Nucleic Acids Research, 2017 **1**
doi: 10.1093/nar/gkx1059

FreePSI: an alignment-free approach to estimating exon-inclusion ratios without a reference transcriptome

Jianyu Zhou[1,2], Shining Ma[3], Dongfang Wang[1], Jianyang Zeng[4] and Tao Jiang[1,2,5,*]

[1]MOE Key Laboratory of Bioinformatics; Bioinformatics Division and Center for Synthetic & Systems Biology, TNLIST, Tsinghua University, Beijing 100084, China, [2]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, [3]Department of Statistics, Stanford University, Stanford, CA 94305, USA, [4]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China and [5]Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA

Received May 17, 2017; Revised October 11, 2017; Editorial Decision October 16, 2017; Accepted October 19, 2017

ABSTRACT

Alternative splicing plays an important role in many cellular processes of eukaryotic organisms. The exon-inclusion ratio, also known as percent spliced in, is often regarded as one of the most effective measures of alternative splicing events. The existing methods for estimating exon-inclusion ratios at the genome scale all require the existence of a reference transcriptome. In this paper, we propose an alignment-free method, FreePSI, to perform genome-wide estimation of exon-inclusion ratios from RNA-Seq data without relying on the guidance of a reference transcriptome. It uses a novel probabilistic generative model based on *k*-mer profiles to quantify the exon-inclusion ratios at the genome scale and an efficient expectation-maximization algorithm based on a divide-and-conquer strategy and ultrafast conjugate gradient projection descent method to solve the model. We compare FreePSI with the existing methods on simulated and real RNA-seq data in terms of both accuracy and efficiency and show that it is able to achieve very good performance even though a reference transcriptome is not provided. Our results suggest that FreePSI may have important applications in performing alternative splicing analysis for organisms that do not have quality reference transcriptomes. FreePSI is implemented in C++ and freely available to the public on GitHub.

INTRODUCTION

Alternative splicing plays a crucial role in many cellular processes of eukaryotic organisms (1). It allows a gene to be transcribed into multiple isoforms (or mRNA transcripts) and hence increases the phenotypic complexity of an organism without increasing its genetic complexity. The exon-inclusion ratio, also known as percent spliced in (PSI), is a popular statistic for measuring alternative splicing events (2). It is defined as the ratio of the relative abundance of all isoforms containing a certain exon over the relative abundance of all isoforms of the gene containing the exon. In other words, the PSI value of an exon tells us how often the exon occurs in all the isoforms of the gene that contains the exon. The PSI values of a gene reflect the intensity of its alternative splicing events and have been widely used in differential expression analysis that aims at detecting spliced exons (3) as well as in the exploration of biological mechanisms of alternative splicing (4–6).

A genome-wide estimation of PSI values remains difficult until the advent of high-throughput RNA-seq technology (7). In recent years, many computational methods have been proposed to analyze RNA-seq data (8), including several for performing genome-wide PSI estimation. The methods for PSI analysis generally fall into two categories: isoform-centric or exon-centric (9). An isoform-centric PSI analysis (10) begins by estimating the relative abundance of each isoform by using a quantification tool such as Cufflinks (11), RSEM (12) CEM (13) or eXpress (14) if a reference transcriptome is given. Once the relative abundance levels of all isoforms have been quantified, the PSI values of each exon in the genome can be easily derived. If no reference transcriptome is available, a transcriptome assembly tool such as Cufflinks, IsoLasso (15), StringTie (16) or TransComb (17) can be used to infer the expressed isoforms as well as their relative abundance from the input RNA-seq data and reference genome.

A common feature of the above quantification/assembly methods is that they all require the input RNA-seq reads to be mapped (or aligned) to the reference genome (or transcriptome) as a preprocessing step. This can be achieved by

---

[*]To whom correspondence should be addressed. Tel: +1 951 8272991; Fax: +1 951 8274643; Email: jiang@cs.ucr.edu

using alignment tools such as Bowtie (18), TopHat (19,20) and HISAT (21). On the other hand, an alignment-free approach for abundance quantification has been proposed recently and implemented in Sailfish (22). The method uses *k*-mer counts to construct profiles of both the input RNA-seq reads and reference transcriptome, and a probabilistic generative model based on the profiles to estimate the abundance of each isoform. As reported in (22), Sailfish is able to achieve a comparable overall accuracy as Cufflinks while maintaining a much higher efficiency. The high efficiency of Sailfish is helped by a light-weight expectation-maximization algorithm for solving the probabilistic model and the parallelizable *k*-mer counting method evolved from Jellyfish (23). Inspired by the alignment-free approach, some 'pseudo-alignment' (or 'quasi-mapping') based methods including Kallisto (24) and Salmon (25) have been proposed very recently in the literature with further improved performance. These methods do not attempt to map reads to precise locations of the reference genome. Instead, they try to identify all isoforms in the reference transcriptome that may potentially contain each specific read. Note that the alignment-free or pseudo-alignment-based approaches for isoform abundance quantification require the existence of a reference transcriptome and their performance clearly depends on the quality of the reference transcriptome.

Exon-centric methods including MISO (26), MATS (27) and rMATS (28) focus on specific exons instead of an entire exome and analyze alternative splicing events such as exon skipping, mutually exclusive exons, intron retention as well as alternative ($5'$ or $3'$) boundaries based on the PSI values of the exons. In particular, MISO can perform alternative splicing analysis on a single biological sample or differential expression analysis on two samples, while MATS and rMATS specialize in the comparison of two samples. These methods all require mapped RNA-seq reads and use Bayesian inference to perform PSI estimation that incurs significant running time. Moreover, the alternative splicing events to be analyzed have to be provided by the user in advance or extracted from a reference transcriptome.

Clearly, the availability of a high quality reference transcriptome is critical for both isoform-centric and exon-centric PSI estimation methods. Although transcriptomes can be assembled from RNA-seq data on-the-fly by using assembly tools such as Cufflinks, IsoLasso, StringTie or TransComb, they are likely to contain a high degree of noise (9). Such noise may significantly affect the accuracy of subsequent PSI estimation. Moreover, even if a reference transcriptome is available, it may not cover all expressed isoforms in the input RNA-seq data. Such an incomplete reference transcriptome may also misguide subsequent PSI estimation.

In this paper, we propose a new method for genome-wide PSI estimation, called FreePSI, that requires neither a reference transcriptome (hence, transcriptome-free) nor the mapping of RNA-seq reads (hence, alignment-free). The first freedom allows FreePSI to work effectively when a high quality reference transcriptome is unavailable and the second freedom not only helps make FreePSI more efficient, it also eliminates the necessity of dealing with multi-reads, which is a challenging problem by itself. Note that this is the first alignment-free method in RNA-seq data analysis that does not require a reference transcriptome. An outline of the method is given below.

FreePSI takes as the input a reference genome with exon boundary annotation and a set of RNA-seq reads. Since a reference transcriptome is not assumed, it uses a weighted directed bipartite graph (called an abundance flow graph) to represent all possible isoforms of a gene and their expression levels. In such a graph, each vertex represents an exon boundary and each edge represents either an exon or an exon junction. The weight of an edge represents the total relative abundance of all isoforms covering the corresponding exon or junction. Obviously, to estimate the PSI value of each exon, it suffices to infer the edge weights in every abundance flow graph. By regarding each edge as a sequence of *k*-mers, FreePSI constructs a novel probabilistic model for generating all observed *k*-mers in the input RNA-seq reads based on the abundance flow graphs for all genes. It then employs the expectation-maximization (EM) framework to solve a genome-wide maximum likelihood estimation (MLE) of the model and a divide-and-conquer strategy to factorize the key optimization problem in the M-step into independent subproblems for each gene, which are then solved by an ultrafast algorithm, conjugate gradient projection descent. The above factorization is crucial for the efficiency of FreePSI because unlike Sailfish whose EM algorithm involves an M-step with a closed-form solution due to the given reference transcriptome, the key optimization problem in the M-step of the EM algorithm of FreePSI does not have a closed-form solution. Finally, it uses a post-processing procedure based on straightforward correlation analysis to "smooth out" the PSI values in each gene.

To evaluate the performance of FreePSI, we compare it with isoform-centric methods including Salmon (the most recent isoform abundance quantification method) and Cufflinks (the most popular transcriptome assembly and isoform quantification method) as well as a representative exon-centric method MISO on both simulated and real data. Our experimental results demonstrate that although FreePSI is unable to match the overall performance of Salmon on simulated data where the correct reference transcriptome is provided, it performs better than Cufflinks without assuming a reference transcriptome (denoted as Cufflinks-A) and MISO in terms of both accuracy and efficiency. In particular, for genes that have large proportions of multi-mapped reads, FreePSI achieves significantly better accuracy than Cufflinks-A. On the other hand, on a real dataset where the true reference transcriptome is unknown, both FreePSI and Cufflinks-A are able to outperform Salmon significantly in terms of accuracy. These results suggest that FreePSI may have important applications in alternative splicing analysis when a high quality reference transcriptome is unavailable.

