Nature Methods

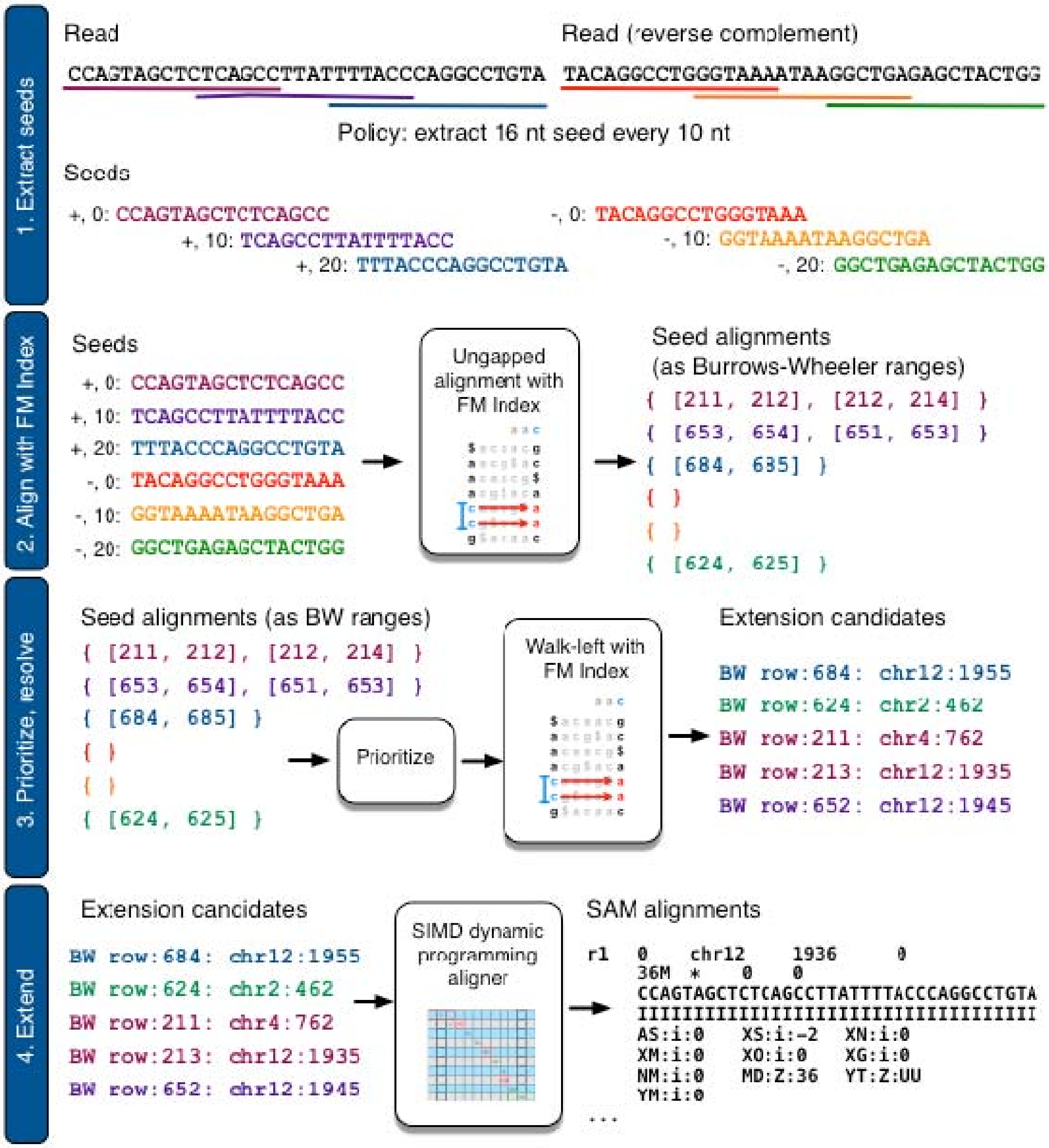
Fast gapped-read alignment with Bowtie 2

Ben Langmead & Steven L Salzberg

Supplementary Figure 1	Alignment workflow.
Supplementary Figure 2	Sensitive and accurate alignment of simulated reads.
Supplementary Figure 3	Comparison of Bowtie 2 and Bowtie 1 aligning unpaired HiSeq 2000 reads.
Supplementary Figure 4	Comparison of Bowtie 2 and Bowtie 1 aligning paired HiSeq 2000 reads.
Supplementary Table 1	Command-line arguments and full results from experiments using real data.
Supplementary Table 2	Alignment overlap between Bowtie 2 and BWA/BWA-SW.
Supplementary Table 4	Comparison of Bowtie 2 and Bowtie 1 aligning unpaired HiSeq 2000 reads.
Supplementary Table 5	Comparison of Bowtie 2 and Bowtie 1 aligning paired HiSeq 2000 reads.
Supplementary Table 6	Comparison including aligners not based on the FM Index.
Supplementary Note	Bowtie 2 design.
Supplementary Results	Additional software comparisons.

Note: Supplementary Table 3 and Supplementary Software are available on the Nature Methods website.

