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.

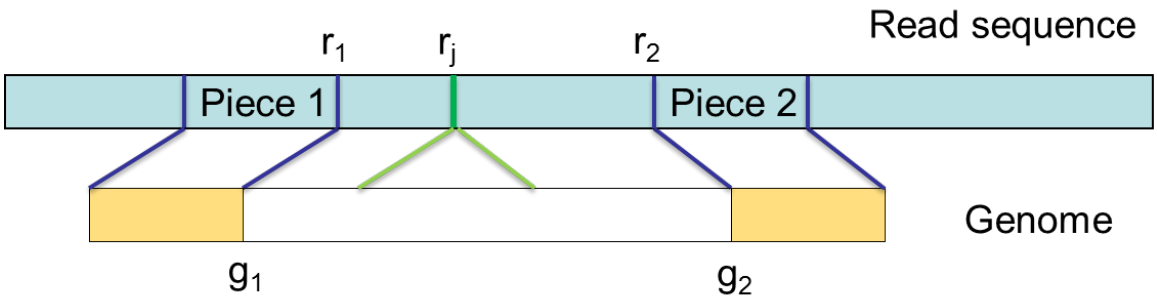
The gap opening penalties are user-defined and can be set independently for GT/AG, GC/AG, AT/AC and all other (non-canonical) motifs. The penalties for different intron motifs have to be selected according to the frequency expectations of different intron motifs in the species under study. The default 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

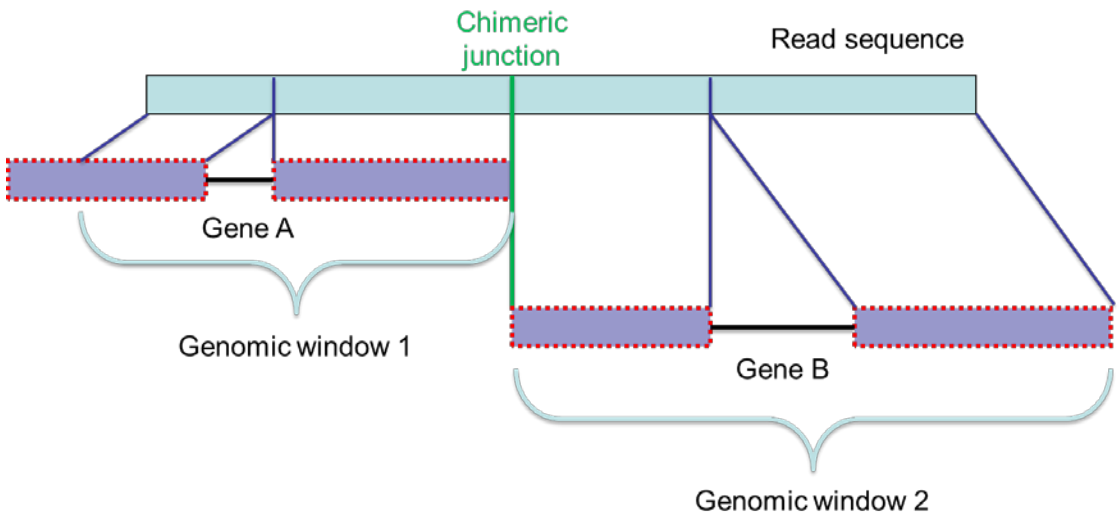
Alignments from all windows are collected and sorted by their score. All the alignments scored within a user-defined range of the maximum score are considered multi-mappers. Some additional user-configurable filtering can be done before the alignments are output.

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.

