Bioinformatics Advance Access published October 25, 2012

STAR: ultrafast universal RNA-seq aligner

Alexander Dobin[1*], Carrie A. Davis[1], Felix Schlesinger[1], Jorg Drenkow[1], Chris Zaleski[1], Sonali Jha[1], Philippe Batut[1], Mark Chaisson[2] and Thomas R. Gingeras[1]

[1]Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA.

[2]Pacific Biosciences, Menlo Park, California, USA.

Associate Editor: Dr. Inanc Birol

ABSTRACT

Motivation: Accurate alignment of high-throughput RNA-seq data is a challenging and yet unsolved problem because of the non-contiguous transcript structure, relatively short read lengths and constantly increasing throughput of the sequencing technologies. Currently available RNA-seq aligners suffer from high mapping error rates, low mapping speed, read length limitation and mapping biases.

Results: To align our large (exceeding 80 billon reads) ENCODE Transcriptome RNA-seq dataset we developed the Spliced Transcripts Alignment to a Reference (STAR) software based on a previously un-described RNA-seq alignment algorithm which utilizes sequential maximum mappable seed search in uncompressed suffix arrays followed by seed clustering and stitching procedure. STAR outperforms other aligners by more than a factor of 50 in mapping speed, aligning to the human genome 550 Million 2x76bp paired-end reads per hour on a modest 12-core server, while at the same time improving alignment sensitivity and precision. In addition to unbiased *de novo* detection of canonical junctions, STAR can discover non-canonical splices and chimeric (fusion) transcripts, and is also capable of mapping full length RNA sequences. Using Roche 454 sequencing of RT-PCR amplicons, we experimentally validated 1,960 novel intergenic splice junctions with an 80-90% success rate, corroborating the high precision of the STAR mapping strategy.

Implementation and Availability: STAR is implemented as a standalone C++ code. STAR is free open source software distributed under GPLv3 license and can be downloaded from http://code.google.com/p/rna-star/

Contact: dobin@cshl.edu

1 INTRODUCTION

While genomes are composed of linearly ordered sequences of nucleic acids, eukaryotic cells generally reorganize the information in the transcriptome by splicing together non-contiguous exons to create mature transcripts (Hastings and Krainer, 2001). The detection and characterization of these spliced RNAs have been a critical focus of functional analyses of genomes in both the normal and disease cell states. Recent advances in sequencing technologies have made transcriptome analyses at the single nucleotide level almost routine. However, hundreds of millions of short (36nt) to medium (200nt) length sequences (reads) generated by such high throughput sequencing experiments present unique challenges to detection and characterization of spliced transcripts. Two key tasks make these analyses very computationally intensive. The first task is an accurate alignment of reads that contain mismatches, insertions and deletions caused by genomic variations and sequencing errors. The second task involves mapping sequences derived from non-contiguous genomic regions comprising spliced sequence modules that are joined together to form spliced RNAs. While the first task is shared with DNA re-sequencing efforts, the second task is specific and crucial to the RNA-seq, since it provides the connectivity information needed to reconstruct the full extent of spliced RNA molecules. These alignment challenges are further compounded by the presence of multiple copies of identical or related genomic sequences that are themselves transcribed, making precise mapping difficult.

Various sequence alignment algorithms have been recently developed to tackle these challenges (Au, et al., 2010; De Bona, et al., 2008; Grant, et al., 2011; Han, et al., 2011; Trapnell, et al., 2009; Wang, et al., 2010; Wu and Nacu, 2010; Zhang, et al., 2012). However, application of these algorithms invokes compromises in the areas of mapping accuracy (sensitivity and precision) and computational resources (run time and disk space) (Grant, et al., 2011). With current advances in sequencing technologies, the computational component is increasingly becoming a throughput bottleneck. High mapping speed is especially important for large consortia efforts such as ENCODE, which continuously generate large amounts of sequencing data.

Furthermore, most of the cited algorithms were designed to deal with relatively short reads (typically less than or around 200 bases), and are ill-suited for aligning long read sequences generated by the emerging third-generation sequencing technologies (Flusberg, et al., 2010; Rothberg, et al., 2011). The longer read sequences, ideally reaching full lengths of RNA molecules, have a great potential for enhancing transcriptome studies by providing more complete RNA connectivity information.

This report describes an alignment algorithm entitled Spliced Transcripts Alignment to a Reference (STAR), which was designed to specifically address many of the challenges of RNA-seq data mapping, and utilizes a novel strategy for spliced alignments. We performed high throughput validation experiments which corroborated STAR's precision for detection of novel splice junctions. STAR's high mapping speed and accuracy were crucial for analyzing the large ENCODE transcriptome (Djebali, et al.,

2012) dataset (exceeding 80 billion Illumina reads). We also demonstrated that STAR has a potential for accurately aligning long (several kilo-bases) reads that are emerging from the third-generation sequencing technologies.

2 ALGORITHM

Many previously described RNA-seq aligners were developed as extensions of contiguous (DNA) short read mappers, which were used to either align short reads to a database of splice junctions, or align split read portions contiguously to a reference genome, or a combination thereof. In contrast to these approaches, STAR was designed to align the non-contiguous sequences directly to the reference genome. STAR algorithm consists of two major steps: seed searching step and clustering/stitching/scoring step.

2.1 Seed search

The central idea of STAR seed finding phase is the sequential search for a Maximal Mappable Prefix (*MMP*). *MMP* is similar to the Maximal Exact (Unique) Match concept used by the large-scale genome alignment tools Mummer (Delcher, et al., 1999; Delcher, et al., 2002; Kurtz, et al.) and MAUVE (Darling, et al., 2004; Darling, et al., 2010). Given a read sequence *R,* read location *i* and a reference genome sequence *G*, the $MMP(R,i,G)$ is defined as the longest substring $[R_i , R_{i+1} , \dots , R_{i+MML-1}]$ which matches exactly one or more substrings of *G*, where *MML* is the maximum mappable length. We will explain this concept using a simple example of a read which contains a single splice junction and no mismatches (Fig. 1a). In the first step the algorithm finds the *MMP* starting from the first base of the read. Since the read in this example comprises a splice junction, it cannot be mapped contiguously to the genome, and thus the first seed will be mapped to a donor splice site. Next, the *MMP* search is repeated for the unmapped portion of the read, which, in this case, will be mapped to an acceptor splice site. Note that this sequential application of *MMP* search only to the unmapped portions of the read makes STAR algorithm extremely fast and distinguishes it from Mummer and MAUVE which find all possible Maximal Exact Matches. This approach represents a natural way of finding precise locations of splice junctions in a read sequence and is advantageous over an arbitrary splitting of read sequences used in the split-read methods. The splice junctions are detected in a single alignment pass without any *a priori* knowledge of splice junctions loci or properties, and without a preliminary contiguous alignment pass needed by the junction database approaches. The *MMP* in STAR search is implemented through un-compressed suffix arrays (Manber and Myers, 1993). Notably, finding *MMP* is an inherent outcome of the standard binary string search in un-compressed suffix arrays (SA), and does not require any additional computational effort compared to the full length exact match searches. The binary nature of the SA search results in a favorable logarithmic scaling of the search time with the reference genome length, allowing very fast searching even against large genomes. Advantageously, for each *MMP* the SA search can find all distinct exact genomic matches with little computational overhead, which facilitates an accurate alignment of the reads that map to multiple genomic loci ("multi-mapping" reads).

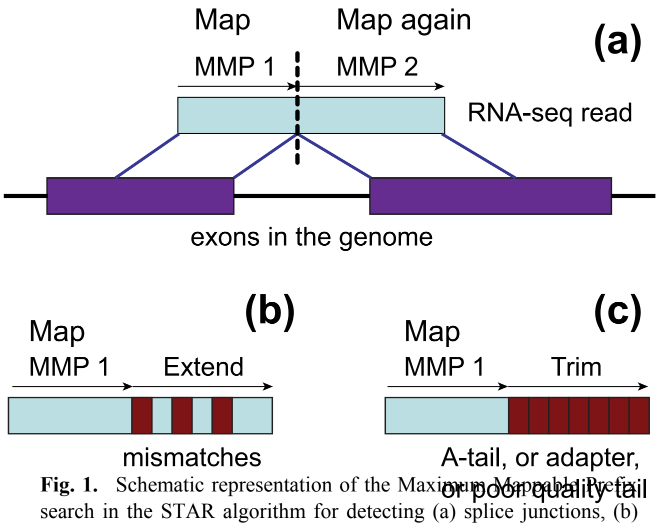


Fig. 1. Schematic representation of the Maximum Mappable Prefix search in the STAR algorithm for detecting (a) splice junctions, (b) mismatches, (c) tails

In addition to detecting splice junctions, the *MMP* search, implemented in STAR, enables finding multiple mismatches and indels, as illustrated in Fig. 1b. If the *MMP* search does not reach the end of a read because of the presence of one or more mismatches, the *MMP*s will serve as anchors in the genome that can be extended allowing for alignments with mismatches. In some cases the extension procedure does not yield a good genomic alignment, which allows identification of poly-A tails, library adapter sequences, or poor sequencing quality tails (Fig. 1c). The *MMP* search is performed in both forward and reverse directions of the read sequence, and can be started from user-defined search start points throughout the read sequence, which facilitates finding anchors for reads with errors near the ends and improves mapping sensitivity for high sequencing error rate conditions.

Besides the efficient MMP search algorithm, uncompressed suffix arrays also demonstrate a significant speed advantage over the compressed suffix arrays implemented in many popular short read aligners (see SM-1.8). This speed advantage is traded off against the increased memory usage by uncompressed arrays, which is assessed further in Section 3.3.

2.2 Clustering, stitching and scoring

In the second phase of the algorithm, STAR builds alignments of the entire read sequence by stitching together all the seeds that were aligned to the genome in the first phase. First, the seeds are clustered together by proximity to a selected set of "anchor" seeds. We found that an optimal procedure for anchor selection is through limiting the number of genomic loci the anchors align to. All the seeds that map within user-defined genomic windows around the anchors are stitched together assuming a local linear transcription model. The size of the genomic windows determines the maximum intron size for the spliced alignments. A frugal dynamic programming algorithm (see SM-1.5 for details) is used to stitch each pair of seeds allowing for any number of mismatches but only one insertion or deletion (gap).

