Inference of Isoforms from Short Sequence Reads (Extended Abstract)

Jianxing Feng[1], Wei Li[2], and Tao Jiang[3]

[1] Comp. Sci. Dept., Tsinghua Univ., Beijing, China
fengjx06@mails.tsinghua.edu.cn
[2] Comp. Sci. Dept., Univ. of California, Riverside, CA
liw@cs.ucr.edu
[3] Comp. Sci. Dept., Univ. of California, Riverside, CA, and Tsinghua Univ., Beijing
jiang@cs.ucr.edu

Abstract. Due to alternative splicing events in eukaryotic species, the identification of mRNA isoforms (or splicing variants) is a difficult problem. Traditional experimental methods for this purpose are time consuming and cost ineffective. The emerging RNA-Seq technology provides a possible effective method to address this problem. Although the advantages of RNA-Seq over traditional methods in transcriptome analysis have been confirmed by many studies, the inference of isoforms from millions of short sequence reads (*e.g.*, Illumina/Solexa reads) has remained computationally challenging. In this work, we propose a method to calculate the expression levels of isoforms and infer isoforms from short RNA-Seq reads using exon-intron boundary, transcription start site (TSS) and poly-A site (PAS) information. We first formulate the relationship among exons, isoforms, and single-end reads as a convex quadratic program, and then use an efficient algorithm (called IsoInfer) to search for isoforms. IsoInfer can calculate the expression levels of isoforms accurately if all the isoforms are known and infer novel isoforms from scratch. Our experimental tests on known mouse isoforms with both simulated expression levels and reads demonstrate that IsoInfer is able to calculate the expression levels of isoforms with an accuracy comparable to the state-of-the-art statistical method and a 60 times faster speed. Moreover, our tests on both simulated and real reads show that it achieves a good precision and sensitivity in inferring isoforms when given accurate exon-intron boundary, TSS and PAS information, especially for isoforms whose expression levels are significantly high.

1 Introduction

Transcriptome study (or transcriptomics) aims to discover all the transcripts and their quantities in a cell or an organism under different external environmental conditions. A large amount of work has been devoted to transcriptomics, which includes the international projects EST [1, 2], FANTOM [3], and ENCODE [4, 5]. Many technologies have been introduced in recent years including array-based experimental methods such as tiling arrays [6], exon arrays [7], and exon-junction

arrays [8, 9], and tag-based approaches such as MPSS [10, 11], SAGE [12, 13], CAGE [14, 15], PMAGE [16], and GIS [17]. However, due to various constraints intrinsic to these technologies, the speed of advance in transcriptomics is far from being satisfactory, especially on eukaryotic species because of widespread alternative splicing events.

Applying next generation sequencing technologies to transcriptomes, the recently developed RNA-Seq technology is quickly becoming an important tool in functional genomics and transcriptomics. It can be used to identify all genes and exons and their boundaries [18, 19] and to study gene functions and perform transcriptome analysis [20]. For example, based on an unannotated genomic sequence and millions of short reads from RNA-Seq, [21] developed a general method for the discovery of a complete transcriptome, including the identification of coding regions, ends of transcripts, splice junctions, splice site variations, *etc.* Their application of the method to *S.cerevisiae* (yeast) showed a high degree of agreement with the existing knowledge of the yeast transcriptome. Besides yeast [22, 18], RNA-Seq has been applied to the transcriptome analysis of mouse [23, 24] and human [25, 26]. These results demonstrate that RNA-Seq is a powerful quantitative method to sample a transcriptome deeply at an unprecedented resolution. Moreover, DNA sequencing technologies are under fast development. Some of them now could provide long reads, paired-end reads, DNA-strand-sequencing of mRNA transcripts, etc. See [27] for a comprehensive analysis of the advantages of RNA-Seq over traditional methods in genome-wide transcriptome analysis, and the challenges faced by this technology.

Very recently, several methods have been proposed to characterize the expression level of each transcript [28, 29] using RNA-Seq data. In [28], the authors showed that short (single-end or paired-end) read sequences cannot theoretically guarantee a unique solution to the *transcriptome reconstruction* problem (*i.e.*, the reconstruction of all expressed isoforms and their expression levels) in general even if the reads are sampled perfectly according to the length of each transcript (without random distortions and noise). However, under the same assumption, the authors also showed that paired-end reads could help reconstruct the transcripts uniquely and determine their expression levels for most of the currently known isoforms of human, and single-end reads could allow us to determine the expression levels correctly if all the isoforms are known. However, these results are mostly of theoretical interest because sequence read data are random in nature and may contain noise in practice. [29] presented a more practical way to estimate the expression levels of known isoforms. The method uses maximum likelihood estimation followed by importance sampling from the posterior distribution.

The availability of all the isoforms is the basis of the accurate estimation of isoform expression levels [29], which could be used to infer all splicing variants quantitatively and qualitatively. The variations in isoform expression levels and splicing are important for many studies, *e.g.*, the study of diseases [30, 31] and drug development [32]. A lot of effort has been devoted to the identification of transcripts/isoforms from the more traditional EST, cDNA data. Instead of a

comprehensive review, we will just name a few results below. To enumerate all possible isoforms, a core ingredient is the *splicing graph* [33, 34]. A predetermined parameter "dimension" decides how many transcripts are compared simultaneously. The parameter is usually fixed to two, but recently, [34] extended the method to arbitrary dimensions. There are several methods that assemble transcripts from EST data using the splicing graph and its variations [35, 36]. Newly proposed experimental methods in [37, 38] could be used to identify new isoforms. However, it is still unclear whether these methods can be applied in a large scale.

RNA-Seq has shown great success in transcriptome analysis, but it has not been used to infer isoforms. Although it is straightforward to infer the existence of novel isoforms from RNA-Seq data that exhibit novel transcribed regions [24, 6], it is not so obvious how to use RNA-Seq data to infer the existence of novel isoforms in known transcribed regions, because the observed reads could be sampled from either known or unknown isoforms. The problem has remained challenging for two reasons. The first is that RNA-Seq reads are usually very short. The second is due to the randomness and biases of the reads sampled from all the transcripts. In fact, to our best knowledge, there has been no published work to computationally infer isoforms from (realistic) short RNA-Seq reads.

Due to the high combinatorial complexity of isoforms of genes with a (moderately) large number of exons, the inference of isoforms from short reads (and other available biological information) should be realistically divided into two sub-problems. The first is to discover all the exon-intron boundaries as well as the transcription start site (TSS) and poly-A site (PAS) of each isoform. As mentioned above, there are several effective methods for detecting exon-intron boundaries from RNA-Seq read data [18, 19]. The identification of TSS's and PAS's is an indispensable part of many large genomics projects [3, 4]. The technology of GIS-PET (Gene Identification Signature Paired-End Tags) can also be used to identify TSS-PAS pairs [17, 39]. The second sub-problem is to find combinations of exons that can properly explain the RNA-Seq data, given the exon-intron boundary and TSS-PAS pair information.

In this paper, we are concerned with the second sub-problem in isoform inference. Assuming that the exon-intron boundary and TSS-PAS pair information is given, we propose a method (called IsoInfer) to infer isoforms from short RNA-Seq reads (*e.g.*, Illumina/Solexa data). Although our method works for single-end data and data with both single-end and paired-end reads, we will use single-end reads as the primary source of data and paired-end reads as a secondary data which can be used to filter out false positives. We formulate the relationship among exons, isoforms, and single-end reads as a convex quadratic program, and design an efficient algorithm to search for isoforms. Our method can calculate the expression levels of isoforms accurately if all the isoforms are known. To demonstrate this, we have compared IsoInfer with the simple counting method in [40, 41] and the method in [29] on simulated expression levels and reads, and found that our method is much more accurate than the simple counting method and has a comparable accuracy as the method in [29] but is 60 times

faster. Most importantly, IsoInfer can infer isoforms from scratch when they are sufficiently expressed, by trying to find a minimum set of isoforms to explain the read data. Our experimental tests on both simulated and real reads show that it is possible to infer the precise combination of exons in a sufficiently expressed isoform from RNA-Seq short read data with a reasonably good accuracy, when accurate exon-intron boundary and TSS-PAS pair information is provided. To our best knowledge, this is the first computational method to infer isoforms from short RNA-Seq reads. Due to the page limit, some proofs and tables are omitted in this extended abstract but can be found in the full paper [42]

2 Methods

2.1 Assumptions and terminology

Traditionally, only five types of alternating splicing (AS) events have been proposed, including exon skipping, mutually exclusive exons, intron retention, alternative donor and acceptor sites [43]. However, these events are not adequate to describe complex AS events as more experimental knowledge has become available [44]. In this work, we describe isoforms or AS events in a much general way, which is referred to as a "bit matrix" in [44].

