Accurate, fast, and model-aware transcript expression quantification with Salmon

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

October 2, 2015

Existing methods for quantifying transcript abundance require a fundamental compromise: either use high quality read alignments and experiment-specific models or sacrifice them for speed. We introduce Salmon, a quantification method that overcomes this restriction by combining a novel 'lightweight' alignment procedure with a streaming parallel inference algorithm and a feature-rich bias model. These innovations yield both exceptional accuracy and order-of-magnitude speed benefits over traditional alignment-based methods.

Estimating transcript abundance across cell types, species, and conditions is a fundamental task in genomics. For example, these estimates are used for the classification of diseases and their subtypes [1], for understanding expression changes during development [2], and tracking the progression of cancer [3]. Efficient quantification of transcript abundance from RNA-seq data is an especially pressing problem due to the exponentially increasing number of experiments and the

*rob.patro@cs.stonybrook.edu
†geet@cs.cmu.edu
‡carlk@cs.cmu.edu

growing adoption of expression data for medical diagnosis [4]. However, various methods that address this problem achieve accurate results at the cost of requiring significant computational resources and do not scale well with the rate at which data is produced [5]. The recently developed quantification tool Sailfish [6] achieves an order of magnitude speed improvement over previous approaches, but Sailfish can sometimes produce slightly less accurate estimates for paired-end data or for stranded protocols and does not take advantage of high quality alignment information and experiment-specific models.

We introduce a quantification procedure, called Salmon (**Supplementary Fig. 1**), that achieves best-in-class accuracy, takes advantage of high quality alignment information and experiment-specific models and provides the same order-of-magnitude speed benefits as Sailfish. Using synthetic data from both the RSEM simulator [7] and the Flux Simulator [8] as well as experimental quantitative PCR data [9], we show that Salmon generally outperforms Sailfish and eXpress [10] with respect to accuracy (**Fig. 1a-b,e**; **Supplementary Tables 1&2**) and is also faster than Sailfish (**Fig. 1c**). The transcript abundance estimation problem is particularly difficult for genes with many isoforms since reads derived from these genes can map to many more transcripts, and we find that Salmon is also generally more accurate in this case (**Fig. 1d**). Salmon is designed to run in parallel so that the procedure scales better with the number of reads in an experiment. Salmon can quantify abundance either via a lightweight alignment procedure (**Online methods**, **Lightweight alignment** and **Supplementary Fig. 2**), or using pre-computed alignments provided in SAM or BAM format — we find that the quantification accuracy is robust to this choice of input (**Supplementary Fig. 3**). Salmon is also typically more accurate than a recent unpublished procedure Kallisto (**Supplementary Figs. 4&5**, **Supplementary Table 1**).

An innovation contributing to Salmon's speed and accuracy is its novel lightweight alignment procedure. Salmon attempts to find a chain of super-maximal exact matches (SMEMs) and maximal exact matches (MEMs) to the transcriptome that cover a read. A maximal exact match is a substring that is shared by the read and reference transcript that cannot be extended in either direction without introducing a mismatch, and a super-maximal exact match [11] is a MEM that is not contained within any other MEM on the query. Salmon's lightweight alignment procedure finds co-linear chains of SMEMs. The SMEMS in these chains must be approximately consistent in the sense that the sizes of the gaps between SMEMs in the read and the transcript need not be identical (**Online methods**, **Lightweight alignment** and **Supplementary Fig. 2**). Using a Burrows-Wheeler-based index, this approach allows for the computation of much more accurate alignments than using k-mers at a speed much faster than full alignment (**Fig. 1c**). This approach overcomes potential inaccuracies of using k-mers as in Sailfish while providing some of the benefits of a full alignment. If errors or mutations are uniformly distributed in a read, very few k-mers could map to a transcript even if the read and the transcript share a high-quality alignment. Salmon's improvement in overall accuracy may be due in large part to lightweight alignment since a modification of Sailfish that incorporates this type of efficient alignment starts to approach Salmon's accuracy (**Supplementary Figs. 6&7**, **Supplementary Table 1**). The primary insight 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, it is sufficient to identify the transcripts and positions within them that match the fragments reasonably well.

Salmon also incorporates a rich model of experimental biases, which allows it to account for the affects of experiment-specific parameters and biases including non-uniform read mapping at transcript start sites, strand-specific protocols, and the fragment length distribution. These biases are automatically learned in the online phase of the algorithm, and are encoded in a fragment-transcript agreement model (**Online methods**, **Fragment-transcript agreement model**). In this model, fragment-transcript assignment scores are defined as proportional to (1) the chance of observing a fragment length given a particular transcript/isoform of a gene (2) the chance that a fragment starts at a particular position on the transcript, (3) the concordance of the fragment aligning with a user-defined sequencing library format (e.g. a paired ended, stranded protocol), and (4) the chance that the fragment came from the transcript based on a score obtained from the the lightweight alignment procedure. Salmon additionally incorporates these biases and experimental parameters by maintaining ‘rich equivalence classes’ of fragments (**Online methods**, **Equivalence classes**) that contain the information in these models and speed up the process of estimating transcript abundances.

