Salmon: Accurate, Versatile and Ultrafast Quantification from RNA-seq Data using Lightweight-Alignment

Rob Patro [*1], Geet Duggal[†2] and Carl Kingsford[‡2]

[1]Department of Computer Science, Stony Brook University
[2]Department of Computational Biology, Carnegie Mellon University

Abstract

Transcript quantification is a central task in the analysis of RNA-seq data. Accurate computational methods for the quantification of transcript abundances are essential for downstream analysis. However, most existing approaches are much slower than is necessary for their degree of accuracy. We introduce Salmon, a novel method and software tool for transcript quantification that exhibits state-of-the-art accuracy while being significantly faster than most other tools. Salmon achieves this through the combined application of a two-phase inference procedure, a reduced data representation, and a novel lightweight read alignment algorithm. Salmon is written in `C++11`, and is available under the GPL v3 license as open-source software at https://combine-lab.github.io/salmon.

1 Note

This pre-print presents a description of the Salmon method for RNA-seq quantification, which, after a brief period of private development, has been developed in public as open source software since Spring, 2014 (initially as part of the Sailfish package). Our aim is to provide a description of the method and some preliminary results on its accuracy. A more complete evaluation of the method is in preparation.

2 Introduction

Accurate quantification of transcript-level abundance is one of the fundamental computational challenges in the processing of high-throughput RNA-seq data. Many approaches have been suggested for improving the quality and speed of this task. For example, Li et al. [1] introduced a formal statistical model to account for multi-mapping reads, one of the main factors prohibiting accurate quantification of transcript abundance in the presence of alternative splicing and homologous genes. They introduced an expectation maximization (EM) procedure to determine the maximum

[*]rob.patro@cs.stonybrook.edu
[†]gduggal@cs.cmu.edu
[‡]carlk@cs.cmu.edu

likelihood (ML) estimates of the model parameters given the observed alignments. Under a similar generative model, the use of variational Bayesian (VB) methods have been suggested as an alternative, and superior, approach for estimating relative transcript abundance [2, 3]. Recently, a number of approaches have been proposed to reduce the time and memory required to perform the statistical inference. For example, eXpress [4] adopts an online-EM inference procedure that enables accurate inference of transcript abundance after only a single pass over the read alignments (for sufficiently large data). Other tools like Sailfish [5] and RNA-Skim [6] avoid read mapping entirely, and instead re-formulate the inference problem in terms of k-mers, which allows for a drastic decrease in the time required to process the data. However, these k-mer approaches can sacrifice some accuracy when dealing with complex mixtures of isoforms.

We present a new method for the rapid and accurate inference of transcript-level relative abundance from RNA-seq data that matches or exceeds the accuracy of these "alignment-free" approaches while maintaining or improving on their speed and provide a very efficient implementation of this method in the software tool Salmon. Salmon is a versatile, accurate, and fast tool for estimating transcript abundance from RNA-seq data. Our computational experiments below show that Salmon meets or exceeds the accuracy of existing methods, especially on expressed genes with more than 1 known isoform. Further, Salmon is able to learn and account for the effects experiment-specific parameters and biases. It supports strand-specific protocols and makes use of read-pair information, incuding the learning of an empirical fragment length distribution, for paired-end experiments. All of this is achieved while being computational faster than the k-mer-based approach of Sailfish [5]. The main innovations that allow this advancement are the use of a streaming variational Bayes (VB) inference algorithm combined with a batched VB or batched EM refinement algorithm, the development of a fast, lightweight-alignment procedure that uses chains of maximal exact matches instead of k-mers, and the use of "rich" equivalence classes that maintain the necessary information to support model-adjusted fragment assignments while drastically reducing the amount of data that must be manipulated.

Salmon uses a two-phase parallel inference algorithm designed to scale well with the number of reads in a sequencing experiment. The first phase uses a variant of stochastic collapsed variational Bayesian inference (SCVB0) [7] to simultaneously infer transcript-level abundances and learn the parameters for auxiliary models (e.g. the fragment length distribution and non-uniform fragment start position distribution). Salmon's implementation of SCVB0 is an online inference algorithm that exploits massive parallelism and asynchronous updates, in the spirit of Hogwild! [8]. To our knowledge, Salmon is the first method to apply this parallel inference approach to transcript abundance estimation. This algorithm works in terms of the collapsed variational objective [3], which provides a tighter variational lower-bound on the true posterior compared to the more traditional, non-collapsed objective. Further, during this online phase, a set of "rich" equivalence classes are also built up over the observed fragments, which allows a vastly reduced data representation for subsequent inference. In the second phase of the method, an expectation maximization (EM) (or, optionally, Variational Bayesian inference) algorithm is used to refine the initial transcript abundance estimates until a data-dependent convergence criterion is met.

Salmon also provides a level of versatility not currently possible with any existing approach by abstracting the inference algorithm from the manner in which reads are mapped. Salmon allows users to estimate transcript abundances using two different modes. The alignment-based mode, similar to the majority of existing tools, accepts aligned reads in a standard format (i.e. `bam`, `sam` or `cram` format). Conversely, the lightweight-alignment-based mode allows the user to pass in unmapped reads (i.e. in `fasta/q` format). The potential loci of origin of these reads are then computed using a novel lightweight-alignment algorithm, and, in a streaming fashion, the computed mappings are given as input to the inference algorithm. This versatility allows Salmon to

be used efficiently in a number of different contexts. If an RNA-seq analysis pipeline produces read alignments for purposes other than quantification, those alignments can be used by Salmon to avoid redundant work. Such alignments can often be produced very efficiently using high-performance aligners like STAR [9] and HISAT [10]. On the other hand, if a particular analysis only requires transcript expression estimates (which may then be used for e.g. testing differential expression), the often time-consuming alignment of reads can be avoided.

We compare Salmon to a fast traditional estimator, eXpress, and find it to produce significantly more accurate estimates on simulated reads using two different RNA-seq simulation approaches. We also compare with the recently released method of Kallisto [11] which employs an idea similar in some respects to (but significantly different than) our lightweight-alignment algorithm and again find that Salmon tends to produce more accurate estimates in general, and in particular is better able estimate abundances for multi-isoform genes.

3 Objectives and Models for Abundance Estimation

Assume that, for a particular sequencing experiment, the underlying true transcriptome is given as $\mathcal{T} = \{(t_1, \ldots, t_M), (c_1, \ldots, c_M)\}$, where each t_i is the nucleotide sequence of some transcript (an isoform of some gene) and each c_i is the corresponding number of copies of t_i in the sample. Further, we denote by $\ell(t_i)$ the length of transcript t_i.

The model of the sequencing experiment dictates that, in the absence of experimental bias, library fragments are sampled proportional to $c_i \cdot \ell(t_i)$. That is, the probability of drawing a sequencing fragment from some position on a particular transcript t_i is proportional the total fraction of all nucleotides in the sample that originate from a copy of t_i. This quantity is called the nucleotide fraction [1]:

$$\eta_i = \frac{c_i \cdot \ell(t_i)}{\sum_{j=1}^{M} c_j \cdot \ell(t_j)}.$$

The true nucleotide fractions, $\boldsymbol{\eta}$, though not directly observable, would provide us with a way to measure the true relative abundance of each transcript in our sample. Specifically, if we normalize the η_i by the transcript length $\ell(t_i)$, we obtain a quantity

$$\tau_i = \frac{\frac{\eta_i}{\ell(t_i)}}{\sum_{j=1}^{M} \frac{\eta_j}{\ell(t_j)}},$$

called the transcript fraction [1]. These $\boldsymbol{\tau}$ can be used to immediately compute common measures of relative transcript abundance like transcripts per million (TPM). The TPM measure for a particular transcript is the number of copies of this transcript we would expect to exist in a collection of one million transcripts, assuming this collection had exactly same distribution of abundances as our sample. The TPM for transcript t_i, is given by $\texttt{TPM}_i = \tau_i 10^6$. Of course, in a real sequencing experiment, there are numerous biases, confounding factors, and sampling effects that may alter the above assumptions, and accounting for them is important for making inference accurate, which we will discuss below.

