JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 29, Number 2, 2022
Mary Ann Liebert, Inc.
Pp. 121–139
DOI: 10.1089/cmb.2021.0444

Deriving Ranges of Optimal Estimated Transcript Expression due to Nonidentifiability

HONGYU ZHENG,[1,*,i] CONG MA,[2,*] and CARL KINGSFORD[1]

ABSTRACT

Current expression quantification methods suffer from a fundamental but under-characterized type of error: the most likely estimates for transcript abundances are not unique. This means multiple estimates of transcript abundances generate the observed RNA-seq reads with equal likelihood, and the underlying true expression cannot be determined. This is called nonidentifiability in probabilistic modeling. It is further exacerbated by incomplete reference transcriptomes where reads may be sequenced from unannotated transcripts. Graph quantification is a generalization to transcript quantification, accounting for the reference incompleteness by allowing exponentially many unannotated transcripts to express reads. We propose methods to calculate a "confidence range of expression" for each transcript, representing its possible abundance across equally optimal estimates for both quantification models. This range informs both whether a transcript has potential estimation error due to nonidentifiability and the extent of the error. Applying our methods to the Human Body Map data, we observe that 35%–50% of transcripts potentially suffer from inaccurate quantification caused by nonidentifiability. When comparing the expression between isoforms in one sample, we find that the degree of inaccuracy of 20%–47% transcripts can be so large that the ranking of expression between the transcript and other isoforms from the same gene cannot be determined. When comparing the expression of a transcript between two groups of RNA-seq samples in differential expression analysis, we observe that the majority of detected differentially expressed transcripts are reliable with a few exceptions after considering the ranges of the optimal expression estimates.

Keywords: alternative splicing, expression quantification, nonidentifiability and differential expression, uncertainty.

[1]Computational Biology Department, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.
[2]Computer Science Department, Princeton University, Princeton, New Jersey, USA.
*These authors contributed equally to this work.
[i]ORCID ID (https://orcid.org/0000-0002-7668-2090).
An earlier draft of this article was posted as a preprint at bioRxiv (DOI: 10.1101/2019.12.13.875625).

1. INTRODUCTION

Despite the improvements of transcript expression estimation methods based on RNA-seq data (Li and Dewey, 2011; Hensman et al., 2015; Bray et al., 2016; Patro et al., 2017), the estimated transcript expression can still be inaccurate and uncertain. One source of uncertainty in expression estimation is that multiple sets of expression estimates can optimally explain the observed RNA-seq reads. Therefore, the ''best'' estimation cannot be uniquely identified. The state-of-the-art methods to quantify transcripts' expression are based on probabilistic models, and, in probabilistic model inference terminology, the phenomenon of nonuniqueness in optimal parameters under infinite data is called model ''nonidentifiability.''

In this work, we relax the concept and use this term to refer to the nonuniqueness of optimal parameters under a given finite data set. See Figure 1 for a toy example of model nonidentifiability in expression quantification. The two main problems for evaluating the accuracy of transcript expression estimates under nonidentifiability are (1) detecting the transcripts whose expression estimates are nonidentifiable and (2) bounding the range of the uncertain expression of the transcripts.

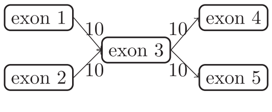
Expression of transcripts is used in many analyses, and understanding the accuracy and uncertainty of the estimated expression helps us evaluate the confidence of the conclusions of such analyses. Transcript expression estimates are used for detecting splicing isoform switching (Nowicka and Robinson, 2016; Guo et al., 2017; Vitting-Seerup and Sandelin, 2017), for identifying differential expression (McCarthy et al., 2012; Love et al., 2014; Ritchie et al., 2015; Soneson et al., 2015; Pimentel et al., 2017), and for predicting disease status and treatment outcome (Morán et al., 2012; Hoadley et al., 2014). Ranges of uncertain expression estimates provide useful insights into the reliability of these studies.

We focus on the uncertainty of expression estimates due to model nonidentifiability, but there are other causes of estimation uncertainty or inaccuracy. The small sample sizes (low sequencing depth of RNA-seq) also lead to estimation errors in transcripts' expression. Statistical methods such as bootstrapping and Gibbs sampling (Turro et al., 2011; Glaus et al., 2012; Al Seesi et al., 2014) provide an estimate on the error in expression levels due to the sample size. This estimation error can be reduced by increasing the sequencing depth. In contrast, the estimation uncertainty due to model nonidentifiability is more fundamental because it cannot be addressed even under infinite sample size.

In RNA-seq data, exon sequences are usually shared among multiple transcripts, and RNA-seq reads usually cannot be mapped to a unique transcript. Due to the high rate of multimapping events in RNA-seq data, the best set of transcripts' expression cannot be uniquely resolved, and the phenomenon of nonidentifiability occurs. Multimapped reads are prevalent in RNA-seq data regardless of the sequencing depth. Thus, the uncertainty of expression due to model nonidentifiability cannot be easily addressed.

Previous works have analyzed the nonidentifiability phenomenon in transcript expression quantification. However, they mainly focused on the first problem, to identify the transcripts for which the expression estimates are nonidentifiable, but not the second, which is to bound the ranges of the true expressions for the transcripts with uncertain expression estimates. Lacroix et al. (2008) and Hiller et al. (2009) developed methods to list the transcripts that have nonunique expression estimates, but their methods do not provide information about the values of optimal abundances. Roberts and Pachter (2013) incorporated the detection of nonidentifiable abundances into their quantification method, and designed a tool to output an identifiability flag for each transcript along with a single-expression estimate.

In this work, we develop methods that address the range of optimal expression estimates for transcripts under model nonidentifiability. For each transcript, we calculate the minimum and



$$\begin{cases} \text{abundance}(1-3-4) = 10 \\ \text{abundance}(2-3-5) = 10 \end{cases}$$

$$\begin{cases} \text{abundance}(1-3-5) = 10 \\ \text{abundance}(2-3-4) = 10 \end{cases}$$

FIG. 1. Example of a nonidentifiable quantification model. Transcripts are the paths in the splice graph, denoted by the concatenation of exon indices. The number on each edge indicates the number of observed reads mapped to this splice junction. The set of transcript abundances is optimal if it perfectly explains the observed reads. That is, for each junction, the total abundances of transcripts containing that junction sum up to the number displayed on the edge. The right side of the figure shows two co-optimal expression abundances. It can be verified that both solutions explain the observed reads perfectly as they both predict 10 reads on each junction.

maximum values across all optimal expression estimates. That is, for any expression value between the computed minimum and maximum, there exists a set of expression of the other transcripts such that the estimation of all transcripts' expression (combining both the expression of this transcript and the other transcripts) leads to the largest likelihood in the probabilistic model of expression quantification. Compared with a list of names of the transcripts for which the expressions are nonidentifiable, the range of optimal expression of a transcript provides more information on the accuracy (or inaccuracy) of the estimate.

Most widely used quantification software (Li and Dewey, 2011; Hensman et al., 2015; Bray et al., 2016; Patro et al., 2017) take a set of reference transcript sequences as input and assume that the reference transcripts are the complete set of sequences that can be expressed. This is called reference-transcript-based expression quantification. Another line of expression quantification models (LeGault and Dewey, 2013; Bernard et al., 2014; Ma et al., 2020) called ‘‘graph quantification'' assumes that the current reference transcriptome database is incomplete. Instead, the splice graph that encodes the exon/exon connection relationships is assumed to be correct and complete, meaning every possibly expressed transcript corresponds to a path in the splice graph and vice versa. Those models infer the splice graph edge expression or edge selection propensity in the quantification probabilistic models.

We provide a more detailed overview in Section 2.3. Many transcript assembly methods also adopt a similar setup, assuming a mixture of reference transcripts and novel isoforms are expressed (Trapnell et al., 2010; Tomescu et al., 2013; Pertea et al., 2015; Liu and Dickerson, 2017; Gatter and Stadler, 2019).

We develop methods to bound the range of optimal expression estimates for both reference-transcript-based and graph quantification models. Our method for the reference-transcript-based quantification is based on linear programming over sufficient statistics (Section 2.2), and our method for graph quantification is based on max-flow formulations to ‘‘introspect'' the graph quantification model (Section 2.4). Our introspection algorithm can not only bound the uncertain expression of full transcripts, but also extends to graph structures. For example, given a set of edges, our method computes the range of the optimal total expression of transcripts that cover any edge in the set.

Combining our methods for quantification models and interpolating between the complete and incomplete reference transcriptome assumption, we can additionally compute the range of optimal expression estimates under the assumption that a given percentage of the expression comes from the reference and the remaining expression comes from the full paths in splice graphs (Section 2.5).

Applying our method to 16 Human Body Map samples, we analyze to what degree the expressions of transcripts are estimated inaccurately due to nonidentifiability. We observe that around 35%–50% of transcripts potentially suffer from expression estimation error across the 16 samples. Most of these transcripts (or 20%–47% of total transcripts) have very uncertain expression estimates such that the ranking of expression between the transcript and its sibling isoforms from the same gene is inconclusive. Around half of the transcripts with uncertain expression estimates due to nonidentifiability are different from those due to finite sample size.

Applying our method on sequencing data sets of an MCF10 cell line and of CD8 T cells, we use the ranges of optima to evaluate the reliability of detected differentially expressed (DE) transcripts within each data set. A DE detection is unreliable if the ranges of optimal expression between the DE groups largely overlap. We observe that the majority of the DE calls are reliable and robust to the uncertain expression estimation due to nonidentifiability when the reference transcriptome contributes to >40% of expression. However, there are 5 unreliable DE calls (out of 257 detections) in the MCF10 data set, and 19 unreliable DE calls (out of 3152 detections) in the CD8 T cell data set. It requires further investigation to determine whether these transcripts are actually DE, and analyses based on the DE status of these transcripts require extra caution.

2. METHODS

This study does not collect new biological data, and thus IRB approval is not required.

