SUPPLEMENTAL MATERIAL

RazerS 3: Faster, fully sensitive read mapping

David Weese, Manuel Holtgrewe and Knut Reinert

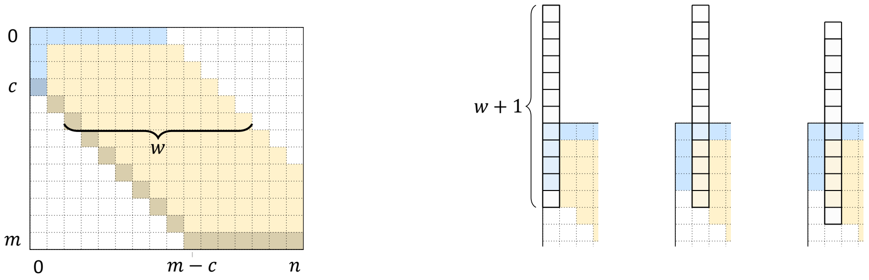
August 26, 2012

S1 BANDED EDIT DISTANCE VERIFICATION

We propose a bit-parallel banded approximate string search algorithm that can be used to search for semi-global alignments between a pattern and a text within a parallelogram of the DP matrix. For a given text t of length n and a given pattern p of length m we consider the DP matrix which has $m+1$ rows and $n+1$ columns. Let a band of w consecutive diagonals be given where the left-most diagonal is the main diagonal shifted by c diagonals to the left, see Figure S1a. The algorithm diagonally slides a column vector D of $w+1$ cells over the band. D is encoded by delta bit-vectors VP and VN of size w and a variable $score$ that tracks the cell values of the lower band boundary (dark cells in Figure S1a). Each sliding step consist of a horizontal and a vertical step. In the horizontal step the delta vectors $D0$, HP and HN are computed as in Myers' algorithm, see Figure S1b and lines 12–15 in Algorithm 1. Then VP and VN of the next column are deduced from these delta vectors and shifted by 1 bit to the right in the vertical step, see Figure S1b and lines 16–18.

In the beginning, D covers the intersection of the first column and all band diagonals plus the diagonal left of the band as shown Figure S1b. As D initially represents cells beyond the DP matrix, they have to be initialized such that they have no unintended influence on the cells within the DP matrix and such that the first DP row contains zero values for the approximate search or increasing values for the edit distance calculation. Setting the pattern bitmasks to zero for cells beyond the DP matrix, $VN = 0^w$ and $VP = 1^w$ for the approximate search or $VP = 1^{c+1}0^{w-c-1}$ for edit distance results in the desired initialization patterns depicted in Figure S2.

Before the pattern bitmasks can be used to compute the next D vector, they need to be shifted by 1 bit to the right and for $pos + c < m$ in one bitmask bit w must be set to represent the next pattern character $p[pos + c]$. This is done in line 11. Eventually, $score$ must be tracked properly. As long as the second



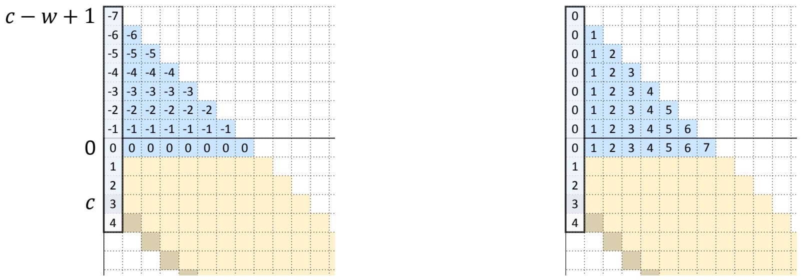
(a) band parameters

(b) compute next vector and shift down

Figure S1: Band parameters (a). For two sequences of length m and n the band is uniquely defined by the number of consecutive diagonals w and the row c that intersects the left-most diagonal and the first column. The initial state and the two substeps of one DP recursion step are shown on the right.

Algorithm 1: bandedMyers(t, p, k, w, c)

```
input  : text $t$, pattern $p$, errors $k$, and parameters $w$, $c$
output : text end positions of matches with up to $k$ errors

foreach $x \in \Sigma$ do                                           // initialize pattern bitmasks
    $B[x] \leftarrow 0^{w}$
for $j \leftarrow 0$ to $c - 1$ do
    $B\big[p[j]\big] \leftarrow B\big[p[j]\big] \mid 0^{c-j-1}10^{w-c+j}$

$VP \leftarrow 1^{w}, VN \leftarrow 0^{w}$                                     // initialize bit-vectors
$score \leftarrow c$

for $pos \leftarrow 0$ to $n - 1$ do
    foreach $x \in \Sigma$ do                                       // shift pattern bitmasks
        $B[x] \leftarrow B[x] \gg 1$
    if $pos + c < m$ then
        $B\big[p[pos + c]\big] \leftarrow B\big[p[pos + c]\big] \mid 10^{w-1}$

    $X \leftarrow B\big[t[pos]\big] \mid VN$                              // compute horizontal bit-vectors
    $D0 \leftarrow \big((VP + (X \mathbin{\&} VP)) \wedge VP\big) \mid X$
    $HN \leftarrow VP \mathbin{\&} D0$
    $HP \leftarrow VN \mid \sim (VP \mid D0)$
    $X \leftarrow D0 \gg 1$                                  // compute and shift vertical bit-vectors
    $VN \leftarrow X \mathbin{\&} HP$
    $VP \leftarrow HN \mid \sim (X \mid HP)$

    if $pos \leq m - c$ then                                            // scoring and output
        $score \leftarrow score + 1 - \big((D0 \gg (w - 1)) \mathbin{\&} 1\big)$
    else
        $s = (w - 2) - \big(pos - (m - c + 1)\big)$
        $score \leftarrow score + \big((HP \gg s) \mathbin{\&} 1\big)$
        $score \leftarrow score - \big((HN \gg s) \mathbin{\&} 1\big)$
    if $pos \geq m - c$ and $score \leq k$ then
        report occurrence ending at $pos$
```



(a) approximate search

(b) edit distance

Figure S2: Initialization of the column vector for approximate search (a) or edit distance computation (b).

last cell of D is within the DP matrix the last bit of $D0$ is used to track $score$ down the left-most band diagonal in line 20. Otherwise, the horizontal deltas of the last matrix row are used in lines 23 and 24 to update $score$. A simple heuristic allows to stop the verification earlier and improve the average running time. Cell values along a DP diagonal are monotonically increasing from top to bottom and $score$ can only decrease along the last matrix row. The last row contains $n + c - m$ band cells and thus the search can be stopped if $score > k + n + c - m$.

S2 SPECIFICITY TRADE-OFFS

As described in Section 3.1, the pigeonhole filter achieves better running times than the SWIFT filter for smaller error rates and for edit distance. This statement is made for the parallel execution of RazerS 3 using the new banded verification. In this section, we will consider the improvements contributed by both the introduction of banded verification and the pigeonhole filter seperately.

S2.1 Influence of banded verification

Table S1 shows the total mapping time, average filtration time, and average verification time when running Razers 3 with one and eight threads with different combinations of filter and verification variants. The number of verifications (candidates) and successful verifications (matches) is also shown. The upper half of the table shows the results when RazerS 3 is run with one thread. We mapped 100 bp reads from read set ERR012100 against chr. 2 of the human genome. RazerS 3 was run with 99 % sensitivity (default setting) and an error rate of 4 %.

When using one thread, the verification time using SWIFT and unbanded Myers is 34.5 minutes while it is 10.13 minutes when using banded Myers instead. When using the pigeonhole filter, the time decreases

filter	Myers variant	threads	avg. filtration time	avg. verification time	total time	candidates	matches
SWIFT	unbanded	1	14:14:04	34:33	14:49:06	489 M	29 M
SWIFT	banded	1	13:42:41	10:08	13:53:19	489 M	24 M
pigeonhole	unbanded	1	12:28	5:10:45	5:23:41	5 490 M	24 M
pigeonhole	banded	1	12:25	56:41	1:09:34	5 490 M	24 M
SWIFT	unbanded	8	1:01:13	4:27	1:05:47	489 M	29 M
SWIFT	banded	8	1:04:25	1:17	1:11:09	489 M	29 M
pigeonhole	unbanded	8	2:31	39:22	41:59	5 490 M	24 M
pigeonhole	banded	8	2:36	6:41	9:24	5 490 M	24 M

Table S1: Running time of the RazerS 3 mapping step in different variants, excluding I/O. The time is given as [h:min:s]. Note that the column "total time" also includes the I/O of references as well as running time for match management.

from 310.75 minutes to 69.56 minutes. The speedup thus is 3.4 for SWIFT and 4.47 for pigeonhole.