Given a collection of observations (raw sequenced fragments or alignments thereof), and a model similar to the one described above, there are numerous approaches to inferring the relative abundance of the transcripts in the target transcriptome, $\mathcal{T}$. Here we describe two basic inference schemes, both available in Salmon, which are commonly used to perform inference in models similar to the one defined above.

3.1 Maximum likelihood objective

The first scheme takes a maximum likelihood approach to solving for the quantities of interest. Specifically, if we assume that all fragments are generated independently and we are given a vector of known nucleotide fractions $\boldsymbol{\eta}$, a binary matrix of transcript-fragment assignment $\boldsymbol{Z}$ where $z_{ji} = 1$ if fragment j is derived from transcript i, and the set of transcripts $\mathcal{T}$, we can write the probability of observing a set of sequenced fragments $\mathcal{F}$ as:

$$\Pr\{\mathcal{F} \mid \boldsymbol{\eta}, \boldsymbol{Z}, \mathcal{T}\} = \prod_{j=1}^{N} \Pr\{f_j \mid \boldsymbol{\eta}, \boldsymbol{Z}, \mathcal{T}\} = \prod_{j=1}^{N} \sum_{i=1}^{M} \Pr\{t_i \mid \boldsymbol{\eta}\} \cdot \Pr\{f_j \mid t_i, \boldsymbol{z}_{ji} = 1\}. \tag{1}$$

$\Pr\{f_j \mid t_i, z_{ji} = 1\}$ is the probability of generating fragment j given that it came from transcript i. We will use $\Pr\{f_j \mid t_i\}$ as shorthand for $\Pr\{f_j \mid t_i, z_{ji} = 1\}$ since $\Pr\{f_j \mid t_i, z_{ji} = 0\}$ is uniformly 0. The determination of $\Pr\{f_j \mid t_i\}$ is defined in further detail in section 3.3. The likelihood associated with this objective can be optimized using the EM-algorithm as in [1].

3.2 Bayesian objective

One can also take a Bayesian approach to transcript abundance inference as done in [2, 3]. In this approach, rather than directly seeking maximum likelihood estimates of the parameters of interest, we want to infer the posterior distribution of $\boldsymbol{\eta}$. In the notation of [3], we wish to infer $\Pr\{\boldsymbol{\eta} \mid \mathcal{F}, \mathcal{T}, \mathcal{Z}\}$ — the posterior distribution of nucleotide fractions given the transcriptome $\mathcal{T}$ and the observed fragments $\mathcal{F}$. This distribution can be written as:

$$\Pr\{\boldsymbol{\eta} \mid \mathcal{F}, \mathcal{T}, \mathcal{Z}\} \propto \sum_{\boldsymbol{Z} \in \mathcal{Z}} \Pr\{\mathcal{F} \mid \mathcal{T}, \boldsymbol{Z}\} \cdot \Pr\{\boldsymbol{Z} \mid \boldsymbol{\eta}\} \cdot \Pr\{\boldsymbol{\eta}\}, \tag{2}$$

where

$$\Pr\{\boldsymbol{Z} \mid \boldsymbol{\eta}\} = \prod_{i=1}^{M} \prod_{j=1}^{N} \eta_j^{z_{ji}}, \tag{3}$$

and

$$\Pr\{\mathcal{F} \mid \mathcal{T}, \boldsymbol{Z}\} = \prod_{i=1}^{M} \prod_{j=1}^{N} \Pr\{f_j \mid t_i\}^{\boldsymbol{z}_{ji}}. \tag{4}$$

Unfortunately, direct inference on the distribution $\Pr\{\boldsymbol{\eta} \mid \mathcal{F}, \mathcal{T}, \mathcal{Z}\}$ of interest is intractable because its evaluation requires the summation over the exponentially large latent variable configuration space $\mathcal{Z}$. Since the posterior distribution cannot be directly estimated, we must rely on some form of approximate inference. One particularly attractive approach is to apply variational Bayesian (VB) inference in which some tractable approximation to the posterior distribution is assumed.

Subsequently, one seeks the parameters for the approximate posterior under which it best matches the true posterior. Essentially, this turns the inference problem into an optimization problem — finding the optimal set of parameters — which can be efficiently solved by a number of different algorithms. In particular, variational inference seeks to find the parameters for the approximate posterior that minimizes the Kullback-Leibler (KL) divergence between the approximate and true posterior distribution. Though the true posterior may be intractable, this minimization can be achieved by maximizing a lower-bound on the marginal likelihood of the posterior distribution [12], written in terms of the approximate posterior. Salmon optimizes the collapsed variational Bayesian objective [3] in its online phase and the full variational Bayesian objective [2] in the variational Bayesian mode of its offline phase (see section 4.4).

3.3 Fragment-transcript agreement model

We model the conditional probability $\Pr\{f_j \mid t_i\}$ for generating f_j given t_i using a number of auxiliary terms. These terms come from auxiliary models whose parameters do not explicitly depend upon the current estimates of transcript abundances. Thus, once the parameters of these these models have been learned and are fixed, these terms do not change even when the estimate for $\Pr\{t_i \mid \boldsymbol{\eta}\} = \eta_i$ needs to be updated. Salmon uses the following auxiliary terms:

$$\Pr\{f_j \mid t_i\} = \Pr\{\ell \mid t_i\} \cdot \Pr\{p \mid t_i, \ell\} \cdot \Pr\{o \mid t_i\} \cdot \Pr\{a \mid f_j, t_i, p, o, \ell\} \tag{5}$$

Where $\Pr\{\ell \mid t_i\}$ is the probability of drawing a fragment of the inferred length given t_i, and is evaluated based on an observed empirical fragment length distribution. $\Pr\{p \mid t_i, \ell\}$ is the probability of the fragment starting at position p on t_i, computed using an empirical fragment start position distribution as defined in [1]. $\Pr\{o \mid t_i\}$ is the probability of obtaining a fragment aligning with the given orientation to t_i. This is determined by the concordance of the fragment with the user-specified library format. It is 1 if the alignment agrees with the library format and a user-defined prior value $p_{\bar{o}}$ otherwise. Finally, $\Pr\{a \mid f_j, t_i, p, o, \ell\}$ is the probability of generating alignment a of fragment f_j, given that it is drawn from t_i, with orientation o, and starting at position p and is of length ℓ; this term is defined as the coverage score (see section 4.1) for lightweight alignments, and is given by eq. (6) for traditional alignments. The parameters for all auxiliary models are learned during the streaming phase of the inference algorithm from the first N' observations ($5,000,000$ by default). These auxiliary terms can then be applied to all subsequent observations.