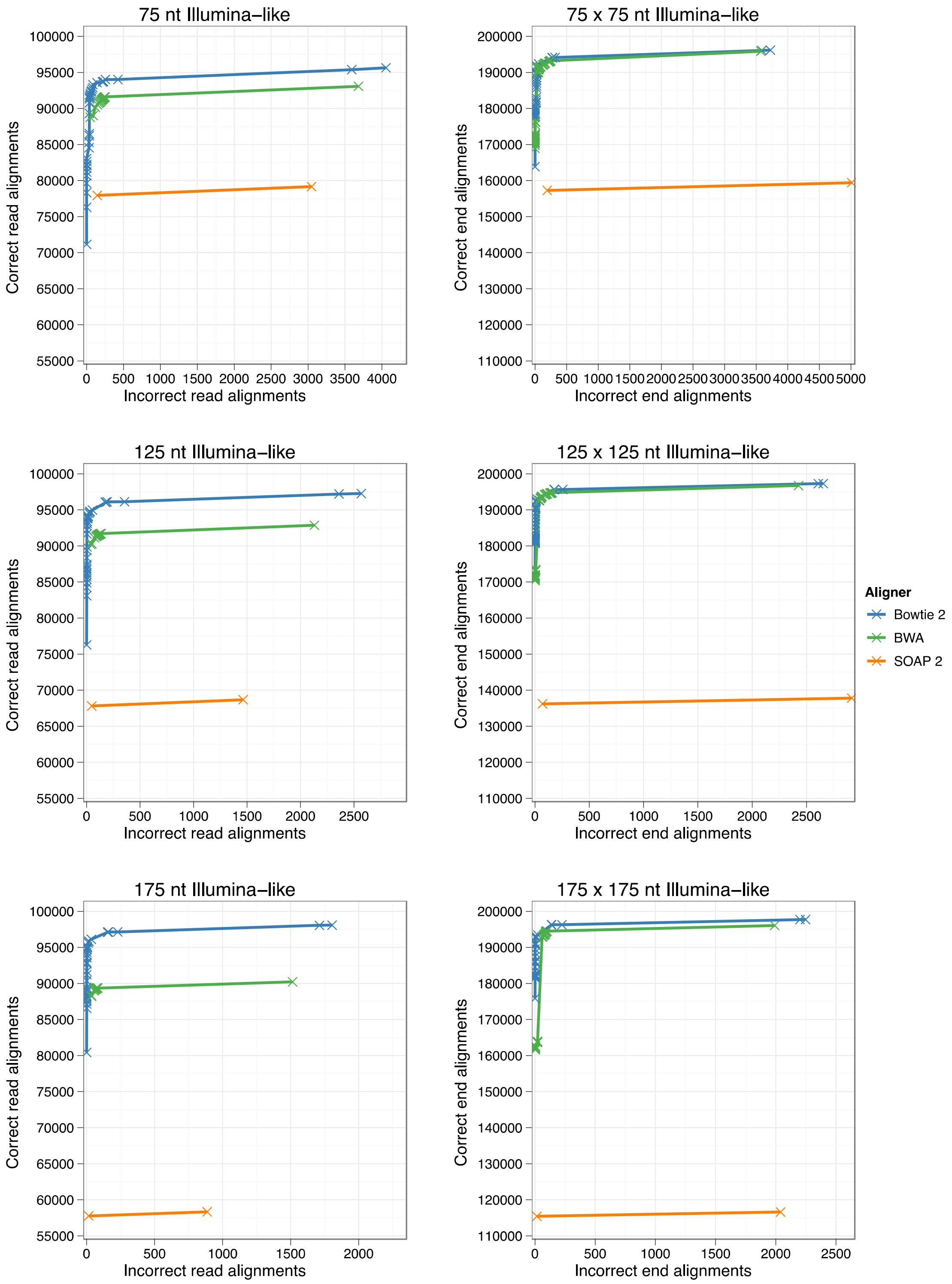
Supplementary Figure 1



```
r1  0   chr12    1936     0
    36M  *    0    0
    CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCTGTA
    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
    AS:i:0    XS:i:-2   XN:i:0
    XM:i:0    XO:i:0    XG:i:0
    NM:i:0    MD:Z:36   YT:Z:UU
    YM:i:0
...
```

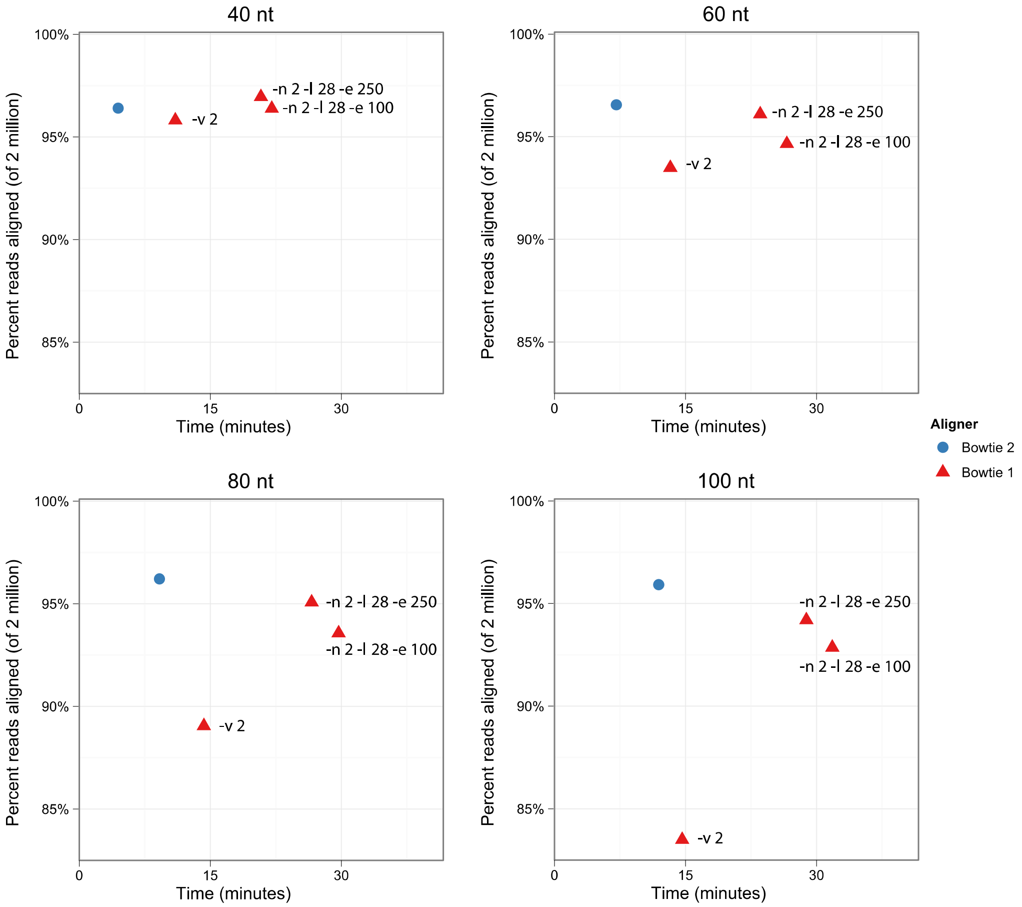
Bowtie 2 alignment workflow. For each read, Bowtie 2 proceeds in four steps. In Step 1, Bowtie 2 extracts "seed" substrings from the read and its reverse complement. In Step 2, seed substrings are aligned to the genome in an ungapped fashion using the FM Index, yielding Burrows-Wheeler (BW) ranges. Step 3 takes the BW ranges and prioritizes rows such that rows from smaller ranges receive a higher priority. Bowtie 2 then repeatedly chooses rows randomly, weighted by priority, and resolves each selected row's offset into the reference genome using the FM Index "walk-left" procedure. Step 4 takes prioritized, resolved alignments from step 3 and performs Single Instruction Multiple Data (SIMD)-accelerated dynamic programming alignment in the vicinity of each until all seed hits are examined, until a sufficient number of alignments are examined, or until the dynamic programming effort limit is reached.

