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

Supplementary information

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

1 MOE Key Laboratory of Bioinformatics; Bioinformatics Division and Center for Synthetic & Systems Biology, TNLIST, Tsinghua University

2 Department of Computer Science and Technology, Tsinghua University

3 Department of Statistics, Stanford University

4 Institute for Interdisciplinary Information Sciences, Tsinghua University

5 Department of Computer Science and Engineering, University of California, Riverside

## Contents

# S1 Abundance flow graph

## S1.1 Formal definitions of $\alpha$ and PSI

Let $\alpha_h$ denote the relative abundance of isoform $h$ of a gene. For convenience, denote the junction segment formed by exon segments $i$ and $j$ as a pair $(i, j)$, where $i \neq j$. For simplicity, we will also denote an exon segment $i$ as the identity pair $(i, i)$. All such pairs will uniformly be referred to as segments. The indicator variable $\mathcal{I}_{hij}$ is defined as

$$\mathcal{I}_{hij} = \begin{cases} 1 & \text{if isoform } h \text{ covers segment } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

where the segment $(i, j)$ represents an exon segment if $i = j$ or a junction segment otherwise. The parameter $\alpha_{ij}$ for segment $(i, j)$ is formally defined as

$$\alpha_{ij} = \sum_h \mathcal{I}_{hij}\alpha_h \tag{S1.1}$$

Note that isoforms will not be explicitly dealt with in our method (*i.e.*, they will be marginalized out), but they are used here to help explain the construction of our model, and will be referred to frequently below.

PSI is defined as the ratio of the total relative abundance of all isoforms containing a given exon over the total relative abundance of all isoforms of the gene containing the exon. Hence, the PSI value of exon segment $i$ can be expressed by the following equation:

$$\psi_i = \frac{\sum_h \mathcal{I}_{hii}\alpha_h}{\sum_h \alpha_h} = \frac{\alpha_{ii}}{\sum_h \alpha_h} \tag{S1.2}$$

## S1.2 Properties of the abundance flow graph

In order to consider the first and last exon segments of an isoform, we extend the above indicator variables as follows:

$$\mathcal{I}_{hsi} = \begin{cases} 1 & \text{if exon segment } i \text{ is the first exon segment of isoform } h \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{I}_{hit} = \begin{cases} 1 & \text{if exon segment } i \text{ is the last exon segment of isoform } h \\ 0 & \text{otherwise} \end{cases}$$

Then, the edge weights $\alpha_{si}$ and $\alpha_{it}$ in an abundance flow graph can be defined formally as

$$\alpha_{si} = \sum_h \mathcal{I}_{hsi}\alpha_h, \qquad \alpha_{it} = \sum_h \mathcal{I}_{hit}\alpha_h \tag{S1.3}$$

**Theorem S1.** *The following equalities hold:*

$$\sum_i \alpha_{si} = \sum_h \alpha_h = \sum_i \alpha_{it} \tag{S1.4}$$

*Proof.* Since every isoform has a unique first exon segment, we have the following equality:

$$\sum_i \mathcal{I}_{hsi} = 1$$

Hence,

$$\begin{aligned} \sum_i \alpha_{si} &= \sum_i \sum_h \mathcal{I}_{hsi}\alpha_h \\ &= \sum_h \alpha_h \sum_i \mathcal{I}_{hsi} \\ &= \sum_h \alpha_h \end{aligned}$$

The second equality can be derived similarly. □

The abundance flow graph satisfies the flow conservation property, if we consider the edge weights as flow amounts. Figure S1 illustrates an example of this flow conservation property.

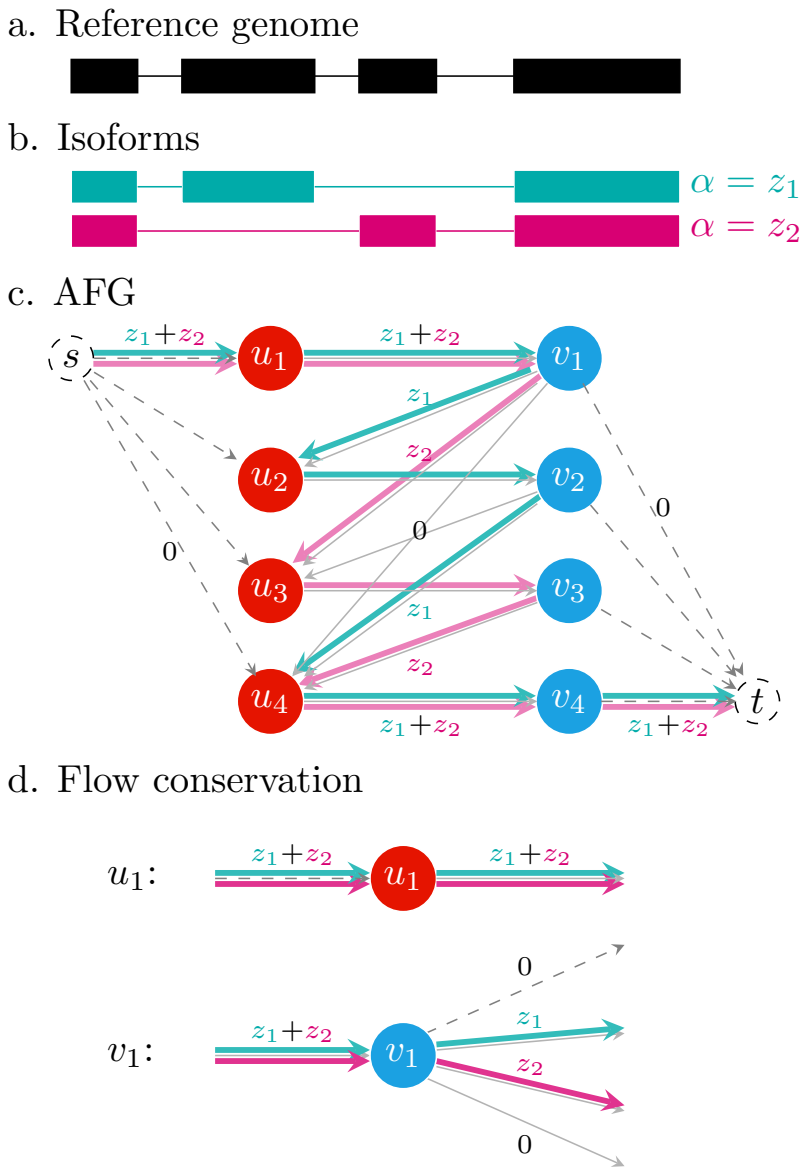



Figure S1: This figure illustrates the flow conservation property of an example AFG for a gene consisting of four exon segments and two isoforms. The exon boundaries are shown in (a) and the relative abundance of the two isoforms ($z_1$ and $z_2$) are shown in (b). The resulting AFG is shown in (c). The two isoforms are highlighted as two paths with colors cyan and pink from $s$ to $t$, respectively. Considering the edge weights as flow amounts, the flow conservation property clearly holds for all vertices in $U$ and $V$. In particular, Figure (d) shows that the flow conservation property holds for vertices $u_1$ and $v_1$.

The following theorem provides a formal derivation of the flow conservation property.

