IQSeq: Integrated Isoform Quantification Analysis Based on Next-Generation Sequencing

Jiang Du[1], Jing Leng[2], Lukas Habegger[2], Andrea Sboner[3], Drew McDermott[1], Mark Gerstein[1,2,3]*

[1] Department of Computer Science, Yale University, New Haven, Connecticut, United States of America, [2] Program in Computational Biology and Bioinformatics, Yale University, New Haven, Connecticut, United States of America, [3] Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, Connecticut, United States of America

Abstract

With the recent advances in high-throughput RNA sequencing (RNA-Seq), biologists are able to measure transcription with unprecedented precision. One problem that can now be tackled is that of isoform quantification: here one tries to reconstruct the abundances of isoforms of a gene. We have developed a statistical solution for this problem, based on analyzing a set of RNA-Seq reads, and a practical implementation, available from archive.gersteinlab.org/proj/rnaseq/IQSeq, in a tool we call IQSeq (Isoform Quantification in next-generation Sequencing). Here, we present theoretical results which IQSeq is based on, and then use both simulated and real datasets to illustrate various applications of the tool. In order to measure the accuracy of an isoform-quantification result, one would try to estimate the average variance of the estimated isoform abundances for each gene (based on resampling the RNA-seq reads), and IQSeq has a particularly fast algorithm (based on the Fisher Information Matrix) for calculating this, achieving a speedup of ~ 500 times compared to brute-force resampling. IQSeq also calculates an information theoretic measure of overall transcriptome complexity to describe isoform abundance for a whole experiment. IQSeq has many features that are particularly useful in RNA-Seq experimental design, allowing one to optimally model the integration of different sequencing technologies in a cost-effective way. In particular, the IQSeq formalism integrates the analysis of different sample (i.e. read) sets generated from different technologies within the same statistical framework. It also supports a generalized statistical partial-sample-generation function to model the sequencing process. This allows one to have a modular, "plugin-able" read-generation function to support the particularities of the many evolving sequencing technologies.

Citation: Du J, Leng J, Habegger L, Sboner A, McDermott D, et al. (2012) IQSeq: Integrated Isoform Quantification Analysis Based on Next-Generation Sequencing. PLoS ONE 7(1): e29175. doi:10.1371/journal.pone.0029175

Editor: Andrey Rzhetsky, University of Chicago, United States of America

Received August 17, 2011; **Accepted** November 22, 2011; **Published** January 6, 2012

Funding: These authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: Mark.Gerstein@yale.edu

Introduction

The concepts of genes and isoforms have evolved and become more complex [1]: the discovery of splicing [2–4] revealed that the gene was a series of exons, coding for, in some cases, discrete protein domains, and separated by long noncoding stretches called introns. With alternative splicing, one genetic locus could code for multiple different mRNA transcripts (isoform transcripts). This discovery complicated the concept of the gene radically. For instance, as of 2007, the GENCODE annotation [5] contained on average 5.4 transcripts per locus.

With the recent development of high-throughput RNA sequencing (RNA-Seq) technology, it is possible for biologists to measure transcription with unprecedented precision. One problem that can now be tackled is that of isoform quantification, where one tries to reconstruct the abundances of similar isoforms based on a set of RNA-Seq reads. Various methods have been developed to solve this problem. In previous work, researchers proposed different statistical frameworks to solve this problem. Xing et al. [6] proposed a maximum likelihood problem, an expectation maximization solution, and a Fisher information measurement for performance estimation; Jiang et al. [7], based on Poisson model assumption, formulated a maximum likelihood problem and its numerical solution, and also utilized the observed Fisher information matrix to sample the posterior distribution of isoform quantity; Trapnell et al. [8] used variable read-length model (normal distribution by default) and a sampling method similar to [7] to derive the posterior distribution of isoform quantity; Richard et al. [9] with a Poisson model, also used bootstrapping to study the robustness of their method against non-uniform sequencing effects; Lacroix et al. [10] studied the conditions under which the problem can be solved, revealing that although neither single nor paired-end sequencing guarantee a unique solution, paired-end reads may be sufficient to solve the vast majority of the transcript variants in practice.

These studies, however, have not fully addressed the problem of isoform quantification in a couple of respects: First of all, they usually assume that only one sequencing technique is used in an experiment, and that the reads are uniformly sampled along the transcripts. These are not necessarily good approximations to real data. Second, while some theoretical results have been presented on estimating the accuracy (e.g. average variance) of quantification results, there does not yet exist a method to efficiently compute these measurements other than using brute-force simulation, which is computationally infeasible in large scale expriments involving tens of thousands of genes and millions of sequencing reads. On the other hand, fast estimation of quantification accuracy would not only enable researchers to better understand the analysis results being obtained, but also will be useful in RNA-Seq experiment design to optimally integrate different sequencing technologies in a cost-efficient way.

In order to fill in these gaps, we have developed a generalized statistical solution for the problem of isoform quantification, and a practical implementation in a tool we call IQSeq (Isoform Quantification in next-generation SEQuencing). IQSeq has the following features which represent improvements over previous work in isoform quantification in the following aspects:

1. It has a generalized statistical read generation function during the sequencing process (i.e. a customizable function describing how reads are randomly sampled from isoforms). This provide a flexible way to incorporate characteristics of different sequencing technologies (e.g. $3'$ end sequencing bias of transcripts).
2. It integrates the analysis of different sample sets generated from different sampling technologies (e.g. long and short reads).
3. It has a fast algorithm for estimating the average variance of the results provided by our expectation maximization based solution.
4. Given the estimated isoform abundance output, IQSeq also provides an information theoretical method to measure the overall transcriptome complexity.

In this paper, we will first introduce a mathematical definition of the generalized partial sampling and distribution estimation problem (which IQSeq is based on), and provide a expectation maximization based iterative solution. Then we discuss in detail on how to estimate the performance of this solution using Fisher information based heuristics, and present fast algorithms that implement the computation of these heuristics. Finally, we show results of applying our methods to both simulated and real-world data, illustrating scenarios where such integrated analysis can be the most informative.

Methods

First, we formally define the isoform quantification with multiple sequencing technologies as a generalized statistical partial sampling problem, and present a computational solution based on maximum likelihood estimation and expectation maximization. We then show both analytical results and practical fast algorithms to estimate the average variance of the solution on isoform quantification, and compare their computational complexity against brute-force methods. We present the main theoretical results in this section, and detailed derivations can be found in Text S1.

Problem Definition

We start by defining the generalized process of batch partial sampling, which represents the sequencing process in RNA-Seq experiments, and the relationships between partial samples and the objects being sampled.

Definition 1. (Batch Partial Sampling) Let $I=\{I_1,...,I_K\}$ be all the possible isoforms for a given gene, with relative abundances $\Theta=(\theta_1,...,\theta_K)^T$, where $\sum_{k=1}^{K}\theta_k=1$. We assume that there are M different partial sampling methods (sequencing techniques with difference characteristics, e.g. long/medium/short, single/paired end): $Samp_1,...,Samp_M$, and let S denote all the samples (reads): $S=\{s \text{ from } Samp_m|m=1,...,M\}$. We also define $\delta_{s,k}=Ind$ (partial sample (read) s is compatible with I_k), where Ind is the indicator function. There are in total $N=\sum_{m=1}^{M}N_m$ samples, where N_m is the total number of partial samples from $Samp_m$.

Here we assume a two-step sampling process: First, a sampling method $Samp_m$ chooses an isoform instance I_k according to Θ. Second, the sampling method generates a partial sample s according to a local partial sample generation model (the read generation function) $G_{s,k}^{(m)}=Pr$ (generating $s|I_k,Samp_m$).

Definition 2. (Distribution Estimation based on Batch Partial Samples) *Given* I, and S as defined in Definition 1, estimate Θ.

As shown in Figure 1, I are the isoforms with different relative abundances Θ, and S are the single- and paired-end reads whose sequences align with part of this gene region. Some of these reads (e.g. read 2, 3 and 5) are compatible with multiple isoforms. The ultimate problem is to estimate Θ based on I and S, i.e., reconstructing a distribution based on partial observations.

In the remaining part of this paper, we will use two notations to describe a partial sample s: $s_{m,i}$ is the ith sample from $Samp_m$; and $s_{[a,b)}^{(k)}$ stands for a partial sample from I_k, starting (inclusive) from position a and ending (exclusive) at b in that isoform. We also define exons as those nodes in the splicing graph of a gene, so that there are no exons that overlap with each other (i.e. an exon in a transcript may be a combination of multiple nodes of the splicing graph). We have included in our software package a preprocessing tool for grouping transcripts into gene clusters and formulating corresponding splicing graphs.

Maximum Likelihood Estimation (MLE)

Definition 2 does not give an explicit criterion for a “good” estimation of Θ. Since the problem is defined in a statistical sampling framework, it is natural to consider using Maximum Likelihood as such a criterion.

Definition 3. (Maximum-Likelihood Distribution Estimation based on Batch Partial Samples) Given I, and S as defined in Definition 1, find $\hat{\Theta}$ such that:

$$\hat{\Theta}=argmax_{\Theta}\log(Pr(S|\Theta)) \tag{1}$$

By plugging in the partial samples $s_{m,i}$s and $G_{s,k}^{(m)}$s, we can rewrite the formula above as follows:

$$\hat{\Theta}=argmax_{\Theta}\sum_{m=1}^{M}\sum_{s=s_{m,*}}\log\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)} \tag{2}$$

In the next subsection, we demonstrate how this problem can be solved by introducing a hidden variable $Z_{s,k}$ and using the technique of Expectation Maximization [11].

Applying the Expectation Maximization Method

We define $Z_{s,k}=Ind(s \text{ is from } I_k)$, which are the hidden variables in this problem. Since Expectation Maximization gives an iterative solution, we denote the estimation for Θ in the nth step as $\Theta^{(n)}$, and further define $\zeta_{s,k}^{(n)}=\mathbf{E}_{Z|S,\Theta^{(n)}}[Z_{s,k}]$, which is the expectation of $Z_{s,k}$ given $\Theta^{(n)}$ (the estimated paramters at the nth step) and the reads S.

$$\zeta_{s,k}^{(n)}=\mathbf{E}_{Z|S,\Theta^{(n)}}[Z_{s,k}] \tag{3}$$

$$=\frac{\delta_{s,k}\theta_k^{(n)}G_{s,k}^{(m)}}{\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}^{(n)}G_{s,k'}^{(m)}} \quad , \tag{4}$$

where s is generated by $Samp_m$.

