Supplementary Materials — The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote

Yang Liao[1,2], Gordon K Smyth[1,3] and Wei Shi[1,2,#]

[1]Bioinformatics Division, The Walter and Eliza Hall Institute of Medical Research, 1G Royal Parade, Parkville, VIC 3052,
[2]Department of Computing and Information Systems
[3]Department of Mathematics and Statistics, The University of Melbourne, Parkville, VIC 3010, Australia

[#] To whom correspondence should be addressed.
Wei Shi
Bioinformatics Division
Walter and Eliza Hall Institute
1G Royal Parade, Parkville, VIC 3052, Australia
Email: shi@wehi.edu.au
Phone: +61 3 9345 2629
Fax: +61 3 9347 0852

Supplementary Methods

Determine number of subreads selected for voting and consensus threshold

We created a series of calibration datasets to find the best parameter setting for Subread aligner. These datasets had different sequencing error rates, ranging from 0 to 10 percent. Each dataset included 10 million 101bp reads which were randomly extracted from a modified human reference genome (GRCh37). In the modified genome, duplicated 80bp long sequences were removed so as to make extracted reads have unique locations. Combinations of different numbers of subreads selected for voting and different consensus thresholds were examined to determine the best values for these two parameters used by Subread aligner.

Removing non-informative subreads

Figure S2 shows the cumulative frequency distribution of all possible 16bp sequences extracted from the latest version of human genome (GRCh37). 81% of these sequences were found to occur 24 or fewer times in the genome. We investigated the effect of removing non-informative subreads using different occurrence cut-offs on mapping accuracy and sensitivity. Figure S4 shows the results from using cut-off values of 24 and 128. Different numbers of selected subreads ranging from 7 to 28 were examined in this evaluation. There was a slight increase in mapping sensitivity when the occurrence cut-off increased from 24 to 128, but here was also a little decrease observed in mapping accuracy. Therefore, the overall mapping performance was larged unchanged. Using other occurrence cut-offs yielded similar results. Furthermore, running time and memory usage were found to increase by 4.4% and 1%, respectively, when the cut-off increased from 24 to 128. We decided to use 24 as the occurrence cut-off for removing non-informative reads.

Memory management of Subread and Subjunc

Subread (or Subjunc) extracts 16bp sequences from the reference sequence and then builds a hash table with keys being 16bp sequences and values being their chromosomal locations. Keys are strings of 0's and 1's, which are the encoded values for each 16bp sequence. Each base in the 16bp sequence is encoded by a 2-bit binary number (A:00, T:01, G:10, C:11), permitting each 16bp sequence to take exactly four-byte storage. Each 16bp sequence can therefore be loaded in a register in Central Processing Unit (CPU) via one memory request. Figure S1 shows the memory management scheme used by Subread (or Subjunc).

Mapping paired-end reads

To map paired-end reads, Subread (or Subjunc) firstly maps two reads from the same pair individually as if they were single-end ends. Mapping location of a read, which was mapped with higher confidence than the other read from the same pair (e.g. larger number of votes obtained), was determined as the "anchor" location for the fragment. Paired-end distance information is then used to help map the other read accurately. Due to the availability of distance information, we use a relaxed consensus threshold for calling mapping locations for the unmapped pair, which is as low as only one vote. This will ensure a good sensitivity is achieved.

When there are more than one mapping locations found for this pair, we choose the one which satisfies the distance criteria. If there are more than one such location, we

choose the one which has larger number of votes. If there is still a tie, we can break it by using mapping quality score and Hamming distance.

If none of the mapping locations for the two ends satisfies the distance criteria, Subread and Subjunc still report the mapping locations for these two reads, as long as they meet the requirement for the minimal number of consensus subreads (3 by default). These read pairs may have a fragment length greater or less than the specified fragment length, or they may originate from chimeric sequences.

Implementation and usage

Subread and Subjunc were implemented using programming language C. To use them to map reads or discover exon-exon junctions, an index has to be built first. At this step, the size of requested memory may be tuned by users. It takes about 1 hour to build an index for human or mouse genome on a Linux computer with a few gigabytes of memory. The index needs only to be built once and can then be re-used in the subsequent read mapping operations. Subread and Subjunc output mapping results in a SAM format text file. They can be freely downloaded from `http://subread.sourceforge.net/`. An R version of the program has also been developed, which is called Rsubread and can be downloaded from the Bioconductor R project `http://www.bioconductor.org/packages/release/bioc/html/Rsubread.html`. The R package calls the underlying C functions for read mapping.