3.4 Alignment model

When Salmon is given read alignments as input, it can learn and apply a model of read alignments to help assess the probability that a fragment originated from a particular locus. Specifically, Salmon's alignment model is a spatially varying first-order Markov model over the set of `CIGAR` symbols and nucleotides. To account for the fact that substitution and indel rates can vary spatially over the length of a read, we partition each read into a fixed number of bins (4 by default) and learn a separate model for each of these bins. This allows us to learn spatially varying effects without making the model itself too large (as if, for example, we had attempted to learn a separate model for each position in the read). Given the `CIGAR` string $s = s_0, \ldots, s_{|s|}$ for an alignment a, we compute the probability of a as:

$$\Pr\{a \mid f_j, t_i, p, o, \ell\} = \Pr\{s_0\} \prod_{k=1}^{|s|} \Pr_{(\mathcal{M}_k)}\{s_{k-1} \rightarrow s_k \mid f_j, t_i, p, o, \ell\} \tag{6}$$

where $\Pr\{s_0\}$ is the start probability and $\Pr_{(\mathcal{M}_k)}\{\cdot\}$ is the transition probability under the model at the k^{th} position of the read (i.e., in the bin corresponding to position k). To compute these probabilities, Salmon parses the `CIGAR` string s and moves appropriately along both the fragment f_j and the reference transcript t_i, and computes the probability of transitioning to the next observed state in the alignment (a tuple consisting of the `CIGAR` operation, and the nucleotides in the fragment and reference) given the current state of the model. The parameters of this Markov model are learned from sampled alignments in the online phase of the algorithm (see Algorithm 1).

4 Methods

Salmon consists of three components: a lightweight-alignment model, an online phase that estimates initial expression levels and model parameters and constructs equivalence classes over the input

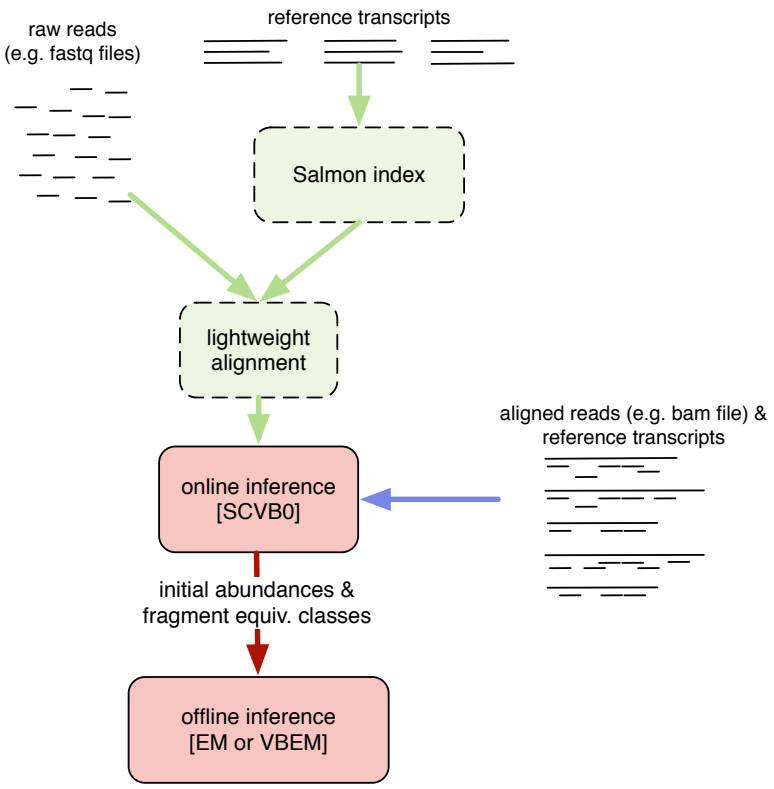

Figure 1: Overview of Salmon's method and components. Salmon excepts either raw or aligned reads as input, performs an online inference when processing fragments or alignments, builds equivalence classes over these fragments and subsequently refines abundance estimates using an offline inference algorithm on a reduced representation of the data.

fragments, and an offline phase that refines the expression estimates. The online and offline phases together optimize the estimates of the latent parameters $\boldsymbol{\alpha}$, and each method can compute $\boldsymbol{\eta}$ directly from these parameters.

The online phase uses a variant of stochastic, collapsed variational Bayesian inference [7]. The offline phase applies the variational Bayesian EM algorithm [12] over a reduced representation of the data represented by the equivalence classes until a data-dependent convergence criterion is satisfied. An overview of our method is given in Figure 1, and we describe each component in more detail below.

4.1 Lightweight-alignment

A key computational challenge in inferring relative transcript abundances is to determine the potential loci-of-origin for a sequenced fragment. To make the optimization tractable, all positions cannot be considered. However, if the sequence of a fragment is substantially different from the sequence of a given transcript at a particular position, it is very unlikely that the fragment originated from this transcript and position — these positions will have their probability truncated to 0 and will be omitted from the optimization. Determining a set of potential loci-of-origin for a sequenced fragment is typically done by aligning the reads to the genome or transcriptome using tools like Bowtie2 [13], STAR [9], or HISAT [10]. While Salmon can process the alignments generated by

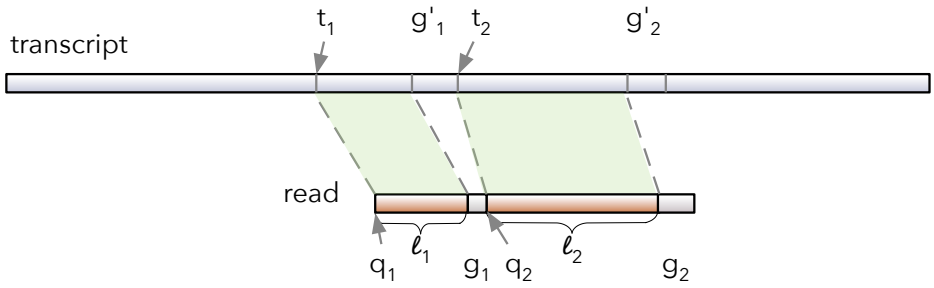

Figure 2: A δ-consistent chain of matches covering the read. Here, the coverage (score) of the chain is $s = \frac{\ell_1+\ell_2}{\ell_1+g_1+\ell_2+g_2}$, and $\delta = |(t_2 - t_1) - (q_2 - q_1)| = |g'_1 - g_1|$.

such tools (when they are given with respect to the transcriptome), it provides another method to determine the potential loci-of-origin of the fragments directly, using a procedure that we call *lightweight-alignment*.

The main motivation behind lightweight-alignment is that achieving accurate quantification of transcript abundance from RNA-seq data does not require knowing the optimal alignment between the sequenced fragment and the transcript for every potential locus of origin. Rather, simply knowing which transcripts (and positions within these transcripts) match the fragments reasonably well is sufficient. Formally, we define lightweight-alignment as a procedure that, given the transcripts $\mathcal{T}$ and a fragment f_i, returns a set of 3-tuples $A(\mathcal{T}, f_i) = \{(t_{i_1}, p_{i_1}, s_{i_1}), \ldots, \}$. Each tuple consists of 3 elements: a transcript $t_{i'}$, a position $p_{i'}$ within this transcript, and a score $s_{i'}$ that summarizes the quality of the match between f_i and $t_{i'}$ at position $p_{i'}$.