We also note that the running time of the filtration step also decreases slightly when switching from unbanded to banded verification. This can be explained by the new verification algorithm not needing any precomputation data and having a better memory locality. Thus the filtration step can use more of the processor cache.

When using eight threads, the pigeonhole/unbanded variant uses 39.37 minutes for verification, pigeonhole/banded uses 9.3 minutes, a speedup of 4.2. The SWIFT/unbanded variant uses 4.45 minutes, SWIFT/banded uses 1.28 minutes, a speedup of 3.47.

Interestingly, the total mapping time does not decrease as greatly for SWIFT when changing from banded to unbanded while there is a speedup for pigeonhole. Thus, we also have to investigate the influence of the pigeonhole filter.

S2.2 Influence of the pigeonhole filter

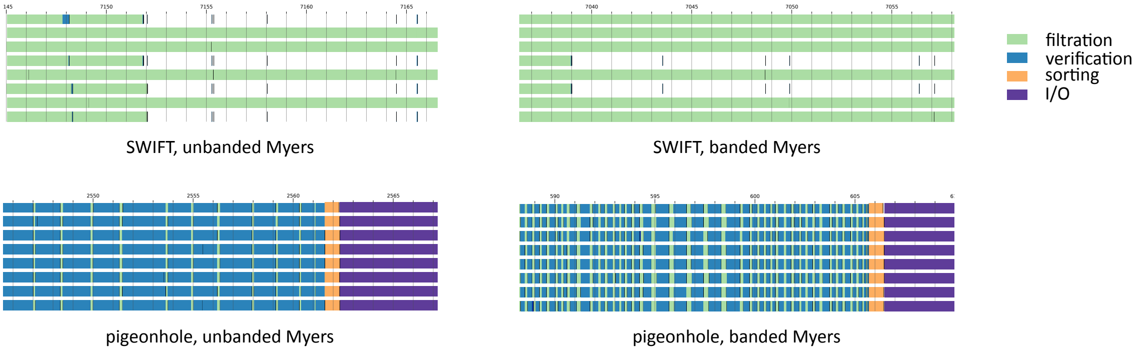
Using the SWIFT filter identifies 489 M candidates while the pigeonhole filter identifies 5 490 M candidates: More than 11 times as many (see Table S1). However, using the pigeonhole filter, RazerS 3 is faster than when using the SWIFT filter. Depending on whether using one or eight threads and banded or unbanded verification, the difference in mapping time is a factor between 1.56 (eight threads, unbanded) and 11.98 (one thread, banded). The factor is 2.75 for one thread, unbanded and 7.56 for eight threads, banded).

When changing the filter from SWIFT to pigeonhole, the filtration time decreases greatly, at the cost of more candidates and thus more time spent in verification. The time spent in verification can be greatly reduced by using the banded instead of the unbanded verification algorithm.

Another interesting effect of the faster verification phase is the impact on load balancing. We instrumented the RazerS 3 code and recorded the start and end time of each step of each step and thus also the times where a thread waits for others. Figure S3 shows excerpts from the time charts of the first two threads from each run towards the end of the run. The x-axis indicates the time, where all charts have the same time scale. Each green bar represents the time spent for filtering candidate regions between an eighth of all reads and a reference window of length 500 kbp.

The blue bars represent time spent for verification (which is load balanced in packages between the blocks). Colored parts of the bar indicate that the thread was actively doing work whereas white parts indicate waiting.

As can be seen, when using the SWIFT filter, running time is dominated by filtration. Because filtration is not load balanced dynamically, multiple threads are idle when they have completed their filtration.



SWIFT, unbanded Myers

SWIFT, banded Myers

pigeonhole, unbanded Myers

pigeonhole, banded Myers

Figure S3: Running time profiles for RazerS 3 with different variants for the filtration and verification phase. The charts show the work performed by each of the 8 threads towards the end of the program run. In pigeonhole mode, all threads finish almost simultaneously because the time spent for verification dominates the running time and the verification can be load balanced dynamically. In SWIFT mode, many threads are idle from some point of time since the filtration time dominates and filtration is load balanced only statically. Note that while the charts all have the same scale and show an interval of the same length, they all show an interval starting at different points of times. For the pigeonhole filter case, the times are taken around the end of the mapping phase. In the case of the SWIFT filter this was not possible on this scale since the threads finish so irregularly.

When running with the pigeonhole filter, the running time is dominated by verification. Verification can be load balanced dynamically and consequently, the threads finish almost simultaneously. This can be seen in the two charts on the bottom which show the last 25 s of the pigeonhole runs. Furthermore, we can observe that the time spent on verification is shorter when using banded verification but the remaining verification work still allows good load balancing.

S3 EVALUATION OF THE PARALLELIZATION

This section analyzes the employed parallelization of RazerS 3 in detail.

S3.1 Achieved speedup

To evaluate how much our implementation benefits from widely available multi-core architectures, we mapped a relatively large dataset (10 M reads of set ERR012100) against chr. 2 of the human genome. We ran RazerS 3 with 1, 2, 4, and 8 threads (*dynamic* load balancing). The results were compared with the trivial parallelization (*static* load balancing) of splitting the read set into t parts of equal size and running t separate RazerS 3 processes in parallel that use one thread each.

Both the runs with dynamic and static load balancing required about 89.5 min with one thread. Mapping reads with dynamic load balancing scaled almost linearly with speedups of 1.95, 3.95, and 7.46 for 2, 4, and 8 threads. Static load balancing scaled worse: The speedups were 1.90, 3.63, and 6.61. With 8 threads we effectively gained one more processor core with our dynamic balancing scheme compared to the static load balancing.

S3.2 Alternative load-balancing methods

In Section S3.1 we compared static and dynamic load balancing schemes. Another possible load balancing scheme would split the read set into many packages which are then distributed through a queue to a group of worker threads. Each thread processes one package in a sequential fashion and the next package is given to the earliest idle thread.

This approach has a major drawback: The filtration step of RazerS 3 has to scan over the text for each package. While the time for verification is linear in the number of matches and the number of verifications does not change, this is not the case for the filtration: Doubling the number of packages, leads to an increase by a factor of 1.7−1.8 for the total CPU time spent in filtration in our experiments.

10 M reads were mapped against the D. melanogaster genome. We split the reads into 8, 16, 32, and 40 packages. Each package was then mapped using a separate RazerS 3 process. The work queue program `ts`[1] was used for running the mapping processes such that 8 were run in parallel. The CPU time spent for filtering was 216 s, 388 s, 694 s, and 1025 s using a splitting scheme of 8, 16, 32, and 40 packages. When using dynamic load balancing, the time for filtration was 243 s (the difference to the time for 8 packages can be explained by the better isolation and fewer synchronizations using separate processes).

When using 8 packages, the load balancing is the same as for the static scheme described in Section S3.1. While there is no additional overhead for the filtration, the load balancing is inferior to the dynamic load balancing implemented in RazerS 3. Using more packages, the overhead for the additional filtration makes this approach even worse than static load balancing.

S4 OPEN ADDRESSING Q-GRAM INDEX

For each thread a q-gram index stores the occurrences of all overlapping read q-grams. We changed the q-gram index implementation that originally used the rank of a q-gram in the set of all possible q-grams to address the q-gram index. The index implementation in RazerS 3 uses an open addressing scheme which reduces the memory consumption from originally $\mathcal{O}(|\Sigma|^q + n)$ to $\mathcal{O}(n)$ where n is the number of overlapping q-grams. Through this change the overall memory consumption of RazerS 3 by the q-gram index does not depend on the number of threads.

[1] http://vicerveza.homeunix.net/~viric/soft/ts/

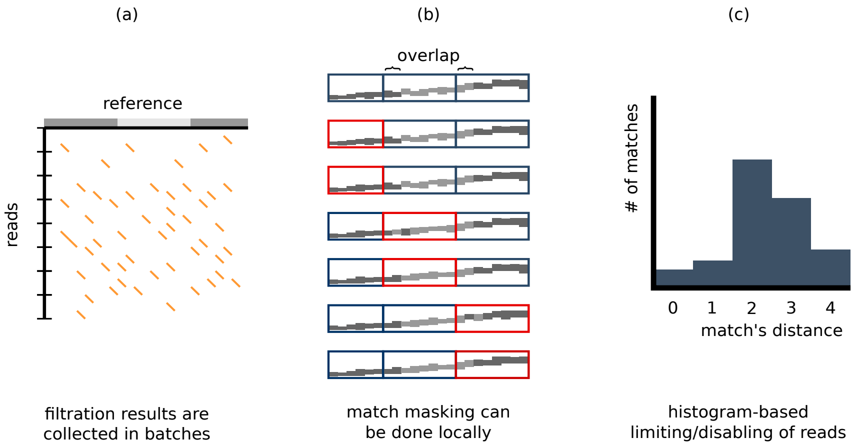

(a)
reference
reads
filtration results are collected in batches
(b)
overlap
match masking can be done locally
(c)
# of matches
0
1
2
3
4
match's distance
histogram-based limiting/disabling of reads

Figure S4: (a) Window-based partitioning of the genome for filtration. (b) Window-based masking of matches. (c) Histogram of all matches of one read. In part (b), the bars represent matches where the lower border of a bar is the starting point of the match in the genome, the upper border is the end point of the match in the genome. The color of the bar show from which window the match comes from. Note that RazerS 3 stores one histogram for each read. For more information, see Sections 2.4 and S5.