**Theorem S2.**

$$\alpha_{ii} = \alpha_{si} + \sum_{j<i} \alpha_{ji}, \quad \textit{for all } i \tag{S1.5}$$

$$\alpha_{ii} = \alpha_{it} + \sum_{j>i} \alpha_{ij}, \quad \textit{for all } i \tag{S1.6}$$

*Proof.* We first prove that equality S1.5 holds. For any given isoform, an exon segment $i$ of the isoform is either its first exon segment or next to exon segment $j$ for some $j < i$. Therefore, for each isoform $h$,

$$\mathcal{I}_{hii} = \mathcal{I}_{hsi} + \sum_{j<i} \mathcal{I}_{hji} \tag{S1.7}$$

Multiplying both sides of Equation S1.7 by $\alpha_h$ and summing up the equations for all isoforms $h$, we get

$$\begin{aligned}
& \sum_h \mathcal{I}_{hii}\alpha_h = \sum_h \left( \mathcal{I}_{hsi} + \sum_{j<i} \mathcal{I}_{hji} \right) \alpha_h \\
\Rightarrow \quad & \sum_h \mathcal{I}_{hii}\alpha_h = \sum_h \mathcal{I}_{hsi}\alpha_h + \sum_{j<i} \sum_h \mathcal{I}_{hji}\alpha_h \\
\Rightarrow \quad & \alpha_{ii} = \alpha_{si} + \sum_{j<i} \alpha_{ji} \qquad \text{(by Equations S1.1 and S1.3)}
\end{aligned}$$

Hence, equality S1.5 holds. Equality S1.6 can be proven similarly. □

**Corollary S1.** *The PSI values can be rewritten as*

$$\psi_i = \frac{\alpha_{ii}}{\sum_i \alpha_{ii} - \sum_i \sum_{j>i} \alpha_{ij}} \tag{S1.8}$$

*Proof.* By Equation S1.2, we only need to focus on the denominator.

$$\begin{aligned}
\sum_h \alpha_h &= \sum_i \alpha_{si} && \text{(by Equation S1.4)} \\
&= \sum_i \left( \alpha_{ii} - \sum_{j<i} \alpha_{ji} \right) && \text{(by Equation S1.5)} \\
&= \sum_i \alpha_{ii} - \sum_i \sum_{j<i} \alpha_{ji} \\
&= \sum_i \alpha_{ii} - \sum_i \sum_{j>i} \alpha_{ij}
\end{aligned}$$

□

Since $\alpha_{si} \geq 0$ and $\alpha_{it} \geq 0$ for all $i$, we have the following constraints for the parameters $\alpha_{ij}$:

**Corollary S2.**

$$\alpha_{ii} \geq \sum_{j<i} \alpha_{ji}, \quad \textit{for all } i \tag{S1.9}$$

$$\alpha_{ii} \geq \sum_{j>i} \alpha_{ij}, \quad \textit{for all } i \tag{S1.10}$$

These constraints are crucial for an accurate estimation of our probabilistic model.

# S2 Probabilistic generative model

## S2.1 Subscript rearrangement

In order to perform a genome-wide analysis, we use a single index $s$ to indicate a segment (exon or junction) of any gene $g$, and $\alpha_{gs}$ to denote the total relative abundance of all isoforms containing segment $s$ of gene $g$. Hence, the PSI value of exon segment $i$ of gene $g$ can be rewritten as

$$\psi_{gi} = \frac{\alpha_{gi}}{\underset{\substack{s\in g \\ s \text{ is an exon segment}}}{\sum} \alpha_{gs} - \underset{\substack{s\in g \\ s \text{ is a junction segment}}}{\sum} \alpha_{gs}} \tag{S2.1}$$

## S2.2 Random variables and parameters

Three random variables are defined for the probabilistic generative model.

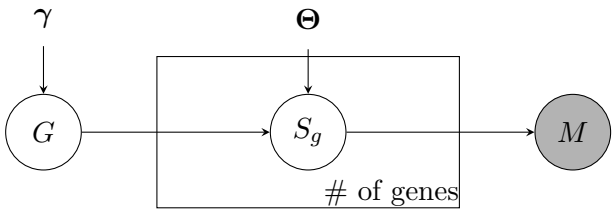
1. $G$ represents a random gene. Thus, $\mathrm{P}(G = g) = \gamma_g$ indicates that the probability that a read is generated from gene $g$ is $\gamma_g$.

2. $S_g$ represents a random segment of gene $g$, and $\mathrm{P}(S_g = s|G = g) = \theta_{gs}$ indicates that each read from gene $g$ is generated from segment $s$ with conditional probability $\theta_{gs}$.

3. $M$ represents a random k-mer, and $\mathrm{P}(M = m)$ indicates the probability of k-mer $m$ being observed.

## S2.3 Graphical model

The probabilistic generative model can be represented by the following graphical model:




The probability of observing a k-mer $m$ can be calculated as

$$\begin{aligned}
&\mathrm{P}(M = m) \\
&= \sum_g \sum_{s\in g} \mathrm{P}(M = m, S_g = s, G = g) \\
&= \sum_g \sum_{s\in g} \mathrm{P}(M = m|S_g = s, G = g)\,\mathrm{P}(S_g = s|G = g)\,\mathrm{P}(G = g) \\
&= \sum_g \mathrm{P}(G = g) \sum_{s\in g} \mathrm{P}(S_g = s|G = g)\,\mathrm{P}(M = m|S_g = s, G = g) \\
&= \sum_g \gamma_g \sum_{s\in g} \theta_{gs} \mathrm{P}(M = m|S_g = s, G = g)
\end{aligned}$$

where $\mathrm{P}(M = m|S_g = s, G = g)$ denotes the theoretical distribution of k-mers on segment $s$.

## S2.4 Relationship between $\gamma$, $\theta$ and $\alpha$

Let $L_{gs}$ denote the length of segment $s$ of gene $g$ and $L_{gh}$ denote the length of isoform $h$ of gene $g$. The number of possible starting sites for a read in the segment or isoform are

$$\begin{aligned} \widetilde{L}_{gs} &= L_{gs} - L_{\text{read}} + 1 \\ \widetilde{L}_{gh} &= L_{gh} - L_{\text{read}} + 1 \end{aligned}$$

respectively, where $L_{\text{read}}$ denotes the read length. If the relative abundance is measured by TPM, $\alpha_{gh}$ can be calculated as

$$\alpha_{gh} = \frac{\frac{f_{gh}}{\widetilde{L}_{gh}}}{\sum_g \sum_{h' \in g} \frac{f_{gh'}}{\widetilde{L}_{gh'}} \times 10^{-6}} \tag{S2.2}$$

where $f_{gh}$ represents the number of reads from isoform $h$ of gene $g$. Similarly, let $f_{gs}$ denote the number of reads from segment $s$ of gene $g$, and the indicator variable $\mathcal{I}_{ghs}$ to indicate whether isoform $h$ contains segment $s$ of gene $g$. Assuming that the reads are uniformly distributed in each segment/isoform, we get

$$f_{gs} = \sum_{h \in g} \mathcal{I}_{ghs} f_{gh} \frac{\widetilde{L}_{gs}}{\widetilde{L}_{gh}} \tag{S2.3}$$

Note that the reads are generated by an implicit two-level process in the above probabilistic model: from genes to isoforms and then to segments. Equation S2.3 marginalizes out isoforms to associate the reads with segments directly.

Since the reads are assumed to be generated from the probabilistic model, we have

$$\gamma_g \theta_{gs} \approx \frac{f_{gs}}{\sum_g \sum_{h \in g} f_{gh}} \tag{S2.4}$$

The following proposition shows a relationship among $\gamma$, $\theta$ and $\alpha$.