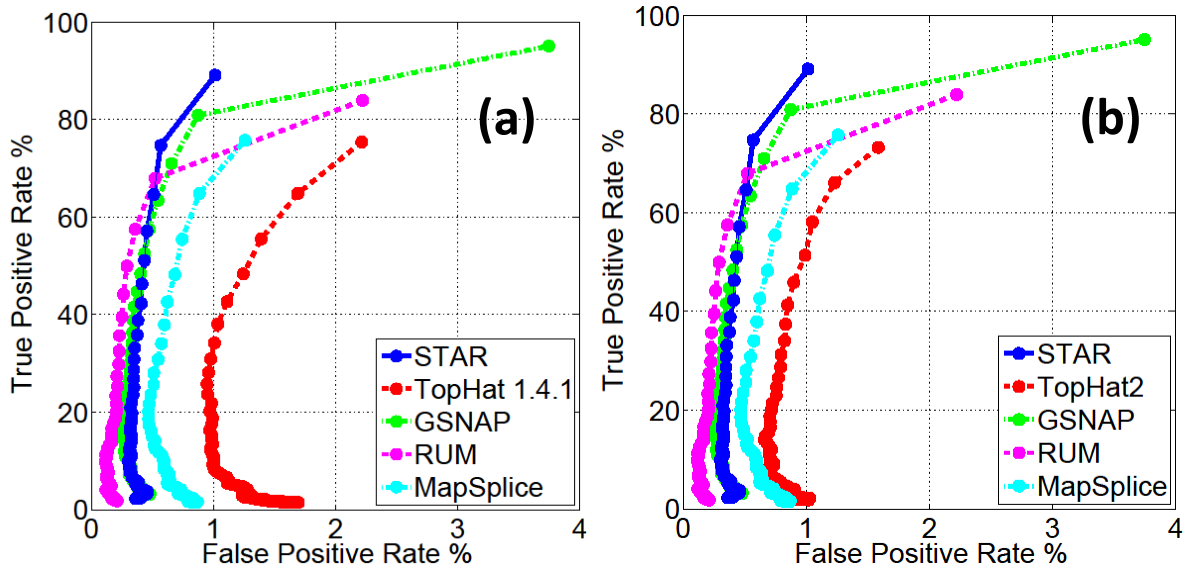
For speed benchmarking, the test were run using the Linux "virtual disk" device /dev/shm for all the input, output and temporary files to avoid hard drive bandwidth and latency issues.

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).

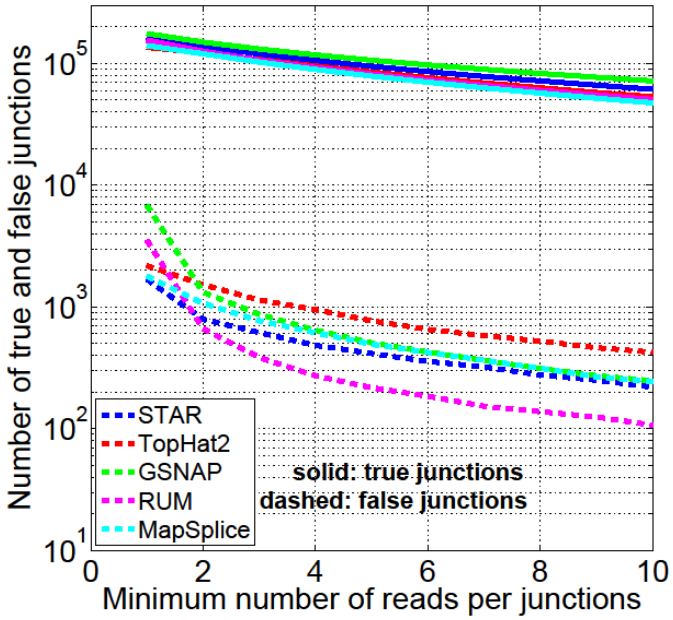


TopHat2's accuracy is improved significantly compared to TopHat 1.4.1. Moreover, the mapping speed of TopHat 2.0.0 has increased by ≈30%.

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.