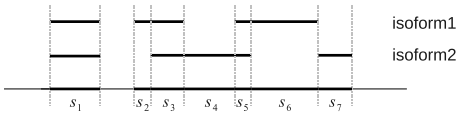

Fig. 1. Expressed segments. Every exon-intron boundary introduces a boundary of some segment. Every expressed segment is a part of an exon.

The exon-intron boundaries of a gene divide the gene into disjoint *segments*, as shown in Figure 1. A segment is *expressed* if it has mapped reads. Thus, every expressed isoform consists of a subset of expressed segments. Two segments are adjacent if they are adjacent in the reference genome (*i.e.*, they share a common boundary). For example, in Figure 1, s_2 and s_3 are adjacent but s_1 and s_2 are not. Any two segments may form a *segment junction* which is not necessarily an exon junction in the traditional sense. For example, s_2 and s_3 form a segment junction, which is not an exon junction. In the following, "junction" refers to "segment junction" unless otherwise stated.

As stated in the introduction, we first assume that exon-intron boundaries are known. Our second assumption is that the short reads are uniformly randomly sampled from all the expressed isoforms (*i.e.*, mRNA transcripts). We have to further assume that the short reads have been mapped to the referenced genome. The mapping of RNA-Seq reads can be done by many recent tools, *e.g.*, Bowtie [45], Maq [46], SOAP [47], RNA-MATE [48] and mrFAST [49]. The mapping of multi-reads (*i.e.*, reads that match several locations of the reference genome) is addressed in [24, 50]. We will use Bowtie in our work due to its efficiency and

accuracy. The last assumption concerns paired-end reads, which will be stated in section 2.3.

2.2 Quadratic programming formulation

$\mathcal{G}$ denotes the set of all the genes. Each g gene defines a set of expressed segments $S_g = \{s_1, s_2, \ldots, s_{|S_g|}\}$ (given exon-intron boundaries), where the expressed segments are sorted according to their positions in the reference genome. The junctions on this gene are all the pairs of expressed segments $(s_i, s_j), 1 \leq i < j \leq |S_g|$. The length of segment s_i is l_i. Denote the set of all known isoforms of this gene as F_g. Each isoform $f \in F_g$ consists of a subset of expressed segments. The expression level (*i.e.*, the number of reads per base) of isoform f is denoted by x_f. The sum of the length of all transcripts, weighted by their expression levels, over all genes, is $L_0 = C \cdot \sum_{g \in \mathcal{G}} \sum_{e \in f, f \in F_g} l_e x_f$, for some constant C that defines the linear relationship between the expression level and the number of transcripts corresponding to an isoform. C can be inferred from data as shown in [24].

From now on, we will consider a fixed gene g and omit the subscript g when there is no ambiguity. Let M be the total number of single-end reads mapped to the reference genome and d_i the number of reads falling into expressed segment s_i. Under the uniform sampling assumption, d_i is the observed value of a random variable (denoted as r_i) that follows the binomial distribution $B(M, p_i)$, where $p_i = Cy_i l_i / L_0$ and $y_i = \sum_{s_i \in f} x_f$. Because M is usually very large, p_i is very small and Mp_i is sufficiently large in most cases, the binomial distribution can be approximated by a normal distribution $N(\mu_i, \sigma_i^2)$, with $\mu_i = Mp_i, \sigma_i^2 = Mp_i(1 - p_i) \approx Mp_i = \mu_i$, similar to the approximation in [29]. Therefore, the random variables $\frac{r_i - \mu_i}{\sigma_i}$, for every expressed segment s_i, follow the same distribution approximately. Define $\epsilon_i = |r_i - \mu_i|$. Then, the variable $\frac{\epsilon}{\sigma_i}$ also follows the same distribution approximately for every s_i.

Let L_1 denote the length of a single-end read. In order to map reads to junctions, we will also think of each junction (s_i, s_j) as a segment of length $2L_1 - 2$, consisting of the last $L_1 - 1$ bases of s_i and the first $L_1 - 1$ bases of s_j. Denote the set of the junctions as $J = \{s_{|S|+1}, s_{|S|+2}, \ldots, s_{|S|+|J|}\}$. The relationship among the expressed segments of gene g, its expressed isoforms, and the single-end reads mapped to each expressed segment and junction can be captured by the following quadratic program (QP):

$$\begin{aligned} min \quad & z = \sum_{s_i \in S \cup J} (\tfrac{\epsilon_i}{\sigma_i})^2 \\ s.t. \; & \sum_{s_i \in f} x_f l_i + \epsilon_i = d_i, \; s_i \in S \cup J \\ & x_f \geq 0, \qquad f \in F \end{aligned}$$

where σ_i is the standard deviation in the normal distribution $N(\mu_i, \sigma_i^2)$ and will be empirically estimated from d_i.

Note that if each r_i follows the normal distribution strictly, then the random variables $\frac{\epsilon_i}{\sigma_i}$ is i.i.d. and thus the solution of the above QP would correspond to the maximum likelihood estimation of the x_f's if each σ_i is fixed [51], and the objective function z is a random variable obeying the χ^2 distribution with freedom

$|S| + |J|$. This QP can be easily shown to be a convex QP by a simple transformation and solved in polynomial time by a public program QuadProg++ which implements the dual method of Goldfarb and Idnani [52] for convex quadratic programming. Since σ_i is unknown, we substitute $\sqrt{d_i}$ for σ_i as an approximation. Let QPsolver denote the above algorithm for solving the convex QP program. Given S, F, and d_i's, QPsolver returns the values of x_i's and z.

When the isoforms in F are given, minimizing the objective function means to find a combination of the expression level (x_f) of each isoform in F such that the observed values (d_i's) can be explained the best. In this case, the value of the objective function serves as an indicator of whether the isoforms in F can explain the observed data. More specifically, *p-value*(z) denotes the probability of $P(Z \geq z)$, where Z is a random variable following the χ^2 distribution with freedom $|S| + |J|$. We empirically choose a cutoff of 0.05. If p-value(z) is less than 0.05 we conclude that F cannot explain d.

2.3 Paired-end reads

Figure 2(left) illustrates some concepts concerning paired-end reads. A paired-end read consists of a pair of short (single-end) reads separated by a *gap*. The figure also defines the *read length*, *span*, *start position*, *center position* and *end position* of a paired-end read. If the span of a paired-end read is a random variable following some probability distribution $h(x)$, then three possible strategies for generating paired-end reads will be considered in this paper.

- Strategy (a): The start position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution $h(x)$. If the end position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the end position of this read is at the end of the isoform.
- Strategy (b): The center position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution $h(x)$. This strategy has been adopted in [53]. Again, if the start (or end) position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the start (or end, respectively) position of this read is at the start (or end, respectively) of the isoform.
- Strategy (c): The end position of a paired-end read is uniformly and randomly sampled from all the expressed isoforms. Then the span of this paired-end is generated following the distribution $h(x)$. If the start position of this paired-end read falls out of the isoform, the paired-end read is truncated such that the start position of this read is at the start of the isoform.

Let w_1, w_2, w_3 be the lengths of three consecutive intervals on an isoform as shown in Figure 2(right). When any of the strategies (a-c) is applied to generate a certain number of paired-end reads, the following Theorem 1 gives a non-trivial upper bound on the probability of not observing any reads with start positions in the first interval and end positions in the third interval.

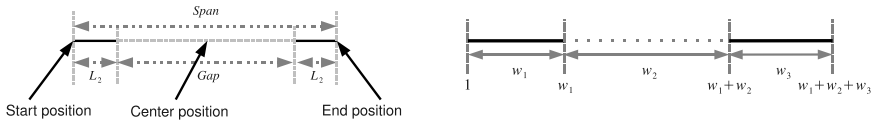

Fig. 2. Left: A paired-end read consisting of two short reads of length L_2 that are separated by a gap. Right: Three consecutive intervals on an isoform.

Theorem 1 *Suppose that the expression level of this isoform is α RPKM (i.e., reads per kilobase of exon model per million mapped reads [24]), and the span of each paired-end read follows some distribution $h(x)$. If M paired-end reads are generated by any of the strategies (a-c), the probability that there are no paired-end reads that have start positions in the first interval and end positions in the third interval is upper bounded by*

$$P_{M,h,\alpha}(w_1, w_2, w_3) = (1 - P_0)^M \approx e^{-MP_0}$$

where $P_0 = 10^{-9}\alpha \sum_{0 \le i < w_1} \int_{l(i)}^{u(i)} h(x)dx$, $l(i) = w_1 - i + w_2$, and $u(i) = w_1 - i + w_2 + w_3$.