Salmon’s two-phase parallel inference procedure (**Online methods**, **Online phase** and **Offline phase**; Illustration of method in **Supplementary Fig. 1**) allows it to scale well with the number of reads in an experiment and make use of large multicore machines that are already commonly used to run bioinformatics pipelines. For example, Salmon can quantify a data set of approximately 200 million reads in approximately 5 minutes using 64 cores. Unlike the Sailfish k-mer-based index, 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, allowing for rapid experimentation of quantification parameters. As an alternative to computing lightweight alignments, Salmon’s design also allows the user to provide alignments that have already been computed and uses an alternative alignment scoring model in this case (**Online methods**, **Alignment model**).

The insight behind Salmon’s lightweight alignment approach and sophisticated inference model allows for the use of more sequence information in the read and produces some of the most accurate expression estimates to date. Salmon’s ability to compute high quality estimates of transcript abundances at the previously prohibitive scale of thousands of experiments will also enable individual expression experiments to be interpreted in the context of many rapidly growing sequence expression databases. This will allow for a more comprehensive comparison of the similarity of experiments across large populations of individuals across different environmental conditions and cell types.

Acknowledgements

This research is funded in part by the Gordon and Betty Moore Foundation’s Data- Driven Discovery Initiative through Grant GBMF4554 to C.K. It is partially funded by the US National

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.

Author Contributions R.P. and C.K. designed the method, which was implemented by R.P. R.P., G.D., and C.K. designed the experiments and R.P. and G.D. conducted the experiments. R.P. G.D. C.K. wrote the manuscript.

References

[1] Katherine A Hoadley, Christina Yau, Denise M Wolf, Andrew D Cherniack, David Tamborero, Sam Ng, Max DM Leiserson, Beifang Niu, Michael D McLellan, Vladislav Uzunangelov, et al. Multiplatform analysis of 12 cancer types reveals molecular classification within and across tissues of origin. *Cell*, 158(4):929–944, 2014.

[2] Jingyi Jessica Li, Haiyan Huang, Peter J Bickel, and Steven E Brenner. Comparison of *D. melanogaster* and *C. elegans* developmental stages, tissues, and cells by modENCODE RNA-seq data. *Genome Research*, 24(7):1086–1101, 2014.

[3] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113–1120, 2013.

[4] Ignasi Morán, İldem Akerman, Martijn van de Bunt, Ruiyu Xie, Marion Benazra, Takao Nammo, Luis Arnes, Nikolina Nakić, Javier García-Hurtado, Santiago Rodríguez-Seguí, et al. Human β cell transcriptome analysis uncovers lncRNAs that are tissue-specific, dynamically regulated, and abnormally expressed in type 2 diabetes. *Cell Metabolism*, 16(4):435–448, 2012.

[5] Yuichi Kodama, Martin Shumway, and Rasko Leinonen. The sequence read archive: explosive growth of sequencing data. *Nucleic Acids Research*, 40(D1):D54–D56, 2012.

[6] Rob Patro, Stephen M Mount, and Carl Kingsford. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature Biotechnology*, 32(5):462–464, 2014.

[7] Nicolas Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal RNA-Seq quantification. *arXiv preprint arXiv:1505.02710*, 2015.

[8] Thasso Griebel, Benedikt Zacher, Paolo Ribeca, Emanuele Raineri, Vincent Lacroix, Roderic Guigó, and Michael Sammeth. Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Research*, 40(20):10073–10083, 2012.

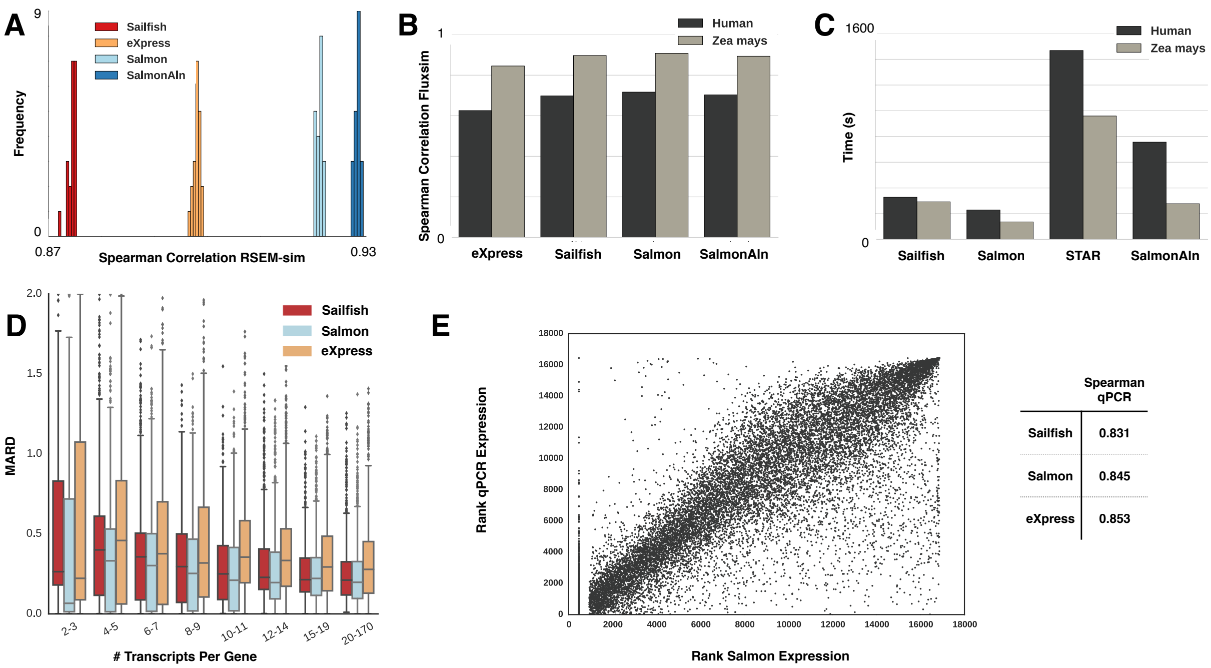
[9] SEQC/MAQC-III Consortium et al. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control Consortium. *Nature Biotechnology*, 32(9):903–914, 2014.