S5 IMPROVED MATCH MANAGEMENT

Both the SWIFT and the pigeonhole filter slide a q-gram sequentially over the reference sequence and use the q-gram indices to look up its occurrences in the read set. Although the genomic position of the common q-grams is monotonically increasing, the position in the read may vary. Consequently, the left-most diagonals of found candidate parallelograms are not monotonically increasing in general. Thus, a match can have a lower coordinate than the previously found match for the same read. This non-monotonicity is limited by some relatively small overlap that can be computed from the filter settings.

Thus, the sorting for the masking step does not have to be performed on all matches. Masking can be done in windows on the match array:

First, all matches from the first window (the right border is shifted to the left by the overlap) are sorted by start, then by end position (as described above) and masked after each sorting. Then, all matches from the second window (the left border is shifted left by the overlap) are sorted by start, then by end position and masked after each sorting. This is repeated for all following windows. Thus, masking can be performed locally while still providing masking requirement (i). Requirement (ii) is ensured in a final compaction step. Lowering the filter sensitivity (iii) is done by keeping a histogram of found matches (counted after masking) for each read. This is illustrated in Figure S4.

This allows us to manage the arrays in memory mapped files and only access a relatively small part at the end of the file. Only this active part has to be kept in memory and the operating system will swap out the rest of the file if necessary. This strongly reduces the memory requirements of the program and makes them independent of the total number of matches. Rather, the memory required for matches is now bounded by the maximal number of matches of all reads that fall into one window. Note that the operating system will buffer as much of the file as possible in main memory for better performance. The sorting of the final compaction step can be done using external sorting for large result sets.

S6 READ MAPPER PARAMETRIZATION

In the following we describe the parameters, we used for the comparison with other read mappers. `MIN` and `MAX` were placeholders for minimal and maximal insert size, `INS` is the mean insert size and `IERR` the allowed deviation (`INS = (MIN + MAX) / 2`, `IERR = (MAX - MIN) / 2`). For the tools using indices, we built the index using default options.

Bowtie 2. Version 2.0.0-beta6 was used. The number of threads was selected using the parameter (`-p`). We used the parameter `--end-to-end` to enforce semi-global read alignments. For the Rabema experiment, we used the parameters `-k 100`. For all other experiments, we used the parameters `-k 1`. In paired-end mode, we used the parameters `--minins MIN --maxins MAX`.

BWA. Version 0.6.1-r104 was used. We used the parameter `-t` to select the number of reads in the `aln` step. The `sampe` and `samse` steps were performed using one thread since BWA does not offer a parallelization here. When mapping for the Rabema experiment, we passed the parameter `-N` to `aln` and `-n 100` to `samse`. Otherwise, we passed the parameter `-n 1` to `samse`. The insert size was not passed to BWA, however we pass the insert size and allowed error from BWA's output to the other read mappers.

Hobbes. Version 1.3 was used. Since we focus on edit distance, we used the 16 bit bit-vector version as described in (Ahmadi *et al.*, 2011). We built the index using the recommended[2] q-gram length 11. Indels were enabled using `--indels`. Maximal edit distance was set using `-v`. Multi-threading was enabled using `-p`. For resource measurement, we used the output without CIGAR, for analyzing the results, we enabled CIGAR output using `--cigar`. In paired-end mode, we used the parameters `--pe --min MIN --max MAX`.

mrFAST. Version 2.1.0.6 was used. It was used as explained in the manual[3]. mrFAST does not support multithreading. We divided the input into blocks of 500 k reads and processed each chunk in a separate process using the program `ts`[4]. Long reads were split into packages of 100 k reads. This way, always 8 processes were executed in parallel. We set the edit distance error rate to 4 % of the read length.

RazerS 3. Version 3.0 was used. RazerS 3 was parametrized as follows: The native or SAM output format was selected with `-of 0` or `-of 4`. Indel support was disabled with `--no-gaps` when required. The number of threads was set with the `-tc` parameter. The percent recognition rate was set using the `-rr` parameter, e.g. `-rr 100` or `-rr 99`. The error rate was set through the `-i` parameter, e.g. `-i 96` to map with 4 % errors[5]. The pigeonhole or SWIFT filter was selected using `-fl pigeonhole` or `-fl swift`. As an all-mapper, the parameter `-m 1000000` was used and as a best-mapper `-m 1` was used. In paired-end mode, the parameters used were `--library-length INS --library-error IERR`.

SHRiMP 2. Version 2.2.2 was used. The number of threads was selected with `--threads`. In paired-end mode, the options used are `--pair-mode opp-in --isize MIN,MAX`.

Soap 2. Version 2.1 was used. The number of threads was selected with `-p`. In paired-end mode, the options used are `-m MIN -x MAX`.

S7 MEMORY REQUIREMENTS

As mentioned earlier, RazerS 3 improves on the memory requirements of RazerS by using open addressing for the q-gram index. The peak memory consumption can be determined as follows: First, contigs of the reference genome are processed and loaded one after another, thus the size C of the largest contig is one summand. Another summand is the total number of bases in the input read set, e.g. $n \cdot m$ for n Illumina reads of length m. Then, each thread uses a q-gram index whose size is linear in the number of contained q-grams. Therefore, the overall size of the indices is $\mathcal{O}(n \cdot (m - \max Q))$ when using a SWIFT filter with q-grams of shape Q, or $\mathcal{O}(n \cdot m/\Delta)$ when using a (q, Δ)-seed pigeonhole filter. Finally, enough space for the matches has to be allocated, which can be estimated by $\mathcal{O}(n \cdot \alpha)$ where α is the average number of matches per read.

Practically, we observe a peak memory consumption of about 10 GB when mapping 10 M reads against the human genome with indels and an error rate of 4 %. This amount grows linearly in the number of reads, i.e. the memory consumption for 20 M reads is about 20 GB.

[2] http://hobbes.ics.uci.edu/manual.jsp

[3] http://mrfast.sourceforge.net/manual.html

[4] http://vicerveza.homeunix.net/~viric/soft/ts/

[5] RazerS uses the percent identity, which is 100 minus error rate in percents.

S8 EXPERIMENTAL MAPS READ SETS

Table S2 shows how the read sets were obtained for computing the experimental maps. Since we were not able to obtain read sets with consistent lengths for each reference organism, we had to trim the reads in most cases.

reference	read length	read set ID	original length
E. coli	100	ERR022075	100
E. coli	70	ERR022075	100
E. coli	50	ERR022075	100
E. coli	30	ERR032371	36
C. elegans	100	SRR065390	100
C. elegans	70	SRR065390	100
C. elegans	50	SRR065390	100
C. elegans	30	SRR107574	34
human chr. 2	100	ERR012100	101
human chr. 2	70	SRR029194	88
human chr. 2	50	SRR029194	88
human chr. 2	30	ERR003244	37

Table S2: This table gives information which datasets were used when creating the experimental maps. If original length and read length m are not equal then the first m bases were used.

S9 SIMULATION DETAILS

We simulated reads with Mason (Holtgrewe, 2010) using the Illumina model. The haplotype SNP rate was set to 0.6 %, indel rate to 0.1 %, and indel size was randomly chosen between 1 and 32. This was the variation rate we observed from the 1000 genomes individual NA19240. For the Rabema and variation detection experiments, the read length was set to 100 bp. Here, sequencing errors were simulated using an indel probability of 0.03 % at each position and replacements were simulated with the standard model using a factor of 3. The positional replacement, insert, and deletion probabilities can be seen in Figure S5. The probabilities were determined after realigning the reads with the Needleman-Wunsch algorithm and a score that slightly prefers mismatches over gaps. This explains the artifacts of the curves at the ends and in the center.

For the long simulated reads, we used an indel probability of 0.01 % and used the standard model with a factor of 1.