We describe, here, the lightweight-alignment approach for a single read (it extends naturally to paired-end reads by looking for lightweight-alignments for read pairs that are appropriately positioned on the same transcript). Salmon attempts to find a chain of super-maximal exact matches (SMEMs) and maximal exact matches (MEMs) that cover a read. Recall, a maximal exact match is a substring that is shared by the query (read) and reference (transcript) that cannot be extended in either direction without introducing a mismatch. A super-maximal exact match [14] is a MEM that is not contained within any other MEM on the query.

Salmon attempts to cover the read using SMEMs. Differences — whether due to read errors or true variation of the sample being sequenced from the reference — will often prevent SMEMs from spanning an entire read. However, one will often be able to find *approximately*-consistent, co-linear chains of SMEMs that are shared between the read and target transcripts. A chain of SMEMs is a collection of 3-tuples $c = \{(q_1, t_1, \ell_1), \ldots\}$ where each q_i is a position on the query (read), t_i is a position on the reference (transcript), and ℓ_i is the length of the SMEM. If $\sum_i |(q_{i+1} - q_i) - (t_{i+1} - t_i)| = 0$, then we say that the chains are consistent — the space between the location of SMEMs on the query and the reference are the same. If, instead, we require that $\sum_i |(q_{i+1} - q_i) - (t_{i+1} - t_i)| \leq \delta$, then we say that the chain is *approximately* consistent, or δ-consistent. Consistent chains can deal only with substitution errors and mutations, while δ-consistent chains can also account for indels. Figure 2 shows an example.

While the discussion above is in terms of SMEMs, the chains constructed by Salmon typically consist of a mix of SMEMs and MEMs. This is because, like BWA-mem [14], Salmon breaks SMEMs that are too large (by default, greater than 1.5 times the minimum required MEM length), to prevent them from masking potentially high-scoring MEM chains. In order for Salmon to consider a read to match a transcript locus sufficiently well, there must be a δ-consistent chain between the read and the transcript sequence, beginning at the locus, that covers a user-specified fraction of

the read (65% by default).

Using this procedure, Salmon implements lightweight-alignment by finding, for a fragment f_i, all transcript position pairs $(t_{i'}, p_{i'})$ that share a δ-consistent chain with f_i covering at least fraction c of the fragment. The score, $s_{i'}$, of this lightweight-alignment is simply the fraction of the fragment covered by the chain.

Salmon searches for SMEMs using the FMD-index [15]. Specifically, Salmon uses a slightly-modified version of the BWA [15] index, replacing the default sparse sampling with a dense sampling to improve speed. When Salmon is run in lightweight-alignment mode, one must have first prepared an index for the target transcriptome against which lightweight-alignment is to be performed. The Salmon index is built using the `index` command of Salmon. Unlike k-mer-based indices (e.g. as used in Sailfish [5] or Kallisto [11]), the parameters for lightweight-alignment (e.g. the fraction of the read required to be covered, or the minimum length MEMs considered in chains) can be modified without re-building the index. This allows one to easily modify the sensitivity and specificity of the lightweight-alignment procedure without the need to re-create the index (which often takes longer than quantification).

4.2 Online phase

The online phase of Salmon attempts to solve the variational Bayesian inference problem described in section 3, and optimizes a collapsed variational objective function [3] using a variant of stochastic collapsed Variational Bayesian inference [7]. The inference procedure is a streaming algorithm that updates estimated read counts $\boldsymbol{\alpha}$ after every small group B^t (called a mini-batch) of observations. The pseudo-code for the algorithm is given in Algorithm 1.

Algorithm 1 Laissez-faire SCVB0

1: **while** $B^t \leftarrow$ pop(work-queue) **do**
2: $\quad \hat{\boldsymbol{x}} \leftarrow \mathbf{0}$
3: $\quad$ **for** read $r \in B^t$ **do**
4: $\quad\quad \boldsymbol{x} \leftarrow \mathbf{0}$
5: $\quad\quad$ **for** alignment a of r **do**
6: $\quad\quad\quad y \leftarrow$ the transcript involved in alignment a
7: $\quad\quad\quad x_y \leftarrow x_y + \alpha_y \cdot \Pr\{a \mid y\}$ ▷ Add a's contribution to the local weight for transcript y
8: $\quad\quad$ **end for** ▷ Normalize the contributions for all alignments of r
9: $\quad\quad$ **for** alignment a of r **do**
10: $\quad\quad\quad y \leftarrow$ the transcript involved in alignment a
11: $\quad\quad\quad \hat{x}_y \leftarrow \hat{x}_y + \frac{x_y}{\sum_{y' \in r} x_{y'}}$
12: $\quad\quad$ **end for**
13: $\quad\quad$ Sample $a \in r$ and update auxiliary models using a
14: $\quad$ **end for**
15: $\quad \boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + v^t \cdot \hat{\boldsymbol{x}}$ ▷ Update the global weights with local observations from B^t
16: **end while**

The observation weight for mini-batch t, v^t, in line 15 of Algorithm 1 is an increasing sequence sequence in t, and is set, as in [4], to adhere to the Robbins-Monroe conditions. Here, the $\boldsymbol{\alpha}$ represent the (weighted) estimated counts of fragments originating from each transcript. Using this method, the expected value of $\boldsymbol{\eta}$ can be computed directly from $\boldsymbol{\alpha}$ using equation 16. We employ a *weak* Dirichlet conjugate-prior with $\alpha_i^0 = 0.01$ for all $t_i \in \mathcal{T}$. As outlined in [7], the

SCVB0 inference algorithm is similar to variants of the online-EM [16] algorithm with a modified prior.

The procedure in algorithm 1 is run independently by as many worker threads as the user has specified. The threads share a single work-queue upon which a parsing thread places mini-batches of alignment groups. An alignment group is simply the collection of all alignments (i.e. all multi-mapping locations) for a particular read. The mini-batch itself consists of a collection of some small, fixed number of alignment groups ($1,000$ by default). Each worker thread processes one alignment group at a time, using the current weights of each transcript and the current auxiliary parameters to estimate the probability that a read came from each potential transcript of origin. The processing of mini-batches occurs in parallel, so that very little synchronization is required, only an atomic compare-and-swap loop to update the global transcript weights at the end of processing of each mini-batch — hence the moniker laissez-faire. This lack of synchronization means that when estimating x_y, we can not be certain that the most up-to-date values of $\boldsymbol{\alpha}$ are being used. However, due to the stochastic and additive nature of the updates, this has little to no detrimental effect [17].

The inference procedure itself is generic over the type of alignments being processed; they may be either regular alignments (e.g. coming from a `bam` file), or lightweight-alignments generated as described in section 4.1 above. After the entire mini-batch has been processed, the global weights for each transcript $\boldsymbol{\alpha}$ are updated. These updates are *sparse*; i.e. only transcripts which appeared in some alignment in mini-batch B^t will have their global weight updated after B^t has been processed. This ensures, as in [4], that updates to the parameters $\boldsymbol{\alpha}$ can be performed efficiently.

4.3 Streaming determination of equivalence classes during the online phase

During its online phase, in addition to performing streaming inference of transcript abundances, Salmon also constructs a highly-reduced representation of the sequencing experiment. Specifically, Salmon constructs "rich" equivalence classes over all of the sequenced fragments. We define an equivalence relation $\sim$ over fragments. Let $M(f_x) = \{t_i \mid (t_i, p_i, s_i) \in A(\mathcal{T}, f_i)\}$ be the set of transcripts to which f_x maps (this can also be analogously computed using traditional alignments). We say $f_x \sim f_y$ if and only if $M(f_x) = M(f_y)$. Related, but distinct notions of alignment-based equivlance classes have been introduced previously (e.g. [18]), and shown to greatly reduce the time required to perform iterative optimization such as that described in Section 4.4.