[10] Adam Roberts and Lior Pachter. Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods*, 10(1):71–73, 2013.

[11] Heng Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:1303.3997*, 2013.

[12] Bo Li, Victor Ruotti, Ron M Stewart, James A Thomson, and Colin N Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, 2010.

Figure Caption



	Spearman qPCR
Sailfish	0.831
Salmon	0.845
eXpress	0.853

Figure 1: (A) Distribution of Spearman correlations between estimated and ground truth expression for Sailfish, eXpress, and two variants of Salmon across 20 replicates using RSEM-sim simulated data. Salmon uses lightweight alignment and SalmonAln uses read alignments from the STAR aligner (SalmonAln). (B) Spearman correlations between estimated and ground truth expression for Sailfish, eXpress, Salmon, and SalmonAln using Fluxsim synthetic data on both Human and *Zea mays*. (C) Running time of Sailfish, Salmon, the STAR Aligner, and SalmonAln. (D) Accuracy of Sailfish, Salmon, and eXpress for sets of human transcripts grouped by the number of transcripts associated with the gene. Accuracy is computed using the mean absolute relative difference (MARD) between the estimated expression and ground truth expression (**Online methods**, **Validation**) and is computed on Fluxsim data. We omit single isoform genes, as the methods are near indistinguishable in this category, and genes with 0 truly-expressed isoforms 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. 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. (E) Correlation between qPCR and Salmon estimates of transcript abundance. A scatter plot of expression ranks for Salmon and qPCR is shown to the left and Spearman correlations qPCR data was obtained from a recent study by the SEQC consortium.

.

Online methods

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 [12]:

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

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 [12]. 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.

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:

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

$\Pr\{f_j \mid t_i, z_{ji} = 1\}$ 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 **Fragment-transcript agreement model**. The likelihood associated with this objective can be optimized using the EM algorithm as in [12].

Bayesian objective

One can also take a Bayesian approach to transcript abundance inference as done in [13, 14]. 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 [13], 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:

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

where

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

and

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

Unfortunately, direct inference on the distribution $\Pr\left\{\boldsymbol{\eta} \mid \mathcal{F}, \mathcal{T}, \mathcal{Z}\right\}$ 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 [15], written in terms of the approximate posterior. Salmon optimizes the collapsed variational Bayesian objective [13] in its online phase and the full variational Bayesian objective [14] in the variational Bayesian mode of its offline phase (see **Offline phase**).

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 [12]. $\Pr\{o \mid t_i\}$ 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 Algorithms, Lightweight Alignment) for lightweight alignments, and is given by equation (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.

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**). When lightweight alignments are used instead of user-provided alignments, $\Pr\{a \mid f_j, t_i, p, o, \ell\}$ is taken to be proportional to the normalized coverage of fragment f_j on transcript t_i: coverage$(f_j, t_i)/\max_k$ coverage(f_j, t_k).

Algorithms

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 fragments, and an offline phase that refines the expression estimates. The online and offline phases together optimize the estimates of $\boldsymbol{\alpha}$ which is a vector of weighted estimates of read counts. Each method can compute $\boldsymbol{\eta}$ directly from these parameters.

The online phase uses a variant of stochastic, collapsed variational Bayesian inference [16].

The offline phase applies the variational Bayesian EM algorithm [15] 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 **Supplementary Fig. 1**, and we describe each component in more detail below.

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 [17], STAR [18], or HISAT [19]. While Salmon can process the alignments generated by 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 $A\left(\mathcal{T}, f_i\right) = \{\left(t_{i_1}, p_{i_1}, s_{i_1}\right), \ldots, \}$. 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 [11] 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. **Supplementary Fig. 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 [11], 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 [20]. Specifically, Salmon uses a slightly-modified version of the BWA [20] 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 [6] or Kallisto [7]), 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).