**Proposition S1.**

$$\alpha_{gs} \approx \frac{Z_2}{Z_1} \frac{\gamma_g \theta_{gs}}{\widetilde{L}_{gs}} \tag{S2.5}$$

*where* $Z_1 = \sum_g \sum_{h \in g} \frac{f_{gh}}{\widetilde{L}_{gh}} \times 10^{-6}$ *and* $Z_2 = \sum_g \sum_{h \in g} f_{gh}$.

*Proof.*

$$\begin{aligned} \gamma_g \theta_{gs} &\approx \frac{f_{gs}}{Z_2} && \text{(by Equation S2.4)} \\ &= \frac{1}{Z_2} \sum_{h \in g} \mathcal{I}_{ghs} f_{gh} \frac{\widetilde{L}_{gs}}{\widetilde{L}_{gh}} && \text{(by Equation S2.3)} \\ &= \frac{\widetilde{L}_{gs}}{Z_2} \sum_{h \in g} \mathcal{I}_{ghs} \frac{f_{gh}}{\widetilde{L}_{gh}} \\ &= \frac{\widetilde{L}_{gs}}{Z_2} \sum_{h \in g} \mathcal{I}_{ghs} \alpha_{gh} Z_1 && \text{(by Equation S2.2)} \\ &= \frac{Z_1}{Z_2} \widetilde{L}_{gs} \sum_{h \in g} \mathcal{I}_{ghs} \alpha_{gh} \\ &= \frac{Z_1}{Z_2} \widetilde{L}_{gs} \alpha_{gs} && \text{(by Equation S1.1)} \end{aligned}$$

□

Therefore, the definition of PSI given in Equation S2.1 can be rewritten in terms of $\theta$ as follows:

$$\psi_{gi} \approx \frac{\frac{\theta_{gi}}{\widetilde{L}_{gi}}}{\sum\limits_{\substack{s\in g \\ s \text{ is an exon segment}}} \frac{\theta_{gs}}{\widetilde{L}_{gs}} - \sum\limits_{\substack{s\in g \\ s \text{ is a junction segment}}} \frac{\theta_{gs}}{\widetilde{L}_{gs}}} \tag{S2.6}$$

The linear inequalities S1.9 and S1.10 can also be transformed into the constraints on $\theta$. For gene $g$, the parameter $\theta$ should satisfy $\boldsymbol{A}_g\boldsymbol{\theta}_g \geq 0$, which is the matrix-form of the constraints. More specifically, $\boldsymbol{A}_g$ is composed of three parts:

$$\boldsymbol{A}_g = \begin{pmatrix} \boldsymbol{D}_g \\ \boldsymbol{U}_g \\ \boldsymbol{B}_g \end{pmatrix} \tag{S2.7}$$

Each row of $\boldsymbol{D}_g \in \mathbb{R}^{N_e(g)\times N_s(g)}$ denotes the coefficients of the linear constraints between each exon segment and its downstream junction segments (*i.e.*, the inequality in Equation S1.10). $N_e(g)$ denotes the number of exon segments in gene $g$ and $N_s(g)$ the number of (exon and junction) segments in gene $g$. The element at row $i$ and column $s$ is defined as

$$\boldsymbol{D}_g(i,s) = \begin{cases} \frac{1}{\widetilde{L}_{gs}} & \text{if } s \text{ is an exon segment } i \\ -\frac{1}{\widetilde{L}_{gs}} & \text{if } s \text{ is a junction segment beginning at exon segment } i \\ 0 & \text{otherwise} \end{cases} \tag{S2.8}$$

Similarly, each row of $\boldsymbol{U}_g \in \mathbb{R}^{N_e(g)\times N_s(g)}$ denotes the coefficients of the linear constraints between each exon segment and its upstream junction segments (*i.e.*, the inequality in Equation S1.9). Its elements are defined as

$$\boldsymbol{U}_g(i,s) = \begin{cases} \frac{1}{\widetilde{L}_{gs}} & \text{if } s \text{ is exon segment } i \\ -\frac{1}{\widetilde{L}_{gs}} & \text{if } s \text{ is a junction segment ending at exon segment } i \\ 0 & \text{otherwise} \end{cases} \tag{S2.9}$$

$\boldsymbol{B}_g \in \mathbb{R}^{(N_s(g)-N_e(g))\times N_s(g)}$ represents the non-negativity constraints on the relative abundance of junction segments, which is defined as

$$\begin{aligned} & \boldsymbol{B}_g = (\boldsymbol{0}, \boldsymbol{I}) \\ \text{where} \quad & \boldsymbol{0} \in \mathbb{R}^{(N_s(g)-N_e(g))\times N_e(g)} \\ & \boldsymbol{I} \in \mathbb{R}^{(N_s(g)-N_e(g))\times(N_s(g)-N_e(g))} \end{aligned} \tag{S2.10}$$

The non-negativity constraints on the relative abundance of exon segments are omitted here since they are already implied by the constraints in $\boldsymbol{D_g}, \boldsymbol{U_g}$ and $\boldsymbol{B_g}$.

## S2.5 Theoretical distribution of k-mers

Let $r$ represent a read and $m_r$ represent a k-mer in $r$. Define the following indicator function:

$$\mathcal{I}(m, m_r) = \begin{cases} 1 & \text{if } m = m_r \\ 0 & \text{if } m \neq m_r \end{cases}$$

Recall $c_{gsm} = \mathrm{P}(M = m | S_g = s, G = g)$ denotes the theoretical distribution of k-mers on segment $s$ of gene $g$, under the assumption that the reads are uniformly distributed on each segment. The following proposition is easy to prove.

**Proposition S2.**

$$c_{gsm} := P(M = m|S_g = s, G = g) = \frac{\mathcal{F}_{gsm}}{\widetilde{L}_{gs}\left(L_{read} - K + 1\right)} \tag{S2.11}$$

*where*

$$\mathcal{F}_{gsm} = \sum_{r \in s} \sum_{m_r} \mathcal{I}\left(m_r, m\right) \tag{S2.12}$$

Finally, the probability of observing a k-mer $m$ is

$$\mathrm{P}\left(M = m\right) = \sum_{g} \gamma_g \sum_{s \in g} \theta_{gs} c_{gsm} \tag{S2.13}$$

where $\gamma_g$ and $\theta_{gs}$ are model parameters to be estimated.

# S3 Algorithms

## S3.1 Maximum likelihood estimation

Let $n_m$ denote the number of occurrences of k-mer $m$ in the input RNA-seq reads. The likelihood of observing all k-mers in the input is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\Theta}) &= \prod_m \mathrm{P}(M = m)^{n_m} \\ \log \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\Theta}) &= \sum_m n_m \log \mathrm{P}(M = m) \\ &= \sum_m n_m \log \left( \sum_g \gamma_g \sum_{s \in g} \theta_{gs} c_{gsm} \right) \end{aligned}$$

The maximum likelihood estimation is to solve the following nonlinear constrained optimization:

$$\begin{aligned} \max \quad & \log \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\Theta}) \\ \text{s.t.} \quad & \boldsymbol{A}_g \boldsymbol{\theta}_g \geq 0, \quad \text{for all gene } g \\ & \sum_{s \in g} \theta_{gs} = 1, \quad \text{for all gene } g \\ & \sum_g \gamma_g = 1, \quad \forall \gamma_g \geq 0, \quad \forall \theta_{gs} \geq 0 \end{aligned}$$

## S3.2 The expectation-maximization algorithm

An initial feasible solution is obtained via the following algorithm.