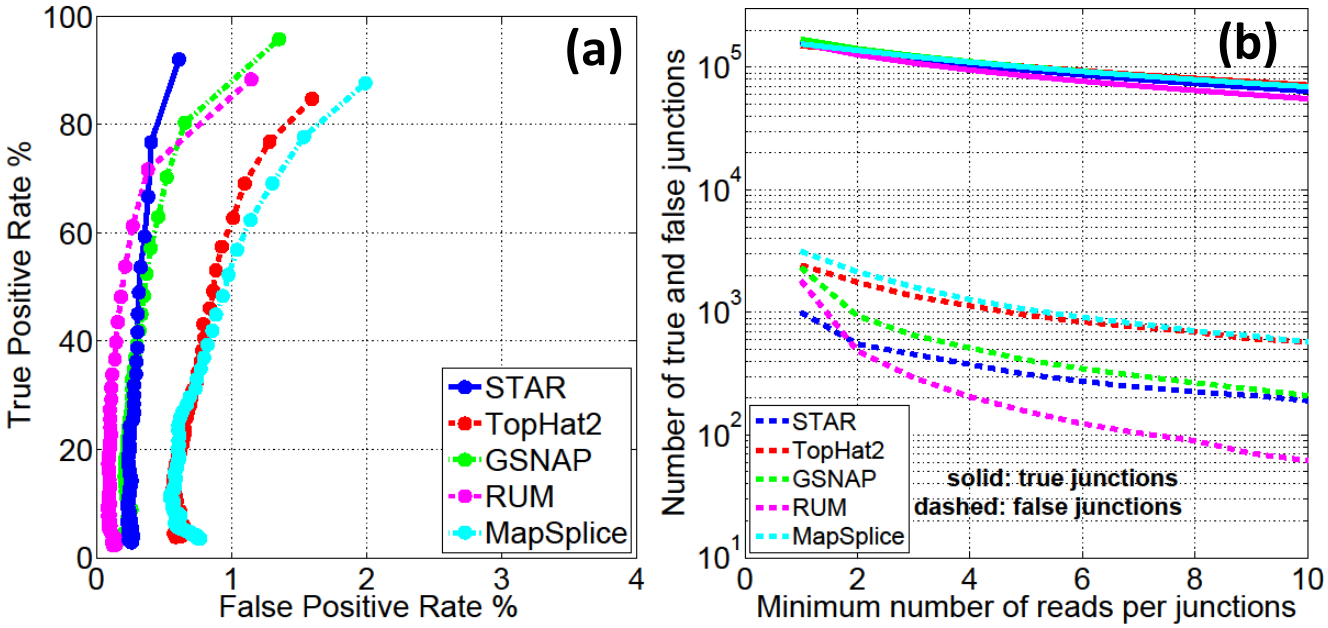
2.3. Simulated dataset SIM1_TEST1

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.).



