can take any value, independently of their observed coverage; this is done in order to accommodate the transformations we will apply to the graph. Throughout this proof we assume $p<\infty$; otherwise, it suffices to replace the summation operation with the one of taking the maximum.

Given an input DAG G, we construct the graph $\widetilde{G}$. The weights of $\widetilde{G}$ are the same as the weights of G, with the exception that if an arc (u,v) was subdivided into arcs $(u,z_1),(z_1,z_2),\ldots,(z_{rank(u)-rank(v)-1},v)$, then $w(u,z_1) = w(u,v)$, and all other arcs $(z_1,z_2),\ldots,(z_{rank(u)-rank(v)-1},v)$ are added to Z.

Denote by r_{max} the maximum rank of $\widetilde{G}$. We further transform $\widetilde{G}$ by making it such that all of its sources have rank r_{max}. This can be done by adding, for each source s of rank $r_s<r_{max}$, a directed path of length $r_{max}-r_s$ ending in s, whose starting point, thus, has rank r_{max}; the arcs of this path are added to Z (e.g., the path (z_4,z_5,x_6) ending in x_6 in Fig. 4(c)). The resulting graph $\widetilde{G}$ has at most $\langle G\rangle$ vertices at each rank, by Lemma 1.

Let $\widetilde{G}_r$ denote the subgraph of $\widetilde{G}$ induced by the vertices of rank at least r. We will solve the problem on the subgraphs $\widetilde{G}_{r_{max}},\widetilde{G}_{r_{max}-1},\ldots,\widetilde{G}_1,\widetilde{G}_0=\widetilde{G}$, as follows.

For every rank $r\in\{r_{max},\ldots,0\}$ we compute a table sol_r, which for every k-tuple $(v_1,\ldots,v_k)$ of sinks of $\widetilde{G}_r$ (that is, of vertices of rank r of $\widetilde{G}$), stores the value of the p-norm, raised to the power p, of the k-paths optimal for $\widetilde{G}_r$ and ending in $(v_1,\ldots,v_k)$. Stated formally,

$$\mathsf{sol}_r(v_1,\ldots,v_k) := \min_{\substack{\text{paths } P_1,\ldots,P_k \text{ in } \widetilde{G}_r,\\ \text{each } P_i \text{ is from a source to } v_i}} (\|err(P_1,\ldots,P_k,r)\|_p)^p,$$

where $err(P_1,\ldots,P_k,r)$ is the vector with value

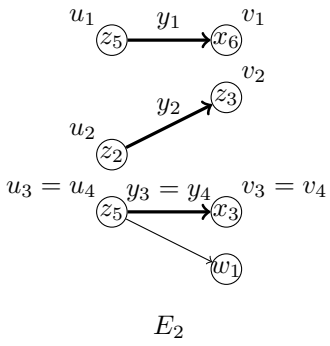
$$f\left(w(a),\sum_{j\in\{1\leqslant t\leqslant k\mid a\in P_t\}} w_j\right)$$

on the component corresponding to arc a of $\widetilde{G}_r$, for each arc a of $\widetilde{G}_r$.

The solution for Problem (f,k,p)-GGMA with the given expression levels will be obtained by taking the minimum over all k-tuples of sinks of $\widetilde{G}_0=\widetilde{G}$. In tables sol_r, we can also store the predecessors of the k-tuples of endpoints on the optimal paths, in order to retrieve these paths.

We initialize the table $\mathsf{sol}_{r_{max}}$ with 0, for every k-tuple of vertices of rank r_{max}. For every rank i, $r_{max}>i\geqslant 1$ in decreasing order, we initialize each entry in table sol_i by ∞, and compute it as follows.