We start with relevant definitions in Section 2.1. Section 2.2 provides a high-level overview of probabilistic modeling of transcript quantification, and the linear programming to derive a range of optimal abundance estimates for this setup. Section 2.3 provides an overview of graph quantification, and Section 2.4 describes our introspection algorithm to derive the ranges under this setup.

2.1. Definitions

A ''splice graph'' is a directed acyclic graph representing alternative splicing events in a gene. The graph has two special vertices: S represents start of transcripts and T represents termination of transcripts. Every other vertex represents a (partial) exon. Edges in the splice graph represent ''splice junctions,'' that is, potential adjacency between exons in transcripts. Each transcript corresponds to a unique $S-T$ path in the splice graph, and the set of known transcripts is called the ''reference transcriptome.'' $S-T$ paths that are not present in the reference transcriptome correspond to ''unannotated transcripts.'' We use the phrase ''quantified transcript set'' to denote a set of transcripts with corresponding abundances.

2.2. Ranges of optimal estimates for reference-transcript-based quantification

Recall the problem of reference-transcript-based expression quantification. We focus on paired-end reads for now, however, this formulation naturally extends to other types. We also focus on a particular probabilistic modeling, which lies at the heart of most modern transcript quantifiers (Li and Dewey, 2011; Hensman et al., 2015; Bray et al., 2016; Patro et al., 2017). Assume that the paired-end reads from an RNA-seq experiment are error-free and uniquely aligned to a reference genome as fragments. We denote the set of fragments as F, and the set of transcripts as $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$ with abundance (copies of molecules) $c_1, c_2, \dots, c_n$. The probability of observing F is as follows:

$$P(F|\mathcal{T}, \{c_i\}) = \prod_{z \in F} \sum_{i \in A_T(z)} P(T_i)P(z|T_i).$$

To generate a fragment, first a transcript is sampled, then the fragment is sampled from the selected transcript, and we only observe the resulting fragments. $P(T_i)$ denotes the probability of sequencing a fragment from transcript T_i, and $P(z|T_i)$ denotes the probability of sampling the fragment z given that it comes from T_i. $A_T(z)$ is the set of transcript indices onto which z can map. Usually, $P(T_i) \propto c_i$, and $P(z|T_i)$ is a known quantity derived from fragment length distribution, bias correction, and similar factors. Recent transcript quantifiers (Bray et al., 2016; Patro et al., 2017) use sophisticated techniques to ensure accurate estimation of $P(z|T_i)$ and fast inference of $\{c_i\}$.

We now introduce the idea of reparameterization. Assume, for now, each fragment spans exactly one junction. If two sets of quantified transcripts result in exactly the same total abundance on each junction, they yield the same generative model. In other words, these two sets will be indistinguishable from each other. Let $\{g_i\}$ be the set of total abundances for each junction J_i, and we use $J_i \in T_j$ to denote that junction J_i is in transcript T_j. The following condition is sufficient to guarantee that the set of transcript abundances $\{c'_i\}$ is valid, and indistinguishable from $\{c_i\}$:

$$\sum_{j:J_i \in T_j} c_j = \sum_{j:J_i \in T_j} c'_j, \quad \text{for each } J_i,$$
$$c'_j \geq 0, \quad \text{for each } T_j.$$

As this defines a linear system, we calculate the maximum and minimum possible value of c'_j for each transcript T_j, which would be the confidence interval for this transcript assuming $\{c_i\}$ is an optimal solution for the inference, which can be readily obtained using existing software. More specifically, we use either $\max c'_j$ or $\min c'_j$ as the sole optimization objective with the linear system as the constraints, then solve the resulting linear program, for each transcript of interest. It is an interval because for every abundance value in between the extremes, there exists an optimal solution that allocates this exact abundance to the transcript.

This argument no longer holds when fragments can span multiple junctions. However, as we have shown previously (Ma et al., 2020), to ensure that two sets of quantified transcripts yield the same fragment distribution, it suffices that for each phasing path (i.e., the set of junctions in a fragment, as a path on the splice graph), the total abundance of transcripts containing the phasing path in full is equal. In other words, the form of the linear program remains unchanged, with the only change being that J_i is now a phasing path instead of a single junction.

2.3. Bird's-eye view: graph quantification

We now provide a high-level overview of graph quantification, and draw similarities with its transcript-based counterpart. The problem of splice graph expression quantification is very similar to the transcript version of the problem, but with the variables associated with transcripts $\{T_i\}$, $\{c_i\}$ replaced by a graph $G=(V, \{e_i\})$ and a network flow $\{f_i\}$ on the graph. We start by assuming that G is the splice graph in the usual sense, and every fragment z is a junction read, which can be mapped to an edge in G. Under an analogous probabilistic model, the probability of observing F is as follows:

$$P(F|G, \{f_i\}) = \prod_{z\in F} \sum_{i\in A_G(z)} P(e_i)P(z|e_i).$$

To generate a fragment, first an edge in G is sampled, then the fragment is sampled from the selected edge, and we only observe the resulting fragments. Again, $P(e_i)$ denotes the probability of sampling a fragment from edge e_i, and $P(z|e_i)$ denotes the probability of sampling the fragment z given it comes from e_i. $A_G(z)$ denotes the set of edges from which z can be generated. We also have $P(e_i) \propto f_i$ up to a normalization factor, and $P(z|e_i)$ being a known quantity.

After inferring the splice flow value from the set of observed fragments, we obtain the optimal splice graph flow that explains the observation. While the graph flow itself finds use in certain cases, for most analysis it is imperative that the flow will be decomposed into a weighted set of $S-T$ paths, corresponding to a quantified set of transcripts. Importantly, the flow decomposition is not unique in many cases, and while these decompositions lead to a different set of transcripts, they define the same generative model. This is a direct manifestation of nonidentifiability, if we consider the transcript abundances to be the variables of the probabilistic model.

In the next section, we describe methods that consider all possible flow decompositions of a given graph flow. As a particular use case, these methods are able to determine the minimal and maximal possible abundance for a given transcript, across all possible flow decompositions.

Similarly to the transcript-based quantification, the conclusion no longer holds when fragments span multiple junctions, and it is much harder to adjust the graph quantification model to properly take these into account. However, for graph quantification, G can be any supergraph of the splice graph, as long as each $S-T$ path in G corresponds to a valid transcript to generate fragments, and each valid transcript corresponds to a unique $S-T$ path in G. Several approaches are possible, and we will use the one outlined in our previous work (Ma et al., 2020), which properly handles phasing reads.

2.4. Subgraph quantification

Nonidentifiability manifests itself in graph quantification as different flow decompositions from the inferred splice graph flow. Specifically for a transcript, the corresponding $S-T$ path can be assigned with different weights from different flow decompositions. Similar to Section 2.2, we only need to know the minimal and maximal possible weight to construct the confidence interval.

In addition, we are also interested in the related problem of ‘‘local quantification,’’ that is, to estimate the abundance of a specific splicing pattern by aggregating expression of full-length transcripts containing the pattern. This is natural for analysis of complex gene loci, where certain splicing patterns within a region might be of larger interest than the patterns outside. Our reasoning for deriving a ‘‘confidence interval’’ can similarly apply for splicing patterns, that is, we are interested in the total weight of $S-T$ paths containing a specific splicing pattern as a confidence interval. Since a transcript consisting of s exons is a combination of $s-1$ consecutive junctions, we are interested in splicing patterns that are defined by co-occurrence of k disjoint junctions. This motivates the following formal definition of the subgraph quantification problem in graph theory language, generalizing the transcript confidence interval calculation to splicing patterns:

Definition 1 (AND-Quant). *Let $G=(V, E)$ be a directed acyclic graph with an edge flow, and $\{E_k\}$ be a list of edge sets with the well-ordering property: If a path visits $e_i \in E_i$, then visits $e_j \in E_j$ at a later step, it must be that $i<j$. An $S-T$ path is ‘‘good’’ if it intersects each E_k. For a flow decomposition of G, the total ‘‘good flow’’ is the total flow from good $S-T$ paths.* AND-Quant$(G, \{E_k\})$ *asks for the minimum and maximum total good flow for all decompositions of G.*

For quantifying a full transcript, E_k consists of single-edge sets, each corresponding to an edge in the $S-T$ path representing the transcript. The range of optima for this transcript is exactly AND-Quant$(G, \{E_k\})$. For the aforementioned ‘‘local quantification,’’ the definition of $\{E_k\}$ depends on the specific construction of the graph quantification instance, but in general it is easy to construct one E_i for each junction that satisfies the well-ordering property.

To solve AND-Quant, the first step is to define a similar problem called OR-Quant as follows:

Definition 2 (OR-Quant) *Let $G=(V,E)$ be a directed acyclic graph with an edge flow, and E' be an arbitrary subset of E. An $S-T$ path is ''good'' if it intersects E'. The total ''good flow'' and the objective are defined in the same way as in* AND-Quant.

The OR-Quant problem is complementary to AND-Quant and is interesting on its own, as it is suitable to represent analyses where we are interested in aggregated expression that includes any of the several junctions or exons. We now convert an instance of AND-Quant to OR-Quant by constructing a graph G_B, called the block graph, that contains every edge that is either in one of E_k, or is between an edge in E_{k-1}, and an edge in E_k for $1 \le k \le m$, where m is the number of sets in $\{E_k\}$ (with some abuse of notion, assume E_0 consists of a virtual edge ending at S, and E_{m+1} consists of a virtual edge starting at T). We claim the following for AND-Quant:

Lemma 1. *An $S-T$ path is ''good'' in the sense of Definition 1 if and only if it is a subgraph of G_B.*

If we know the minimum and maximum total ''bad flow'' (negative of good flow), we can obtain the answer to AND-Quant by complementing the result with U, the total flow of G. From the lemma, an $S-T$ path is bad if and only if it intersects with $G-G_B$, which turns the problem into an instance of OR-Quant.