Supplementary Figure 2



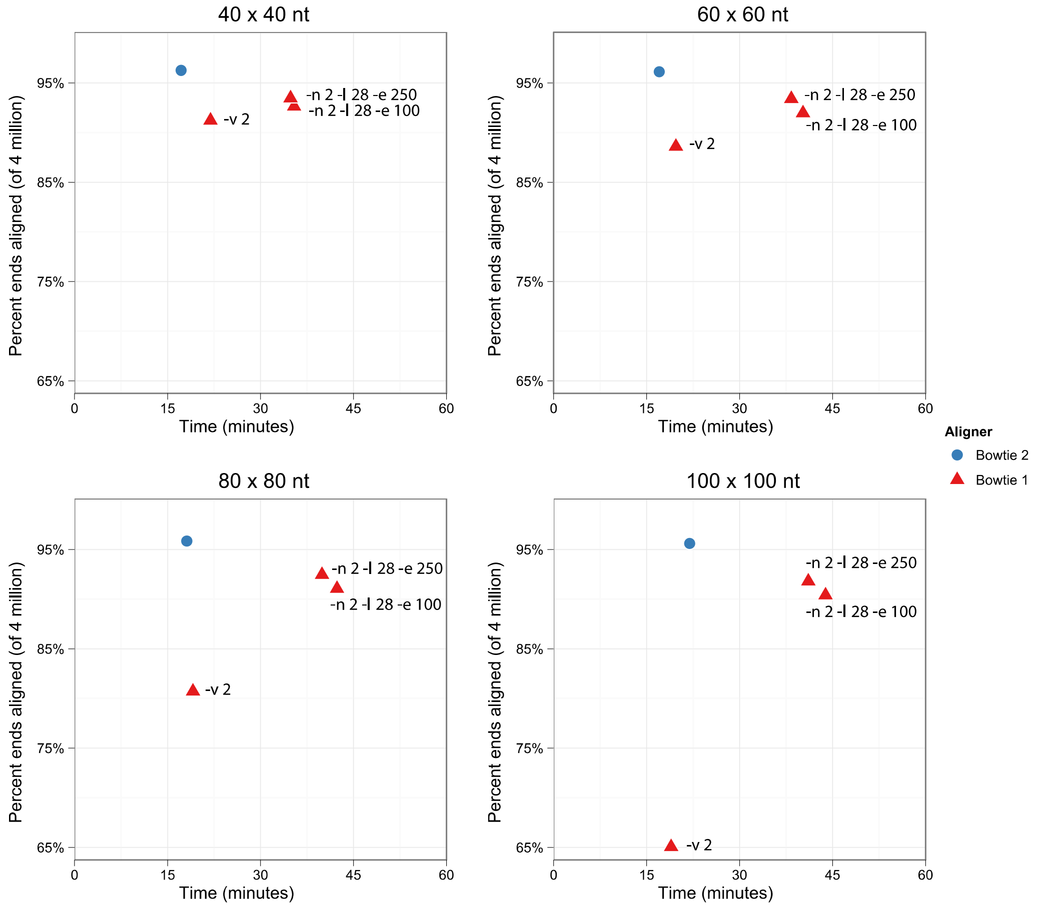
Sensitive and accurate alignment of simulated reads. Six datasets were simulated using the Mason simulator. For each aligner and each dataset, we plot the cumulative number of correct and incorrect alignments, accumulated from high to low mapping quality, on the vertical and horizontal axes. See Online Methods for details on how reads were simulated.

Supplementary Figure 3



Comparison of Bowtie 2 and Bowtie 1 aligning unpaired HiSeq 2000 reads. We ran Bowtie 2 and Bowtie 1 on the set of 100 nt unpaired HiSeq 2000 reads from **Figure 1a**. 40, 60 and 80 nt datasets were generated by trimming bases from the 3' end of the reads in the 100 nt dataset. Bowtie 2 was run in its default mode. We set Bowtie 1's reporting options to be comparable to Bowtie 2's defaults (-M 1 --best). Bowtie 1 was run in '-v 2' mode, which allows up to 2 mismatches in the entire alignment. Bowtie 1 was also run in '-l 28 -n 2' mode, which uses the first 28 nt of the read as a "seed" and allows at most 2 mismatches in that portion. The -e option sets a ceiling on the sum of the quality scores at mismatched positions, where quality scores are rounded to the nearest 10 and scores greater than 30 are rounded to 30. Full results are in **Supplementary Table 4**.

Supplementary Figure 4



Comparison of Bowtie 2 and Bowtie 1 aligning paired HiSeq 2000 reads. We ran Bowtie 2 and Bowtie 1 on the set of 100 x 100 nt paired HiSeq 2000 reads from **Figure 1b**. 40 x 40, 60 x 60 and 80 x 80 nt datasets were generated by trimming bases from the 3' end of the reads in the 100 x 100 nt dataset. The minimum and maximum insert lengths were set to 0 and 500 for both tools. We set Bowtie 1's reporting options to be comparable to Bowtie 2's defaults (-M 1 --best). Bowtie 1 was run in '-v 2' mode, which allows up to 2 mismatches in the entire alignment. Bowtie 1 was also run in '-l 28 -n 2' mode, which uses the first 28 nt of the read as a "seed" and allows at most 2 mismatches in that portion. The -e option sets a ceiling on the sum of the quality scores at mismatched positions, where quality scores are rounded to the nearest 10 and scores greater than 30 are rounded to 30. Note that Bowtie 2 will attempt to find and report alignments for each end separately if the ends cannot be aligned concordantly as a pair. Bowtie 1, on the other hand, reports no alignment for either end in this case. Full results are in **Supplementary Table 5.**

Supplementary Table 1

Aligner	Options	Label in Fig 1	Running time	% reads aligned (of 2 million)	Peak virtual memory footprint (gigabytes)
(a) Unpaired 100 nt HiSeq 2000 data					
Bowtie 2	-D 5 -R 1 -N 0 -L 22 -i S,0,2.50 (--very-fast)	1	6m:02s	94.62%	3.24
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,0,2.50 (--fast)	2	8m:08s	95.32%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.50		9m:15s	95.49%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.20		9m:23s	95.52%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.65		10m:36s	95.80%	3.24
Bowtie 2	**-D 15 -R 2 -N 0 -L 22 -i S,1,1.15 (--sensitive)**	3	11m:32s	95.92%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 21 -i S,1,1.00		12m:59s	96.03%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.75		16m:01s	96.07%	3.24
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.50		17m:43s	96.11%	3.24
Bowtie 2	-D 20 -R 3 -N 0 -L 20 -i S,1,0.50 (--very-sensitive)	4	23m:44s	96.26%	3.24
Bowtie 2	-D 25 -R 4 -N 0 -L 20 -i S,1,0.50		30m:20s	96.34%	3.24
Bowtie	-l 28 -n 2 -e 250 -M 1 --best	5	27m:56s	94.20%	2.34
BWA	-k 1 -l 32 -o 1	6	11m:42s	91.36%	2.39
BWA	-k 1 -l 32 -o 2		13m:05s	91.40%	2.44
BWA	-k 1 -l 28 -o 1		13m:55s	91.47%	2.40
BWA	-k 1 -l 32 -o 3		14m:41s	91.40%	2.52
BWA	-k 1 -l 28 -o 2		15m:48s	91.51%	2.48
BWA	-k 1 -l 24 -o 1	7	16m:50s	91.57%	2.41
BWA	-k 1 -l 28 -o 3		17m:25s	91.51%	2.56