2.4 Valid isoforms

For a gene with expressed segments $S = \{s_1, s_2, \ldots, s_{|S|}\}$, an isoform f of this gene can be expressed as a binary vector with length $|S|$. The ith element $f[i]$ of f is 1 if and only if expressed segment s_i is contained in f. Denote the set of all possible binary vectors with n elements as $B(n)$. Similarly, a single-end or paired-end short read that is mapped to a subset $S' \subseteq S$ of expressed segments can be represented as a binary vector $r \in B(|E|)$ such that $r[i] = 1$ if and only if $s_i \in E'$. A subset E' of expressed segments is *supported* by single-end or paired-end reads if there is at least one single-end or paired-end read r such that $r[i] = 1, i \in E'$.

Although single-end reads, paired-end reads, and TSS-PAS information data do not provide exact combinations of expressed segments of isoforms, they can be used to eliminate many isoforms from consideration. Each of these types of data provides some information that can be used to define a condition which will be satisfied by all isoforms inferred by our algorithm (to be described in the next subsection).

- Junction information. A junction (s_i, s_j) is on an isoform f if $f[i] = f[j] = 1$ and $f[k] = 0, i < k < j$. If s_i and s_j are adjacent, then junction (s_i, s_j) is an *adjacent junction*. An isoform satisfies *condition I* if all the non-adjacent junctions on this isoform are supported by single-end short reads. In practice, most sufficiently expressed isoforms satisfy this condition. For example, when 40 millions single-end reads with length 30bps are mapped, the probability of an isoform with expression level 6 RPKM satisfying condition I is 99.3% and 92.8%, if this isoform contains 10 and 100 exons, respectively. See Theorem 2 below for the details.
- Start-end segment pair information. For an isoform f, expressed segment s_i is the *start* expressed segment of f if $f[i] = 1$ and $f[j] = 0, 1 \le j < i$. Expressed segment s_i is the *end* expressed segment of f if $f[i] = 1$ and $f[j] = 0, i < j \le |S|$.

The TSS-PAS pair information describes the start and end expressed segments of each isoform and will be referred to as the *start-end segment pair* data. An isoform satisfies *condition II* if the start-end segment pair of this isoform appears in the given set of start-end segment pairs. If the TSS-PAS pair information is missing, then any expressed segment can theoretically be the start or end expressed segment. However, in this case, many short (and thus unrealistic) isoforms could be introduced, which will make isoform inference difficult. Therefore, when the TSS-PAS pair information is missing, we allow an expressed segment s_i to be the start (or end) expressed segment of any isoform if there is no expressed segment s_j with $j < i$ (or $i < j$) such that junction (s_j, s_i) (or (s_i, s_j), respectively) is adjacent or supported by some read.

– Paired-end read data. A pair of expressed segments $(s_i, s_j), i < j$ on an isoform f is an *informative pair* if $f[i] = f[j] = 1$ and $P_{M,h,\alpha}(l_i + L_2 - 1, g_{i,j}, l_j + L_2 - 1) < 0.05$, assuming that the span of a paired-end read follows some probability distribution $h(x)$, the expression level of this isoform is α RPKM and M paired-end reads have been mapped. Here, L_2 is the read length of a paired-end read, $g_{i,j} = \sum_{i<k<j} l_k f[k]$, and $P_{M,h,\alpha}$ is defined in Theorem 1. According to the theorem, if (s_i, s_j) is informative, then the probability that there are no paired-end reads with start positions in segment s_i and end positions in segment s_j is less than 0.05. A triple of expressed segments $(s_i, s_{i+1}, s_j), i + 1 < j$ is an *informative triple* if $f[i] = f[i + 1] = f[j] = 1$ and $P_{M,h,\alpha}(L_2 - 1, g_{i,j}, l_j + L_2 - 1) < 0.05$. Similarly, $(s_i, s_{i+1}, s_j), j < i$ is an informative triple if $P_{M,h,\alpha}(L_2-1, g_{j,i+1}, l_j+L_2-1) < 0.05$. An isoform satisfies *condition III* if every informative pair or triple on this isoform is supported by paired-end reads. A larger α makes this condition more stringent. Because in many cases, two isoforms can only be distinguished by a pair or triple of segments, it is necessary to require that every informative pairs or triple (instead of some of them) are supported by paired-end reads.

Note that while the junction information is always available given the single-end read data and exon-intron boundary information, the start-end segment pair information and paired-end read data are not necessarily always available. We define an isoform as *valid* if it satisfies conditions I, II and/or III whenever the corresponding types of data are provided. The following theorem gives a lower bound on the probability that type I condition is satisfied by an isoform.

Theorem 2 *Under the uniform sampling assumption, the probability that an isoform f consisting of t exons with expression level x RPKM satisfies type I condition is at least $(1-e^{-xL_1M/10^9})^{t-1}$, where e is the base of natural logarithm, M the number of single-end reads mapped, and L_1 the length of single-end reads.*

2.5 Isoform inference algorithm

We now describe our algorithm, IsoInfer, for inferring isoforms. The algorithm uses the following types of data: the reference genome, single-end short reads, exon-intron boundaries, TSS-PAS pairs, gene boundary information from the reference genome annotation, and paired-end short reads. The first three pieces of information (*i.e.*, the reference genome, exon-intron boundaries and single-end short reads) are required in the algorithm. If TSS-PAS pairs are not provided,

gene boundaries would be required. The flow of data processing in IsoInfer is illustrated in Figure 3. The third step of the algorithm requires an external tool (*e.g.*, Bowtie [45]) to map the short reads to the reference genome and junction sequences. In the fifth step, any two segments that are adjacent or supported by a junction read will be clustered together. Note that, such a cluster may contain expressed segments from more than one gene or contain only a subset of expressed segments from a single gene, but these cases do not happen very often. Furthermore, in each cluster, if there is a sequence of consecutive expressed segments such that every internal segment has no non-adjacent junction with any other expressed segment other than its left or right neighbor in the sequence, then we will combine the expressed segments into a single segment. This compression will be important because it reduces the problem size drastically for some isoforms containing a very large number of expressed segments. The details of the clustering and compression step are straightforward and omitted.

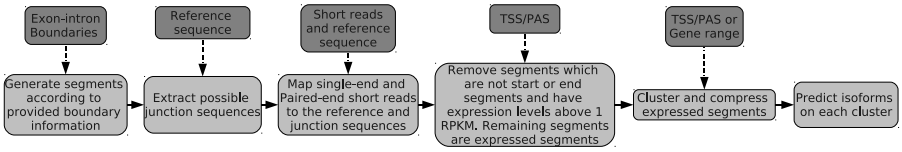

Fig. 3. The flow of data processing in algorithm IsoInfer.

In the following, we give more details of the last step in IsoInfer, *i.e.*, inferring isoforms. Each cluster of expressed segments defines an instance of the isoform inference problem. Denote such an instance as $I(S, R, T, d)$, where $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is the set of expressed segments in the cluster, R the set of short (single-end and paired-end) reads mapped to the segments in the cluster, T the set of start-end segment pairs, and d a function such that $d(i), s_i \in S$, denotes the number of single-end reads mapped to segment s_i and $d(i, j), 1 \leq i < j \leq |S|$, denotes the number of single-end reads mapped to junction (s_i, s_j).

The inference procedure is summarized in Algorithm 1. It first enumerates all the valid isoforms in step 1. However, for a cluster with a large number of expressed segments and isoforms, the number of valid isoforms could be too large to be enumerated efficiently even though conditions I, II and/or III could be used to filter out many invalid isoforms. Therefore, the algorithm enumerates valid isoforms with high expression levels first, where the expression level of an isoform is defined by the least number of single-end reads on any junction of the isoform. The enumeration terminates when a preset number (denoted as γ) of valid isoforms are enumerated. The parameter γ is used to avoid the rare cases that the number of valid isoforms is too large to be handled by subsequent steps of IsoInfer. We set $\gamma = 1000$ by default based on our empirical knowledge of the real data considered in section 4. For example, over 97.5%, 98.5%, and 99% cases, the number of valid isoforms is no more than 1000 in the tests on mouse brain, liver and muscle tissues, respectively, when the exact boundary and TSS-PAS information is extracted from the UCSC knownGene table. The

impact of the omitted isoforms is minimized because highly expressed isoforms are enumerated first.

