Benchmarking Short Sequence Mapping Tools[a]

Ayat Hatem[1,2], Doruk Bozdağ[2], and Ümit V. Çatalyürek[1,2]
[1]*Department of Biomedical Informatics*
[2]*Department of Electrical and Computer Engineering*
The Ohio State University

dayat@bmi.osu.edu, bozdagd@bmi.osu.edu, umit@bmi.osu.edu

September 22, 2011

Technical Report
High Performance Computing Lab
Department of Biomedical Informatics
The Ohio State University

http://bmi.osu.edu/hpc/

---

http://bmi.osu.edu/hpc/papers/Hatem11-Bench.pdf

[a]A shorter version of this paper appears in IEEE International Conference on Bioinformatics and Biomedicine (BIBM) 2011.

Benchmarking Short Sequence Mapping Tools

Ayat Hatem[1,2], Doruk Bozdağ[2], and Ümit V. Çatalyürek[1,2]
[1]*Department of Biomedical Informatics*
[2]*Department of Electrical and Computer Engineering*
The Ohio State University

September 22, 2011

Abstract

The development of next-generation sequencing instruments has lead to the generation of millions of short sequences in a single run. The process of aligning these reads to a reference genome is time consuming and demands the development of fast and accurate alignment tools. However, the current proposed tools make different compromises between the accuracy and the speed of mapping. Moreover, many important aspects are overlooked when comparing the performance of a newly developed tool to the state of the art. Therefore, there is a need for an objective evaluation method that covers the various aspects. In this work, we introduce a benchmarking suite to extensively analyze various tools with respect to the different comparison aspects and provide an objective comparison. In order to assess our work, we applied our benchmarking tests on seven well known mapping tools, namely, Bowtie, BWA, SOAP, MAQ, RMAP, GSNAP, and FANGS. Bowtie, BWA, SOAP, GSNAP, and FANGS are based on indexing the reference genome, whereas MAQ and RMAP are based on building hash tables for the reads. It is shown that the benchmarking tests reveal the strengths and weaknesses of each tool. In addition, the tests can be further applied to other tools. The results show that there is no clear winner. However, Bowtie maintained the best throughput for most of the tests while BWA performed better for longer read lengths.

1 Introduction

Next-generation sequencing (NGS) technology has evolved rapidly, leading to the generation of hundreds of millions of sequences (reads) in a single run. The number of generated reads varies between 1 million for long reads (≈400 bps) and 2.4 billion for short reads (≈75 bps). Efficient alignment (mapping) of this large amount of reads with high accuracy is a crucial part in many applications' workflow such as genome resequencing [1], DNA methylation [2], RNA-seq, and ChIP sequencing. To undertake this challenging task, numerous tools have been developed, including MAQ [3], RMAP [4], GSNAP [5], Bowtie [6], BWA [7], SOAP [8], Mosaik[1], FANGS [9], SHRIMP [10] and Novoalign[2]. However, due to using different mapping techniques, each tool provides different trade-offs between speed and quality of the results. For instance, quality is often compromised in the following ways to reduce runtime:

- Neglecting base quality score.
- Limiting the number of allowed mismatches.

[1]http://bioinformatics.bc.edu/marthlab/Mosaik
[2]http://www.novocraft.com

- Disabling gapped alignment or limiting the gap length.
- Ignoring SNP information.

In most cases, it is unclear how such compromises affect the performance of newly developed tools in comparison to the state of the art tools. In addition, in most of the previous studies (e.g., [5] , [7], [11], and [12]), either the default options were only used or only one comparison aspect was considered. Moreover, the main aim of these studies was providing new tools rather than providing a thorough comparison. Therefore, there is a need for a quantitative evaluation method to systematically compare mapping tools in different aspects. In this paper, we tackle this problem and present a benchmarking suite to evaluate and understand the strengths and weaknesses of each tool. The set of tests provided in this suite cover a variety of input properties and algorithmic features. Input properties refer to the type of the reference genome and the properties of the reads including their length and source. Algorithmic features, on the other hand, pertain to the features provided by the mapping tool regarding its performance and utility. To assess our work, we apply these tests on six well known short sequence mapping tools, namely, Bowtie, BWA, SOAP, MAQ, RMAP, and GSNAP. Moreover, we compare the performance of these tools with that of FANGS, a long read mapping tool, to show their effectiveness in handling long reads. The results of the benchmarking tests depend on the features provided by the tools and the performance with default options. Therefore, in this paper, we will first present these major points and show their effect on the comparison.