Fragments which are equivalent can be grouped together for the purpose of inference. Salmon builds up a set of fragment-level equivalence classes by maintaining an efficient concurrent cuckoo hash map [19]. To construct this map, we associate each fragment f_x with $\boldsymbol{t}^x = M(f_x)$, which we will call the label of the fragment. Then, we query the hash map for $\boldsymbol{t}^x$. If this key is not in the map, we create a new equivalence class with this label, and set its count to 1. Otherwise, we increment the count of the equivalence class with this label that we find in the map. The efficient, concurrent nature of the data structure means that many threads can simultaneously query and write to the map while encountering very little contention.

Each key in the hash map is associated with a value that we call a "rich" equivalence class. For each equivalence class $\mathcal{C}^j$, we retain a count d^j, which is the total number of fragments contained within this class. We also maintain, for each class, a weight vector $\boldsymbol{w}^j$. The entries of this vector are in one-to-one correspondence with transcripts i in the label of this equivalence class such that

$$w_i^j = \frac{\sum_{f \in \mathcal{C}^j} \Pr\{f \mid t_i\}}{\sum_{t_k \in \boldsymbol{t}^j} \sum_{f \in \mathcal{C}^j} \Pr\{f \mid t_k\}}. \tag{7}$$

That is, w_i^j is the average conditional probability of observing a fragment from $\mathcal{C}^j$ given t_i over all fragments in this equivalence class. Since the fragments in $\mathcal{C}^j$ are all exchangeable, the pairing between the conditional probability for a particular fragment and a particular transcript need not be maintained, as the following series of equalities holds:

$$w_i^j = \frac{\sum_{f \in \mathcal{C}^j} \Pr\{f \mid t_i\}}{\sum_{f \in \mathcal{C}^j} \sum_{t_k \in \boldsymbol{t}} \Pr\{f \mid t_k\}} = \frac{\sum_{f \in \mathcal{C}^j} \Pr\{f \mid t_i\}}{\sum_{f \in \mathcal{C}^j} 1} = \frac{1}{d^j} \left(\sum_{f \in \mathcal{C}^j} \Pr\{f \mid t_i\} \right) \tag{8}$$

Thus, the aggregate weights stored in the "rich" equivalence classes gives us the power of considering the conditional probabilities specified in the full model, without having to continuously reconsider each of the fragments in $\mathcal{F}$.

4.4 Offline phase

In its offline phase, Salmon uses the "rich" equivalence classes learned during the online phase to refine the inference. Given the set $\boldsymbol{\mathcal{C}}$ of rich equivalence classes of fragments, we can use an expectation maximization (EM) algorithm to optimize the likelihood of the parameters given the data. The abundances $\boldsymbol{\eta}$ can be computed directly from $\boldsymbol{\alpha}$, and we compute maximum likelihood estimates of these parameters which represent the estimated counts (i.e. number of fragments) deriving from each transcript, where:

$$\mathcal{L}\{\boldsymbol{\alpha} \mid \mathcal{F}, \boldsymbol{Z}, \mathcal{T}\} = \prod_{j=1}^{N} \sum_{i=1}^{M} \hat{\eta}_i \Pr\{f_j \mid t_i\} \tag{9}$$

and $\hat{\eta}_i = \frac{\alpha_i}{\sum_j \alpha_j}$. If we write this same likelihood in terms of the equivalence classes $\boldsymbol{\mathcal{C}}$, we have:

$$\mathcal{L}\{\boldsymbol{\alpha} \mid \mathcal{F}, \boldsymbol{Z}, \mathcal{T}\} = \prod_{\mathcal{C}^j \in \boldsymbol{\mathcal{C}}} \left(\sum_{t_i \in \boldsymbol{t}^j} \hat{\eta}_i w_i^j \right)^{d^j}. \tag{10}$$

EM update rule. This likelihood, and hence that represented in eq. (9), can then be optimized by applying the following update equation iteratively

$$\alpha_i^{u+1} = \sum_{\mathcal{C}^j \in \boldsymbol{\mathcal{C}}} d_j \left(\frac{\alpha_i^u w_i^j}{\sum_{t_k \in \boldsymbol{t}^j} \alpha_k^u w_k^j} \right). \tag{11}$$

We apply this update equation until the maximum relative difference in the $\boldsymbol{\alpha}$ parameters satisfies:

$$\Delta\left(\boldsymbol{\alpha^u}, \boldsymbol{\alpha^{u+1}}\right) = \max \frac{\left|\alpha_i^u - \alpha_i^{u+1}\right|}{\alpha_i^{u+1}} < 1 \times 10^{-2} \tag{12}$$

for all $\alpha_i^{u+1} > 1 \times 10^{-8}$.

Let $\boldsymbol{\alpha}'$ be the estimates after having achieved convergence. We can then approximate η_i by $\hat{\eta}_i$, where:

$$\hat{\eta}_i = \frac{\alpha_i'}{\sum_j \alpha_j'}. \tag{13}$$

Variational Bayes optimization. Instead of the standard EM updates of eq. (11), we can, optionally, perform Variational Bayesian optimization by applying VBEM updates as in [2], but adapted to be with respect to the equivalence classes:

$$\alpha_i^{u+1} = \sum_{\mathcal{C}^j \in \boldsymbol{\mathcal{C}}} d^j \left(\frac{e^{\gamma_i^u} w_i^j}{\sum_{t_k \in \boldsymbol{t}^j} e^{\gamma_k^u} w_k^j} \right), \tag{14}$$

where:

$$\gamma_i^u = \Psi\left(\alpha_i^0 + \alpha_i^u\right) - \Psi\left(\sum_k \alpha_k^0 + \alpha_k^u\right). \tag{15}$$

Here, $\Psi(\cdot)$ is the digamma function, and, upon convergence of the parameters, we can obtain an estimate of the expected value of the posterior nucleotide fractions as:

$$\mathbb{E}\{\eta_i\} = \frac{\alpha_i^0 + \alpha_i'}{\sum_j \alpha_j^0 + \alpha_j'} = \frac{\alpha_i^0 + \alpha_i'}{\hat{\alpha}^0 + N}, \tag{16}$$

where $\hat{\alpha}^0 = \sum_{i=1}^{M} \alpha_i^0$. Variational Bayesian optimization in the offline-phase of Salmon is selected by passing the `--useVBOpt` flag to the Salmon `quant` command.

4.5 Sampling from the posterior

After the convergence of the parameter estimates has been achieved in the offline phase, it is possible to draw samples from the posterior distribution using collapsed, blockwise Gibbs sampling over the equivalence classes. Samples can be drawn by iterating over the equivalence classes, and re-sampling assignments for some fraction of fragments in each class according to the multinomial distribution defined by holding the assignments for all other fragments fixed. Many samples can be drawn quickly, since many Gibbs chains can be run in parallel. Further, due to the accuracy of the preceding inference, the chains begin sampling from a good position in the latent variable space almost immediately. These posterior samples can be used to obtain estimates for quantities of interest about the posterior distribution, such as its variance, or to produce confidence intervals. When Salmon is passed the `--useGSOpt` parameter, it will draw a number of posterior samples that can be specified with the `--numGibbsSamples` parameter.