A short read r is *validated* by a set of isoforms if the set contains an isoform f such that $f[i] = 1$ when $r[i] = 1$. A start-end segment pair is validated by a set of isoforms if this pair is the start-end segment pair of some f in the set. A set of isoforms is a *feasible solution* of $I(S, R, T, d)$ if every read in R and start-end segment pair in T are validated by the set. Due to possible noise in sequencing and the incompleteness of the enumeration of valid isoforms in step 1, it may happen that some reads or start-end segment pairs are not supported by the set of isoforms F enumerated in step 1. Step 2 of the algorithm removes such invalidated reads and start-end segment pairs to make F feasible.

Algorithm 1 IsoformInference. Given an instance $I(S, R, T, d)$, the algorithm infers a set of isoforms to explain the read data.

1: Among all segment junctions of an isoform f, denote $m(f)$ as the minimum number of single-end reads mapped to any of these junctions. Enumerate all the valid isoforms f in the descending order of $m(f)$ until a preset number (γ) of valid isoforms is obtained. Denote the set of all the enumerated valid isoforms as F.
2: Remove all the short reads and start-end segment pairs that are not validated by F.
3: **for** $5 \leq u \leq \beta$ **do**
4: $w(f) \leftarrow 0$ for $f \in F$.
5: **for** $0 \leq m \leq |S| - u$ **do**
6: $n \leftarrow m + u$.
7: $V^{(m,n)} \leftarrow BestCombination(I^{(m,n)})$.
8: For each $v \in V^{(m,n)}$, define $G(v) = \{f | f \in F, f^{(m,n)} = v\}$ and for each $f \in G(v)$, let $w(f) = w(f) + 1/|G(v)|$.
9: **end for**
10: Sort F by w in increasing order.
11: **for** $f \in F$ **do**
12: **if** $w(f) < 1$ and $F - \{f\}$ is a feasible solution of I **then**
13: $F \leftarrow F - \{f\}$.
14: **end if**
15: **end for**
16: **end for**
17: $w'(f) \leftarrow 1/w(f)$ for $f \in F$.
18: Solve the weighted set cover instance $(U, \mathcal{C}, w')$, where $U = R \cup T, \mathcal{C} = \{S_f | f \in F\}$, and $r \in S_f$ if r is validated by f for $r \in U$ for each $f \in F$ by the branch-and-bound method implemented in GNU package GLPK. Return the set of the valid isoforms corresponding to the optimal solution of set cover.

To find a subset of valid isoforms to explain the data, a simple idea is to try all possible combinations of the valid isoforms in F and find a minimum combination that can explain all the short reads, as done in procedure *BestCombination* (*i.e.*, Algorithm 2). The procedure BestCombination gradually increases the number

of valid isoforms considered and enumerates all possible combinations of such a number of isoforms until a preset condition is met.

Algorithm 2 BestCombination. Given an instance $I(S, R, F, d)$, find a "best" subset of F such that the read data can be explained by enumerating all possible subsets of F.

```
1: for 1 ≤ i ≤ |S| do
2:    p ← 0 and F' ← ∅.
3:    for each F'' ⊂ F where |F''| = i and F'' is a feasible solution of I do
4:       {z, x} ←QPsolver(I(S, F'', d)).
5:       if p < p-value(z) then
6:          p ← p-value(z) and F' ← F''.
7:       end if
8:    end for
9:    if p ≥ 0.05 then
10:      Return F'.
11:   end if
12: end for
```

However, it is often infeasible to enumerate all possible combinations of the valid isoforms of a given size. When this happens, we decompose an the instance into some sub-instances. In each sub-instance, only a subset of expressed segments are considered. More specifically, for an instance $I(S, R, F, d)$, where F is the set of valid isoforms enumerated, a sub-instance $I^{(m,n)} = I(S^{(m,n)}, R^{(m,n)}, d^{(m,n)}, F^{(m,n)})$, $0 \leq m < n \leq |S|$, is defined concerning the subset $S^{(m,n)} = \{s_{m+1}, \ldots, s_n\}$ of expressed segments of S. It is formally defined as follows. For each $f \in B(|S|)$, define $f^{(m,n)} \in B(n-m)$ and $f^{(m,n)}[i] = f[i+m], 1 \leq i \leq n-m$. In other words, $f^{(m,n)}$ denotes the sub-vector of f spanning the interval $[m+1, n]$. Let $F^{(m,n)} = \{f^{(m,n)} | f \in F\}$, $R^{(m,n)} = \{r^{(m,n)} | r \in R\}$, $d^{(m,n)}(i) = d(i+m), 1 \leq i \leq n-m$, and $d^{(m,n)}(i,j) = d(i+m, j+m), 1 \leq i < j \leq n-m$. Note that the start-end segment information is not needed in sub-instances.

The parameter β appearing in step 3 controls the maximum size of a sub-instance. Larger sub-instances make the results of procedure BestCombination more reliable. However, the execution time of BestCombination increases exponentially with the number of valid isoforms which grows with the size of the sub-instance. Therefore, instead of a fixed size, a set of sub-instance sizes from the interval $[5, \beta]$ are attempted. For a fixed sub-instance size, BestCombination is executed on each sub-instance of the size in step 7. According to the results of BestCombination, each valid isoform is assigned a weight in Step 8 which roughly indicates the frequency that the isoform appears in the combinations found by BestCombination. A subset of valid isoforms with weights less than 1 are removed in steps 11-15 without making F infeasible.

In steps 17 and 18 of the algorithm, a weighted set cover instance is constructed such that an optimal solution implies a subset of valid isoforms with a minimum total weight such that all the short reads and start-end segments

can be explained. The set cover problem can be solved by using the branch-and-bound method implemented in GNU package GLPK, since it involves only small instances.

3 Simulation test results

We test IsoInfer on mouse genes. The reference genomic sequence and known isoforms of all mouse genes are downloaded from UCSC (mm9, NCBI Build 37) [54]. All exon-intron boundaries of the known isoforms are extracted. This dataset contains 26,989 genes and 49,409 isoforms. 16,392 (60.7%) of the genes have only one isoform and 59 (0.2%) of the genes have more than 10 isoforms. 5830 (21.6%) of the genes have only one exon and 384 (1.4%) of the genes have more than 40 exon-intron boundaries. For the simulation study, only genes with at least two known isoforms are used, which result in 10,595 genes. We further extract all the start-end segments and randomly generate relative expression levels of every isoform. Although it would be natural to assume that expression levels follow a uniform distribution, it is reported in [55–57] that the expression levels of isoforms tend to obey a log-normal distribution. Therefore, we consider three types of distributions.

- Base10: For each isoform, a random number r following the standard normal distribution is generated and then 10^r is assigned as the relative expression level of this isoform.
- Base2: For each isoform, a random number r following the standard normal distribution is generated and then 2^r is assigned as the relative expression level of this isoform.
- Uniform: For each isoform, a random number r uniformly generated from [0,1] is assigned as the relative expression level of this isoform.

Then 40M single-end and 10M paired-end short reads are randomly generated according to the relative expression levels of the isoforms. In the simulation, we assume that the span of a paired-end read is a random variable obeying the normal distribution $N(\mu, \sigma^2)$ [58] so we could evaluate the impact of the mean and deviation of the spans of paired-end reads on the performance of IsoInfer. Note that IsoInfer does not depend on this assumption and works for paired-end reads drawn from any distribution.

Finally, IsoInfer is used to recover all the known isoforms using the start-end segments and single-end and paired-end reads. In the simulation, the read lengths of single-end and paired-end reads are 25bps and 20bps, respectively. The parameter α is set to 1 RPKM, $\beta = 7$ and $\gamma = 1000$. We consider three measures of the performance, *sensitivity*, *effective sensitivity* and *precision*. A known isoform is *recovered* if it is in the output of IsoInfer. Sensitivity is defined as the number of recovered isoforms divided by the number of all known isoforms. Specificity is defined as the number of recovered isoforms divided by the number of isoforms inferred. Since IsoInfer only intends to infer isoforms that are sufficiently expressed, it is useful to consider how many sufficiently expressed

isoforms are recovered by the algorithm. Since Theorem 2 shows that an isoform with a sufficiently high expression level is likely to satisfy condition I (*i.e.*, all its exon-intron junctions are supported by the read data) with high probability, we define *effective sensitivity* as the number of recovered isoforms divided by the number of known isoforms whose exon-intron junctions are supported by the read data.