2 Methods

For most existing tools (and all the ones we consider), the mapping process starts by building an index for the reference genome or the reads. Then, the index is used to find the corresponding genomic positions for each read. The two most common techniques to build the index are based on hash tables and the Burrows-Wheeler Transform [13]. In this section, we first give a brief description of the two techniques. Then, we explain the features the tools support and the default options setup.

Hash Tables: The hash based method is divided into two types: hashing the reads and hashing the genome. In general, the main idea for both types is to build a hash table for subsequences of the reads/genome. The key of each entry is a subsequence while the value is a list of positions where the subsequence can be found. Tools based on hashing the reads include MAQ and RMAP while GSNAP and FANGS are based on hashing the genome.

Burrows-Wheeler Transform (BWT): BWT [14] is an efficient data indexing technique that maintains a relatively small memory footprint when searching through a given data block. BWT was extended by Ferragina and Manzini [15] to a newer data structure, called FM-index, to support exact matching. The FM-index is used in several recent short sequence mapping tools such as BWA, Bowtie and SOAP. By transforming the genome into an FM-index, the lookup performance of the algorithm improves for the cases where a single read matches multiple locations in the genome. However, the improved performance comes with a significantly large index build up time.

2.1 Features

Inexact matching of DNA sequences to a genome is a special case of string matching. Incorporating known properties of DNA sequences and sequencing technologies brings additional complexity to the mapping process. Such properties include:

1. *Seeding* represents the first few tens of base pairs of a read. The seed part of a read is expected to contain less erroneous characters due to the specifics of the NGS technologies. Therefore, most tools use the seeding property to maximize performance and accuracy.

2. *Base quality scores* provide a measure on correctness of each character in the read. Some tools use them to decide mismatch locations. Others accept or reject the read based on the sum of the quality scores at mismatch positions.

3. *Existence of indels* necessitates inserting or deleting nucleotides when mapping a sequence to a reference genome (gaps). The complexity of choosing a gap location increases with the read length. Therefore, some tools do not allow any gaps while others limit their locations and numbers.

4. *Mate-pair (Paired-end) reads* result from sequencing both ends of a DNA molecule. Mapping mate-pair reads increases the confidence in the mapping locations due to having an estimation of the distance between the two ends.

5. *Color space reads* are a different type of reads generated by SOLiD sequencers. Each base is given a number (color) out of four numbers depending on the value of the previous base. The reads can be converted into bases, however, performing the mapping in color space has advantages in terms of error detection.

6. *Splicing* occurs when transcriptional reads are located across exon-exon junctions. It scatters the two parts of the read up to a distance of 20000 bps. Detection of such long gaps is difficult especially for short reads.

7. *SNPs* are variations of a single nucleotide between members of the same species. SNPs are not mismatches. Therefore, their location should be identified before mapping reads in order to correctly identify actual mismatch positions.

8. *Bisulfite treatment* is a method used for the study of the SP mythlation state of the DNA [2]. In bisulfite treated reads, each unmethylated cytosine is converted to uracil. Therefore, they require special handling in order to not misalign the reads.

A discussion related to these features can be found in [13]. In general, each tool supports only a subset of the features. Moreover, there are differences in the way the features are handled, which are explained in Table 1. For instance, BWA, SOAP, and GSNAP accept or reject an alignment based on counting the number of mismatches between the read and the corresponding genomic position. On the other hand, Bowtie and MAQ use a quality threshold to perform the same function, where the quality threshold is the sum of base quality scores at mismatch positions. In some cases, the features are partially supported. For example, SOAP supports gapped alignment only for paired end reads, and BWA limits the gap size. Therefore, considering only one of the above features when comparing between the tools would lead to under- or over-estimation of tools' performance.

2.2 Default options

In general, using a tool's default options yields good performance while maintaining a good output quality. Most users use the tools with the default options or by only tweaking some of them. Therefore, it is important to understand the effect of using these options and the kind of compromises made when using them. For the seven tools considered in this paper, the most crucial default options are the following:

1. Maximum number of mismatches in the seed: all tools use a default value of 2.

2. Maximum number of mismatches in the read: BWA and GSNAP determine it based on the read length. It is 10 for RMAP and 5 for SOAP and FANGS.
3. Seed length: It is 24 for MAQ, 32 for RMAP, and 28 for Bowtie. BWA disables seeding while SOAP considers the whole read as the seed.
4. Quality threshold: It is equal 70 for MAQ and Bowtie.
5. Splicing: This option is enabled for GSNAP.
6. Gapped alignment: It is enabled for GSNAP, BWA, and MAQ by default while it is disabled for SOAP.
7. Minimum and maximum insert sizes for paired-end mapping: The insert size represents the distance between the two ends. The values used for the minimum and the maximum insert sizes are 0 and 250 for Bowtie, 0 and 500 for BWA, 400 and 500 for SOAP, 0 and 250 for MAQ, and 100 and 400 for RMAP.