Mapping for reads generated from major sequencing platforms, including Illumina Genome Analyzer/HiSeq, ABI SOLiD and Roche 454, is all supported.

Supplementary Figures and Tables

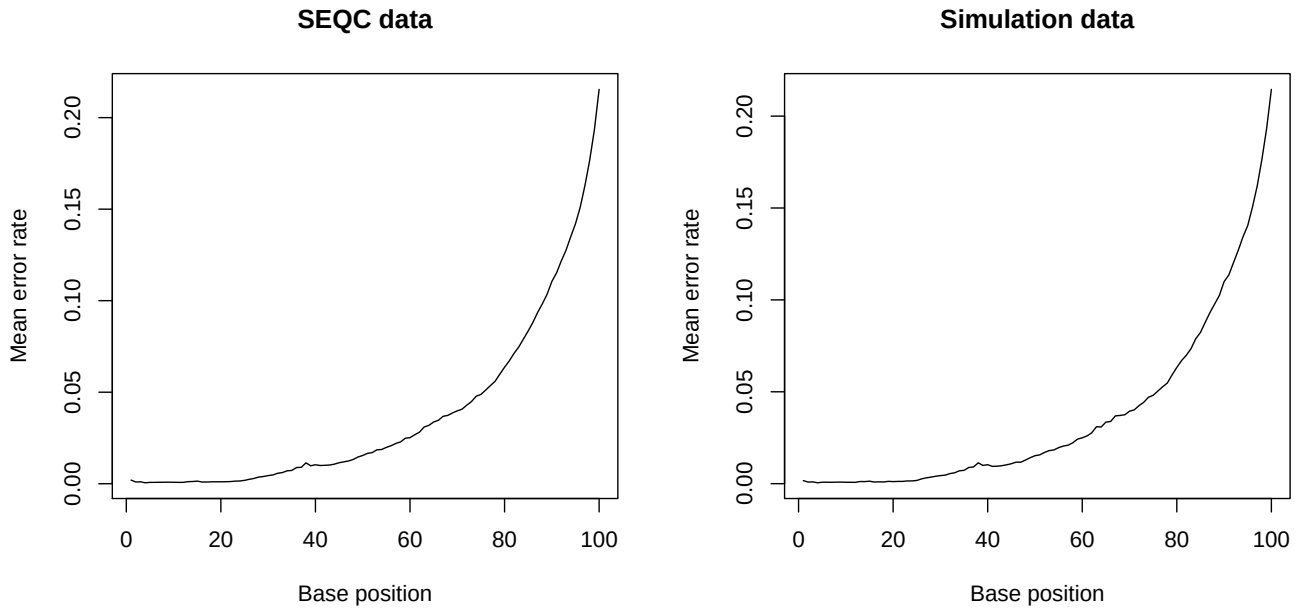

Figure S1: Error rates at different base positions in SEQC data and simulation data. Mean error rates calculated for SEQC data were based on the base calling p values of read bases. Mean error rates calculated for simulation data were based on the numbers of errors introduced to read bases.