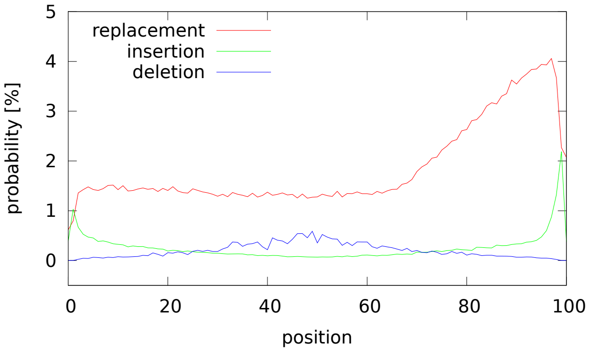


Figure S5: Positional replacement, insert, and deletion probabilities for the simulated reads in the Rabema experiment.

S10 FULL EXPERIMENTAL MAPS

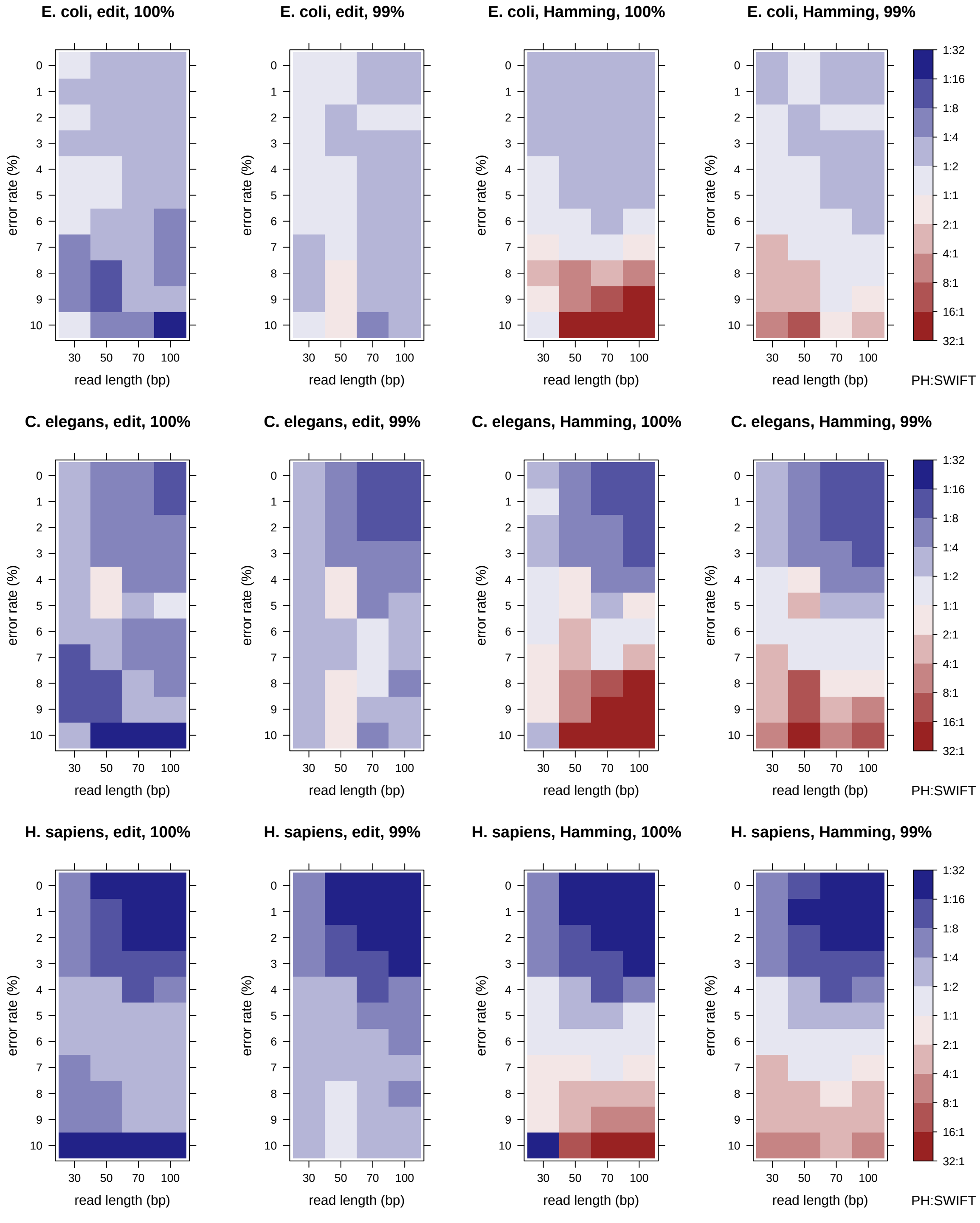


Figure S6: Experimental map for reference sequences of E. coli, C. elegans, and chr. 2 of human. For read sets from different organisms with compared the time between mapping with the pigeonhole and SWIFT filter, while varying read length, string metric, error rate, and sensitivity. Only the mapping time was measured to eliminate the variance of the I/O time on the cluster as much as possible. RazerS 3 was run with 12 threads. The color of each cell indicates the ratio of the running time between the pigeonhole and the SWIFT variant, where the pigeonhole variant was faster for blue cells. Ratios less/greater than 1:32/32:1 are plotted as 1:32/32:1. Because the C. elegans reference sequence is unmasked and contains large repeats, the pigeonhole filter generates a large number of matches for an error rate of 10 %. Subsequently, it uses too much memory with the default window size. We alleviated this by running RazerS 3 with a window size of 50 kbp on these datasets.

S11 EXTENDED VARIATION DETECTION TABLES

		(0,0)		(1,0)		(2,0)		(3,0)		(4,0)		(1,1)		(1,2)		(0,3)		(0,4)	
	method	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.
best-mappers	Bowtie 2	97.6	97.3	95.6	94.8	94.6	92.0	93.3	88.7	92.6	82.5	95.3	93.3	93.5	92.3	96.1	95.4	97.6	97.4
	BWA	98.2	97.9	97.1	96.4	97.6	95.3	96.5	90.2	94.9	85.1	97.4	90.9	97.1	80.3	96.3	66.5	97.5	67.1
	Soap 2	98.1	82.9	97.0	63.6	97.4	31.0	0.0	0.0	0.0	0.0	90.6	6.2	0.0	0.0	0.0	0.0	0.0	0.0
	R3-100	98.4	98.4	97.7	97.7	98.2	98.2	97.5	97.5	96.3	96.3	98.1	98.1	97.9	97.9	97.6	97.6	98.4	98.4
	R3-99	98.4	98.4	97.7	97.7	98.2	98.0	97.4	96.6	96.2	95.1	98.2	98.1	97.9	97.9	97.6	97.6	98.4	98.4
	R3-95	98.4	98.3	97.7	97.5	98.2	97.3	97.5	94.9	96.1	91.7	98.2	97.6	97.9	97.6	97.5	97.5	98.4	98.4
all-mappers	Hobbes	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	100.0	100.0	100.0	99.8	100.0	93.6	99.6	90.5	99.6	87.6
	mrFAST	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	99.3
	SHRiMP 2	100.0	99.4	100.0	99.5	100.0	99.7	100.0	99.9	100.0	99.7	100.0	99.5	100.0	99.2	100.0	99.6	100.0	99.6
	R3-100	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	R3-99	100.0	100.0	100.0	99.9	100.0	99.8	100.0	99.1	100.0	98.9	100.0	99.9	100.0	100.0	100.0	100.0	100.0	100.0
	R3-95	100.0	99.9	100.0	99.7	100.0	99.0	100.0	97.3	100.0	95.4	100.0	99.4	100.0	99.6	100.0	99.9	100.0	100.0

Table S3: Full results for the variation detection experiment using **single-end** reads. This table extends the upper right part of Table 1.

		(0,0)		(2,0)		(4,0)		(6,0)		(8,0)		(2,2)		(2,4)		(0,5)		(0,7)	
	method	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.	prec.	recl.
best-mappers	Bowtie 2	98.8	98.8	98.6	98.1	98.3	96.4	97.3	93.0	100.0	92.0	98.5	97.6	98.3	97.5	98.5	98.3	97.5	97.5
	BWA	99.0	98.4	99.0	96.7	99.2	93.5	99.3	86.6	100.0	80.0	98.5	79.9	100.0	65.2	99.1	69.0	100.0	60.0
	Soap 2	98.7	95.5	98.8	83.9	98.9	53.2	100.0	8.3	0.0	0.0	97.3	53.3	96.6	46.3	98.5	79.7	0.0	0.0
	R3-100	99.0	99.0	99.0	99.0	99.0	99.0	98.7	98.7	100.0	100.0	99.7	99.7	99.6	99.6	99.0	99.0	100.0	100.0
	R3-99	99.0	99.0	99.0	98.8	99.2	98.6	98.7	96.2	100.0	92.0	99.4	98.5	99.2	99.2	98.7	98.7	100.0	100.0
	R3-95	99.0	98.9	99.0	98.3	99.1	96.3	98.7	93.6	100.0	88.0	99.1	97.9	100.0	99.2	98.7	98.6	100.0	100.0
all-mappers	Hobbes	97.5	93.2	97.6	93.7	98.0	94.6	95.3	92.7	100.0	100.0	96.5	80.1	99.5	86.0	97.7	85.2	100.0	86.4
	mrFAST	98.8	98.8	98.9	98.9	98.9	98.9	98.7	98.7	100.0	100.0	99.1	99.1	98.8	98.8	91.8	7.8	96.3	65.0
	SHRiMP 2	100.0	99.7	100.0	99.7	100.0	99.9	100.0	100.0	100.0	100.0	100.0	99.7	100.0	99.6	100.0	99.7	100.0	100.0
	R3-100	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	R3-99	100.0	100.0	100.0	99.8	100.0	99.4	100.0	97.5	100.0	92.0	100.0	99.1	100.0	100.0	100.0	100.0	100.0	100.0
	R3-95	100.0	99.9	100.0	99.2	100.0	97.3	100.0	94.9	100.0	88.0	100.0	98.8	100.0	99.2	100.0	99.9	100.0	100.0

Table S4: Full results for the variation detection experiment using **paired-end** reads. This table extends the lower right part of Table 1.