We now solve OR-Quant. Recall E' is the set of edges that a ''good'' path needs to intersect. To that end, we define two auxiliary graphs. G^- is a copy of G without the edges in E'. G^+ is a copy of G with an extra vertex T', which replaces T as the flow sink, and all edges in E' have their destination moved to T'. We claim that running MaxFlow on both graphs yields the solution:

Theorem 1. *Let G^+ and G^- be constructed as described above. Then* OR-Quant$(G,E')=$ $[U-$ MaxFlow(G^-), MaxFlow$(G^+)]$.

Constructing G^+ and G^- takes linear time, so the time complexity of OR-Quant depends only on the time to solve the MaxFlow. Combining the arguments leads to the solution to AND-Quant (recall U is the total flow on G):

Theorem 2. *Let G_B be the block graph, and $[l,\ r]=$*OR-Quant$(G,\ G-G_B)$. *Then* AND-Quant$(G,\{E_i\})=[U-r,\ U-l]$.

See Figure 2 for an example of both problems. In the following three sections, we provide proofs to Theorems 1 and 2, as well as an algorithm to construct G_B in linear time.

2.4.1. Algorithms for OR-Quant. In this section, we state the algorithms for OR-Quant from a pure graph theoretical perspective. Recall our setup consists of a directed acyclic graph (DAG) G with predetermined source S and sink T, a flow $F=\{f_e\}$ on the graph, and an edge subset. We also use $C(F)$ to denote the total flow, and in this way we write $U=C(F)$.

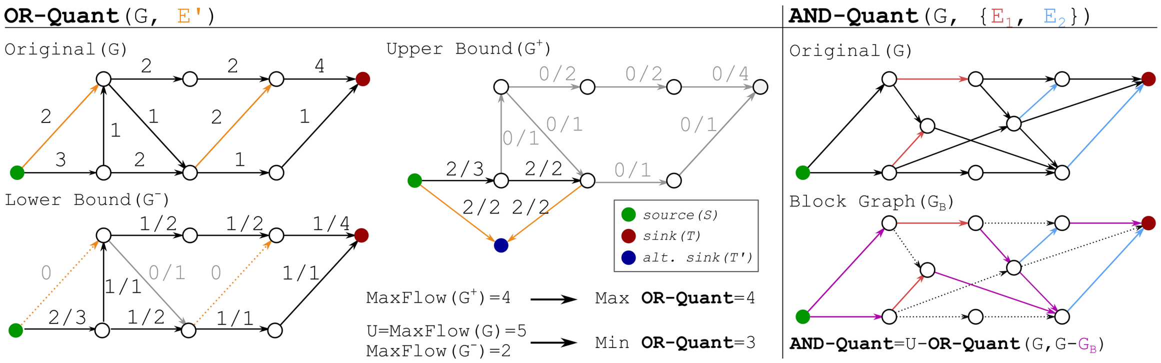


FIG. 2. Examples for subgraph quantification. Left: Example for OR-Quant, showing G and the auxiliary graphs G^+ and G^-. Right: Example for AND-Quant, showing G and the block graph G_B. As noted in the figure, G_B consists of all colored/nondashed edges in the image.

We start with several basic definitions.

Definition 3 (Flow). *Given DAG G with predetermined source S and sink T, a flow F is a set of edge weights of G satisfying non-negativity on every edge weight and flow balance on every vertex except S and T.*

Definition 4 (Decompositions). *Given graph G and a flow $F=\{f_e\}$, a decomposition R is written as $\{T_i, c_i\}$ where each T_i is an $S-T$ path on G and c_i is a nonnegative number, satisfying $\sum_{i:\, e\in T_i} c_i = f_e$ for every edge e. We use $|R|$ to denote the number of $S-T$ paths in R, and $C(R)=\sum c_i$ to denote the total flow/capacity of the decomposition.*

Definition 5 (Partial Decompositions). *A partial decomposition of a set of non-negative edge weights $W=\{w_e\}$ on a graph G is written as $\{T_i,\ c_i\}$ where each T_i is again an $S-T$ path on G, and c_i is a non-negative number satisfying $\sum_{i:\, e\in T_i} c_i \leq w_e$ for every edge e. Alternatively, if W is a flow, a partial decomposition can simply be defined as a subset of a (full) decomposition of F.*

We define partial decompositions on arbitrary non-negative edge weights, as required for a later proof. We present the following lemma without proof.

Lemma 2 (Finite Decomposition). *Every flow on a graph of finite size has a decomposition of finite size.*

We now restate the definition of OR-Quant slightly more formally.

Definition 6. *Let $G=(V,E)$ be a DAG with a flow F, and $E' \subseteq E$. An $S-T$ path is good if it intersects E'. (We also say the path is bad if it does not intersect E'.) For a decomposition $R=\{T_i,\ c_i\}$, the total good flow is defined as $Q(R)=\sum_{i:|T_i\cap E'|>0} c_i \leq C(R)$, that is, total flow from good paths in the decomposition.* OR-Quant*(G, E′) asks for the minimum and maximum $Q(R)$ for any possible decomposition R of F.*

We now state our algorithms and prove the correctness of the algorithms, starting from the lower bound.

Theorem 3 (Diff-Flow). *The auxiliary graph G^- is built with edge set $E-E'$. $Z=U-$* MaxFlow*(G^-) is the minimum total good flow $Q(R)$ from any possible decomposition R of F.*

Proof. We prove the theorem in two parts: First, we show there is a decomposition r such that $Q(R)=Z$, then we show that any decomposition satisfies $Q(R)\geq Z$.

The key observation is that all $S-T$ paths that are bad are fully in G^-. For the first part, fix a maximum flow of G^- as F^-. We let R^- be an arbitrary decomposition of F^-, and R^+ be an arbitrary decomposition of $F-F^-$.

We have $Q(R^-)=0$ because no path in R^- intersects with E'. We now claim $Q(R^+)=Z$. Because $C(R^+)=C(R)-C(R^-)=Z$, $Q(R^+)=Z$ if and only if every path in R^+ intersects with E'. Assuming otherwise, we can remove that path from R^+ and add it to R^-, which leads to larger $C(R^-)$. This is impossible: it implies F^- is not the MaxFlow of G^-, because the flow of R^- is a strictly better solution. Having proved $Q(R^+)=Z$, we now know R^-+R^+, a (full) decomposition, has exactly Z total good flow.

Now for any decomposition R, we split it into two parts. R^+ contains all good paths in R, and R^- contains all bad paths in R. We have $C(R^-) \leq$ MaxFlow(G^-), because the flow of R^- is a flow on G^-. Because $Q(R^-)=0$, we have $Q(R)=Q(R^+)=C(R^+)\geq Z$. □

Theorem 4 (Split-Flow). *The auxiliary graph G^* is built by adding a vertex T' in G and, for each edge in E' changing its destination to T' (we change it from G^+ to G^* for technical reasons here). T' replaces T as the sink of flows. $Y=$*MaxFlow*(G^*) is maximum total good flow for any possible decomposition of F.*

Proof. We prove the theorem in a similar style: First, we show any decomposition satisfies $Q(R)\leq Y$, then that there is a decomposition R of F such that $Q(R)=Y$.

For the first part of the theorem, for a decomposition R of F, we can write $R=R^+ + R^-$ where R^+ contains all good paths in R and R^- contains all bad paths in R. We can map R^+ to G^* and denote the resulting partial decomposition R^* (R^* is indeed a partial decomposition by noting there is a one-to-one mapping between edges of F^* and edges of F). We have $Q(R)=Q(R^+)=C(R^+)=C(R^*)\leq$ MaxFlow(G^*).

The second part of the proof, that is, to show there is a decomposition R of F such that $Q(R)=Y$, is more involved. Thus, we provide an overview before proceeding to the actual proof.

2.4.1.1. Overview

This part of the proof involves actually constructing the ‘‘correct'' decomposition of R such that $Q(R)$ is correct. We only know that there exists a MaxFlow with correct flow value on G^*, and we can decompose this flow. However, this only gets us a ‘‘truncated'' decomposition, where all paths end in an edge in E' instead of T. Thus, our task will be to complete each of these paths in the decomposition by extending them to T, which yields the correct decomposition as we desire. We cannot arbitrarily extend the paths as we also need to respect the flow capacity on F.

So, we may need to further split the weight of a path in the truncated decomposition when trying to complete it. Our constructive proof follows a strict protocol of doing these splits and completions, one path at a time, so flow capacity is respected in every step.

We start by building an arbitrary decomposition R^* from a maximum flow of G^*. However, R^* is not a valid partial decomposition of F, because G^* and G have different sets of edges. Our goal is to obtain a partial decomposition R^+ of F that is a ‘‘reconstruction'' of R^*. We will define several terms for convenience of our proof.

2.4.1.2. Path mapping for G^*

A good $S-T$ path on G is mapped to an $S-T'$ path on G^* by finding the first edge of the path that is in E', changing the destination of that edge to T', and discarding the edges after that so that the path ends at T'. (We will never map a bad $S-T$ path on G this way.)

Similarly, an $S-T'$ path on G^* is mapped to a (incomplete) path on G by moving the destination of last edge (that was T') back to its original node before the transformation. We assume a label containing its original destination is kept on each edge that ends at T'. This implies that multiple edges can exist between a node and T', and the movement follows the label. The resulting path is guaranteed to intersect with E', but is not a complete $S-T$ path.

We apply an induction argument, formally defined as follows. Figure 3 provides a concrete example for the induction argument.