MATERIALS AND METHODS

Overview

As outlined in Introduction, FreePSI estimates the PSI values of all annotated exons on the reference genome from RNA-seq reads and is both transcriptome-free and alignment-free. It uses a weighted directed bipartite graph, called an abundance flow graph, to represent all possible

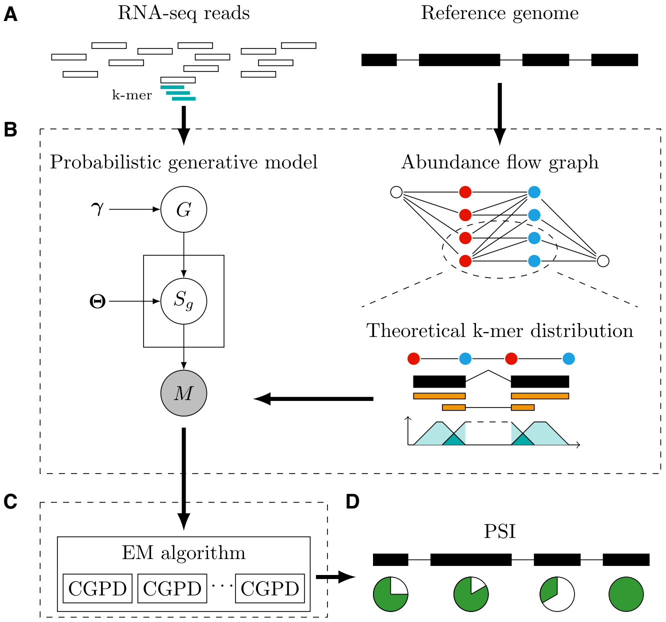

Figure 1. An overview of FreePSI. (**A**) The input of FreePSI includes a reference genome with exon boundary annotation and a set of RNA-seq reads. (**B**) The main component of FreePSI is a probabilistic generative model. The abundance flow graph represents all possible isoforms and their abundance levels. For each exon (or junction), the (theoretical) distribution of *k*-mers in the exon (or junction, respectively) is derived by assuming that the reads were uniformly sequenced. (**C**) An EM algorithm is employed to perform genome-wide inference for the model, and a divided-and-conquer strategy decomposes the key optimization problem in the M-step into independent subproblems for each gene. Each subproblem is solved using a conjugate gradient projection algorithm. (**D**) The output of FreePSI includes estimated PSI values for all exons.

isoforms of a gene and their abundance levels. The PSI values of all exons in the gene can be easily derived from the weights of the edges in the graph, where each edge represents an exon or junction in all possible isoforms. The edge weights are constrained by linear inequalities and can be estimated via an alignment-free approach. The alignment-free estimation formulates (theoretical) *k*-mer distributions on exons/junctions by assuming that the reads were uniformly sequenced and uses a novel probabilistic generative model to describe all observed *k*-mers in the reads. Then it computes a genome-wide MLE of the model by the EM framework, and uses a divided-and-conquer strategy to decompose the key optimization problem in the M-step into independent constrained nonlinear optimization subproblems for each gene. These subproblems are solved in parallel by using an elaborate implementation of an ultrafast conjugate gradient projection descent algorithm. Figure 1 illustrates a flowchart of FreePSI. The details of FreePSI are given in the following subsections.

Abundance flow graph

An *abundance flow graph* (AFG) represents all possible isoforms of a gene and their relative abundance based on the concept of *segments*. There are two types of segments: exon segments and junction segments. An exon segment is defined as an interval on the reference genome sandwiched between two consecutive exon boundaries. For any pair of exon segments i and j that can potentially be joined by a junction read, a junction segment is defined as the concatenation of the length $L_{\text{read}}-1$ suffix of i and the length $L_{\text{read}}-1$ prefix of j, where L_{read} represents the read length. Note that exon segments and junction segments are referred to as expressed segments and junctions, respectively, in (15). Let α_h denote the relative abundance of isoform h and α_{ij} denote the total relative abundance of all isoforms covering the junction segment formed by exon segments i and j. For convenience, we use the notation α_{ii} to denote the total relative abundance of all isoforms covering exon segment i. The PSI value of exon segment i can be calculated by the following equation:

$$\psi_i = \frac{\alpha_{ii}}{\sum_h \alpha_h}$$

Figure 2A illustrates an example of segments. Supplementary Section S1.1 gives the formal definitions of α and PSI as well as a detailed derivation of the above equation.

An AFG is essentially a weighted directed bipartite graph (U, V, E). Here, $U = \{u_i | 1 \le i \le n_{\text{exon}}\}$ represents the left part of the vertices, where u_i denotes the starting boundary of exon segment i and n_{exon} the number of exon segments, and $V = \{v_i | 1 \le i \le n_{\text{exon}}\}$ represents the right part, where v_i denotes the ending boundary of exon segment i. The edges are separated into the forward edges and backward edges, denoted as $E = (E_{\rightarrow}, E_{\leftarrow})$. The forward edges, $E_{\rightarrow} = \{<u_i, v_i> | 1 \le i \le n_{\text{exon}}\}$, represent the exon segments and are weighted as α_{ii}. The backward edges, $E_{\leftarrow} = \{<v_i, u_j> | 1 \le i < j \le n_{\text{exon}}\}$, represent the junction segments and are weighted as α_{ij}. In addition, two vertices s and t representing dummy (source and sink) exons are introduced in the graph to accommodate isoforms with alternative transcription start sites and/or polyadenylation cleavage sites. For each i, an edge $<s, u_i>$ is added with weight α_{si} to denote the total relative abundance of all isoforms starting with exon segment i and an edge $<v_i, t>$ is added with weight α_{it} to denote the total relative abundance of all isoforms ending with exon segment i. An example AFG for a gene consisting of four exon segments is shown in Figure 2B. Note that although an AFG looks very similar to a *splicing graph* introduced in (29), its edges represent exon and junction segments rather than exons and introns.

Clearly, every isoform of the gene corresponds to a path in the AFG from s to t, and vice versa. Hence, the total relative abundance of all isoforms of the gene is equal to the summation of all α_{si}. Moreover, the edge weights in the AFG should satisfy the "flow conservation" property. In other words, for each vertex in $U \cup V$, the total weight of all its in-edges is equal to the total weight of all its out-edges. Supplementary Figure S1 provides an example of the flow conservation property. Using this property, the PSI value of an exon segment can be expressed as

$$\psi_i = \frac{\alpha_{ii}}{\sum_i \alpha_{ii} - \sum_i \sum_{j>i} \alpha_{ij}} \tag{1}$$