**Algorithm S1:** Initial feasible solution construction

**Input:** $c_{gsm}$, $n_m$ and $\boldsymbol{A}_g$ (for all $g, s$ and $m$)
**Output:** $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$

**begin**
  $Z_m \leftarrow \sum_g \sum_{s \in g} c_{gsm}$, $\theta_{gs} \leftarrow \sum_m n_m \frac{c_{gsm}}{Z_m}$, $\gamma_g \leftarrow \sum_{s \in g} \theta_{gs}$ ▷ distribute $n_m$ to segments
  $Y \leftarrow \sum_g \gamma_g$, $\gamma_g \leftarrow \frac{\gamma_g}{Y}$ ▷ normalize $\gamma$
  **forall** $g$ **do**
    **forall** *row vector* $\boldsymbol{a}_r$ *in* $\boldsymbol{A}_g$ *with* $\boldsymbol{a}_r \boldsymbol{\theta}_g < 0$ **do**
      **if** $\theta_{gs} = 0$ *for some exon segment* $s$ **then**
        set all $\theta_{gs'} \leftarrow 0$ for all adjacent junction segments $s'$ ▷ make $\theta$ feasible
      **end**
      **while** $\boldsymbol{a}_r \boldsymbol{\theta}_g < 0$ **do**
        $\theta_{gs} \leftarrow \theta_{gs} \times 10$ for all exon segments $s$ ▷ make $\theta$ feasible
      **end**
      $X_g \leftarrow \sum_{s \in g} \theta_{gs}$, $\theta_{gs} \leftarrow \frac{\theta_{gs}}{X_g}$ ▷ normalize $\theta$
    **end**
  **end**
  **return** $\boldsymbol{\theta}^{(0)}$ *and* $\boldsymbol{\gamma}^{(0)}$
**end**

In the E-step of the EM algorithm, we derive the expected log-likelihood using the current

estimation of $\gamma_g^{(t)}$ and $\theta_{gs}^{(t)}$ as follows:

$$\mathcal{Q}\left(\boldsymbol{\gamma}, \boldsymbol{\Theta}\right) = \sum_m n_m \sum_g \mu_{gm}^{(t)} \log\left(\gamma_g \sum_{s \in g} c_{gsm} \theta_{gs}\right) \tag{S3.1}$$

where

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

By expanding the product in the logarithmic term of Equation S3.1, $\mathcal{Q}\left(\boldsymbol{\gamma}, \boldsymbol{\Theta}\right)$ can be decomposed as the summation of two independent parts:

$$\mathcal{Q}\left(\boldsymbol{\gamma}, \boldsymbol{\Theta}\right) \;=\; \mathcal{Q}^{\mathrm{I}}\left(\boldsymbol{\gamma}\right) + \sum_g \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right) \tag{S3.2}$$

where

$$\begin{aligned} \mathcal{Q}^{\mathrm{I}}\left(\boldsymbol{\gamma}\right) &= \sum_m \sum_g \mu_{gm}^{(t)} \log\left(\gamma_g\right) \\ \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right) &= \sum_m \mu_{gm}^{(t)} \log\left(\sum_{s \in g} \theta_{gs} c_{gsm}\right) \end{aligned}$$

The M-step of the algorithm is to maximize the expectation of the log-likelihood given in Equation S3.2. This is divided into two independent parts. The first part is to solve

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

By using the Lagrangian multiplier method, a closed-form solution for this part can be derived:

$$\gamma_g^{(t+1)} = \frac{\sum\limits_m \mu_{gm}^{(t)}}{\sum\limits_m \sum\limits_g \mu_{gm}^{(t)}}$$

The second part consists of a similar optimization problem for each gene $g$:

$$\begin{aligned} \max \quad & \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right) \\ \text{s.t.} \quad & \boldsymbol{A}_g \boldsymbol{\theta}_g \geq 0, \quad \sum_{s \in g} \theta_{gs} = 1, \quad \forall \theta_{gs} \geq 0 \end{aligned} \tag{S3.3}$$

Since a closed-form solution for this problem is unavailable due to the linear inequality constraints, the conjugate gradient projection descent (CGPD) algorithm is applied to solve the problem for all genes concurrently.

### S3.3 The conjugate gradient projection descent algorithm

The CGPD algorithm is an extension of the well-known gradient projection descent (GPD) algorithm [1]. For completeness, a pseudocode of the CGPD algorithm is given below.

**Algorithm S2:** The CGPD algorithm

**Input:** $\boldsymbol{\theta}_g$, $\boldsymbol{A}_g$

**Output:** $\arg\max \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{\theta}_g\right)$ s.t. $\boldsymbol{A}_g\boldsymbol{\theta}_g \geq 0,\ \sum_{s\in g}\theta_{gs}=1,\ \forall\theta_{gs}\geq 0$

**begin**
  $i \leftarrow 0$ , $\boldsymbol{x}^{(i)} \leftarrow \boldsymbol{\theta}_g$ , $\boldsymbol{g}^{(i)} \leftarrow \nabla\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{x}^{(i)}\right)$
  $q \leftarrow 1$ , $\boldsymbol{N}_q \leftarrow (1,1,\cdots,1)^\top$
  **forall** *row vector* $\boldsymbol{a}_j$ *in* $\boldsymbol{A}_g$ **do**
    **if** $\boldsymbol{a}_j\boldsymbol{x}^{(i)}=0$ **then**
      $\boldsymbol{N}_{q+1} \leftarrow \left(\boldsymbol{N}_q, \boldsymbol{a}_j^\top\right)$
      $q \leftarrow q+1$
    **end**
  **end**
  $\boldsymbol{H}_q^{(i)} \leftarrow \boldsymbol{I}^{q\times q}$
  **repeat**
    $\boldsymbol{s}^{(i)} = \boldsymbol{H}_q^{(i)}\boldsymbol{g}^{(i)}$ ▷ construct conjugate gradient direction
    $\boldsymbol{\alpha} \leftarrow \left(\boldsymbol{N}_q^\top\boldsymbol{N}_q\right)^{-1}\boldsymbol{N}_q^\top\boldsymbol{g}^{(i)}$ ▷ construct projected gradient direction
    $b_{jj} \leftarrow \left(\boldsymbol{N}_q^\top\boldsymbol{N}_q\right)^{-1}_{jj}$
    **if** $\boldsymbol{s}^{(i)}=\boldsymbol{0}$ *and* $\forall\alpha_j\leq 0$ **then**
      **return** $\boldsymbol{x}^{(i)}$
    **else if** $\left\|\boldsymbol{s}^{(i)}\right\| \leq \max\left\{\frac{1}{2}\alpha_j b_{jj}^{-1/2}\right\}$ **then**
      update $\boldsymbol{N}_q$ as $\boldsymbol{N}_{q-1}$ ▷ deactivate a constraint
      update $\boldsymbol{H}_q^{(i)}$ as $\boldsymbol{H}_{q-1}^{(i)}$
      $q \leftarrow q-1$
    **else**
      $\lambda^{(i)} \leftarrow \arg\max \mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{x}^{(i)}+\lambda\boldsymbol{s}^{(i)}\right)$ s.t. $0\leq\lambda\leq\lambda_{\mathrm{bound}}$ ▷ perform line search
      **if** $\lambda^{(i)}=0$ **then**
        **return** $\boldsymbol{x}^{(i)}$
      **else if** $\lambda^{(i)}=\lambda_{bound}$ **then**
        update $\boldsymbol{N}_q$ as $\boldsymbol{N}_{q+1}$ ▷ activate a constraint
        update $\boldsymbol{H}_q^{(i)}$ as $\boldsymbol{H}_{q+1}^{(i+1)}$
        $q \leftarrow q+1$
      **else**
        update $\boldsymbol{H}_q^{(i)}$ as $\boldsymbol{H}_q^{(i+1)}$ ▷ keep the constraints
      **end**
      $\boldsymbol{x}^{(i+1)} \leftarrow \boldsymbol{x}^{(i)}+\lambda^{(i)}\boldsymbol{s}^{(i)}$ , $\boldsymbol{g}^{(i+1)} \leftarrow \nabla\mathcal{Q}_g^{\mathrm{II}}\left(\boldsymbol{x}^{(i+1)}\right)$
      $i \leftarrow i+1$
    **end**
  **end**
**end**