Online phase

The online phase of Salmon attempts to solve the variational Bayesian inference problem described in **Objectives and models for abundance estimation**, and optimizes a collapsed variational objective function [13] using a variant of stochastic collapsed Variational Bayesian inference [16]. The inference procedure is a streaming algorithm that updates estimated read counts $\boldsymbol{\alpha}$ after every small group B^{τ} (called a mini-batch) of observations. The pseudo-code for the algorithm is given in **Algorithm 1**.

The observation weight for mini-batch B^{τ}, v^{τ}, in line 15 of **Algorithm 1** is an increasing sequence sequence in τ, and is set, as in [10], 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 [16], the SCVB0 inference algorithm is similar to variants of the online-EM [21] 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

Algorithm 1 Laissez-faire SCVB0

```
while $B^{\tau} \leftarrow$ pop(work-queue) do
    $\hat{\boldsymbol{x}} \leftarrow \mathbf{0}$
    for read $r \in B^{\tau}$ do
        $\boldsymbol{x} \leftarrow \mathbf{0}$
        for alignment $a$ of $r$ do
            $y \leftarrow$ the transcript involved in alignment $a$
            $x_y \leftarrow x_y + \alpha_y \cdot \Pr\{a \mid y\}$ ▷ Add $a$'s contribution to the local weight for transcript $y$
        end for                                   ▷ Normalize the contributions for all alignments of $r$
        for alignment $a$ of $r$ do
            $y \leftarrow$ the transcript involved in alignment $a$
            $\hat{x}_y \leftarrow \hat{x}_y + \frac{x_y}{\sum_{y' \in r} x_{y'}}$
        end for
        Sample $a \in r$ and update auxiliary models using $a$
    end for
    $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} + v^{\tau} \cdot \hat{\boldsymbol{x}}$              ▷ Update the global weights with local observations from $B^{\tau}$
end while
```

(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 [22]. 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 **Lightweight alignment** 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^{τ} will have their global weight updated after B^{τ} has been processed. This ensures, as in [10], that updates to the parameters $\boldsymbol{\alpha}$ can be performed efficiently.

Equivalence classes

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 $M\left(f_x\right) = \left\{t_i \mid \left(t_i, p_i, s_i\right) \in A\left(\mathcal{T}, f_i\right)\right\}$ be the set of transcripts to which f_x maps according to alignments A. We say $f_x \sim f_y$ if and only if $M\left(f_x\right) = M\left(f_y\right)$. Related, but distinct notions of alignment-based equivlance classes have been introduced previously (e.g. [23]), and shown to greatly reduce the time required to perform iterative optimization such as that described in **Offline phase**. 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 [24]. To construct this map, we associate each fragment f_x with $\boldsymbol{t}^x = M\left(f_x\right)$, 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 $d^j = |\mathcal{C}^j|$, 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

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

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:

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

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 $\mathcal{F}$.

Offline phase

In its offline phase, which follows the online 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:

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

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:

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

EM update rule. This likelihood, and hence that represented in equation (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 equation (11), we can, optionally, perform Variational Bayesian optimization by applying VBEM updates as in [14], 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:

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

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.

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.

Validation

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. [25] 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.

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 [8] and the `RSEM-sim` procedure used in [7].

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, and thus may be a more unbiased simulation method. 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 [7] — RSEM was run on sample `NA12716_7` of the Geuvadis RNA-seq data [26] 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 75 bp 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.

qPCR data

We compared quantification performance of the methods using qPCR data from the SEQC consortium [9]. We obtained normalized Prime PCR estimates for genes from http://abrf.masonlab.net/Files.html and compared abundance estimates of Sample A (Universal Human Reference RNA) with abundance estimates on RNA-seq data from sample A obtained at the BGI site (SEQC_ILM_BGI_A_1, GEO ID: GSE47792). While all tested methods for quantifying abundance seem to produce high concordance with qPCR-based estimates, we find that Salmon performs better than most other methods (**Supplementary Table 2**).

Comparison with Stringtie

We also performed accuracy analyses using a recent transcript assembly and quantification program Stringtie [27]. After quantifying with Stringtie, we noticed that many transcripts that are highly expressed in the ground truth and by other quantifiers are shown as unexpressed in the Stringtie output, resulting in low overall correlation with the ground truth. This may be due to Stringtie's conservative approach. It requires that each exon-intron-exon junction is supported by at least one spliced read in order to be considered in the pool of expressed transcripts. For longer genes with many introns, it may therefore be more likely that transcripts associated with this gene are discarded. We chose not to include these results here to not penalize methods like Stringtie that will attempt to reconstruct rather than just quantify transcripts.

References for Online Methods

[13] J. Hensman, P. Papastamoulis, P. Glaus, A. Honkela, and M. Rattray. Fast and accurate approximate inference of transcript expression from RNA-seq data. *Bioinformatics*, Aug 2015.