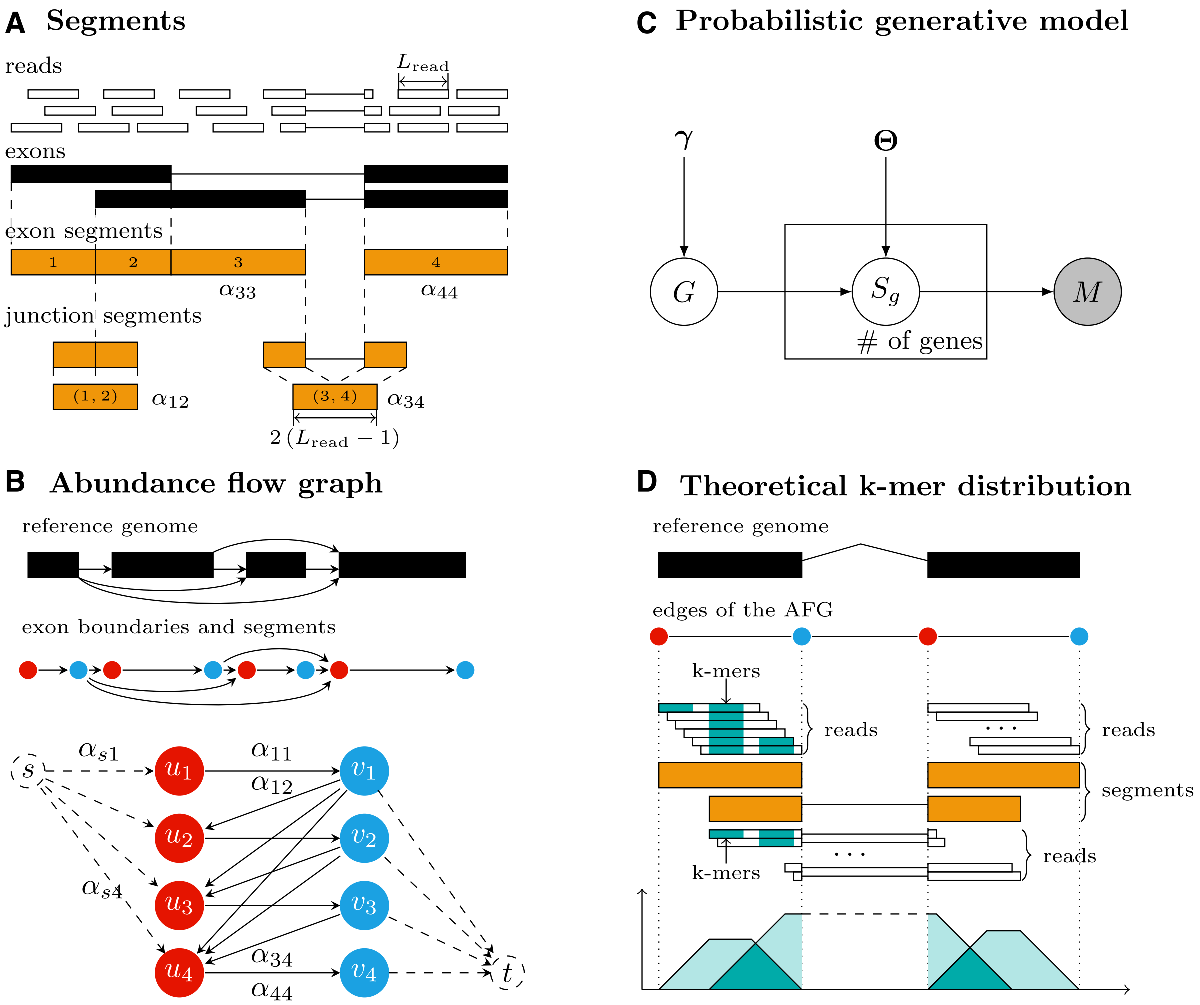

Figure 2. (**A**) Segments. Every annotated exon (black bar) corresponds to one or more exon segments and has two boundaries. If the annotated exons from different isoforms overlap, they are partitioned into disjoint exon segments. If exon segment *i* is joined with exon segment *j* in an isoform, a junction segment of length $2(L_{\text{read}}-1)$ would be formed. For examples, the junction segment (1, 2) is formed by exon segments 1 and 2 and the junction segment (3, 4) by exon segments 3 and 4. The parameter α associated with each segment represents the total relative abundance of all isoforms covering the segment. (**B**) Abundance flow graph. The figure shows an example AFG for a gene consisting of four exon segments. The AFG is constructed according to the exon boundary annotation of the reference genome. The red vertices represent the starting boundaries of the exon segments, and the blue vertices represent the ending boundaries of the exon segments. The forward edges from the red vertices to the blue vertices represent the exon segments and the backward edges from the blue vertices to the red vertices represent the junction segments. The parameter α defined for each segment is assigned as its corresponding edge weight. Two dummy vertices *s* and *t* are introduced to handle isoforms that begin and/or end with different exon segments. (**C**) Probabilistic generative model. The graphical structure of the probabilistic generative model is a three-layer Bayesian network. The distributions of the random variables *G* and S_g are determined by the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\Theta}$. S_g has gene number replicates, and the random variable *M* is observable. (**D**) Theoretical *k*-mer distribution. The theoretical distribution of *k*-mers is derived from the assumption that the reads are uniformly distributed in an isoform. Since each read belongs to one segment, they are also uniformly distributed in a segment. Hence, *k*-mers near the middle of a segment are usually covered by more reads in the segment than *k*-mers near the boundaries of the segment. This gives rise to a trapezoid shaped theoretical distribution of *k*-mers in a segment. Note that a *k*-mer may be shared by multiple (exon and junction) segments.

The following inequalities will help us in the estimation of the α values:

$$\alpha_{ii} \geq \sum_{j<i} \alpha_{ji}, \quad \alpha_{ii} \geq \sum_{j>i} \alpha_{ij} \qquad (2)$$

The detailed derivations of these linear constraints are given in Supplementary Section S1.2.

Probabilistic generative model

Since each read is generated from a segment randomly and each read defines a set of *k*-mers, each segment generates a random set of *k*-mers. We construct a three-layer Bayesian network to model the mixture of all *k*-mers generated by the segments from all genes, as shown in Figure 2C. In the following, *s* denotes a segment (exon or junction) and *g* a gene. Let *G* represent a random gene, S_g a random segment of gene *g*, for each *g*, and *M* a random *k*-mer. We use $P(G = g) = \gamma_g$ to represent the probability that a read is generated from gene *g*, $P(S_g = s|G = g) = \theta_{gs}$ to represent the conditional probability that a read generated from gene *g* belongs to segment *s*, and $P(M = m)$ to denote the probability of observing *k*-mer *m*. Then, the probability of an observed k-mer m can be expressed as

$$\mathrm{P}(M=m)=\sum_{g}\gamma_g\sum_{s\in g}\theta_{gs}\mathrm{P}\left(M=m|S_g=s,G=g\right)$$

where $\mathrm{P}(M=m|S_g=s,G=g)$ denotes the (theoretical) distribution of k-mers on segment s in gene g assuming that the reads are sampled uniformly. See Supplementary Section S2.3 for a detailed derivation of this probability.

Again assuming that the reads are uniformly distributed on each segment, then the parameters α can be approximated by γ and θ as follows, if the relative abundance is measured by TPM (transcripts per million) :

$$\alpha_{gs}\approx\frac{Z_2}{Z_1}\frac{\gamma_g\theta_{gs}}{\widetilde{L}_{gs}} \tag{3}$$

where α_{gs} denotes the total relative abundance of all isoforms covering segment s in gene g, Z_1 and Z_2 are normalization constants, and $\widetilde{L}_{gs}$ is the effective length of segment s in gene g. That is, if L_{gs} denotes the length of segment s in gene g, then $\widetilde{L}_{gs}=L_{gs}-L_{\text{read}}+1$. Hence, the PSI value defined in Equation 1 can be estimated from θ and the linear constraints in Equation 2 can also be applied to θ. The detailed derivation of the above approximation can be found in Supplementary Section S2.4.

The theoretical k-mer distribution over a segment is necessary because even if the reads are sampled uniformly, the k-mers are not distributed uniformly across the segment. See Figure 2D for an illustration. Note that the segment defines $\widetilde{L}_{gs}$ distinct reads, and each read contains $L_{\text{read}}-K+1$ k-mers. Let $\mathcal{F}_{gsm}$ denote the number of distinct reads covering a k-mer m on segment s in gene g. Then, the theoretical distribution of all k-mers generated from the segment can be written as

$$\mathrm{P}\left(M=m|S_g=s,G=g\right)=\frac{\mathcal{F}_{gsm}}{\widetilde{L}_{gs}\left(L_{\text{read}}-K+1\right)} \tag{4}$$

From now on, let c_{gsm} denote $\mathrm{P}(M=m|S_g=s,G=g)$. More details of the above discussion are given in Supplementary Section S2.5.

Therefore, to quantify PSI values, it suffices to perform a MLE of the parameters γ and θ.

Expectation-maximization algorithm

The MLE can be formulated as the following nonlinear constrained optimization problem:

$$\begin{aligned}\max\quad & \sum_{m}n_m\log\left(\sum_{g}\gamma_g\sum_{s\in g}\theta_{gs}c_{gsm}\right)\\ \text{s.t.}\quad & \boldsymbol{A}_g\boldsymbol{\theta}_g\geq 0,\quad \text{for all genes } g\\ & \sum_{s\in g}\theta_{gs}=1,\quad \text{for all genes } g\\ & \sum_{g}\gamma_g=1,\quad \forall\gamma_g\geq 0,\quad \forall\theta_{gs}\geq 0\end{aligned}$$

where n_m denotes the number of occurrences of k-mer m in all input reads, $\boldsymbol{\theta}_g$ the vector formed by all θ_{gs}, $s\in g$, and $\boldsymbol{A}_g\boldsymbol{\theta}_g$ the matrix form of the linear constraints in Equation (2) (see Supplementary Equations S2.7–S2.10 for details). See Supplementary Section S3.1 for this formulation. We develop an EM algorithm below to solve the optimization problem iteratively.