# S4 Some implementation details

## S4.1 Linear indexing algorithm

**Algorithm S3:** Linear indexing algorithm

**Input:** sequence $S$ with length $L$
**Output:** indices of all k-mers in $S$

**begin**
  $a \leftarrow 0$
  **for** $i = 1, \cdots, K-1$ **do**
    $a \leftarrow (a \text{ lsh } 2) \text{ or } H(S_i)$
  **end**
  $mask \leftarrow (1 \text{ lsh } K) - 1$
  $A \leftarrow \{\}$
  **for** $i = K, \cdots, L$ **do**
    $a \leftarrow (a \text{ lsh } 2) \text{ or } H(S_i)$
    $a \leftarrow a \text{ and } mask$
    $A \leftarrow A \cup \{a\}$
  **end**
  **return** $A$
**end**
**Input:** base pair $s$
**Output:** index of $s$
**function** $H$
  **switch** $s$ **do**
    **case** *'A'* **do return** 0
    **case** *'C'* **do return** 1
    **case** *'G'* **do return** 2
    **case** *'T'* **do return** 3
  **end**
**end**

## S4.2 Techniques for improving the efficiency of CGPD

### S4.2.1 Offline computation for part of $\nabla \mathcal{Q}_g^{\mathrm{II}}(\boldsymbol{\theta}_g)$

$\nabla \mathcal{Q}_g^{\mathrm{II}}(\boldsymbol{\theta}_g)$ can be represented by

$$\nabla \mathcal{Q}_g^{\mathrm{II}}(\boldsymbol{\theta}_g) = \mathrm{diag}^{-1}\left(\boldsymbol{C}_g \boldsymbol{C}_g^{\top} \boldsymbol{\theta}_g\right)(\boldsymbol{C}_g \boldsymbol{\mu}_g) \tag{S4.1}$$

where $\boldsymbol{C}_g \in \mathbb{R}^{n_s(g) \times n_k}$ is the matrix form of $c_{gsm}$ and $\boldsymbol{\mu}_g \in \mathbb{R}^{n_k \times 1}$ is the vector form of $\mu_{gm}$, with $n_k$ denoting the number of k-mers and $n_s(g)$ the number of segments in gene $g$. Assuming that the CGPD algorithm converges in $T$ iterations, its time complexity is then $O\left(T n_k n_s(g)^2\right)$, if $\nabla \mathcal{Q}_g^{\mathrm{II}}(\boldsymbol{\theta}_g)$ is computed directly according to Equation S4.1. Since only $\boldsymbol{\theta}_g$ is changed during the iterations, $\boldsymbol{C}_g \boldsymbol{C}_g^{\top}$ and $\boldsymbol{C}_g \boldsymbol{\mu}_g$ (the iteration-invariant parts) can be computed in advance. This way, the time complexity is reduced into $O\left(n_k n_s(g)^2 + T n_s(g)^2\right)$.

### S4.2.2 Replacing outer product of vectors

The CGPD algorithm performs many vector outer product operation in the following form:

$$\boldsymbol{H}^{(t+1)} = \boldsymbol{H}^{(t)} \pm \boldsymbol{q}\boldsymbol{q}^{\top}$$

where $\boldsymbol{H} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{q} \in \mathbb{R}^{n\times 1}$. A direct computation requires allocating $n \times n$ new memory to store the matrix $\boldsymbol{q}\boldsymbol{q}^{\top}$, which is redundant and becomes an efficiency bottleneck of FreePSI. To speed up this frequent operation, we update $\boldsymbol{H}$ by in-space column-wise operations as follows:

$$\boldsymbol{H}_{\cdot i}^{(t+1)} \leftarrow \boldsymbol{H}_{\cdot i}^{(t)} \pm q_i \boldsymbol{q} \qquad \text{for } i = 1, \cdots, n$$

which does not require temporary memory allocation.

