nature biotechnology

Near-optimal probabilistic RNA-seq quantification

Nicolas L Bray[1], Harold Pimentel[2], Páll Melsted[3] & Lior Pachter[2,4,5]

We present kallisto, an RNA-seq quantification program that is two orders of magnitude faster than previous approaches and achieves similar accuracy. Kallisto pseudoaligns reads to a reference, producing a list of transcripts that are compatible with each read while avoiding alignment of individual bases. We use kallisto to analyze 30 million unaligned paired-end RNA-seq reads in <10 min on a standard laptop computer. This removes a major computational bottleneck in RNA-seq analysis.

The first two steps in typical transcript-level RNA-seq processing workflows are alignment to a transcriptome or a reference genome and estimation of transcript abundances. These steps can be time consuming. For example, aligning 20 samples, each with 30 million RNA-seq reads, using the widely used program TopHat2 (ref. 1) takes 28 core hours on 20 cores, while quantification with the companion program Cufflinks[2] takes another 14 h. Such running times are likely to become prohibitive as sequence data from increasing numbers of samples are generated. Although the quantification of aligned reads can be sped up with streaming algorithms[3] or by naive counting of reads[4], these approaches have resulted in a decrease in quantification accuracy. To circumvent the alignment step, a recent study proposed quantifying samples by extraction of k-mers from reads followed by exact matching of the k-mers using a hash table[5]. However, shredding reads into k-mers discards valuable information present in complete reads since each k-mer can align to more transcripts than the read itself. This results in a substantial loss of accuracy (**Supplementary Fig. 1**).

Although the direct use of k-mers is inadequate for accurate quantification, the hash-based approach provides a basis for speeding up RNA-seq processing. Here we investigate whether information from k-mers within reads can be combined to maintain the accuracy of alignment-based quantification. We examine the central difficulty and key requirement for accurate quantification, which is the assignment of reads that cannot be uniquely aligned[6]. Typically, these multi-mapping reads are accounted for using a statistical model of RNA-seq[6] that probabilistically assigns such reads while inferring maximum likelihood estimates of transcript abundances. However, it has been shown that accurate quantification does not require information on where inside transcripts the reads may have originated from, but rather which transcripts could have generated them[7]. On the basis of this information, we develop a method based on pseudoalignment of reads and fragments, which focuses only on identifying the transcripts from which the reads could have originated and does not try to pinpoint exactly how the sequences of the reads and transcripts align.

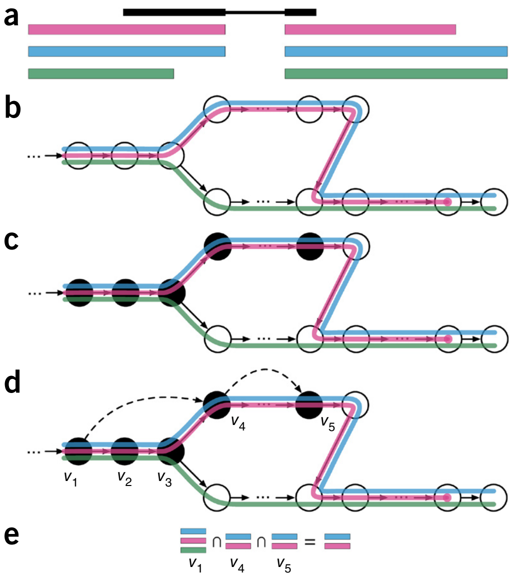
A pseudoalignment of a read to a set of transcripts, T, is a subset, $S \subseteq T$, without specific coordinates mapping each base in the read to specific positions in each of the transcripts in S. Accurate pseudoalignments of reads to a transcriptome can be obtained using fast hashing of k-mers together with the transcriptome de Bruijn graph (T-DBG). de Bruijn graphs have been crucial for DNA and RNA assembly[8], where they are usually constructed from reads. Kallisto uses a T-DBG, which is a de Bruijn graph constructed from k-mers present in the transcriptome (**Fig. 1a**), and a path covering of the graph, a set of paths whose union covers all edges of the graph, where the paths correspond to transcripts (**Fig. 1b**). This path covering of a T-DBG induces multi-sets on the vertices, called k-compatibility classes. A compatibility class can be associated to an error-free read by representing it as a path in the graph and defining the k-compatibility class of a path in the graph as the intersection of the k-compatibility classes of its constituent k-mers (**Fig. 1c**). An equivalence class for a read is a multi-set of transcripts associated with the read; ideally it represents the transcripts a read could have originated from and provides a sufficient statistic for quantification. A key point is that the k-compatibility class of an error-free read coincides with the minimal equivalence class consisting of transcripts containing the read for large k (Online Methods).