[14] Naoki Nariai, Kaname Kojima, Takahiro Mimori, Yukuto Sato, Yosuke Kawai, Yumi Yamaguchi-Kabata, and Masao Nagasaki. TIGAR2: sensitive and accurate estimation of transcript isoform expression with longer RNA-Seq readsonline. *BMC Genomics*, 15(Suppl 10):S5, 2014.

[15] Christopher M Bishop et al. *Pattern Recognition and Machine Learning*, volume 4. Springer, New York, 2006.

[16] James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 446–454. ACM, 2013.

[17] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature Methods*, 9(4):357–359, 2012.

[18] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[19] Daehwan Kim, Ben Langmead, and Steven L Salzberg. HISAT: a fast spliced aligner with low memory requirements. *Nature Methods*, 12(4):357–360, 2015.

[20] Heng Li. Exploring single-sample SNP and INDEL calling with whole-genome *de novo* assembly. *Bioinformatics*, 28(14):1838–1844, 2012.

[21] Olivier Cappé. Online expectation-maximisation. *Mixtures: Estimation and Applications*, pages 1–53, 2011.

[22] Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit S Dhillon. PASSCoDe: Parallel ASynchronous Stochastic dual Co-ordinate Descent. *arXiv preprint arXiv:1504.01365*, 2015.

[23] Marius Nicolae, Serghei Mangul, Ion I Mandoiu, and Alex Zelikovsky. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for Molecular Biology*, 6(1):9, 2011.

[24] Xiaozhou Li, David G Andersen, Michael Kaminsky, and Michael J Freedman. Algorithmic improvements for fast concurrent cuckoo hashing. In *Proceedings of the Ninth European Conference on Computer Systems*, page 27. ACM, 2014.

[25] David Lovell, Vera Pawlowsky-Glahn, Juan José Egozcue, Samuel Marguerat, and Jürg Bähler. Proportionality: a valid alternative to correlation for relative data. *PLoS Computational Biology*, 11(3):e1004075, 2015.

[26] Tuuli Lappalainen, Michael Sammeth, Marc R Friedländer, Peter ACt Hoen, Jean Monlong, Manuel A Rivas, Mar Gonzàlez-Porta, Natalja Kurbatova, Thasso Griebel, Pedro G Ferreira,

et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 501(7468):506–511, 2013.

[27] Mihaela Pertea, Geo M Pertea, Corina M Antonescu, Tsung-Cheng Chang, Joshua T Mendell, and Steven L Salzberg. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology*, 33(3):290–295, 2015.

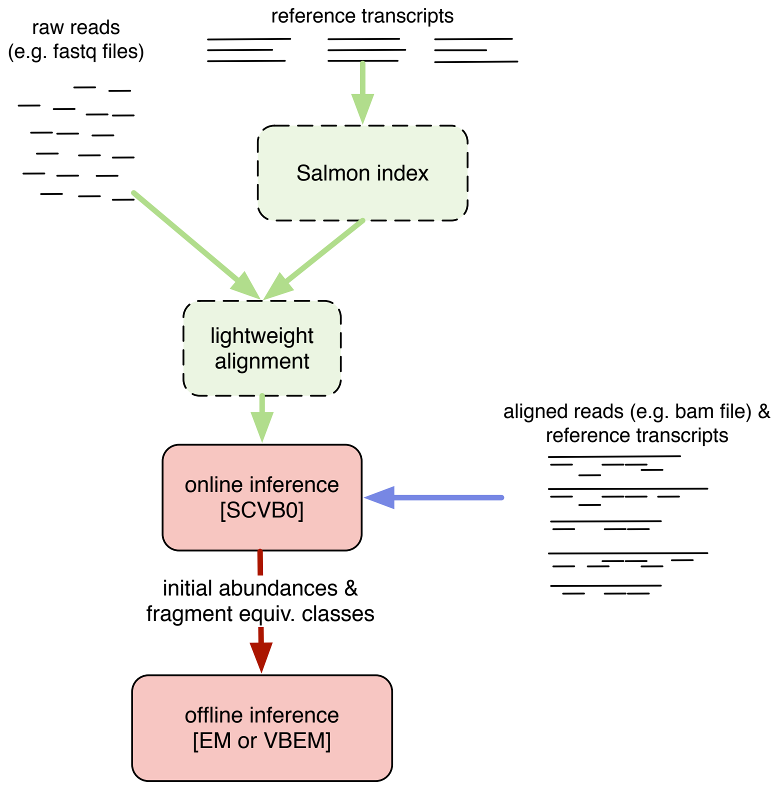
Supplementary Material

for

"Accurate, fast, and model-aware transcript expression quantification with Salmon"

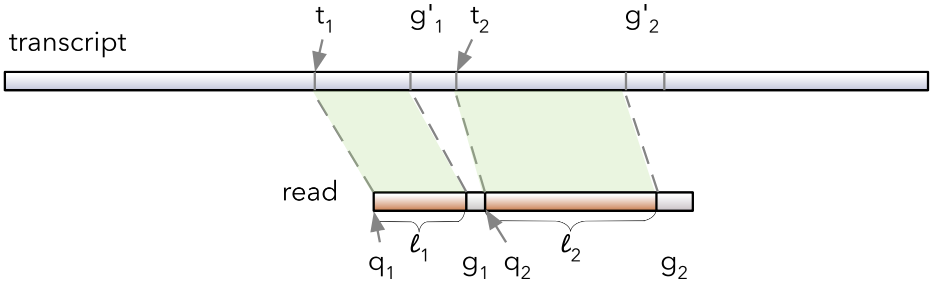
by Rob Patro, Geet Duggal, and Carl Kingsford

Supplementary Figure 1: Overview of Salmon's method and components.



