BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 20 2011, pages 2790–2796
doi:10.1093/bioinformatics/btr477

Sequence analysis

Advance Access publication August 19, 2011

Comparative analysis of algorithms for next-generation sequencing read alignment

Matthew Ruffalo[1,*], Thomas LaFramboise[2,3] and Mehmet Koyutürk[1,3]

[1]Department of Electrical Engineering and Computer Science, [2]Department of Genetics and [3]Center for Proteomics and Bioinformatics, Case Western Reserve University, Cleveland, OH 44106, USA

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: The advent of next-generation sequencing (NGS) techniques presents many novel opportunities for many applications in life sciences. The vast number of short reads produced by these techniques, however, pose significant computational challenges. The first step in many types of genomic analysis is the mapping of short reads to a reference genome, and several groups have developed dedicated algorithms and software packages to perform this function. As the developers of these packages optimize their algorithms with respect to various considerations, the relative merits of different software packages remain unclear. However, for scientists who generate and use NGS data for their specific research projects, an important consideration is choosing the software that is most suitable for their application.

Results: With a view to comparing existing short read alignment software, we develop a simulation and evaluation suite, SEAL, which simulates NGS runs for different configurations of various factors, including sequencing error, indels and coverage. We also develop criteria to compare the performances of software with disparate output structure (e.g.some packages return a single alignment while some return multiple possible alignments). Using these criteria, we comprehensively evaluate the performances of Bowtie, BWA, mr- and mrsFAST, Novoalign, SHRiMP and SOAPv2, with regard to accuracy and runtime.

Conclusion: We expect that the results presented here will be useful to investigators in choosing the alignment software that is most suitable for their specific research aims. Our results also provide insights into the factors that should be considered to use alignment results effectively. SEAL can also be used to evaluate the performance of algorithms that use deep sequencing data for various purposes (e.g.identification of genomic variants).

Availability: SEAL is available as open source at http://compbio.case.edu/seal/.

Contact: matthew.ruffalo@case.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on March 25, 2011; revised on July 20, 2011; accepted on August 10, 2011

*To whom correspondence should be addressed.

1 INTRODUCTION

Next-generation sequencing techniques are demonstrating promise in transforming research in life sciences (Schuster, 2007). These techniques support many applications including metagenomics (Qin *et al.*, 2010), detection of SNPs (Van Tassell *et al.*, 2008) and genomic structural variants (Alkan *et al.*, 2009; Medvedev *et al.*, 2009) in a population, DNA methylation studies (Taylor *et al.*, 2007), analysis of mRNA expression (Sultan *et al.*, 2008), cancer genomics (Guffanti *et al.*, 2009) and personalized medicine (Auffray *et al.*, 2009). Some applications (e.g. metagenomics) require *de novo* sequencing of a sample (Miller *et al.*, 2010), while many others (e.g. variant detection, cancer genomics) require resequencing. For all of these applications, the vast amount of data produced by sequencing runs poses many computational challenges (Horner *et al.*, 2010).

In resequencing, a reference genome is already available for the species (e.g. the human genome) and one is interested in comparing short reads obtained from the genome of one or more donors (individual members of the species) to the reference genome. Therefore, the first step in any kind of analysis is the mapping of short reads to a reference genome. This task is complicated by many factors, including genetic variation in the population, sequencing error, short read length and the huge volume of short reads to be mapped. So far, many algorithms have been developed to overcome these challenges and these algorithms have been made available to the scientific community as software packages (Li and Homer, 2010). Currently available software packages for short read alignment include Bowtie (Langmead *et al.*, 2009), SOAP (Li *et al.*, 2009), BWA (Li and Durbin, 2009, 2010), mrFAST (Alkan *et al.*, 2009), mrsFAST (Hach *et al.*, 2010), Novoalign (Novocraft, 2010) and SHRiMP (Rumble *et al.*, 2009).

In this article, we assess the performance of currently available alignment algorithms, with a view to (i) understanding the effect of various factors on accuracy and runtime performance and (ii) comparing existing algorithms in terms of their performance in various settings. For this purpose, we develop a simulation and evaluation suite, SEAL, that simulates short read sequencing runs for a given set of configurations and evaluates the output of each software using novel performance criteria that are specifically designed for the current application. Our results show significant differences in performance and accuracy as quality of the reads and the characteristics of the genome vary. In the next section, we briefly describe the alignment algorithms that are evaluated in this article. Subsequently, in Section 3, we describe the simulation suite implemented in SEAL and our performance criteria in detail.