3.1 Calculation of expression levels

To estimate the effectiveness of our QP formulation, we randomly generate Base10 expression levels and single-end short reads on the known mouse isoforms and check whether it can recover the correct expression levels of the known isoforms. For an isoform f with expression level x_f and calculated expression level x'_f, the relative difference $\frac{|x'_f - x_f|}{x_f}$ is used to measure the accuracy of calculation. A simple and widely used method of calculating expression levels of isoforms is based on counting reads mapped to its unique exons and exon junctions [41, 40]. Clearly, this simple strategy fails if the isoform does not have any unique exons or exon junctions. We compare our method with the simple method (simply denoted as *Uniq* in this paper) and the method based on maximum likelihood estimation (MLE) and importance sampling (IS) [29]. The comparison is depicted in Figure 4.

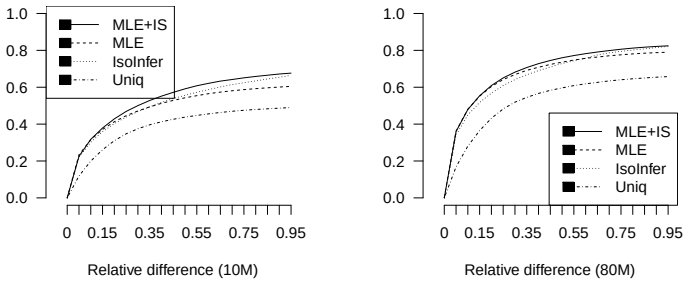

Fig. 4. Comparison of the accuracies of different methods in estimating isoform expression levels. The Y-axis shows the percentage of isoforms whose estimated/calculated expression levels are within a certain relative difference range from the truth. 10 million reads (left) and 80 million reads (right) are sampled in each of the figures.

The comparison shows that MLE followed by IS (MLE+IS) is the most accurate and Uniq is the worst. IsoInfer achieves comparable performances with MLE (followed by IS). An advantage of MLE+IS is that it also provides a 95% confidence interval for each expression level estimation. On the other hand, IsoInfer calculates the expression levels much faster than MLE+IS does (3 minutes vs 3 hours for all mouse genes on a standard desktop PC). The efficiency of IsoInfer makes the search for novel isoforms possible.

3.2 The influence of the distribution of expression levels

In this section, we analyze the influence of the distribution of expression levels on the performance of IsoInfer in inferring isoforms. The distribution of the span of paired-end reads are fixed as the normal distribution $N(300, 30^2)$. The sensitivities and precisions grouped by number of known isoforms per gene are depicted in Figure 5.

The overall sensitivities and precisions of IsoInfer on (Base10, Base2, Uniform) expression levels are (39.7%,75.0%,72.5%) and (79.3%,82.1%,81.3%), respectively. The sensitivities for Base10 expression levels are much lower than those for Base2 and Uniform expression levels, because a large faction of the isoforms are not significant expressed. The effective sensitivity of three cases are 83.5%, 77.4% and 77.4%, respectively. Figure 5 gives detailed sensitivity, effective sensitivity and precision of IsoInfer on genes with a certain number of isoforms. The high effective sensitivity shown in the figure is also confirmed by the sensitivity results on different expression levels, also given in Figure 5 which shows that isoforms with high expression levels are identified with high sensitivities. For example, for Base10 expression levels, isoforms with expression level above 3 (or 6) RPKM are identified with sensitivity above 56.0% (or 81.0%, respectively).

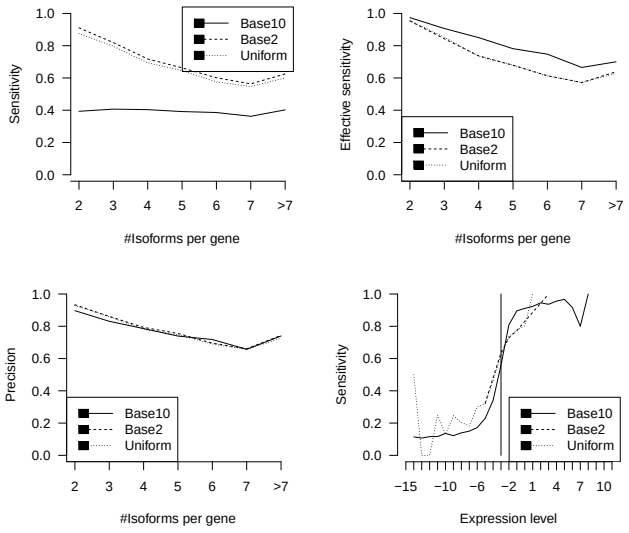

Fig. 5. The sensitivity (top left), effective sensitivity (top right) and precision (bottom left) of IsoInfer on genes with a certain number of isoforms when different distributions of expression levels are generated. The bottom right graph shows the sensitivity of IsoInfer on different expression levels when different distributions of expression level are applied. In the graph, the expression levels are log_2 transformed. Expression level x corresponds to $25 \cdot 2^x$ RPKM. The vertical line corresponds to expression level 1/8 = 3.125 RPKM.

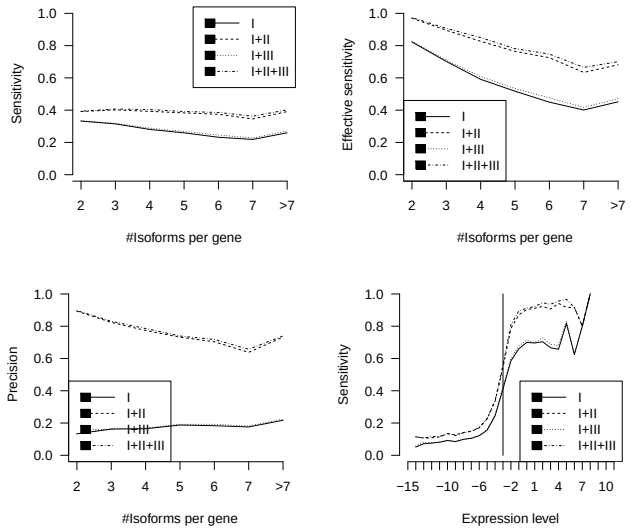

Fig. 6. The sensitivity (top left), effective sensitivity (top right) and precision (bottom left) of IsoInfer on genes with a certain number of isoforms when different combinations of type I, II and III data are provided. The bottom right graph shows the sensitivity of IsoInfer on different expression levels when different combinations of type I, II and III data are used. Again, the expression levels are log_2 transformed. Expression level x corresponds to $25 \cdot 2^x$ RPKM. The vertical line corresponds to expression level $1/8 = 3.125$ RPKM.

3.3 The importance of start-end expressed segment pairs

As mentioned before, single-end short reads are necessary for our algorithm but start-end segment pairs and paired-end reads are optional. To estimate the importance of the last two pieces of information, we compare the results when different types of data are available. Four combinations are possible, denoted as I, I+II, I+III, and I+II+III, where I, II and III correspond to single-end reads (which provide the junction information), start-end segment pairs and paired-end data, respectively. The combination I+III means that the single-end and paired-end read data are available but not the start-end segment pairs. In the simulation, Base10 expression levels are generated and the span distribution of paired-end reads is fixed as $N(300, 30^2)$. Figure 6 shows that start-end segment pairs are much more important than paired-end reads for our algorithm. For example, the sensitivities and precisions for combinations I+II and I+III are (38.9%,78.5%) and (29.5%,16.5%), respectively.

3.4 The influence of span distribution

The span of paired-end reads follows the normal distribution $N(\mu, \sigma^2)$. We run IsoInfer on different combinations of μ and σ. On each combination, 10 million pair-end reads are randomly generated. Since start-end segment pairs are much more important than paired-end reads, as shown in the above subsection, the span distribution should not have a significant influence on the inference results

when start-end segment pairs are available. This is confirmed by Tables 3 and 4 given in [42]. The precision and sensitivity of IsoInfer vary by at most 1.5% when different span distributions are applied.

The above small effect of paired-end read data on the performance of IsoInfer is because the parameter α is set to 1. When a large α is applied, IsoInfer trades sensitivity for precision. For example, when the span distribution of paired-end read is fixed as $N(300, 30^2)$, if α is set to 1, the sensitivity and precision on genes with at least 8 isoforms are 40.2% and 74.0%, respectively. The two measures will change to 35.4% and 78.1%, respectively, when α is set to 20. The performance of IsoInfer when α is set to different values is shown in Tables 5 and 6 of [42].