# S5 Supplementary results and discussion

## S5.1 Simulated data evaluation

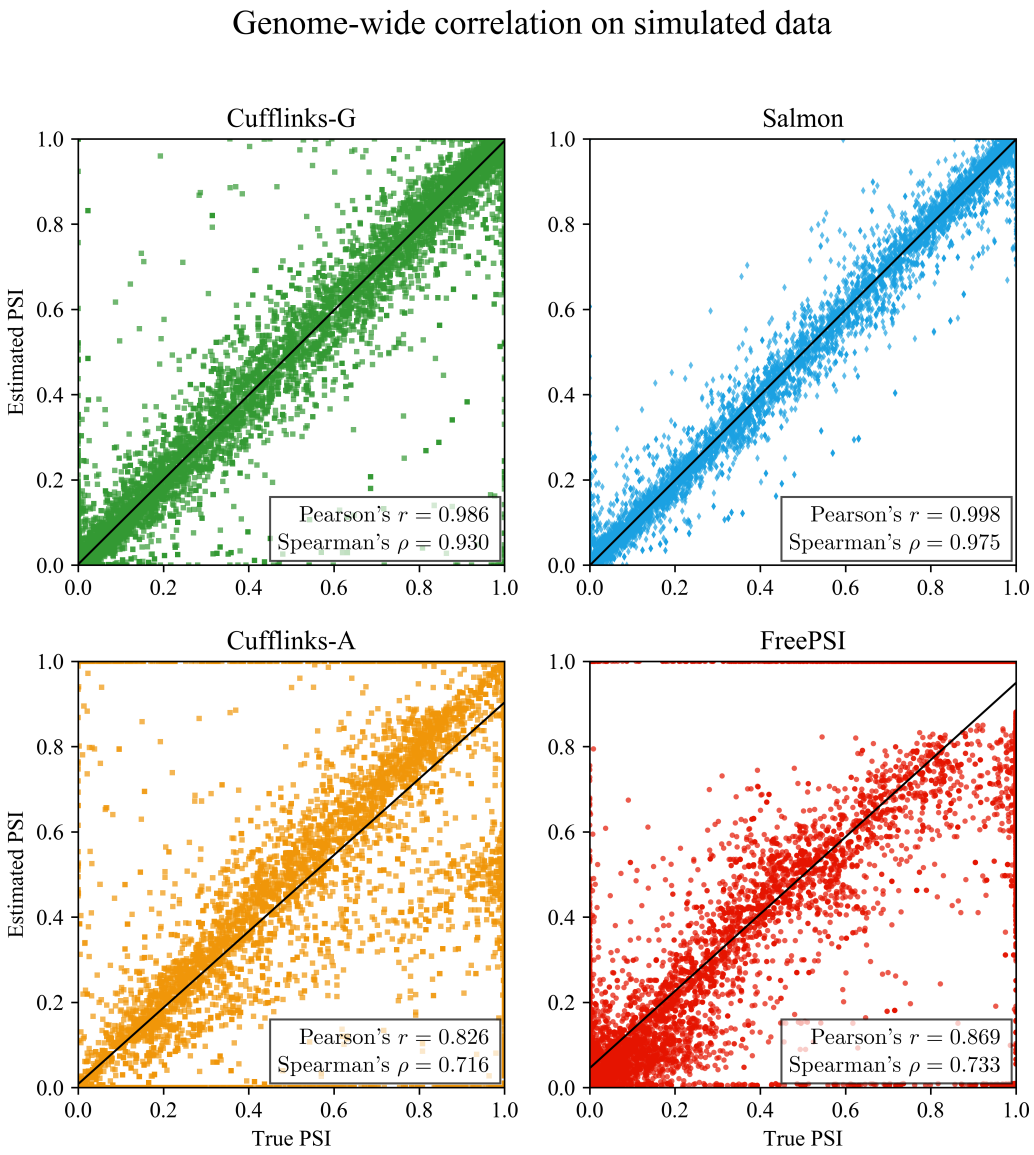



Figure S2: The scatter plot for genome-wide evaluation of different methods on the simulated data. The X-axis shows the true PSI values in the simulation and the Y-axis the PSI values estimated by different methods.

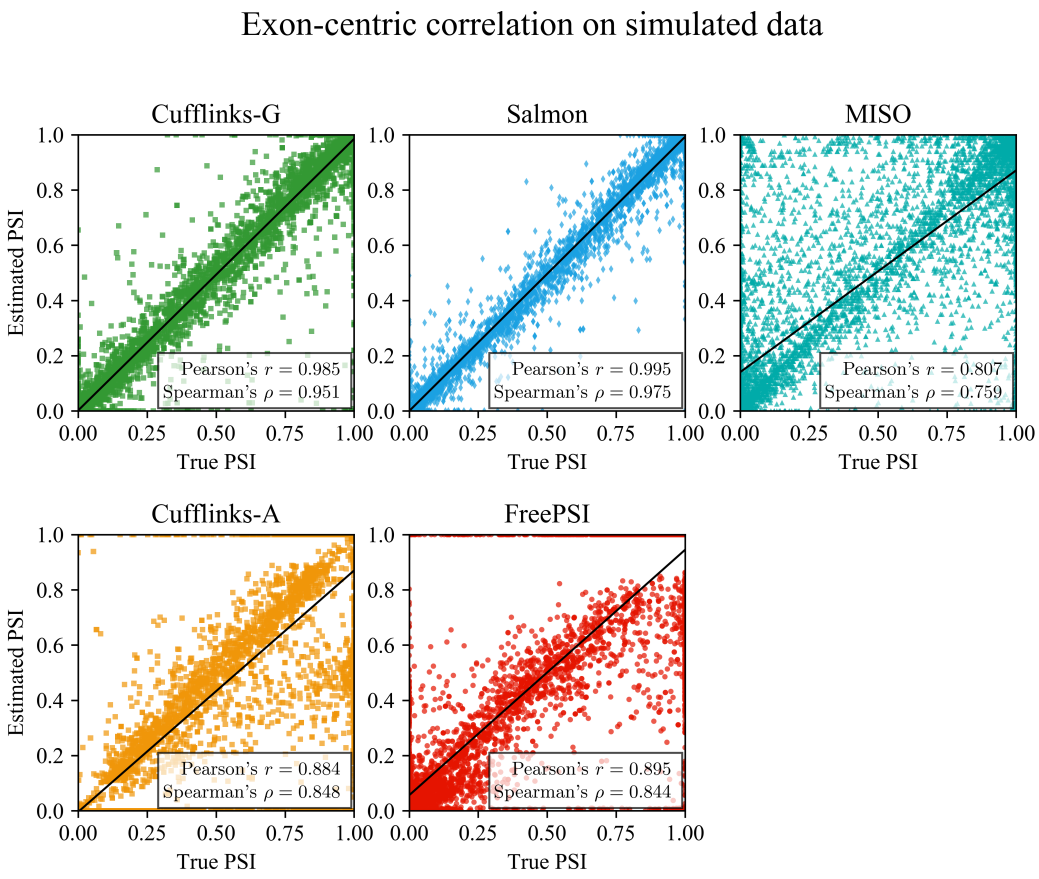



Figure S3: The scatter plot for exon-centric evaluation of different methods on the simulated data. The X-axis shows the true PSI values in the simulation and the Y-axis the PSI values estimated by different methods.

## S5.2 Real data evaluation

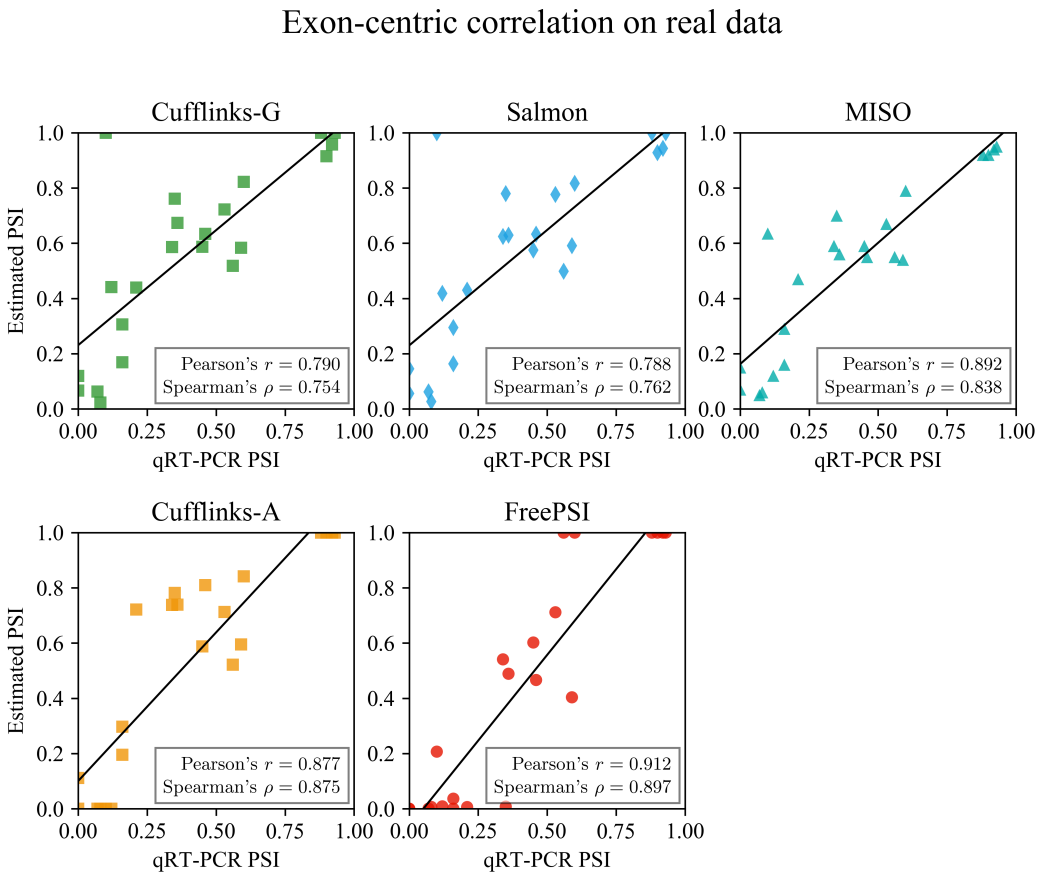



Figure S4: The scatter plot for exon-centric evaluation of different methods on the real data. The X-axis shows the true PSI values calculated from the qRT-PCR PSI results and the Y-axis the PSI values estimated by different methods.