S12 EXTENDED PERFORMANCE COMPARISON TABLES

	dataset	ERR022075 E. coli							SRR065390 C. elegans						
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]		mapped reads [%]		time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]		mapped reads [%]	
best-mappers	Bowtie 2	1:19	15:21	162	99.77	100.00 99.52 97.51 94.39 90.72	99.32	84.64 93.43 95.73 96.84 97.51	2:08	25:14	277	99.38	100.00 99.30 93.38 88.61 84.03	92.58	75.01 83.74 86.20 87.61 88.57
	BWA	3:04	13:17	184	99.78	100.00 99.51 97.52 94.41 91.83	97.98	84.64 93.43 95.72 96.85 97.52	5:21	36:50	325	99.46	100.00 99.09 95.57 89.70 85.86	89.33	75.01 83.72 86.25 87.64 88.59
	Soap 2	1:30	3:16	729	98.00	100.00 99.09 96.75 0.46 0.12	95.68	84.64 93.39 95.67 95.68 95.68	1:32	5:26	813	96.74	100.00 96.57 92.38 0.33 0.04	85.95	75.01 83.50 85.94 85.95 85.95
	R3-100	0:50	2:00	5705	100.00	100.00 100.00 100.00 100.00 100.00	97.59	84.64 93.47 95.79 96.92 97.59	1:42	11:42	5841	100.00	100.00 100.00 100.00 100.00 100.00	88.79	75.01 83.80 86.38 87.83 88.79
	R3-99	0:48	1:59	5705	100.00	100.00 100.00 100.00 99.99 99.94	97.59	84.64 93.47 95.79 96.92 97.59	1:38	11:17	5841	100.00	100.00 100.00 100.00 99.99 99.80	88.78	75.01 83.80 86.38 87.83 88.78
	R3-95	0:47	1:57	5705	100.00	100.00 100.00 100.00 99.87 99.56	97.59	84.64 93.47 95.79 96.92 97.59	1:30	9:52	5841	99.98	100.00 100.00 100.00 99.75 98.81	88.77	75.01 83.80 86.38 87.82 88.77
all-mappers	Hobbes	2:02	12:55	769	91.46	91.45 91.42 92.39 91.30 90.88	89.28	77.40 85.49 87.63 88.66 89.28	26:46	287:26	3136	93.27	93.21 93.86 93.69 93.42 90.88	82.83	69.92 78.18 80.60 81.95 82.83
	mrFAST	0:36	4:30	8269	100.00	100.00 100.00 100.00 100.00 99.99	97.59	84.64 93.47 95.79 96.92 97.59	6:01	57:42	10497	100.00	100.00 99.98 99.91 99.98 100.00	88.79	75.01 83.80 86.38 87.83 88.79
	SHRiMP 2	4:17	49:42	1553	99.86	99.99 99.69 98.66 97.11 94.49	99.28	84.63 93.44 95.72 96.83 97.48	52:44	608:58	3372	98.70	99.59 96.81 91.76 87.60 81.88	91.91	74.71 83.22 85.59 86.88 87.69
	R3-100	0:50	2:10	5705	100.00	100.00 100.00 100.00 100.00 100.00	97.59	84.64 93.47 95.79 96.92 97.59	2:20	16:57	9203	100.00	100.00 100.00 100.00 100.00 100.00	88.79	75.01 83.80 86.38 87.83 88.79
	R3-99	0:49	2:09	5705	100.00	100.00 100.00 100.00 99.99 99.94	97.59	84.64 93.47 95.79 96.92 97.59	2:16	16:07	9036	100.00	100.00 100.00 100.00 99.99 99.80	88.78	75.01 83.80 86.38 87.83 88.78
	R3-95	0:49	2:08	5705	100.00	100.00 100.00 100.00 99.87 99.56	97.59	84.64 93.47 95.79 96.92 97.59	2:11	14:24	8598	99.98	100.00 100.00 100.00 99.75 98.81	88.77	75.01 83.80 86.38 87.82 88.77

	dataset	SRR497711 D. melanogaster							ERR012100 H. sapiens						
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]		mapped reads [%]		time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]		mapped reads [%]	
best-mappers	Bowtie 2	2:00	23:43	299	99.65	100.00 99.77 99.02 97.65 94.92	85.71	52.08 67.27 73.62 76.88 78.81	5:37	66:40	3374	99.62	100.00 99.75 96.02 92.88 87.86	96.72	75.99 87.81 90.54 91.85 92.76
	BWA	5:35	36:30	413	98.96	100.00 99.57 98.40 90.72 82.08	79.37	52.08 67.24 73.57 76.62 78.31	13:45	98:30	4475	99.66	100.00 99.50 98.01 93.39 88.92	93.53	75.99 87.78 90.59 91.91 92.82
	Soap 2	1:55	8:37	900	91.78	100.00 96.24 89.35 0.09 0.02	72.49	52.08 66.73 72.48 72.49 72.49	2:34	15:11	5481	96.45	100.00 94.94 86.54 0.32 0.16	89.73	75.99 87.24 89.73 89.73 89.73
	R3-100	1:28	9:45	5795	100.00	100.00 100.00 100.00 100.00 100.00	78.92	52.08 67.31 73.69 76.97 78.92	85:56	1001:18	9096	100.00	100.00 100.00 100.00 100.00 100.00	92.99	75.99 87.84 90.67 92.02 92.99
	R3-99	1:26	9:14	5795	99.98	100.00 100.00 100.00 99.93 99.32	78.90	52.08 67.31 73.69 76.96 78.90	73:09	848:59	8679	99.99	100.00 100.00 100.00 99.94 99.41	92.98	75.99 87.84 90.67 92.02 92.98
	R3-95	1:26	9:24	5795	99.87	100.00 100.00 100.00 99.11 96.34	78.82	52.08 67.31 73.69 76.94 78.82	43:16	493:42	7640	99.96	100.00 100.00 100.00 99.30 96.92	92.95	75.99 87.84 90.67 92.01 92.95
all-mappers	Hobbes	4:51	39:57	2141	96.49	96.55 96.46 96.94 96.28 93.86	76.16	50.28 64.98 71.16 74.33 76.16	265:48	2851:00	70683	95.97	95.94 96.14 96.39 96.10 94.63	89.24	72.90 84.30 87.02 88.33 89.24
	mrFAST	4:01	37:32	10844	100.00	100.00 100.00 100.00 100.00 100.00	78.92	52.08 67.31 73.69 76.97 78.92	413:40	3987:49	11324	100.00	100.00 100.00 100.00 100.00 100.00	92.99	75.99 87.84 90.67 92.02 92.99
	SHRiMP 2	23:40	255:09	3801	99.83	99.99 99.99 99.74 98.71 96.33	89.91	52.07 67.30 73.66 76.92 78.83	1312:09	14466:34	38188	99.81	99.89 99.83 99.39 98.29 96.81	99.06	75.90 87.74 90.56 91.91 92.87
	R3-100	1:51	11:59	7329	100.00	100.00 100.00 100.00 100.00 100.00	78.92	52.08 67.31 73.69 76.97 78.92	118:26	1384:18	15298	100.00	100.00 100.00 100.00 100.00 100.00	92.99	75.99 87.84 90.67 92.02 92.99
	R3-99	1:49	11:21	7302	99.98	100.00 100.00 100.00 99.93 99.32	78.90	52.08 67.31 73.69 76.96 78.90	100:19	1169:22	15238	99.99	100.00 100.00 100.00 99.94 99.41	92.98	75.99 87.84 90.67 92.02 92.98
	R3-95	1:45	10:40	7270	99.87	100.00 100.00 100.00 99.11 96.34	78.82	52.08 67.31 73.69 76.94 78.82	58:13	665:47	14384	99.96	100.00 100.00 100.00 99.30 96.92	92.95	75.99 87.84 90.67 92.01 92.95