Previously, the equivalence classes of reads have been determined via the time-consuming alignment of the reads to the transcriptome. However, since a hash of k-mers provides a fast way to determine their k-compatibility classes, the equivalence class of (error-free) reads can be efficiently determined by selecting suitably large k and then intersecting the reads' constituent k-compatibility classes. The difficulty of implementing such an approach for RNA-seq lies in the fact that reads have errors. However, with very high probability, an error in a k-mer will result in it not appearing in the transcriptome, and such k-mers are simply ignored. The issue of errors is also ameliorated by a technique that we implemented to improve the efficiency of pseudoalignment that removes redundant k-mers from the computation on the basis of information contained in the T-DBG (Online Methods). Because fewer k-mers are inspected, there is less opportunity for erroneous k-mers to produce misleading results. With pseudoalignments efficiently computable, we explored the use of the expectation-maximization (EM) algorithm applied to equivalence classes for quantification[5] (Online Methods). Although the likelihood function is simpler than some other models used for RNA-seq[2,3,9], it still includes a model for bias, and its use has the advantage that the EM algorithm can be applied for many rounds very rapidly.

[1]Innovative Genomics Initiative, University of California, Berkeley, California, USA. [2]Department of Computer Science, University of California, Berkeley, California, USA. [3]Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Reykjavik, Iceland. [4]Department of Mathematics, University of California, Berkeley, California, USA. [5]Department of Molecular & Cell Biology, University of California, Berkeley, California, USA. Correspondence should be addressed to L.P. (lpachter@math.berkeley.edu).

Received 15 October 2015; accepted 25 February 2016; published online 4 April 2016; doi:10.1038/nbt.3519

Figure 1 Overview of kallisto. The input consists of a reference transcriptome and reads from an RNA-seq experiment. (**a**) An example of a read (in black) and three overlapping transcripts with exonic regions as shown. (**b**) An index is constructed by creating the transcriptome de Bruijn Graph (T-DBG) where nodes (v_1, v_2, v_3, ...) are *k*-mers, each transcript corresponds to a colored path as shown and the path cover of the transcriptome induces a *k*-compatibility class for each *k*-mer. (**c**) Conceptually, the *k*-mers of a read are hashed (black nodes) to find the *k*-compatibility class of a read. (**d**) Skipping (black dashed lines) uses the information stored in the T-DBG to skip *k*-mers that are redundant because they have the same *k*-compatibility class. (**e**) The *k*-compatibility class of the read is determined by taking the intersection of the *k*-compatibility classes of its constituent *k*-mers.



To validate and benchmark kallisto, we tested it on a set of 20 RNA-seq simulations generated with the program RSEM (RNA-Seq by Expectation Maximization)[9], as well as on RNA-seq data from the Sequencing Quality Control Consortium (SEQC)[10] for which quantitative PCR (qPCR) can be used as an independent validation of quantification. The transcript abundances and error profiles for the simulated data were based on the quantification of sample NA12716_7 from the Genetic European Variation in Health and Disease (GEUVADIS) data set[11]. To accord with GEUVADIS samples, the simulations consisted of 30 million reads. We examine the quality of the kallisto pseudoalignments as compared to pseudoalignments extracted from Bowtie2 alignments. The two methods agreed exactly on the set of reported transcripts for 70.7% of the reads, but when they differed on the (pseudo)alignment of a read, Bowtie2 reported 8.02 transcripts on average compared to 4.96 for kallisto. Despite being much more specific than Bowtie2, kallisto had almost 100% sensitivity. The transcript of origin was contained in the set of reported transcripts for 99.89% of the reads, only 0.1% less than with Bowtie2 (99.99%). On the real data used as the basis for the simulations (NA12716_7), the programs displayed similar characteristics. The two methods agreed exactly for 66.22% of reads where both (pseudo)aligned, and for differing reads Bowtie2 aligned to 8.94 transcripts on average, versus 4.86 for kallisto. As expected, the number of (pseudo)aligned reads was lower for the real data, with 86.5% of the reads aligned by Bowtie2 versus 90.8% pseudoaligned by kallisto.