BWA	-k 1 -l 24 -o 2		20m:42s	91.61%	2.51
BWA	-k 1 -l 24 -o 3		21m:17s	91.61%	2.59
BWA	**-k 2 -l 32 -o 1**	8	31m:24s	91.80%	2.41
BWA	-k 2 -l 28 -o 1		36m:01s	91.83%	2.41
BWA	-k 2 -l 32 -o 2		36m:43s	91.84%	2.51
BWA	-k 2 -l 32 -o 3		38m:25s	91.84%	2.59
BWA	-k 2 -l 28 -o 2		43m:13s	91.87%	2.52
BWA	-k 2 -l 24 -o 1		43m:17s	91.85%	2.42
BWA	-k 2 -l 28 -o 3		43m:44s	91.87%	2.59
BWA	-k 2 -l 24 -o 2		47m:52s	91.89%	2.53
BWA	-k 2 -l 24 -o 3		50m:09s	91.89%	2.63
SOAP2	-l 256 -v 3 -g 0		5m:20s	84.43%	5.34
SOAP2	**-l 256 -v 5 -g 0**	9	5m:23s	84.43%	5.34
SOAP2	-l 256 -v 7 -g 0		5m:30s	84.43%	5.34
SOAP2	-l 75 -v 5 -g 0		6m:20s	89.47%	5.34
SOAP2	-l 75 -v 7 -g 0	10	6m:22s	89.78%	5.34
SOAP2	-l 75 -v 3 -g 0		6m:33s	88.62%	5.34
SOAP2	-l 40 -v 7 -g 0	11	8m:44s	92.40%	5.34
SOAP2	-l 40 -v 5 -g 0		9m:15s	91.29%	5.34
SOAP2	-l 40 -v 3 -g 0		11m:34s	88.84%	5.34
(b) Paired-end 100 x 100 nt HiSeq 2000 data					
Bowtie 2	-D 5 -R 1 -N 0 -L 22 -i S,0,2.50 (--very-fast)	1	16m:09s	94.80%	3.25
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,0,2.50 (--fast)	2	17m:51s	95.04%	3.25
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.50		20m:08s	95.16%	3.25
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.20		20m:09s	95.20%	3.25
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.65		22m:43s	95.61%	3.26
Bowtie 2	**-D 15 -R 2 -N 0 -L 22 -i S,1,1.15 (--sensitive)**	3	25m:52s	95.90%	3.26

Bowtie 2	-D 15 -R 2 -N 0 -L 21 -i S,1,1.00		27m:16s	95.92%	3.26
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.75		30m:28s	95.98%	3.26
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.50		33m:16s	96.00%	3.26
Bowtie 2	-D 20 -R 3 -N 0 -L 20 -i S,1,0.50 (--very-sensitive)	4	44m:09s	96.19%	3.26
Bowtie 2	-D 25 -R 4 -N 0 -L 20 -i S,1,0.50		48m:24s	96.27%	3.26
Bowtie	-l 28 -n 2 -e 250 -M 1 -best -X 500	5	41m:01s	91.80%	3.01
BWA	-k 1 -l 32 -o 1	6	26m:23s	93.38%	3.20
BWA	-k 1 -l 32 -o 2		31m:39s	93.41%	3.20
BWA	-k 1 -l 28 -o 1		31m:40s	93.45%	3.20
BWA	-k 1 -l 32 -o 3		35m:17s	93.41%	3.20
BWA	-k 1 -l 28 -o 2		38m:03s	93.47%	3.20
BWA	-k 1 -l 24 -o 1	7	39m:11s	93.51%	3.20
BWA	-k 1 -l 28 -o 3		41m:40s	93.47%	3.20
BWA	-k 1 -l 24 -o 2		47m:02s	93.53%	3.20
BWA	-k 1 -l 24 -o 3		50m:53s	93.53%	3.20
BWA	**-k 2 -l 32 -o 1**	8	83m:58s	93.63%	3.20
BWA	-k 2 -l 32 -o 2		95m:28s	93.65%	3.20
BWA	-k 2 -l 28 -o 1		95m:35s	93.65%	3.20
BWA	-k 2 -l 32 -o 3		97m:04s	93.66%	3.20
BWA	-k 2 -l 24 -o 1		108m:30s	93.66%	3.20
BWA	-k 2 -l 28 -o 2		109m:31s	93.67%	3.20
BWA	-k 2 -l 28 -o 3		114m:09s	93.67%	3.20
BWA	-k 2 -l 24 -o 2		127m:20s	93.68%	3.20
BWA	-k 2 -l 24 -o 3		131m:48s	93.68%	3.20
SOAP2	-l 256 -v 3 -g 0 -m 250 -x 500		11m:10s	78.28%	5.34
SOAP2	**-l 256 -v 5 -g 0** -m 250 -x 500	9	11m:10s	78.28%	5.34
SOAP2	-l 256 -v 7 -g 0 -m 250 -x 500		11m:14s	78.28%	5.34
SOAP2	-l 75 -v 7 -g 0 -m 250 -x 500		12m:55s	86.97%	5.35

SOAP2	-l 75 -v 5 -g 0 -m 250 -x 500		12m:56s	86.20%	5.35
SOAP2	-l 75 -v 3 -g 0 -m 250 -x 500		13m:34s	84.27%	5.35
SOAP2	-l 75 -v 7 -g 3 -m 250 -x 500	10	16m:48s	90.63%	5.35
SOAP2	-l 75 -v 5 -g 3 -m 250 -x 500		17m:15s	89.23%	5.35
SOAP2	-l 256 -v 7 -g 3 -m 250 -x 500		17m:50s	87.76%	5.35
SOAP2	-l 256 -v 5 -g 3 -m 250 -x 500		17m:51s	86.26%	5.35
SOAP2	-l 75 -v 3 -g 3 -m 250 -x 500		17m:54s	85.97%	5.35
SOAP2	-l 256 -v 3 -g 3 -m 250 -x 500		18m:01s	81.19%	5.36
SOAP2	-l 40 -v 7 -g 0 -m 250 -x 500		21m:20s	89.11%	5.35
SOAP2	-l 40 -v 5 -g 0 -m 250 -x 500		22m:38s	87.45%	5.35
SOAP2	-l 40 -v 3 -g 0 -m 250 -x 500		25m:57s	84.29%	5.35
SOAP2	-l 40 -v 7 -g 3 -m 250 -x 500	11	26m:43s	92.08%	5.35
SOAP2	-l 40 -v 5 -g 3 -m 250 -x 500		27m:53s	90.07%	5.35
SOAP2	-l 40 -v 3 -g 3 -m 250 -x 500		30m:02s	86.04%	5.35
(c) 454 data					
Bowtie 2	-D 5 -R 1 -N 0 -L 25 -i S,1,2.0 --bwa-sw-like	1	58m:41s	98.29%	3.27
Bowtie 2	-D 5 -R 1 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		61m:12s	98.40%	3.27
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		63m:28s	98.51%	3.27
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,1,1.75 --bwa-sw-like	2	65m:05s	98.80%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		65m:25s	98.54%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.20 --bwa-sw-like		65m:59s	98.66%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.65 --bwa-sw-like		67m:09s	98.85%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.15 --bwa-sw-like		70m:08s	99.02%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 21 -i S,1,1.00 --bwa-sw-like		75m:28s	99.13%	3.27
Bowtie 2	**-D 15 -R 2 -N 0 -L 20 -i S,1,0.75** --bwa-sw-like	3	82m:14s	99.23%	3.27
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like		88m:06s	99.28%	3.28
Bowtie 2	-D 20 -R 3 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like	4	93m:55s	99.29%	3.28