Table S5: Extended experimental results for real-world **single-end** data, extending Table 2. The results are shown for the first 10 M × 100 bp single-end reads of Illumina datasets. In large we show the percentage of totally mapped reads and in small the percentages of read that are mapped with up to $\left(\begin{smallmatrix} 0 & 1\% & 2\% \\ 3\% & 4\% & \end{smallmatrix}\right)$ errors.

	dataset	simulated, m = 200 D. melanogaster					simulated, m = 400 D. melanogaster					simulated, m = 800 D. melanogaster				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]
best-mappers	Bowtie 2	0:29	5:23	318	99.46 100.00 99.86 99.47 97.23 88.00	99.70 12.51 61.89 86.40 91.42 92.75	1:35	18:15	355	98.68 100.00 99.80 99.12 91.93 83.01	99.94 1.56 52.43 82.17 86.62 89.21	13:48	164:45	628	96.73 97.47 99.67 98.05 87.95 85.17	99.99 0.03 41.07 73.95 82.31 90.03
	BWA	1:09	6:41	440	98.49 100.00 99.82 98.45 90.30 67.15	91.83 12.51 61.87 86.11 90.78 91.82	2:40	18:03	935	93.95 100.00 99.59 96.69 63.29 4.45	84.57 1.56 52.31 81.30 84.38 84.57	5:38	36:44	1055	74.96 97.47 98.62 82.47 0.00 0.00	68.09 0.03 40.61 68.09 68.09 68.09
	Soap 2	0:21	1:40	1074	65.88 100.00 96.41 4.12 0.008 0.00	61.20 12.51 60.18 61.20 61.20 61.20	0:38	3:40	1554	56.07 100.00 79.04 27.73 2.09 0.22	50.99 1.56 41.87 50.37 50.69 50.92	0:54	4:33	1546	41.21 97.47 67.99 28.10 0.22 0.00	38.14 0.03 28.17 37.88 38.14 38.14
	R3-100	0:46	6:32	1529	100.00 100.00 100.00 100.00 100.00 100.00	92.87 12.51 61.94 86.47 91.50 92.87	0:56	7:05	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:17	8:10	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43
	R3-99	0:46	6:59	1529	100.00 100.00 100.00 100.00 99.98 99.88	92.87 12.51 61.94 86.47 91.50 92.87	0:56	6:46	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:17	7:52	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43
	R3-95	0:46	6:34	1529	99.99 100.00 100.00 100.00 99.93 99.63	92.86 12.51 61.94 86.47 91.50 92.86	0:55	6:51	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:15	7:52	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43
all-mappers	mrFAST	1:11	10:29	6909	99.24 100.00 100.00 100.00 96.34 62.09	92.17 12.51 61.94 86.47 91.32 92.17	2:17	20:54	7972	94.15 100.00 100.00 98.39 54.15 0.00	85.02 1.56 52.48 81.79 85.02 85.02	5:16	49:22	9115	65.25 93.14 95.65 59.59 0.00 0.00	69.32 0.03 39.34 69.32 69.32 69.32
	SHRiMP 2	7:25	64:30	3758	99.49 99.98 99.97 99.78 96.97 81.86	99.96 12.51 61.93 86.41 91.36 92.60	42:53	484:02	3814	98.44 99.95 99.91 99.21 88.64 77.47	99.85 1.55 52.44 82.09 86.39 88.83	796:06	9442:21	4023	95.70 97.47 99.75 97.60 82.95 80.14	99.31 0.03 41.04 73.67 81.63 89.14
	R3-100	0:49	6:55	1529	100.00 100.00 100.00 100.00 100.00 100.00	92.87 12.51 61.94 86.47 91.50 92.87	0:58	6:57	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:20	8:48	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43
	R3-99	0:48	6:19	1529	100.00 100.00 100.00 100.00 99.98 99.88	92.87 12.51 61.94 86.47 91.50 92.87	0:58	7:24	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:20	8:38	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43
	R3-95	0:48	6:43	1529	99.99 100.00 100.00 100.00 99.93 99.63	92.86 12.51 61.94 86.47 91.50 92.86	0:55	6:42	2674	100.00 100.00 100.00 100.00 100.00 100.00	89.47 1.56 52.48 82.26 86.79 89.47	1:20	8:35	4963	100.00 100.00 100.00 100.00 100.00 100.00	90.43 0.03 41.13 74.13 82.65 90.43

	dataset	simulated, m = 200 H. sapiens					simulated, m = 400 H. sapiens					simulated, m = 800 H. sapiens				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped reads [%]	mapped reads [%]
best-mappers	Bowtie 2	1:19	15:13	3372	99.02 100.00 99.58 98.62 96.32 87.23	99.57 12.57 61.74 86.07 91.17 92.54	4:25	52:06	3413	98.56 100.00 99.70 98.96 91.74 83.08	99.94 1.59 52.46 82.11 86.61 89.24	63:15	756:12	3686	96.83 96.57 99.73 98.16 88.25 85.29	100.00 0.02 41.20 73.97 82.36 90.11
	BWA	4:53	44:30	6619	98.44 100.00 99.81 98.36 90.31 66.51	91.82 12.57 61.85 86.05 90.78 91.81	20:21	185:24	11043	93.64 100.00 99.53 96.06 62.45 4.39	84.33 1.59 52.36 81.10 84.15 84.33	26:00	257:32	10946	74.38 96.57 97.72 82.00 0.00 0.00	67.60 0.02 40.37 67.60 67.60 67.60
	Soap 2	0:49	5:08	5651	65.82 100.00 96.41 3.99 0.004 0.00	61.15 12.57 60.17 61.15 61.15 61.15	1:37	12:30	5938	56.22 100.00 79.19 27.86 2.00 0.17	51.09 1.59 42.00 50.48 50.79 51.02	2:05	15:04	6122	41.51 96.57 68.43 28.24 0.27 0.00	38.22 0.02 28.33 37.95 38.22 38.22
	R3-100	26:34	299:17	3341	100.00 100.00 100.00 100.00 100.00 100.00	92.88 12.57 61.93 86.43 91.51 92.88	44:21	511:05	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	102:48	1205:31	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49
	R3-99	16:17	179:00	3341	100.00 100.00 100.00 100.00 99.98 99.85	92.88 12.57 61.93 86.43 91.51 92.88	20:31	226:47	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	38:50	442:12	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49
	R3-95	13:55	153:13	3341	99.99 100.00 100.00 100.00 99.94 99.63	92.87 12.57 61.93 86.43 91.51 92.87	20:31	227:05	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	37:21	424:50	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49
all-mappers	mrFAST	103:02	1024:12	7395	99.25 100.00 100.00 100.00 96.47 61.90	92.18 12.57 61.93 86.43 91.34 92.18	321:50	4143:26	7926	65.33 69.17 69.39 68.36 37.33 0.00	59.38 1.10 36.49 56.87 59.38 59.38	1116:03	13750:58	9601	44.19 62.66 64.69 40.39 0.00 0.00	47.24 0.01 26.70 47.24 47.24 47.24
	SHRiMP 2	546:29	5323:53	38570	99.32 99.78 99.79 99.60 96.96 82.05	99.95 12.54 61.80 86.23 91.23 92.48	1705:04	92111:02	45100	98.10 99.47 99.56 98.86 88.49 77.61	99.75 1.58 52.34 81.82 86.14 88.60	–	–	–	–	–
	R3-100	28:30	323:07	3341	100.00 100.00 100.00 100.00 100.00 100.00	92.88 12.57 61.93 86.43 91.51 92.88	48:52	564:57	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	105:48	1243:35	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49
	R3-99	17:22	191:32	3341	100.00 100.00 100.00 100.00 99.98 99.85	92.88 12.57 61.93 86.43 91.51 92.88	21:46	241:16	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	41:41	477:03	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49
	R3-95	14:40	160:48	3341	99.99 100.00 100.00 100.00 99.94 99.63	92.87 12.57 61.93 86.43 91.51 92.87	21:13	236:06	3532	100.00 100.00 100.00 100.00 100.00 100.00	89.53 1.59 52.57 82.28 86.83 89.53	39:52	455:24	4964	100.00 100.00 100.00 100.00 100.00 100.00	90.49 0.02 41.27 74.17 82.70 90.49