By performing an E step that computes

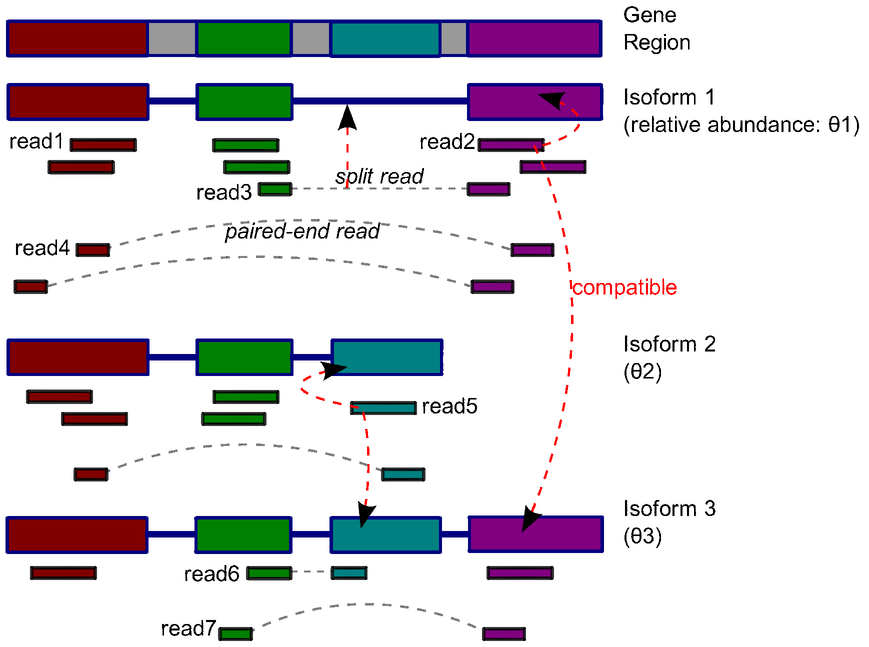


Figure 1. Reads (partial samples) in the isoform quantification problem.
doi:10.1371/journal.pone.0029175.g001

$$Q^{(n)}(\Theta) = \mathbf{E}_{Z|S,\Theta^{(n)}}[\log(Pr(Z,S|\Theta))] \tag{5}$$

$$= \sum_{m=1}^{M} \sum_{s=s_{m,*}} \sum_{k=1}^{K} \zeta_{s,k}^{(n)} \log\theta_k + C \tag{6}$$

and a M step which maximizes $Q^{(n)}(\Theta)$ with constraint: $\sum_{k=1}^{K} \theta_k = 1$, we have:

$$\theta_k^{(n+1)} = \frac{\sum_{m=1}^{M} \sum_{s=s_{m,*}} \frac{\delta_{s,k}\theta_k^{(n)} G_{s,k}^{(m)}}{\sum_{k'=1}^{K} \delta_{s,k'}\theta_{k'}^{(n)} G_{s,k'}^{(m)}}}{N} \tag{7}$$

as the new estimation for Θ.

The iterative estimation in Equation 7 is intuitively consistent with the case of estimating a distribution based on full samples: consider the scenario in which for each s, there is only one $k \in 1,\ldots,K$ satisfying $\delta_{s,k} > 0$, the right hand side of Equation 7 thus becomes $\frac{\sum_{m=1}^{M} \sum_{s=s_{m,*}} \delta_{s,k}}{N}$, which is exactly how the distribution estimation problem with traditional full samples can be solved. In the case of partial samples, our solution provides a way to adjust the "weight" each sample s contributes to the θ_ks of different objects.

Analyzing the Performance of Estimation

Given $\hat{\Theta}$ obtained from the MLE solution presented in the previous section, we would like to understand how much this estimate will deviate from the "true" Θ on average. Here we focus on the variance of the $\hat{\Theta}$, which describes how stable the MLE result is over many different partial sample sets (obtained via additional experiments or re-sampling) drawn from the same isoform set:

$$Average\left(var(\hat{\theta}_k)\right) = \frac{\sum_{k=1}^{K-1} var(\hat{\theta}_k)}{K-1} \tag{8}$$

As we will show later, although brute-force simulation can be performed to obtain a relatively accurate estimation of this measurement, it is may become computationally intractable when there are too many reads and genes to be considered. We thus propose to use a Fisher information based heuristic for estimating $Average\left(var(\hat{\theta}_k)\right)$, and present a fast algorithm to compute the exact value of this heuristic.

We first introduce the Fisher information matrix [12,13] as a basis for further discussion. The Fisher information is a way of measuring the amount of information that the random samples S carries about the unknown parameter Θ upon which the likelihood function of Θ, $Pr(S|\Theta)$, depends. An important use of the Fisher information matrix in statistical analyses is its contribution to the calculation of the covariance matrices of estimates of parameters fitted by maximum likelihood.

Let $\theta_1,\ldots,\theta_{K-1}$ be the free parameters, and $\theta_K = 1 - \sum_{k=1}^{K-1} \theta_k$.

Definition 4. (Observed Fisher information matrix).

$$\Im(\Theta)_{p,q} = -\frac{\partial^2 \log(Pr(S|\Theta))}{\partial\theta_p \partial\theta_q}, \text{ where } p,q = 1,\ldots,K-1 \tag{9}$$

$$= \sum_{m=1}^{M} \sum_{s=s_{m,*}} \frac{\left(\delta_{s,p} G_{s,p}^{(m)} - \delta_{s,K} G_{s,K}^{(m)}\right)\left(\delta_{s,q} G_{s,q}^{(m)} - \delta_{s,K} G_{s,K}^{(m)}\right)}{\left[\sum_{k=1}^{K} \delta_{s,k}\theta_k G_{s,k}^{(m)}\right]^2} \tag{10}$$

Definition 5. (Expected Fisher information matrix).

$$\mathcal{I}(\Theta)_{p,q} = \mathbf{E}\left[\Im(\Theta)_{p,q}\right] \tag{11}$$

Covariance matrix of the maximum likelihood estimator

Let $T(S) = (\hat{\theta}_1,\ldots,\hat{\theta}_{K-1}, 1 - \sum_{k=1}^{K-1} \hat{\theta}_k)^T$, and $\psi(\Theta) = \mathbf{E}[T(S)]$. The Cramér-Rao bound [13] states that:

$$cov_{\Theta}(T(S)) \geq \frac{\partial\psi(\Theta)}{\partial\Theta}[\mathcal{I}(\Theta)]^{-1}\left(\frac{\partial\psi(\Theta)}{\partial\Theta}\right)^{T} \quad , \tag{12}$$

where $[\partial\psi(\Theta)/\partial\Theta]_{u,v} = \partial\psi_u(\Theta)/\partial\theta_v$, $u = 1,...,K$; $v = 1,...,K-1$.

We then estimate $\psi(\Theta)$ by Θ, and use the bound above to estimate the covariance matrix:

$$cov_{\Theta}(T(S)) \approx \begin{bmatrix} & & & -\sum_{k=1}^{K-1}\mathcal{I}_{1,k}^{-1} \\ & \mathcal{I}_{(K-1)\times(K-1)}^{-1} & & \vdots \\ & & & -\sum_{k=1}^{K-1}\mathcal{I}_{K-1,k}^{-1} \\ -\sum_{k=1}^{K-1}\mathcal{I}_{k,1}^{-1} & \cdots & -\sum_{k=1}^{K-1}\mathcal{I}_{k,K-1}^{-1} & \sum_{i=1}^{K-1}\sum_{j=1}^{K-1}\mathcal{I}_{i,j}^{-1} \end{bmatrix}_{K\times K} \tag{13}$$

This means that we only need $\mathcal{I}(\Theta)$ in order to estimate the performance of our MLE with different sampling method combinations.

Heuristic for MLE performance estimation

In order to provide a single value measure for the expected performance of Maximum Likelihood estimation, we propose to use the following heuristic to estimate the average variance of $\hat{\Theta}$:

$$Average\left(var(\hat{\theta}_k)\right) \approx \frac{\sum_{k=1}^{K-1}\frac{1}{\mathcal{I}(\Theta)_{k,k}}}{K-1} \tag{14}$$

This heuristic avoids the potential computational intensive and numerically unstable computation of the inverse of $\mathcal{I}$, and is consistent with the theoretical result on the lower-bound of $var(\hat{\theta})$ in one dimensional case:

$$var(\hat{\theta}) \geq \frac{1}{\mathcal{I}(\theta)} \tag{15}$$

which is a specialization of the result in the previous subsection. In other words, the precision to which we can estimate Θ is fundamentally limited by the Fisher information.

In order to compute this heuristic, all we need is $\mathcal{I}(\Theta)$ itself. However, the brute-force computation (according to Definition 4 and 5) of this matrix will be time-consuming since its time complexity is proportional to the total number of possible sample sets (which in turn grows exponentially with the number of samples). In the next section, we will present algorithms that can compute this matrix in a more efficient fashion.

Efficient Computation of $\mathcal{I}(\Theta)$

First of all, we can decompose $\mathcal{I}(\Theta)$ in the following way:

$$\mathcal{I}(\Theta)_{p,q} = \sum_{m=1}^{M} N_m \mathcal{I}^{(m)}(\Theta)_{p,q} \tag{16}$$

where

$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \mathbf{E}_{s\sim Sampm}\left[-\frac{\partial^2 \log\sum_{k=1}^{K}\delta_{s,k}\theta_k G_{s,k}^{(m)}}{\partial\theta_p\partial\theta_q}\right] \tag{17}$$

is the expected Fisher information matrix of a single partial sample based on $Samp_m$. Thus we need to be able to compute $\mathcal{I}^{(m)}(\Theta)$ in order to obtain $\mathcal{I}(\Theta)$.

Further decomposing $\mathcal{I}^{(m)}(\Theta)$

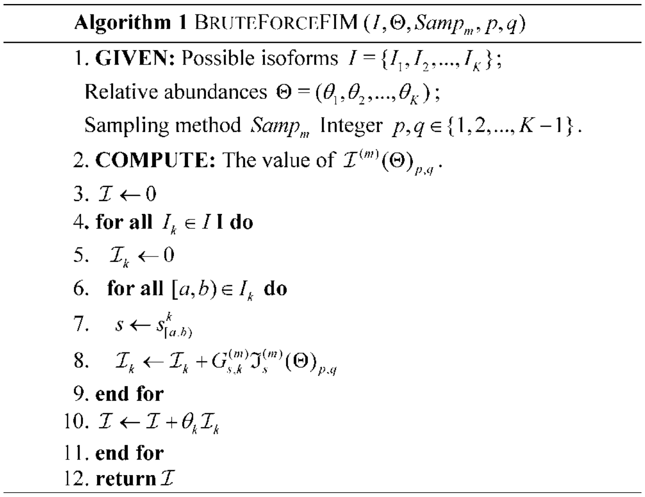
$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \sum_{k=1}^{K}\theta_k \sum_{s=s_{[a,b)}^{(k)};\forall[a,b)\in I_k} G_{s,k}^{(m)}\Im_{s=s_{[a,b)}^{(k)}}^{(m)}(\Theta)_{p,q} \tag{18}$$

where

$$\Im_{s=s_{[a,b)}^{(k)}}^{(m)}(\Theta)_{p,q} = \frac{\left(\delta_{s,p}G_{s,p}^{(m)} - \delta_{s,K}G_{s,K}^{(m)}\right)\left(\delta_{s,q}G_{s,q}^{(m)} - \delta_{s,K}G_{s,K}^{(m)}\right)}{\left[\sum_{k'=1}^{K}\delta_{s,k'}\theta_{k'}G_{s,k'}^{(m)}\right]^2} \tag{19}$$

is the Fisher information matrix of a partial sample s from $Samp_m$ at $[a,b)$ in I_k.

A brute-force algorithm for computing $\Im_{s=s_{[a,b)}^{(k)}}^{(m)}(\Theta)$ can thus be described as follows:



Algorithm 1 BRUTEFORCEFIM $(I,\Theta,Samp_m,p,q)$

```
GIVEN: Possible isoforms $I=\{I_1,I_2,...,I_K\}$;
 Relative abundances $\Theta=(\theta_1,\theta_2,...,\theta_K)$;
 Sampling method $Samp_m$ Integer $p,q\in\{1,2,...,K-1\}$.
COMPUTE: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.
$\mathcal{I}\leftarrow 0$
for all $I_k\in I$ l do
  $\mathcal{I}_k\leftarrow 0$
  for all $[a,b)\in I_k$ do
    $s\leftarrow s_{[a.b)}^{k}$
    $\mathcal{I}_k\leftarrow\mathcal{I}_k+G_{s,k}^{(m)}\Im_s^{(m)}(\Theta)_{p,q}$
  end for
  $\mathcal{I}\leftarrow\mathcal{I}+\theta_k\mathcal{I}_k$
end for
return $\mathcal{I}$
```

In Algorithm 4, if length is the length of a given sequence I_k, then the whole algorithm consists of $\sim\sum_{k=1}^{K}$ length(I_k) computations of $\Im_s^{(m)}(\Theta)$.