Let $\boldsymbol{\gamma}$ denote the vector formed by all γ_g and $\boldsymbol{\Theta}$ the matrix formed by stacking all vectors $\boldsymbol{\theta}_g$. Before the iteration starts, an initial feasible solution $\boldsymbol{\gamma}^{(0)}$ and $\boldsymbol{\Theta}^{(0)}$ is obtained based on the k-mer profiles of the input RNA-seq reads, as sketched in Algorithm S1 of Supplementary Section S3.2. In general, the E-step of the EM algorithm is to generate a function for the expected log-likelihood based on the current estimation of the parameters. Assuming that t iterations have been completed, the expected log-likelihood is then

$$\mathcal{Q}(\boldsymbol{\gamma},\boldsymbol{\Theta})=\mathcal{Q}^{\mathrm{I}}(\boldsymbol{\gamma})+\sum_{g}\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$$

where

$$\mathcal{Q}^{\mathrm{I}}(\boldsymbol{\gamma})=\sum_{m}\sum_{g}\mu_{gm}^{(t)}\log\left(\gamma_g\right)$$

$$\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)=\sum_{m}\mu_{gm}^{(t)}\log\left(\sum_{s\in g}\theta_{gs}c_{gsm}\right)$$

$$\mu_{gm}^{(t)}=\frac{\gamma_g^{(t)}\sum_{s\in g}\theta_{gs}^{(t)}c_{gsm}}{\sum_{g}\gamma_g^{(t)}\sum_{s\in g}\theta_{gs}^{(t)}c_{gsm}}$$

where $\mu_{gm}^{(t)}$ denotes the posterior probability of k-mer m being generated from gene g based on the last estimations of $\boldsymbol{\gamma}^{(t)}$ and $\boldsymbol{\Theta}^{(t)}$. More details of the derivation are given in Supplementary Section S3.2. The expectation $\mathcal{Q}(\boldsymbol{\gamma},\boldsymbol{\Theta})$ is decomposed into the summation of two terms, $\mathcal{Q}^{\mathrm{I}}(\boldsymbol{\gamma})$ and $\sum_g\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$, that involve independent parameters and constraints.

The M-step is to maximize the expected log-likelihood given in the E-step. Given the above decomposition, the maximization problem can be divided into two parts. The first part is

$$\begin{aligned}\max\quad & \mathcal{Q}^{\mathrm{I}}(\boldsymbol{\gamma})\\ \text{s.t.}\quad & \sum_{g}\gamma_g=1,\quad \forall\gamma_g\geq 0\end{aligned}$$

This part has a closed-form solution (see Supplementary Section S3.2). The second part $\sum_g\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ can be solved by a divide-and-conquer strategy, which leads to an optimization subproblem for each gene g:

$$\begin{aligned}\max\quad & \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)\\ \text{s.t.}\quad & \boldsymbol{A}_g\boldsymbol{\theta}_g\geq 0,\quad \sum_{s\in g}\theta_{gs}=1,\quad \forall\theta_{gs}\geq 0\end{aligned}$$

Unfortunately, these subproblems do not have closed-form solutions due to the presence of linear inequality constraints. Hence, we use a conjugate gradient projection descent algorithm to solve them.

The conjugate gradient projection descent (CGPD) algorithm (30) is an efficient algorithm for convex optimization under linear constraints. The detailed CGPD algorithm is given in Algorithm S2 of Supplementary Section S3.3. Its key idea is to perform line search along the conjugate directions in the null space of active constraints. Since the objective function $\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ in our problem is a continuous differentiable convex function and all the constraints are linear, the CGPD algorithm is particularly suitable. Because $\nabla\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ is much easier to compute than $\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ (Supplementary Section S4.2.1), we choose the secant method for line search in CGPD, which only requires the first-order derivatives of the objective function and has a super linear convergence rate.

Post-processing

The above EM algorithm results in an estimation of the parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\Theta}$. Since these parameters only provide an approximation of the parameters α (and thus PSI), as shown in equation 3, some post-processing could be applied to refine the raw estimation of PSI values. We adopt a post-processing procedure based on two assumptions: for each gene, (i) a small number of isoforms are expressed and (ii) there exists an exon segment included in all expressed isoform.

The first assumption implies that the PSI values of the exon segments from each gene should fall into a small set of distinct numbers. Hence, we could potentially reduce noise by clustering similar PSI values. An average-linkage hierarchical clustering algorithm with Euclidean distance is adopted here. Once a hierarchical clustering tree is obtained, we cut it to result in the least number of clusters such that either the maximum standard deviation of PSI values in each cluster is less than 0.06 or the mean of the standard deviations of PSI values in all clusters is less than 0.05. Then, the raw estimates of PSI values in each cluster are revised to the mean values of the cluster.

The second assumption implies that there should be at least one exon segment with PSI value equal to 100% PSI. Hence, we rescale the above revised PSI values by dividing each by the maximum PSI value of any exon segment in the same gene.

It turns out that both the linear constraints in Equation 2 and the above processing steps are crucial for FreePSI to obtain a good estimation of exon-inclusion ratios.

Implementation

FreePSI is implemented mainly in `C++`. We utilize the third-part library `Eigen` for matrix manipulations. Parallelization of the program is achieved by using `OpenMP`. Some key issues of the implementation are discussed below.

K-mer hash table. As a preprocessing step, Jellyfish is invoked to count k-mers in the input RNA-seq reads. Then, FreePSI indexes each k-mer as a 64-bit integer using a linear algorithm that scans the reads and segments only once, as shown in Algorithm S3 of Supplementary Section S4.1. The k-mer indices are then hashed into a `C++11` built-in hash table, `unordered map`. Each entry of the hash table stores two pieces of information: one is the count of the k-mer in the reads and the other is a list of segments containing the k-mer as well as the corresponding coefficient c_{gsm}. During hashing, all k-mers that share the same segments are combined into one representative k-mer, and their counts and the coefficients are also combined. This shrinks the size of the hash table significantly. In our simulation experiments, the shrinkage rates were ∼80% on average, which greatly reduced the computational complexity of subsequent steps. A similar strategy was also adopted in (22).

Implementation of the EM algorithm. The convergence criterion of the EM algorithm is that the log-likelihood increases by less than 10^{-6} in an iteration. In order to speed up the computation, sparse matrix and parallelization techniques are adopted. In particular, the `reduction` function in `OpenMP` is employed to allow for concurrent calculation of the summation in the E-step of the algorithm. On the other hand, the M-step, composed of many independent subproblems, can be easily parallelized with `OpenMP`.

Efficiency improvements for CGPD. The CGPD algorithm used in the M-step is the efficiency bottleneck of FreePSI. Four techniques are implemented to improve its efficiency.

The first one is reordering matrix multiplications, which has also been considered carefully in all implementations of the CGPD algorithm. It is well-known that rearranging the order of matrix multiplications can potentially reduce time complexity drastically. In particular, the matrix multiplications in CGPD can be reordered so that only matrix-vector multiplications are performed. In our simulation experiments, we found that this technique contributed significantly to the efficiency of FreePSI.

The second strategy is the compaction of sparse parameters. Before calling CGPD, the zero entries in $\boldsymbol{\theta}_g$ as well as their associated c_{gsm}'s and optimization constraints are removed. Only the remaining parameters are updated by the CGPD algorithm. The compaction may reduce the number of iterations by lowering the dimensionality of the search space in CGPD.

The third technique is offline computation for a part of $\nabla\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$. The gradient $\nabla\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ is required in every iteration of CGPD. We observe that the gradient can be decomposed into the product of an iteration-invariant part and an iteration-variant part, while the iteration-invariant part consumes a large amount of running time (see Supplementary Section S4.2.1 for the details). So, computing the iteration-invariant part in advance can surely enhance FreePSI's efficiency.

The last technique is replacing outer products of vectors into in-space column-wise operations. The CGPD algorithm computes outer products of vectors during its iterations. A direct implementation allocates new memory space for storing the result matrix of each outer product, which is then added to or subtracted from another matrix. However, the storage of intermediate matrices is unnecessary. Hence, we replace an outer product operation by some column-wise operations that can be performed in-space. That is, the column vector is first multiplied with an element of the row vector. The result is then added to or subtracted from the

Table 1. Accuracy measured by both Pearson and Spearman correlations of the methods on both simulated and real RNA-seq datasets

Data	Evaluation	Correlation	MISO	Salmon	Cufflinks-G	Cufflinks-A	FreePSI
Simulated data	Genome-wide	Pearson	-	0.998	0.986	0.826	0.869
		Spearman	-	0.975	0.930	0.716	0.733
	Exon-centric	Pearson	0.807	0.995	0.985	0.884	0.895
		Spearman	0.759	0.975	0.951	0.848	0.844
Real data	Exon-centric	Pearson	0.892	0.788	0.790	0.877	0.912
		Spearman	0.838	0.762	0.754	0.875	0.897

corresponding column of another matrix (see Supplementary Section S4.2.2 for more details).