We present detailed experimental results in Section 4. We conclude with a detailed discussion of our results in Section 5.

2 BACKGROUND

The problem of short read alignment is formulated as follows. Given a reference genome (for which the entire nucleotide sequence is available) and a donor genome (for which the nucleotide sequence is not known), a sequencing run produces many short reads from the donor genome. These short reads are generated by taking relatively long (200–8000 bp) fragments from the donor genome and sequencing a number of bases (35–150 bp for Illumina, 400 bp on average for 454) from either only one end (*single-end read*) or both ends (*paired-end read*) of the fragment. Given this set of short reads from the donor genome, the objective of alignment is to correctly determine each read's corresponding location in the reference genome.

Here, we briefly describe the alignment algorithms and the major differences between their approaches. Many of these algorithms have undergone major revisions in recent years, with their authors producing either 'version 2' of their tools [e.g. SOAP (Li *et al.*, 2009)] or using a different name for the new version [e.g. Mapping and Assembly with Qualities (MAQ) (Li *et al.*, 2008a) versus Burrows-Wheeler Alignment (BWA) (Li and Durbin, 2009)]. We only consider the most recent version of each tool for brevity; the developers of these tools perform their own evaluation to demonstrate the superiority of their newer approaches.

2.1 Bowtie

Bowtie (Langmead *et al.*, 2009) uses an index built with the Burrows-Wheeler transformation (Burrows *et al.*, 1994; Ferragina and Manzini, 2000) and claims a small memory footprint—about 1.3 GB for the entire human genome. Bowtie makes some compromises to provide its speed and memory usage; notably that it does not guarantee the highest quality read mapping if no exact match exists. Additionally, it may fail to align some reads with valid mappings when configured for maximum speed. If a user desires higher accuracy, Bowtie provides options to adjust this trade-off.

2.2 BWA

BWA (Li and Durbin, 2009, 2010) can be considered as 'MAQ (Li *et al.*, 2008a) version 2'. Whereas MAQ uses a hash-based index to search the genome, BWA uses an index built with the Burrows-Wheeler transformation that allows for much faster searching than its predecessor. Like its predecessor, BWA reports a meaningful quality score for the mapping that can be used to discard mappings that are not well supported due to e.g. a high number of mismatches.

2.3 mrFAST and mrsFAST

The mr- and mrsFAST tools (Alkan *et al.*, 2009; Hach *et al.*, 2010) are notable in that they report all mappings of a read to a genome rather than a single 'best' mapping. The ability to report all possible reference genome locations is useful in the detection of copy number variants (Bailey *et al.*, 2002). Indeed, these algorithms are developed primarily for applications that involve detection of structural variants (Alkan *et al.*, 2009). mr- and mrsFAST use a seed-and-extend method for alignment, and create hash table indices for the reference genome. Each read is split into first, middle and last k-mers (the default $k = 12$), and each of these k-mers are searched in the hash index to place initial alignment seeds.

2.4 Novoalign

Novoalign is a proprietary product of Novocraft (Novocraft, 2010) that uses a hashing strategy similar to that of MAQ (Li *et al.*, 2008a). It has become quite popular in recent publications due to its accuracy claims, and it allows up to eight mismatches per read for single end mapping.

2.5 SHRiMP

SHRiMP (Rumble *et al.*, 2009) specializes in mapping SOLiD color-space reads, but is also usable for the reads simulated in our evaluation. It takes advantage of recent advances in sequence alignment: q-gram filters (Rasmussen *et al.*, 2005), which allow multiple matching seeds to start the alignment process; spaced seeds (Califano and Rigoutsos, 1993), which allow predetermined sections of mismatches in seed sequences; and specialized hardware implementations/instructions to speed up the standard Smith–Waterman (Smith and Waterman, 1981) alignment algorithm.

2.6 SOAPv2

SOAP is an alignment algorithm specifically designed for detecting and genotyping single nucleotide polymorphisms. Like BWA and its predecessor MAQ, SOAP version 2 (Li *et al.*, 2009) improves on SOAPv1 (Li *et al.*, 2008b) by using an index based on the Burrows-Wheeler transformation (BWT). This improved index significantly improves alignment speed and memory usage. SOAPv2 determines matches by building a hash table to accelerate the searching of the BWT reference index.