Observe that the set of arcs between ranks $i+1$ and i, which we denote here E_i, has cardinality at most $\langle G\rangle$, from the fact that G is acyclic and by the



E_2

Fig. 5. The set E_2 of arcs from the vertices of rank 3 (on the left) to those of rank 2 (on the left) of the graph $\widetilde{G}$ from Fig. 4(c). In this particular example $k = 4$, and the first path uses the arc (z_5, x_6), the second uses the arc (w_2, w_3), while the third and the fourth paths both use the arc (w_5, x_3).

definition of rank. We enumerate all ways of assigning the arcs in E_i to the k solution paths, that is, through all k-tuples $(y_1, \ldots, y_k) \in (E_i)^k$ where $a \in E_i$ equals some y_j if arc a is assigned to the jth path. Note that this assignment induces an assignment $(u_1, \ldots, u_k)$ of the vertices of rank $i+1$ to the k solution paths, and an assignment $(v_1, \ldots, v_k)$ of the vertices of rank i to the k solution paths.

For each arc $a \in E_i \setminus Z$ we can then compute its predicted coverage by looking at which paths it belongs to in the assignment $(y_1, \ldots, y_k)$, and which are the weights of the corresponding paths in the given tuple $(w_1, \ldots, w_k)$.

If leading to a lower value, we update $\mathsf{sol}_i(v_1, \ldots, v_k)$ with $\mathsf{sol}_{i+1}(u_1, \ldots, u_k)$ plus the fitting function applied to the observed and predicted coverages of all arcs in $E_i \setminus Z$.

Graph $\widetilde{G}$ can be constructed in time $O(m)$. The update at each rank takes time $O(\langle G\rangle^k(\langle G\rangle + k))$, since each of the $O(\langle G\rangle)$ arcs in E_i must be inspected and the fitting function must be applied to their observed and predicted coverage. Since the maximum rank in $\widetilde{G}$ is at most the maximum rank in G, which is n, then the entire procedure takes time $O(\langle G\rangle^k(\langle G\rangle + k)n)$. □

Corollary 4. *If Properties 1 or 2 hold for the fitting function f, and the expression levels of the solution paths are allowed to take only integer values, then Problem (f, k, p)-GGMA can be solved in time $O(W^k\langle G\rangle^k(\langle G\rangle + k)n)$, where W is the maximum weight of an arc of the input DAG G.*

3.3 Some approximation results

Our first approximation result is a negative one, stating that for fitting functions $f(x) = |x|$, or $f(x) = x^2$, Problems k-UTEC or k-UTEO are hard to approximate. The proof of this fact is an immediate consequence of the reduction given in the proofs of Thms. 1 and 2.

Corollary 5. *For all $\alpha > 0$, there exists no polynomial-time approximation algorithm, with multiplicative approximation factor α, for Problems k-UTEC or k-UTEO with fitting functions $f(x) = |x|$, or $f(x) = x^2$, unless P = NP.*

Proof: In both reductions given in the proofs of Thms. 1 and 2, for fitting functions $f(x) = |x|$, or $f(x) = x^2$, an instance of the 3-PARTITION, or SUBSET SUM problem, respectively, is a 'yes' instance if and only if Problem k-UTEC or k-UTEO, respectively, admits a solution with total cost 0. □

Despite this hardness result, the DP algorithm in Sec. 3.2 can lead to an additive approximation algorithm, obtained by the simple strategy of discretizing the set of possible expression levels of the solution paths, according to an arithmetic progression. Given $\varepsilon > 0$, let

$$\mathcal{W} := \{1, \varepsilon W, 2\varepsilon W, 3\varepsilon W, \ldots, W\}.$$