RESULTS

In this section, we evaluate the performance of FreePSI by comparing it with the other state-of-the-art PSI quantification methods on both simulated and real RNA-seq data. More specifically, we compare two transcriptome-free methods including FreePSI and Cufflinks (v2.2.1) and three transcriptome-guided methods including Salmon (v0.7.2), Cufflinks (v2.2.1) and MISO (v0.5.3). Here, Cufflinks is considered as both a transcriptome-free method and a transcriptome-guided method. In the former case (denoted as Cufflinks-A), it is used to perform both transcriptome assembly and abundance quantification; but in the latter case (denoted as Cufflinks-G), it is only run to quantify the relative abundance of the annotated isoforms. In addition, HISAT (v2.0.4) is used for mapping reads in the alignment-based methods (Cufflinks and MISO) and Jellyfish (v2.2.6) is used for counting *k*-mers in FreePSI. All the methods are run on a 64-bit Linux server consisting of two CPUs with 16 cores each and 96 GB memory.

Performance on simulated data

We use Flux Simulator (31) to simulate RNA-seq data. Here, UCSC *hg38* is used as the reference genome and the RefSeq *refGene* annotation (23,983 genes and 57,822 isoforms) is used as the reference transcriptome. The expression level of each isoform is assigned according to a power-law distribution, and roughly 100 million strand-specific paired-end reads of length 76bp are simulated with the default sequencing error profile. The overall mapping rate of the simulated reads to the reference genome is 92.2%.

FreePSI requires annotated exon boundaries as a part of its input. Although such information is provided in the reference genome (UCSC *hg38*), in order to be consistent with the simulation, we extract exon boundaries from the reference transcriptome by merging the annotated isoforms (since the reads are simulated from them directly). In particular, overlapping exons from different isoforms are split into disjoint "short exons", and all short exons of lengths >30 bp are retained in the exon annotation. To avoid dealing with genes with too many short exons, genes with more than 40 short exons are removed from the annotation, which accounts for 1.4% of all genes. Finally, the exon annotation is represented as a series of disjoint intervals on the reference genome. The annotation of alternative splicing events required by MISO is extracted from the reference transcriptome using a built-in toolkit of MISO. It includes five types of alternative splicing events: skipped exon, retained intron, mutually exclusive exons and alternative ($5'$ and $3'$) boundaries.

The *k*-mer length is a key parameter in FreePSI. It is set as 27 bp in the experiment. (See Supplementary Section S5.4 for a discussion on the impact of the *k*-mer length on the performance of FreePSI.) Jellyfish is used to count 27-mers from the simulated RNA-seq reads while filtering out *k*-mers that contain any base with error probability over 1%. Both Cufflinks-A and Cufflinks-G are run with the "rescue method" for multi-read refinement and the positional bias correction enabled, while MISO and Salmon are run with the default configurations. More details of the running configurations are given in Supplementary Section S6.

To obtain the ground truth for evaluation, we transform the simulated expression levels of the annotated isoforms into PSI values of each annotated exon. The accuracy performance is evaluated in two ways: genome-wide and exon-centric. The genome-wide evaluation tests the overall accuracy of estimated PSI values across all exons of all genes. MISO is excluded from this evaluation because it does not provide a genome-wide estimation of PSI values. Totally, 7032 genes with expression levels over 10 TPM are selected for the evaluation. The accuracy is measured by the Pearson and Spearman correlations between estimated and true PSI values of all annotated exons in the selected genes. The results of the compared methods are listed in Table 1 (rows 1 and 2).

The exon-centric evaluation is concerned with the accuracy of estimated PSI values of alternatively spliced exons, which are exons with over 95% of their regions covered by the annotated alternative splicing events extracted above. According to this definition, 10,919 alternatively spliced exons are selected for the evaluation. The Pearson and Spearman correlation coefficients between estimated and true PSI values of alternatively spliced exons are listed in Table 1 (rows 3 and 4).

Both genome-wide and exon-centric evaluations arrive at similar conclusions. The transcriptome-guided isoform-centric methods, Salmon and Cufflinks-G, delivered nearly perfect estimates of PSI. This is clearly due to the fact that the methods used the same reference transcriptome as employed in the simulation. On the other hand, the accuracy of the exon-centric method, MISO, is just acceptable. The transcriptome-free methods, FreePSI and Cufflinks-A, were also able to deliver strong correlation results, with FreePSI performing slightly better than Cufflinks-A. A scatter plot of the estimation results of FreePSI is shown in Figure 3A. Similar scatter plots for the other methods can be found in Figures S2 and S3 of Supplementary Section S5.1.

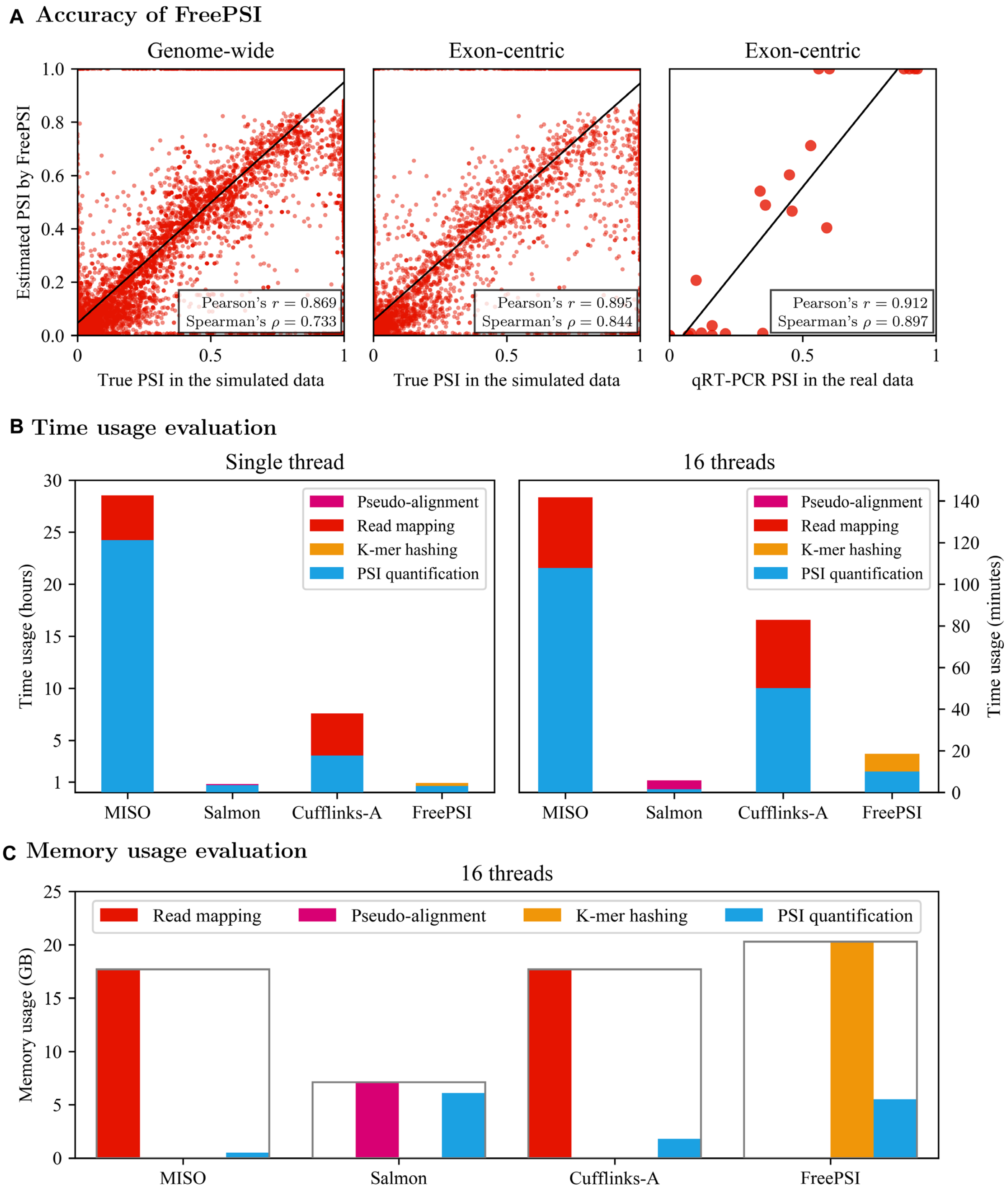

Figure 3. (**A**) Accuracy of FreePSI. The left and center scatter plots show the correlation between the true PSI values and PSI values estimated by FreePSI on the simulated data using genome-wide and exon-centric evaluation methods, respectively. The right scatter plot shows the result of FreePSI on the real data. The Y-axis in these plots represents the estimated values of PSI and the X-axis the ground truth, respectively. (**B**) Time usage evaluation. The left histogram shows the time usages in hours of different methods with a single thread, while the right histogram shows the time usages in minutes of the methods with 16 threads. Color is used to break the running time of a method into preprocessing time and quantification time. (**C**) Memory usage evaluation. This histogram shows the memory usages in GB of different methods with 16 threads. Color is used to break the memory usage of a method into preprocessing memory and quantification memory. The frame boxing the bars represents the peak memory usage of each method.