Equivalent partial samples

In order to continue our discussion on faster algorithms to compute $\Im_{s=s_{[a,b)}^{(k)}}^{(m)}(\Theta)$, we introduce the concept of equivalent partial samples below (the relavant proofs can be found in Text S1):

Definition 6. Two partial samples s_1 and s_2 are equivalent w.r.t. $Samp_m$ if and only if $\Im_{s_1}^{(m)}(\Theta) = \Im_{s_2}^{(m)}(\Theta)$.

Lemma 1. *If* $\forall I_k\in I$, $\delta_{s_1,k}G_{s_1,k}^{(m)} = \delta_{s_2,k}G_{s_2,k}^{(m)}$, *then* s_1 *and* s_2 *are equivalent w.r.t.* $Samp_m$.

Definition 7. A set of partial samples S is an equivalent sample set w.r.t. $Samp_m$ if and only if $\forall s_1,s_2\in S$, s_1 and s_2 are equivalent w.r.t. $Samp_m$.

Lemma 2. *Given an isoform* I_k *and a sampling method* $Samp_m$, *if we divide all its possible partial samples into* n *non-overlapping equivalent sample sets* $S_1,S_2,...,S_n$, *then:*

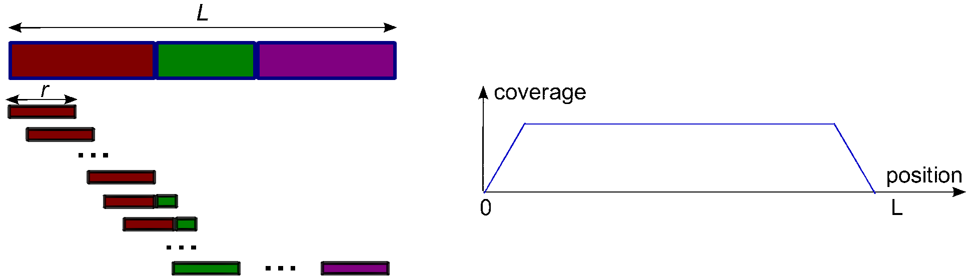


Figure 2. A simple shotgun read generation model.
doi:10.1371/journal.pone.0029175.g002

$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \sum_{k=1}^{K} \theta_k \sum_{i=1}^{n} |S_i| G_{s_i,k}^{(m)} \Im_{s_i}^{(m)}(\Theta)_{p,q}, \textit{for any } s_i \in S_i \tag{20}$$

Results from a simple shotgun read generation model

In this subsection, we consider a simplified partial sample generation model:

Definition 8. A simple shotgun sampling method $Samp_m$ generates samples with fixed read length r_m. When sampling from an isoform I_k with length l_k, there are in total $l_k - r_m + 1$ different samples $s_{[a,b)}^{(k)}$, where $a = 0,1,2,\ldots,(l_k - r_m)$; and $b = a + r_m$. Each of these samples has equal probability of being generated from I_k: $G_{s,k}^{(m)} = 1/(l_k - r_m + 1)$.

Figure 2 illustrates simple shotgun sampling process and its corresponding per-base coverage on the isoform being sampled.

Lemma 3. *Given the sample generation model $Samp_m$ above, if two samples s_1 and s_2 generated by this method are compatible with the same set of isoforms, i.e. $\delta_{s_1,k} = \delta_{s_2,k}, \forall I_k \in I$, then s_1 and s_2 are equivalent w.r.t. $Samp_m$.*

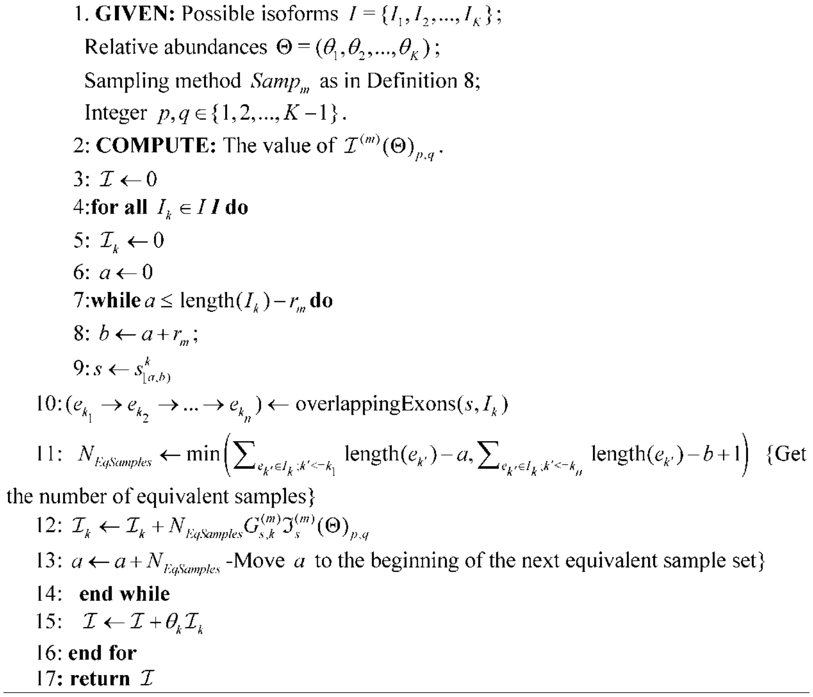
Theorem 1. *Given the sample generation model $Samp_m$ above, if two samples s_1 and s_2 generated by this method overlap with all the junctions in the same set of connected exons $e_{k_1} \rightarrow e_{k_2} \rightarrow \ldots \rightarrow e_{k_n}$, then s_1 and s_2 are equivalent w.r.t. $Samp_m$.*

For example, in Figure 3, where the reads are generated from a simple shotgun sampling process, the equivalent partial samples are -read1, read2, read9}, -read10, read11}. Also, if we consider a paired-end read as a long shotgun read with its gap filled, the samples read5 and read6 are also (approximately) equivalent, if their insert sizes are close to each other. However, read12 is not equivalent to these reads, since its shotgun version overlaps with a different exon junction set (with an addition exon).

Algorithms for efficiently computing $\mathcal{I}^{(m)}(\Theta)$

Based on Definition 8 and Theorem 1, we can design the following algorithm for efficiently computing $\mathcal{I}^{(m)}(\Theta)$ by combining the computation of this value for equivalent partial samples from each isoform.

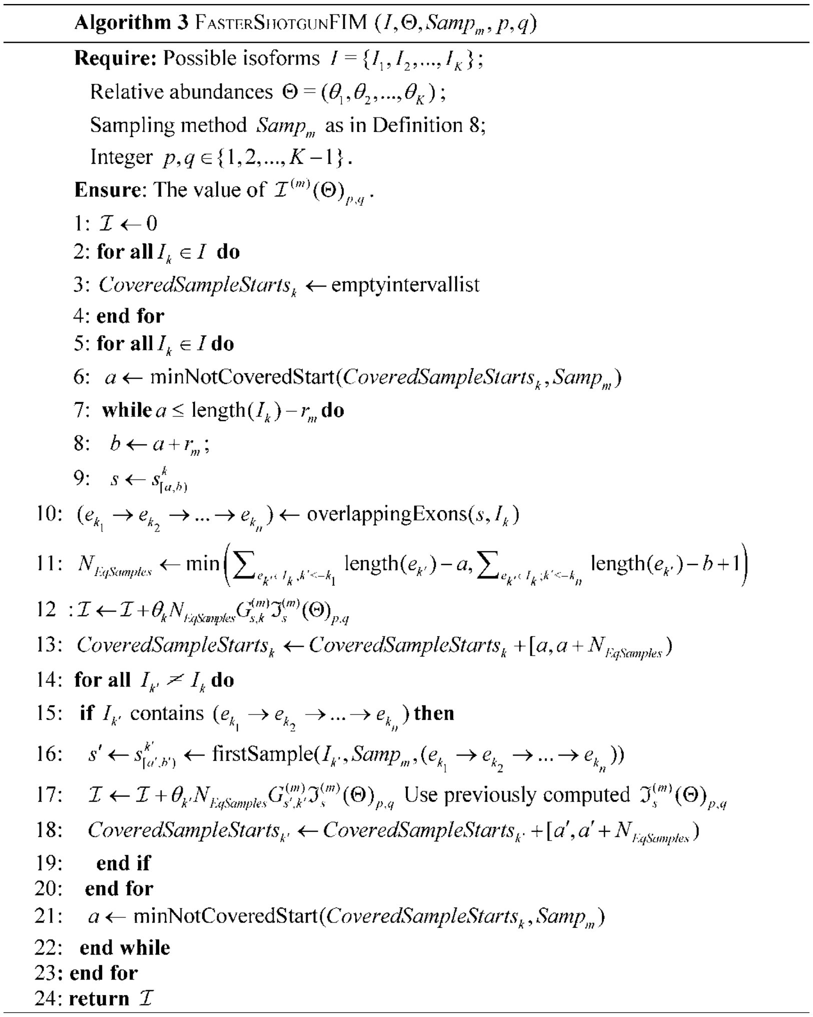
Algorithm 2 FasterShotgunFIM $(I, \Theta, Samp_m, p, q)$



```
GIVEN: Possible isoforms $I = \{I_1, I_2, \ldots, I_K\}$;
 Relative abundances $\Theta = (\theta_1, \theta_2, \ldots, \theta_K)$;
 Sampling method $Samp_m$ as in Definition 8;
 Integer $p, q \in \{1, 2, \ldots, K-1\}$.
COMPUTE: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.
$\mathcal{I} \leftarrow 0$
for all $I_k \in I$ I do
$\mathcal{I}_k \leftarrow 0$
$a \leftarrow 0$
while $a \le \text{length}(I_k) - r_m$ do
$b \leftarrow a + r_m$;
$s \leftarrow s_{[a,b)}^{k}$
    $(e_{k_1} \rightarrow e_{k_2} \rightarrow \ldots \rightarrow e_{k_n}) \leftarrow \text{overlappingExons}(s, I_k)$
    $N_{EqSamples} \leftarrow \min\left(\sum_{e_{k'} \in I_k ; k' <- k_1} \text{length}(e_{k'}) - a, \sum_{e_{k'} \in I_k ; k' <- k_n} \text{length}(e_{k'}) - b + 1\right)$ {Get
the number of equivalent samples}
    $\mathcal{I}_k \leftarrow \mathcal{I}_k + N_{EqSamples} G_{s,k}^{(m)} \Im_s^{(m)}(\Theta)_{p,q}$
    $a \leftarrow a + N_{EqSamples}$ -Move $a$ to the beginning of the next equivalent sample set}
     end while
     $\mathcal{I} \leftarrow \mathcal{I} + \theta_k \mathcal{I}_k$
    end for
    return $\mathcal{I}$
```

In Algorithm 2, overlappingExons(s,I_k) identifies the connected exons set in I_k that overlaps with a given partial sample s, and can be implemented with $O(\log NumExons_k)$ time complexity by pre-computing an exon-position index table for the isoforms.