Bowtie 2	-D 25 -R 4 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like		110m:39s	99.30%	3.28
BWA-SW	-c 5.5 -z 1 -s 1	5	83m:41s	98.12%	3.66
BWA-SW	-c 5.5 -z 1 -s 2		112m:26s	98.12%	3.66
BWA-SW	-c 5.5 -z 2 -s 1	6	124m:28s	98.74%	3.68
BWA-SW	**-c 5.5 -z 1 -s 3**	7	141m:00s	98.12%	3.66
BWA-SW	-c 5.5 -z 3 -s 1	8	161m:21s	98.82%	3.69
BWA-SW	-c 5.5 -z 2 -s 2		169m:20s	98.74%	3.68
BWA-SW	-c 5.5 -z 2 -s 3		213m:47s	98.74%	3.68
BWA-SW	-c 5.5 -z 3 -s 2		220m:01s	98.83%	3.69
BWA-SW	-c 5.5 -z 3 -s 3		276m:22s	98.82%	3.69

(d) Ion Torrent data

Bowtie 2	-D 5 -R 1 -N 0 -L 25 -i S,1,2.0 --bwa-sw-like	1	3m:52s	49.51%	3.37
Bowtie 2	-D 5 -R 1 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		4m:10s	49.64%	3.37
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		4m:55s	50.00%	3.37
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.50 --bwa-sw-like		5m:26s	50.09%	3.37
Bowtie 2	-D 10 -R 2 -N 0 -L 22 -i S,1,1.75 --bwa-sw-like	2	5m:30s	51.72%	3.37
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,2.20 --bwa-sw-like		5m:40s	50.74%	3.37
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.65 --bwa-sw-like		6m:20s	52.05%	3.37
Bowtie 2	-D 15 -R 2 -N 0 -L 22 -i S,1,1.15 --bwa-sw-like		7m:09s	53.11%	3.37
Bowtie 2	-D 15 -R 2 -N 0 -L 21 -i S,1,1.00 --bwa-sw-like		8m:13s	53.82%	3.38
Bowtie 2	**-D 15 -R 2 -N 0 -L 20 -i S,1,0.75** --bwa-sw-like	3	10m:00s	54.71%	3.38
Bowtie 2	-D 15 -R 2 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like		11m:48s	55.19%	3.39
Bowtie 2	-D 20 -R 3 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like	4	14m:20s	55.32%	3.39
Bowtie 2	-D 25 -R 4 -N 0 -L 20 -i S,1,0.50 --bwa-sw-like		17m:11s	55.38%	3.39
BWA-SW	-c 5.5 -z 1 -s 1	5	22m:16s	47.80%	3.66
BWA-SW	-c 5.5 -z 1 -s 2		23m:23s	47.80%	3.66
BWA-SW	**-c 5.5 -z 1 -s 3**	7	24m:26s	47.80%	3.66

BWA-SW	-c 5.5 -z 2 -s 1	6	37m:34s	51.58%	3.67
BWA-SW	-c 5.5 -z 2 -s 2		39m:19s	51.58%	3.67
BWA-SW	-c 5.5 -z 2 -s 3		40m:58s	51.58%	3.67
BWA-SW	-c 5.5 -z 3 -s 1	8	47m:14s	52.01%	3.67
BWA-SW	-c 5.5 -z 3 -s 2		49m:27s	52.01%	3.67
BWA-SW	-c 5.5 -z 3 -s 3		51m:30s	52.01%	3.67

Command-line arguments and full results from experiments using real data. The data in sections (a), (b), (c) and (d) above correspond to panels (a), (b), (c) and (d) of **Figure 1**. The tools' default parameter combinations are shown in boldface. Note that, for the comparisons to BWA-SW Bowtie 2's --bwa-sw-like option is used, which sets Bowtie 2 scoring parameters to mimic BWA-SW's. However, --bwa-sw-like is not enabled in Bowtie 2 by default.

Supplementary Table 2

Dataset	Bowtie 2 versus	Reads or ends aligned by neither	Reads or ends aligned by only Bowtie 2	Reads or ends aligned by only other tool	Reads or ends aligned by both
Unpaired HiSeq 2K	BWA	79,842 (3.99%)	84,136 (4.21%)	449 (0.09%)	1,834,243 (91.71%)
Paired HiSeq 2K	BWA	154,799 (3.87%)	99,852 (2.50%)	9,137 (0.23%)	3,736,212 (93.41%)
454	BWA-SW	7,458 (0.75%)	11,344 (1.13%)	266 (0.03%)	988,390 (98.84%)
Ion Torrent	BWA-SW	450,602 (45.06%)	71,423 (7.14%)	2,270 (0.23%)	475,705 (47.57%)

Alignment overlap between Bowtie 2 and BWA/BWA-SW. Shows the number and fraction of reads that are aligned by both tools, only one tool, or neither tool.

Supplementary Table 4

Aligner	Options	Running time	% reads aligned (out of 2 million)	Peak virtual memory footprint (gigabytes)
Length: 40 nt				
Bowtie 2	(defaults)	4m:27s	96.40%	3.35
Bowtie 1	-v 2 -M 1 --best	11m:00s	95.81%	2.34
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best	22m:02s	96.39%	2.34
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best	20m:48s	96.95%	2.34
Length: 60 nt				
Bowtie 2	(defaults)	6m:09s	96.55%	3.24
Bowtie 1	-v 2 -M 1 --best	13m:16s	93.49%	2.34
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best	26m:36s	94.66%	2.34
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best	23m:33s	96.10%	2.34
Length: 80 nt				
Bowtie 2	(defaults)	9m:11s	96.21%	3.24
Bowtie 1	-v 2 -M 1 --best	14m:16s	89.05%	2.34
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best	29m:41s	93.57%	2.34
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best	26m:36s	95.07%	2.34
Length: 100 nt				
Bowtie 2	(defaults)	11m:56s	95.92%	3.24
Bowtie 1	-v 2 -M 1 --best	14m:37s	83.50%	2.34
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best	31m:48s	92.86%	2.34

Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best	28m:50s	94.20%	2.34

Comparison of Bowtie 2 and Bowtie 1 aligning unpaired HiSeq 2000 reads. We ran Bowtie 2 and Bowtie 1 on the set of 100 nt unpaired HiSeq 2000 reads described in Figure 1a and Table 1. We used options for Bowtie 1 that are comparable to Bowtie 2's default mode of searching for at least 2 alignments and reporting a representative alignment with mapping quality (-M 1 --best). Bowtie 1 was run in '-v 2' mode, which allows up to 2 mismatches in the entire alignment. Bowtie 1 was also run in '-l 28 -n 2' mode, which uses the first 28 nt of the read as a "seed" and allows at most 2 mismatches in that portion. The -e option sets a ceiling on the sum of the quality scores at mismatched positions, where quality scores are rounded to the nearest 10 and scores greater than 30 are rounded to 30. The results show that Bowtie 2 achieves a superior combination of speed and sensitivity with equal memory footprint. Bowtie 2's advantage is more pronounced for longer reads.

Supplementary Table 5

Aligner	Options	Running time	% reads aligned (out of 2 million)	Peak virtual memory footprint (gigabytes)
Length: 40 nt				
Bowtie 2	--sensitive -I 0 -X 500	17m:11s	96.26%	3.34
Bowtie 1	-v 2 -M 1 --best -I 0 -X 500	21m:55s	91.23%	3.01
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best -I 0 -X 500	35m:25s	92.63%	3.01
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best -I 0 -X 500	34m:50s	93.46%	3.01
Length: 60 nt				
Bowtie 2	--sensitive -I 0 -X 500	17m:02s	96.12%	3.28
Bowtie 1	-v 2 -M 1 --best -I 0 -X 500	19m:43s	88.60%	3.01
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best -I 0 -X 500	40m:13s	91.96%	3.01
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best -I 0 -X 500	38m:20s	93.40%	3.01
Length: 80 nt				
Bowtie 2	--sensitive -I 0 -X 500	18m:06s	95.84%	3.26
Bowtie 1	-v 2 -M 1 --best -I 0 -X 500	19m:06s	80.72%	3.01
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best -I 0 -X 500	42m:20s	91.06%	3.01
Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best -I 0 -X 500	39m:55s	92.47%	3.01
Length: 100 nt				
Bowtie 2	--sensitive -I 0 -X 500	21m:56s	95.60%	3.26
Bowtie 1	-v 2 -M 1 --best -I 0 -X 500	18m:57s	65.06%	3.01
Bowtie 1	-l 28 -n 2 -e 100 -M 1 --best -I 0 -X 500	43m:51s	90.40%	3.01

Bowtie 1	-l 28 -n 2 -e 250 -M 1 --best -I 0 -X 500	41m:05s	91.80%	3.01

Comparison of Bowtie 2 and Bowtie 1 aligning paired HiSeq 2000 reads. We ran Bowtie 2 and Bowtie 1 on set of 100 x 100 nt paired HiSeq 2000 reads described in Figure 1b and Table 2. The minimum and maximum insert lengths were set to 0 and 500 for both tools. We used options for Bowtie 1 that are comparable to Bowtie 2's default mode of searching for at least 2 alignments and reporting a representative alignment with mapping quality (-M 1 --best). Bowtie 1 was run in '-v 2' mode, which allows up to 2 mismatches in the entire alignment. Bowtie 1 was also run in '-l 28 -n 2' mode, which uses the first 28 nt of the read as a "seed" and allows at most 2 mismatches in that portion. The -e option sets a ceiling on the sum of the quality scores at mismatched positions, where quality scores are rounded to the nearest 10 and scores greater than 30 are rounded to 30. Note that Bowtie 2 will attempt to find and report alignments for each end separately if the ends cannot be aligned concordantly as a pair. Bowtie 1, on the other hand, reports no alignment for either end in this case. Thus, this comparison lends a small speed advantage to Bowtie 1 and a sensitivity advantage to Bowtie 2.

Supplementary Table 6

Aligner	Options	Running time	% reads aligned (out of 200,000)	Peak virtual memory footprint (gigabytes)
Bowtie 2	(defaults)	39s	95.89%	3.24
BWA	(defaults)	1m:42s	91.81%	2.32
SOAP2	(defaults)	31s	84.45%	5.32
GSNAP	(defaults)	20m:56s	93.99%	4.91
MOSAIK	-mm 15 -act 35 -bw 35 -mhp 100	30m:27s	95.64%	61.70
SHRiMP2	(defaults)	251m:38s	97.67%	36.90

Comparison including aligners not based on the FM Index. We ran Bowtie 2 (v2.0.0-beta4), BWA (v0.5.9), SOAP2 (v2.21), GSNAP (2011-03-28.v3), MOSAIK (v1.1.0021), and SHRiMP 2 (v2_2_0), using each tool to align the first 100,000 reads from the set of 100 nt unpaired HiSeq 2000 reads described in Figure 1a and Table 1. We ran each aligner with the options shown in the second column; we used default options for all tools but MOSAIK, where we used the recommended options for reads of around 100 nt. Not all tools are being run in comparable reporting modes; e.g. Bowtie, BWA and SOAP2 report one representative alignment for each input read by default, but GSNAP, SHRiMP, and MOSAIK report many alignments per read by default. Also, a substantial fraction of running time for MOSAIK and SHRiMP 2 is spent building the reference index, a cost that can be amortized in practice by aligning large collections of reads at once. For these reasons, and because only one set of parameters is tried for each tool, we emphasize that these results are not a comprehensive comparison of these tools. This experiment was run on a single Intel Xeon X5550 Nehalem 2.66GHz processor of a High-Memory Quadruple Extra Large Instance (m2.4xlarge) rented from Amazon's Elastic Compute Cloud (EC2) service. The instance had 68.4 gigabytes of physical memory and was running the Basic 64-bit Amazon Linux AMI 2011.02.1 Beta.

Supplementary Note

When aligning an unpaired read, Bowtie 2 proceeds in four steps, (**Supplementary Fig. 1**). In step 1, Bowtie 2 extracts substrings ("seed" strings) from the read and its reverse complement. In step 2, the seed strings are aligned to the genome in an ungapped fashion with the aid of the FM Index. In step 3, seed alignments are prioritized and their offsets with respect to the reference genome are determined. Step 4 takes prioritized, resolved alignments from step 3 and performs SIMD-accelerated dynamic programming alignment in the vicinity of each until all are examined, until a sufficient number of alignments were examined, or until the dynamic programming effort limit (described below) is reached. These steps are described in greater detail below.