4 Recovery of known isoforms from real reads

The evaluation uses the following four data sets: (1) known mouse isoforms downloaded from UCSC [54], which contains 49,409 transcripts, (2) mouse mRNAs expressed in various tissues downloaded from UCSC containing 228,779 mRNAs, (3) RNA-Seq data from brain, liver and skeletal muscle tissues of mouse [24], which contains 47,781,892, 44,279,807 and 38,210,358 single-end reads for brain, liver and muscle, respectively, and (4) 104,710 exon junctions that were predicted by TopHat from the above RNA-Seq data for mouse brain tissue [19].

As in the simulation tests, on a specific tissue, one can only expect that isoforms with expression levels above a certain threshold can be detected by RNA-Seq experiments, so as to be inferred by IsoInfer. Given a set of mapped reads, an isoform is said to be *theoretically expressed* if each exon except for the first and last one of this isoform has expression level at least 1 RPKM and every exon junction on this isoform is supported by short reads. (Note that this does not really guarantee that the isoform is actually expressed.) The expression levels of the first and last exons are ignored here because of the possible $3'$ and $5'$ sampling biases in RNA-Seq [27, 24]. The theoretically expressed isoforms among known mouse isoforms and mRNAs are used as benchmarks. Note that the benchmarks change when different tissues are considered, because the expression levels of isoforms change from tissue to tissue.

We have done two group of tests. The first one is to use the TSS-PAS pair and exon-intron boundary information from the known mouse isoforms and/or mRNAs from UCSC and RNA-Seq short reads to infer isoforms. The predicted isoforms are compared with the theoretically expressed isoforms in the corresponding benchmark. An isoform is recovered by IsoInfer if one of isoforms inferred by IsoInfer matches this isoform *precisely* (*i.e.*, the two isoforms contain exactly the same set of exons with exactly the same boundaries). The inference results are shown in Table 1. These results demonstrate that when accurate exon-intron boundary and TSS-PAS pair information is provided, IsoInfer achieves a reasonably good precision, and the precision increases as the size of the benchmark increases. When known mouse isoforms are used, IsoInfer achieves decent effective sensitivities (*i.e.*, 72.9% for brain, 82.2% for liver and 83.0% for muscle). Because mRNAs were collected from different sources and tissues, a large frac-

tion of them may not really be expressed in a specific tissue. Therefore, effective sensitivity of IsoInfer drops when mRNAs are used as the benchmark.

Table 1. The performance of IsoInfer when different exon-intron boundary and TSS-PAS pair information and corresponding benchmarks are used. Here, "Union" means that the exon-intron boundary and TSS-PAS pair information is extracted from both known mouse isoforms and mRNAs and the benchmark is the union of the known mouse isoforms and mRNAs.

	Known isoforms			mRNAs			Union		
Tissue	Brain	Liver	Muscle	Brain	Liver	Muscle	Brain	Liver	Muscle
#Theoretically expressed	18521	12411	11723	87178	72594	69086	101392	82199	78298
Specificity	0.493	0.592	0.627	0.572	0.670	0.712	0.591	0.697	0.737
Effective sensitivity	0.729	0.822	0.830	0.328	0.352	0.366	0.335	0.365	0.381

The second test measures the performance of IsoInfer when the exact exon-intron boundary information is unavailable. The test uses exon-intron boundaries predicted by TopHat from the RNA-Seq read data on the mouse brain tissue and the TSS-PAS pair information extracted from the known mouse isoforms and/or mRNAs. The test results are shown in Table 2. Although it is reported in [19] that over 80% of the exon junctions predicted by TopHat are also exon junctions in the UCSC known mouse isoforms, the inference result on the known mouse isoforms is much worse than the result when exact exon-intron boundary information is provided. On the other hand, when mRNA is used as the benchmark, the exon-intron boundaries provided by TopHat lead IsoInfer to a more aggressive prediction (and thus achieving a better effective sensitivity).

In each of the above tests, the last three steps of IsoInfer shown in Figure 3 took less than 80 minutes on an Intel P8600 processor.

Table 2. The performance of IsoInfer when the exon-intron boundary information is extracted from the exon junctions predicted by TopHat. These results are all on the mouse brain tissue. The TSS-PAS pair information is extracted from the known mouse isoforms and/or mRNAs, depending on the benchmark. Again, "Union" means that the TSS-PAS pair information is extracted from both known mouse isoforms and the benchmark is the union of the known mouse isoforms and mRNAs.

	Known isoforms	mRNAs	Union
Specificity	0.240	0.362	0.378
Effective sensitivity	0.496	0.532	0.508

Acknowledgment

We thank Pirola Yuri for useful discussions. The research is supported in part by a CSC scholarship, NSF grant IIS-0711129, and NIH grants LM008991 and AI078885.

References

1. Boguski, M.S., Tolstoshev, C.M., Bassett, D.E.: Gene discovery in dbEST. Science **265**(5181) (1994) 1993–1994
2. Boguski, M.S.: The turning point in genome research. Trends in Biochemical Sciences **20**(8) (1995) 295 – 296
3. The FANTOM Consortium: The transcriptional landscape of the mammalian genome. Science **309**(5740) (2005) 1559–1563
4. The ENCODE Project Consortium: Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. Nature **447**(7146) (2007) 799–816
5. Weinstock, G.M.: ENCODE: more genomic empowerment. Genome Res **17**(6) (2007) 667–668
6. Bertone, P., Stolc, V., Royce, T.E., Rozowsky, J.S., Urban, A.E., Zhu, X., Rinn, J.L., Tongprasit, W., Samanta, M., Weissman, S., Gerstein, M., Snyder, M.: Global identification of human transcribed sequences with genome tiling arrays. Science **306**(5705) (2004) 2242–2246
7. Kwan, T., Benovoy, D., Dias, C., Gurd, S., Provencher, C., Beaulieu, P., Hudson, T.J., Sladek, R., Majewski, J.: Genome-wide analysis of transcript isoform variation in humans. Nat Genetics (2008)
8. Johnson, J.M., Castle, J., Garrett-Engele, P., Kan, Z., Loerch, P.M., Armour, C.D., Santos, R., Schadt, E.E., Stoughton, R., Shoemaker, D.D.: Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. Science **302**(5653) (2003) 2141–2144
9. Kapranov, P., Cheng, J., Dike, S., Nix, D.A., Duttagupta, R., Willingham, A.T., Stadler, P.F., Hertel, J., Hackermuller, J., Hofacker, I.L., Bell, I., Cheung, E., Drenkow, J., Dumais, E., Patel, S., Helt, G., Ganesh, M., Ghosh, S., Piccolboni, A., Sementchenko, V., Tammana, H., Gingeras, T.R.: RNA maps reveal new RNA classes and a possible function for pervasive transcription. Science **316**(5830) (2007) 1484–1488
10. Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D.H., Johnson, D., Luo, S., McCurdy, S., Foy, M., Ewan, M., Roth, R., George, D., Eletr, S., Albrecht, G., Vermaas, E., Williams, S.R., Moon, K., Burcham, T., Pallas, M., DuBridge, R.B., Kirchner, J., Fearon, K., Mao, J., Corcoran, K.: Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. Nat Biotechnol **18**(6) (2000) 630–634
11. Reinartz, J., Bruyns, E., Lin, J.Z., Burcham, T., Brenner, S., Bowen, B., Kramer, M., Woychik, R.: Massively parallel signature sequencing (MPSS) as a tool for in-depth quantitative gene expression profiling in all organisms. Brief Funct Genomic Proteomic **1**(1) (2002) 95–104
12. Velculescu, V.E., Zhang, L., Vogelstein, B., Kinzler, K.W.: Serial analysis of gene expression. Science **270**(5235) (1995) 484–487
13. Harbers, M., Carninci, P.: Tag-based approaches for transcriptome research and genome annotation. Nat Meth **2**(7) (2005) 495–502
14. Shiraki, T., Kondo, S., Katayama, S., Waki, K., Kasukawa, T., Kawaji, H., Kodzius, R., Watahiki, A., Nakamura, M., Arakawa, T., Fukuda, S., Sasaki, D., Podhajska, A., Harbers, M., Kawai, J., Carninci, P., Hayashizaki, Y.: Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. Proceedings of the National Academy of Sciences of the United States of America **100**(26) (2003) 15776–15781