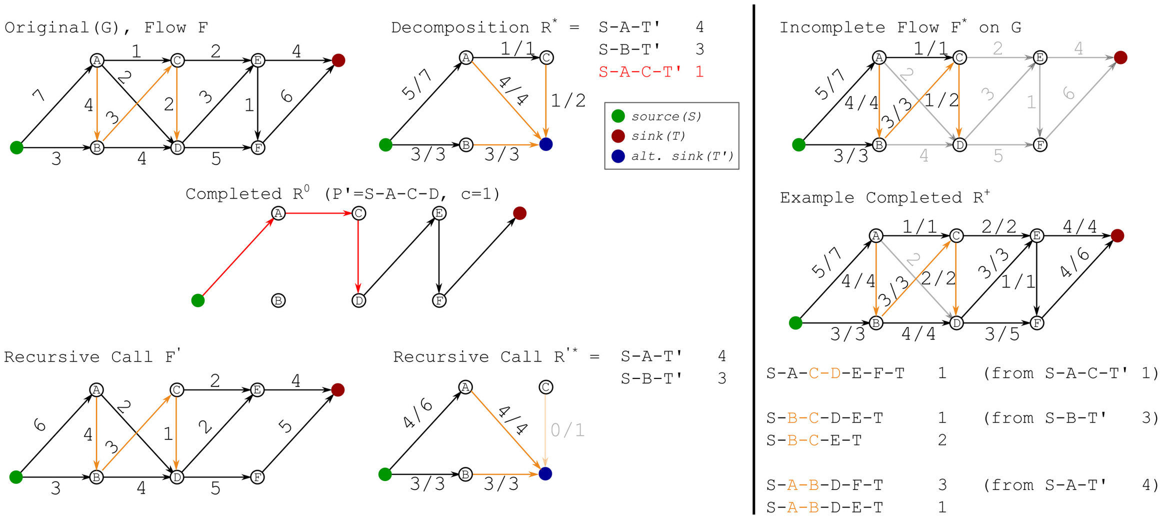


FIG. 3. Example for flow reconstruction. Input graph G is shown on the top left, with edges in E' marked in orange. The vertexes $A-F$ are in topological order. On the left-hand side, we show one step of the recursive proof. Left top: Input pairs (F, R^*), with the picked path (P, c) marked in red. We hide vertices D, E, F, T and associated edges in R^*, as those edges cannot be used in R^*. Left middle: Constructed R^0, which contains only one path with weight 1. The part belonging to P' marked in red. Left bottom: The recursive call (F', R'^*), which is F with R^0 removed and R^* with (P, c) removed, respectively. On the right-hand side, we show a solution to the OR-Quant upper bound. Right top: The MaxFlow on G^* directly mapped as F^* onto G without any completion. Right middle: A completed flow with matching flow value of 8. Right bottom: The solution R^+ obtained from the recursive process. For each path the first edge to appear in E' is marked in orange.

2.4.1.3. Flow reconstruction instance

An instance for flow reconstruction has two inputs: (F, R^*). F is a flow on G, and R^* is a partial decomposition on F^* (of G^*), where F^* is constructed in the same way as G^* by moving endpoints of edges in E' to a new node T'. F^* is actually not a flow because load balance is violated at the vertices that have an incoming edge moved away. Nevertheless, partial decomposition of F^* is still well-defined as R^* contains $S-T'$ paths. The output of the instance is R^+, a partial decomposition of F, satisfying that (1) each path in R^+ is good and (2) if we map each path in R^+ back to G^*, the resulting partial decomposition has the same flow as R^*.

We will apply the induction on $|R^*|$, number of paths in R^*. The base case is then $|R^*|=0$, where we can simply return an empty R^+. Assume now the flow reconstruction can be solved for $|R^*| \leq k-1$, we solve the instance with $|R^*|=k$. We first state the algorithm, and then prove its correctness.

2.4.1.4. Induction algorithm

We first pick the path (P, c) in R^* whose second-to-last vertex (right before T') is the last among all paths in topological ordering of G. Denote the last edge of this path, mapped back to G, as (u, v). P without its last edge is then a path from S to u. We let P' denote P with last edge changed back to (u, v).

We next run a MaxFlow on G from v to T with total flow c. This call always returns a full flow (with total flow c), because there is at least c incoming flow for v from the existence of edge (u, v) alone. This flow is then decomposed, and for each path in the decomposition, we prepend it with P' (which is a path from S to v) so that it becomes an $S-T$ path. We call the resulting partial decomposition R^0, and we next show it is valid, meaning the total flow of R^0 on each edge does not exceed F. For edges in P', the total flow on this edge in R^0 is exactly c. As (P, c) is a path in R^*, each of the edges in P has at least c flow in F^*, which immediately means each edge in P' has at least c flow in F. For all other edges, its flow comes from the $v-T$ MaxFlow, so the flow value is always below the capacity of this edge.

We continue the process by recursively calling the procedure on (F', R'^*), where F' is F minus the flow of R, and R'^* is R^* without (P, c). We return the partial decomposition from the recursive call merged with R^0.

2.4.1.5. Proof of induction correctness

The new instance to be called is an instance with size $k-1$, since one path in R^* is removed. We first prove that the recursive call is valid, that is, R'^* is indeed a partial decomposition of F'^* (F'^* is constructed from F' by moving the endpoint of edges in E' to T').

If the call is invalid, it means some edge in F'^* has less flow than the total flow from R'^*. Since R^* is a partial decomposition of F^*, the condition will only be invalidated for an edge with positive flow in R^0, as these are the only edges where the flow on F'^* is less than the flow on F^*. This edge can either be an edge in P, or an edge whose starting point is later than v in the topological order of G by construction of R^0 (with a $v-T$ MaxFlow). If the edge is in P and is not the last edge, exactly c flow is subtracted on this edge from F to F' and from R^* to R'^*, so this would contradict with the condition that R^* is a partial decomposition of F^*. If this is the edge (u, T') that is mapped from (u, v), again, exactly c flow is subtracted in both F to F' and R^* to R'^*.

If this is an edge not in P, we claim the edge has zero flow in R'^*. This is because the way we pick (P, c), where the path with the latest second-to-last vertex is picked. If the edge has positive flow in R'^*, it implies there is a path in R'^* that includes this edge, and the ending point of the path would be later than v, meaning the path would have been chosen in the place of P during the process.

Given the recursive call is valid, we can prove the correctness of the algorithm by showing that R^+ outputted by the algorithm indeed satisfies the two output conditions. For (1), each path in R^+ is good because it comes from an R^0 at some iteration, and all paths in R^0 have P' as prefix whose last edge (u, v) is in E'. For (2), at each recursion call, R^0 mapped to F^* becomes one path P with flow c, which matches (P, c) in R^*, so the condition is maintained similarly by an induction argument. This finishes the correctness proof for the induction.

2.4.1.6. Completing the proof

With the induction algorithm, we can reconstruct a partial decomposition R^+ of G from R^* (a decomposition of F^*), then similar to the previous proof, construct another partial decomposition R^- from F minus the flow of R^+. By construction of R^+, we have $Q(R^+) = \text{MaxFlow}(G^*)$. If some path in R^- intersects with E', we can remove that path from R^- and add it to R^+, then map the new R^+ to G^* to obtain a flow better than MaxFlow(G^*), a contradiction. This means $Q(R^-)=0$ and the decomposition $R=R^+ + R^-$ satisfies $Q(R)=$ MaxFlow(G^*), completing the whole proof. □

With both bounds proved, we can formally finish the proof of the main theorem:

Theorem 1. *Let G^+ and G^- be constructed as described above. Then* OR-Quant(G, E') = [U − MaxFlow(G^-), MaxFlow(G^+)].

Proof. This is implied by Theorem 3 for the lower bound, and Theorem 4 for the upper bound. □

2.4.2. Algorithms for AND-Quant. In this section, we prove that the complementarity argument mentioned above is correct. Our setup consists of a DAG G with predetermined source S and sink T, a flow $F=\{f_e\}$ on G, and a list of edge sets $\{E_k\}$. We define flows and decompositions in the same way as the last section (Definitions 3 and 4). Now recall the definition of AND-Quant, written slightly different with the new notations:

Definition 7 (Well-Ordering Property). *A list of edge sets $\{E_k\}$ satisfies the well-ordering property if a path visiting $e_i \in E_i$ then $e_j \in E_j$ at a later step implies $i < j$.*

Definition 8 (AND-Quant). *Let $G=(V, E)$ be a directed acyclic graph with an edge flow, and $\{E_k\}$ be a list of edge sets satisfying the well-ordering property. An $S-T$ path is good if it intersects each E_k. For a decomposition of F, the total good flow is the total flow from good $S-T$ paths.* AND-Quant(G,$\{E_k\}$) *asks for the minimum and maximum total good flow for all possible decompositions of F.*

We start by discarding edges e in any of E_i such that no good $S-T$ paths would use e. By definition, removal of these edges will not change the answer to AND-Quant as no good $S-T$ paths would be excluded by removing these edges. This also means each edge e in any E_i satisfies that there is a good $S-T$ path that includes e.

Now given G is a DAG and the edge sets satisfy the well-ordering property, we can define the following:

Definition 9 (Start/End-Sets and Natural Order). *Define $U_i=\{u : (u, v) \in E_i\}$, $V_i=\{v : (u, v) \in E_i\}$ and for convenience, $V_0=\{S\}$, $U_{m+1}=\{T\}$.*

We define the natural order $u \le v$ if there is a directed path starting at u and ending at v. We define $U_i \to x$ as the condition that there is some $u_i \in U_i$ such that $u_i \le x$ (intuitively, x can be reached from U_i), and $x \to U_i$ as the condition that $x \le u_i$ for some $u_i \in U_i$ (intuitively, x can reach U_i). The conditions $V_i \to x$ and $x \to V_i$ can be similarly defined.

Definition 10 (The Block Graph). *Define B_i, the i^{th} block subgraph, the set of edges (u, v) that satisfies $V_{j-1} \to u$ and $v \to U_j$, for $1 \le i \le m+1$. The full block graph G_B is the union of all B_i and E_i.*

The filtering steps and construction of the block graph can be done in linear time, as discussed in Section 2.4.3.

The rest of this section is dedicated to prove the following lemma, which directly implies the main theorem:

Lemma 3 (Main Lemma for AND-Quant). *An $S-T$ path in G is good if and only if it is a subset of G_B.*

We now provide a quick overview before proceeding to the actual proof.