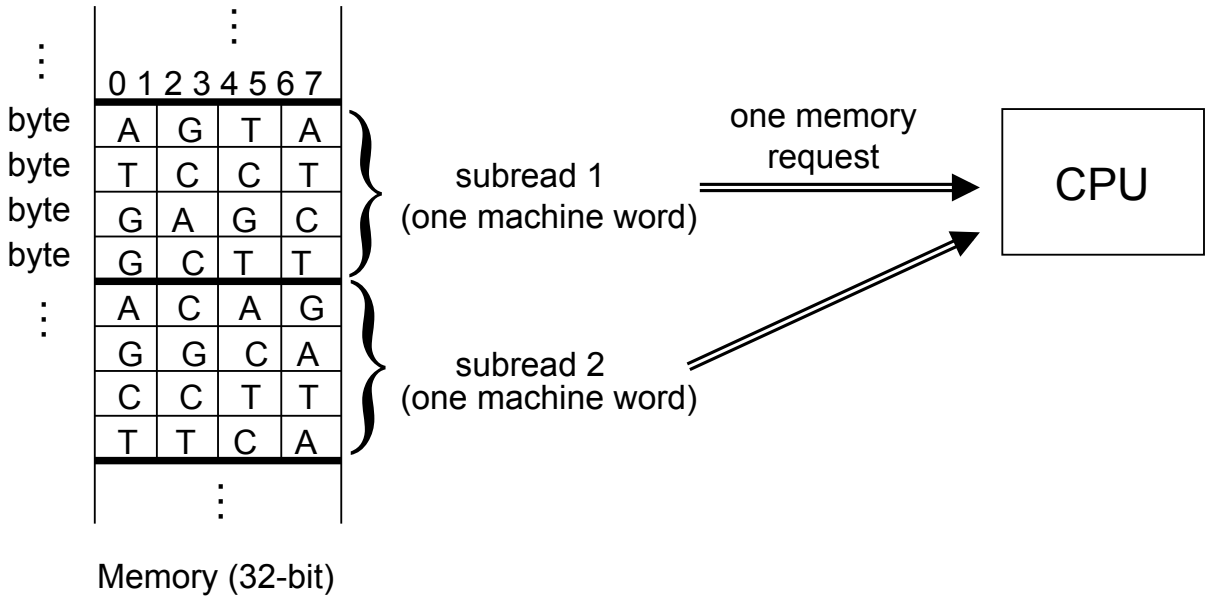

Figure S2: Memory management scheme used by Subread an Subjunc. Each 16bp sequence is saved into 4 bytes which is one machine word on a 32-bit computer system(half a machine word on a 64-bit computer system). Each base in the sequence is encoded by 2 bits. Each subread can be loaded into a CPU register by one memory request.

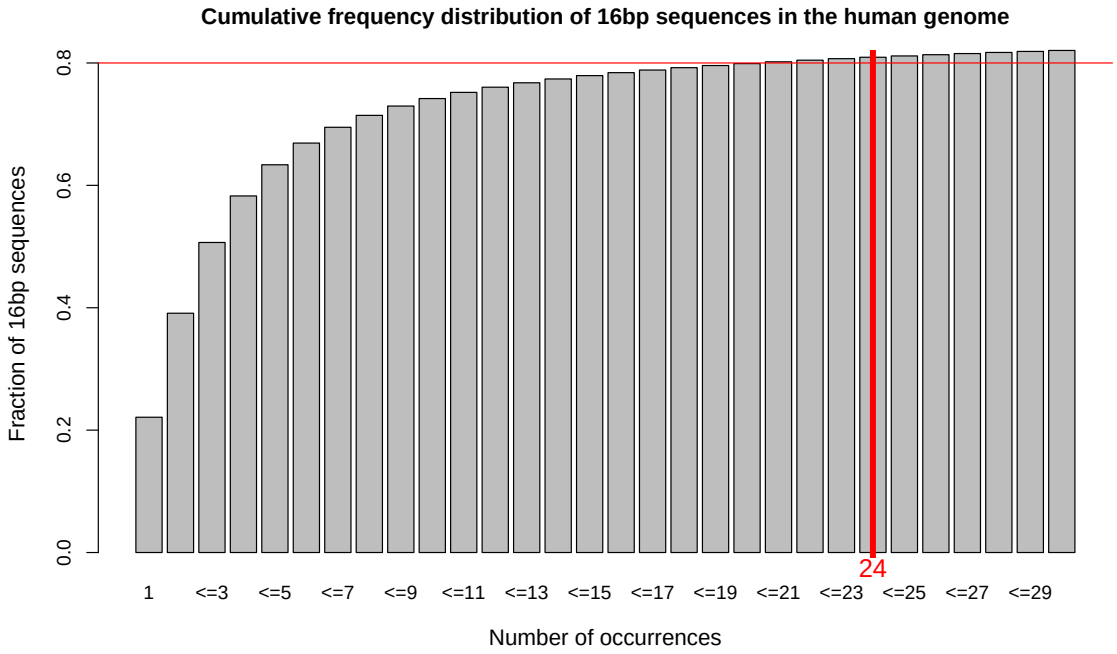

Figure S3: Cumulative distribution of occurrences of 16bp sequences in the human genome (GRCh37). 16bp sequences which occur no more than 30 times in human genome are shown in this figure. Each group in the figure gives the fraction of 16bp sequences in human genome (including repetitive sequences), which occur no more than a particular number of times (shown in x-axis). 81 percent of 16bp sequences occur 24 or fewer times.