3 METHODS

3.1 Simulation

We develop SEAL (SEquence ALignment evaluation suite), a comprehensive sequencing simulation and alignment tool evaluation suite. This software (implemented in Java) provides several utilities that can be used to evaluate alignment algorithms, including:

- Reading a pre-existing reference genome from one or more FASTA files.
- Alternatively, generating an artificial reference genome based on input parameters (length, repeat count, repeat length, repeat variability rate).
- Simulating reads from random locations in the genome based on input parameters of read length, coverage, sequencing error rate and indel rate.
- Applying alignment tools to the genome and the reads through a standardized interface.
- Parsing the output of the alignment tool and calculating the number of reads that were correctly or incorrectly mapped.
- Computing runtimes and measures of accuracy.

The ability to generate random reference genomes enables systematic studies of the effect of various factors on performance. In particular, besides specifying the length of the reference genome, the user can also adjust different repeat parameters—repeat count, repeat length and repeat variability rate (the probability of altering a base at each genome location during a repeat). This repeat variability rate is intended to introduce variability in the potential mappings of a read. Repeats are quite common in real genomes (Cheung *et al.*, 2003).

Our evaluation simulates reads from a reference genome, choosing uniformly distributed locations at random and making reads from fragments of normally distributed sizes. In the paired-end case, the underlying fragment is of normally distributed size and the read length at each end is fixed. The user can evaluate the effect of various factors by adjusting the following parameters:

- *Read length*: this is the average number of bases in each read. In current platforms, read length ranges from 30 to hundreds of base pairs.
- *Sequencing error rate*: this is the fraction of miscalled bases in a sequencing run. It also implicitly accounts for single nucleotide variants in the population. The error rate reported by current platforms is around 1% (Illumina, 2010).
- *Indel rate*: in addition to base read errors, we also consider short insertions and deletions, which may be caused by sequencing errors or variations in the population. This parameter specifies the fraction of short insertions and deletions in simulated reads.
- *Indel length*: this parameter controls the length of short insertions and deletions in the reference genome and is used to assess the robustness of an alignment tool. The length of indels is selected from a normal distribution and the indel length parameter determines the mean of this distribution.
- *Coverage*: since the reads come from random locations on the genome, it is important to have sufficient number of reads to adequately cover the entire genome. If the length of the reference genome is n, read length is m, and the number of mapped reads is k, then coverage is defined as mk/n, i.e. the expected number of aligned read bases that cover a given reference base position. Note that coverage does not have a direct effect on the accuracy of an alignment algorithm since each read is aligned independently. However, it is important for an alignment algorithm to scale to realistic levels of coverage in terms of runtime performance.

3.2 Evaluation

Most tools report a quality score for the mapping of a read to the reference genome. These scores mirror Phred scores (Ewing and Green, 1998); they represent the log-scaled probability that the mapping is incorrect. Meaningful scores typically range from 0 to 60, where 0 corresponds to very low-quality mapping and scores of > 30 are considered to be very good. A score of 30 denotes a 10^{-3} chance that the mapping is incorrect; as the score increases to 40, the chance of an incorrect mapping theoretically drops to 10^{-4}.

Our evaluation incorporates a threshold on this mapping quality; we only consider reads whose quality is reported to be greater than or equal to a certain value. This threshold value is used as a parameter in calculating the accuracy of a set of read mappings. We define the performance figures from the perspective of reads, i.e. the true location of a read is considered the truth and an alignment is considered a prediction.

Note that the use of one evaluation method for all tools is not appropriate, since some tools (mrFAST and mrsFAST in particular) report all matching genome positions while others report only the 'best' mapping. Standard definitions of an 'incorrect mapping' would unfairly penalize tools that report multiple mappings since a read may map equally well to multiple locations due to paralogous sequences in the reference genome. Motivated by this observation, we use two alternate definitions of an incorrect mapping, namely a *strict incorrect mapping* and a *relaxed incorrect mapping*. For a fixed threshold on mapping quality, we classify the accuracy of the mapping(s) of a read as follows.

- *Correctly mapped read (CM)*: the read is mapped to the correct location in the genome and its quality score is greater than or equal to the threshold.
- *Incorrectly mapped read—strict (IM-S)*: the read is mapped to an incorrect location in the genome and its quality score is greater than or equal to the threshold.
- *Incorrectly mapped read—relaxed (IM-R)*: the read is mapped to an incorrect location in the genome, its quality score is greater than or equal to the threshold and there is no correct alignment for that read with quality score higher than the threshold.
- *Unmapped read (UM)*: the read is not mapped at all by the alignment tool or the quality score is less than the threshold.