Seed extraction. Substrings of the read ("seed strings") are extracted at regular intervals along the read and its reverse complement. Seed strings are contiguous (i.e. they are not spaced seeds) and may or may not overlap each other. For instance, if we extract a 20 nt substring every 10 nt along the read, consecutive substrings will overlap by 10 nt. If we extract a 18 nt substring every 20 nt, then substrings will not overlap and there will be a gap of 2 nt between adjacent substrings. Seed length is configured using Bowtie 2's -L option. The -L option can take any value from 4 through 32. Values for this option that performed well in our experiments ranged from 20 to 25.

When the input comprises reads of various lengths (e.g. for 454 or Ion Torrent data), it is advantageous to set the interval length using a sublinear function of read length. For instance, the default function used in Bowtie 2 end-to-end mode is $I(x) = \max(1, \mathrm{floor}(1 + 1.15 * \sqrt{x}))$, where I is interval length as a function of the length of the read, x. For a 100 nt read, this causes seeds to start 12 nt apart, with the first seed starting at offset 0 from the 5' end, the second seed starting at offset 12, etc. The constant term, coefficient and function used are configurable via Bowtie 2's -i option.

FM Index-assisted seed alignment. Given seed strings, Bowtie 2 then uses FM Index-assisted alignment to find ungapped alignments for each. The alignment process makes use of the same reference pruning, policy pruning and double indexing approaches used in Bowtie 1[1]. Bowtie 2 also uses bi-directional BWT[2], an approach that allows the aligner to efficiently switch between alignment in a right-to-left direction and alignment in a left-to-right direction.

Seed strings can be aligned with up to 1 mismatch. The number of mismatches to permit is configurable. Option -N 1 allows seed alignments to have up to 1 mismatch, whereas option -N 0 requires that seeds match exactly.

Seed alignment prioritization. The output from the seed alignment step is a set of zero or more Burrows-Wheeler ranges per seed string. We call such a range a “seed-hit range.” A seed-hit range describes a range of rows in the Burrows-Wheeler matrix that begin with a reference substring that is within 0 or 1 mismatches of the seed substring. A single seed string may be associated with multiple Burrows-Wheeler ranges, since a seed string may be within 1 mismatch of many distinct reference substrings. Each row of each seed-hit range corresponds to a locus on the reference genome where we might search for a full alignment. Bowtie 2 assigns a priority to each row (i.e. locus) equal to $1/r^2$ where r is the total number of rows in the range. E.g. a row from a seed-hit range with 3 elements gets 1/9th the weight of a row from a seed-hit range with 1 element.

In this step, Bowtie 2 proceeds by repeatedly selecting a row in a random weighted fashion using these weights. When a row is selected, its offset into the reference genome is calculated using the typical FM Index algorithm, which applies the “LF mapping” properly repeatedly until a “marked row” is reached, at which point the offset can be resolved with a lookup (see section 3.2 of the FM Index publication[3]). We call this the “walk-left” procedure. Each resolved offset is passed to the SIMD-accelerated dynamic programming algorithm along with information about which seed string gave rise to the hit.

SIMD-accelerated dynamic programming. For each resolved seed hit, Bowtie 2 extracts flanking characters from the reference and solves a rectangular dynamic programming problem to find high-scoring full alignments in the vicinity of the seed hit. Dynamic programming alignment algorithms such as Needleman-Wunsch[4], Smith-Waterman[5], and extensions thereof[6] enable efficient computation of the optimal alignment between two sequences, even in the presence of many gaps and mismatches.

Dynamic programming algorithms can be visualized as acting on a matrix with rows corresponding to characters in the read and columns corresponding to characters in the reference. The algorithm calculates all elements in the matrix moving from upper left corner to the lower right, with each element (i, j) set to the alignment score that results from aligning the length-i prefix of the read to the length-j prefix of the reference. Because a given cell (i, j) can be calculated by considering only values in the cells above $(i\text{-}1, j)$, to the left $(i, j\text{-}1)$ and to the upper-left $(i\text{-}1, j\text{-}1)$, it is possible to parallelize these algorithms. Consider, for instance, a matrix for which all the elements in the first N anti-diagonals have already been calculated. All of the elements in the N+1th anti-diagonal can be calculated simultaneously in parallel. That is, the inputs to the calculations are available in previous anti-diagonals and none of the calculations depend on each other.

Many parallel dynamic programming approaches have been proposed, including implementations using single-instruction multiple-data (SIMD) instructions (also called “vector” or “streaming” instructions) available on general purpose CPUs[7-9]. Bowtie 2 builds on the approach used by the swsse2 tool[7], which fills striped, vertical chunks of the dynamic programming matrix with the help of SIMD instructions available on modern Intel and AMD computer processors. Because the chunks are oriented vertically, the read can be preprocessed into a “query profile,” and diagonal score contributions can be calculated with a lookup table. Because the chunks are “striped” along the read, the work required to propagate vertical contributions is reduced compared to non-striped approaches.

While the swsse2 tool is geared toward scoring protein alignments, Bowtie 2 adapts and extends the approach for read alignment. Specifically, Bowtie 2’s approach (a) works for end-to-end alignment in addition to local alignment, (b) implements a restriction on which positions may contain gaps, (c) implements separately configurable read and reference gap penalties, (d) permits scoring functions that account for quality values, and (e) implements a backtrace procedure so that alignments can be derived directly from the algorithm’s output.

Dynamic programming effort limit. Reads with seed strings that match many places on the genome can spur an excessively large number of dynamic programming problems. A homopolymer of Ts, for instance, could match hundreds of thousands of loci in the genome. Bowtie 2 avoids executing an excessive number of dynamic programming problems by imposing a ceiling on the number of dynamic programming attempts that can “fail” consecutively. We say an attempt “fails” if it fails to yield an alignment with a score that exceeds the best or the second-best alignment found so far. If the ceiling is set to 15, for example, and Bowtie 2 attempts 16 dynamic programming alignments in a row that fail, Bowtie 2 will simply report the alignments found so far and move on to the next read. This ceiling is set with the -D option.

Reseeding. Bowtie 2 has a “reseeding” facility designed to maximize accuracy with respect to reads with repetitive seed strings. After the seed string alignment step (step 2), Bowtie 2 will calculate the average number of seed hits per seed string and, if this average rises above a certain threshold (1,000 by default), the read is classified as having repetitive seed strings. If a read is classified as such, alignment for that read proceeds in multiple “rounds” where, before each round, Bowtie 2 extracts a new set of seed strings. This is called “re-seeding.” For instance, if Bowtie 2 is configured to extract a 20 nt seed every 10 positions and is aligning a read that has been classified as having repetitive seeds, it will proceed in two rounds: in the first round it extracts a 20 nt seed every 10 positions starting at offset 0 from the 5’ end, and in the second round it does the same but starting at offset 5 from the 5’ end. By re-seeding in this way, Bowtie 2 increases the chance that it will find the best alignment for the read. The maximum number of re-seeding rounds is set with option -R.