Performance on real data

Although experiments on real data can provide a more realistic assessment of performance, ground truth is often difficult to obtain for real data. In the evaluation of many RNA-seq quantification methods on real data, the results of quantitative real-time polymerase chain reaction (qRT-PCR) experiments have been used as the ground truth of expression levels of isoforms/genes. The limited number of isoforms or splicing events considered in qRT-PCR experiments makes it difficult to perform a genome-wide evaluation, but we can still use it to conduct an exon-centric evaluation of the PSI estimation methods.

We download the qRT-PCR data together with RNA-seq data studied in (28) (SRA accession: SRR536348). The RNA-seq dataset consists of ∼250 million strand-specific paired-end reads with length 101 bp. The qRT-PCR data concern 34 skipped exon events under the UCSC *hg19* annotation, and provide the true PSI values of these events. To process the RNA-seq data, we first use Sickle (32) (https://github.com/najoshi/sickle) to perform quality control on the reads. Then, the tolerated error probability in Jellyfish is decreased to 0.1% for each base when it is used to count *k*-mers. *k*-mers that occur fewer than 10 times are removed. The other processing steps are identical to those in the simulation experiment.

Out of the 34 skipped exons detected by the qRT-PCR data, 22 can be mapped to our annotated exon boundaries. Hence, we perform an exon-centric evaluation only on these 22 skipped exons. The Pearson and Spearman correlation coefficients between qRT-PCR and estimated PSI values estimated by different methods on these 22 exons are listed in Table 1 (rows 5 and 6).

We observe that the transcriptome-guided methods Salmon and Cufflinks-G performed much worse than the other methods on this real dataset. This is perhaps due to the difference between the reference transcriptome and the true transcriptome expressed in the real data. In particular, Salmon and Cufflinks-G only estimated the relative abundance for the annotated isoforms, and would ignore all isoforms that are actually expressed in the data but missing in the reference transcriptome. Such reliance on a correct reference transcriptome might explain the 20% accuracy performance drop on real data compared with the simulation experiment. On the other hand, MISO performed much better on this real data than on the simulated data. The good performance of MISO on these 22 splicing events can perhaps be explained by the fact that it is designed for estimating PSI values of specific splicing events. The transcriptome-free methods, FreePSI and Cufflinks-A, continued to deliver strong correlation coefficients, again with FreePSI performing better than Cufflinks-A. Since these methods do not rely a given reference transcriptome, they are able to deal with any set of expressed isoforms and provide robust performance on data with unknown (or incomplete) transcriptomes. A scatter plot of the results of FreePSI is shown in Figure 3A, and scatter plots for the other methods are given in Supplementary Figrue S4 of Supplementary Section S5.2.

Efficiency evaluation

The efficiency of a quantification method is as important as its accuracy. We present the running time of the above PSI quantification methods using a single thread or 16 threads separately in the above simulation experiment. While single-thread running time represents the sequential time-efficiency of an algorithm, 16-thread running time could suggest the parallelizability of the algorithm as well as its practical time efficiency when computer clusters (or multi-core machines) are available. Since the methods preprocess data differently, we also show the time spent on preprocessing in each method besides quantification. In particular, the alignment-based methods (Cufflinks and MISO) begin by mapping reads to the reference genome using HISAT, the pseudo-alignment-based method (Salmon) starts by constructing a pseudo-alignment and the alignment-free method (FreePSI) begins by building the *k*-mer hash table. Figure 3B shows the running time of these methods.

As shown in Figure 3B, the running time of all the methods compared is quite acceptable with 16 threads. Salmon and FreePSI were able to complete the job within 20 min, while the alignment-based methods (Cufflinks and MISO) required more than an hour for both read mapping and PSI quantification. In the case of using a single thread, both Salmon and FreePSI finished the job within one hour, while Cufflinks-A spent about 8 h and MISO about 28 h. Comparing the two transcriptome-free methods, we observe that FreePSI ran about four times as fast as Cufflinks-A with 16 threads and eight times with a single thread. This suggests that the time efficiency of FreePSI is much better than that of Cufflinks-A.

We also present the memory footprints of these methods with 16 threads in Figure 3C. The amounts of memory required by all methods are acceptable. The alignment-based methods (Cufflinks and MISO) exhibited similar memory complexity patterns. The peak memory of both methods took place in read mapping, and the memory cost in the quantification process was very low (under 2GB). Salmon and FreePSI showed another pattern of memory complexity. Both methods required similar amounts of memory when estimating PSI, while FreePSI used about three times of memory as Salmon in the preprocessing step. This seems to be reasonable because FreePSI has to model all possible isoforms without a reference transcriptome, i.e. it has to build a large hash table to store the relationship between *k*-mers and all exon segments and possible junction segments covering all genes.

DISCUSSION

The above experimental results on simulated and real RNA-seq data demonstrate that FreePSI performs well in both accuracy and efficiency. In this subsection, we discuss important factors that may affect the performance of FreePSI as well as the other methods.

Impact of sequencing depth

Sequencing depth is often considered as impact factor in RNA-seq analysis. In order to observe how sequencing depth may affect the accuracy of the PSI quantification methods, we simulate two more RNA-seq datasets, with 20 million reads and 50 million reads each, respectively. The accuracy results of MISO, Salmon, Cufflinks-A, and FreePSI on all three simulated datasets are shown in Figure 4A. Clearly, the trend is the same for all the methods and the accuracy gets better when the sequencing depth increases. While Salmon maintained a high accuracy for all three sequencing depths, FreePSI was able to achieve a decent correlation at 0.817 on the dataset with only 20 million reads. In other words, FreePSI can be used to provide a robust estimate of PSI values on RNA-seq data with a broad range of sequencing depths, especially when a high quality reference transcriptome is unavailable.

Impact of reference transcriptome

As shown in Table 1, the performance of the transcriptome-guided methods dropped sharply in the real data experiment, although they were nearly perfect in the simulation experiment. On the contrary, the performance of transcriptome-free methods remained robust on both simulated and real datasets. A plausible explanation of the poor performance of the transcriptome-guided methods on the real dataset is the mismatch between the reference transcriptome and true transcriptome expressed in the data. In order to test how the PSI quantification methods are affected by the quality of the reference transcriptome, we conduct the following simulation experiment.

We use the simulated dataset with 100 million reads, but provide a randomly selected subset of isoforms from the RefSeq *refGene* annotation as the input reference transcriptome for Salmon and Cufflinks-G. In other words, the transcriptome used in the simulation (i.e. the RefSeq *refGene* annotation) is the true transcriptome, but we assume that only an incomplete reference transcriptome is known. We use the sampling rate of the reference transcriptome to represent the coverage of the true transcriptome. As shown in Figure 4B, the accuracy of Salmon and Cufflinks-G (measured by Pearson correlation) decreased almost linearly with the drop of the sampling rate. When the sampling rate decreased to 80%, the performance of Salmon and Cufflinks-G was much worse than that of FreePSI.

In practice, reference transcriptomes are often incomplete and many organisms do not have well-annotated transcriptomes. Hence, a plausible approach would be to assemble the transcriptome first and then quantify PSI values, as illustrated in Cufflinks-A. In order to test how the quality of quantification would impact PSI estimation, we conduct another experiment that applies one of the best isoform abundance quantification methods, Salmon, to perform quantification based on the transcriptome assembled by Cufflinks-A (denoted as Cufflinks-A-Salmon). As Supplementary Figure S5 demonstrates, Cufflinks-A and Cufflinks-A-Salmon performed very similarly on both simulated and real data. This may suggest that the bottleneck of estimating PSI values without a reference transcriptome for methods based on transcript quantification is still the quality of transcriptome assembly. Given the difficulty of transcriptome assembly, transcriptome-free methods such as FreePSI are expected to have important applications in the analysis of many real RNA-seq data.