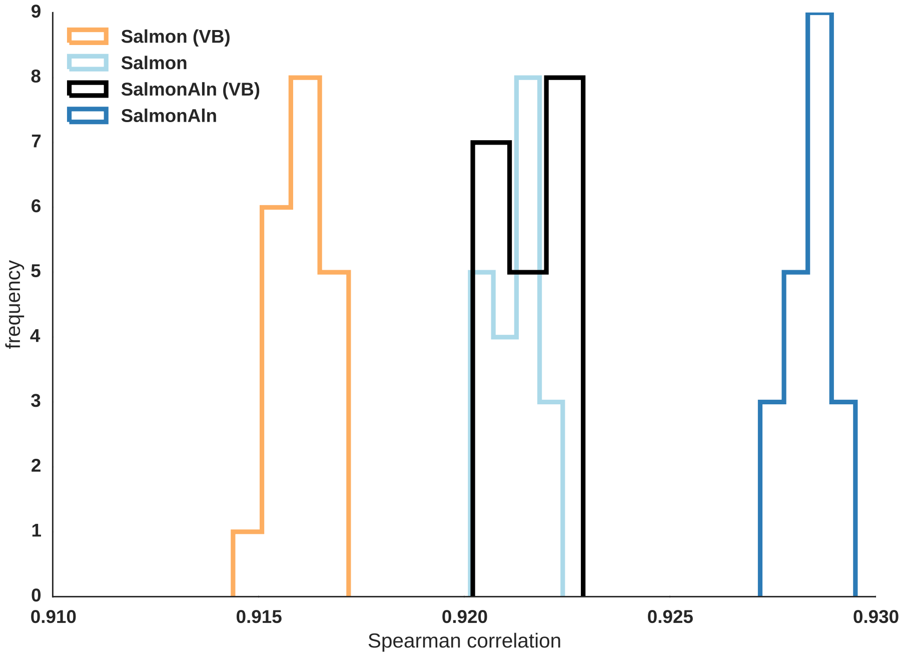
Supplementary Figure 1: Overview of Salmon's method and components. Salmon excepts either raw (green arrows) or aligned reads (blue arrow) as input, performs an online inference when processing fragments or alignments, builds equivalence classes over these fragments and subsequently refines abundance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

Supplementary Figure 2: Illustration of a chained SMEM alignment.



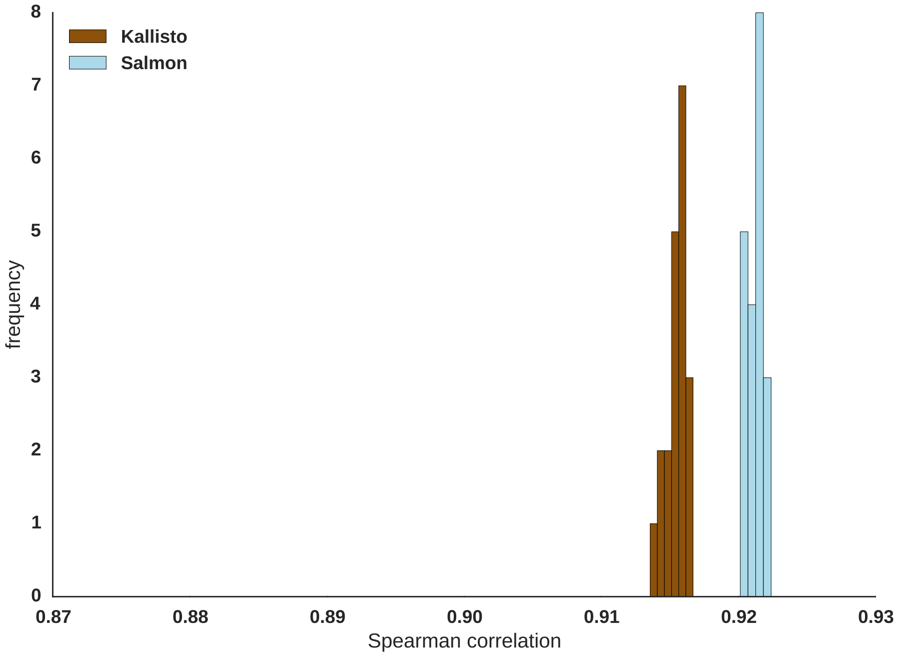
Supplementary Figure 2: A δ-consistent chain of matches to a transcript that covers a read. Here, the coverage (score) of the chain is $s = \frac{\ell_1+\ell_2}{\ell_1+g_1+\ell_2+g_2}$, and $\delta = |(t_2 - t_1) - (q_2 - q_1)| = |g'_1 - g_1|$.