Importantly, the seeds from the mates of paired-end RNA-seq reads are clustered and stitched concurrently, with each paired-end read represented as a single sequence, allowing for a possible genomic gap or overlap between the inner ends of the mates. This is a principled way to utilize the paired-end information as it reflects better the nature of the paired-end reads, namely, the fact that the mates are pieces (ends) of the same sequence. This approach increases the sensitivity of the algorithm, since only one correct anchor from one of the mates is sufficient to accurately align the whole read.

If an alignment within one genomic window does not cover the entire read sequence, STAR will try to find two or more windows that cover the whole read, resulting in a chimeric alignment with different parts of the read mapping to distal genomic loci, or different chromosomes, or different strands (see Fig. S-1). STAR can find chimeric alignments in which the mates are chimeric to each other, with a chimeric junction located in the un-sequenced portion of the RNA molecule between two mates. STAR can also find chimeric alignments in which one or both mates are internally chimerically aligned, thus pinpointing the precise location of the chimeric junction in the genome. An example of the BCR-ABL fusion transcript detection from the K562 erythroleukemia cell line is given in the Supplementary Materials SM-1.7 (Fig. S-2).

The stitching is guided by a local alignment scoring scheme, with user-defined scores (penalties) for matches, mismatches, insertions, deletions, and splice junction gaps, allowing for a quantitative assessment of the alignment qualities and ranks (see SM-1.4 for details). The stitched combination with the highest score is chosen as the best alignment of a read. For multi-mapping reads, all alignments with scores within a certain user-defined range below the highest score are reported.

Although the sequential *MMP* search only finds the seeds exactly matching the genome, the subsequent stitching procedure is capable of aligning reads with a large, scalable with the read length number of mismatches, indels and splice junctions. This characteristic has become ever more important with the emerging of the third generation sequencing technologies (such as Pacific Biosciences or Ion Torrent) that produce longer reads with elevated error rates.

3 RESULTS

3.1 Performance on simulated RNA-seq data

First we used simulated data to evaluate performance of STAR and compare it to other RNA-seq mappers. Simulations allow for a precise calculation of false positive and negative rates, even though artificial error models, used to generate simulated reads, may not adequately represent experimental errors. We used a simulated dataset from a recent study (Grant, et al., 2011) et al. 2011), in which 10 million of 2x100nt Illumina-like read sequences with a reasonably high error rate were generated from the mouse transcriptome, including annotated transcripts as well as artificial ones. Various types of genomic variations and sequencing errors were introduced to mimic real RNA-seq data.

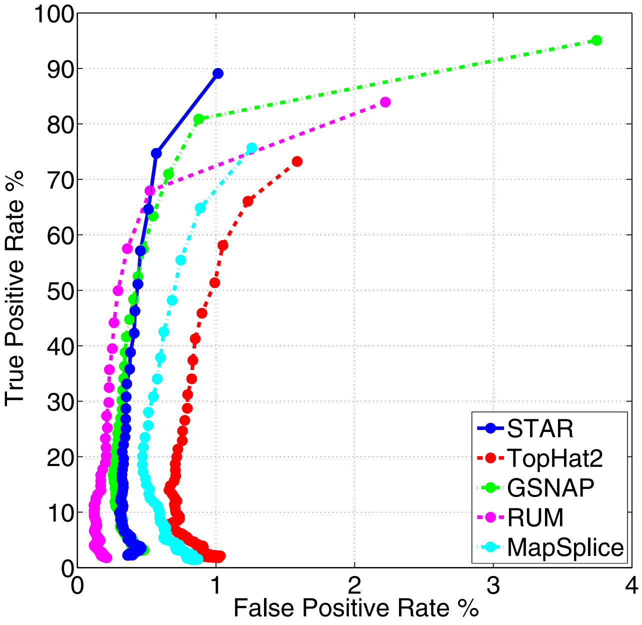


Fig. 2 True positive rate vs. false positive rate (ROC-curve) for simulated RNA-seq data for STAR, TopHat2, GSNAP, RUM and MapSplice.

The latest available versions of STAR 2.1.3, TopHat2 2.0.0 (Trapnell, et al., 2009), GSNAP 2012-07-03 (Wu and Nacu, 2010), RUM 1.11 (Grant, et al., 2011) and MapSplice 1.15.2 (Wang, et al., 2010) were run on the simulated dataset labeled as “SIM1-TEST2” in (Grant, et al., 2011). Since the TopHat2 2.0.0 release represents a major new development of the TopHat aligner, which has not been peer reviewed yet, we also made the comparisons with the previous TopHat version 1.4. We found that the new version yields a slightly better accuracy and faster mapping speed (see SM-2.1, Fig.S-3). All aligners were run in the *de novo* mode, i.e. without using gene/transcript annotations. The maximum number of mismatches was set at 10 per paired-end read, the minimum/maximum intron sizes were set at 20b/500kb (see SM-2 for additional information). Note, that running comparison between mappers with their default parameters is a reasonable and commonly accepted practice, because all considered aligners were, by default, optimized for mammalian genomes and recent RNA-seq data.

The resulting alignments were compared against the true genomic origin of the simulated reads, and true/false positive rates of splice junction detection were calculated using procedures and scripts developed by (Grant, et al., 2011). ROC curves (Fig. 2), were computed with the detection (discrimination) threshold given by the number of reads mapped across each junction, i.e. for each aligner only junctions supported by at least N reads were selected for each point along the ROC curves, with N varied from 1 (lowest threshold) to 100 (high threshold). All aligners exhibit desirable steep ROC curves at high values of detection threshold. At the

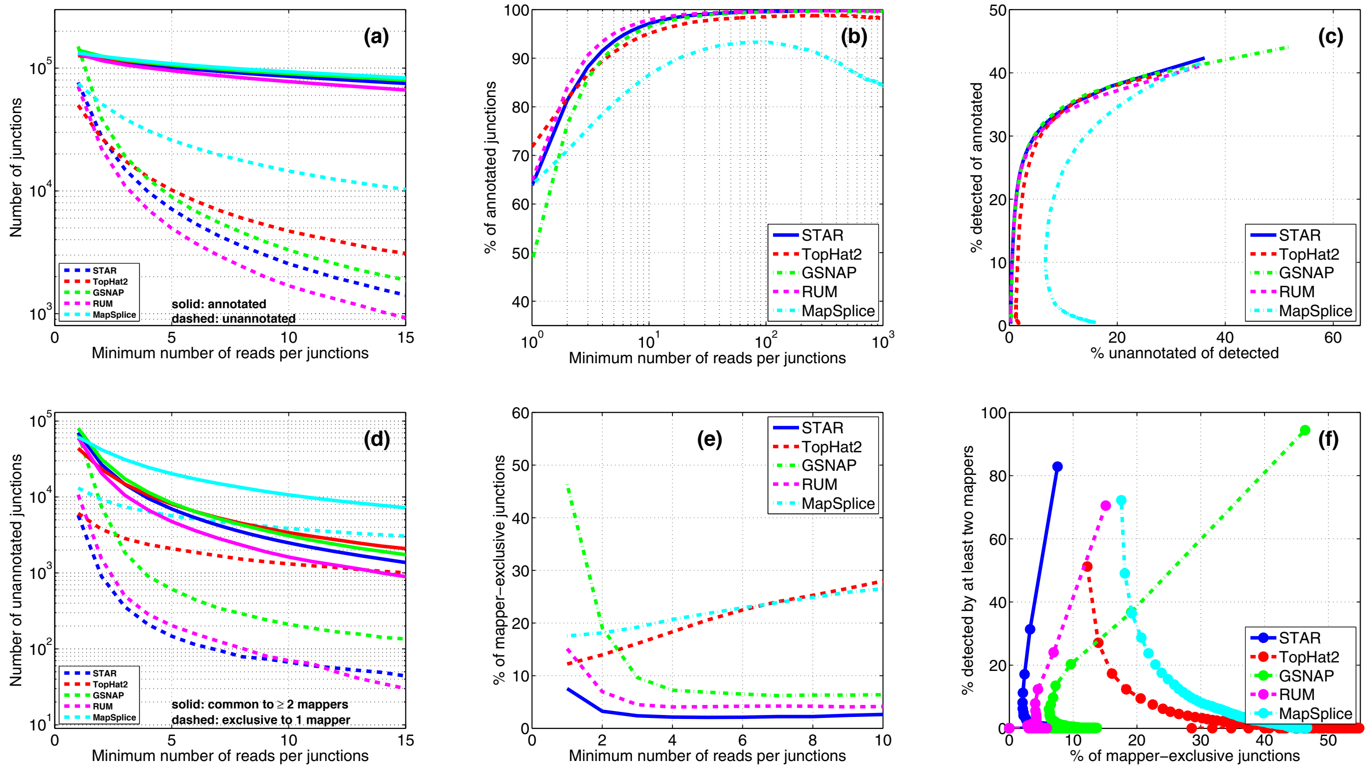


Fig. 3 Various accuracy metrics for splice junction detection in the experimental RNA-seq data. The color-coding scheme for mappers is the same in all plots. X-axis in plots (a), (b), (d) and (e) is the detection threshold defined as the number of reads mapped across each junction, i.e. each point with the X-value of N represents all junctions that are supported by at least N reads mapped by a given aligner. (a) Total number of detected junctions, annotated (solid lines) and unannotated (dashed lines); (b) percentage of detected junctions that are annotated; (c) pseudo-ROC curve: percentage of all annotated junctions that are detected vs. percentage of detected junctions that are unannotated; (d) number of unannotated junctions detected by at least two mappers (solid lines) and number of unannotated junctions detected exclusively by only one mapper (dashed lines); (e) percentage of detected unannotated junctions that are detected exclusively by only one mapper; (f) pseudo-ROC curve: percentage of unannotated junctions that are detected by at least two mappers vs. percentage of detected unannotated junctions that are detected exclusively by only one mapper.

lowest detection threshold of 1 read per junction, STAR exhibits the lowest false positive rate while achieving high sensitivity. Supplementary Fig. S-5 shows the same analysis for a low error rate simulated dataset, which yields similar conclusions.