Impact of gene expression level

The above genome-wide performance evaluation on simulated data focused on the performance of the methods on highly expressed genes (i.e. abundance $\geq$ 10 TPM). To study how the expression level of a gene may influence the accuracy of PSI estimation, we consider subsets of genes with abundance above various TPM thresholds (i.e. 0, 1, 2, 5 and 10) in the simulated data with 100M reads. The performance of the methods on these subsets of genes is shown in Figure 4C. When the TPM threshold increases from 0 to 10, the accuracy measured by the Pearson correlation of all methods generally increases. Clearly, the expression level of a gene has significant impact on the performance of the two transcriptome-free methods (i.e. Cufflinks-A and FreePSI). More highly expressed genes are generally expected to produce more reads and thus correctly assembled isoforms, which lead to more correctly estimated PSI values. The accuracy of FreePSI is always better than that of Cufflinks-A for all the TPM thresholds. Interestingly, the performance of both methods decrease significantly when the TPM threshold increases from 0 to 1. This suggests that the methods are able to deal with unexpressed genes better than lowly expressed genes. This is true for Cufflinks-A because when a gene is not expressed, no isoform will likely be assembled and thus all PSI values of the gene will be output as 0 (correctly) in our experiment. As for FreePSI, although an unexpressed gene may attract a few noisy k-mers, their effect will likely be diminished by the linear constraints in the EM algorithm and the post-processing step of FreePSI, leading to (correctly) estimated PSI values of 0 for the gene. The robust performance of transcriptome-guided methods (*i.e.*, Cufflinks-G and Salmon) suggests that a correct reference transcriptome is important to PSI estimation especially for lowly expressed genes.

Impact of multi-mapped reads

Table 1 suggests that FreePSI provides a better estimate than Cufflinks-A and Salmon performs better than Cufflinks-G (and MISO) on simulated data. In other words, the alignment-free or pseudo-alignment-based methods generally perform better than the alignment-based methods, with or without the reference transcriptome. The advantage of alignment-free and pseudo-alignment-based methods in PSI estimation can perhaps be explained by considering the impact of multi-mapped reads. Cufflinks first uses uniquely mapped reads to estimate the relative abundance of isoforms and then employs a 'rescue method' to refine the estimates using multi-mapped reads. On the other hand, the alignment-free methods and pseudo-alignment-based methods do not distinguish multi-mapped reads from uniquely mapped reads, and use all reads simultaneously to perform quantification. We conduct a simple simulation experiment below to study the impact of these different treatments of multi-mapped reads on PSI estimation.

We consider the simulated dataset with 100 million reads again. Among all mapped reads, 2.48% are mapped to mul-

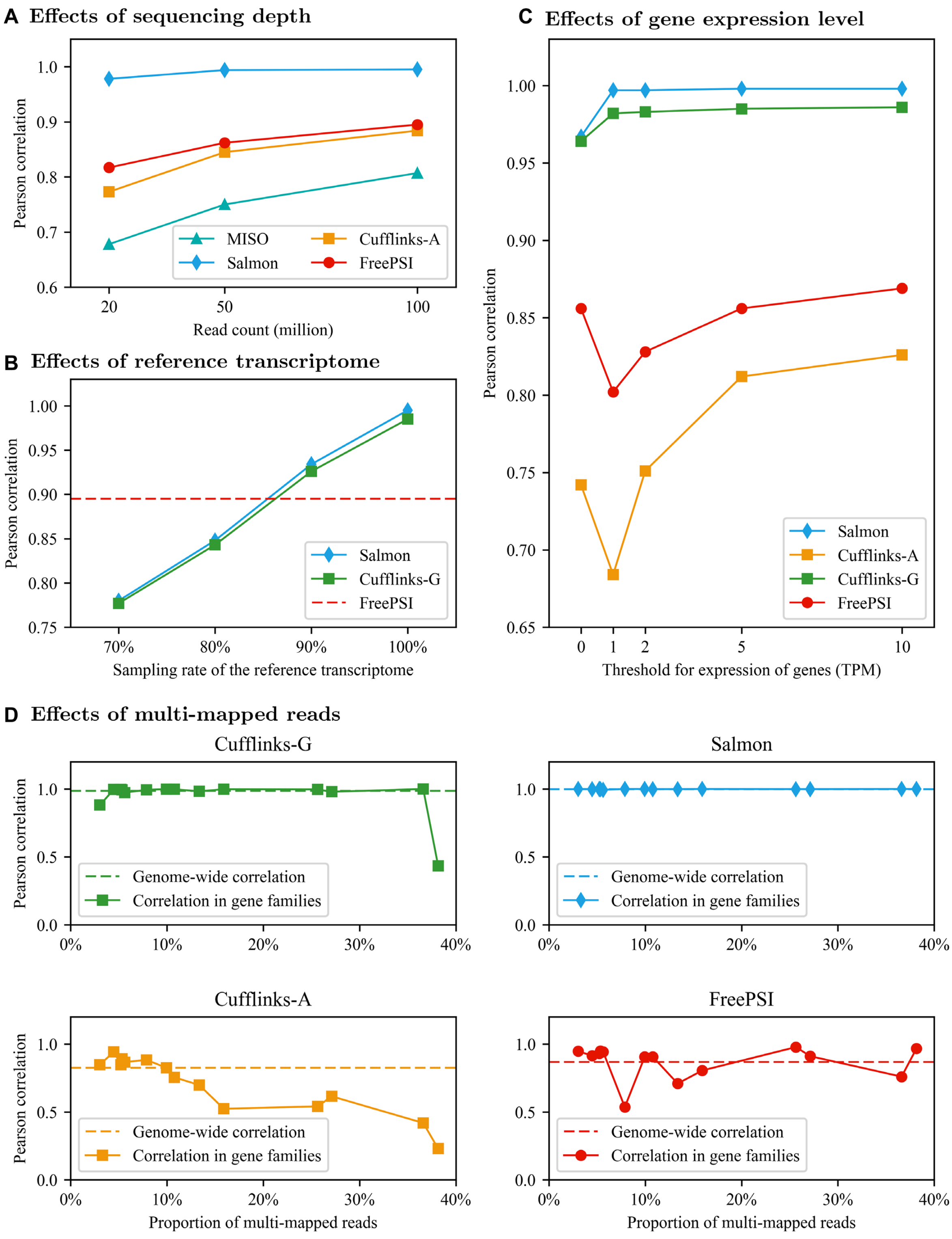

Figure 4. (**A**) Impact of sequencing depth. The figure shows Pearson correlation between estimated PSI values and the ground truth on the spliced exons (i.e. exon-centric evaluation) under various sequencing depths. (**B**) Impact of reference transcriptome. The X-axis represents the sampling rate used for creating the reference transcriptome for Salmon and Cufflinks-G. The Y-axis represents Pearson correlation between estimated PSI values and the ground truth on the spliced exons (i.e. exon-centric evaluation). The dashed line denotes the performance of FreePSI in the exon-centric evaluation as a reference. (**C**) Impact of gene expression level. The figure shows the Pearson correlation between estimated PSI values and the ground truth on expressed genes under different TPM thresholds. (**D**) Impact of multi-mapped reads. The four plots show the performance of four PSI estimation methods on 14 gene families with high proportions of multi-mapped reads. Each point represents Pearson correlation on all exons of the genes in the corresponding family. The X-axis represents the proportion of multi-mapped reads in each gene family. The dashed line denotes the Pearson correlation coefficient obtained by the method in genome-wide evaluation as a reference. The full details of all results discussed in this figure can be found in Supplementary Tables S1, S2, S3 and S4 of Supplementary Section S5.5.

tiple positions. Since multi-mapped RNA-seq reads are generally from genes with similar sequences, we retrieve gene families from the HGNC database and consider large gene families that have large numbers of isoforms. These gene families are expected to result in large portions of multi-mapped reads in the simulated dataset with 100 million reads. Altogether, 93 gene families are selected, each of which contains >20 genes and at least twice as many isoforms. Out of these gene families, 14 contain multi-mapped reads that are more than 2.48% of their total numbers of mapped reads. Figure 4D shows the accuracy of the PSI estimation methods on the 14 gene families with different proportions of multi-mapped reads.

While Salmon's performance was robust across all gene families and remained nearly optimal, Cufflinks-G performed well on most gene families but failed to obtain an acceptable estimate of PSI values on the gene family with the highest proportion of multi-mapped reads. The performance of FreePSI fluctuated slightly on the gene families around its genome-wide performance. However, the performance of Cufflinks-A clearly decreased with the increased proportion of multi-mapped reads. This simple experiment illustrates that the performance of alignment-free and pseudo-alignment-based methods are generally not affected by the existence of multi-mapped reads, but the performance of alignment-based methods may suffer from a proportion of multi-mapped reads. In particular, although the “rescue method” was enabled, without the guidance of the reference transcriptome, the performance of Cufflinks-A still suffered significantly from multi-mapped reads. Therefore, the advantage of FreePSI over Cufflinks-A is magnified on genes or gene families that involve large proportions of multi-mapped reads.