For a given set of reads, we compute *strict accuracy* as $\frac{|\text{CM}|}{|\text{CM}|+|\text{IM-S}|}$ and *relaxed accuracy* as $\frac{|\text{CM}|}{|\text{CM}|+|\text{IM-R}|}$, where CM denotes the set of correctly mapped reads, IM-S denotes the set of incorrectly mapped reads in the strict sense and IM-R denotes the set of incorrectly mapped reads in the relaxed sense. For example, if a read is mapped to four locations in the reference genome and one of those mappings is correct, the other three alignments are not counted as incorrect mappings in the relaxed sense. Note that strict and relaxed accuracy provide two extreme (respectively pessimistic and optimistic) measures of accuracy; therefore, they provide an interval for the accuracy of an algorithm. These two measures are equal if the tool reports a single genome location for each read. Furthermore, to assess the ability of a tool in finding a mapping for all reads, we define the used read ratio for an alignment tool as $\frac{|\text{CM}|+|\text{IM-S}|}{|\text{CM}|+|\text{IM-S}|+|\text{UM}|}$.

4 RESULTS

4.1 Accuracy

We simulate reads from two genomes: an artificially generated genome and the human genome [release 19 (International Human Genome Sequencing Consortium, 2001; Rhead *et al.*, 2010)]. The generated genome is of length 500 Mb, with 100 repeats of length 500 bp each. Results from the simulated genome are available in the Supplementary Material. Due to computational considerations, SHRiMP's accuracy results are only available for the simulated genome.

Table 1 shows the details of each experiment.

4.1.1 Varying error rate The accuracy of all algorithms on the human genome for varying error rate is compared in Figure 1. The results for quality threshold 0 (accepting all reads) are shown in Figure 1a, whereas Figure 1b shows the mapping accuracy when considering reads of quality $\geq$10. We can see that Bowtie, BWA and Novoalign are the most sensitive to mapping quality threshold at high error rates; their accuracy significantly increases as reads of

Table 1. Experimental setup for each simulation type: genome size(s), read length and read count

Evaluation type	Genome size (s) (Gb, Mb)	Read length (bp)	Read count	Error rate	Indel size	Indel freq.
Accuracy: varying error rate	3, 500	50	500 000	[0, 0.1]	0	0
Accuracy: varying indel size	3, 500	50	500 000	0.01	[2, 16]	0.02
Accuracy: varying indel frequency	3, 500	50	500 000	0.01	2	$[10^{-5}, 10^{-2}]$

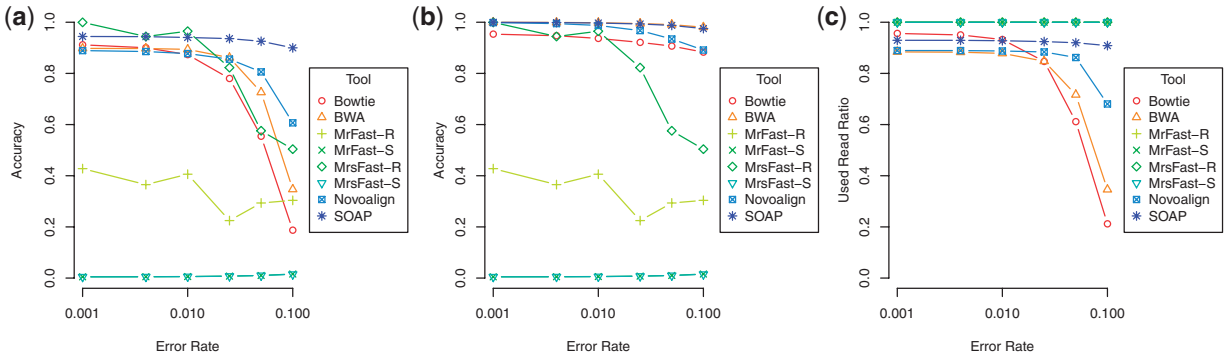

Fig. 1. Human genome: accuracy with varying error rate. (**a**) Shows mapping quality threshold 0, (**b**) shows threshold 10 and (**c**) shows the proportion of reads that have mapping quality of at least 10. -R and -S suffixes denote relaxed and strict accuracy, respectively.