3.2 Performance on experimental RNA-seq data

For evaluation of the RNA-seq mappers' performance on experimental RNA-seq data STAR, TopHat2, GSNAP, RUM and MapSplice were run (see SM-2 for additional information) on an ENCODE long RNA-seq dataset (K562 whole cell A+ sample, 1 Illumina GAIIx lane of 40 million 2x76 reads). STAR and GSNAP aligned the largest percentage of reads (94% both), followed by RUM (86%), MapSplice (85%) and TopHat2 (71%).

Different accuracy metrics for splice junction detection with respect to the Gencode 7 (Harrow, et al., 2012) annotations are plotted in Fig. 3a-c as a function of the detection threshold, defined as the minimum number of RNA-seq reads per junction. While all aligners detect a similar number of annotated junctions (Fig. 3a, solid lines), there are noticeable differences between mappers in the number of detected unannotated junctions (Fig. 3a, dashed lines). The percentage of the unannotated among all detected junctions is plotted in Fig. 3b as a function of detection threshold. Since all aligners show similar sensitivities to annotated junctions, the proportion of annotated among all detected junctions may serve as a surrogate of precision. STAR, RUM and TopHat2 perform similarly, while GSNAP exhibits lower precision at lower detection threshold, and MapSplice shows unusual non-monotonic and non-saturating behavior which was also noted in (Zhang, et al., 2012). Pseudo-ROC curve, i.e. the proportion of annotated junctions that are detected (pseudo-sensitivity) vs. the proportion of detected junctions that are unannotated (pseudo-false positive rate) is plotted in the Fig. 3c. All aligners (except MapSplice) perform similarly at high values of the detection threshold.

Since many unannotated junctions represent true novel splicing events and are not false positives, the percentage of unannotated among all detected junctions is not a very accurate proxy for the false positive rate. To obtain a more accurate estimate of the false positive rate, we followed another frequently used approach (Zhang, et al., 2012) and plotted (Fig. 3d) the number of junctions detected by at least two mappers (pseudo-true positive) and the number of junctions detected exclusively by each mapper (pseudo-false positive). STAR alignments yield the lowest pseudo-false positive rate, i.e. the lowest proportion of exclusively detected

Table 1. Mapping speed and RAM benchmarks on the experimental RNA-seq dataset.

Aligner	Mapping speed: Million read pairs / hour		Peak physical RAM, GB	
	6 threads	12 threads	6 threads	12 threads
STAR	309.2	549.9	27.0	28.4
STAR sparse	227.6	423.1	15.6	16.0
TopHat2	8.0	10.1	4.1	11.3
RUM	5.1	7.6	26.9	53.8
MapSplice	3.0	3.1	3.3	3.3
GSNAP	1.8	2.8	25.9	27.0

junctions (Fig. 3e), while at the same time achieving the second in class pseudo-sensitivity (Fig. 3f). GSNAP exhibits the highest pseudo-sensitivity at the cost of high pseudo-false positive rate. These results qualitatively agree with the aligners' performance on the simulated data, whereas the quantitative differences may be attributed to disparities between real and simulated errors. Supplementary Fig. S-6 shows the same analysis for shorter RNA-seq dataset (2x50b), which indicates that STAR retains high sensitivity and precision even for short reads.

Note that the pseudo-true/false positive definitions are based on the assumption that junctions detected by only one aligner are more likely to be false positive than the junctions detected by two or more aligners; however, these definitions are not rigorous since the true/false assessments cannot be made for experimental data. We would also like to stress that these comparisons were done for current versions of each tool, with the default parameters and for the present state of Illumina sequencing technology. As both sequencing technologies and tools improve, these rankings may change and have to be reevaluated.

Similarly to other RNA-seq aligners, STAR's default parameters are optimized for mammalian genomes. Other species may require significant modifications of some alignment parameters; in particular, the maximum and minimum intron sizes have to be reduced for organisms with smaller introns.

3.3 Speed benchmarks

Speed benchmarks were performed on a server equipped with two 6-core Intel Xeon CPUs X5680@ 3.33GHz, 148GB of RAM and RAID-5 array of six 2TB 7200rpm SATA hard drives. 6 or 12 threads were requested for each run, utilizing half or full capacity of the server. All mappers were run with their default parameters on the ~40 Million 2x76 Illumina human RNA-seq dataset described in the previous section.

The "wall" time (i.e. the total run time required to complete the mapping) and RAM usage are presented in Table 1. STAR achieves a speed of 550 million 2x76 Illumina paired-end reads per hour using 12 threads (full capacity of the server), i.e. 45 Million paired reads per hour per processor, outperforming the second fastest mapper (TopHat2) by more than a factor of 50. STAR exhibits close to linear scaling of the throughput rate with the number of threads, losing ~10% of per thread mapping speed when the number of threads is increased from 6 to 12.

STAR's high mapping speed is traded off against RAM usage: STAR requires ~27GB of RAM for aligning to the human genome. Like all other aligners, with the exception of RUM, the amount of RAM used by STAR does not increase significantly with the number of threads, because the suffix array is shared among all threads. Although STAR's RAM requirements would have been prohibitively expensive several years ago, at the time when the first short read aligners were developed, recent progress in semiconductor technologies resulted in a substantial drop of RAM prices, and modern high performance servers are commonly equipped with RAM exceeding 32GB. STAR has an option to utilize sparse suffix arrays reducing the RAM consumption to below 16GB for the human genome at the cost of ~25% decrease in the mapping speed, while maintaining the alignment accuracy.

3.4 Experimental validation

As part of the characterization of human transcriptome by the ENCODE (Djebali, et al., 2012), STAR was used to map polyadenylated (poly A+) long (>200nts) transcripts isolated from whole cell extracts of primary human H1ES and HUVEC cell lines. These RNAs were sequenced using a Duplex-Specific Nuclease protocol (Parkhomchuk, et al., 2009) that generated 2x76bp strand-specific reads.

Not surprisingly, unannotated (novel) splice sites show lower abundance levels than the annotated junctions, as indicated by the significant drop in the number of unannotated junctions with the number of supporting reads (see Fig. S-7). Since each of the cell lines was sequenced in biological duplicates, a collection of high confidence splice sites could be identified based on their reproducibility between replicas. To assess the reproducibility of the detected splice junctions, we developed a non-parametric Irreproducible Discovery Rate (npIDR) approach, specifically suitable for the discrete nature of the RNA-seq data (see Supplementary Materials for the detailed description). This approach is similar to the Irreproducibility Discovery Rate concept extensively used in the analysis of the ENCODE ChIP-seq experiments (Landt, et al., 2012). Fig. S-8 shows the dependence of npIDR=0.1 on the read count per junction, providing a principled method for selecting the read count threshold with a desired level of reproducibility. For example, five staggered reads per junction are required to achieve npIDR of 0.1, i.e. the 90% likelihood that these junctions will be observed again in another experiment on the same cell line with the same sequencing depth.

Experimental validation was carried out on 1,920 novel splice junctions in a wide range of RNA-seq reads support, both below and above the npIDR threshold. Only splice junctions mapped to intergenic or antisense loci to Gencode 7 genes (Harrow, et al., 2012) were chosen for validation, since these junc-

Table 2. Number of selected junctions and percentage of selected junctions that were validated by at least two 454 reads, as a function of the RNA-seq read count per junction.

H1ES			HUVEC		
Read count per junction from two replicates	Number of tested junctions	Proportion of junctions validated by at least two 454 reads	Read count per junction from two replicates	Number of tested junctions	Proportion of junctions validated by at least two 454 reads
2	192	72.4%	2	192	74.0%
3	192	77.6%	3	192	75.0%
4	96	74.0%	4	96	76.0%
5	96	82.3%	5-6	96	84.4%
6-7	96	79.2%	7-8	96	84.4%
8-11	96	81.3%	9-12	96	86.5%
12-24	96	87.5%	13-23	96	94.8%
>=25	96	88.5%	>=24	96	90.6%

tions are more likely to be false positive than the junctions that map within the annotated genes. The high-throughput validation pipeline involved RT-PCR amplification of targeted regions followed by Roche 454 sequencing of the pooled products. The RT-PCR primer design took advantage of the ~250nt insert length of the paired-end reads supporting targeted junctions, and entailed the production of long 300 to 600nt amplicons. These amplicons were pooled and sequenced by a Roche 454 sequencer to provide long and more confidently mappable reads that were aligned to the genome with BLAT. Detailed description of the experimental protocols can be found in (Djebali, et al., 2012).

We selected 1,920 intergenic and antisense splice junctions from H1ES and HUVEC cell lines, including both highly (npIDR<0.1) and poorly (npIDR>0.1) reproducible junctions. Approximately 82% to 89% (H1ES) and 84% to 95% (HUVEC) of all the tested novel intergenic/antisense junctions supported by at least five RNA-seq reads (corresponding to npIDR<0.1) were corroborated by at least two amplicons sequenced by 454 (see Table 2). Notably, the validation rate remains at a high level of 72% (H1ES) and 74% (HUVEC) even for the candidate junctions that were supported by as few as two RNA-seq reads. These results confirm high precision of the STAR's splicing detection algorithm even for rare novel junctions.

The upper bound of the False Discovery Rate (FDR) can be estimated from the validation rate (≡VR) as FDR≤1-VR. For low abundance junctions the experimental FDR is lower than the npIDR predicted from the dissimilarity between the replicates: for example, even though 45% of junctions, supported by just two reads, are not reproducible (Fig. S-8), more than 70% of them are successfully validated (Table 2). Hence, npIDR can serve as a conservative upper bound FDR estimate in cases where validation experiments are impractical.

4 DISCUSSION

Despite several years of ongoing improvements, alignment of the non-contiguous RNA-seq reads to a reference genome is not a solved problem yet, owing both to its intrinsic complexity and rapid transformations of the sequencing technologies. Several critical problems have been found to afflict previously published approaches, such as high mapping error rate for contiguous and especially spliced alignments; mapping biases whereas some types of alignments are artificially favored over others; low sensitivity for un-annotated transcripts; poor scalability with the read length; restrictions in the number of junctions, mismatches and indels per read; inability to detect non-linear transcripts such as chimeric RNAs; and, crucially, very high requirements on computational resources owing to the low mapping throughput.