15. Kodzius, R., Kojima, M., Nishiyori, H., Nakamura, M., Fukuda, S., Tagami, M., Sasaki, D., Imamura, K., Kai, C., Harbers, M., Hayashizaki, Y., Carninci, P.: CAGE: cap analysis of gene expression. Nat Meth **3**(3) (2005) 211–222
16. Kim, J.B., Porreca, G.J., Song, L., Greenway, S.C., Gorham, J.M., Church, G.M., Seidman, C.E., Seidman, J.G.: Polony multiplex analysis of gene expression (PMAGE) in mouse hypertrophic cardiomyopathy. Science **316**(5830) (2007) 1481–1484
17. Ng, P., Wei, C.L., Sung, W.K., Chiu, K.P., Lipovich, L., Ang, C.C., Gupta, S., Shahab, A., Ridwan, A., Wong, C.H., Liu, E.T., Ruan, Y.: Gene identification signature (GIS) analysis for transcriptome characterization and genome annotation. Nat Methods **2** (2005) 105 – 111
18. Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., Snyder, M.: The transcriptional landscape of the yeast genome defined by RNA sequencing. Science **320**(5881) (2008) 1344–1349
19. Trapnell, C., Pachter, L., Salzberg, S.L.: TopHat: discovering splice junctions with RNA-Seq. Bioinformatics **25**(9) (2009) 1105–1111
20. Graveley, B.R.: Molecular biology: power sequencing. Nature **453**(7199) (2008) 1197–1198
21. Yassour, M., Kaplan, T., Fraser, H.B., Levin, J.Z., Pfiffner, J., Adiconis, X., Schroth, G., Luo, S., Khrebtukova, I., Gnirke, A., Nusbaum, C., Thompson, D.A., Friedman, N., Regev, A.: Ab initio construction of a eukaryotic transcriptome by massively parallel mRNA sequencing. Proceedings of the National Academy of Sciences **106**(9) (2009) 3264–3269
22. Wilhelm, B.T., Marguerat, S., Watt, S., Schubert, F., Wood, V., Goodhead, I., Penkett, C.J., Rogers, J., Bähler, J.: Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. Nature **453**(7199) (2008) 1239–43
23. Cloonan, N., Forrest, A.R., Kolle, G., Gardiner, B.B., Faulkner, G.J., Brown, M.K., Taylor, D.F., Steptoe, A.L., Wani, S., Bethel, G., Robertson, A.J., Perkins, A.C., Bruce, S.J., Lee, C.C., Ranade, S.S., Peckham, H.E., Manning, J.M., McKernan, K.J., Grimmond, S.M.: Stem cell transcriptome profiling via massive-scale mRNA sequencing. Nat Methods (2008)
24. Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., Wold, B.: Mapping and quantifying mammalian transcriptomes by RNA-Seq. Nat Methods **5**(7) (2008) 621 – 628
25. Marioni, J., Mason, C., Mane, S., Stephens, M., Gilad, Y.: RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. Genome Res **18**(9) (2008) 1509–17
26. Sultan, M., Schulz, M.H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H., Yaspo, M.L.: A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. Science **321**(5891) (2008) 956–960
27. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nature reviews. Genetics (2008)
28. Lacroix, V., Sammeth, M., Guigó, R., Bergeron, A.: Exact transcriptome reconstruction from short sequence reads. In: WABI '08: Proceedings of the 8th international workshop on Algorithms in Bioinformatics, Berlin, Heidelberg, Springer-Verlag (2008) 50–63
29. Jiang, H., Wong, W.H.: Statistical inferences for isoform expression in RNA-Seq. Bioinformatics **25**(8) (2009) 1026–1032

30. Pagani, F., Baralle, F.E.: Genomic variants in exons and introns: identifying the splicing spoilers. Nat Rev Genet **5**(5) (2004) 389–396
31. Srebrow, A., Kornblihtt, A.R.: The connection between splicing and cancer. J Cell Sci **119**(13) (2006) 2635–2641
32. Williams, W.V.: Editorial hot topic: Transcriptome analysis in drug development (executive editor: William v. williams). Current Molecular Medicine **5** (2005) 1–2(2)
33. Heber, S., Alekseyev, M., Sze, S.H., Tang, H., Pevzner, P.A.: Splicing graphs and EST assembly problem. Bioinformatics **18**(suppl-1) (2002) S181–188
34. Sammeth, M., Valiente, G., Guigó, R.: Bubbles: alternative splicing events of arbitrary dimension in splicing graphs. In Vingron, M., Wong, L., eds.: RECOMB. Volume 4955 of Lecture Notes in Computer Science., Springer (2008) 372–395
35. Xing, Y., Resch, A., Lee, C.: The multiassembly problem: reconstructing multiple transcript isoforms from EST fragment mixtures. Genome Res **14**(3) (2004) 426–441
36. Bonizzoni, P., Mauri, G., Pesole, G., Picardi, E., Pirola, Y., Rizzi, R.: Detecting alternative gene structures from spliced ESTs: a computational approach. Journal of Computational Biology **16**(1) (2009) 43–66
37. Djebali, S., Kapranov, P., Foissac, S., Lagarde, J., Reymond, A., Ucla, C., Wyss, C., Drenkow, J., Dumais, E., Murray, R.R., Lin, C., Szeto, D., Denoeud, F., Calvo, M., Frankish, A., Harrow, J., Makrythanasis, P., Vidal, M., Salehi-Ashtiani, K., Antonarakis, S.E., Gingeras, T.R., Guigó, R.: Efficient targeted transcript discovery via array-based normalization of RACE libraries. Nat Meth **5**(7) (2008) 629–635
38. Salehi-Ashtiani, K., Yang, X., Derti, A., Tian, W., Hao, T., Lin, C., Makowski, K., Shen, L., Murray, R.R., Szeto, D., Tusneem, N., Smith, D.R., Cusick, M.E., Hill, D.E., Roth, F.P., Vidal, M.: Isoform discovery by targeted cloning, 'deep-well' pooling and parallel sequencing. Nat Meth **5**(7) (2008) 597 – 600
39. Fullwood, M.J., Wei, C.L., Liu, E.T., Ruan, Y.: Next-generation DNA sequencing of paired-end tags (PET) for transcriptome and genome analyses. Genome Res **19**(4) (2009) 521–532
40. Pan, Q., Shai, O., Lee, L.J., Frey, B.J., Blencowe, B.J.: Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. Nat Genet **40**(12) (2008) 1413–1415
41. Wang, E.T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., Kingsmore, S.F., Schroth, G.P., Burge, C.B.: Alternative isoform regulation in human tissue transcriptomes. Nature **456**(7221) (2008) 470–476
42. Feng, J., Li, W., Jiang, T.: Inference of isoforms from short sequence reads. Manuscript (Jan 2010)
43. Breitbart, R.E., Andreadis, A., Nadal-Ginard, B.: Alternative splicing: a ubiquitous mechanism for the generation of multiple protein isoforms from single genes. Annual Review of Biochemistry **56**(1) (1987) 467–495
44. Sammeth, M., Foissac, S., Guig, R.: A general definition and nomenclature for alternative splicing events. PLoS Comput Biol **4**(8) (2008) e1000147
45. Langmead, B., Trapnell, C., Pop, M., Salzberg, S.: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biology **10**(3) (2009) R25
46. Li, H., Ruan, J., Durbin, R.: Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res **18**(11) (2008) 1851–1858
47. Li, R., Li, Y., Kristiansen, K., Wang, J.: SOAP: short oligonucleotide alignment program. Bioinformatics **24**(5) (2008) 713–714

48. Cloonan, N., Xu, Q., Faulkner, G.J., Taylor, D.F., Tang, D.T., Kolle, G., Grimmond, S.M.: RNA-MATE: a recursive mapping strategy for high-throughput RNA-sequencing data. Bioinformatics (2009) btp459
49. Alkan, C., Kidd, J.M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., Kitzman, J.O., Baker, C., Malig, M., Mutlu, O., Sahinalp, S.C., Gibbs, R.A., Eichler, E.E.: Personalized copy number and segmental duplication maps using next-generation sequencing. Nat Genet **41**(10) (2009) 1061–1067
50. Hashimoto, T., Hoon, M.J.d., Grimmond, S.M., Daub, C.O., Hayashizaki, Y., Faulkner, G.J.: Probabilistic resolution of multi-mapping reads in massively parallel sequencing data using MuMRescueLite. Bioinformatics (2009) btp438
51. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2007)
52. Goldfarb D, I.A.: A numerically stable dual method for solving strictly convex quadratic programs. Math Program **27** (1983) 1–33
53. Korbel, J., Abyzov, A., Mu, X., Carriero, N., Cayting, P., Zhang, Z., Snyder, M., Gerstein, M.: PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. Genome Biology **10**(2) (2009) R23
54. Karolchik, D., Kuhn, R.M., Baertsch, R., Barber, G.P., Clawson, H., Diekhans, M., Giardine, B., Harte, R.A., Hinrichs, A.S., Hsu, F., Kober, K.M., Miller, W., Pedersen, J.S., Pohl, A., Raney, B.J., Rhead, B., Rosenbloom, K.R., Smith, K.E., Stanke, M., Thakkapallayil, A., Trumbower, H., Wang, T., Zweig, A.S., Haussler, D., Kent, W.J.: The UCSC genome browser database: 2008 update. Nucl Acids Res **36**(Database issue) (2008) D773–9
55. Alter, M.D., Rubin, D.B., Ramsey, K., Halpern, R., Stephan, D.A., Abbott, L.F., Hen, R.: Variation in the large-scale organization of gene expression levels in the hippocampus relates to stable epigenetic variability in behavior. PLoS ONE **3**(10) (10 2008) e3344
56. Konishi, T.: Three-parameter lognormal distribution ubiquitously found in cdna microarray data and its application to parametric data treatment. BMC Bioinformatics **5**(1) (2004) 5
57. Wijaya, E., Harada, H., Horton, P.: Modeling the marginal distribution of gene expression with mixture models. In: FGCN '08: Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking, Washington, DC, USA, IEEE Computer Society (2008) 84–89
58. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: MetaSima sequencing simulator for genomics and metagenomics. PLoS ONE **3**(10) (2008) e3373