Supplementary Figure 3: Quantification accuracy for Salmon variants using RSEM-sim data



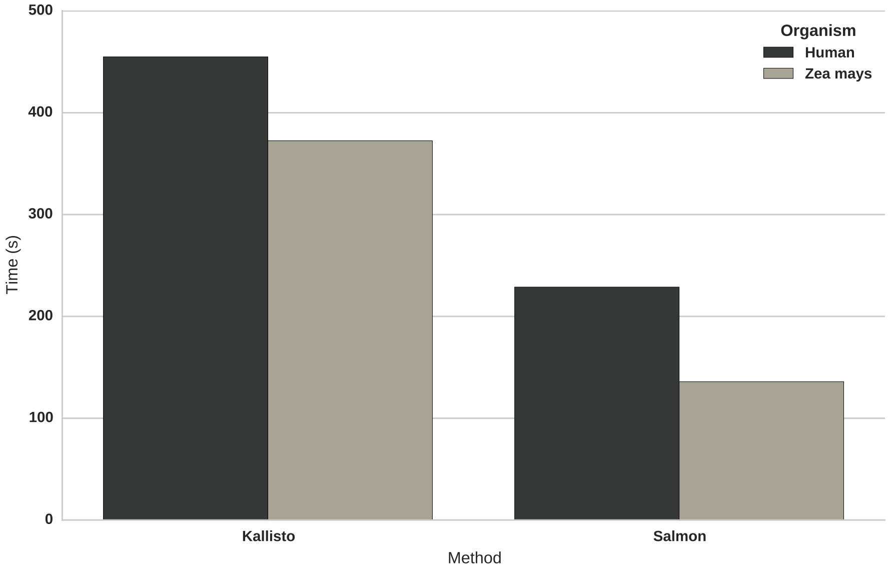
Supplementary Figure 3: Analogous to **Fig. 1a**, this plot compares various the quantification accuracy of various modes of Salmon. Specifically, the distribution of Spearman correlation coefficients across 20 replicates of RSEM-sim data are shown for each variant: Salmon, Salmon with a Variational Bayes offline component (VB), Salmon using read alignments from STAR (SalmonAln), and SalmonAln with a Variational Bayes offline component.

Supplementary Figure 4: Quantification accuracy for Kallisto using RSEM-sim data.



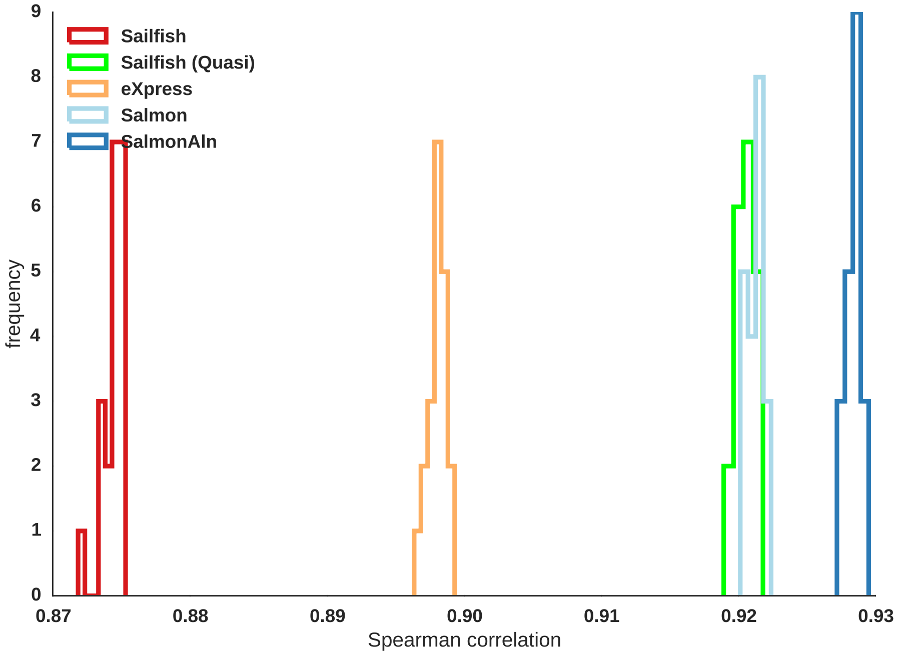
Supplementary Figure 4: Analogous to **Fig. 1a**, this plot compares various the quantification accuracy of Kallisto with Salmon. Specifically, the distribution of Spearman correlation coefficients across 20 replicates of RSEM-sim data are shown for both methods.