Paired-end alignment. Bowtie 2 supports alignment of pairs of reads (variously called "paired ends" or "mate pairs") in which both ends of a single DNA fragment are sequenced. The user sets expected minimum and maximum fragment lengths as well as expected orientations of the ends. A paired-end alignment that matches these expectations is called “concordant” and an alignment that violates these expectations is “discordant.” If a pair fails to align in a concordant fashion, Bowtie 2 attempts to align each end in an unpaired fashion. This is similar to both BWA’s and SOAP2’s behavior. When a pair fails to align concordantly but both ends align uniquely in an unpaired fashion, Bowtie 2 reports this as a “discordant” alignment.

The procedure for aligning a paired-end read largely follows the four steps described above. These steps are run first for one end of the paired-end read, then for the other. When the dynamic programming algorithm discovers a full alignment for one end (the “anchor” end), Bowtie 2 performs an additional step: it calculates the reference window where the other end (the “opposite” end) could potentially appear, given the user-configurable minimum and maximum fragment lengths and orientations (set with the -I, -X, --ff, --fr, and --rf options). Bowtie 2 then uses SIMD-accelerated dynamic programming alignment to search for a high-scoring alignment of the opposite end in that window. Bowtie 2 proceeds in this way until all anchor ends and their respective windows have been examined, until a sufficient number of alignments have been found, or until the dynamic programming effort limit is reached.

Bowtie 2’s unpaired and paired-end alignment modes can be set independently of its end-to-end and local alignment modes. For instance, it is possible to run Bowtie 2 in a mode that is both paired-end and local, such that both ends of the pair might align locally. This is in contrast to BWA and SOAP2, which allow only one end to align locally in their paired-end alignment modes.

Supplementary Results

Accuracy and sensitivity comparisons on simulated data. See main text and Online Methods. In addition, we tested simulated unpaired, Illumina-like datasets of lengths 75, 125 and 175 nt and paired-end, Illumina-like datasets of lengths 75 x 75 nt, 125 x 125 nt and 175 x 175 nt. We then ran Bowtie 2, BWA, and SOAP2 with their default arguments and evaluated with the same methodology described in the main text (**Supplementary Fig 2**). The read length was set with Mason’s ‘-n’ option.

Comparison of Bowtie 1 and Bowtie 2. To see how Bowtie 1 and Bowtie 2 compare in terms of speed and fraction of reads aligned, we ran Bowtie 2 and Bowtie 1 on the same set of 100 x 100 nt reads used in the Paired HiSeq 2000 comparison (**Supplementary Figs 2**, **3** and **Supplementary Tables 4, 5**). Bowtie 1 was designed to align relatively short reads (i.e. shorter than 100 nt), so here we compare Bowtie 1 and 2 for both unpaired and paired-end reads of length 40, 60, 80 and 100 nt. The 40, 60, and 80 nt datasets were constructed by trimming bases from the 3’ end of the 100 nt dataset.

The minimum and maximum insert lengths were set to 0 and 500 for both tools. Bowtie 2 was run in its default mode. We set Bowtie 1’s reporting options to be as comparable as possible to Bowtie 2’s defaults (-M 1 --best). Bowtie 1 was run in ‘-v 2’ mode, which allows up to 2 mismatches in the entire alignment. Bowtie 1 was also run in '-l 28 -n 2' mode, which uses the first 28 nt of the read as a "seed" and allows at most 2 mismatches in that portion. The -e option sets a ceiling on the sum of the quality scores at mismatched positions, where quality scores are rounded to the nearest 10 and scores greater than 30 are rounded to 30. Two settings for -e were used: 100, and 250.

Note that Bowtie 2 will attempt to find and report alignments for each end separately if the ends cannot be aligned concordantly as a pair. Bowtie 1, on the other hand, reports no alignment for either end in this case. Thus, the paired-end comparison (**Supplementary Fig 3** and **Supplementary Table 5**) lends a small speed advantage to Bowtie 1 but gives a substantial sensitivity advantage to Bowtie 2.

Comparison to additional tools. To assess the fraction of reads aligned by Bowtie 2 versus other tools, we compared the fraction of reads aligned by Bowtie 2, BWA[10], SOAP2[11], GSNAP[12], MOSAIK (http://bioinformatics.bc.edu/marthlab/Mosaik), and SHRiMP 2[13] for a subset of 200,000 reads from the unpaired HiSeq 2000 dataset examined in **Fig 1a**.

We used default options for all tools except MOSAIK, where we used the recommended options for reads of around 100 nt. Not all tools were run in comparable reporting modes; e.g. Bowtie 2, BWA and SOAP2 report one representative alignment for each input read by default, but GSNAP, SHRiMP, and MOSAIK report many alignments per read by default. Also, a substantial fraction of running time for MOSAIK and SHRiMP 2 is spent building the reference index, a cost that can be amortized in practice by aligning large collections of reads at once. For these reasons, and because only one set of parameters was tried for each tool, we emphasize that these results do not constitute a comprehensive comparison of these tools. Rather, the purpose is to get a rough impression of how the tools compare in terms of speed and fraction of reads aligned.

This experiment was run on a single Intel Xeon X5550 Nehalem 2.66GHz processor of a High-Memory Quadruple Extra Large Instance (m2.4xlarge) rented from Amazon's Elastic Compute Cloud (EC2) service. The instance had 68.4 gigabytes of physical memory and was running the Basic 64-bit Amazon Linux AMI 2011.02.1 Beta.

Literature Cited

1. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S.L. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol* **10**, R25 (2009).
2. Lam, T. et al. 31-36 (IEEE, 2009).
3. Ferragina, P. & Manzini, G. 390-398 (IEEE, 2000).
4. Needleman, S.B. & Wunsch, C.D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* **48**, 443-453 (1970).
5. Smith, T.F. & Waterman, M.S. Identification of common molecular subsequences. *J Mol Biol* **147**, 195-197 (1981).
6. Gotoh, O. An improved algorithm for matching biological sequences. *J Mol Biol* **162**, 705-708 (1982).
7. Farrar, M. Striped Smith-Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics* **23**, 156-161 (2007).
8. Rognes, T. Faster Smith-Waterman database searches with inter-sequence SIMD parallelisation. *BMC Bioinformatics* **12**, 221 (2011).
9. Rognes, T. & Seeberg, E. Six-fold speed-up of SmithñWaterman sequence database searches using parallel processing on common microprocessors. *Bioinformatics* **16**, 699 (2000).
10. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**, 1754-1760 (2009).
11. Li, R. et al. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics* **25**, 1966-1967 (2009).
12. Wu, T.D. & Nacu, S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics* **26**, 873-881 (2010).
13. David, M., Dzamba, M., Lister, D., Ilie, L. & Brudno, M. SHRiMP2: sensitive yet practical SHort Read Mapping. *Bioinformatics* **27**, 1011-1012 (2011).