In this work we described STAR, a novel algorithm for aligning high-throughput long and short RNA-seq data to a reference genome, developed to overcome the aforementioned issues. Unlike many other RNA-seq mappers, STAR is not an extension of a short-read DNA mapper, but was developed as a stand-alone C++ code. STAR is capable of running parallel threads on multi-core systems with close to linear scaling of productivity with the number of cores. STAR is very fast: on a modern but not overly expensive 12-core server it can align 550 million 2x76nt reads per hour to the human genome, surpassing all other existing RNA-seq aligners by a factor of 50. At the same time, STAR exhibits better alignment precision and sensitivity than other RNA-seq aligners for both experimental and simulated data.

One of the main inherent problems of all *de novo* RNA-seq aligners is the inability to accurately detect splicing events which involve short (<5-10nt) sequence overhangs on the donor or acceptor sides of a junction. This causes a significant under-detection of splicing events, and also increases significantly the misalignment rate, since such reads are likely to be mapped with a few mismatches to a similar contiguous genomic region. In addition, this effect also biases the alignments towards processed pseudogenes which are abundant in the human genome. Similarly to other RNA-seq aligners, to mitigate this problem STAR has an option to obtain information about possible splice junction loci from annotation databases (see SM-4). It is also possible to run a 2nd mapping pass supplying it with splice junctions loci found in the 1st mapping pass. In this case STAR will not discover any new junctions, but will align spliced reads with short overhangs across the previously detected junctions.

To demonstrate STAR's ability to align long reads we have mapped the long (0.5-5kb) human mRNA sequences from GenBank (see SM-5 for details). The accuracy of STAR alignments is similar or higher than that of BLAT, a popular EST/mRNA aligner. At the same time, STAR outperforms BLAT by more than two orders of magnitude in the alignment speed, which is important for high-throughput sequencing applications.

The algorithm extensibility to long reads shows that STAR has a potential to serve as a universal alignment tool across a broad spectrum of emerging sequencing platforms. STAR can align reads in a continuous streaming mode which makes it compatible with novel sequencing technologies such as the one recently announced by Oxford Nanopore Technologies. As the sequencing technologies and protocols evolve, new mapping strategies will have to be developed and STAR core algorithm can provide a flexible framework to address arising alignment challenges.

FUNDING

This work was funded by NHGRI grant U54HG004557.

DATA ACCESS

GEO: GSE38886 (Roche 454 sequencing)
GEO: GSE30567 (Illumina long RNA-Seq)

REFERENCES

Au, K.F., *et al.* (2010) Detection of splice junctions from paired-end RNA-seq data by SpliceMap, *Nucleic Acids Res*, **38**, 4570-4578.

Darling, A.C., *et al.* (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements, *Genome Research*, **14**, 1394-1403.

Darling, A.E., Mau, B. and Perna, N.T. (2010) progressiveMauve: multiple genome alignment with gene gain, loss and rearrangement, *PLoS One*, **5**, e11147.

De Bona, F., *et al.* (2008) Optimal spliced alignments of short sequence reads, *Bioinformatics*, **24**, i174-180.

Delcher, A.L., *et al.* (1999) Alignment of whole genomes, *Nucleic Acids Res*, **27**, 2369-2376.

Delcher, A.L., *et al.* (2002) Fast algorithms for large-scale genome alignment and comparison, *Nucleic Acids Res*, **30**, 2478-2483.

Djebali, S., *et al.* (2012) Landscape of transcription in human cells, *Nature*, **489**, 101-108.

Flusberg, B.A., *et al.* (2010) Direct detection of DNA methylation during single-molecule, real-time sequencing, *Nat Methods*, **7**, 461-465.

Grant, G.R., *et al.* (2011) Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM), *Bioinformatics*, **27**, 2518-2528.

Han, J., *et al.* (2011) Pre-mRNA splicing: where and when in the nucleus, *Trends Cell Biol*, **21**, 336-343.

Harrow, J., *et al.* (2012) GENCODE: The reference human genome annotation for The ENCODE Project, *Genome Research*, **22**, 1760-1774.

Hastings, M.L. and Krainer, A.R. (2001) Pre-mRNA splicing in the new millennium, *Curr Opin Cell Biol*, **13**, 302-309.

Kurtz, S., *et al.* (2004) Versatile and open software for comparing large genomes, *Genome Biol*, **5**, R12.

Landt, S.G., *et al.* (2012) ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia, *Genome Research*, **22**, 1813-1831.

Manber, U. and Myers, G. (1993) Suffix Arrays - a New Method for Online String Searches, *Siam J Comput*, **22**, 935-948.

Parkhomchuk, D., *et al.* (2009) Transcriptome analysis by strand-specific sequencing of complementary DNA, *Nucleic Acids Res*, **37**, e123.

Rothberg, J.M., *et al.* (2011) An integrated semiconductor device enabling non-optical genome sequencing, *Nature*, **475**, 348-352.

Trapnell, C., Pachter, L. and Salzberg, S.L. (2009) TopHat: discovering splice junctions with RNA-Seq, *Bioinformatics*, **25**, 1105-1111.

Wang, K., *et al.* (2010) MapSplice: accurate mapping of RNA-seq reads for splice junction discovery, *Nucleic Acids Res*, **38**, e178.

Wu, T.D. and Nacu, S. (2010) Fast and SNP-tolerant detection of complex variants and splicing in short reads, *Bioinformatics*, **26**, 873-881.

Zhang, Y., *et al.* (2012) PASSion: a pattern growth algorithm-based pipeline for splice junction detection in paired-end RNA-Seq data, *Bioinformatics*, **28**, 479-486.

RNA-STAR: ultrafast universal spliced sequences aligner: Supplementary materials

Alexander Dobin[1], Carrie A. Davis[1], Felix Schlesinger[1], Jorg Drenkow[1], Chris Zaleski[1], Sonali Jha[1], Philippe Batut[1], Mark Chaisson[2] and Thomas R. Gingeras[1]

[1]Cold Spring Harbor Laboratory, Cold Spring Harbor, NY, USA.

[2]Pacific Biosciences, Menlo Park, California, USA.

Corresponding author: Alexander Dobin

1 Bungtown Rd, Cold Spring Harbor, NY 11724

Email: dobin@cshl.edu

Phone: 516-422-4123

1. STAR algorithm details

1.1. Suffix array search against a reference genome

Suffix Array (SA) of the whole genome is utilized to find the Maximum Mappable Prefixes (MMP). The MMP search is originated at 5' of the reads, and also at arbitrary user defined positions along the read. All the possible alignments with the length equal to Maximum Mappable Length (MML) are collected, which allows a comprehensive alignment of multi-mappers. If the MMP does not cover the whole read, the remaining unmapped portion is aligned again using the same procedure, continuing until the end of the read sequence. The whole procedure is performed in both 5' to 3' and 3' to 5' directions. Suffix Array is generated prior to the alignment and stored on disk. Before the alignment begins, SA and genome sequence are loaded into RAM and are stored in the Linux shared memory, allowing access from multiple processes (threads). The SA contains both the positive and negative strand of the genome. In case of the genomes larger than 2 Gigabases, the SA indices require fractional bytes, for example, for genomes 2 to 4 Gigabase-long, each SA index occupies 33 bits.

1.2. Pre-indexing of suffix arrays

While suffix array search is theoretically fast owing to its binary nature, in practice it may suffer from non-locality resulting in persistent cache misses which deteriorate the performance. To alleviate this problem we developed a pre-indexing strategy. After the SA is generated, we find the locations of all possible L-mers in the SA, $L<=L_{max}$, where L_{max} is user defined and is typically 12-15. Since the nucleotide alphabet contains only four letters, there are $N_L=2^{2L}$ different L-mers for which the SA locations have to be stored. For example, if $L=L_{max}=14$, N_L~268M and for 33-bit SA indices it will require 1GB of storage. All L-mers with $L<L_{max}$ will require 1/3 more of storage space. Using the L-mer indices we can immediately bound each search in the SA for all strings longer than L_{max} , and obtain the complete answer for all strings shorter than L_{max}. This procedure makes the SA search more local and speeds it up by a factor of 2-4.

1.3. Anchors and alignment windows

The SA search (step 1) yields a collection of alignments that cover all or just portions of the read, possibly multiple times. In the next step the "anchor" alignments are selected, defining the genomic regions to which the read is similar. In the current implementation, all the alignments that map less than a user defined value (typically 20-50) are selected as anchors. Alignment windows are genomic regions selected around the anchors. All the alignments, anchor and non-anchor, located within an alignment window will be stitched to each other in an attempt to find the best "linear" alignment (step 5). The genome is split into equally size bins, and all the anchor bins that are within a user defined distance to each other are lumped into one window. Alignment windows are necessary to include short pieces of the read which map too many times (and hence cannot be anchors) such as short donor/acceptor portions of splice junctions, or micro-exons.

1.4. Scoring scheme

Total score for each alignment is calculated as a sum of match scores, minus sum mismatch scores for mismatched bases, minus the penalties for insertions, deletions and genomic gaps:

$$S = +\sum_{match} P_m - \sum_{mismatch} P_{mm} - \sum_{inserion} P_{ins} - \sum_{deletion} P_{del} - \sum_{gap} P_{gap} .$$

In the present version of STAR matches and mismatches are scored as +/-1.

For short deletions and all insertions the penalty is a sum of user-defined indel opening penalty and indel extension penalty which proportional to the indel length:

$$P_{ins/del} = P_{ins/del}^{open} + P_{ins/del}^{extend} \cdot L_{ins/del}$$

Deletions that are longer than a user defined minimum intron size are considered splice junction (gaps), and their penalties consist of a constant gap opening penalty and a penalty which depends logarithmically on the gap length.