The accuracy of kallisto is similar to those of existing RNA-seq quantification tools (**Fig. 2a** and **Supplementary Fig. 2**) and enables a substantial improvement over Cufflinks[2] and Sailfish[5]. The inferior performance of Cufflinks can be attributed to its limited application of the EM algorithm in cases where reads multi-map across genomic locations[12]. Unlike Sailfish[5], which shreds reads into *k*-mers for fast hashing, resulting in a loss of information, kallisto's pseudoalignments explicitly preserve the information provided by *k*-mers across reads (**Supplementary Fig. 1**).

All programs have reduced performance on paralogs owing to the similarity among genes within a family, but kallisto remains highly competitive, again almost matching RSEM's performance (**Supplementary Figs. 3** and **4**). To test kallisto's suitability for allele-specific expression quantification, we simulate reads from a transcriptome with two distinct haplotypes. The Spearman's correlation for kallisto was 0.833 vs. 0.848 for RSEM, 0.830 for eXpress and 0.706 for Sailfish, showing that kallisto is suitable for allele-specific expression. Notably, the simulation was based on RSEM, for generating both the parameters and then the data using them.

We also tested kallisto on SEQC data that has independently been quantified with qPCR. Kallisto performed similarly to other programs

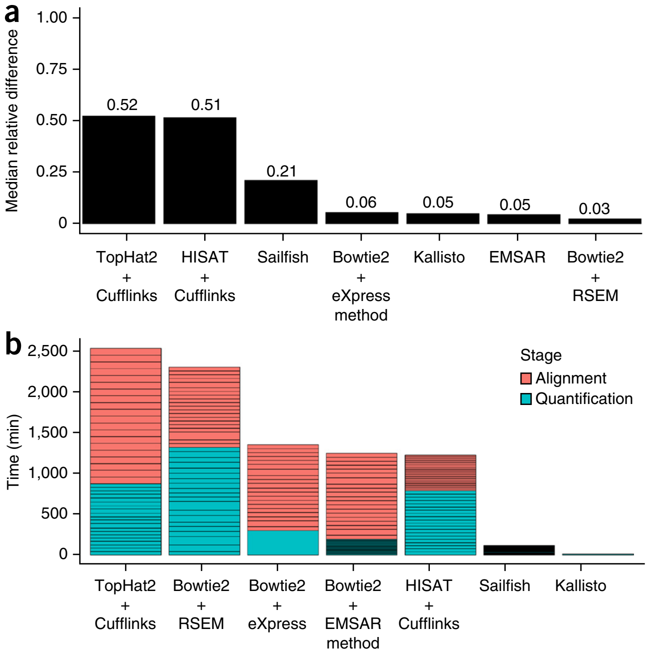


Figure 2 Performance of kallisto and other methods. (**a**) Accuracy of kallisto, Cufflinks, Sailfish, EMSAR, eXpress and RSEM on 20 RSEM simulations of 30 million 75-bp paired-end reads based on the abundances and error profile of GEUVADIS sample NA12716_7 (selected for its depth of sequencing). For each simulation, we report the accuracy as the median relative difference in the estimated read count of each transcript. Estimated counts were used rather than transcripts per million (TPM) because the latter is based on both the assignment of ambiguous reads and the estimation of effective lengths of transcripts, so a program might be penalized for having a differing notion of effective length despite accurately assigning reads. The values reported are means across the 20 simulations (the variance was too small to be visible in this plot). Relative difference is defined as the absolute difference between the estimated abundance and the ground truth divided by the average of the two. (**b**) Total running time in minutes for processing the 20 simulated data sets of 30 million paired-end reads described in **a**. All processing was done using 20 cores, with programs being run with 20 threads when possible (Bowtie2, TopHat2, RSEM, Cufflinks) and 20 parallel processes otherwise (eXpress, kallisto). Each box represents one dataset. Since eXpress and kallisto process all datasets in parallel, the only quantification time shown is the maximum of all the quantifications.