2.4.2.1. Overview

The proof of the above lemma is done in several parts. The if direction of the statement is straightforward, and the only-if direction is harder to prove. We start by deriving some helpful corollaries of the well-ordering property. Next, we show that any vertex x in the block graph and any edge set E_i in the query list, x is either strictly before E_i, meaning there is a path starting at x and ending at an edge in E_i (this includes the case where $x \in U_i$, that is, an edge in E_i originates from x), or strictly after E_i, meaning there is a path starting at an edge in E_i and ending at x. This implies each E_i is a cut of G_B, which implies the only-if part of the statement.

Lemma 4 (Well-Ordering Property Extended). If $u \in U_i,\ v \in V_j,\ v \le u$, *then* $i > j$.

Proof. We can derive this from the well-ordering property of $\{E_k\}$, by constructing an $S-T$ path. First, as $v \in V_j$ there is an edge in E_j that contains v, and by our previous assumption there is a path from S to v that uses this edge. As $v \leq u$, there is a path from v to u. As $u \in U_i$, there is an edge in E_i that contains u, and similarly there is a path from u to T that uses this edge. Combining the three segments together, we have an $S-T$ path that first visits an edge in E_j, then an edge in E_i later, so by the well-ordering property of $\{E_k\}$, $i > j$. □

Lemma 5. *For each $v_i \in V_i$, $v_i \to U_{i+1}$.*

Proof. Recall that we removed all edges in E_i that do not belong to any good paths. $v_i \in V_i$ implies there is an edge $(u_i, v_i) \in E_i$ that is contained in a good path, and later in this good path there is an edge in E_{i+1} that starts at some vertices in U_{i+1}. This implies $v_i \to U_{i+1}$. □

Lemma 6. *For each $u_i \in U_i$, $u_i \to U_j$ where $j \geq i$.*

Proof. $u_i \in U_i$ means there is an edge $(u_i, v_i) \in E_i$, and $u_i \to U_i$. As shown in the last lemma, $v_i \to U_j$ for $j > i$. This means the same also holds for u_i as $u_i < v_i$. □

We can prove the symmetric statements about V_i.

Lemma 7. *For a fixed i, each vertex x in G_B either satisfies $x \to U_i$ or $V_i \to x$, but not both.*

Proof. If x is in one of U_j, $x \to U_i$ if $i \leq j$ by Lemma 6. Similarly, if $x \in U_j$, $V_i \to x$ if $i > j$ by both Lemma 5.

We can similarly prove that the condition holds for all $x \in V_j$, where $x \to U_i$ if $j < i$, and $V_i \to x$ if $j \geq i$.

If x is in none of U_j or V_j, it is in an edge in B_j, which by definition of B_j means both $V_{j-1} \to x$ and $x \to U_j$. Combined with Lemma 6, this means $x \to U_i$ if $j \leq i$, and $V_i \to x$ if $j > i$.

We have proved for any vertex in x and any i, one of $x \to U_i$ or $V_i \to x$ must be satisfied. No vertices can satisfy both, otherwise we have $u \in U_i$, $v \in V_i$ and $v \leq x \leq u$, which would imply $i < i$ by Lemma 4, contradiction. □

Lemma 3 (Main Lemma for AND-QUANT). *An $S-T$ path in G is good if and only if it is a subset of G_B.*

Proof. We first prove that if a path is good, it is within G_B. An edge (u, v) on the path is either in some E_i, or between some edge in E_{i-1} and some edge in E_i. In the latter case, we know $V_{i-1} \to u$ and $v \to U_i$, which directly imply $(u, v) \in B_i$.

To prove the other direction, we show that removing any of E_i results in disconnection of S to T. By the previous lemma, the vertices of G_B can be partitioned into two disjoint sets: $\bar{S} = \{x : x \to U_i, x \in G_B\}$, $\bar{T} = \{x : V_i \to x, x \in G_B\}$. For an edge $(u, v) \in G_B$ that starts in $\bar{S}$ and ends in $\bar{T}$:

- If $(u, v) \in B_j$, we know $V_{j-1} \to u \to U_i$ (first part from definition of B_j and second part from $u \in \bar{S}$, similar for the rest of this proof), so $j-1 < i$, and at the same time $V_i \to v \to U_j$ so $i < j$, but there is not an integer i between $j-1$ and j, contradiction.
- If $(u, v) \in E_j$, we know $U_j \to u \to U_i$ so $j \leq i$ (because $V_{j-1} \to u$ by using the fact there is a good path including u), and $V_i \to v \to V_j$ so $i \leq j$ (because $v \leq v_j \to U_{j+1}$ for some $v_j \in V_j$ by Lemma 5), which implies $i = j$.

So we have $(u, v) \in E_i$. In other words, removing all edges in E_i results in disconnection between $\bar{S}$ and $\bar{T}$, so each $S-T$ path in G_B uses an edge in E_i, for each i. This means the path is good. □

We can now finish the proof of the main theorem:

Theorem 2. *Let G_B be the block graph, and $[l, r] =$* OR-QUANT*$(G, G-G_B)$. Then* AND-QUANT*$(G, \{E_i\}) = [U-r, U-l]$.*

Proof. By Lemma 3, a path is a bad path if and only if the path intersects with $G-G_B$. Thus, we can convert an instance of AND-QUANT to an instance of OR-QUANT by constructing the block graph G_B, then complementing the results of OR-QUANT with $E' = G-G_B$. The correctness of our OR-QUANT algorithms is guaranteed by Theorem 1. □

2.4.3. *Constructing* G_B.

In this section we discuss how to construct G_B in linear time using notations from Section 2.4.2.

We first run two breadth-first searches from S and T to mark the set of vertices that are reachable from S and can reach T. All vertices that are unable to do both will never appear in an $S-T$ path, and will be removed first. Let $s(v)$ denote the largest i such that there is a path from S, via an edge in each of $E_1, E_2, \ldots, E_i$ in order, to v. Generate a topological order of G, and let $s(S)=0$. For every vertex in the topological order other than S, the value of $s(v)$ is determined as follows. $s(v)$ is initialized as the largest of $s(u)$ from all its predecessors. Then for every $(u, v) \in E_j$ and $s(u)=j-1$, set $s(v)=\max(s(v), j)$.

We now show that the values of $s(v)$ are correctly derived. If v is indeed reachable from S via an edge in each of $E_1, E_2, \ldots, E_i$, we have $s(v) \geq i$ as we will visit all nodes on the path in topological order and after an edge in E_i we always have $s(v) \geq i$. If $s(v) \geq i$, we show v is reachable from S via an edge in each of $E_1, E_2, \ldots, E_i$. We let $p(v)$ be the predecessor of v where $s(v)$ is calculated from, either by passing the value or passing an edge in E_j.

By chaining $p(v)$, we obtain a path from S to v where each vertex is responsible for the value of $s(v)$ of its successor in the path. This means along the path the value will only increase by passing through an edge in E_j, and this can only bring $s(v)$ from $j-1$ to j, so the path must contain an edge from each $E_1, E_2, \ldots, E_i$ in order.

Similarly, we can obtain $t(v)$, which is defined as the smallest i such that there is a path from v, via an edge in each of $E_i, E_{i+1}, \ldots, E_m$ to T by setting $t(T)=m+1$ and traverse the graph in reverse topological order. The filtering process then works as follows. For each $(u, v) \in E_i$, the edge is kept in E_i if and only if $s(u)=i-1$ and $t(v)=i+1$. We prove this procedure is correct. A good $S-T$ path containing (u, v) must visit an edge in each of $\{E_1, E_2, \ldots, E_{i-1}\}$ in order, visit $(u, v) \in E_i$, then an edge in each of $\{E_{i+1}, E_{i+2}, \ldots, E_m\}$.

The first part is possible if and only if $s(u) \geq i-1$, and the last part is possible if and only if $t(v) \leq i+1$. It remains to prove that $s(u) \leq i-1$ (the other part is symmetric). This is because otherwise there will be an edge in E_i that precedes u. As (u, v) is also in E_i, this means there will exist a path containing two edges in E_i, which violates the well-ordering property.

To construct G_B, we need to construct the set of blocks B_i. This can be done using the values of $s(v)$ and $t(v)$. For an edge (u, v) not in any of E_i, if $s(u)+1=t(v)$, the edge is allocated to $B_{t(v)}$. We prove this procedure is correct. If an edge satisfies $s(u)+1=t(v)=k$, there is an edge in E_{k-1} that leads to u, meaning $V_{k-1} \rightarrow u$, and similarly $u \rightarrow U_k$, which is the definition of B_k. Similarly, if $(u, v) \in B_k$, we show $s(u)+1=t(v)=k$. Given $V_{k-1} \rightarrow u$ and the edge set is filtered, there exists $v_{k-1} \in V_{k-1}$, $v_{k-1} \leq v$, and $s(v) \geq s(v_{k-1})=k-1$. If $s(v) \geq k$, this means there is an edge in E_k that leads to u and $V_k \rightarrow u$. Given $v \rightarrow U_k$ at the same time, we have $V_k \rightarrow u \leq v \rightarrow U_k$, and there is a path that visits an edge in E_k twice, violating the well-ordering property. We can similarly prove it is necessary and sufficient that $t(v)=k$.

Combining the results, we have the following lemma for preprocessing (note that topological sorting is a linear time algorithm):

Lemma 8. *There is an* $O(n+m)$ *algorithm that filters* $\{E_i\}$ *such that every remaining edge in every* E_i *is included in at least one good* $S-T$ *path, and constructs* $\{B_i\}$ *as described in Definition* 10, *where n and m are the number of nodes and edges in G.*

2.5. *Structured analysis of differential expression*

We have discussed nonidentifiability-aware transcript quantification under two assumptions. In this section, we model the quantification problem under a hybrid assumption. Some fragments are generated from the reference transcriptome, while others are generated by combining known junctions (valid under the setup of graph quantification). This model is more realistic than the model under the two extreme assumptions.