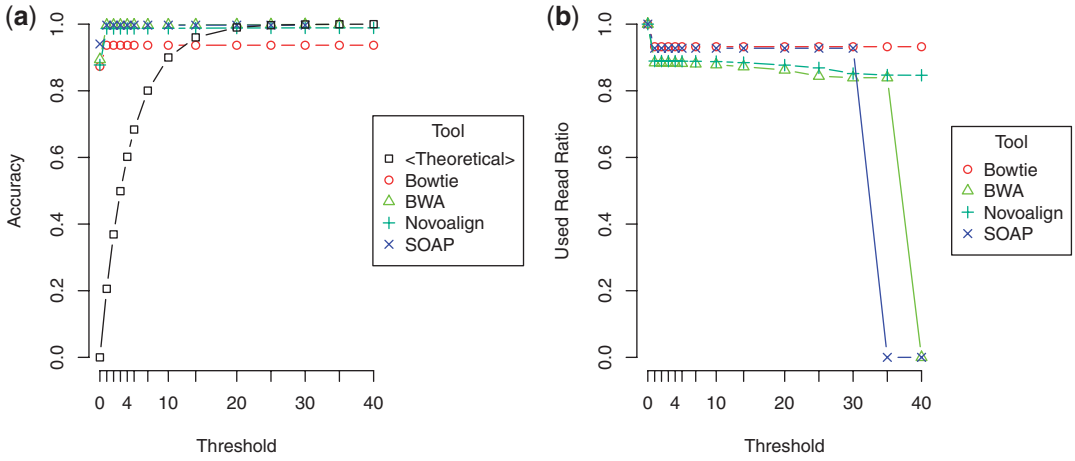

Fig. 2. Human genome: comparison of reported accuracy versus theoretical accuracy for 0.1% base call error rate (only tools that report meaningful quality scores are included). (**a**) Shows a comparison of the theoretical accuracy for each mapping quality score versus each tool's accuracy at that quality threshold. (**b**) Shows the proportion of reads with a mapping quality greater than or equal to each threshold value.

mapping quality 0 are discarded. SOAP's mapping accuracy is quite high even at quality threshold 0, which is consistent with its intended usage for genotyping SNPs. Figure 1c shows the proportion of mapping results that are used to create Figure 1b, i.e. the proportion of reads that have mapping quality of at least 10.

Figure 2 shows a direct comparison of the theoretical accuracy at each quality score against each tool's actual accuracy. The mapping quality Q is defined as the log-scaled probability P that the mapping is *incorrect*: $Q=-10\log_{10}P$, giving a theoretical accuracy A for each quality score: $A=1-P=1-10^{-Q/10}$. Figure 2a shows that most tools underestimate their mapping quality; most incorrect mappings can be discarded simply by considering mapping qualities of at least 1.

4.1.2 Varying indel sizes Figure 3 shows the accuracy of the alignment tools with fixed indel rate (0.05/bp) as the average indel size varies. This level of indel rate can be considered 'frequent' (as seen in Fig. 4). These figures again emphasize that SOAP is better suited for SNP analysis than indel calling—as the average indel size approaches 10, SOAP fails to align any reads and its accuracy drops to 0. Bowtie, BWA and Novoalign show very unfavorable accuracy when all reads are considered (i.e. when the mapping quality threshold is low); however, it can be seen that they report many of the incorrect mappings with low-quality scores, since their accuracy with quality threshold 10 is significantly improved. It can also be seen in these figures that mr(s)FAST and Novoalign are most robust to longer indels and Novoalign's mapping quality scores become particularly useful as indels get longer.

4.1.3 Varying indel frequencies The accuracy provided by each tool for fixed indel length (2) and varying indel rate on the human genome is shown in Figure 4. As seen in this figure, the accuracy of all algorithms depend strictly on indel rate; the accuracy provided by all algorithms is almost always >95% for indel rate <0.001. It should be also noted that mr- and mrsFAST show a tremendous difference in the relaxed and strict precision measures with varying indel frequency; they not only report the correct genome location, but also report many incorrect locations.