We can further reduce the number of times of computing $\Im_s^{(m)}(\Theta)$ by combining equivalent partial samples from across isoforms: when an equivalent sample set from an isoform has been identified, all the same samples from other isoforms can be recorded in lists of intervals to avoid redundant computation of their $\Im_s^{(m)}(\Theta)$s. The algorithm is shown below:



```
Algorithm 3 FasterShotgunFIM ($I,\Theta,Samp_m,p,q$)
Require: Possible isoforms $I=\{I_1,I_2,...,I_K\}$;
  Relative abundances $\Theta=(\theta_1,\theta_2,...,\theta_K)$;
  Sampling method $Samp_m$ as in Definition 8;
  Integer $p,q\in\{1,2,...,K-1\}$.
Ensure: The value of $\mathcal{I}^{(m)}(\Theta)_{p,q}$.
$\mathcal{I}\leftarrow 0$
for all $I_k \in I$ do
  $CoveredSampleStarts_k \leftarrow$ emptyintervallist
end for
for all $I_k \in I$ do
  $a \leftarrow$ minNotCoveredStart($CoveredSampleStarts_k, Samp_m$)
  while $a \le$ length($I_k$) $- r_m$ do
    $b \leftarrow a + r_m$;
    $s \leftarrow s^k_{[a,b)}$
    $(e_{k_1}\rightarrow e_{k_2}\rightarrow ...\rightarrow e_{k_n}) \leftarrow$ overlappingExons($s, I_k$)
    $N_{EqSamples} \leftarrow \min\left(\sum_{e_{k'}\in I_k, k'<-k_1}\text{length}(e_{k'})-a, \sum_{e_{k'}\in I_k; k'<-k_n}\text{length}(e_{k'})-b+1\right)$
    $\mathcal{I}\leftarrow\mathcal{I}+\theta_k N_{EqSamples}G^{(m)}_{s,k}\Im^{(m)}_s(\Theta)_{p,q}$
    $CoveredSampleStarts_k \leftarrow CoveredSampleStarts_k + [a, a+N_{EqSamples})$
    for all $I_{k'} \neq I_k$ do
      if $I_{k'}$ contains $(e_{k_1}\rightarrow e_{k_2}\rightarrow ...\rightarrow e_{k_n})$ then
        $s' \leftarrow s^{k'}_{[a',b')} \leftarrow$ firstSample($I_{k'}, Samp_m, (e_{k_1}\rightarrow e_{k_2}\rightarrow ...\rightarrow e_{k_n})$)
        $\mathcal{I}\leftarrow\mathcal{I}+\theta_{k'} N_{EqSamples}G^{(m)}_{s',k'}\Im^{(m)}_s(\Theta)_{p,q}$ Use previously computed $\Im^{(m)}_s(\Theta)_{p,q}$
        $CoveredSampleStarts_{k'} \leftarrow CoveredSampleStarts_{k'} + [a', a'+N_{EqSamples})$
      end if
    end for
    $a \leftarrow$ minNotCoveredStart($CoveredSampleStarts_k, Samp_m$)
  end while
end for
return $\mathcal{I}$
```

In Algorithm 3, minNotCoveredStart($CoveredSampleStarts_k$, $Samp_m$) finds the minimum position $a\in\{0,1,...,\text{length}(I_k)-r_m+1\}$ that is outside a given interval list $CoveredSampleStarts_k$; firstSample(I_k,$Samp_m$,$ConnectedExonSet$) returns the partial sample $s^k_{[a,b)}$ from I_k covering all the exon junctions in $ConnectedExonSet$ with a minimum a, and can be implemented with a worst-case $O(\log NumExons_k + |ConnectedExonSet|)$ time complexity by using a pre-computed exon position index table for the isoforms.

Complexity analysis

Given a set of K possible isoforms $I=\{I_1,I_2,...,I_K\}$, with lengths $l_1,l_2,...,l_K$, respectively, and a shotgun sampling method $Samp_m$ with sample read length r_m as described in Definition 8, Algorithm 1 requires $K(\bar{l_k})$ steps of computing $\Im_s^{(m)}(\Theta)_{p,q}$. Thus computing $\mathcal{I}^{(m)}(\Theta)$ using this brute-force algorithm requires $(K-1)^2\cdot\sum_{k=1}^{K} l_k$ operations of calculating $\Im_s^{(m)}(\Theta)_{p,q}$. If we assume that the average length of an isoform is $l_{AvgIsoform}$, this corresponds to $\sim K^3\cdot l_{AvgIsoform}$ computations of $\Im_s^{(m)}(\Theta)_{p,q}$.

Suppose that on average an isoform can be divided into $N_{EqSampleSets}$ equivalent sample sets by Algorithm 2, this algorithm will then require $\sim K^3\cdot N_{EqSampleSets}$ steps of computing $\Im_s^{(m)}(\Theta)_{p,q}$ to obtain the Fisher information matrix $\mathcal{I}^{(m)}(\Theta)$ for the given sampling method, thus being more efficient than Algorithm 1 by a ratio of $l_{AvgIsoform}/N_{EqSampleSets}$. Algorithm 3 will obviously be even more efficient by avoiding the redundant computation of some of the equivalent sample sets in Algorithm 2.

Using more complex G function in Algorithm 3

The sequencing technology being used in an RNA-Seq experiment is usually more complicated than the simplified G function described in Definition 8, which assumes equal sample-length and uniform generative probability. In reality, a typical G usually involves reads with different lengths within a certain range, and also biased sample generation probability at different locations

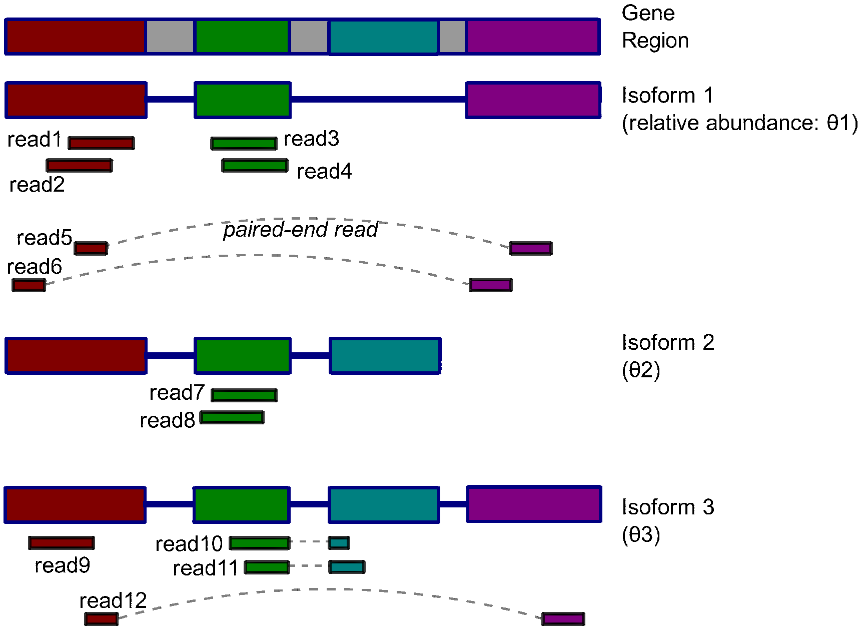


Figure 3. Equivalent samples in a simple shotgun read generation model.
doi:10.1371/journal.pone.0029175.g003

of a full-length isoform. Although once such a G is defined, our MLE solution can treat it in the same way as it does for simplified versions, Algorithm 3 no longer works "out of the box" due to its dependency on Definition 8 to find equivalent partial samples. We discuss briefly in this subsection on how to handle more complex features.

When the assumption of uniform sample generation still holds, it is straightforward to handle samples with different lengths in FIM computation. We can treat one sampling method as a combination of multiple simplified methods as in Definition 8, with different sample lengths $\{l_1, \cdots, l_L\}$:

$$\mathcal{I}^{(m)}(\Theta)_{p,q} = \sum_{l=l_1}^{l_L} Pr_m\{length(s)=l\}\mathcal{I}^{(m_l)}(\Theta)_{p,q} \tag{21}$$

$$= \sum_{l=l_1}^{l_L} Pr_m\{length(s)=l\}\left[-\frac{\partial^2 \log \sum_{k=1}^{K} \delta_{s_l,k}\theta_k G_{s_l,k}^{(m,l)}}{\partial\theta_p \partial\theta_q}\right] \tag{22}$$

where $Pr_m\{length(s)=l\}$ represents the probability of generating a sample with length l in sampling method $Samp_m$, s_l is a sample with length l, and $G_{s_l,k}^{(m,l)}$ is the simplified sample generation probability as in Definition 8, with sample length l.

In the case of non-uniform sample generation along the isoform, if $G_{s,k}^{(m)}$ is a step function (piece-wise constant function) for sample s along isoform I_k, we will still be able to find equivalent sample sets as described in Definition 7, based on both the isoform structures and the intervals in G. If, however, very few such constant components exist in G, we will need to relax the definition of equivalent partial samples to satisfying $\delta_{s_1,k} = \delta_{s_2,k}$ only. With this relaxed definition, we can find samples S_{eq} with equivalent structural similarities to all the isoforms. In this case, if the isoforms contain regions where any s_1 and s_2 from it satisfy $G_{s_1,k}^{(m)} = c_{s_1,s_2} \cdot G_{s_2,k}^{(m)}$ with a constant c_{s_1,s_2} for all k, we still have $\Im_{s_1}^{(m)}(\Theta)_{p,q} = \Im_{s_2}^{(m)}(\Theta)_{p,q}$ according to Equation 19, and the $\mathcal{I}^{(m)}$ can thus be efficiently computed using a variant of Algorithm 3 by combining the computation for such equivalent partial samples. For more complex G functions, however, approximation algorithms may have to be introduced for fast computation of $\mathcal{I}^{(m)}$.

Results

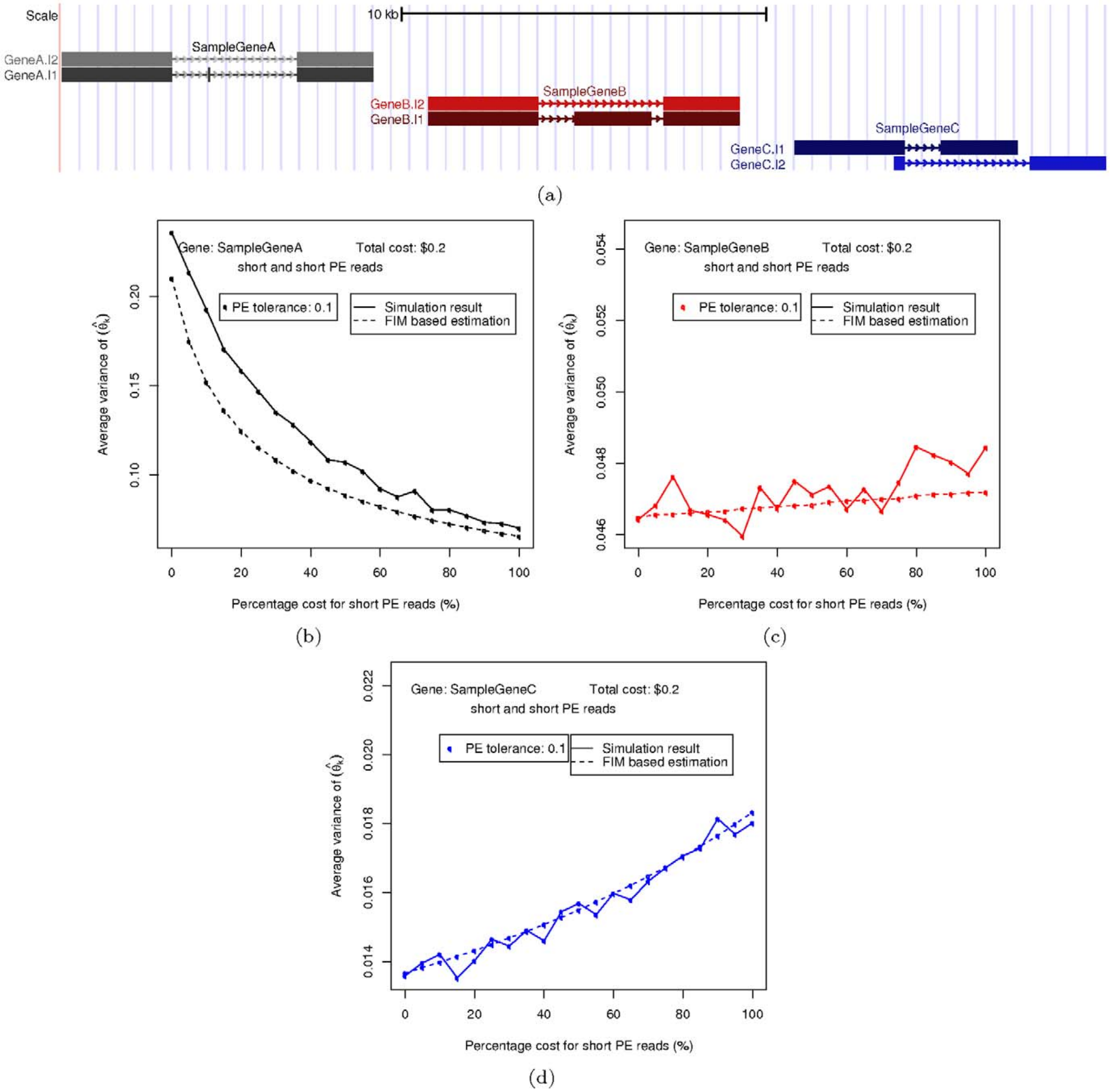
Simulation Results

Here we present our results on a set of simulated datasets. In order to demonstrate the accuracy and efficiency of the methods we developed, we first use simulations to show the performance of our approach on simplified gene models and a real gene. These simulations are useful in designing optimal sequencing experiments for isoform quantification.