Observe that the set $\mathcal{W}$ of approximated expression levels for the paths has cardinality $1 + 1/\varepsilon$. For every tuple $(w_1, \ldots, w_k) \in \{1, \ldots, W\}^k$ of exact expression levels for the k paths, there exists a tuple of expression levels $(w'_1, \ldots, w'_k) \in \mathcal{W}^k$ such that for every $1 \leqslant i \leqslant k$, it holds that

$$-\varepsilon W \leqslant w_i - w'_i \leqslant \varepsilon W.$$

For example, this strategy leads to the following approximation result for Problem (f_{abs}, k, ∞)-GGMA, where $f_{abs}(x, y) = |x - y|$. If OPT is the value of the objective function for the optimal paths of an optimal tuple of expression levels $(w_1, \ldots, w_k) \in \{1, \ldots, W\}^k$, and OPT' is the value of the objective function for the optimal paths with approximated expression levels $(w'_1, \ldots, w'_k)$, such that for every $1 \leqslant i \leqslant k$, $-\varepsilon W \leqslant w_i - w'_i \leqslant \varepsilon W$ holds, then it also holds that

$$OPT - \varepsilon W \leqslant OPT' \leqslant OPT + \varepsilon W.$$

Therefore, in order to find an εW-additive approximation for Problem $(|\cdot|, k, \infty)$-GGMA, we enumerate over k-tuples in $\mathcal{W}^k$, and for each such tuple compute the optimal paths using Thm. 3.

Corollary 6. *If the optimal solution for Problem $(|\cdot|, k, \infty)$-GGMA has cost OPT, then for every $\varepsilon > 0$, in time $O\big(\big(1 + \frac{1}{\varepsilon}\big)^k \langle G\rangle^k(\langle G\rangle + k)n\big)$, we can find a solution of cost OPT', such that*

$$OPT - \varepsilon W \leqslant OPT' \leqslant OPT + \varepsilon W,$$

where W is the maximum weight of an arc.

In the case of Problem (f_r, k, p)-GGMA, however, we can write a multiplicative approximation algorithm, using the same method, but this time using a geometric progression. Given $\varepsilon > 0$, let

$$\mathcal{W}' := \{1, (1+\varepsilon), (1+\varepsilon)^2, \ldots, (1+\varepsilon)^{\lfloor \log_{1+\varepsilon} W \rfloor}\}.$$

The set $\mathcal{W}'$ has cardinality $1+\lfloor \log_{1+\varepsilon} W \rfloor$. For every tuple $(w_1, \ldots, w_k) \in \{1, \ldots, W\}^k$ of exact expression levels for the k paths, there exists a tuple of weights $(w'_1, \ldots, w'_k) \in (\mathcal{W}')^k$ such that for every $1 \leqslant i \leqslant k$, it holds that

$$w_i \leqslant w'_i \leqslant (1+\varepsilon) w_i.$$

If OPT is the value of the objective function for the optimal paths of optimal expression levels $(w_1, \ldots, w_k)$, and OPT' is the value of the objective function for the optimal paths of approximated expression levels $(w'_1, \ldots, w'_k)$ such that for every $1 \leqslant i \leqslant k$, $w_i \leqslant w'_i \leqslant (1+\varepsilon) w_i$ holds, then it also holds that

$$\frac{1}{1+\varepsilon} OPT \leqslant OPT' \leqslant (1+\varepsilon) OPT.$$

Thus, just as before, we have

Corollary 7. *If the optimal solution for Problem (f_r, k, p)-GGMA has cost OPT, then for every $\varepsilon > 0$, in time $O((\log_{1+\varepsilon} W)^k \langle G \rangle^k (s+k)n)$, where W is the maximum weight of an arc, we can find a solution of cost OPT', such that*

$$\frac{1}{1+\varepsilon} OPT \leqslant OPT' \leqslant (1+\varepsilon) OPT.$$

Thus, when k is bounded, Problem (f_r, k, p)-GGMA admits an FPTAS, since the running time is then polynomial in n, $1/\varepsilon$ and $\log W$.

4 Experiments

One of these optimization objectives, namely the one in Problem k-UTEC, or equivalently Problem $(f_c, k, 1)$-GGMA, was already shown to be a relevant model on real instances in the conference version of this article [1].[3] There the RNA transcripts predicted by the cited dynamic programming algorithm for finding the k-paths solution were shown to be more accurate than those predicted by competing methods. As neither the optimization goals nor the implementation have changed, we do not repeat these tests verbatim here, but we concentrate on the running time, as we have made several improvements to the dynamic programming algorithms.