(**Supplementary Table 1**). Learning sequence specific bias (Online Methods and **Supplementary Table 2**) provides a slight improvement in agreement with qPCR, similar to improvements with bias learning in Cufflinks and eXpress.

Kallisto outperformed all other methods in speed, thanks to optimizations made possible by the pseudoalignment framework (**Fig. 1d**,**e**, and Methods). Each simulation was processed on average in less than 7.5 min on a single core. The total runtime for kallisto on the simulated data was 11.47 min (**Fig. 2b**). A simple word count of a simulated data set took 75 s, providing a lower bound for optimal quantification time and demonstrating that kallisto's speed is near optimal. The software is also memory efficient, requiring a maximum of 3.2 Gb of RAM per sample. This allows kallisto to process 30 million read simulations in less than 10 min on a small laptop with a 1.3-GHz processor, demonstrating that with kallisto, RNA-seq analysis of even large data sets is tractable on non-specialized hardware.

The speed of kallisto also enables uncertainty of abundance estimates to be quantified via the bootstrap technique of repeating analyses after resampling with replacement from the data. After the equivalence classes of the original reads have been computed, kallisto samples multinomially from the equivalence classes according to their counts and runs the EM algorithm on those newly sampled equivalence class counts. The running time for each bootstrap sample depends on the number of equivalence classes, which is much smaller than, and roughly independent of, the number of reads. While run times are transcriptome-specific, each sample typically takes on the order of 10 s, and kallisto can multithread the bootstrapping. Since the data associated with each bootstrap consists solely of a set of equivalence class counts and transcript abundances, the memory usage is trivial. We explore the accuracy with which the bootstrap can estimate the uncertainty inherent in a dataset by examining repeated 30 million read subsamples of a deep 216-million-read human RNA-seq dataset from the SEQC-MAQCIII[12] consortium (**Supplementary Fig. 5**). We perform 40 bootstraps (see **Supplementary Fig. 6** for an analysis of convergence) on only a single sample of 30 million reads, yet the variance in estimates correlated highly ($R = 0.933$) with the variance of abundance estimates obtained from the other subsamples. While it is expected that the variance on abundance estimates should increase approximately linearly with abundance[13], our results show that there is high variability in uncertainty of estimates as a result of the complex structure of similarity among transcripts, especially multiple isoforms of genes. A naive attribution of Poisson variance to the shot noise in read count estimates from transcripts, as is commonly done in gene-level RNA-seq analyses, is revealed to be a poor proxy for the true variance (**Supplementary Figs. 7** and **8**). Thus, the bootstrap should prove to be valuable in downstream applications of RNA-seq, as kallisto now allows the uncertainty in estimates to be factored in to downstream statistical computations.

The simplicity of kallisto means that the software has few parameters. Only the *k*-mer length and the mean of the fragment length distribution are required for quantification. The latter is estimated during run-time when paired-end reads are provided. The *k*-mer length must be large enough that random sequences of length *k* do not match to the transcriptome and short enough to ensure robustness to errors. Subject to those constraints, the performance of kallisto is robust to the *k*-mer length chosen (**Supplementary Figs. 9** and **10**). Although we have focused on the performance of kallisto on RNA-seq, the method should be generally applicable to quantification of sequence census datasets[14].

METHODS

Methods and any associated references are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

N.L.B., H.P. and L.P. were partially funded by NIH R01 HG006129. P.M. was partially funded by a Fulbright fellowship.