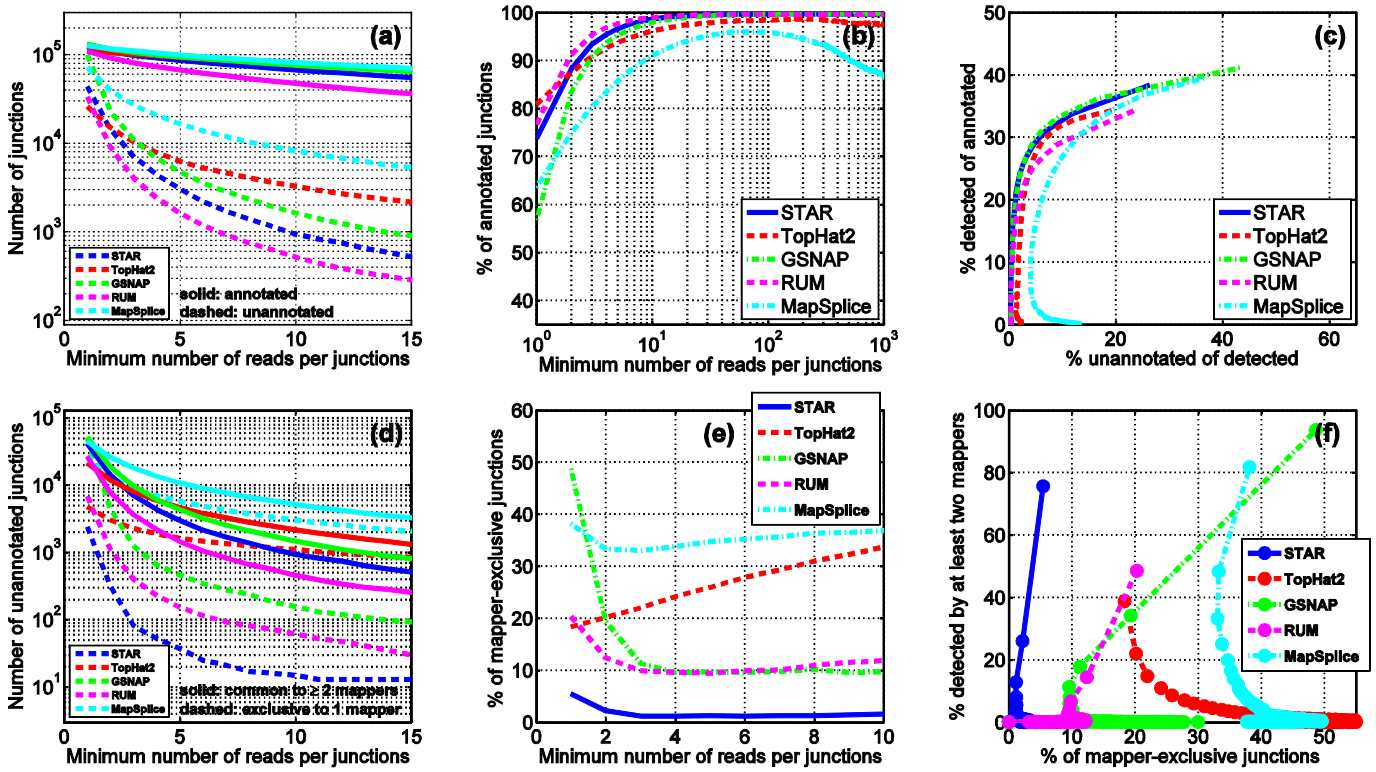
2.4. Experimental 2x50b RNA-seq data

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.

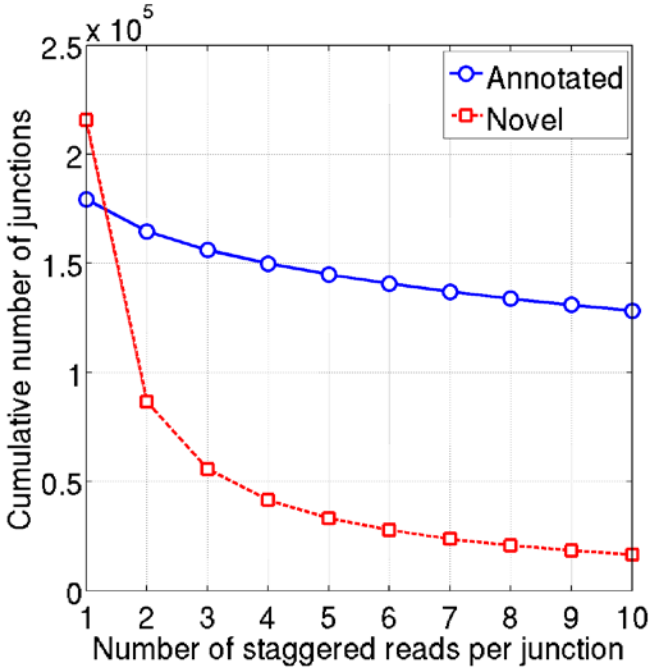
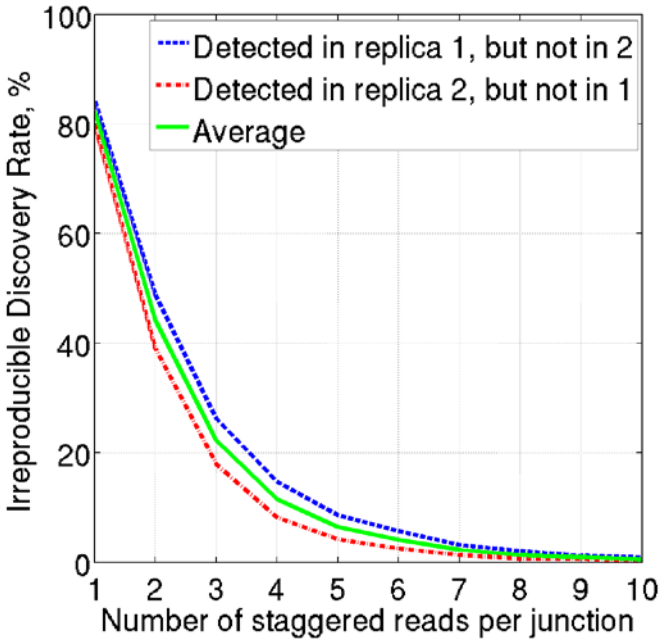


Figure S-8

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.