Table S5 (continued): Extended experimental results for **long simulated single-end** data, extending Table 2. The results are shown for 1 M single-end reads of the given lengths simulated with Mason using the default Illumina error model. In large we show the percentage of totally mapped reads and in small the percentages of read that are mapped with up to $\left(\begin{smallmatrix}0 & 1\% & 2\% \\ 3\% & 4\% & \end{smallmatrix}\right)$ errors. Hobbes is not capable of mapping long reads and thus not shown here. Some mrFAST processes crashed for the D. melanogaster and H. sapiens m = 400 and m = 800 datasets which explains the low number of mapped reads. SHRiMP 2 was not able to map the 800 bp human dataset within 96 hours.

	dataset	ERR022075 E. coli					SRR065390 C. elegans				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]
best-mappers	Bowtie 2	4:09	49:15	195	99.69 100.00 99.94 99.51 97.97 95.89	98.69 18.92 78.51 88.74 91.84 93.46	6:19	74:50	322	97.30 99.98 99.35 96.91 93.85 87.33	84.59 13.20 16.42 17.74 18.64 19.33
	BWA	7:35	30:58	343	99.66 100.00 99.94 99.48 97.54 95.39	95.87 18.92 78.51 88.74 91.82 93.44	12:32	75:47	479	92.01 99.97 95.50 82.94 67.12 56.05	81.20 13.56 16.88 18.21 19.00 19.56
	Soap 2	3:51	17:29	743	96.58 100.00 99.81 98.87 68.19 38.61	94.70 18.93 78.48 89.09 91.58 92.37	5:56	42:02	833	89.51 100.00 94.23 88.19 59.50 29.87	24.48 13.25 16.34 17.70 18.34 18.73
	R3-100	1:53	7:33	11113	100.00 100.00 100.00 100.00 100.00 100.00	94.66 18.92 78.55 88.79 91.91 93.54	6:29	64:44	11230	100.00 100.00 100.00 100.00 100.00 100.00	20.38 13.20 16.43 17.78 18.70 19.43
	R3-99	1:45	6:08	11113	99.99 100.00 100.00 100.00 99.98 99.79	94.66 18.92 78.55 88.79 91.91 93.54	6:20	62:39	11230	99.97 100.00 100.00 99.99 99.98 99.85	20.38 13.20 16.43 17.78 18.70 19.43
	R3-95	1:44	6:22	11113	99.96 100.00 100.00 100.00 99.84 98.96	94.62 18.92 78.55 88.79 91.90 93.52	6:07	60:22	11230	99.81 100.00 100.00 99.99 99.83 99.21	20.35 13.20 16.43 17.78 18.70 19.43
all-mappers	Hobbes	5:21	39:44	1196	90.57 90.81 90.62 90.19 89.95 90.21	86.15 17.18 71.25 80.61 83.50 85.05	10:27	103:38	2852	86.49 92.91 85.94 72.90 65.00 62.82	18.66 12.26 15.09 16.14 16.82 17.41
	mrFAST	1:21	10:22	52868	99.99 99.99 99.99 99.99 99.99 99.99	95.31 18.92 78.55 88.78 91.90 93.54	14:02	134:13	54510	99.29 99.75 99.48 98.82 98.22 97.68	79.37 13.18 16.41 17.75 18.66 19.40
	SHRiMP 2	8:18	97:16	1612	99.85 100.00 99.99 99.80 99.06 97.80	97.74 18.92 78.54 88.76 91.85 93.46	105:30	1238:32	3468	94.81 99.77 96.89 88.75 80.41 73.48	53.92 13.32 16.58 17.88 18.72 19.36
	R3-100	1:53	7:34	11113	100.00 100.00 100.00 100.00 100.00 100.00	94.66 18.92 78.55 88.79 91.91 93.54	7:49	78:35	11816	100.00 100.00 100.00 100.00 100.00 100.00	20.38 13.20 16.43 17.78 18.70 19.43
	R3-99	1:48	6:40	11113	99.99 100.00 100.00 100.00 99.98 99.79	94.66 18.92 78.55 88.79 91.91 93.54	6:37	64:39	11588	99.97 100.00 100.00 99.99 99.98 99.85	20.38 13.20 16.43 17.78 18.70 19.43
	R3-95	1:45	6:13	11113	99.96 100.00 100.00 100.00 99.84 98.96	94.62 18.92 78.55 88.79 91.90 93.52	6:33	63:35	11230	99.81 100.00 100.00 99.99 99.83 99.21	20.35 13.20 16.43 17.78 18.70 19.43

	dataset	SRR497711 D. melanogaster					ERR012100 H. sapiens				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]
best-mappers	Bowtie 2	6:32	77:40	349	98.94 100.00 99.03 98.43 97.52 96.11	81.94 32.50 50.89 60.48 66.19 69.88	10:51	128:58	3555	99.51 99.97 99.90 99.46 98.34 96.40	94.19 15.04 62.97 77.57 82.66 85.16
	BWA	13:33	77:14	503	97.47 100.00 98.85 97.77 93.52 87.15	73.41 32.51 50.87 60.41 65.93 69.30	34:35	241:41	4662	98.84 99.99 99.88 98.95 95.16 90.79	88.06 15.04 62.97 77.50 82.47 84.86
	Soap 2	5:29	38:22	940	88.67 100.00 94.93 89.40 70.79 41.11	72.77 32.58 50.33 59.65 64.13 65.93	8:24	61:32	5463	91.58 99.99 98.89 93.72 51.38 26.08	87.47 15.07 62.57 77.33 80.49 81.46
	R3-100	9:01	93:39	11199	100.00 100.00 100.00 100.00 100.00 100.00	72.95 32.50 51.07 60.63 66.35 70.04	176:29	2077:35	18568	100.00 100.00 100.00 100.00 100.00 100.00	86.93 15.04 63.02 77.65 82.76 85.27
	R3-99	7:00	69:26	11199	99.97 100.00 100.00 100.00 99.97 99.80	72.93 32.50 51.07 60.63 66.34 70.04	159:03	1872:33	16568	99.98 100.00 100.00 100.00 99.98 99.75	86.91 15.04 63.02 77.65 82.76 85.26
	R3-95	6:56	68:48	11199	99.78 100.00 100.00 100.00 99.68 98.65	72.80 32.50 51.07 60.63 66.33 69.98	135:44	1599:10	13678	99.89 100.00 100.00 100.00 99.81 98.78	86.84 15.04 63.02 77.65 82.75 85.23
all-mappers	Hobbes	8:43	75:57	5870	84.78 84.27 85.76 86.51 85.66 83.24	62.48 27.39 43.41 51.81 56.81 59.99	89:35	884:05	47270	95.11 95.68 95.83 94.73 92.92 90.73	84.05 14.39 60.42 74.46 79.44 81.95
	mrFAST	8:26	81:25	48032	100.00 100.00 100.00 99.99 99.99 99.99	73.16 32.50 51.07 60.63 66.35 70.04	779:12	7649:19	57625	99.94 99.98 99.97 99.93 99.85 99.76	87.79 15.04 63.01 77.64 82.75 85.26
	SHRiMP 2	47:07	537:29	3838	99.67 100.00 99.99 99.84 99.21 97.77	87.36 32.50 51.07 60.62 66.30 69.95	2762:32	31710:26	38594	99.74 99.91 99.90 99.81 99.40 98.40	97.51 15.03 62.96 77.57 82.66 85.15
	R3-100	7:59	76:28	13558	100.00 100.00 100.00 100.00 100.00 100.00	72.95 32.50 51.07 60.63 66.35 70.04	184:27	2167:14	27623	100.00 100.00 100.00 100.00 100.00 100.00	86.93 15.04 63.02 77.65 82.76 85.27
	R3-99	7:44	73:54	13485	99.97 100.00 100.00 100.00 99.97 99.80	72.93 32.50 51.07 60.63 66.34 70.04	177:56	2100:43	25866	99.98 100.00 100.00 100.00 99.98 99.75	86.91 15.04 63.02 77.65 82.76 85.26
	R3-95	7:36	72:22	13241	99.78 100.00 100.00 100.00 99.68 98.65	72.80 32.50 51.07 60.63 66.33 69.98	166:22	1956:20	23026	99.89 100.00 100.00 100.00 99.81 98.78	86.84 15.04 63.02 77.65 82.75 85.23

Table S6: Extended experimental results for real-world **paired-end** data, extending Table 3. The results are shown for the first 10 M × 2 × 100 bp pairs of Illumina datasets. In large we show the percentage of totally mapped reads and in small the percentages of pairs that are mapped with up to $\left(\begin{smallmatrix} 0 & 1\% & 2\% \\ 3\% & 4\% & \end{smallmatrix}\right)$ errors.