AUTHOR CONTRIBUTIONS

N.L.B. and L.P. developed the concept of pseudoalignment and conceived the idea for applying it to RNA-seq quantification. P.M. conceived the implementation using De Bruijn graphs. N.L.B., H.P., P.M. and L.P. designed the kallisto software and N.L.B. implemented a prototype. H.P. and P.M. wrote the current kallisto implementation. N.B. and H.P. automated production of the results. N.L.B., H.P., P.M. and L.P. analyzed results and wrote the paper.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

1. Kim, D. *et al. Genome Biol.* **14**, R36 (2013).
2. Trapnell, C. *et al. Nat. Biotechnol.* **28**, 511–515 (2010).
3. Roberts, A. & Pachter, L. *Nat. Methods* **10**, 71–73 (2013).
4. Anders, S., Pyl, P.T. & Huber, W. *Bioinformatics* **31**, 166–169 (2015).
5. Patro, R., Mount, S.M. & Kingsford, C. *Nat. Biotechnol.* **32**, 462–464 (2014).
6. Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L. & Wold, B. *Nat. Methods* **5**, 621–628 (2008).
7. Nicolae, M., Mangul, S., Măndoiu, I. & Zelikovsky, A. in *Algorithms in Bioinformatics* (eds. Moulton, V. & Singh, M.) 202–214 (Springer, 2010).
8. Compeau, P.E.C., Pevzner, P.A. & Tesler, G. *Nat. Biotechnol.* **29**, 987–991 (2011).
9. Li, B. & Dewey, C.N. *BMC Bioinformatics* **12**, 323 (2011).
10. SEQC/MAQC-III Consortium. *Nat. Biotechnol.* **32**, 903–914 (2014).
11. Lappalainen, T. *et al. Nature* **501**, 506–511 (2013).
12. Roberts, A., Trapnell, C., Donaghey, J., Rinn, J.L. & Pachter, L. *Genome Biol.* **12**, R22 (2011).
13. Marioni, J.C., Mason, C.E., Mane, S.M., Stephens, M. & Gilad, Y. *Genome Res.* **18**, 1509–1517 (2008).
14. Wold, B. & Myers, R.M. *Nat. Methods* **5**, 19–21 (2008).

ONLINE METHODS

Index construction. The construction of the index starts with the formation of a colored de Bruijn graph[15] from the transcriptome, where the colors correspond to transcripts. In the colored transcriptome de Bruijn graph, each node corresponds to a k-mer and every k-mer receives a color for each transcript it occurs in. Contigs are defined to be linear stretches of the de Bruijn graph that have identical colorings. This ensures that all k-mers in a contig are associated with the same equivalence class (the converse is not true: two different contigs can be associated with the same equivalence class). Once the graph and contigs have been constructed, kallisto stores a hash table mapping each k-mer to the contig it is contained in, along with the position within the contig. This structure is called the kallisto index.

For error-free reads, there can be a difference between the equivalence class of a read and the intersection of its k-compatibility classes. But for a read of length l this can only happen if there are two transcripts that have the same $l - k + 1$ k-mers occurring in different order. This is unlikely to happen for large k because it would imply that the T-DBG has a directed cycle shorter than $l - k + 1$. This fact also provides a criterion that can be tested.

Pseudoalignment. Reads are pseudoaligned by looking up the k-compatibility class for each k-mer in the read in the kallisto index and then intersecting the identified k-compatibility classes. In the case of paired-end reads, the k-compatibility class lookup is done for both ends of the fragment and all the resulting classes are intersected. Since the T-DBG identifies each k-mer with its reverse complement, the k-mer hashing in kallisto is strand-agnostic; however, the implementation could also be adapted to require specific strandedness of reads from strand-specific protocols. To further speed up the processing, kallisto uses the structural information stored in the index: because all k-mers in a contig of the T-DBG have the same k-compatibility class, it would be redundant to include more than one k-mer from a contig in the intersection of k-compatibility classes. This observation is leveraged in kallisto by finding the distances to the junctions at the end of its contig each time a k-mer is looked up using the hash. If the read does arise from a transcript in the T-DBG, the k-mers up to those distances can be skipped without affecting the result of the intersection, resulting in fewer hash lookups. To help ensure that the read is consistent with the T-DBG, kallisto checks the last k-mer that is skipped to ensure its k-compatibility class is equal as expected. In rare case when there is a mismatch, kallisto defaults to examining each k-mer of the read. For the majority of reads, kallisto ends up performing a hash lookup for only two k-mers (**Supplementary Fig. 11**). While pseudoalignment does not require or make use of the locations of k-mers in transcripts, it is possible to extract such data from the T-DBG, and a "pseudobam output" option of kallisto takes advantage of this to produce an alignment file containing positions of reads within transcripts. With pseudobam it is possible to examine the location of reads within transcripts and genes of interest for quality control and analysis purposes.