Appendix

The proof of Theorem 1

Proof. For simplicity, we assume that the distributions involved in the following proof are discrete. Let $q(x)$ be the probability that the span of a randomly generated paired-end read is x, and $p(x)$ the probability of a uniformly randomly selected position from all isoforms being at position x on the given isoform. Every paired-end read can be represented by its start (or center or end) position and span uniquely. Denote the set of all possible start positions as Ψ and the set of all possible spans as Ω. Let $V \subset \Psi \times \Omega$ defines the set of paired-end reads that have start positions in the first interval and end positions in the third interval. Under strategy (a), the probability of a uniformly randomly generated paired-end read being in V is:

$$\begin{aligned} P_a &= \sum_{\psi \in \Psi} (\sum_{\omega | (\psi,\omega) \in V} q(\omega)) p(\psi) \\ &= \sum_{(\psi,\omega) \in V} q(\omega)\alpha/10^9 \end{aligned}$$

Similarly, we define the set of possible center positions of paired-end reads as Ψ'. Let $V' \subset \Psi' \times \Omega$ define the set of paired end reads that have start positions in the first interval and end positions in the third interval. Under strategy (b), the probability of a uniformly randomly generated paired-end read being in V' is:

$$P_b = \sum_{(\psi,\omega) \in V'} q(\omega)\alpha/10^9$$

Because $|\{\psi|(\psi,\omega) \in V'\}| = |\{\psi|(\psi,\omega) \in V\}|$ for $\omega \in \Omega$, we have $P_a = P_b$. The argument is also applicable to case when strategy (c) is applied.

When strategy (a) is applied and the end position of the third interval is not the end position of the given isoform, if the start position of a uniformly randomly generated paired-end read is $i, 0 \leq i < w_1$ in the first interval, then the probability of the end position of this paired-end read being in the third interval is

$$p_i = P(X \leq u(i)) - P(X \leq l(i)) = \int_{l(i)}^{u(i)} h(x)dx$$

where $l(i) = w_1 - i + w_2, u(i) = w_1 - i + w_2 + w_3$. When the end position of the third interval is the end position of the given isoform and strategy (a) is applied, we have

$$p_i = P(X \leq +\infty) - P(X \leq l(i)) = \int_{l(i)}^{+\infty} h(x)dx \geq \int_{l(i)}^{u(i)} h(x)dx$$

Because the start position of a paired-end read is uniformly randomly selected,

$$P_a = 10^{-9}\alpha \sum_{0 \leq i < w_1} p_i \geq 10^{-9}\alpha \sum_{0 \leq i < w_1} \int_{l(i)}^{u(i)} h(x)dx = P_0$$

Because M paired-end reads are generated, the probability that none of the reads have start positions in the first interval and end positions in the third interval is $(1-P_a)^M \leq (1-P_0)^M = P_{M,h,\alpha}(w_1, w_2, w_3) \approx e^{-MP_0}$.

Similar arguments hold when strategies (b) and (c) are applied to generate the reads. ■

The proof of Theorem 2

Proof. If expression level of y RPKM of the isoform f corresponds to one transcript of f, the total number of the expressed transcripts of f is x/y. Based on the definition of RPKM, $y = (10^6 \cdot 10^3)/L_0 = 10^9/L_0$, where L_0 is the total length of all the expressed transcripts with duplications. For any junction, the probability of a read falling into this junction is xL/yL_0. So, the probability that none of the reads fall into this junction is $(1-xL/yL_0)^M \approx e^{-xLM/yL_0} = e^{-xLM/10^9}$. In order for this isoform to be valid, each of the $t-1$ junctions contains at least one read. Therefore, the probability of this isoform being valid is $(1-e^{-xLM/10^9})^{t-1}$. Note that the sequencing noise does not decrease the above probability although it may provide some spurious junction reads. ■

Table 3. Sensitivities for various span distributions grouped by the number of isoforms per gene. Here, "No PE reads" means that no paired-end reads are applied. The first column lists various combinations of the mean and standard deviation in the span (normal) distributions considered. The corresponding effective sensitivities range from 63.4% to 97.4%.

#isoforms per gene	2	3	4	5	6	7	≥ 8
No PE reads	0.392	0.402	0.392	0.383	0.374	0.346	0.391
300, 10	0.393	0.406	0.402	0.391	0.385	0.357	0.402
300, 30	0.393	0.407	0.404	0.392	0.386	0.362	0.402
300, 50	0.393	0.407	0.402	0.393	0.385	0.366	0.402
300, 100	0.393	0.408	0.404	0.395	0.385	0.359	0.405
1100, 110	0.387	0.401	0.399	0.395	0.392	0.363	0.403
3000, 300	0.392	0.404	0.403	0.400	0.390	0.366	0.413

Table 4. Specificities for various span distributions grouped by the number of isoforms per gene. The first column lists various combinations of the mean and standard deviation in the span (normal) distributions considered.

#isoforms per gene	2	3	4	5	6	7	≥ 8
No PE reads	0.893	0.824	0.774	0.732	0.704	0.638	0.733
300, 10	0.897	0.830	0.784	0.738	0.717	0.648	0.740
300, 30	0.897	0.831	0.786	0.739	0.718	0.657	0.740
300, 50	0.897	0.830	0.784	0.740	0.714	0.663	0.737
300, 100	0.896	0.830	0.786	0.743	0.713	0.649	0.740
1100, 110	0.896	0.829	0.782	0.739	0.720	0.657	0.729
3000, 300	0.896	0.828	0.776	0.741	0.709	0.648	0.736

Table 5. Sensitivities grouped by the number of isoforms per gene when α is set to various values.

#isoforms per gene	2	3	4	5	6	7	≥ 8
α =1	0.393	0.407	0.404	0.392	0.386	0.362	0.402
2	0.379	0.390	0.388	0.373	0.374	0.347	0.389
3	0.375	0.381	0.381	0.363	0.362	0.332	0.381
4	0.370	0.376	0.375	0.355	0.356	0.325	0.377
5	0.368	0.371	0.371	0.353	0.353	0.325	0.375
6	0.364	0.367	0.366	0.348	0.349	0.324	0.374
7	0.363	0.364	0.363	0.344	0.347	0.323	0.370
8	0.361	0.362	0.361	0.342	0.345	0.323	0.370
9	0.360	0.361	0.360	0.340	0.344	0.323	0.367
10	0.359	0.361	0.358	0.340	0.343	0.323	0.367
20	0.350	0.350	0.340	0.327	0.332	0.306	0.354

Table 6. Specificities grouped by the number of isoforms per gene when α is set to various values.

#isoforms per gene	2	3	4	5	6	7	≥ 8
α =1	0.897	0.831	0.786	0.739	0.718	0.657	0.740
2	0.895	0.833	0.785	0.738	0.721	0.664	0.738
3	0.897	0.835	0.786	0.741	0.724	0.659	0.741
4	0.898	0.838	0.792	0.741	0.730	0.662	0.751
5	0.900	0.842	0.797	0.749	0.734	0.668	0.757
6	0.900	0.845	0.797	0.750	0.738	0.675	0.762
7	0.901	0.844	0.797	0.749	0.742	0.686	0.766
8	0.902	0.845	0.799	0.748	0.747	0.696	0.772
9	0.902	0.847	0.803	0.752	0.752	0.698	0.771
10	0.902	0.849	0.804	0.754	0.754	0.703	0.774
20	0.905	0.853	0.804	0.758	0.760	0.697	0.781