For each transcript T_i, let $[l_i^0, u_i^0]$ denote its range of optimal expression calculated under the complete reference transcriptome assumption (as in Section 2.2), and $[l_i^1, u_i^1]$ denote its range of optimal expression calculated under the incomplete reference transcriptome assumption (as in Section 2.4). We use parameter λ to indicate the assumed portion of fragments generated by combining known junctions. For $0 < \lambda < 1$, we define $[l_i^\lambda, u_i^\lambda] = [\lambda l_i^1 + (1-\lambda) l_i^0, \lambda u_i^1 + (1-\lambda) u_i^0]$, interpolating between ranges for the extreme assumptions. These ranges are useful for analyzing the effects of nonidentifiability under milder assumptions, as we now show for differential expression.

In differential expression analysis, for each transcript we receive two sets of abundance estimates $\{A_i\}$, $\{B_i\}$ under two conditions, and the aim is to determine whether a transcript is expressed more in $\{A_i\}$. With fixed λ, we can instead calculate the ranges $\{[l^{\lambda}_{A,i}, u^{\lambda}_{A,i}]\}$ and $\{[l^{\lambda}_{B,i}, u^{\lambda}_{B,i}]\}$ as described above, where $\{[l^{\lambda}_{A,i}, u^{\lambda}_{A,i}]\}$ are the ranges for transcript T_i under the condition corresponding to abundance estimates $\{A_i\}$. Suppose the transcript is detected to be DE by comparing $\{A_i\}$ and $\{B_i\}$ and it is overexpressed in $\{A_i\}$.

When considering nonidentifiability, if the mean of $l^{\lambda}_{A,i}$ is less than that of $u^{\lambda}_{B,i}$ by a threshold, we define this transcript to be a questionable call for differential expression. If λ portion of expression is explained by unannotated transcripts, we cannot determine definitely if the expression of A_i is higher on average than that of B_i. This filtering of differential expression calls is very conservative (expected to filter out few calls), as most differential expression callers require much higher standards for a differential expression call.

3. RESULTS

3.1. Implementation

In the following analyses, we use GENCODE annotation (version 26) (Frankish et al., 2018) as the reference transcriptome for constructing the splice graphs and for expression quantification. Salmon (Patro et al., 2017) and its corresponding graph quantification probabilistic model (Ma et al., 2020) are used to obtain the edge abundances of the splice graphs under complete and incomplete reference assumptions. We evaluate the expression estimation uncertainty due to nonidentifiability by determining whether the ranking of expression can be altered under different optimal abundances. Specifically, the ranking is computed within the same sample across isoforms in Section 3.2, and for the same isoform across samples in Section 3.3, which is statistically defined by DE analysis.

3.2. Expression of 20%–47% transcripts has inconclusive ranking compared with sibling isoforms

We applied our methods to the Human Body Map data set (*The Illumina Body Map 2.0 data*; 2011) (SRA accession ERX011205), which consists of 16 RNA-seq samples from 16 tissues. We are interested in evaluating the expression estimation uncertainty due to nonidentifiability, and focus on the transcripts for which the uncertainty of expression estimation is so large that the ranking of expression between the transcript and its sibling isoforms cannot be determined. We use the term ''sibling isoforms'' of a transcript to refer to the annotated alternative splicing isoforms that belong to the same gene. For each transcript, we enumerate its sibling isoforms, and compare the range of optimal expression estimates for the pair of transcripts to determine whether the two ranges overlap. An overlap between the two ranges indicates an indecisiveness in the ranking of expression between the two transcripts.

We observe that around 35%–50% of transcripts have uncertain expression estimates due to nonidentifiability. That is, the ranges of optimal abundances of these transcripts are not single points. The majority of them (around 20%–47% of total transcripts across all 16 samples) have a very uncertain expression estimate such that the ranking of expression between the transcript and at least one of its sibling isoform is inconclusive (Fig. 4A). The range is computed under various λ values (compositions of reference transcript expression) ranging from 10% to 100%. These transcripts are possibly false positives in isoform switch detection if they are predicted.

When compared with the expression estimation uncertainty caused by finite sample size (or finite sequencing depth), we observe that the sets of transcripts with inconclusive expression ranking due to the two sources of error do not have large overlap (Fig. 4B). In an arbitrarily chosen Human Body Map sample (a prostate sample, SRA accession ERR030877), we set the λ value to be 70% to make the number of transcripts with uncertainty expression estimates under the two cases similar. Only half of transcripts for which the uncertainty estimates are caused by finite sample size are common to the transcripts under the model nonidentifiability case. This observation suggests that the expression uncertainty due to model nonidentifiability cannot be captured by the degree of uncertainty due to finite sample size.

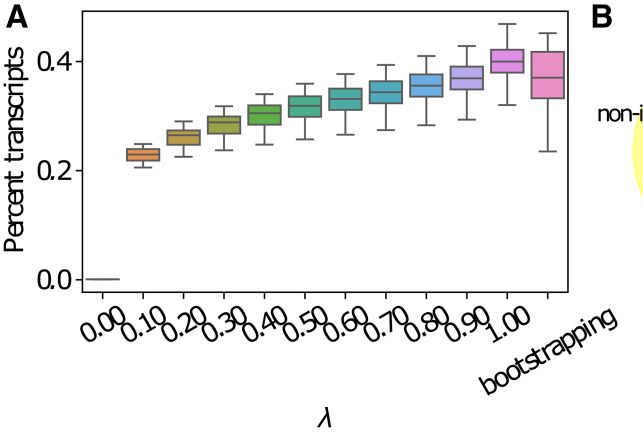


FIG. 4. Transcripts with inconclusive ranking of expression among sibling isoforms due to nonidentifiability. **(A)** Percentages of transcripts that have inconclusive ranking of expression compared with their sibling isoforms due to model nonidentifiability under various λ values (composition of splice graph path expression). The last column of x-axis represents another source of expression uncertainty, finite sample size, which is evaluated by the bootstrapping method implemented in the Salmon quantifier. **(B)** Venn diagram showing the unique and overlapping transcripts with inconclusive ranking of expression due to model nonidentifiability and due to finite sample size. This subplot corresponds to the prostate sample of Human Body Map data set. λ is set to be 70% for the model nonidentifiability case.

The uncertainty caused by finite sample size is evaluated by bootstrapping in Salmon software (Patro et al., 2017). Terminus (Sarkar et al., 2020) identifies groups of transcripts that have smaller quantification variance when quantifying as a group compared with quantifying individually, and is based on bootstrapping or Gibbs sampling of expression estimation. Theoretically, the difference between Terminus and our approach for identifying uncertain expression estimates lies in the source of error and whether incomplete reference is considered. In Figure 5, we show the percentage of commonly identified transcripts with expression estimation uncertainty of the two methods in the Human Body Map data set.

The computed percentages of transcripts with inconclusive ranking of expression are upper bounds. Because the range of optimal expression estimate per transcript is calculated separately, arbitrarily selecting a pair of values from two ranges of optimal expression of two transcripts may not lead to an optimal pair of expression in the expression probabilistic model. For example, selecting the maximum values for both isoforms may lead to nonvalid estimates where the sum of estimated expression (before normalization) exceeds the number of observed RNA-seq reads.

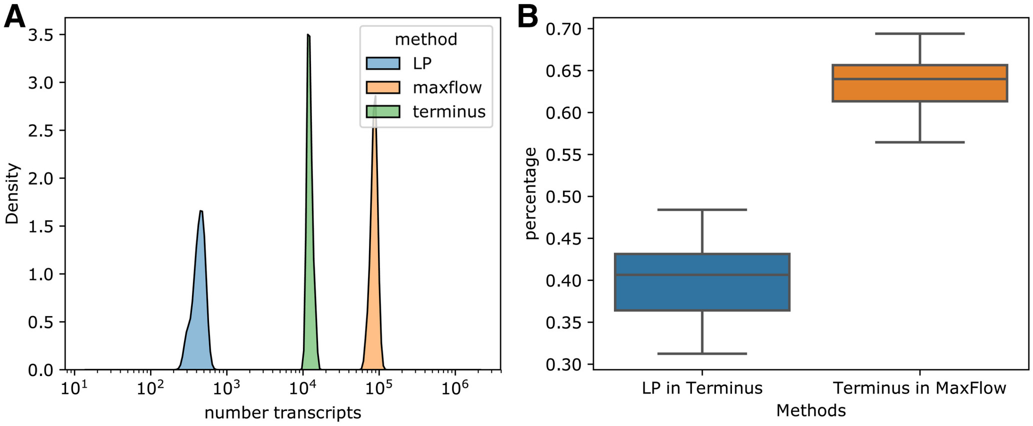


FIG. 5. Counting the number of transcripts with uncertain expression estimates, and comparing with terminus. **(A)** Histogram of number of transcripts identified with uncertain expression estimates under linear programming (LP), MaxFlow in our approach and under Terminus. **(B)** Percentage of transcripts identified under LP that are also identified by Terminus, and percentage of transcripts identified by Terminus that are also identified under MaxFlow.