## S5.3 Impact of the quality of transcriptome assembly

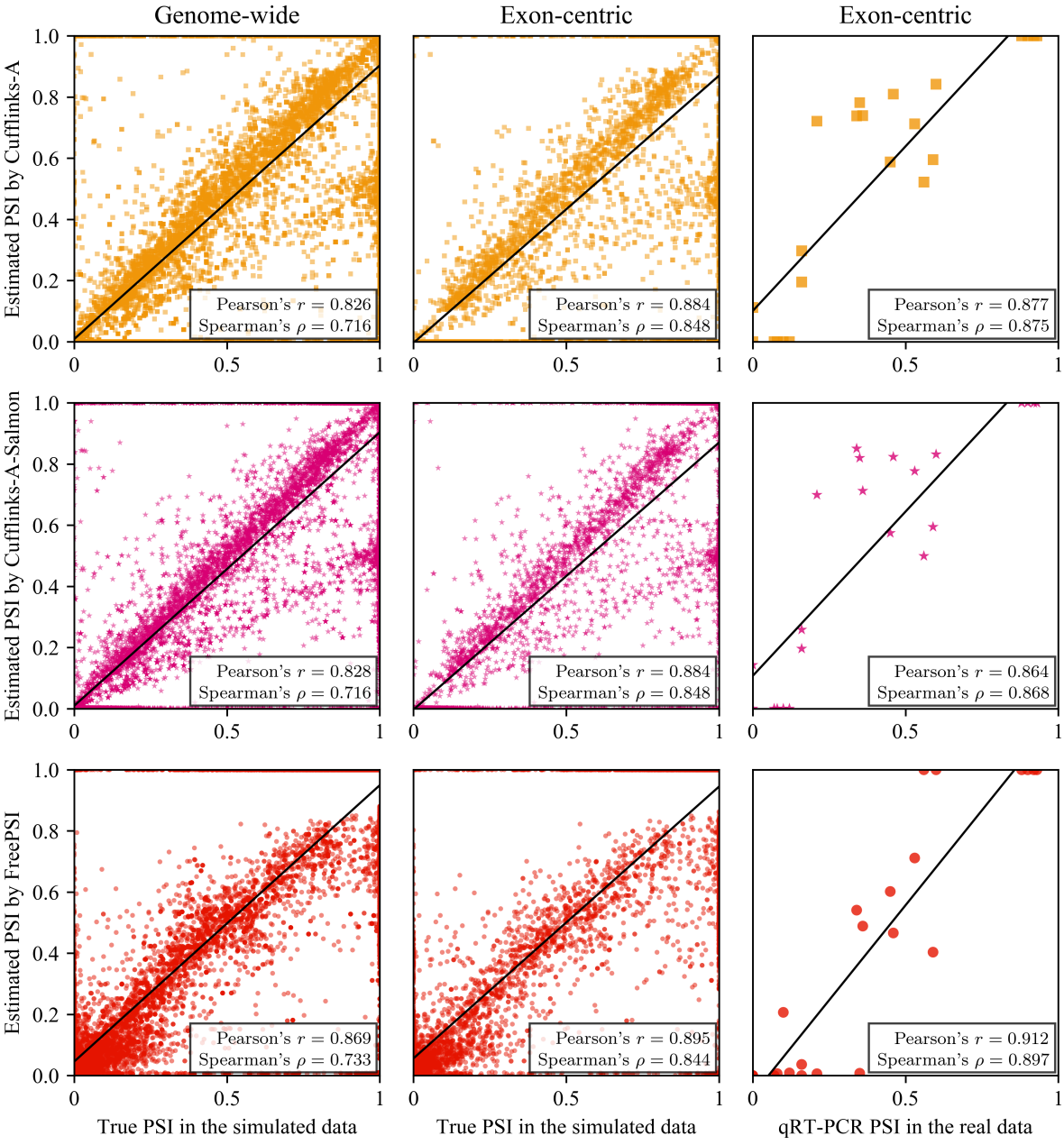



Figure S5: The scatter plots for Cufflinks-A (row 1), Cufflinks-A-Salmon (row 2) and FreePSI (row 3). Note that the plots for Cufflinks-A and FreePSI also appear in Figure 3 and Supplementary Figures S2, S3 and S4. We include them here again for the reader's convenience.

## S5.4 Impact of k-mer length on FreePSI

The parameter $K$ representing the length k-mers considered in FreePSI is critical to the performance of FreePSI. We use the simulated dataset with 100 million reads to study the impact of $K$.

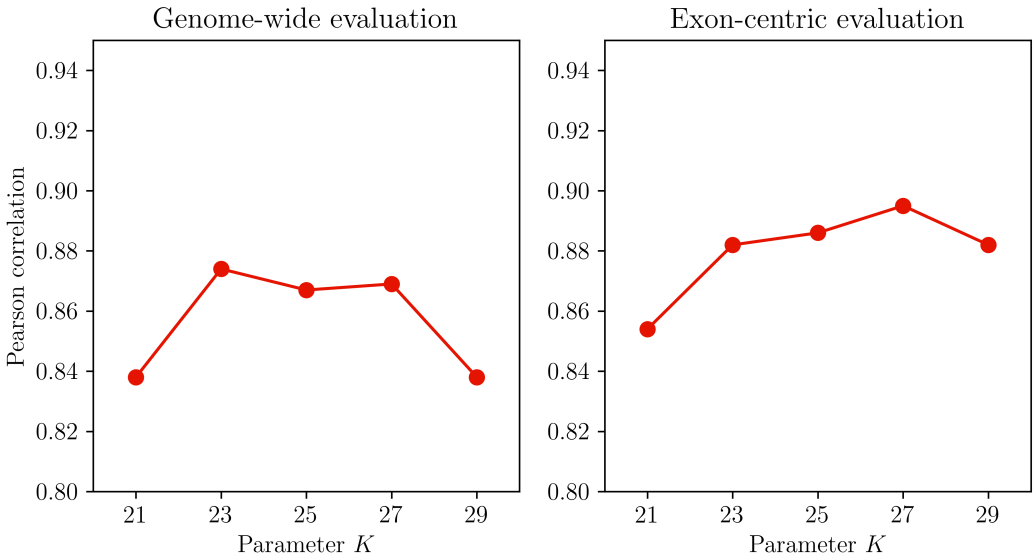



Figure S6: The performance of FreePSI under several choices of $K$.

The above figure shows the performance of FreePSI on the simulated dataset when different values of $K$ was applied. In both evaluations, the performance peaked under a moderate $K$, although the optimal $K$ values were different. The reason is that a smaller $K$ induces more k-mers shared by different segments, which increases the difficulty of the estimation. On the other hand, a larger $K$ results in fewer k-mers representing a segment, which makes the estimation more sensitive to sequencing errors. Hence, a moderate $K$ in the range of 23 and 27 seems to work well for FreePSI generally, and we set the default $K$ as 27.