Namely, we conducted an undirect test to see the effect of the arc-width parameter $\langle G \rangle$ introduced in this article. In [1] the algorithms had an $\Omega(n^k)$ factor in the running time, which we improved here to $\Omega(\langle G \rangle^k)$. In order to see how $\langle G \rangle$ compares to n, we constructed splicing graphs based on simulated data for the same 1,462 genes of the human chromosome 2 as in the experiments of [1]. We repeated this experiment on a real RNA-seq dataset [GenBank:SRR065504] also used in [11] and [1], creating splicing graphs from all the reads that aligned to chromosome 2. Due to the sheer number of the created graphs, we filtered out those graphs that had only one vertex. We computed the arc-width of these graphs, and we illustrate the results Figs. 6 and 7; the arc-width parameter is clearly smaller than the number of vertices on average. For example, for a splicing graph on 50 vertices, the average arc-width is approximately 10 times smaller, both on simulated and real data. The effect to the practical running time of the implementation should also be noticeable: this is left as future work, see Discussion.

3. The corresponding implementation is available at http://sourceforge.net/projects/traph/.

5 Discussion

We focused here on exploring the complexity of the genome-guided multi-assembly problem when the number of allowed paths is bounded, which is the case of most practical interest. For this we took special care to define the problem as generally as possible, in order to show the full power of the applied dynamic programming approach and also to make the hardness results as strong as possible. The generality of the problem definition was exemplified by four quite different optimization objectives, and some further approximability results were derived for some of these.

We expect that the machinery developed in this paper for Problem GGMA to find applications in other multi-assembly -like problems, since, ultimately, this is a quite natural graph problem. Moreover, this is supported by the experimental results showing that the arc-width is small in practice.

Another fundamental aspect is that all the problems studied here assume pair-wise information on the possible consecutive exons in the transcript. However, with longer sequencing reads, one can obtain *subpath constraints* telling which exons should go together in a transcript. A subset of the authors of this article studied how to take this additional information into account in multi-assembly [32]. Those results apply for a version of the problem, where one optimizes only the transcript sequences to satisfy the subpath constraints, but not the coverage values. We are currently studying a combined problem formulation which takes both of these aspects into account. For this reason, we are investing in implementing the dynamic programming improvements of this article for this new combined problem formulation.

Acknowledgments

This work was partially supported by the Academy of Finland under grant 250345 (CoECGR). Travis Gagie was partially supported by the Academy of Finland under grant 268324. Alexandru Tomescu was partially supported by the Academy of Finland under grant 274977.

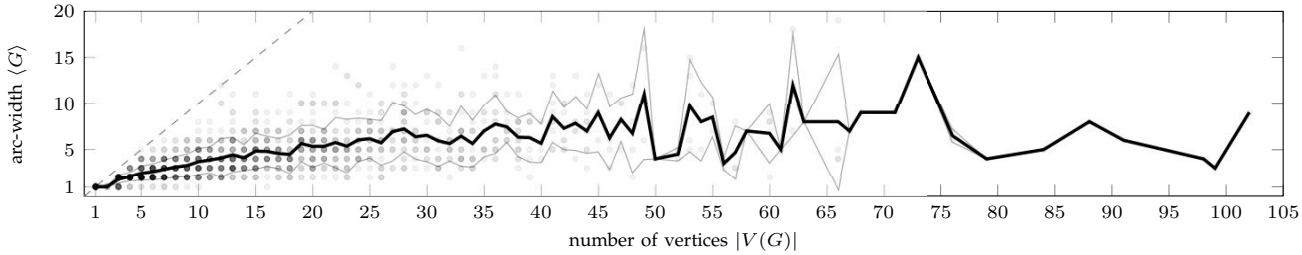


Fig. 6. Comparison of arc-width parameter and number of vertices on simulated data. The color intensity of the data points reflects their frequency in the dataset. The bold line shows the mean and the thinner lines the variance of the arc-width.