4.2 Runtime

Figure 5 compares the runtime of the different tools that we analyze, both in indexing time and alignment time. Figure 5a shows the

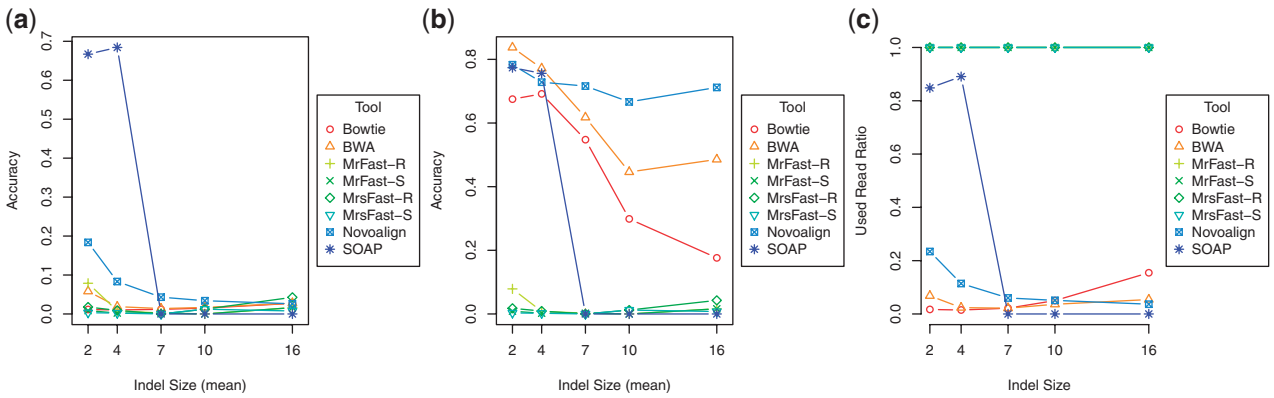

Fig. 3. Human genome: accuracy with varying indel sizes when indel frequency is fixed at 0.05/bp. (**a**) Shows mapping quality threshold 0, (**b**) shows threshold 10 and (**c**) shows the proportion of reads that have mapping quality of at least 10. -R and -S suffixes denote relaxed and strict accuracy, respectively. At indel sizes 10 and 16, SOAP discards all reads, producing missing values in (c).

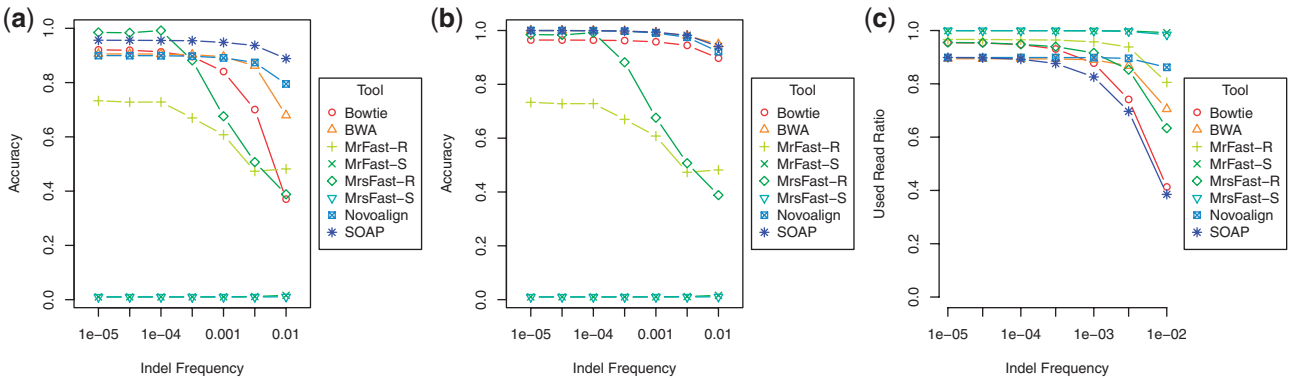

Fig. 4. Human genome: accuracy with varying indel frequencies when mean indel size is fixed at 2. (**a**) Shows mapping quality threshold 0, (**b**) shows threshold 10 and (**c**) shows the proportion of reads that have mapping quality of at least 10. -R and -S suffixes denote relaxed and strict accuracy, respectively.

indexing time for various genome sizes—most tools show a linear relationship between the length of the genome and the time required to build an index. Figure 5b shows the alignment runtime versus read count on a 500 Mb genome.