5 Results

5.1 Ground truth simulated data

To assess accuracy in a situation where the true expression levels are known, we generate synthetic data sets using both the Flux Simulator [20] and the `RSEM-sim` procedure used in [11]. The Flux Simulator attempts to model the different stages of an RNA-seq experiment (e.g. amplification, fragmentation, etc.), and it adopts various mathematical models for different stages of the simulation. However, it does not assume the same generative model used by any of the quantification tools tested here. The Flux Simulator data consisted of 75 million 76bp paired-end reads on a transcript population of 5 million molecules for two separate species: *Homo Sapiens* and *Zea Mays*. To generate data with RSEM-sim, we follow the procedure used in [11] — RSEM was run on sample `NA12716_7` of the Geuvadis RNA-seq data to learn model parameters and estimate true expression, and the learned model was then used to generate 20 different simulated datasets, each consisting of 30 million 75bp paired-end reads. All tests were performed with eXpress v1.5.1, Kallisto v0.42.1,

Salmon v0.4.2 and STAR v2.41d. The flag `--useErrorModel` was passed to alignment-based Salmon. Reads were aligned with STAR using the parameters `--outFilterMultimapNmax 200 --outFilterMismatchNmax 99999 --outFilterMismatchNoverLmax 0.2 --alignIntronMin 1000 --alignIntronMax 0 --outSAMtype BAM Unsorted`. Otherwise, default parameters were used unless noted.

5.2 Metrics for accuracy

We compute three different metrics that summarize the agreement of the predicted number of reads originating from each transcript with the known (simulated) read counts. While these different measures generally give consistent results in our testing, they measure different properties of the underlying estimates. We choose to evaluate these error measures on the estimated read counts to minimize the effect of differences in the manner in which different methods normalize expression estimates by the transcript length (e.g. differences in *effective* length calculations).

The first measure is the mean absolute relative difference (MARD), which is computed using the absolute relative difference ARD_i for each transcript i:

$$\text{ARD}_i = \begin{cases} 0 & \text{if } x_i = y_i = 0 \\ \frac{|x_i - y_i|}{0.5|x_i + y_i|} & \text{otherwise} \end{cases}, \tag{17}$$

where x_i is the true value of the number of reads, and y_i is the predicted value. The relative difference is bounded above by 2, and takes on a value of 0 whenever the prediction perfectly matches the truth. To compute the mean absolute relative difference, we simply take $\text{MARD} = \frac{1}{M}\sum_{i=1}^{M} \text{ARD}_i$.

The second measure is the proportionality correlation, which Lovell et al. [21] argue is a good measure for relative quantities like mRNA expression. The proportionality correlation is defined as:

$$\rho_p = \frac{2\text{Cov}\{\log \boldsymbol{x}, \log \boldsymbol{y}\}}{\text{Var}\{\log \boldsymbol{x}\} + \text{Var}\{\log \boldsymbol{y}\}}. \tag{18}$$

As ρ_p is undefined when either true or estimated measurements take on values of 0, we choose to add a small, positive constant (1×10^{-2}) to all values when computing the proportionality correlation. The ρ_p measure varies from -1 to 1, with a value of 1 being representative of perfect proportional correlation.

Finally, we also compute the Spearman correlation coefficient between the true number of reads deriving from each transcript and the number of reads estimated by each quantification method.

Salmon and Kallisto, by default, truncate very tiny expression values to 0. For example, any transcript estimated to produce $< 1 \times 10^{-8}$ reads is assigned an estimated read count of 0. However, eXpress does not perform such a truncation, and very small, non-zero values may have a negative effect in some of the accuracy metrics we compute. To mitigate such effects, in all of our experiments, we first truncate to 0, in the output of eXpress, all values smaller than the minimum non-zero prediction observed in the output of the other methods.

5.3 Overall accuracy on simulated reads

We compared Salmon to the quantification methods Kallisto [11] and eXpress [4]. Figures 3 to 5 show the distribution of the three measures over all RSEM-sim simulations for each of the quantification tools tested. In these simulations, we observe that variability in results between simulations is minimal, suggesting that all the methods tested are reasonably consistent in terms of their estimates. While all methods appear to perform reasonably well, we notice a few broad trends. On

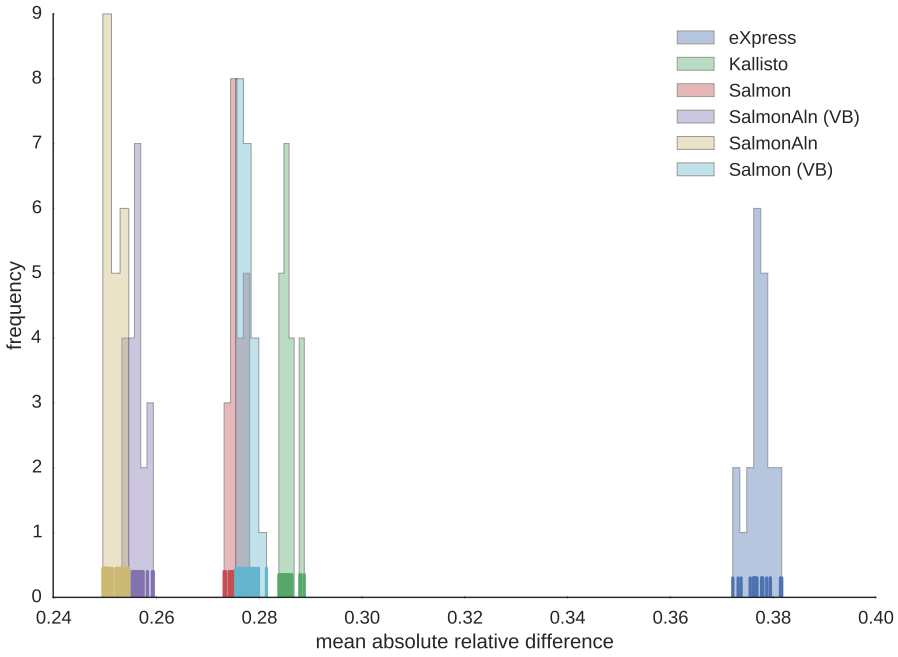

Figure 3: Mean absolute relative difference

this data, variants of Salmon all perform better than the other methods in all metrics except for the Spearman correlation, in which Kalisto performs similar to the variational Bayes inference algorithm of Salmon with lightweight-alignments. This shows that Salmon produces very accurate estimates and that lightweight-alignment is sufficient to produce them.

We observe that alignment-based Salmon tends to outperform lightweight-alignment-based Salmon on these data, and the EM inference algorithm tends to outperform the variational Bayesian inference algorithm. However, the data here is simulated based upon a generative model whose parameters were learned using an EM algorithm. Thus, it seems possible that, in such simulated data, there can be a conflation between accuracy and the agreement of the quantification methods' models with the model underpinning the simulated data. It is possible, for example, that were RSEM [1] performing variational Bayesian inference, other inference methods employing that technique might exhibit more accuracy on its simulated data. Though simulation of highly-realistic RNA-seq data is still an open and active area of research, it is important to account for the fact that simulators based on the same generative models used for inference may introduce certain biases for these inference procedures, and may conflate inference accuracy with model agreement.