dataset		simulated, m = 200 D. melanogaster					simulated, m = 400 D. melanogaster					simulated, m = 800 D. melanogaster				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]
best-mappers	Bowtie 2	1:25	16:37	390	98.91 100.00 99.87 99.59 98.98 96.98	99.24 1.35 17.49 45.66 64.52 71.86	5:45	68:17	515	97.37 100.00 99.79 99.45 98.32 92.65	99.62 0.02 6.16 35.88 57.96 65.03	39:07	468:30	1418	93.64 – 99.61 99.18 96.68 88.10	99.70 0.00 1.00 24.15 47.10 57.94
	BWA	2:13	12:41	674	96.98 100.00 99.82 99.20 97.04 90.51	84.30 1.35 17.49 45.56 64.04 70.88	6:13	38:07	975	88.31 100.00 99.64 98.50 93.79 65.44	71.56 0.02 6.16 35.59 56.62 61.60	11:26	73:40	1569	56.28 – 99.23 95.71 70.82 4.22	46.44 0.00 0.99 23.32 39.95 40.44
	Soap 2	1:55	16:35	1092	74.85 100.00 95.74 87.54 66.82 24.65	71.96 1.37 17.17 42.92 56.01 57.95	4:40	44:16	1377	50.03 100.00 91.25 70.98 34.00 7.57	49.55 0.02 5.71 27.43 35.38 36.19	12:36	124:23	1561	23.55 – 78.27 48.35 20.11 0.94	28.23 0.00 0.80 12.38 17.36 17.64
	R3-100	1:26	12:21	3075	100.00 100.00 100.00 100.00 100.00 100.00	74.91 1.35 17.51 45.74 64.64 72.00	1:45	13:44	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:22	16:05	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.38
	R3-99	1:25	12:44	3075	99.99 100.00 100.00 100.00 100.00 99.98	74.91 1.35 17.51 45.74 64.64 72.00	1:48	14:30	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:19	15:24	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.37
	R3-95	1:23	12:42	3075	99.98 100.00 100.00 100.00 99.99 99.91	74.89 1.35 17.51 45.74 64.64 71.99	1:41	13:31	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:19	15:27	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.37
all-mappers	mrFAST	2:16	20:37	11104	98.46 99.97 99.97 99.97 99.63 95.11	84.68 1.35 17.51 45.73 64.57 71.57	4:51	45:52	12493	88.68 100.00 99.97 99.79 95.80 60.61	72.13 0.02 6.17 35.90 57.24 62.37	10:47	102:53	15392	44.19 – 95.01 91.19 40.34 0.00	49.69 0.00 0.95 24.50 43.35 43.35
	SHRiMP 2	13:10	133:22	3746	98.99 99.99 99.96 99.90 99.43 96.68	99.97 1.35 17.51 45.73 64.56 71.79	83:57	979:35	3945	96.87 100.00 99.88 99.68 98.34 90.03	99.83 0.02 6.17 35.89 57.86 64.72	1617:26	19264:14	4211	91.64 – 99.76 99.33 95.80 83.47	98.62 0.00 1.00 24.12 46.79 57.14
	R3-100	1:33	14:02	3075	100.00 100.00 100.00 100.00 100.00 100.00	74.91 1.35 17.51 45.74 64.64 72.00	1:54	15:36	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:30	15:55	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.38
	R3-99	1:30	13:23	3075	99.99 100.00 100.00 100.00 100.00 99.98	74.91 1.35 17.51 45.74 64.64 72.00	1:53	14:53	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:30	15:45	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.37
	R3-95	1:26	12:32	3075	99.98 100.00 100.00 100.00 99.99 99.91	74.89 1.35 17.51 45.74 64.64 71.99	1:51	15:15	5364	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.17 35.96 58.11 65.29	2:29	15:47	9942	100.00 – 100.00 100.00 100.00 100.00	71.16 0.00 1.00 24.22 47.33 58.37

dataset		simulated, m = 200 H. sapiens					simulated, m = 400 H. sapiens					simulated, m = 800 H. sapiens				
	method	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]	time [min:s]	cpu time [min:s]	memory [Mb]	correctly mapped pairs [%]	mapped pairs [%]
best-mappers	Bowtie 2	2:05	24:14	3439	98.75 99.99 99.86 99.54 98.72 96.62	99.29 1.40 17.45 45.61 64.42 71.75	7:43	90:09	3544	97.34 100.00 99.80 99.43 98.31 92.58	99.81 0.02 6.17 35.96 57.87 64.94	71:12	850:50	4380	93.84 – 99.81 99.28 96.87 88.23	99.90 0.00 0.96 24.23 47.11 57.94
	BWA	10:39	90:33	6790	96.88 100.00 99.84 99.16 96.97 90.29	84.24 1.40 17.46 45.52 64.00 70.86	38:15	378:52	11137	87.72 100.00 99.63 98.25 92.88 64.13	71.10 0.02 6.16 35.61 56.30 61.19	62:08	568:37	10955	55.46 – 98.70 94.13 69.75 4.11	45.78 0.00 0.95 23.00 39.33 39.81
	Soap 2	2:05	16:55	5667	74.84 100.00 95.93 87.57 66.83 24.57	71.89 1.41 17.14 42.86 55.95 57.90	4:46	43:57	6385	50.27 100.00 91.57 71.35 34.06 7.46	49.64 0.02 5.72 27.61 35.48 36.27	11:16	111:05	7385	23.83 – 79.51 49.05 20.12 0.91	28.27 0.00 0.78 12.53 17.48 17.76
	R3-100	28:39	327:39	3759	100.00 100.00 100.00 100.00 100.00 100.00	74.91 1.40 17.47 45.69 64.59 71.97	34:07	390:25	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.26	47:09	539:22	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39
	R3-99	25:03	285:08	3759	99.99 100.00 100.00 100.00 100.00 99.98	74.91 1.40 17.47 45.69 64.59 71.97	27:30	312:19	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.25	38:57	443:16	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39
	R3-95	24:14	275:23	3759	99.98 100.00 100.00 100.00 99.99 99.92	74.90 1.40 17.47 45.69 64.59 71.97	28:03	319:08	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.25	32:30	366:04	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39
all-mappers	mrFAST	217:29	2160:07	14319	98.44 99.96 99.97 99.96 99.64 95.05	84.77 1.40 17.47 45.69 64.52 71.55	328:45	1991:56	13630	22.23 22.71 25.04 25.03 24.08 15.06	18.10 0.005 1.55 9.02 14.35 15.63	1043:59	9427:14	18019	21.01 – 43.36 43.28 19.22 0.00	23.71 0.00 0.42 11.66 20.63 20.63
	SHRiMP 2	996:11	10678:56	39094	98.82 99.82 99.83 99.75 99.27 96.62	99.98 1.40 17.46 45.64 64.45 71.70	3243:49	187639:34	48973	96.64 100.00 99.60 99.43 98.08 90.01	99.86 0.02 6.16 35.90 57.66 64.52	–	–	–	–	–
	R3-100	29:59	343:42	3759	100.00 100.00 100.00 100.00 100.00 100.00	74.91 1.40 17.47 45.69 64.59 71.97	33:11	378:59	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.26	43:27	495:53	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39
	R3-99	26:31	302:45	3759	99.99 100.00 100.00 100.00 100.00 99.98	74.91 1.40 17.47 45.69 64.59 71.97	26:40	299:26	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.25	32:34	365:36	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39
	R3-95	25:21	288:57	3759	99.98 100.00 100.00 100.00 99.99 99.92	74.90 1.40 17.47 45.69 64.59 71.97	26:34	300:32	5366	100.00 100.00 100.00 100.00 100.00 100.00	69.68 0.02 6.18 36.05 58.06 65.25	32:27	366:30	9944	100.00 – 100.00 100.00 100.00 100.00	71.11 0.00 0.96 24.29 47.33 58.39

Table S6 (continued): Extended experimental results for **long simulated paired-end** data, extending Table 3. The results are shown for 1 M paired-end reads of the given lengths simulated with Mason using the default Illumina error model. In large we show the percentage of totally mapped reads and in small the percentages of pairs that are mapped with up to $\left(\begin{smallmatrix} 0 & 1\% & 2\% \\ 3\% & 4\% & \end{smallmatrix}\right)$ errors. Some mrFAST processes crashed for the D. melanogaster $m = 400$ and H. sapiens $m = 400$ and $m = 800$ datasets which explains the low number of mapped reads. SHRiMP 2 was not able to map the 800 bp human dataset within 96 hours.

REFERENCES

Ahmadi, A., Behm, A., Honnalli, N., Li, C., Weng, L., and Xie, X. (2011). Hobbes: optimized gram-based methods for efficient read alignment. *Nucleic Acids Res.*

Holtgrewe, M. (2010). Mason – a read simulator for second generation sequencing data. Technical Report TR-B-10-06, Institut für Mathematik und Informatik, Freie Universität Berlin.