## S5.5 Supplementary tables

Table S1: Impact of sequencing depth on the performance of MISO, Salmon, Cufflinks-A, and FreePSI on simulated data. The numbers of genes and exons selected for the genome-wide and exon-centric evaluations, respectively, are also shown in the table.

| | # reads | 20M | 50M | 100M |
|---|---|---|---|---|
| Pearson correlation for genome-wide evaluation | # genes | 6907 | 7025 | 7032 |
| | Salmon | 0.997 | 0.998 | 0.998 |
| | Cufflinks-A | 0.783 | 0.836 | 0.826 |
| | FreePSI | 0.783 | 0.837 | 0.869 |
| Pearson correlation for exon-centric evaluation | # exons | 10930 | 10958 | 10919 |
| | MISO | 0.678 | 0.750 | 0.807 |
| | Salmon | 0.978 | 0.994 | 0.995 |
| | Cufflinks-A | 0.773 | 0.845 | 0.884 |
| | FreePSI | 0.817 | 0.862 | 0.895 |

Table S2: Performance of Salmon and Cufflinks-G on incomplete reference transcriptomes with different sampling rates. Here, a sampling rate represents what percentage of the true reference transcriptome would be covered in the provided reference transcriptome.

| | Sampling rate | 100% | 90% | 80% | 70% |
|---|---|---|---|---|---|
| Pearson correlation | Salmon | 0.998 | 0.913 | 0.804 | 0.725 |
| for genome-wide evaluation | Cufflinks-G | 0.986 | 0.904 | 0.796 | 0.718 |
| Pearson correlation | Salmon | 0.995 | 0.934 | 0.848 | 0.780 |
| for exon-centric evaluation | Cufflinks-G | 0.985 | 0.926 | 0.843 | 0.777 |

Table S3: Performance of Salmon, Cufflink-G, Cufflinks-A, and FreePSI on genes under different TPM thresholds.

| TPM | 0 | 1 | 2 | 5 | 10 |
|---|---|---|---|---|---|
| Salmon | 0.967 | 0.997 | 0.997 | 0.998 | 0.998 |
| Cufflinks-G | 0.964 | 0.982 | 0.983 | 0.985 | 0.986 |
| Cufflinks-A | 0.742 | 0.684 | 0.751 | 0.812 | 0.826 |
| FreePSI | 0.856 | 0.802 | 0.828 | 0.856 | 0.869 |

Table S4: Performance of Salmon, Cufflink-G, Cufflinks-A, and FreePSI on 14 gene families with large proportions of multi-mapped reads in the simulation. As a comparison, the corresponding numbers for the whole genome are given in the last row of the table.

| Gene family | Multi-read proportion | FreePSI | Cufflinks-A | Salmon | Cufflinks-G | # genes | # isoforms | # expressed genes |
|---|---|---|---|---|---|---|---|---|
| PAR1 | 38.16 % | 0.968 | 0.230 | 1.000 | 0.434 | 21 | 103 | 8 |
| GST | 36.61 % | 0.760 | 0.419 | 1.000 | 1.000 | 23 | 57 | 8 |
| CDK | 27.12 % | 0.911 | 0.615 | 0.999 | 0.980 | 26 | 88 | 13 |
| MAGE | 25.64 % | 0.978 | 0.541 | 0.999 | 0.997 | 39 | 103 | 7 |
| NLR | 15.90 % | 0.805 | 0.524 | 1.000 | 0.998 | 23 | 60 | 9 |
| TTC | 13.37 % | 0.709 | 0.698 | 0.999 | 0.984 | 115 | 373 | 46 |
| TUB | 10.78 % | 0.907 | 0.755 | 0.999 | 0.998 | 24 | 59 | 11 |
| SDR | 9.95 % | 0.907 | 0.825 | 1.000 | 0.999 | 75 | 179 | 23 |
| NUP | 7.88 % | 0.536 | 0.884 | 1.000 | 0.994 | 32 | 89 | 15 |
| DDX | 5.63 % | 0.943 | 0.867 | 0.994 | 0.974 | 42 | 108 | 17 |
| CLEC | 5.34 % | 0.950 | 0.891 | 0.998 | 0.993 | 46 | 130 | 21 |
| TRIM | 5.23 % | 0.931 | 0.849 | 1.000 | 0.997 | 80 | 189 | 26 |
| SCAR | 4.49 % | 0.915 | 0.942 | 0.999 | 0.996 | 27 | 79 | 9 |
| AKAP | 3.04 % | 0.947 | 0.849 | 1.000 | 0.882 | 29 | 73 | 10 |
| Whole genome | 2.48 % | 0.869 | 0.826 | 0.998 | 0.986 | 23983 | 57822 | 7032 |

## S6 Software configurations

- Jellyfish on simulated data
```
jellyfish count -m 27 -s 100M -t 16 -Q 5 ${READS} -o ${OUTPUT}
```
- Jellyfish on real data
```
jellyfish count -m 27 -s 100M -t 16 -Q A -L 10 ${READS} -o ${OUTPUT}
```
- HISAT
```
hisat2 --fr --dta-cufflinks -p 16 -x ${GENOME_INDEX} -1 ${READS-1} -2 ${READS-2}
 | samtools view --threads 16 -Sbo ${BAM_FILE}
```
- FreePSI
```
freePSI build -k 27 -p 16 -g ${REF_GENOME} -1 ${READS-1} -2 ${READS-2}
-a ${EXON_BND} -o ${HASHTABLE}
freePSI quant -k 27 -p 16 -i ${HASHTABLE} -o ${OUTPUT}
```
- Salmon
```
salmon index -t ${REF_TRANSCRIPTOME} -i ${INDEX}
salmon quant -p 16 -l ISF -i ${INDEX} -1 ${READS-1} -2 ${READS-2} -o ${OUTPUT}
```
- Cufflinks-A
```
cufflinks -u -b ${GENOME_INDEX} -p 16 --library-type fr-secondstrand ${BAM}
-o ${OUTPUT}
```
- Cufflinks-G
```
cufflinks -u -b ${GENOME_INDEX} -p 16 --library-type fr-secondstrand ${BAM}
-G ${REF_TRANSCRIPTOME} -o ${OUTPUT}
```
- MISO
```
miso --run ${REF_SPLICING} ${BAM_FILE} --settings-filename=${DEFAULT_MISO_SETTING}
-p 16 --read-len ${READ_LEN} output-dir ${OUTPUT}
```
- Flux Simulator (`.par` file)
```
REF_FILE_NAME ${REF_TRANSCRIPTOME}
GEN_DIR ${REF_GENOME}
NB_MOLECULES 5000000
TSS_MEAN 25
POLYA_SCALE NaN
POLYA_SHAPE NaN
RTRANSCRIPTION YES
RT_PRIMER RH
FRAG_SUBSTRATE RNA
FRAG_METHOD UR
FRAG_UR_ETA NaN
FRAG_UR_D0 1
READ_NUMBER 100000000
READ_LENGTH 76
PAIRED_END YES
ERR_FILE 76
FASTA YES
UNIQUE_IDS YES
```

More details can be found in the source code.

## References

[1] Jo Bo Rosen. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):181–217, 1960.