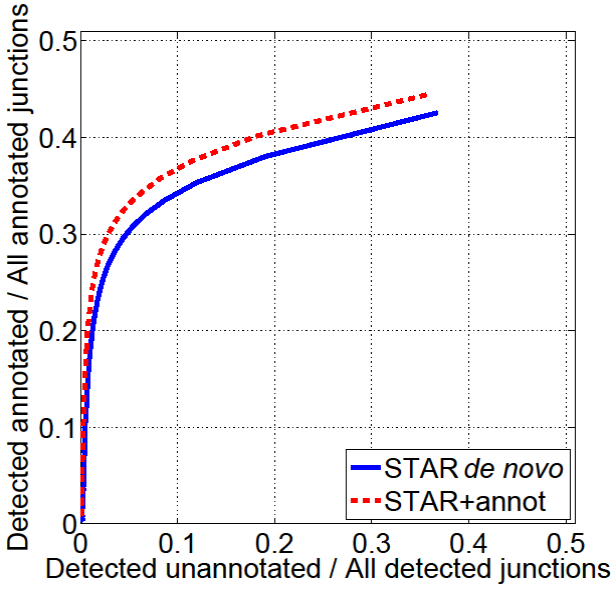
4. Running STAR with annotated junctions database

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

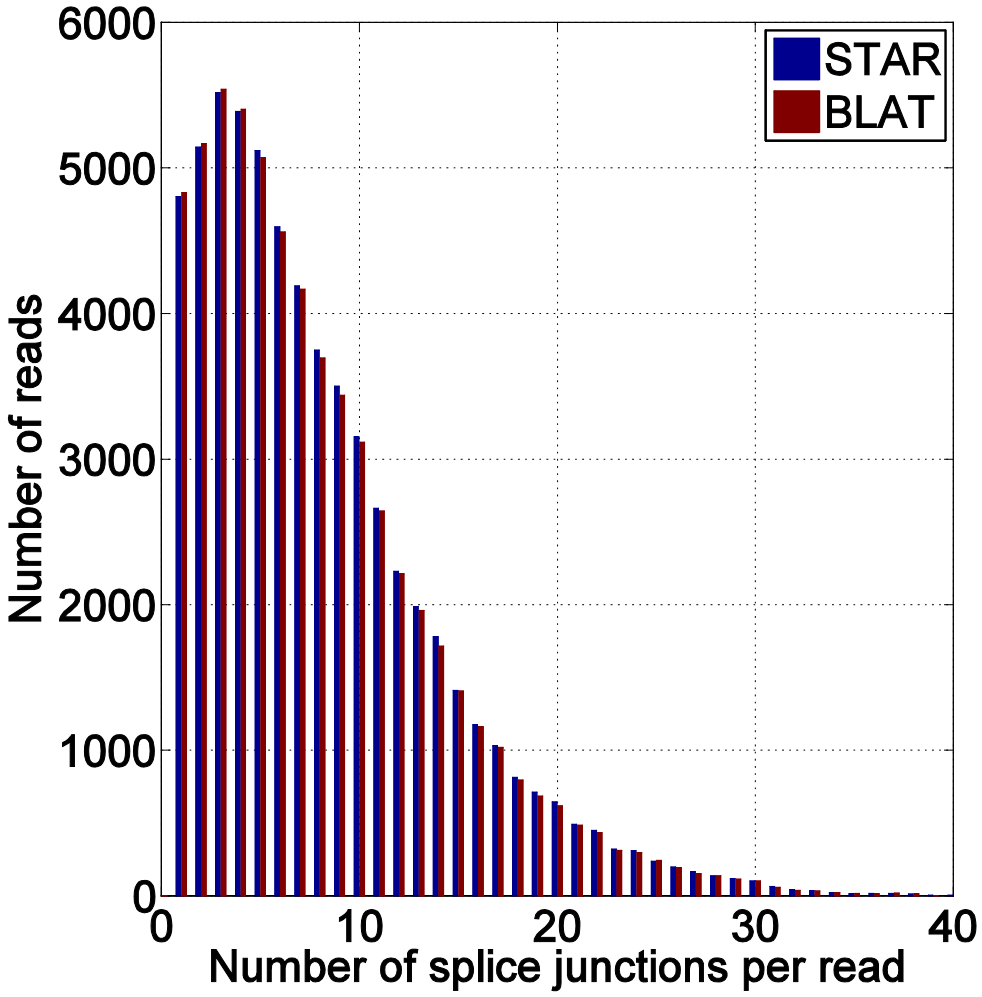
Table S-1

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.