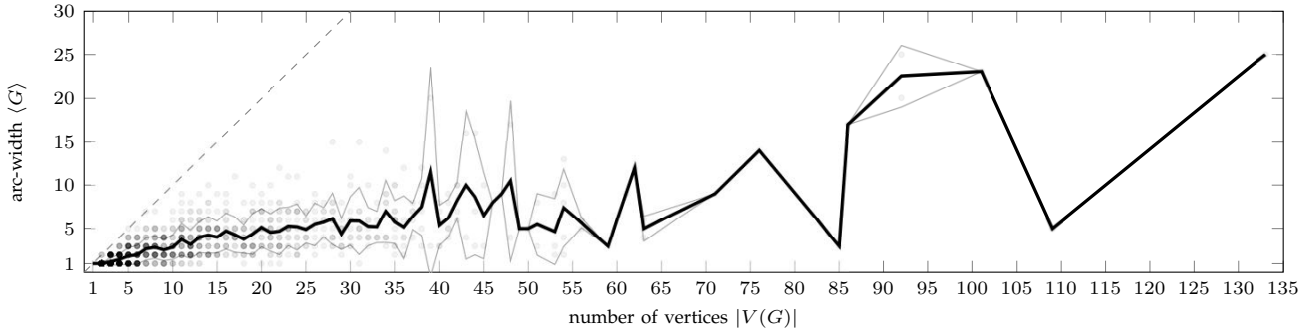


Fig. 7. Comparison of arc-width parameter and number of vertices on real data. The color intensity of the data points reflects their frequency in the dataset. The bold line shows the mean and the thinner lines the variance of the arc-width.

REFERENCES

[1] A. I. Tomescu, A. Kuosmanen, R. Rizzi, and V. Mäkinen, "A Novel Combinatorial Method for Estimating Transcript Expression with RNA-Seq: Bounding the Number of Paths," in *WABI 2013 – 13th Workshop on Algorithms for Bioinformatics*, ser. LNCS, vol. 8126, 2013, pp. 85–98.

[2] Y. Xing, A. Resch, and C. Lee, "The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures," *Genome Res*, vol. 14, no. 3, pp. 426–441, Mar. 2004.

[3] A. Mortazavi, B. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-Seq," *Nature Methods*, vol. 5, pp. 621–628, 2008.

[4] S. Pepke, B. Wold, and A. Mortazavi, "Computation for ChIP-seq and RNA-seq studies," *Nature methods*, vol. 6, no. 11, pp. s22–s32, 2009.

[5] F. Ozsolak and P. M. Milos, "RNA sequencing: advances, challenges and opportunities." *Nature reviews. Genetics*, vol. 12, no. 2, pp. 87–98, Feb. 2011.

[6] I. Birol, S. Jackman, C. Nielsen, J. Qian, R. Varhol, G. Stazyk, R. Morin, Y. Zhao, M. Hirst, J. Schein, D. Horsman, J. Connors, R. Gascoyne, M. Marra, and S. Jones, "De novo transcriptome assembly with ABySS," *Bioinformatics*, vol. 25, no. 21, pp. 2872–2877, Nov. 2009.

[7] C. Trapnell, L. Pachter, and S. L. Salzberg, "TopHat: discovering splice junctions with RNA-Seq," *Bioinformatics*, vol. 25, no. 9, pp. 1105–1111, 2009.

[8] K. F. Au, H. Jiang, L. Lin, Y. Xing, and W. H. Wong, "Detection of splice junctions from paired-end RNA-seq data by SpliceMap," *Nucleic Acids Res*, vol. 38, no. 14, pp. 4570–4578, Aug. 2010.

[9] C. Trapnell, B. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. van Baren, S. Salzberg, B. Wold, and L. Pachter, "Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation," *Nature Biotechnology*, vol. 28, pp. 511–515, 2010.