Supplementary Figure 5: Timing for Kallisto using Fluxsim data



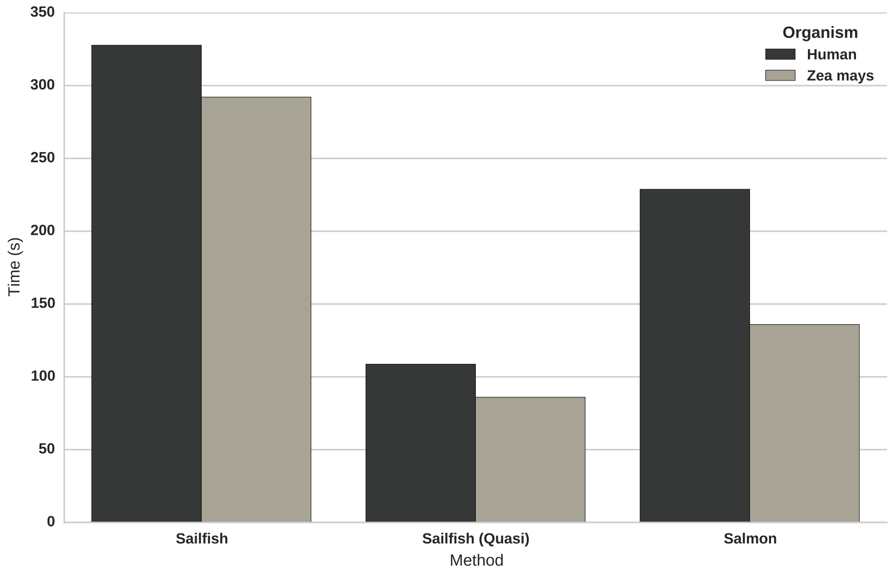
Supplementary Figure 5: Analogous to **Fig. 1c**, this plot compares the run time in seconds of Kallisto with Salmon. Kallisto runs in single thread while Salmon is designed specifically for multi-threaded use and uses 20 threads. In single-threaded mode, Kallisto is 4–6 times faster than Salmon.

Supplementary Figure 6: Quantification accuracy for Sailfish with Quasi-alignment using RSEM-sim data



Supplementary Figure 6: Analogous to **Fig. 1a**, this plot compares various the quantification accuracy of Sailfish using Quasi-alignments with Sailfish, eXpress, Salmon, and SalmonAln. The distribution of Spearman correlation coefficients across 20 replicates of RSEM-sim data are shown for each method. Sailfish with Quasi-mapping is a wrapper around Sailfish that uses techniques similar to lightweight alignment to obtain fragment-transcript mappings. Sailfish then computes its Expectation-Maximization optimization using these mappings. Given that Sailfish with Quasi-alignment approaches the accuracy of Salmon, the quality of the alignment procedure may be most directly related to overall accuracy.

Supplementary Figure 7: Timing for Sailfish with Quasi-alignment using Fluxsim data



Supplementary Figure 7: Analogous to **Fig. 1c**, this plot compares the run time in seconds of Sailfish with Quasi-alignments with Sailfish and Salmon. Sailfish with Quasi-mapping is a wrapper around Sailfish that uses techniques similar to lightweight alignment to obtain fragment-transcript mappings. Sailfish then computes its Expectation-Maximization optimization using these mappings. Sailfish (Quasi) is approximately 3 times faster than Sailfish and more than twice as fast as Salmon. All methods use 20 threads.

Supplementary Table 1: Quantification accuracy of the different methods on the synthetic data generated with the Flux Simulator.

	Kallisto	Sailfish	Sailfish (Quasi)	Salmon	Salmon (VB)	SalmonAln	SalmonAln (VB)	eXpress
H. sapiens								
Proportionality corr.	0.76	0.74	0.76	0.78	0.79	0.76	0.78	0.75
Spearman corr.	0.69	0.70	0.69	0.72	0.73	0.70	0.72	0.63
MARD	0.20	0.21	0.20	0.17	0.14	0.19	0.15	0.25
Z. mays								
Proportionality corr.	0.91	0.90	0.91	0.92	0.92	0.91	0.91	0.89
Spearman corr.	0.89	0.90	0.89	0.91	0.91	0.89	0.90	0.85
MARD	0.20	0.29	0.20	0.17	0.17	0.20	0.19	0.34

Supplementary Table 1: Spearman correlation of abundances for each method with ground truth abundances. The experiments consist of reads generated from Fluxsim, and the data was simulated for both the *H. sapiens* and *Z. mays* transcriptomes. The accuracy is assessed via the three different metrics described above. Sailfish with Quasi-mapping is a wrapper around Sailfish that uses techniques similar to lightweight alignment to obtain fragment-transcript mappings. Sailfish then computes its Expectation-Maximization optimization using these mappings.

Supplementary Table 2: Quantification accuracy of different methods using qPCR data.

H. sapiens				
	Sailfish	Kallisto	Salmon	eXpress
Spearman corr.	0.831	0.837	0.845	0.853

Supplementary Table 2: Spearman correlation of abundances for each method with ground truth abundances derived from qPCR data obtained from the SEQC consortium [9].