Simulation on simplified genes

Due to the complexity of real gene structure, we apply our methods to three artificially constructed genes with simplified isoform structures, so as to better illustrate how different characteristics of the gene structures can affect the outcome of the isoform quantification analysis.

As shown in Figure 4A, each of these genes has two different isoforms, with the more abundant one shown in a darker color. Two sampling techniques, short single and short paired-end (PE), are used to generate reads from them, with a fixed total cost of \$0.20 (roughly corresponding to 12 medium length reads with average size of 250 bp ($\sim 0.6\times$ coverage on the simplied genes), or 950 short reads with average length of 30 bp (~ 6x)). The per-base costs of these sampling techniques are based on [14]. Different cost combinations, e.g. different percentage of the total cost being assigned to a certain sampling technique, are illustrated by the x-axis in Figure 4B–D. For each of these cost combinations, we randomly generate 1000 read sets, and use the MLE solution to estimate $\hat{\Theta}$, based on which $Average\left(var(\hat{\theta}_k)\right)$ are computed (solid lines in Figure 4B–D). We also use Algorithm 3 to estimate the same quantity, and plot the estimations using dashed lines in the same figure for comparison. The results show that the FIM



(a)

(b)

(c)

(d)

Figure 4. Results on simplified genes. (a) Results on gene A (b) Results on gene B (c) Results on gene C.
doi:10.1371/journal.pone.0029175.g004

estimation of $Average\left(var(\hat{\theta}_k)\right)$ are close to the direct simulation results, and also correctly predicts the trend in how this value changes with different cost combinations. Also, different gene structures have noticeable impact on the MLE accuracy, mostly due to the ability of sampling techniques to distinguish isoforms from each other with different gene structures.

Table 1. Total time used by brute-force simulation vs. FIM based heuristic to estimate $Average\left(var(\hat{\theta}_k)\right)$ in simplified genes.

Total trials for one gene	Number of trials $\times$ Number of sampling method combinations $= 1000 \times 21$
Total FIM computation for one gene	Number of sampling methods $= 2$
Total CPU time used by brute-force simulation	~ 52 minutes
Total CPU time used by FIM based heuristic	< 1 second

doi:10.1371/journal.pone.0029175.t001

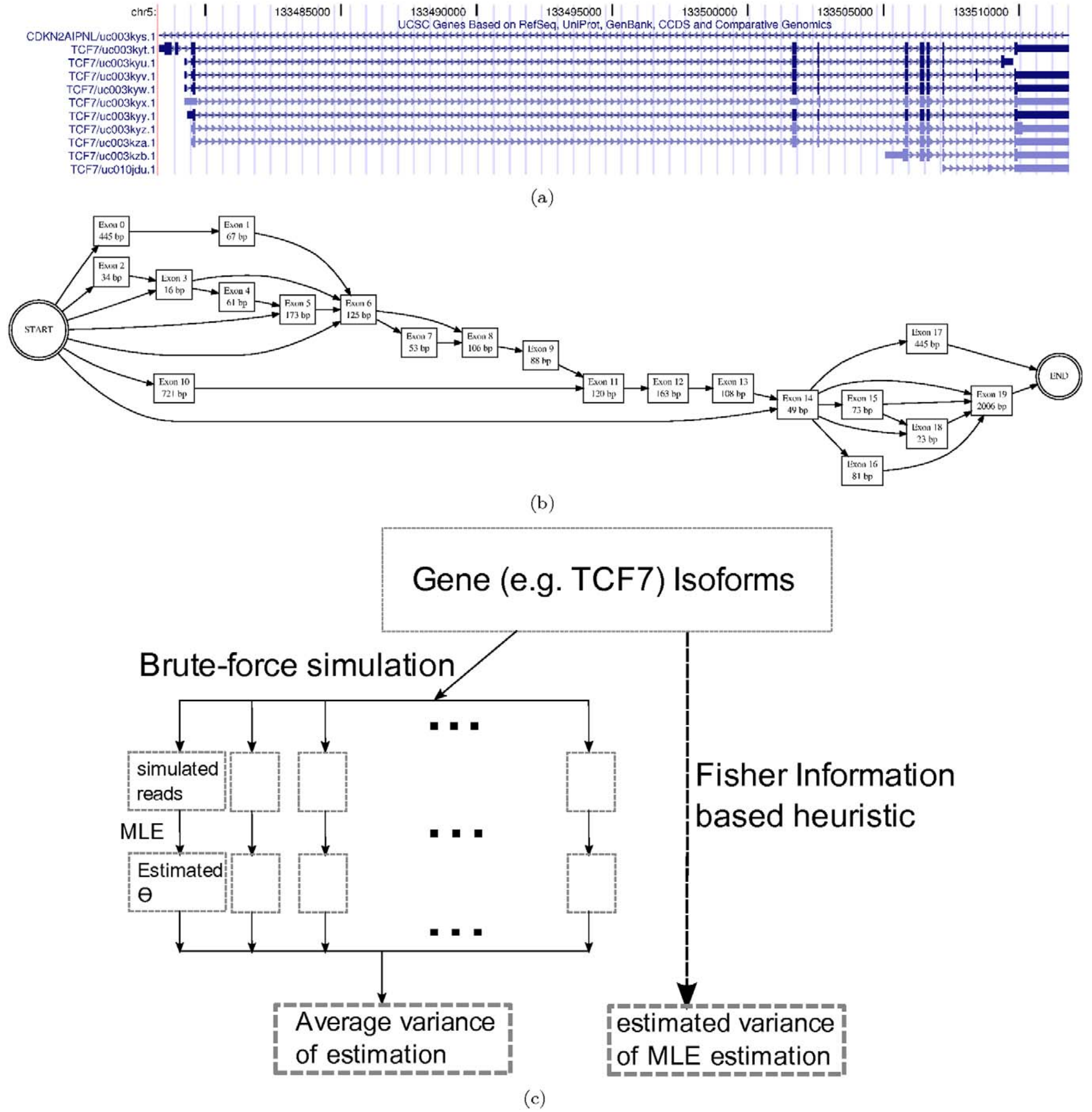


Figure 5. Computations on gene TCF7. (a) Known isoforms (b) Splicing graph (c) Simulation schema.
doi:10.1371/journal.pone.0029175.g005

Not only can the FIM based heuristic correctly approximate how the performance of MLE changes with regard to different sampling technique combinations, it is also able to dramatically shorten the time of computation, as shown in Table 1. This is mainly because while the computation of brute-force simulation depends heavily on the number of reads being generated and the number of trials needed to obtain a relatively stable estimation of $Average\left(var(\hat{\theta}_k)\right)$, the core computation taken by the FIM based heuristic is the evaluation of individual FIMs for the sampling techniques involved, which can be efficiently computed using Algorithm 3, and then combined based on Equation 16 to estimate $Average\left(var(\hat{\theta}_k)\right)$ under different cost combinations. Being able to do these simulations fast is useful in designing optimal expriments.

Efficiently Estimating quantification error: Application on a typical gene

We have developed a Fisher information matrix (FIM) based fast algorithm (Algorithm 3 in Methods section) for estimating the quantification error in $\hat{\Theta}$, and compared its speed with two other benchmark algorithms. Here we consider the gene $TCF7$, which has 10 known isoforms shown in Figure 5A. Figure 5B shows its corresponding splicing graph [6,15], with 19 exon blocks, and 96

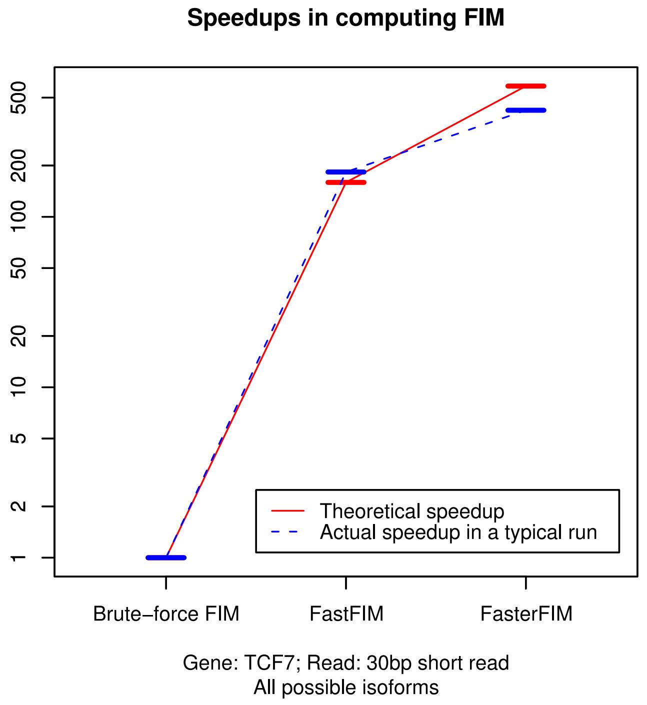


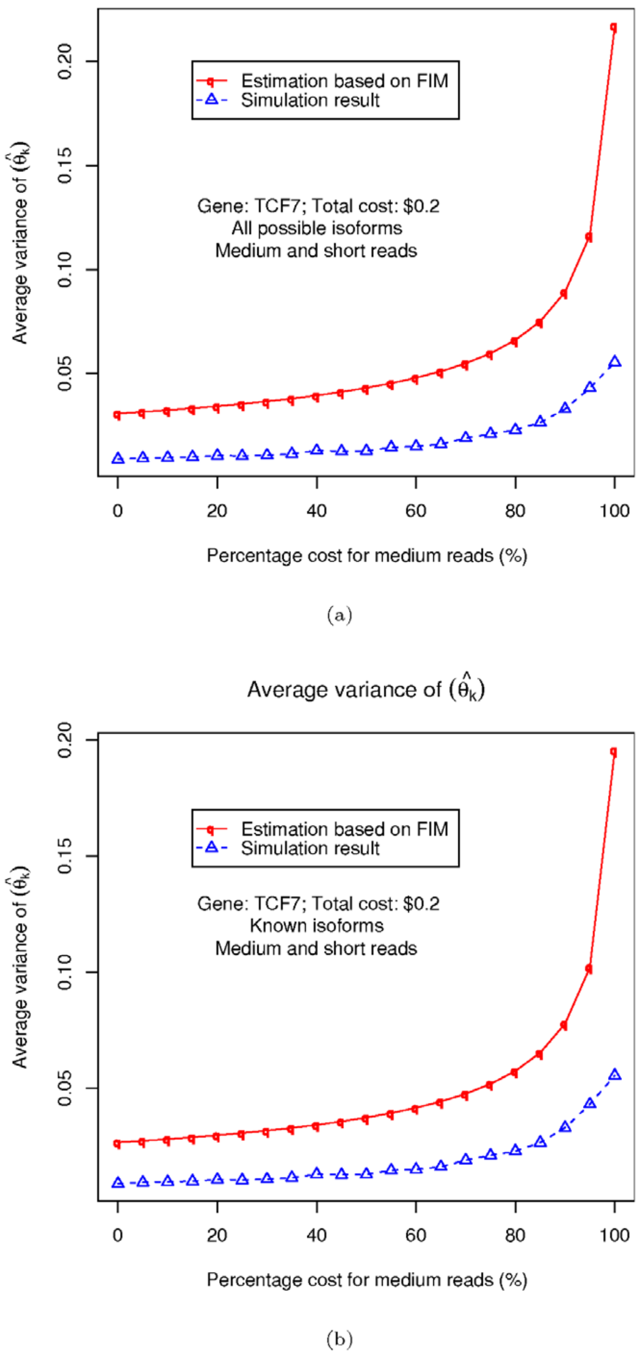
Figure 6. Speedup in FIM computation for gene TCF7.
doi:10.1371/journal.pone.0029175.g006

possible isoforms, which are all the possible paths from node "START" to node "END" in the splicing graph. Figure 5C shows the brute-force way and Fisher information based method to estimate $Average\left(var(\hat{\theta}_k)\right)$.