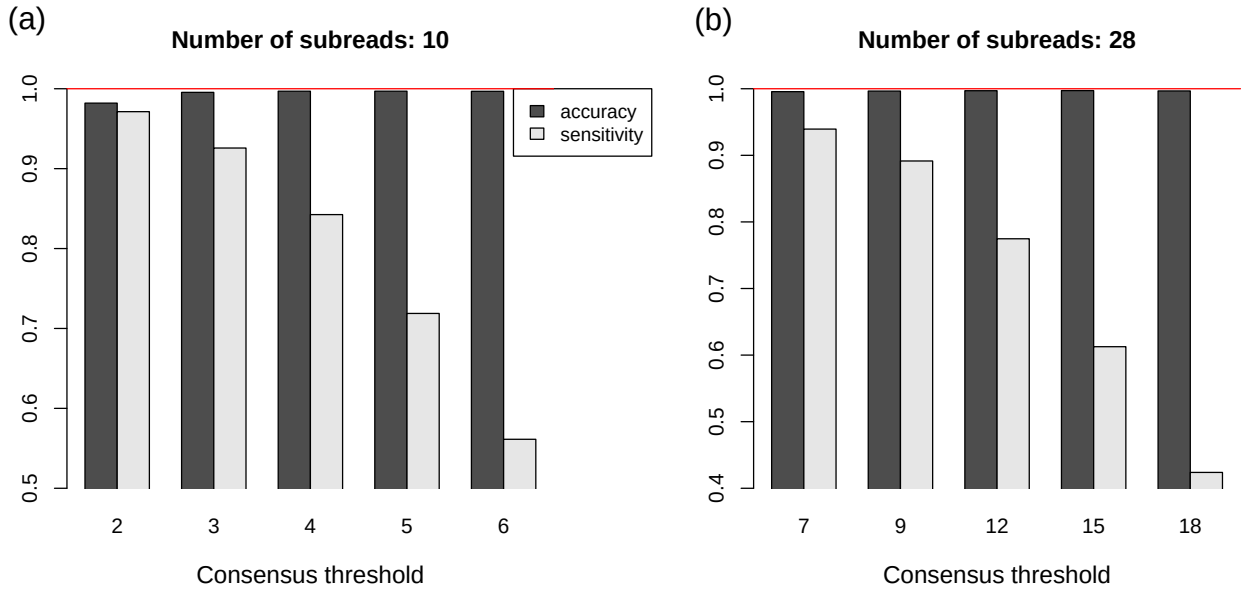

Figure S4: Mapping accuracy and sensitivity from using different numbers of subreads and different consensus thresholds. (a) Accuracy and sensitivity obtained from using 10 subreads and consensus thresholds of 2, 3, 4, 5 and 6. (b) Accuracy and sensitivity obtained from using 28 subreads and consensus thresholds of 7, 9, 12, 15 and 18. Groups in the figures correspond to different consensus thresholds. An occurrence threshold of 24 was used for removing non-informative subreads.