We also test accuracy on data that was generated using the Flux Simulator [20], which uses a simulation procedure that attempts to model different steps in the RNA-seq protocol, and is substantially different from that of `RSEM-sim`. The resulting accuracy metrics appear in Table 1. Specifically, we observe that lightweight-alignment-based Salmon tends to perform best, followed by alignment-based Salmon, followed by Kallisto and then eXpress, with the largest gap appearing between the performance of Kallisto and eXpress.

Interestingly, in this setting, lightweight-alignment tends to produce slightly more accurate results here than traditional alignment (using STAR). We also observe that here variational Bayesian (VB) inference seems to slightly, but uniformly, outperform the standard EM algorithm, which is

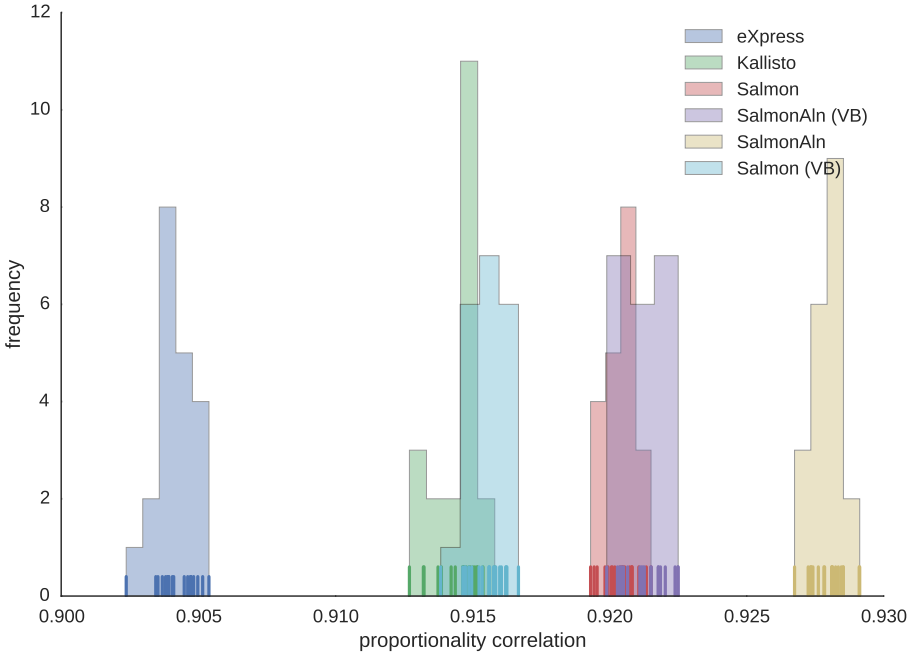

Figure 4: Proportionality correlation coefficients

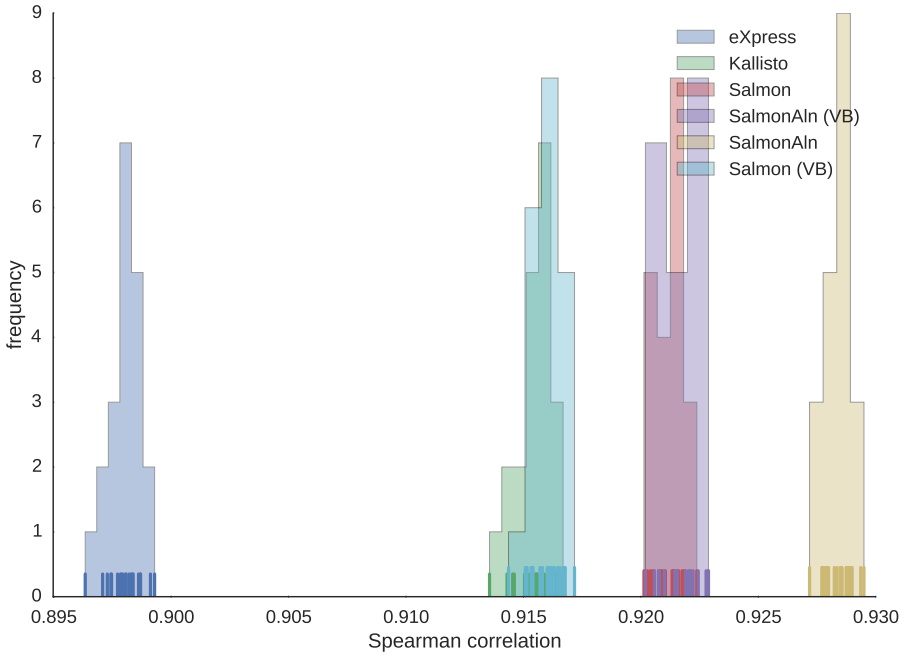

Figure 5: Spearman correlation coefficients

Table 1: The quantification accuracy of the different methods on the synthetic data generated with the Flux Simulator. The experiments consist of 75 Millions 76bp paired-end reads, and the data was simulated for both the *H. sapeins* and *Z. mays* transcriptomes. The accuracy is assessed via the three different metrics described above.

	Kallisto	Salmon	Salmon (VB)	SalmonAln	SalmonAln (VB)	eXpress
H. sapeins						
ρ_p	0.76	0.78	0.79	0.76	0.78	0.75
Spearman corr.	0.69	0.72	0.73	0.70	0.72	0.63
MARD	0.20	0.17	0.14	0.19	0.15	0.25
Z. mays						
ρ_p	0.91	0.92	0.92	0.91	0.91	0.89
Spearman corr.	0.89	0.91	0.91	0.89	0.90	0.85
MARD	0.20	0.17	0.16	0.20	0.19	0.34

consistent with previous observations [2, 3]. This also provides more evidence that lightweight-alignment is sufficient.

5.4 Accuracy on multi-isoform genes

One potential shortcoming of "global" measures of accuracy is that the vast majority of transcripts in any experiment (real or simulated) are likely not expressed at all. Further, a substantial portion of expressed transcripts may originate from single isoform genes. Unless these genes have highly-sequence-similar paralogs within the genome, estimating the abundance of transcripts deriving from them is also a relatively easy task. This is because all or most of the reads that match these transcripts will map uniquely, eliminating one of the primary difficulties in transcript-level abundance estimation.

Thus, we chose to assess the performance of the different methods when stratified by the number of isoforms of the originating gene. This allows us to explore how different methods perform in more difficult scenarios where sequenced fragments must be assigned to one of potentially many different transcripts that share a large proportion of similar sequence. Figure 6 shows the distribution of mean absolute relative differences (MARDs) for the different methods on genes with varying numbers of transcripts. We leave out single isoform genes, as the methods are near indistinguishable in this category, and since this is typically a simple case, and consider only genes with at least 1 truly-expressed isoform to mitigate the effect of the large number of non-expressed genes on the aggregate measurement. Each bar in the plot represents the distribution of MARDs for all transcripts coming from genes with the given range of isoforms. To determine the groupings in Figure 6, we require that each group (except, possibly, the last) have at least $1,000$ examples. The line at the center of the bar denotes the median of the distribution, and the boxes themselves extend from the first to the third quartiles of each distribution. We observe that the different variants of Salmon perform very well on a wide range of isoform counts. Again, the VB variants tend to perform better than the EM variants (not shown), though the differences are often small. eXpress tends to under-perform the other methods, sometimes substantially.