We can see that most tools are designed with a trade-off between indexing runtime and alignment runtime; Bowtie, BWA and SOAP align quickly but require significant amounts of time to build an index of a genome. Novoalign, conversely, requires little indexing time but shows more of a dependence on the number of reads. Interestingly, SHRiMP, seems to show no dependence on read count.

5 DISCUSSION

As expected, these alignment tools are designed with different approaches to trading off speed and accuracy to optimize detection of different types of variations in donor genomes. This trade-off is evident in the performance of BWA and SOAP on the human genome (Fig. 1): without a threshold value to eliminate unreliable reads, BWA is not as accurate even at low error rates ($\sim$0.9 at a base pair substitution rate of 10^{-3} and falling sharply to $\sim$0.37 at an error rate of 10^{-1}). SOAP has a consistently high accuracy ($\sim$0.95) even with no threshold and high error rates. Based on these observations, we can conclude that BWA is specifically designed not to miss any potential mappings, at the cost of reporting many incorrect mappings.

The evaluation of mrFAST, mrsFAST and SHRiMP shows some expected trends; since each fragment is potentially mapped to many locations in the genome, we expect their strict accuracy value to be much lower than that of other tools. As the error rate increases from 0.001 to 0.1, however, we see the strict accuracy measure *increase* for all three of these tools. Intuitively, this seemingly surprising trend makes sense since we expect the number of potential genome mappings to decrease as the reads become less reliable, thus reducing the number of incorrect mappings in relation to the single potential correct mapping. These tools' relaxed accuracy values (as defined in Section 3.2) also show some expected trends; since mrFAST and mrsFAST can report many genome locations for each fragment, we expect their relaxed accuracy to be quite high for low error rates and to decline as the error rate increases.

We must emphasize the large difference between our relaxed and strict accuracy measures in our evaluation of mrFAST, mrsFAST and SHRiMP. The relative usefulness of these two measures depends on the user's specific research aims; one may be more interested in tools with good relaxed accuracy if studying structural variants,

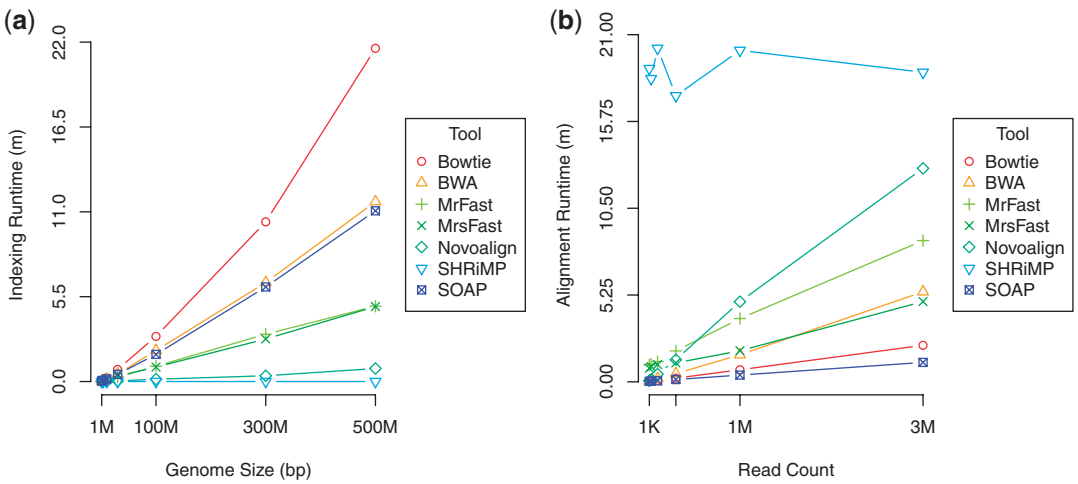

Fig. 5. Runtime measurements: (**a**) Shows indexing time versus genome size, (**b**) shows alignment time versus read count on a 500 Mb genome.

while strict accuracy may be of more use in genotyping SNPs. We must also note that our analysis may not be fair to SHRiMP—this tool is designed for mapping color-space reads and our simulation does not generate this type of data.

As expected, the tools show an overall linear relationship between coverage (number of reads) and the total runtime. For most alignment tools, we can further separate the total runtime into separate measurements for indexing and alignment; if the index can be reused across multiple alignment runs, a high indexing time can be affordable. We believe that our results will be useful to a wide variety of genomic researchers, though we must recognize that we cannot precisely simulate all experimental scenarios or sequencing hardware characteristics. As the state of the art advances, data from new sequencing hardware may challenge the assumptions that today's high-performing algorithms depend on. Similarly, algorithms with unfavorable accuracy or speed on today's data sets may find renewed use in the future.