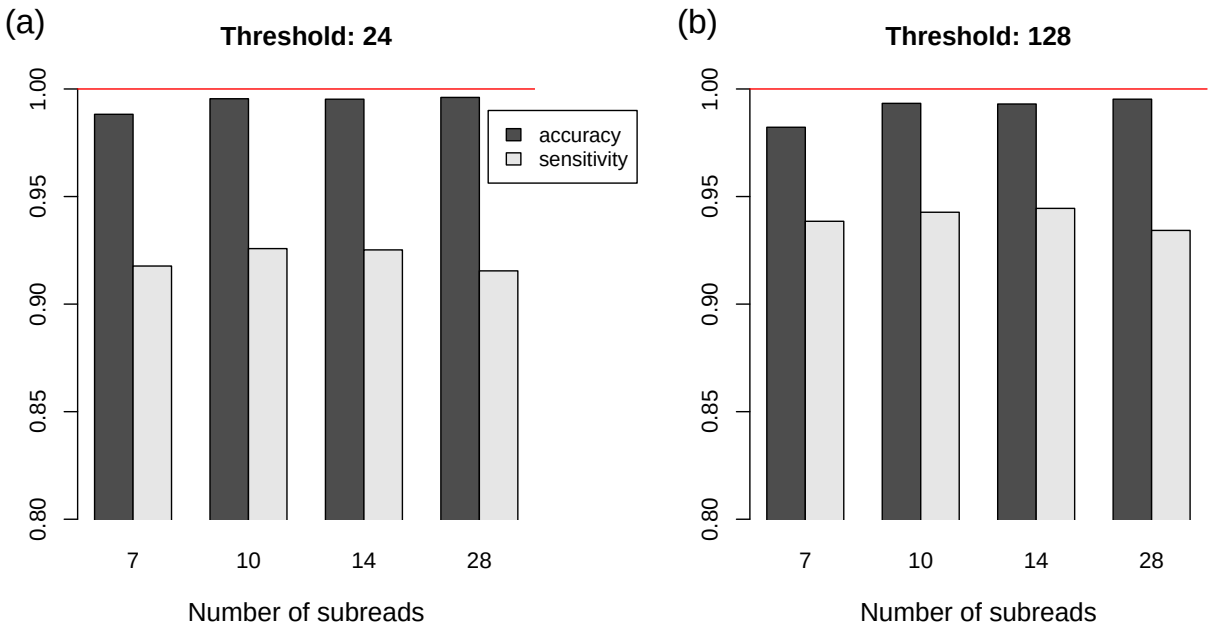

Figure S5: Mapping accuracy and sensitivity when consensus threshold is set to be 30% of subread number. (a) An occurrence threshold of 24 was used. 16bp sequences occurring more than 24 times in human genome were removed. (b) An occurrence threshold of 128 was used. 16bp sequences occurring more than 128 times in human genome were removed. In both (a) and (b), multiple subread numbers are used which are represented by different groups. Consensus thresholds of 2, 3, 4 and 8 were used respectively for subread numbers including 7, 10, 14 and 28. These thresholds are about 30% of corresponding subread numbers.

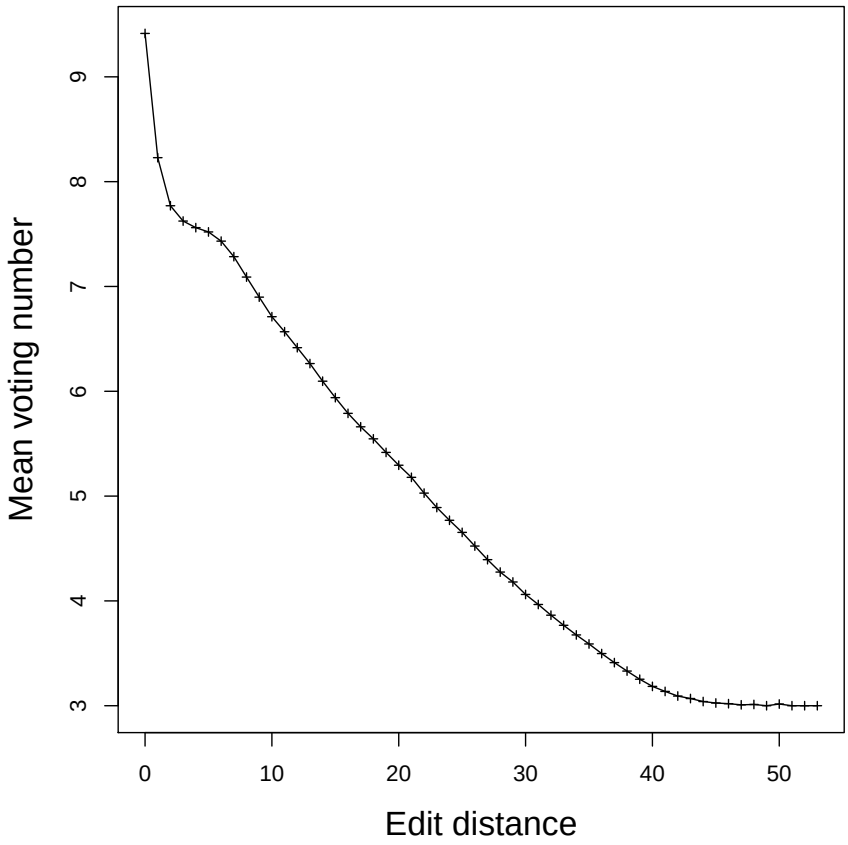

Figure S6: Number of successful votes and edit distances for reads successfully mapped by Subread. Data used here included 10 million 101bp reads generated from filtered human genome using our simulator.

Table S1: Performance of aligners in mapping 10 million simulated 202bp reads including indels. Maq, Novoalign and MrsFast reported errors and were not able to complete the alignments successfully, for both non-indel dataset and the dataset including indels.

Aligner	%Recall	%Accuracy	Time (Min)	Memory(Gb)
Subread	96.2	98.7	18 (31)	7.6 (4.3)
Bowtie2	98.4	98.5	125	3.3
BWA	75.8	97.5	382	2.8
Maq	-	-	-	-
Novoalign	-	-	-	-
MrsFast	-	-	-	-