Quantification. In order to rapidly quantify transcript abundances from pseudoalignments, kallisto makes use of the following form of the likelihood function for RNA-seq:

$$L(\alpha) \propto \prod_{f \in F} \sum_{t \in T} y_{f,t} \frac{\alpha_t}{l_t} = \prod_{e \in E} \left(\sum_{t \in e} \frac{\alpha_t}{l_t} \right)^{c_e} \qquad (1)$$

In equation (1), F is the set of fragments, T is the set of transcripts, l_t is the (effective) length[3] of transcript t and $\mathbf{y}_{f,t}$ is a compatibility matrix defined as 1 if fragment f is compatible with t and 0 otherwise. The parameters are the α_t, the probabilities of selecting fragments from transcripts. The likelihood can be rewritten as a product over equivalence classes, in which similar summation terms have been factored together. In the factorization the numbers c_e are the number of counts observed from equivalence class e. When equation (1) is written in terms of the equivalence classes, the equivalence class counts are sufficient statistics and thus, in the computations, are based on a much smaller set of data (usually hundreds of thousands of equivalence classes instead of tens of millions of reads). The likelihood function is iteratively optimized with the EM algorithm, with iterations terminating when, for every transcript t, $\alpha_t N > 0.01$ (N is the total number of fragments) changes less than 1% from iteration to iteration.

The transcript abundances are output by kallisto in transcripts per million[9] (TPM) units.

Bias correction. There are many sources of bias in RNA-seq, but previous work has identified sequence-specific bias[12] as particularly problematic. Sequence-specific bias arises as a result of nonrandom priming of fragments, where the nucleotide sequences at the 3′ and 5′ ends affect the probability of sampling. The kallisto correction is similar to that of Roberts *et al.*[12]; however, it uses 6-mers of the transcript sequence overlapping the 5′ fragment, starting 2 bp upstream of the fragment. First kallisto measures the empirical frequency of 6-mers as estimated from the first 1 million pseudoalignable reads. To apply the bias correction, it uses an initial estimate for the abundance, using 50 rounds of the EM algorithm. The bias of 6-mers is used to adjust the effective length of each transcript by adding the bias of each 6-mer on both strands. To account for edge effects, kallisto only add the 6-mers from the start up to the length of the transcript minus the average fragment length. This process is repeated once more with an updated expression estimate after 550 rounds of the EM algorithm.

Bootstrap. The bootstrap is highly efficient in kallisto both because the EM algorithm is very fast and because the sufficient statistics of the model are the equivalence class counts. This latter fact means that bootstrap samples can be very rapidly generated once pseudoalignment of the fragments is completed. With the N original fragments having been categorized by equivalence class, generating a new bootstrap sample consists of sampling N counts from a multinomial distribution over the equivalence classes, with the probability of each class being proportional to its count in the original data. The transcript abundances for these new samples are then recomputed using the EM algorithm.

In kallisto the number of bootstraps to be performed is an option passed to the program, and because a large amount of data can be produced, the output is compressed in HDF5. The HDF5 files can be read into another program for processing (for example, R) or can be converted to plain text using kallisto.

Software, simulations and analysis. The kallisto program is available as **Supplementary Software** and for download from http://pachterlab.github.io/kallisto/. The parameters and procedures for Cufflinks, Sailfish, EMSAR[16], eXpress, and RSEM used for the results and figures in the paper are available via a Snakefile[17] at https://github.com/pachterlab/kallisto_paper_analysis/. Source code for reproducing results and figures of the paper is available as **Supplementary Code**.

[15] Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G. *Nat. Genet.* **44**, 226–232 (2012).

[16] Lee, S., Seo, C.H., Alver, B.H., Lee, S. & Park, P.J. *BMC Bioinformatics* **16**, 278 (2015).

[17] Köster, J. & Rahmann, S. *Bioinformatics* **28**, 2520–2522 (2012).

doi:10.1038/nbt.3519