When computing the expected Fisher information matrix $\mathcal{I}_s^{(m)}(\Theta)$, a brute-force algorithm (Algorithm 1) requires 26902 computations of the observed Fisher information matrix $\Im_s^{(m)}(\Theta)$, while an improved algorithm developed by us (Algorithm 2) involves 169 such computations, and the number for our final algorithm (Algorithm 3) is 46, achieving a ~ 585 times speedup compared to the brute-force method. A summary of the speedups is shown in Figure 6. Note that theoretical speedup is calculated based on the number of key computational steps (per-read FIM computation), while the actual speedup depends on the software implementation of all steps in the algorithm.

Integrated analysis with multiple sequencing technologies: Simulation on a typical gene

We present in this section the application of the FIM based heuristic on a real gene, and compared the results to the ones obtained from direct simulations. We pick TCF7 again as a typical example gene with multiple isoforms. Similarly to our procedure in the section on simplified genes, two sampling techniques, medium and short shotgun sequencing (with average length of 250 bp and 30 bp, \$70 and \$7 per 1 million base cost respectively), are used to generate reads from them, with a fixed total cost of \$0.2, with 200 trials being conducted for each cost combination. Two different sets of results are shown in Figure 7, one using all the 96 possible isoforms deduced from its splicing graph, and the other just using its 10 known isoforms. As in the previous section, the results here show that the FIM estimation of $Average\left(var(\hat{\theta}_k)\right)$ are close to the direct simulation results, and also correctly predicts the trend in how this value changes with different cost combinations.



(a)

(b)

Figure 7. Simulation results on TCF7. (a) Results on all possible isoforms (b) Results on known isoforms.
doi:10.1371/journal.pone.0029175.g007

Figure 8 presents a more detailed simulation focusing on short paired-end reads. The *tolerance* value reflects the expectation of the variance in insert size for such experiments: a 0 value means that all the paired-end reads are expected to have exactly the same insert size; the higher the *tolerance* is, the more relaxed are we on the insert size variation (i.e. if the distance of the mapped positions of the two ends of a read on a transcript is within the expected insert size $+/-$ a "tolerated" ratio, the paired-end read will be considered "compatible" with the transcript). As we can see from this figure, the higher the *tolerance*, the larger $Average\left(var(\hat{\theta}_k)\right)$

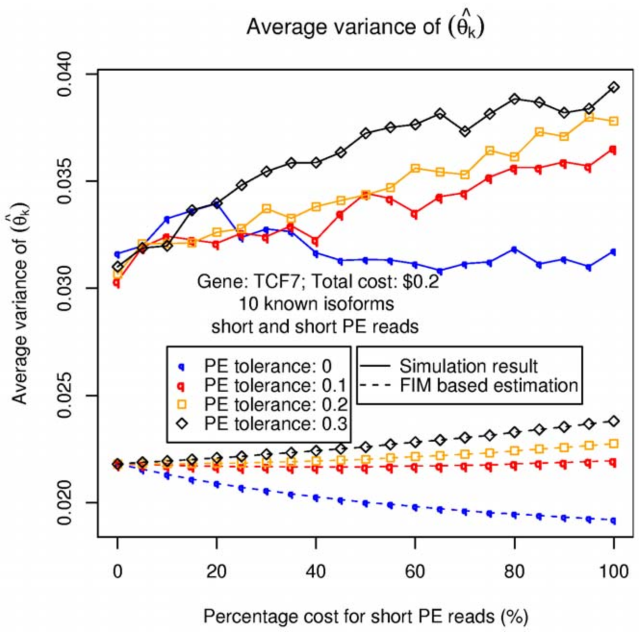


Figure 8. Simulation results on TCF7 with paired-end reads.
doi:10.1371/journal.pone.0029175.g008

becomes, corresponding to a worse expected performance of MLE. This can be explained by the fact that a higher *tolerance* makes the sampling method less capable of distinguishing highly similar isoforms from each other based on a single paired-end read (e.g. GeneA in Figure 4A). The FIM based heuristic is again able to correctly depict the different trends of MLE performance under different cost combinations and *tolerance* settings.

We also show the computation time used by brute-force simulation and FIM based heuristic in Table 2. Note that the brute-force simulation is even more computational consuming, mainly because more isoforms are involved in the MLE process. Given the fact that there exist more than 20000 genes in the human genome and that the simulation has to be rerun for every new experiment to adjust its read counts (the number of reads attributed to a gene region in the experiment), using the FIM based heuristic instead for the purpose of estimating isoform quantification accuracy is obviously a more computationally tractable choice.

Application to a model-organism (worm) dataset

To illustrate how we can interpret the Θ values output, we further apply our MLE solution to a worm dataset [16–18], which is a well-studied model organism. which is a well-studied model organism. The worm has intermediate complexity in isoform structures. It has isoforms but they are significantly simpler in structure than in human, leading to interpretability in the results. This dataset includes multiple developmental stages, and we were able to compare the results on a same set of isoforms under different conditions. The worm genome contains $\sim$20 K genes, and the transcripts from each stage are sequenced with $\sim$50 M short Solexa reads. This dataset is particularly useful for isoform comparison since it contains multiple stages of splicing events that are not overly complex.

Dataset description

Whole transcriptome sequencing data for worm L2, L3, L4 and Young Adult stages, each stage with on average 50 M reads. The annotation set (derived from the modENCODE project, [17,18]) has 21774 total genes. Of these, 12875 genes has multiple isoforms, with an average of 4.344 isoforms per gene.

Comparison of isoform composition between stages

We first present the isoform quantification results on individual genes in two different stages, early embryo (EE) and late embryo (LE), to briefly illustrate the fact that different genes have different isoform composition differences between stages. Here we use the following formula to measure the difference in isoform composition of the same gene in two different stages:

$$Diff_{gene_i}(\Theta^{(Stage1)},\Theta^{(Stage2)}) = \frac{\sum_{k=1}^{K}(\theta_k^{(Stage1)} - \theta_k^{(Stage2)})^2}{K} \qquad (23)$$

where K is the total number of isoforms in gene $gene_i$.

Figure 9 shows two examples of zero and non-zero *Diff* values. The reads are plotted below the isoforms, and the numbers associated with the isoforms are their estimated relative abundances based on MLE. Furthermore, if we compute such values for all the genes in these two stages, we can get a histogram of isoform composition differences as illustrated in Figure 10, which characterizes the general isoform composition difference between stages. The distribution of differences in relative isoform composition for genes is shown: Isoform quantification was applied to RNA-Seq data in 4 developmental stages in worm (L2, L3, L4, YA) and *Diff* score was calculated for each gene in all 6 pairwise comparisons. Because isoform quantification is noisy for genes expressed at very low level, we plotted the distribution for genes that have at least an RPKM value of 0.5 here (RPKM for a gene is the sum of RPKMs of all its isoforms). Red bars represent the average number of genes within the respective *Diff* score range, while error bars indicate the maximum and minimum numbers. *Diff* scores close to 1 indicate big changes in isoform composition, or the relative expression level of isoforms between stages. The histogram indicates that only a few genes ($\approx$43) show dramatic differences in isoform expression between stages. (The number 43 is derived from a cutoff of 0.5 on the *Diff* score.) In Table 3, we include a classification of the structural difference (5′ UTR, 3′ UTR, alternative exon, etc.), between the dominant transcripts in such genes with different isoform compositions. When the different dominant isoforms from a gene differ in two aspects, we assign 0.5 to each category. As shown in this table, many of the structural differences are due to either Distinct 5′ UTR or Overlapping 3′

Table 2. Total time used by brute-force simulation vs. FIM based heuristic to estimate $Average\left(var(\hat{\theta}_k)\right)$ in TCF7.

Total trials for one gene	Number of trials $\times$ Number of sampling method combinations $= 200 \times 21$
Total FIM computation for one gene	Number of sampling methods $= 2$
Total CPU time used by brute-force simulation	~ 10.6 hours
Total CPU time used by FIM based heuristic	< 1 second

doi:10.1371/journal.pone.0029175.t002

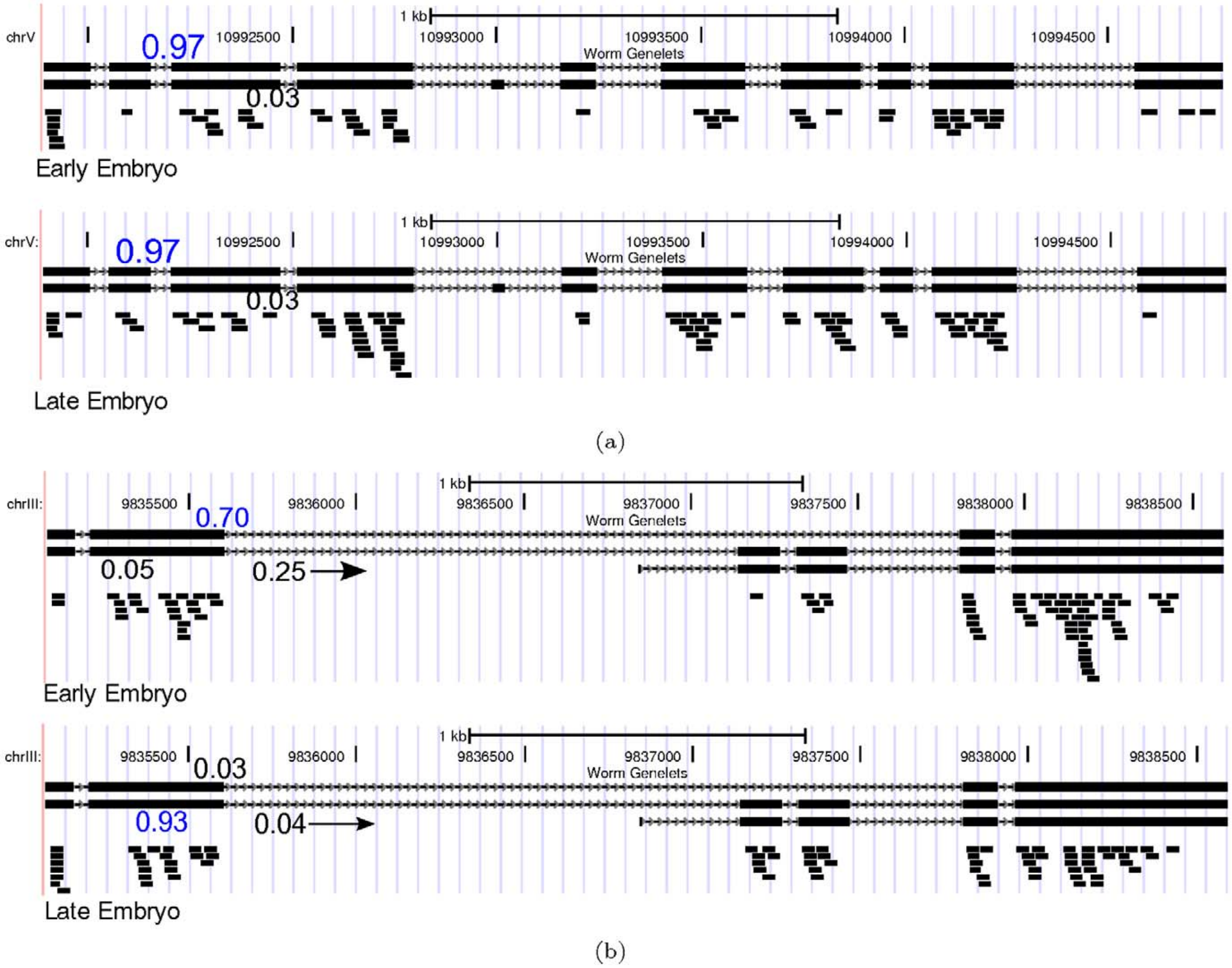


Figure 9. Isoform composition in two stages. (a) Gene 14047 in two stages (*Diff* = 0) (b) Gene 7649 in two stages (*Diff* = 0.42).
doi:10.1371/journal.pone.0029175.g009