[10] J. Feng, W. Li, and T. Jiang, "Inference of isoforms from short sequence reads," in *RECOMB 2010*, ser. LNCS, B. Berger, Ed., vol. 6044. Springer, 2010, pp. 138–157.

[11] W. Li, J. Feng, and T. Jiang, "IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly," *J. Comput. Biol.*, vol. 18, no. 11, pp. 1693–1707, 2011.

[12] J. J. Li, C. Jiang, J. Brown, H. Huang, and P. Bickel, "Sparse linear modeling of next-generation mRNA sequencing (RNA-Seq) data for isoform discovery and abundance estimation," *Proc. of the National Academy of Sciences*, vol. 108, no. 50, pp. 19 867–19 872, 2011.

[13] Y.-Y. Lin, P. Dao, F. Hach, M. Bakhshi, F. Mo, A. Lapuk, C. Collins, and S. C. Sahinalp, "CLIIQ: Accurate Comparative Detection and Quantification of Expressed Isoforms in a Population," in *Proc. WABI 2012*, ser. LNCS, vol. 7534. Springer, 2012, pp. 178–189.

[14] M. Guttman, M. Garber, J. Z. Levin, J. Donaghey, J. Robinson, X. Adiconis, L. Fan, M. J. Koziol, A. Gnirke, C. Nusbaum, J. L. Rinn, E. S. Lander, and A. Regev, "Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs." *Nat Biotechnol*, vol. 28, no. 5, pp. 503–510, May 2010.

[15] A. M. Mezlini, E. J. Smith, M. Fiume, O. Buske, G. Savich, S. Shah, S. Aparicion, D. Chiang, A. Goldenberg, and M. Brudno, "iReckon: Simultaneous isoform discovery and abundance estimation from RNA-seq data," *Genome Research*, vol. 23, no. 3, pp. 519–529, 2012.

[16] S. Mangul, A. Caciula, S. Al Seesi, D. Brinza, A. R. Banday, and R. Kanadia, "An integer programming approach to novel transcript reconstruction from paired-end RNA-Seq reads," in *BCB*, S. Ranka and et al., Eds. ACM, 2012, pp. 369–376.

[17] Z. Xia, J. Wen, C.-C. Chang, and X. Zhou, "NSMAP: A method for spliced isoforms identification and quantification from RNA-Seq," *BMC Bioinformatics*, vol. 12, no. 1, pp. 162+, 2011.

[18] D. Hiller and W. H. H. Wong, "Simultaneous isoform discovery and quantification from RNA-seq," *Statistics in biosciences*, vol. 5, no. 1, pp. 100–118, May 2013.

[19] E. Bernard, L. Jacob, J. Mairal, and J.-P. Vert, "Efficient RNA isoform identification and quantification from RNA-Seq data with network flows," *Bioinformatics*, vol. 30, no. 17, pp. 2447–2455, 2014.