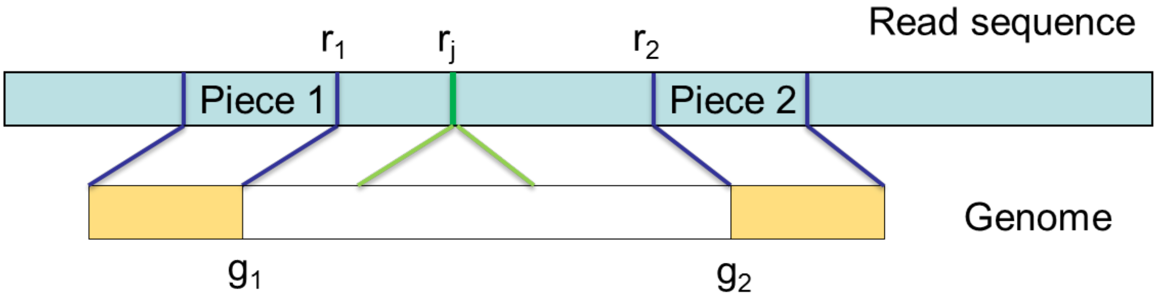
The gap opening penalties are user-defined and can be set independently for GT/AG, GC/AG, AT/AC and all other (non-canonical) motifs. The penalties for different intron motifs have to be selected according to the frequency expectations of different intron motifs in the species under study. The default penalties are adapted for the mammalian genomes, where the major canonical intron motif GT/AG dominates over all the others, followed by GC/AG, and by much less frequent AT/AC and other non-canonical motifs. Note that increasing the gap penalties biases the alignments towards un-spliced alignment with mismatches (for example, pseudogenes).

1.5. Stitching and extension

The mapped seeds within the windows selected in step 1.3 are stitched together into “transcripts” assuming a linear transcription model, i.e. the different blocks of the alignment do not overlap, and blocks that follow each other in the read sequence have to also follow each other in the genome. Two seeds are stitched together using a simple algorithm that allows for one genomic gap and several mismatches. The algorithm searches for the junction position in the read sequence $r_j$that yields the maximum score by finding the maximum of the following quantity:

$$\max_{r_1 < r_j < r_2} \left\{ \sum_{r=1}^{r_j - r_1} \begin{bmatrix} 1 & if\ R(r_1 + r) = G(g_1 + r)\ \&\ R(r_1 + r) \neq G(g_1 + r + \Delta) \\ -1 & if\ R(r_1 + r) \neq G(g_1 + r)\ \&\ R(r_1 + r) = G(g_1 + r + \Delta) \\ 0 & otherwise \end{bmatrix} - P_{gap}(r_j) \right\}$$

Where R and G are read (query) and genome sequences, coordinates r_1, r_2, g_1, g_2 are defined in the diagram below, $\Delta \equiv (g_2 - g_1) - (r_2 - r_1)$ is the alignment gap with the corresponding gap penalty $P_{gap}(r_j)$. The complexity of this algorithm is proportional to the number of unmapped query sequence bases between the mapped seeds, i.e. $r_2 - r_1 - 1$.



Note that current implementation traverses through all the possible paths within a window of aligned pieces, which can be clearly made more efficient by dynamic programming in the future releases. If necessary, the alignments are extended towards unmapped 5' and 3' end of the reads, using a simple algorithm stops the extension when the score reaches the maximum or there are too many mismatches.

1.6. Selecting the best alignments

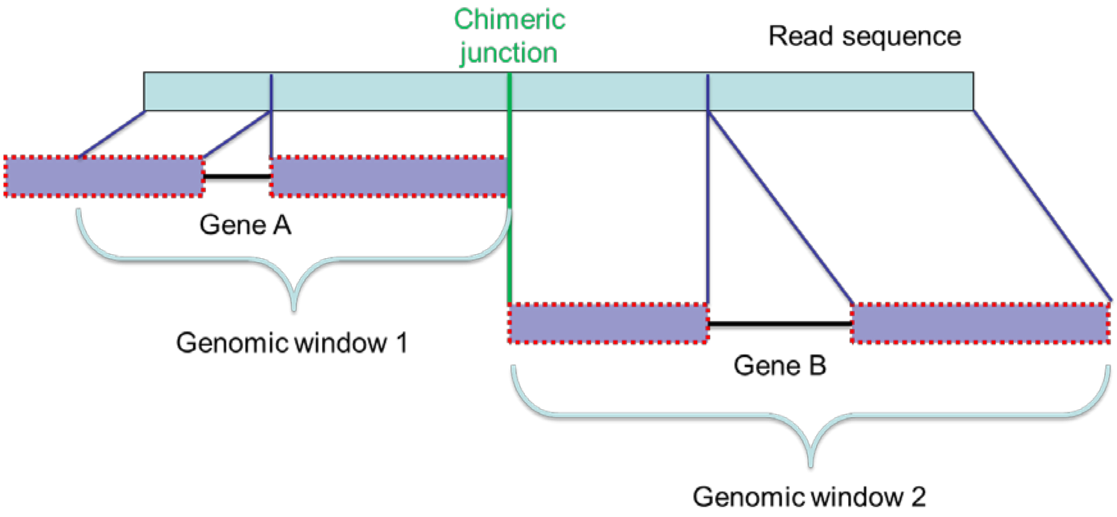
Alignments from all windows are collected and sorted by their score. All the alignments scored within a user-defined range of the maximum score are considered multi-mappers. Some additional user-configurable filtering can be done before the alignments are output.

1.7. Chimeric alignments

If the best scoring ("main") alignment window does not cover the entire read, we report chimeric connections to the other windows that cover portions of the read not covered by the main window. These chimeric connections between windows can span long distance on the same strand, or different strands on the same chromosome, or different chromosomes.

Figure S-1

A diagram illustrating detection of chimeric transcripts

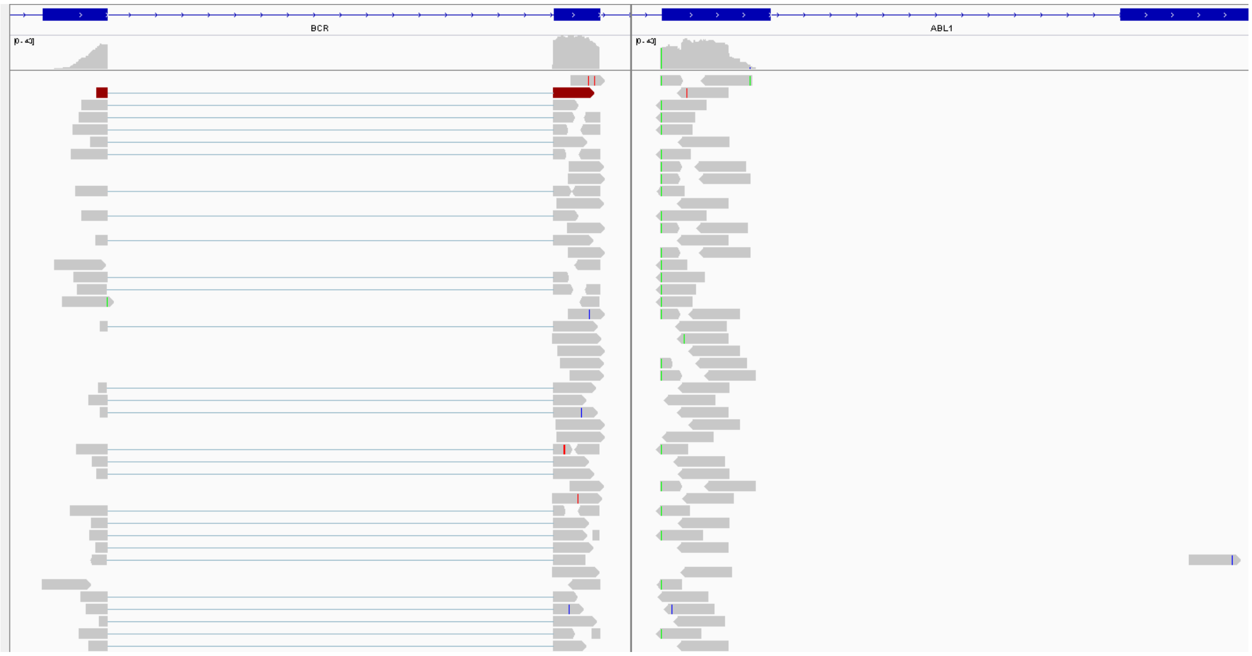


As an example of detecting a chimeric junction, we analyzed the chimeric reads detected by STAR in the ~40M 2x76 reads of K562 RNA-seq dataset used in the main text of the paper. STAR maps 55 reads to a very well-known inter-chromosomal fusion junction between BCR and ABL genes (see

Figure S-2). Some of the reads are aligned with the 1st mate entirely in the BCR and the 2nd mate entirely in ABL, while other reads cross the actual chimeric junction between the exons of the two genes.

Figure S-2

IGV browser snapshot of the BCR-ABL fusion junction with chimeric STAR alignments from K562 RNA-seq data. The panel on the left shows BCR gene locus, while the panel on the right shows ABL gene locus.



1.8. Comparison with the FM-BWT aligners

Many popular short read aligners (BWA, bowtie, Soap2) utilize a compressed form of the suffix arrays - the FM-index based on the Burrows-Wheeler transform. While the compression allows for significant reduction of the memory usage, it also results in diminished efficiency of the string search operations. We compared the performance of STAR and bowtie, short un-spliced reads aligner based on FM-BWT, for the simplest string search operation - exact matching of the reads to the reference genome.

We utilized the first mate sequences (76b) from our real RNA-seq dataset used in the main text. We aligned it to the human genome with bowtie requiring exact matches only (-v0) and at least two alignments (-k2). Because of the first limitation, as expected, bowtie could only align 53% of the reads - these reads map to the genome without mismatches, indels or splicing. These reads were extracted and aligned with both STAR and bowtie.

On this perfectly matching single-end read set, using the 1 thread, STAR aligns 976M reads per hour, compared to bowtie's speed of 154M reads per hour. This demonstrates a factor of ≈6 speed advantage of the uncompressed suffix arrays over the compressed BWT arrays for the exact string match search.

2. Simulated and experimental data analysis details

The maximum intron size in all aligners was set at 500kb. The minimum intron size was set at 20 for STAR, Mapsplice and Tophat (RUM and GSNAP do not allow setting this parameter). The maximum number of mismatches was set at 5 per mate for GSNAP and Mapsplice, 10 per paired-end read for STAR. RUM and Tophat do not allow setting the maximum number of mismatches.