UTR. We have also included in the supplementary website (http://archive.gersteinlab.org/proj/rnaseq/IQSeq) the genes with stage-wise isoform composition differences, ranked by their FIM based estimation variances, and with a thresholded on *Diff* score and RPKM at 0.5.

The effect of different isoform sets on MLE result

We also investigate how different isoform sets (e.g. with a major/minor isoform missing, with an additional "dummy" isoform) will affect the MLE result, especially in terms of the maximized likelihood value. We pick gene No. 7649 as a base isoform set, using the same set of reads and the per-read average maximized likelihood value LL to measure the goodness of fitting:

$$LL_{gene_i} = \sum_{r \in R} \log \sum_{k=1}^{K} \delta_{r,k} \theta_k G_{r,k} \qquad (24)$$

As we can see from Figure 11, the LL value always decreases when we modify the "true" isoform set in an unfavorable fashion. This shows that the likelihood score is an effective metric for ranking isoform sets for a particular gene. We observe that the LL value decreases the most in all cases when the dominant isoform is removed from the isoform set Figure 11b), which indicates that a more important element has a larger contribution in explaining the generated reads. Correspondingly, when a low-probability or dummy isoform (that is not similar to the dominant one) is added to the input isoform set, the LL value decreases less significantly (about half of the case with dominant isoform removal), and also the isoform quantification results remain almost unchanged for the other transcripts in the isoform set. In practice, this characteristic can also be useful to eliminate non-existing isoforms - any isoform that has little effect on either quantification result or LL score can be considered "not important" for explaining the observed reads, and can thus be removed from the isoform set when analyzing a particular dataset.

Use of empirical G function

To illustrate how non-uniform G function works, we modeled the bias of RNA-Seq data by aggregating signal of mapped reads along annotated transcripts. A signal map of the first base of mapped reads was generated. The signal was subsequently mapped onto the transcript and aggregated for all genes with

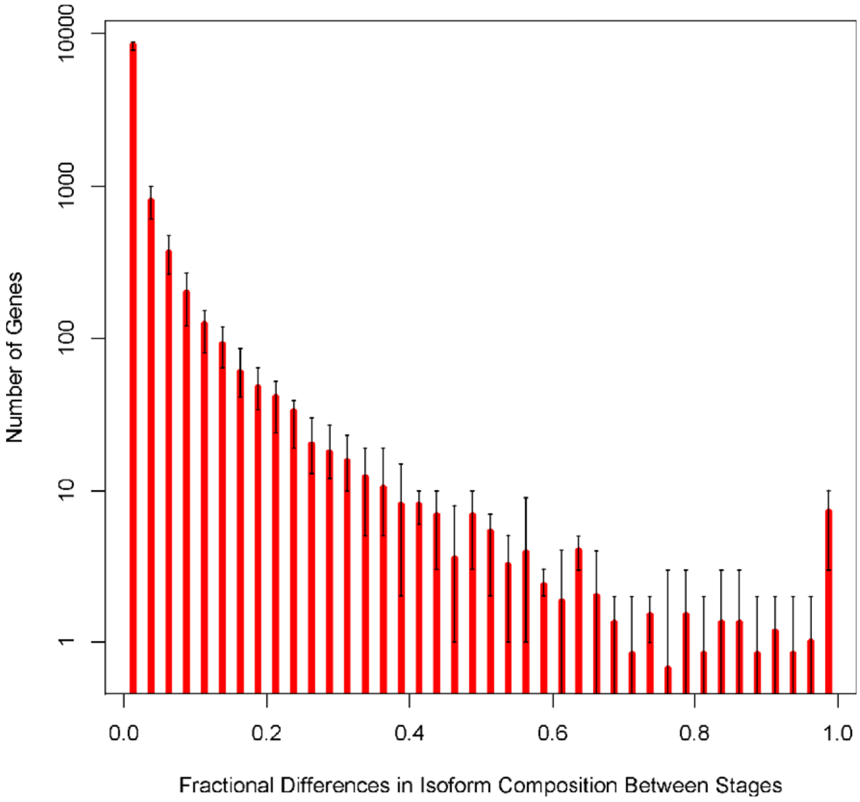


Figure 10. ***Diff*** **of all genes across four stages.**
doi:10.1371/journal.pone.0029175.g010

signal isoform. An aggregation plot of such a signal map for Young Adult is shown in Text S1. In this aggregation, each transcript is divided evenly into 100 bins, with the signals normalized by the sum of signals across all the bins. The normalized signal at each bin thus represents the probability that a read is generated at certain position of the transcript. These non-uniform probabilities gave more realistic estimation of how the reads are generated, and were plugged into the EM calculations. We compared the quantification results for Young Adult worm with uniform and non-uniform G function. The Pearson correlation score for relative abundance is 0.996, and score for absolute abundance is 0.989. The results are similar for other stages. For the majority of genes, the isoform quantification is largely dependent upon whether reads are compatible with the different isoforms of the gene, while the subtle differences in start position probabilities have little influences on final estimation results. Only for a few genes where the isoform structures are highly similar to each other, the quantification results are different.

Also, there have been some recent works [19,20] studying the sequencing biases in RNA-Seq data, with more sophisticated modeling utilizing local sequence composition at different positions along the transcript. Based on different assumptions on sequencing bias, their results can be plugged into the *G* function to derive more realistic quantification results.

Comparison with existing tools

In order to understand the performance of our method compared with other existing tools, we have conducted additional computational analysis by applying IQSeq and Cufflinks on 14 samples from MAQC-3 data [21]. We summarize our result in Figure 12. The genes are categorized by their number of isoforms, and Pearson correlations of the estimated isoform level RPKMs (in logarithmic scale) from the two methods are calculated for each category in each sample. The overall correlation of the isoform quantification results from these two methods is ~0.7 across all samples, which indicates a similar characteristic with a near

Table 3. Classification of different isoform composition between stages.

Type	L3 vs L2	L4 vs L2	L4 vs L3	YA vs L2	YA vs L3	YA vs L4
Overlapping 5′ UTR	6.5	6	2.5	4	4.5	2.5
Distinct 5′ UTR	7.5	26	13.5	20	11	11
Alternative Exon	4.5	6.5	3	4.5	3	4.5
Extended Exon	4	7	6.5	6.5	6	6.5
Overlapping 3′ UTR	13.5	11	8.5	11.5	9.5	10
Distinct 3′ UTR	2	3.5	3	2.5	3	1.5

doi:10.1371/journal.pone.0029175.t003

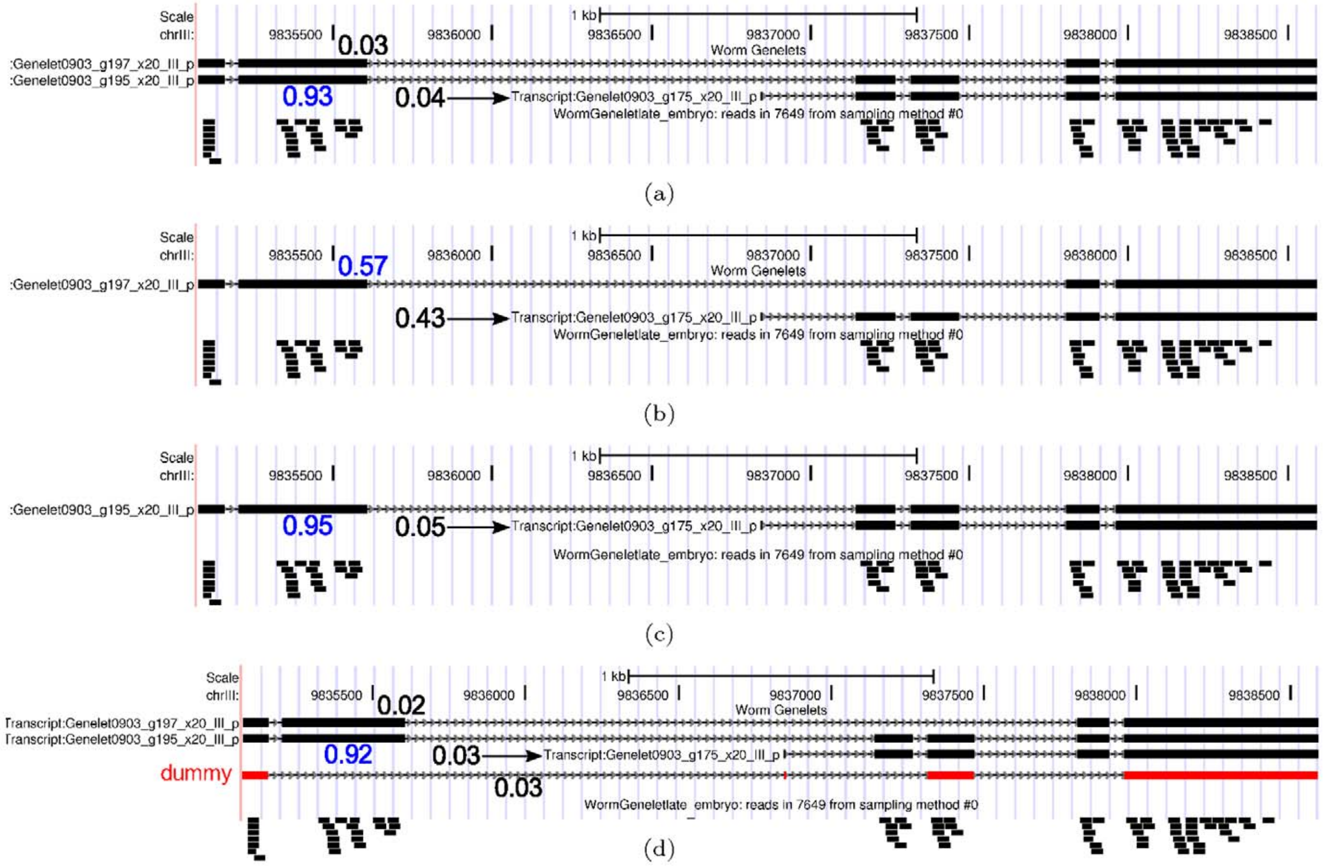

Scale
chrIII:
1 kb
9835500
9836000
9836500
9837000
9837500
9838000
9838500
Worm Genelets
:Genelet0903_g197_x20_III_p
:Genelet0903_g195_x20_III_p
0.03
0.93
0.04
Transcript:Genelet0903_g175_x20_III_p
WormGeneletlate_embryo: reads in 7649 from sampling method #0
(a)
:Genelet0903_g197_x20_III_p
0.57
0.43
(b)
:Genelet0903_g195_x20_III_p
0.95
0.05
(c)
Transcript:Genelet0903_g197_x20_III_p
Transcript:Genelet0903_g195_x20_III_p
dummy
0.02
0.92
0.03
0.03
(d)