ACKNOWLEDGMENTS

We would like to thank the developers of all methods compared in this paper for making their software available. We would also like to thank Gökhan Yavaş for many useful discussions.

Funding: This work was supported by National Science Foundation Award IIS-0916102.

Conflict of Interest: none declared.

REFERENCES

Alkan,C. *et al.* (2009) Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.*, **41**, 1061–1067.

Auffray,C. *et al.* (2009) Systems medicine: the future of medical genomics and healthcare. *Genome Med.*, **1**, 1–2.

Bailey,J.A. *et al.* (2002) Recent segmental duplications in the human genome. *Science*, **297**, 1003–1007.

Burrows,M. *et al.* (1994) A block-sorting lossless data compression algorithm. *Technical Report 124*. Digital Equipment Corporation.

Califano,A. and Rigoutsos,I. (1993) FLASH: a fast look-up algorithm for string homology. In *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, Palo Alto, CA, pp. 56–64.

Cheung,J. *et al.* (2003) Genome-wide detection of segmental duplications and potential assembly errors in the human genome sequence. *Genome Biol.*, **4**, R25.

Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using Phred. II. error probabilities. *Genome Res.*, **8**, 186–194.

Ferragina,P. and Manzini,G. (2000) Opportunistic data structures with applications. In *Proceedings of the 41st Symposium on Foundations of Computer Science (FOCS 2000)*, IEEE Computer Society, pp. 390–398.

Guffanti,A. *et al.* (2009) A transcriptional sketch of a primary human breast cancer by 454 deep sequencing. *BMC Genomics*, **10**, 163–179.

Hach,F. *et al.* (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nat. Methods*, **7**, 576–577.

Horner,D.S. *et al.* (2010) Bioinformatics approaches for genomics and post genomics applications of next-generation sequencing. *Brief. Bioinformatics*, **11**, 181–197.

Illumina,I. (2010) Quality scores data. Available at http://www.illumina.com /truseq/truth_in_data/sbl_comparison/quality_scores_data_.ilmn (last accessed date October 29, 2010).

International Human Genome Sequencing Consortium (2001) Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.

Li,H. and Durbin,R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, **26**, 589–595.

Li,H. and Homer,N. (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinformatics*, **11**, 473–483.

Li,H. *et al.* (2008a) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.*, **18**, 1851–1858.

Li,R. *et al.* (2008b) SOAP: short oligonucleotide alignment program. *Bioinformatics*, **24**, 713–714.

Li,R. *et al.* (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, **25**, 1966–1967.

Medvedev,P. *et al.* (2009) Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods*, **6**, S13–S20.

Miller,J.R. *et al.* (2010) Assembly algorithms for next-generation sequencing data. *Genomics*, **95**, 315–327.

Novocraft (2010) http://www.novocraft.com/. (last accessed date October 28, 2010).

Qin,J. *et al.* (2010) A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, **464**, 59–65.

Rasmussen,K.R. *et al.* (2005) Efficient *q*-gram filters for finding all *e*-matches over a given length. In Miyano,S. *et al.* (eds) *Research in Computational*

Molecular Biology, Vol. 3500 of *Lecture Notes in Computer Science*. Springer, Berlin/Heidelberg, pp. 189–203.

Rhead,B. *et al.* (2010) The UCSC Genome Browser database: update 2010. *Nucleic Acids Res.*, **38** (Suppl. 1), D613–D619.

Rumble,S.M. *et al.* (2009) Shrimp: accurate mapping of short color-space reads. *PLoS Comput. Biol.*, **5**, e1000386.

Schuster,S.C. (2007) Next-generation sequencing transforms today's biology. *Nat. Methods*, **5**, 16–18.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Sultan,M. *et al.* (2008) A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science* , **321**, 956–960.

Taylor,K.H. *et al.* (2007) Ultradeep bisulfite sequencing analysis of DNA methylation patterns in multiple gene promoters by 454 sequencing. *Cancer Res.*, **67**, 8511–8518.

Van Tassell,C.P. *et al.* (2008) SNP discovery and allele frequency estimation by deep sequencing of reduced representation libraries. *Nat. Methods*, **5**, 247–252.