Versions and command line arguments for all aligners are listed below:

STAR 2.1.2d

```
STAR --runThreadN <Nthreads> --genomeDir <genome_path>
--readFilesIn Read1.fastq Read2.fastq --alignIntronMin 20
--alignIntronMax 500000 -- outFilterMismatchNmax 10
```

GSNAP 2012-07-03

```
gsnap -B 5 -t <Nthreads>  -N 1  -A sam  --max-mismatches 5
--pairmax-rna 500000 -D <genome_path>  -d <genome_name> Read1.fastq
Read2.fastq
```

MapSplice 1.15.2

```
python2.6 mapsplice_segments.py  --threads <Nthreads> -u
Read1_mapsplice.fa,Read2_mapsplice.fa -c <chromosomes_path> -B
<genome_name> --min-intron-length 20 --max-intron-length 500000 -m 5
-o output_path paired.cfg
```

RUM 1.11

```
perl RUM_runner.pl <rum.config> Read1.fastq,,,Read2.fastq <out_dir>
<Nthreads> <out_prefix> -genome_only -maxIntron 500000
```

TopHat 2.0.0

```
tophat --solexa1.3-quals -p $1 -r172 --min-segment-intron 20 --max-
segment-intron 500000 --min-intron-length 20 --max-intron-length
500000 <genome_name> Read1.fastq Read2.fastq
```

Bowtie 2 was used as short read aligner for TopHat2.

When the default number of mismatches is used for GSNAP (i.e. the `--max-mismatches` is omitted), it uses an "ultrafast algorithm" and achieves higher speed (5M read pairs per hour for 6 threads, 8.6 read pairs per hour for 12 threads) and lower RAM usage (13GB).

For STAR, RUM and TopHat splice junctions were extracted from the junctions' files generated by the aligners. Because GSNAP does not generate a list of detected junctions, and MapSplices' list contained a large number of false positive junctions, we extracted GSNAP's and MapSplice's junctions from the their uniquely mapped alignments in .sam files using the (Grant, et al.)'s script *sam2junctions.pl*.

All junctions were quantified with the number of the reads crossing it.

The true junctions in the simulations were taken from (Grant, et al.)'s *simulated_reads_junctions-crossed_test1(2).txt* file. The annotated junctions for the experimental data analysis were extracted from Gencode 7 (Harrow, et al., 2012) annotations. The non-canonical junctions (i.e. other than GT/AG, GC/AG and AT/AC, and reverse complementary of those) were "flushed" to the left to avoid the micro-repeat ambiguity. The junctions predicted by the mappers were matched against the true junctions in the simulated data analysis, or to the annotated junctions for the experimental data analysis.

For the calculation of the percentage of mapped reads, we defined mapped reads as those which had one or more alignment with more than 80% mapped bases. We computed the number of mapped bases length as a sum of "M" values in the CIGAR strings of the .sam files.

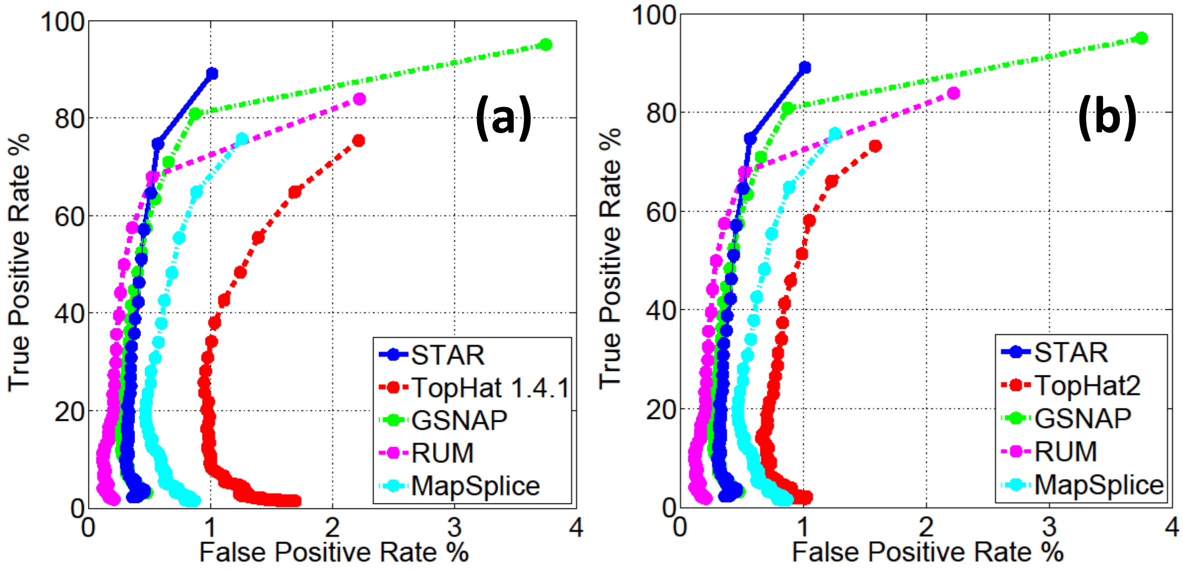
For speed benchmarking, the test were run using the Linux "virtual disk" device /dev/shm for all the input, output and temporary files to avoid hard drive bandwidth and latency issues.

2.1. TopHat 1.4.1 vs. TopHat2 2.0.0

We tested the TopHat 1.4.1, which is the last version before the TopHat2 release. The ROC curves for the simulated dataset SIM1_TEST2 are presented in Figure S-3:

Figure S-3

True positive rate vs. false positive rate (ROC-curve) for simulated RNA-seq data for STAR, TopHat, GSNAP, RUM and MapSplice. (a) TopHat 1.4.1; (b) TopHat2 2.0.0 (identical to Figure 2 of the main text).

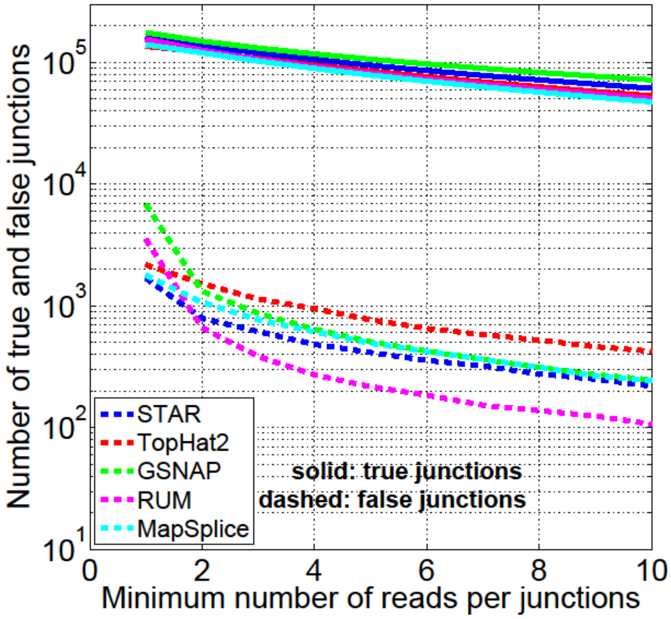


TopHat2's accuracy is improved significantly compared to TopHat 1.4.1. Moreover, the mapping speed of TopHat 2.0.0 has increased by ≈30%.

2.2. Number of predicted junctions for simulated dataset SIM1_TEST2

Figure S-4

Numbers of predicted true and false junctions as a function of read count per junction for the simulated dataset SIM1_TEST2 from (Grant, et al.). See Figure 2 of the main text for the ROC curve.



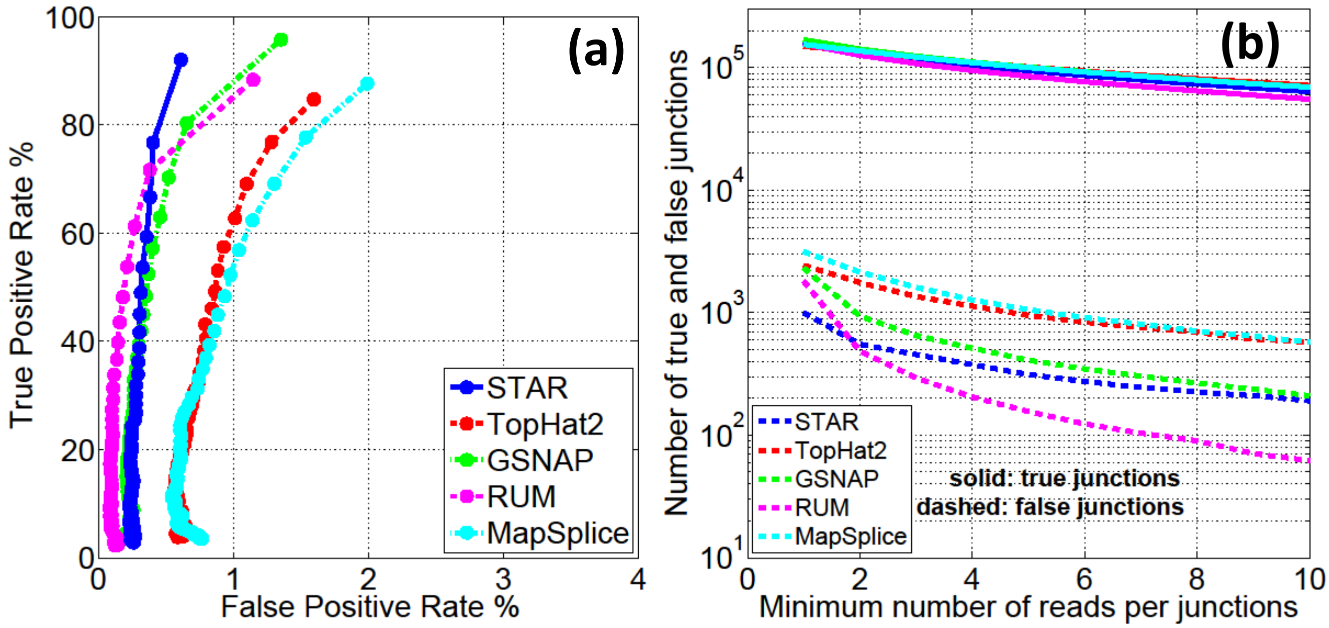
2.3. Simulated dataset SIM1_TEST1

In addition to the *SIM1_TEST2* simulated dataset (Grant, et al.) which was used in the Fig. 2 of the main text, we compared the mappers using a low-error-rate *SIM1_TEST1* dataset:

Figure S-5

(a) True positive rate vs. false positive rate (ROC-curve) for the simulated RNA-seq *SIM1_TEST1* from (Grant, et al.) for STAR, TopHat2, GSNAP, RUM and MapSplic.