Figure 11. Gene 7649: Leave out one isoform, or add a ''dummy'' isoform. (a) Standard calculation with all isoforms: $LL = -7.22$ (b) Leave out the dominant isoform: $LL = -7.35$ (c) Leave out a non-dominant isoform: $LL = -7.29$ (d) Add a ''dummy'' isoform: $LL = -7.29$.
doi:10.1371/journal.pone.0029175.g011

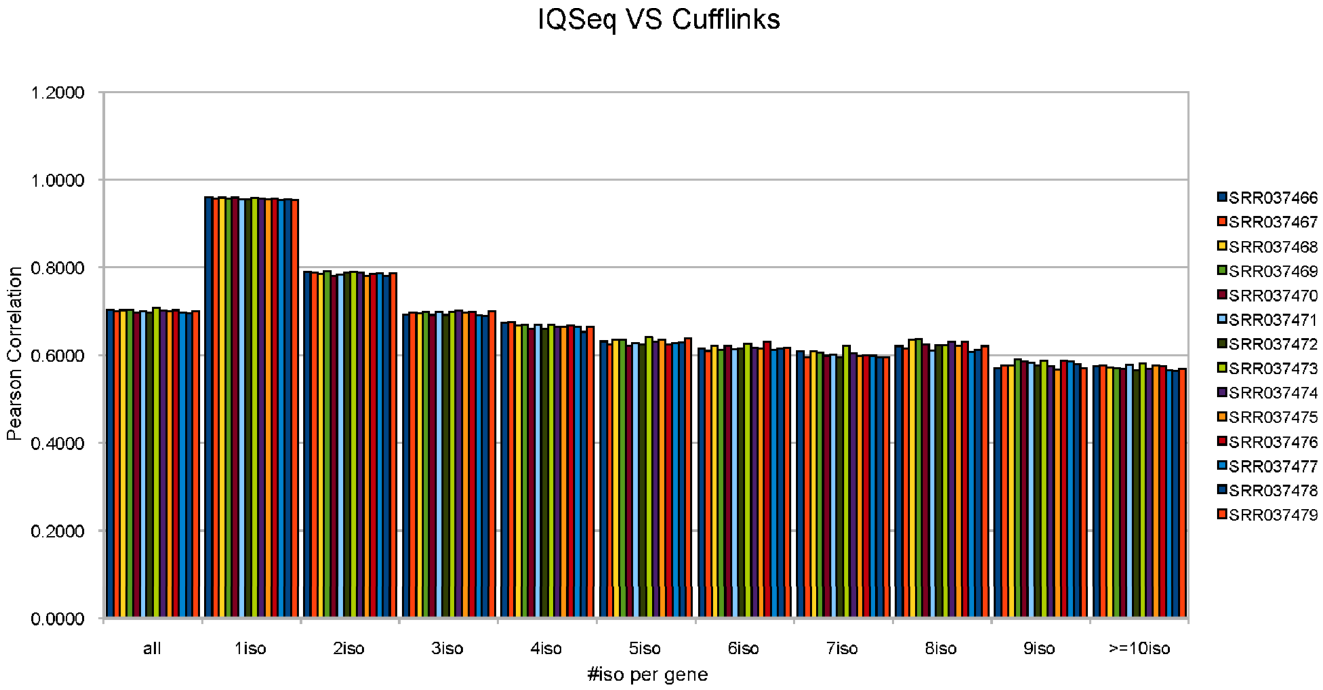


Figure 12. Comparison between IQSeq and Cufflinks.
doi:10.1371/journal.pone.0029175.g012

uniform read-generation assumption. Also, both outputs from IQSeq and Cufflinks have ~ 0.79 correlation with the Taqman assay [22] (an qRT-PCR technique, which can be considered as a gold-standard here). These results confirm consistency of our method with previous work. Note, however, that with our method one can readily "plug-in" more practical read generation models as illustrated in the previous section, making it a more flexible tool to handle and integrate data from different sequencing technologies. Also, we compared the isoform quantification variances between replicates with the FIM based variance estimations, and their logarithmic values have a correlation of 0.59 (Figure 3 in Text S1).

Discussion

In this paper we explore the problem of integrating different sequencing techniques to quantify the relative abundance of different isoform transcripts, which can be generalized to the problem of estimating the distribution based on partial samples from different sampling techniques. We first introduce a statistical framework to model the generative process of the partial samples, using a "plugin-able" function G to allow flexible incorporation of different sampling characteristics, and then present the original problem as a maximum likelihood estimation (MLE) problem, with an iterative solution based on expectation maximization, which guarantees a locally optimum answer. This provides a solution to the question of estimating a distribution based on partial samples.

In order to further investigate the problem involving partial samples, we introduce a heuristic based on the Fisher information matrix (FIM) to estimate the variance of the previously presented MLE solution. Also, in order to accelerate the computation of this measurement, we introduce the concept of equivalent partial samples and develop a fast algorithm, Algorithm 3, to accurately calculate FIM, achieving a speedup of ~ 500 times compared to the brute-force method. Simulation results on both hypothetical and real gene models also show that our FIM-based heuristic gives a good approximation to the value of $Average\left(var(\hat{\theta}_k)\right)$, and accurately predicts the numeric order of this value under different conditions. With this metric, we are also able to demonstrate examples of how to efficiently find low-cost combinations of different sampling techniques to best estimate the isoform compositions in RNA-Seq experiments. Although we are only using individual genes as examples, once we have good assumptions of expression levels of different genes, this procedure can be generalized to all the genes for the low-cost design of actual whole genome RNA-Seq experiments.

What is more, by applying the MLE method to a worm RNA-Seq dataset, we illustrate how we can compare the differential isoform composition between different developmental stages, and how different isoform sets (e.g. with a major/minor isoform missing, with an additional 'dummy" isoform) will affect the MLE result, especially in terms of the maximized likelihood value, showing that the likelihood score is an effective tool for ranking the "fitness" of isoform sets for a particular gene.

Since IQSeq estimates isoform quantity within a probabilistic framework, it does not directly determine the existence of a certain isoform transcript in the data, but rather gives probability measures ($\hat{\Theta}$) and corresponding RPKM values. The result of a secondary experiment with high precision, e.g. qPCR, on a smaller set of genes, can be used as a gold standard dataset to assist answering such existence questions, with either a simple RPKM value threshold that maximizes the prediction accuracy on the gold standard (training) dataset, or more sophisticated classification techniques that takes multiple characteristics (e.g. $\hat{\Theta}$, overall gene expression, FIM-based variance estimation) into account.

The FIM-based variance we are trying to estimate in the proposed algorithm focuses mainly on the expected estimation variance based on different read sets of similar on a same sample, and is a measurement of estimation accuracy. In the case of read sets from different biological replicates, the variances of interest there are usually the actual differences in isoform composition of particular genes between/among the replicates, and analysis on such differences can be generally conducted as a downstream procedure after the isoform quantification calculation.

As sequencing technologies constantly evolve, IQSeq will remain able to provide integrated analysis of different datasets with their own sequencing characteristics, and provide guidelines for optimal RNA-Seq experiment design.

Supporting Information

Text S1 Supplementary material including additional derivation of formulas, proof of lemmas and theorems, and analysis results.
(PDF)

Author Contributions

Conceived and designed the experiments: JD JL MG. Performed the experiments: JD JL. Analyzed the data: JD JL LH AS DM MG. Contributed reagents/materials/analysis tools: JD JL DM MG. Wrote the paper: JD JL MG.

References

1. Gerstein MB, Bruce C, Rozowsky JS, Zheng D, Du J, et al. (2007) What is a gene, post-encode? history and updated definition. Genome Res 17: 669–681.
2. Berget SM, Moore C, Sharp PA (1977) Spliced segments at the 5′ terminus of adenovirus 2 late mrna. Proc Natl Acad Sci U S A 74: 3171–3175.
3. Chow LT, Gelinas RE, Broker TR, Roberts RJ (1977) An amazing sequence arrangement at the 5′ ends of adenovirus 2 messenger rna. Cell 12: 1–8.
4. Gelinas RE, Roberts RJ (1977) One predominant 5′-undecanucleotide in adenovirus 2 late messen- ger rnas. Cell 11: 533–544.
5. Harrow J, Denoeud F, Frankish A, Reymond A, Chen CK, et al. (2006) Gencode: producing a reference annotation for encode. Genome Biol 7 Suppl 1: S4.1–S4.9.
6. Xing Y, Yu T, Wu YN, Roy M, Kim J, et al. (2006) An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. Nucleic Acids Res 34: 3150–3160.
7. Jiang H, Wong WH (2009) Statistical inferences for isoform expression in rna-seq. Bioinformatics 25: 1026–1032.
8. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, et al. (2010) Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol.
9. Richard H, Schulz MH, Sultan M, Nrnberger A, Schrinner S, et al. (2010) Prediction of alternative isoforms from exon expression levels in rna-seq experiments. Nucleic Acids Res 38: e112.
10. Lacroix V, Sammeth M, Guigo R, Bergeron A (2008) Exact transcriptome reconstruction from short sequence reads. In: WABI '08: Proceedings of the 8th international workshop on Algorithms in Bioinformatics. Berlin, Heidelberg: Springer-Verlag. pp 50–63.
11. Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society Series B (Methodological) 39: 1–38.
12. Schervish MJ (1995) Theory of Statistics Springer.
13. Van der Vaart AW (1998) Asymptotic Statistics Cambridge University Press.
14. Du J, Bjornson RD, Zhang ZD, Kong Y, Snyder M, et al. (2009) Integrating sequencing technologies in personal genomics: optimal low cost reconstruction of structural variants. PLoS Comput Biol 5: e1000432.
15. Heber S, Alekseyev M, Sze SH, Tang H, Pevzner PA (2002) Splicing graphs and est assembly problem. Bioinformatics 18 Suppl 1: S181–S188.
16. Hillier LW, Reinke V, Green P, Hirst M, Marra MA, et al. (2009) Massively parallel sequencing of the polyadenylated transcriptome of c. elegans. Genome Res 19: 657–666.

17. Celniker SE, Dillon LAL, Gerstein MB, Gunsalus KC, Henikoff S, et al. (2009) Unlocking the secrets of the genome. Nature 459: 927–930.
18. Gerstein MB, Lu ZJ, Nostrand ELV, Cheng C, Arshinoff BI, et al. (2010) Integrative analysis of the caenorhabditis elegans genome by the modencode project. Science 330: 1775–1787.
19. Li J, Jiang H, Wong WH (2010) Modeling non-uniformity in short-read rates in rna-seq data. Genome Biol 11: R50.
20. Hansen KD, Brenner SE, Dudoit S (2010) Biases in illumina transcriptome sequencing caused by random hexamer priming. Nucleic Acids Res.
21. Bullard JH, Purdom E, Hansen KD, Dudoit S (2010) Evaluation of statistical methods for normal- ization and differential expression in mrna-seq experiments. BMC Bioinformatics 11: 94.
22. Consortium MAQC, Shi L, Reid LH, Jones WD, Shippy R, et al. (2006) The microarray quality control (maqc) project shows inter- and intraplatform reproducibility of gene expression measure- ments. Nat Biotechnol 24: 1151–1161.