[20] S. Heber, M. Alekseyev, S. S.H., T. H., and P. P.A., "Splicing

graphs and EST assembly problem," *Bioinformatics*, vol. 18, no. suppl 1, pp. S181–S188, 2002.
[21] A. I. Tomescu, A. Kuosmanen, R. Rizzi, and V. Mäkinen, "A Novel Min-Cost Flow Method for Estimating Transcript Expression with RNA-Seq," *BMC Bioinformatics*, vol. 14, no. Suppl 5, p. S15, 2013, presented at RECOMB-Seq 2013, Beijing, China.
[22] M. Garber, M. G. Grabherr, M. Guttman, and C. Trapnell, "Computational methods for transcriptome annotation and quantification using RNA-seq," *Nature Methods*, vol. 8, no. 6, pp. 469–477, 2011.
[23] D. Brett, H. Pospisil, J. Valcárcel, J. Reich, and P. Bork, "Alternative splicing and genome complexity," *Nature Genetics*, vol. 30, no. 1, pp. 29–30, Dec. 2001.
[24] T. Maniatis and B. Tasic, "Alternative pre-mRNA splicing and proteome expansion in metazoans," *Nature*, vol. 418, no. 6894, pp. 236–243, 2002.
[25] L. M. McIntyre, K. K. Lopiano, A. M. Morse, V. Amin, A. L. Oberg, L. J. Young, and S. V. Nuzhdin, "RNA-seq: technical variability and sampling," *BMC Genomics*, vol. 12, no. 1, pp. 293+, Jun. 2011.
[26] B. Vatinlen, F. Chauvet, P. Chrétienne, and P. Mahey, "Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1390 – 1401, 2008.
[27] P. D. Grünwald, *The Minimum Description Length Principle*, ser. MIT Press Books. The MIT Press, December 2007, vol. 1, no. 0262072815.
[28] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1979.
[29] R. P. Dilworth, "A Decomposition Theorem for Partially Ordered Sets," *The Annals of Mathematics*, vol. 51, no. 1, 1950.
[30] D. R. Fulkerson, "Note on Dilworth's decomposition theorem for partially ordered sets," *Proceedings of the American Mathematical Society*, vol. 7, no. 4, pp. 701–702, 1956.
[31] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs," *SIAM J. Comput.*, vol. 2, no. 4, pp. 225–231, 1973.
[32] R. Rizzi, A. I. Tomescu, and V. Mäkinen, "On the complexity of minimum path cover with subpath constraints for multi-assembly," *BMC Bioinformatics*, vol. 15, no. Suppl 9, p. S5, 2014.

Alexandru I. Tomescu obtained his PhD in Computer Science from the University of Udine, Italy, in 2012. After spending six months at the Technical University Berlin, Germany, he joined the Genome-scale algorithmics group at the University of Helsinki, Finland, where he currently holds an Academy of Finland Postdoctoral Fellowship.

Travis Gagie received a Dr. rer. nat. in Bioinformatics in 2009 from Bielefeld University, Germany. After three years at the University of Chile and Aalto University, Finland, in 2013 he moved to the Genome-scale algorithmics group at the University of Helsinki as a Postdoctoral Research Fellow, funded by the Helsinki Institute for Information Technology and the Academy of Finland.

Alexandru Popa obtained his PhD in Computer Science from the University of Bristol, UK, in 2011. Then, he was a Postdoctoral Researcher at Aalto University from 2011 to 2013. From September 2013 to January 2015 he was an Assistant Professor at Masaryk University, Brno, Czech Republic. Currently, he is an Assistant Professor at Nazarbayev University, Astana, Kazakhstan.

Romeo Rizzi received in 1997 a Ph.D. in Computational Mathematics and Informatics from the University of Padova, Italy. He held Postdoc and other positions at research centers like CWI (Amsterdam, Netherlands), BRICS (Aarhus, Denmark) and IRST (Trento, Italy), University of Trento and University of Udine, Italy. Since 2011 he is Associate Professor at the University of Verona. He has a background in Operations Research and his main interests are in Combinatorial Optimization and Algorithms. He is an Area Editor of 4OR. He published more than 70 research papers in a broad range of scientific journals in the areas of Discrete Mathematics, Combinatorics, and Algorithms. Including also research papers in refereed conference proceedings, the number of his scientific publications is well over one hundred. Since 2004, he has intensively acted as a trainer of the Italian team for the iOi.

Anna Kuosmanen obtained her M.Sc. in Bioinformatics from University of Helsinki, Finland, in 2013. She is currently pursuing her PhD at University of Helsinki and working in the Genome-scale algorithmics group.

Veli Mäkinen finished his PhD studies in Computer Science in 2003 at the University of Helsinki, Finland. He worked as a Postdoctoral Researcher (2004-2005) at Bielefeld University, Germany, and then back in Helsinki as Postdoctoral Research Fellow (2005-2007) and Academy Research Fellow (2007-2010). In 2010, he was appointed as a Professor in computer science at the University of Helsinki. Veli Mäkinen now heads the Genome-scale algorithmics research group as part of the Center of Excellence in Cancer Genetics Research.