(b) Numbers of predicted true and false junctions as a function of read count per junction for the simulated dataset *SIM1_TEST1* from (Grant, et al.).



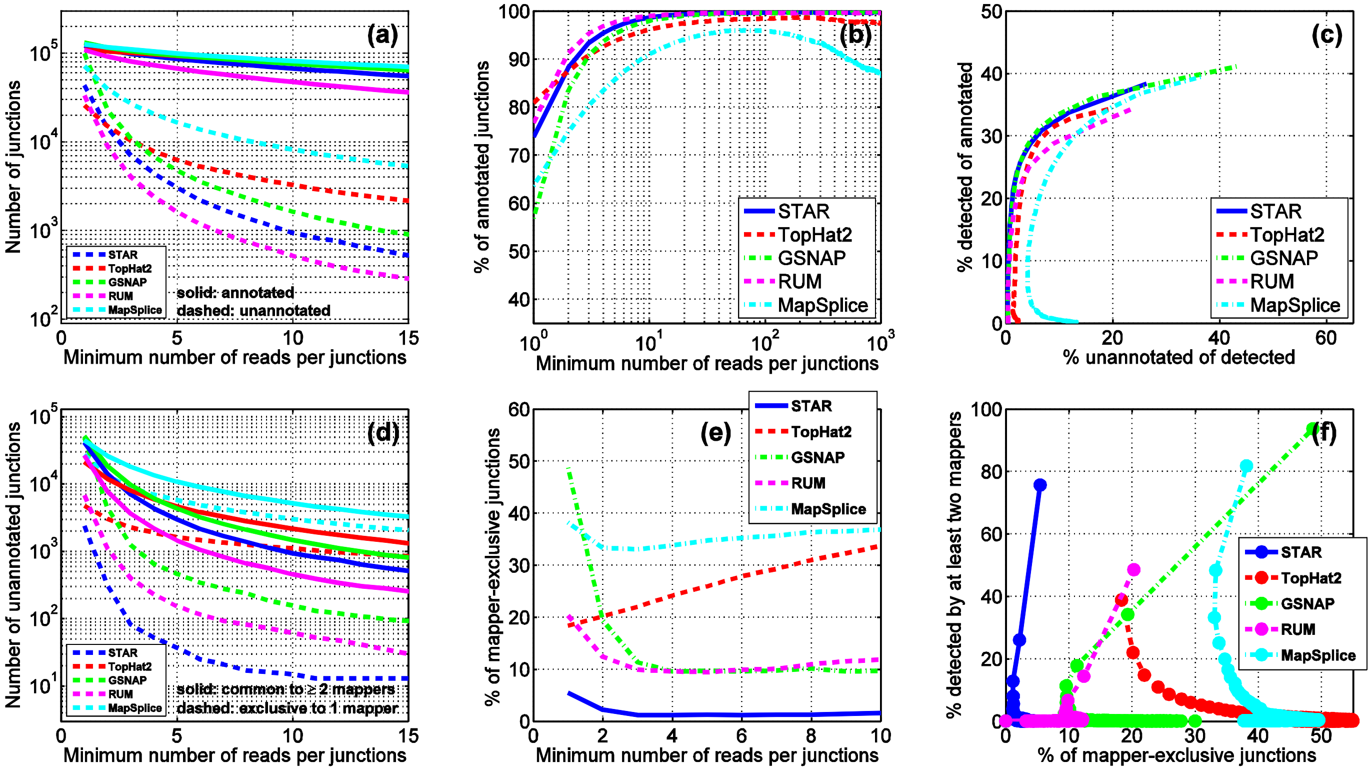
2.4. Experimental 2x50b RNA-seq data

Figure S-6.

Various accuracy metrics for splice junction detection in the experimental 2x50b RNA-seq data.

The 2x50b reads were obtained by trimming the ends of the 2x76b reads from experimental dataset used in the Section 3.2 of the main text (Fig. 2).

The color-coding scheme for mappers is the same in all plots. X-axis in plots (a), (b), (d) and (e) is the detection threshold defined as the number of reads mapped across each junction, i.e. each point with the X-value of N represents all junctions that are supported by at least N reads mapped by a given aligner. (a) Total number of detected junctions, annotated (solid lines) and unannotated (dashed lines); (b) percentage of detected junctions that are annotated; (c) pseudo-ROC curve: percentage of all annotated junctions that are detected vs. percentage of detected junctions that are unannotated; (d) number of unannotated junctions detected by at least two mappers (solid lines) and number of unannotated junctions detected exclusively by only one mapper (dashed lines); (e) percentage of detected unannotated junctions that are detected exclusively by only one mapper; (f) pseudo-ROC curve: percentage of unannotated junctions that are detected by at least two mappers vs. percentage of detected unannotated junctions that are detected exclusively by only one mapper.



3. Non-parametric Irreproducible Discovery Rate (npIDR)

npIDR ascertains reproducibility of the detection of genomic elements (such as splice junctions, exons, transcripts etc.) in RNA-seq experiment with biological replicates, referred to as 1 and 2 below. First, a common set of elements has to be created for the two bio-replicates. This can be a set of annotated elements, or a conjoint set of *de novo* detected elements from the two bio-replicates. Each of the elements in the common set is quantified with RNA-seq reads separately against each bio-replicate. We found that the best quantifier for measuring reproducibility is the plain number of RNA-seq reads supporting each element, rather than normalized values such as FPKM, owing to the discrete nature of RNA-seq signal which, at low levels, is dominated by the sampling noise. The elements in each bio-replicate are binned according to their signal, and for all bins the npIDR_{1in2} is calculated as the proportion of elements in each bin in replicate 1 that have exactly zero signal (i.e. not detected) in replicate 2. Similarly, the npIDR_{2in1} is calculated as the proportion of elements in each bin in replicate 2 that have exactly zero signal (i.e. not detected) in replicate 1. If quantification differences between bio-replicates are caused entirely by random noise, the npIDR_{1in2} and npIDR_{2in1} values should be close to each other. In practice, the difference in sequencing depths (i.e. numbers of mapped reads) of the two bio-replicates causes a systematic bias, and to correct for it we calculate the final npIDR value for each signal bin as the average of npIDR_{1in2} and npIDR_{2in1}. A typical example of npIDR dependence on the signal for *de novo* splice junctions as elements is shown in Figure S-8. Assuming that reproducibility within a sample of junctions with the same signal is equivalent to the reproducibility of individual junctions in an ensemble of experiments, the npIDR determines the probability of an element not to be detected in another experiment of the same depth as bio-replicate 1 or 2. We can also infer the npIDR for an experiment of a combined depth of replicates 1 and 2, by re-quantifying each element with the "pooled" signal value from the two bio-replicates. If signal is the number of RNA-seq reads per element, then the pooled value is calculated as a sum of the signals in two bio-replicates, for a normalized signal such as FPKM this could be an average value, or a maximum value. The npIDR is then assigned to each element according to its pooled signal value, and the npIDR vs. signal dependence calculated in the first step.

Figure S-7

Cumulative number of annotated and novel GT/AG junctions supported by at least a given (X-axis) number of reads per junction in H1ES RNA-seq data. Only staggered reads, i.e. reads with distinct 5' or 3' loci, are counted.

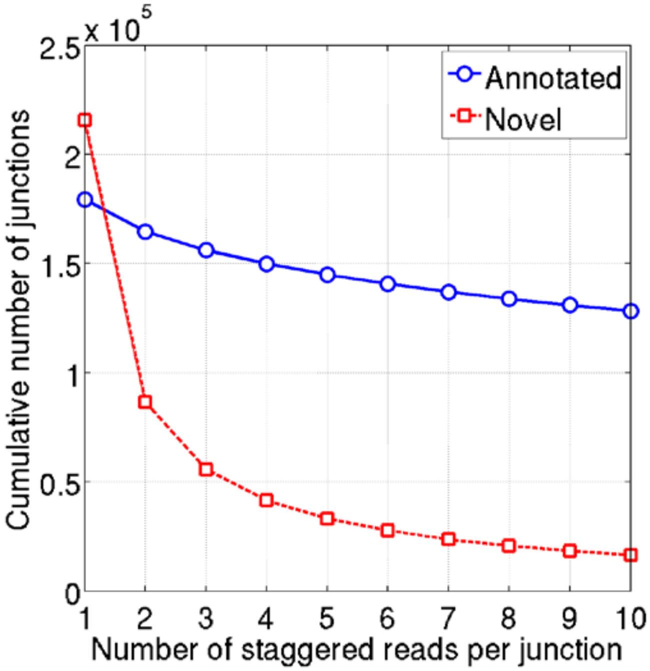
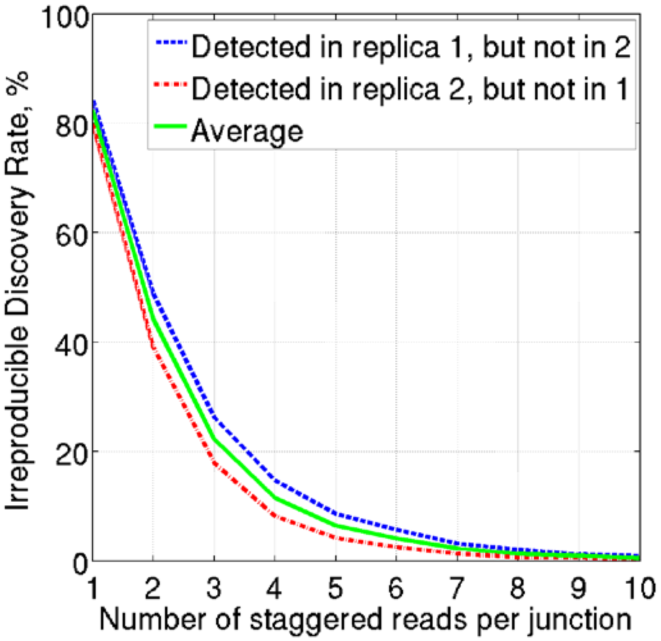


Figure S-8

Percentage of reads present in replica 1 with a given (X-axis) read count and not detected in replica 2, and vice versa. The Average curve represents the npIDR as a function of read count.