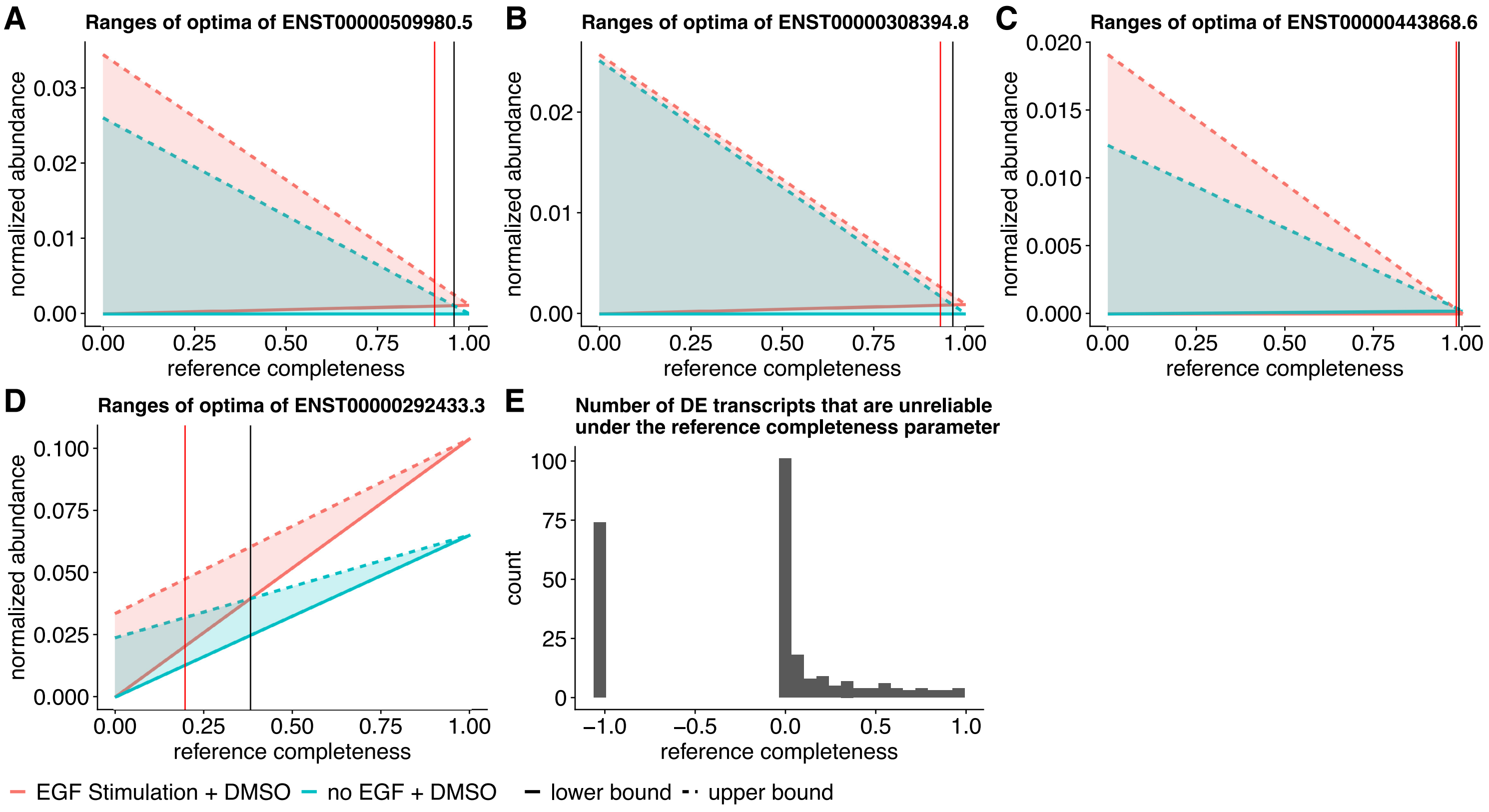


FIG. 6. Overlap between ranges of expression indicates potentially unreliable DE detections in the MCF10 cell line data. **(A–D)** Mean ranges of optima of DE groups. *X* axis is assumed reference completeness $1-\lambda$, the proportion of expression from the reference transcriptome. *Y* axis is the normalized abundances, where Salmon estimates are normalized into transcript per million (TPM) for linear programming under complete reference assumption, and total flow in subgraph quantification is normalized to 10^6 for each sample. Black vertical lines indicate the reference completeness where the mean ranges of optima overlap. Red vertical lines indicate the reference completeness that the ranges have 25% overlap. **(A)** The potential unreliable DE transcript ENST00000509980.5. **(B)** The potential unreliable DE transcript ENST00000308394.8. **(C)** The reliable DE transcript ENST00000443868.6. **(D)** The reliable DE transcript ENST00000527470.5. **(E)** The histogram of the number of unreliable DE transcripts at each λ value. Unreliability is defined as >25% overlap of the ranges of optima. -1.0 in the *x* axis indicates that the overlap is no greater than 25% over all λ values.

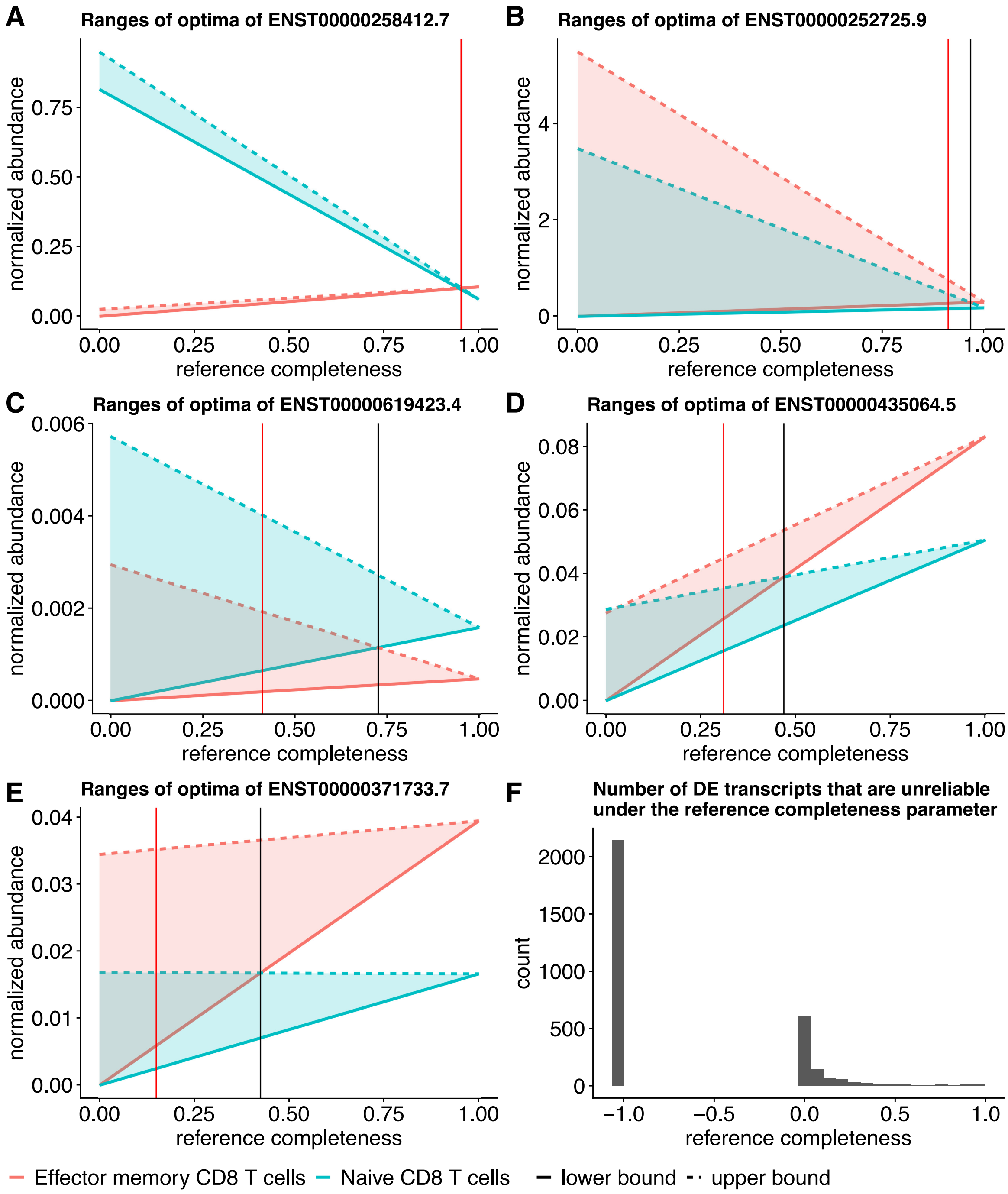


FIG. 7. Overlap between ranges of expression indicates potentially unreliable DE detections in the CD8 T cell data. **(A–E)** Mean ranges of optimal abundances of DE groups. *X* axis is $1-\lambda$, which is the proportion of expression from the reference transcriptome. *Y* axis is the normalized abundances, where Salmon estimates are normalized into TPM for linear programming under complete reference assumption, and total flow in subgraph quantification is normalized to 10^6 for each sample. Black vertical lines indicate the reference completeness where the mean ranges of optima overlap. Red vertical lines indicate the reference completeness that the ranges have 25% overlap. **(F)** The histogram of the number of unreliable DE transcripts at each λ value. Unreliability is defined as >25% overlap of the ranges of optima. -1.0 in the *x* axis indicates that the overlap is no greater than 25% over all λ values.

However, we speculate that these upper bounds are close to the true percentages. Because reversing the ranking requires one to increase the expression of one transcript and decrease the expression of the other, and it is less likely to generate nonviable paired estimates under this operation.

3.3. DE transcripts are generally reliable when assuming the reference transcripts contribute >40% to the expression

Applying our method to the MCF10 cell line samples with and without epidermal growth factor (EGF) treatment (accession SRX1057418) (Kiselev et al., 2015), we analyze the reliability of the detected DE transcripts. We use Salmon (Patro et al., 2017), tximport (Soneson et al., 2015), and DESeq2 (Love et al., 2014) for the differential expression detection pipeline. This pipeline predicts 257 DE transcripts under a false discovery rate (FDR) threshold of 0.01. We use the overlap between the mean ranges of optimal expression estimated in the samples with and without EGF treatment as an indicator for unreliable DE detection (see Section 2.5 for details). The overlap is defined as the size of the intersection over the size of the smaller range of the two ranges of optima. We compute the overlap under various λ values. The threshold to declare an unreliable DE detection is 25% of overlap.