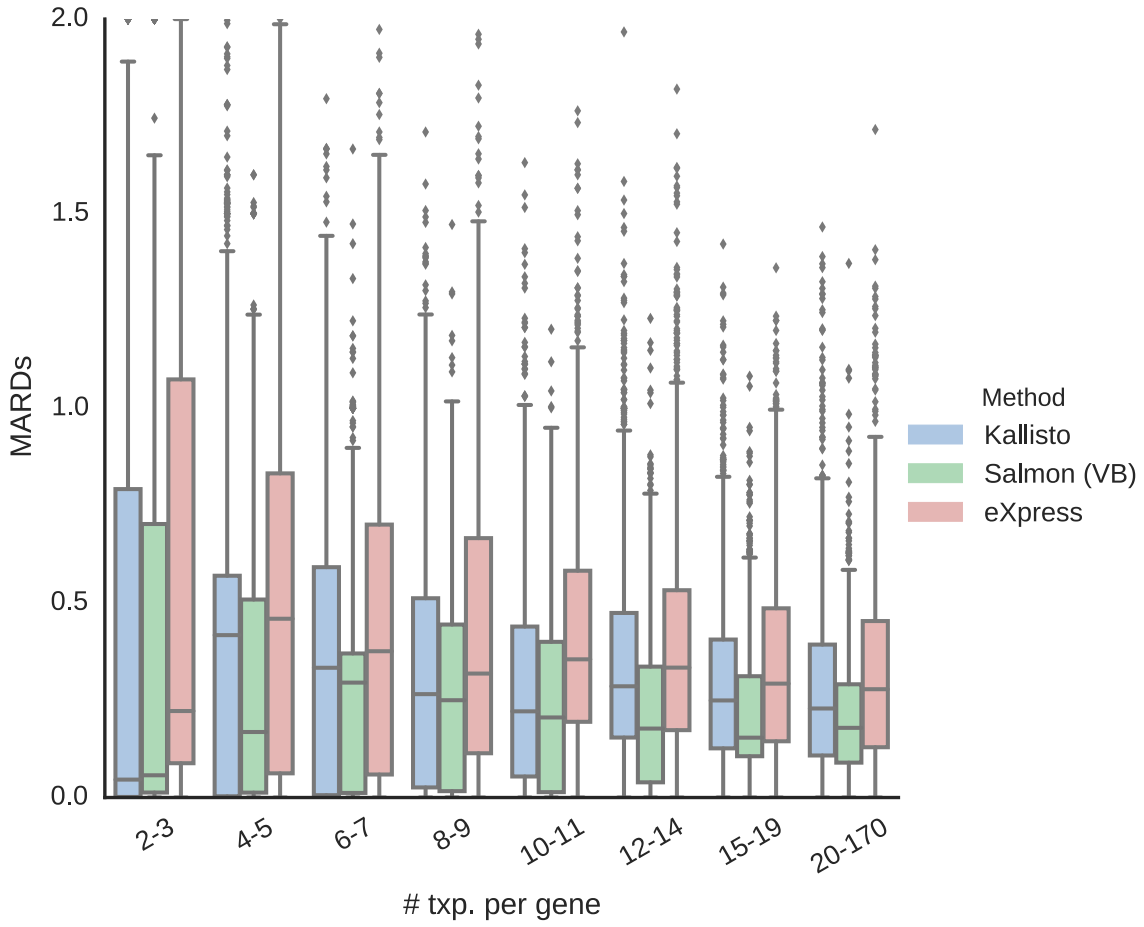

Figure 6: Distributions of mean absolute relative differences (MARDs) for each method, stratified by the number of isoforms that belong to the originating gene.

6 Conclusion

We have introduced Salmon, a fast, accurate and versatile tool for the estimation of transcript-level abundances from RNA-seq data. Salmon achieves its combination of accuracy and speed through a number of innovations, including lightweight-alignment, laissez-faire stochastic inference, and a vastly reduced but expressive representation of an RNA-seq dataset in terms of "rich" equivalence classes over the fragments. On several synthetic RNA-seq datasets, Salmon outperforms existing methods. It also automatically corrects for the effects of a number of experiment-specific parameters and biases. The lightweight-alignment procedure that Salmon employs is a central feature of fast quantification when alignments are not available, and this procedure can also be used independently for aligning reads quickly.

7 Acknowledgements

This research is funded in part by the Gordon and Betty Moore Foundation's Data-Driven Discovery Initiative through Grant GBMF4554 to Carl Kingsford. It is partially funded by the US Na-

tional Science Foundation (CCF-1256087, CCF-1319998) and the US National Institutes of Health (R21HG006913, R01HG007104). C.K. received support as an Alfred P. Sloan Research Fellow.

References

1. Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A. & Dewey, C. N. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics* **26,** 493–500 (2010).
2. Nariai, N. *et al.* TIGAR2: sensitive and accurate estimation of transcript isoform expression with longer RNA-Seq reads. *BMC Genomics* **15,** S5 (2014).
3. Hensman, J., Glaus, P., Honkela, A. & Rattray, M. Fast approximate inference of transcript expression levels from RNA-seq data. *arXiv preprint arXiv:1308.5953v2* (2014).
4. Roberts, A. & Pachter, L. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods* **10,** 71–73 (2013).
5. Patro, R., Mount, S. M. & Kingsford, C. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology* **32,** 462–464 (2014).
6. Zhang, Z. & Wang, W. RNA-Skim: a rapid method for RNA-Seq quantification at transcript level. *Bioinformatics* **30,** i283–i292 (2014).
7. Foulds, J., Boyles, L., DuBois, C., Smyth, P. & Welling, M. *Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation* in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2013), 446–454.
8. Recht, B., Re, C., Wright, S. & Niu, F. *Hogwild!: A lock-free approach to parallelizing stochastic gradient descent* in *Advances in Neural Information Processing Systems* (2011), 693–701.
9. Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29,** 15–21 (2013).
10. Kim, D., Langmead, B. & Salzberg, S. L. HISAT: a fast spliced aligner with low memory requirements. *Nature Methods* **12,** 357–360 (2015).
11. Bray, N., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal RNA-Seq quantification. *arXiv preprint arXiv:1505.02710* (2015).
12. Bishop, C. M. *et al. Pattern recognition and machine learning* **4** (Springer, New York, 2006).
13. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9,** 357–359 (2012).
14. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997* (2013).
15. Li, H. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics* **28,** 1838–1844 (2012).
16. Cappé, O. Online expectation-maximisation. *Mixtures: Estimation and Applications,* 1–53 (2011).
17. Hsieh, C.-J., Yu, H.-F. & Dhillon, I. S. PASSCoDe: Parallel ASynchronous Stochastic dual Co-ordinate Descent. *arXiv preprint arXiv:1504.01365* (2015).
18. Nicolae, M., Mangul, S., Mandoiu, I. I. & Zelikovsky, A. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for Molecular Biology* **6,** 9 (2011).

19. Li, X., Andersen, D. G., Kaminsky, M. & Freedman, M. J. *Algorithmic improvements for fast concurrent cuckoo hashing* in *Proceedings of the Ninth European Conference on Computer Systems* (2014), 27.

20. Griebel, T. *et al.* Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research* **40,** 10073–10083 (2012).

21. Lovell, D., Pawlowsky-Glahn, V., Egozcue, J. J., Marguerat, S. & Bähler, J. Proportionality: a valid alternative to correlation for relative data. *PLoS Computational Biology* **11,** e1004075 (2015).