CONCLUSION

In this paper, we presented an alignment-free approach, FreePSI, for estimating exon-inclusion ratios (or PSI values) without requiring the guidance of a reference transcriptome. FreePSI takes as its input a reference genome with exon boundary annotation and a set of RNA-seq reads, and produces the PSI values of all annotated exons. An abundance flow graph was introduced to represent all possible isoforms and their abundance levels. A novel probabilistic generative model was designed to allow for an alignment-free estimation of the parameters in the abundance flow graph. An efficient EM method based on a divide-and-conquer strategy was proposed to decompose a genome-wide MLE of the model into independent optimization subproblems for each gene. An ultrafast optimization algorithm, conjugate gradient projection descent, was implemented for solving these subproblems in parallel. Finally, a post-processing procedure was adopted to smooth out the estimated PSI values in each gene.

FreePSI is the first quantification method achieving transcriptome-free and alignment-free simultaneously in RNA-seq data analysis. As a result, it not only performs well when high quality reference transcriptomes are not present, but also runs efficiently and is able to deal with data involving a large proportion of multi-mapped reads. We expect that FreePSI will have important applications in the alternative splicing analysis for organisms that do not have well studied transcriptomes.

AVAILABILITY

The FreePSI algorithm is freely available under the GNU General Public License (GPLv3). A version of the source code has been deposited at https://github.com/JY-Zhou/FreePSI. The scripts for generating the simulated RNA-seq datasets analyzed in the paper can be found on the same GitHub page. The actual datasets and detailed experimental results are available from the corresponding author upon request.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

J.Z. and T.J. designed the model, algorithm and experiments. J.Z. implemented the program and conducted the experiments. D.W., S.M. and J.Z. (Jianyang) provided some useful advice. J.Z. and T.J. wrote the paper and all authors have reviewed the paper. We would like to thank Dr Rui Jiang (Tsinghua University) for the support of computational resources, and the anonymous referees for many constructive suggestions.

FUNDING

US National Science Foundation (in part) [DBI-1262107, IIS-1646333 to T.J.]; National Natural Science Foundation of China [61472205 to J.Z.]; Youth 1000-Talent Program of China (to J.Z.); Ningbo Science and Technology Discovery Grant [2014B82014 to T.J.]. Funding for open access charge: US National Science Foundation Grant [IIS-1646333].

Conflict of interest statement. None declared.

REFERENCES

1. Barash,Y., Calarco,J.A., Gao,W., Pan,Q., Wang,X., Shai,O., Blencowe,B.J. and Frey,B.J. (2010) Deciphering the splicing code. *Nature*, **465**, 53–59.
2. Kakaradov,B., Xiong,H.Y., Lee,L.J., Jojic,N. and Frey,B.J. (2012) Challenges in estimating percent inclusion of alternatively spliced junctions from RNA-seq data. *BMC Bioinformatics*, **13**, S11.
3. Saltzman,A.L., Pan,Q. and Blencowe,B.J. (2011) Regulation of alternative splicing by the core spliceosomal machinery. *Genes Dev.*, **25**, 373–384.
4. Ohta,S., Nishida,E., Yamanaka,S. and Yamamoto,T. (2013) Global splicing pattern reversion during somatic cell reprogramming. *Cell Rep.*, **5**, 357–366.
5. Venables,J.P., Klinck,R., Koh,C., Gervais-Bird,J., Bramard,A., Inkel,L., Durand,M., Couture,S., Froehlich,U., Lapointe,E. *et al.* (2009) Cancer-associated regulation of alternative splicing. *Nat. Struct. Mol. Biol.*, **16**, 670–676.
6. Barbosa-Morais,N.L., Irimia,M., Pan,Q., Xiong,H.Y., Gueroussov,S., Lee,L.J., Slobodeniuc,V., Kutter,C., Watt,S., Çolak,R. *et al.* (2012) The evolutionary landscape of alternative splicing in vertebrate species. *Science*, **338**, 1587–1593.
7. Wang,Z., Gerstein,M. and Snyder,M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, **10**, 57–63.

8. Garber,M., Grabherr,M.G., Guttman,M. and Trapnell,C. (2011) Computational methods for transcriptome annotation and quantification using RNA-seq. *Nat. Methods*, **8**, 469–477.
9. Conesa,A., Madrigal,P., Tarazona,S., Gomez-Cabrero,D., Cervera,A., McPherson,A., Szcześniak,M.W., Gaffney,D.J., Elo,L.L., Zhang,X. *et al.* (2016) A survey of best practices for RNA-seq data analysis. *Genome Biol.*, **17**, 13.
10. Alamancos,G.P., Pagès,A., Trincado,J.L., Bellora,N. and Eyras,E. (2015) Leveraging transcript quantification for fast computation of alternative splicing profiles. *RNA*, **21**, 1521–1531.
11. Trapnell,C., Williams,B.A., Pertea,G., Mortazavi,A., Kwan,G., Van Baren,M.J., Salzberg,S.L., Wold,B.J. and Pachter,L. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.*, **28**, 511–515.
12. Li,B. and Dewey,C.N. (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, **12**, 323.
13. Li,W. and Jiang,T. (2012) Transcriptome assembly and isoform expression level estimation from biased RNA-Seq reads. *Bioinformatics*, **28**, 2914–2921.
14. Roberts,A. and Pachter,L. (2013) Streaming fragment assignment for real-time analysis of sequencing experiments. *Nat. Methods*, **10**, 71–73.
15. Li,W., Feng,J. and Jiang,T. (2011) IsoLasso: a LASSO regression approach to RNA-Seq based transcriptome assembly. *J. Computat. Biol.*, **18**, 1693–1707.
16. Pertea,M., Pertea,G.M., Antonescu,C.M., Chang,T.-C., Mendell,J.T. and Salzberg,S.L. (2015) StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.*, **33**, 290–295.
17. Liu,J., Yu,T., Jiang,T. and Li,G. (2016) TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs. *Genome Biol.*, **17**, 213.
18. Langmead,B., Trapnell,C., Pop,M. and Salzberg,S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.
19. Trapnell,C., Pachter,L. and Salzberg,S.L. (2009) TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, **25**, 1105–1111.
20. Kim,D., Pertea,G., Trapnell,C., Pimentel,H., Kelley,R. and Salzberg,S.L. (2013) TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol.*, **14**, R36.
21. Kim,D., Langmead,B. and Salzberg,S.L. (2015) HISAT: a fast spliced aligner with low memory requirements. *Nat. Methods*, **12**, 357–360.
22. Patro,R., Mount,S.M. and Kingsford,C. (2014) Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.*, **32**, 462–464.
23. Marçais,G. and Kingsford,C. (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, **27**, 764–770.
24. Bray,N.L., Pimentel,H., Melsted,P. and Pachter,L. (2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**, 525–527.
25. Patro,R., Duggal,G., Love,M.I., Irizarry,R.A. and Kingsford,C. (2017) Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods*, **14**, 417–419.
26. Katz,Y., Wang,E.T., Airoldi,E.M. and Burge,C.B. (2010) Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nat. Methods*, **7**, 1009–1015.
27. Shen,S., Park,J.W., Huang,J., Dittmar,K.A., Lu,Z.-X., Zhou,Q., Carstens,R.P. and Xing,Y. (2012) MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data. *Nucleic Acids Res.*, **40**, e61.
28. Shen,S., Park,J.W., Lu,Z.-X., Lin,L., Henry,M.D., Wu,Y.N., Zhou,Q. and Xing,Y. (2014) rMATS: robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proc. Natl. Acad. Sci. U.S.A.*, **111**, E5593–E5601.
29. Heber,S., Alekseyev,M., Sze,S.-H., Tang,H. and Pevzner,P.A. (2002) Splicing graphs and EST assembly problem. *Bioinformatics*, **18**(Suppl. 1), S181–S188.
30. Goldfarb,D. (1969) Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints. *SIAM J. Appl. Math.*, **17**, 739–764.
31. Griebel,T., Zacher,B., Ribeca,P., Raineri,E., Lacroix,V., Guigó,R. and Sammeth,M. (2012) Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res.*, **40**, 10073–10083.
32. NA,J. and JN,F. (2011) Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ files. https://github.com/najoshi/sickle.