We identify examples of reliable and unreliable DE predictions. There are five DE transcripts for which their differential expression statuses may change even when the reference expression proportion $(1-\lambda)$ is as high as 90%. Figure6A–C shows the lower bounds and upper bounds of the transcript expression of three examples among the five transcripts. Their expression estimates suffer from great uncertainty such that the ranges of optima between the two DE groups overlap.

The five genes corresponding to the five transcripts are involved in the following cellular processes or pathways: mRNA degradation, cell apoptosis, glucose transportation, DNA repair, inhibition and transportation of certain kinetics (Pruitt et al., 2011). These genes contain between 6 and 22 isoforms. Further analyses based on the DE detection of these five transcripts require caution since they may be falsely detected to be DE due to nonidentifiability.

Other than these transcripts, the detected DE transcripts are generally reliable after considering the potential expression estimation uncertainty due to nonidentifiability. The ranges of optimal expression estimates between the two sample groups do not have large overlaps when the reference transcriptome is relatively complete and contribute >40% to the expression (Fig. 6E). This observation is supported by another data set, where replicates naive CD8 T cells, and four replicates of effector memory CD8 T cells are compared for differential expression detection, as detailed in Figure 7. There are 3152 DE transcripts under FDR threshold 0.01, 19 out of which are unreliable even when reference transcripts compose >90% expression. We observe the similar pattern that most DE predictions are reliable when the reference transcript expression is >40%.

A previous study (Everaert et al., 2017) showed that expression quantification software tends to make slightly more mistakes in deciding the relative expression of isoforms within one sample, compared with deciding the fold change of one isoform across multiple samples. Our results here and in the previous section show agreement with that observation, but with amplified errors in deciding relative expression of isoforms within one sample. Our model for bounding the range of uncertainty due to model nonidentifiability provides an explanation from the theoretical perspective: the short sequencing reads may not be sufficient to uniquely reveal the relative abundances of transcripts from complicated splice graphs.

4. CONCLUSION AND DISCUSSION

We develop algorithms to compute the range of optimal expression estimates due to nonidentifiability for each transcript. We consider both complete and incomplete reference transcript assumptions (quantified with reference-transcript-based quantification and graph quantification, respectively) and further provide the range of uncertain estimates under mixed assumptions: a certain proportion of expression is from reference transcripts and the rest (indicated by λ) is from expression of splice graph paths. The code for computing the range of expression is available at https://github.com/Kingsford-Group/subgraphquant. The code for the involved analyses is available at https://github.com/Kingsford-Group/subgraphquantanalysis.

Applying our methods on Human Body Map samples and two RNA-seq data sets for DE transcript detection, we observe the following expression uncertainty patterns: the ranking of expression between a transcript and its sibling isoforms in a given sample cannot be determined for many (20%–47%) transcripts if the expression estimation uncertainty is considered, but when comparing the expression estimates of a transcript across multiple RNA-seq samples, the detected DE transcripts are mostly reliable.

The λ parameter is unknown in our model, and we address this by investigating the ranges of optima under varying reference completeness values. However, determining the best λ value that fits the data set as an indicator for reference trustfulness is an interesting question in itself, and we believe transcript assembly or related methods might be useful for choosing the correct λ value for each data set.

The nonidentifiability problem in expression quantification is partly due to the contrast between the complex splicing structure of the transcriptome and short length of observed fragments in RNA-seq. Recent developments of full-length transcript sequencing might be able to close this complexity gap by providing longer range phasing information. However, full-length transcript sequencing techniques suffer from problems such as low coverage and high error rate. It is still open whether full-length transcript sequencing is appropriate for quantification and how the current expression quantification methods, including this work, should be adapted to work with full-length transcript sequencing data.

ACKNOWLEDGMENTS

We thank Dr. Guillaume Marçais, Dr. Natalie Sauerwald, and Dr. Jose Lugo-Martinez for insightful comments on the article.

DISCLAIMER

The Department specifically disclaims responsibility for any analyses, interpretations, or conclusions.

AUTHOR DISCLOSURE STATEMENT

C.K. is the cofounder of Ocean Genomics, Inc.

FUNDING INFORMATION

This work has been supported, in part, by the Gordon and Betty Moore Foundations Data-Driven Discovery Initiative through Grant GBMF4554 to C.K., by the U.S. National Institutes of Health (R01GM122935), and the U.S. National Science Foundation (DBI-1937540). This work was partially funded by The Shurl and Kay Curci Foundation. This project is funded, in part, under a grant (No. 4100070287) with the Pennsylvania Department of Health.

REFERENCES

Al Seesi, S., Tiagueu, Y.T., Zelikovsky, A., et al. 2014. Bootstrap-based differential gene expression analysis for RNA-Seq data with and without replicates. *BMC Genomics* 15, S2.

Bernard, E., Jacob, L., Mairal, J., et al. 2014. Efficient RNA isoform identification and quantification from RNA-Seq data with network flows. *Bioinformatics* 30, 2447–2455.

Bray, N.L., Pimentel, H., Melsted, P., et al. 2016. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* 34, 525–527.

Everaert, C., Luypaert, M., Maag, J.L., et al. 2017. Benchmarking of RNA-sequencing analysis workflows using whole-transcriptome RT-qPCR expression data. *Sci. Rep.* 7, 1–11.

Frankish, A., Diekhans, M., Ferreira, A.-M., et al. 2018. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res.* 47(D1), D766–D773.

Gatter, T., and Stadler, P.F. 2019. Ryūtō: Network-flow based transcriptome reconstruction. *BMC Bioinformatics* 20, 190.

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

Guo, W., Calixto, C.P., Brown, J.W., et al. 2017. TSIS: An R package to infer alternative splicing isoform switches for time-series data. *Bioinformatics* 33, 3308–3310.

Hensman, J., Papastamoulis, P., Glaus, P., et al. 2015. Fast and accurate approximate inference of transcript expression from RNA-seq data. *Bioinformatics* 31, 3881–3889.

Hiller, D., Jiang, H., Xu, W., et al. 2009. Identifiability of isoform deconvolution from junction arrays and RNA-Seq. *Bioinformatics* 25, 3056–3059.

Hoadley, K.A., Yau, C., Wolf, D.M., et al. 2014. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell* 158, 929–944.

Kiselev, V.Y., Juvin, V., Malek, M., et al. 2015. Perturbations of PIP3 signalling trigger a global remodelling of mRNA landscape and reveal a transcriptional feedback loop. *Nucleic Acids Res.* 43, 9663–9679.

Lacroix, V., Sammeth, M., Guigo, R., et al. 2008. Exact transcriptome reconstruction from short sequence reads, 50–63. *In* Crandall, K.A., and Lagergren, J., eds. *International Workshop on Algorithms in Bioinformatics*. Springer, Karlsruhe, Germany.

LeGault, L.H., and Dewey, C.N. 2013. Inference of alternative splicing from RNA-Seq data with probabilistic splice graphs. *Bioinformatics* 29, 2300–2310.

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.

Liu, R., and Dickerson, J. 2017. Strawberry: Fast and accurate genome-guided transcript reconstruction and quantification from RNA-Seq. *PLoS Comput. Biol.* 13, e1005851.

Love, M.I., Huber, W., and Anders, S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 15, 550.

Ma, C., Zheng, H., and Kingsford, C. 2020. Exact transcript quantification over splice graphs. *Algorithms Mol Biol.* 16, 5. DOI: https://doi.org/10.1186/s13015-021-00184-7.

McCarthy, D.J., Chen, Y., and Smyth, G.K. 2012. Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation. *Nucleic Acids Res.* 40, 4288–4297.

Morán, I., Akerman, İ., van de Bunt, M., et al. 2012. Human β cell transcriptome analysis uncovers lncRNAs that are tissue-specific, dynamically regulated, and abnormally expressed in type 2 diabetes. *Cell Metab.* 16, 435–448.

Nowicka, M., and Robinson, M.D. 2016. DRIMSeq: A Dirichlet-multinomial framework for multivariate count outcomes in genomics. *F1000Research* 5, 1356.

Patro, R., Duggal, G., Love, M.I., et al. 2017. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* 14, 417–419.

Pertea, M., Pertea, G.M., Antonescu, C.M., et al. 2015. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.* 33, 290–295.

Pimentel, H., Bray, N.L., Puente, S., et al. 2017. Differential analysis of RNA-seq incorporating quantification uncertainty. *Nat. Methods* 14, 687.

Pruitt, K.D., Tatusova, T., Brown, G.R., et al. 2011. NCBI Reference Sequences (RefSeq): Current status, new features and genome annotation policy. *Nucleic Acids Res.* 40(D1): D130–D135.

Ritchie, M.E., Phipson, B., Wu, D., et al. 2015. limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res.* 43, e47–e47.

Roberts, A., and Pachter, L. 2013. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nat. Methods* 10, 71.

Sarkar, H., Srivastava, A., Bravo, H.C., et al. 2020. Terminus enables the discovery of data-driven, robust transcript groups from RNA-seq data. *Bioinformatics* 36(Supplement_1), i102–i110.

Soneson, C., Love, M.I., and Robinson, M.D. 2015. Differential analyses for RNA-seq: Transcript-level estimates improve gene-level inferences. *F1000Research* 4.

The Illumina Body Map 2.0 data. 2011. Available at: https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-513. Last access was June 6, 2018.

Tomescu, A.I., Kuosmanen, A., Rizzi, R., et al. 2013. A novel min-cost flow method for estimating transcript expression with RNA-Seq. *BMC Bioinformatics* 14, S15.

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

Turro, E., Su, S.-Y., Gonçalves, Â., et al. 2011. Haplotype and isoform specific expression estimation using multimapping RNA-seq reads. *Genome Biol.* 12, R13.

Vitting-Seerup, K., and Sandelin, A. 2017. The landscape of isoform switches in human cancers. *Mol. Cancer Res.* 15, 1206–1220.

Address correspondence to:
Prof. Carl Kingsford
Computational Biology Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
USA

E-mail: carlk@cs.cmu.edu