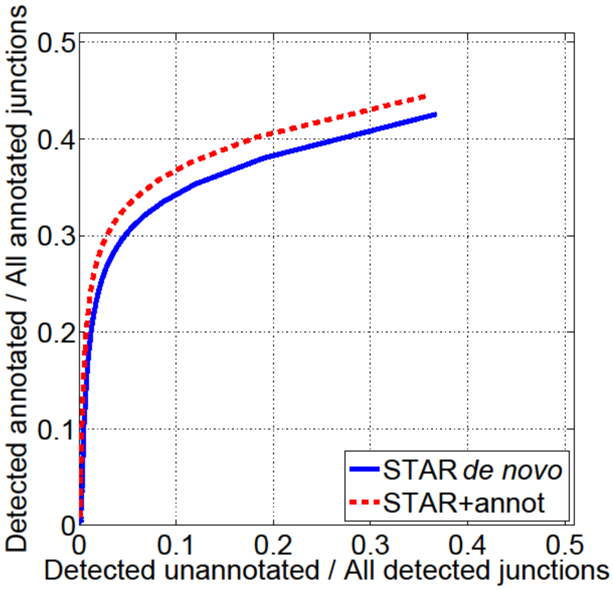
4. Running STAR with annotated junctions database

STAR can utilize annotated splice junctions loci to improve sensitivity of the splice junction detection. STAR incorporates annotated junction sequences into the suffix array and searches the seeds that cross the junctions simultaneously with the seeds that map contiguously to the genome. Stitching and scoring is also done simultaneously for spliced and contiguous seeds, thus allowing detection of annotated and novel junctions in one mapping pass. This procedure makes STAR more sensitive to splicing events that involve short sequence overhangs on either side of a junction.

If we supply Gencode 7 (Harrow, et al., 2012) annotations to STAR, it finds ~5 million more annotated splicing events (i.e. reads crossing annotated junctions), increasing the number of spliced read by ~50%. Importantly, ~6 thousand more annotated junctions are detected (see Figure S-9).
Another option for supplying the splice junctions' loci is to run 2nd pass of STAR alignments utilizing the junctions found in the 1st *de novo* step. In this case new junctions will not be discovered, however more spliced reads crossing the previously detected junctions will be found.

Figure S-9

Pseudo-ROC curve for STAR+annotation and STAR 'de novo' runs: % of all annotated junctions that are detected vs. % of detected junctions that are unannotated.



5. Mapping long mRNA sequences

Human mRNA sequences were downloaded from the UCSC Genome Browser:

http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/mrna.fa.gz

(the file was "Last modified" on 06-Oct-2012).

First 100,000 mRNA sequences between 0.5kb and 5kb long were mapped to the human genome with BLAT and STAR. The mean length of the transcripts was ~2kb. BLAT 3.4 was run with the default parameters since they are optimized for alignment of long EST/mRNA sequences:

```
blat hg19.fa mrna.2012-10-06.500to5000_UpperCase.100k.fa blat.psl.out
```

The "over-occuring tile" file 11.ooc was generated before the alignment run with:

```
blat -makeOoc=11.ooc hg19.fa xxx yyy
```

STAR was compiled with *"make STARlong"* command allowing allocation of large arrays and was run with the following parameters:

```
STAR --runThreadN 1   --outFilterMismatchNmax 100
--seedSearchLmax 30   --seedSearchStartLmax 30
--seedPerReadNmax 100000   --seedPerWindowNmax 100
--alignTranscriptsPerReadNmax 100000
--alignTranscriptsPerWindowNmax 10000
--genomeDir hg19
--readFilesIn mrna.2012-10-06.500to5000_UpperCase.100k.fa
```

For each read, the one best alignment was selected for BLAT and for STAR. Reads were considered mapped if ≥80% of their lengths were aligned to the genome.

The comparison of STAR and BLAT alignments is presented in Table S-1 and Figure S-10. Of the 100,000 reads, STAR aligns 96,557 reads (96.6%), slightly lower than BLAT's 97,441 (97.5%). STAR produces longer alignments more often than BLAT: STAR's alignments are longer than BLAT's for 9,276 reads, while STAR's alignments are shorter than BLAT's for 5,734 reads.

Next we compared splice junction (or intron) chains for STAR and BLAT alignments. STAR yields alignments with at least one junction for 80,459 reads compared with BLAT's 81,884 reads. STAR yields slightly larger number of annotated junction chains: 62,359 of STAR's and 61,881 of BLAT's spliced reads have intron chains with all the junctions annotated in Gencode 13 (Harrow, et al., 2012).

Figure S-10 shows the numbers of reads with fully annotated junction chains as a function of number of junctions per read. STAR finds more alignments with longer annotated junction chains than BLAT:

overall, STAR detected 502,830 splices in reads with fully annotated junction chains compared to 495,470 splices for BLAT.

BLAT's and STAR's junction chains are identical for 63,142 of all spliced reads and 57,038 of reads with fully annotated chains, which demonstrates a good overall agreement between STAR and BLAT alignments.

The mapping time was benchmarked on the same server as described in the section 3.3 of the main text. Only one thread was used for both STAR and BLAT since BLAT is not multi-threaded. STAR demonstrated a 160-fold mapping speed advantage over BLAT: to map 100,000 reads BLAT spent 12 hours, while STAR spent only 4.5 min.

STAR's and BLAT's output files, as well scripts used to process the data, can be downloaded from:

ftp://ftp2.cshl.edu/gingeraslab/tracks/STARpaper/STARpaper_mRNA.tgz

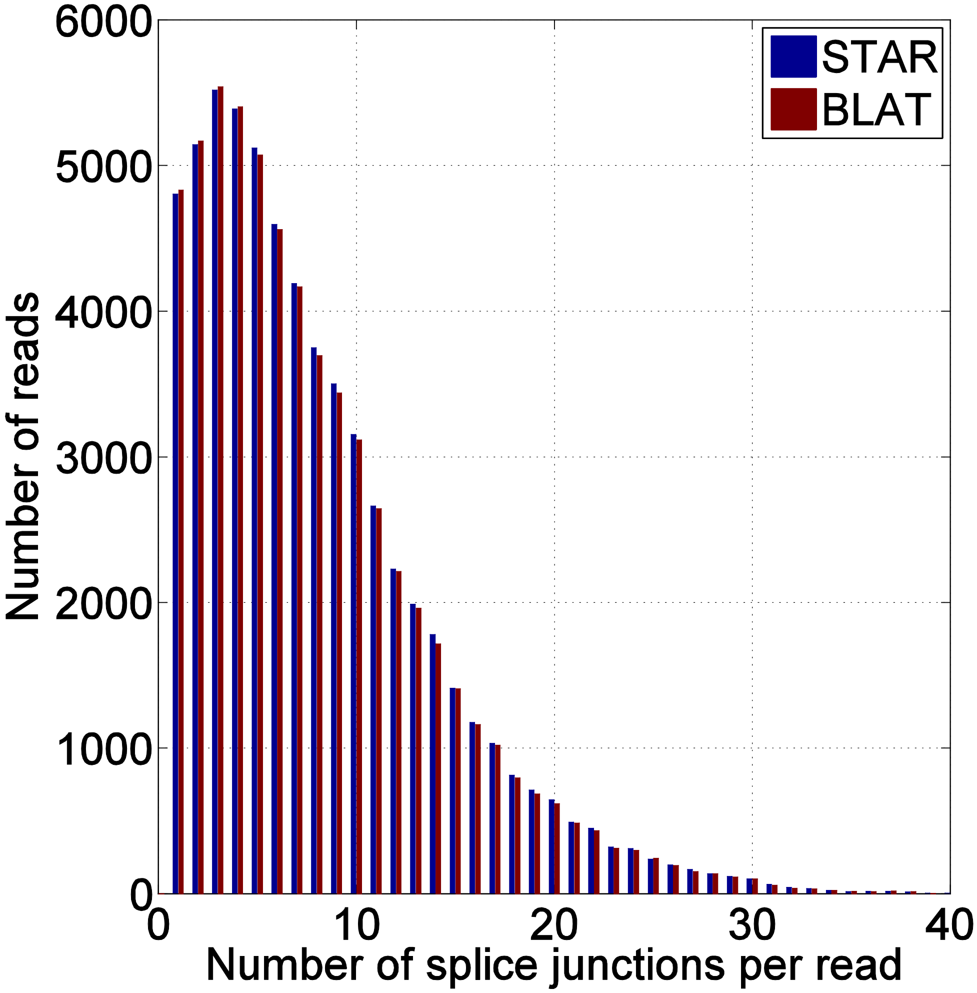
Table S-1

mRNA mapping statistics for STAR and BLAT.

	STAR	BLAT
All reads	100,000	
Mapped reads (>=80% of read length aligned)	96,557	97,441
Alignments that are longer than the other aligner's	9,276	5,734
Reads with one or more splice junctions	80,459	81,884
Reads with fully annotated junction chains	62,359	61,881
Number of junctions in fully annotated introns chains	502,830	495,470
Reads with identical junction chains	63,142	
Reads with identical annotated junction chains	57,038	

Figure S-10

Number of reads with fully annotated junction chains as a function of number of junctions per read for STAR's and BLAT's alignments of mRNA sequences.



6. DATA ACCESS

GEO: GSE38886 (Roche 454 sequencing)
GEO: GSE30567 (Illumina long RNA-Seq)

The Illumina long RNA-seq data utilized in this paper can also be downloaded from the UCSC ENCODE hub:

K562:

http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqK562CellPapFastqRd1Rep1.fastq.gz

http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqK562CellPapFastqRd2Rep1.fastq.gz

H1ES:

http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqH1hescCellPapFastqRd1Rep1.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqH1hescCellPapFastqRd2Rep1.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqH1hescCellPapFastqRd1Rep2.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqH1hescCellPapFastqRd2Rep2.fastq.gz

HUVEC:
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHuvecCellPapFastqRd1Rep1.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHuvecCellPapFastqRd2Rep1.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHuvecCellPapFastqRd1Rep2.fastq.gz
http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHuvecCellPapFastqRd2Rep2.fastq.gz

7. Supplementary References

Grant, G.R., *et al.* (2011) Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM), *Bioinformatics*, **27**, 2518-2528.

Harrow, J., *et al.* (2012) GENCODE: The reference human genome annotation for The ENCODE Project, *Genome Research*, **22**, 1760-1774.