Each tool has different default values, thus leading to different results for the same data set. Hence, using the same values when comparing between the tools is important.

2.3 Features to be tested

In this section, we present the features covered by our benchmarking suite. In addition, we explain how they were previously addressed by the tools we mention in this paper. However, two algorithmic features, namely SNPs and Splicing awareness, are not presented in the results section due to being supported only by one tool. The features are categorized as follows:

1. Mapping options

   Quality threshold: MAQ and Bowtie use the quality threshold to determine the number of allowed mismatches. Therefore, setting a quality threshold is similar to explicitly setting the number of mismatches. However, there is no hard limit on the actual number of mismatches. The impact of varying the quality threshold has not been studied before.

   Number of mismatches: Changing the number of allowed mismatches affects the percentage of mapped reads. This effect was studied in [5], however, the mismatches were generated uniformly on the genome which does not mimic real mismatches distribution.

Table 1: Features supported by each tool. PE.: paired-end only, mm.: mismatches, QS.: base quality score, count: total count of mismatches in the read, and missed means not supported.

	Bowtie	BWA	SOAP	MAQ	GSNAP	RMAP	FANGS
Seed mm.	Up to 3	Any	Up to 2	Any		Any	
Non-seed mm.	QS	Count	Count	QS	Count	Count	Count
Var. seed len.	> 5	Any	> 28				
Mapping qual.		Yes		Yes			
Gapped align.		Yes	PE	PE	Yes		Yes
Colorspace	Yes	Yes		Yes			
Splicing					Yes		
SNP tolerance					Yes		
Bisulfite reads					Yes	Yes	

Seed length: Seeding-based tools impose limits on the number of mismatches in the seed part. As a result, increasing or decreasing the length of the seed part affects the percentage of mapped reads. The effect of the seed length has not been studied in details before.

2. Input properties

   Read length: The read length varies between 30bps for ABI's SOLiD and Illumina's Solexa sequencers up to 500 bps for Roche's 454. Therefore, the impact of read length should be considered for throughput evaluation. Even though the effect of the read length is explored in several studies, the default options were usually used leading to incomparable trade-offs.

   Paired-end reads: Mapping paired reads requires the mapping of both ends within a maximum distance between them. Hence, it adds a constraint when finding the corresponding genomic locations. Paired-end reads were only considered in analysis in [7].

   Genome type: The efficiency of most algorithms are tested by using the Human genome as the reference. However, each genome has its own properties such as the percentage of repeated regions and ambiguous characters. Therefore, using a single genome does not reveal the effect of these properties. To the best of our knowledge, BWA [7] was the only tool to test its performance on different genomes.

3. Algorithmic features

   Gapped alignment: is important for variant discovery due to the ability to detect indel polymorphism [13]. Most of the tools support gapped alignment for paired-end reads while GSNAP and BWA are the only tools to support it for single end reads. However, from the results provided by the different tools, it is not obvious how gapped alignment affects the performance.

   SNP awareness: Incorporating SNP information into mapping allows considering minor alleles as matches rather than mismatches. Currently, this feature is provided only by GSNAP. It was shown in [5] that integrating SNP information affected around 8% of the reads and allowed mapping 0.4% of unmapped reads.

   Splicing awareness: Reads located cross exon-exon junctions would be wrongly aligned using standard alignment algorithms. The only tool that currently supports splicing is GSNAP. It was shown in [5] that the alignment yield increased by 8-9% when splicing detection based on known splice junctions was introduced. However, there was only 0.3-0.6% increase in case of detecting novel splice junctions.

4. Scalability

   Mapping tools may show different scalability under different parallel settings. Many tools support multithreading, which is expected to yield linearly increasing speedup with the increase in the number of cpu cores. On the other hand, using multiprocesses is more general and may improve the throughput even for tools that do not support multithreading (e.g., MAQ and RMAP), where multiprocesses refers to using more than one process in a distributed memory fashion that communicate through a message passaging inferace. Scalability analysis had only been performed for Bowtie in a multi-threading setting.

3 Results and Discussion

In this section, we present results from our benchmarking tests. The experiments were performed on a cluster of quad core AMD 2378 Opteron CPUs at 2.4 GHz and 34 GB of RAM. We used SOAP v2.20, Bowtie v0.12.6, BWA v0.5.0, MAQ v0.7.0, RMAP v2.05.0, FANGS v0.2.3 and GSNAP v2010-07-27. We evaluated the tools on synthetic single and paired-end data sets generated by

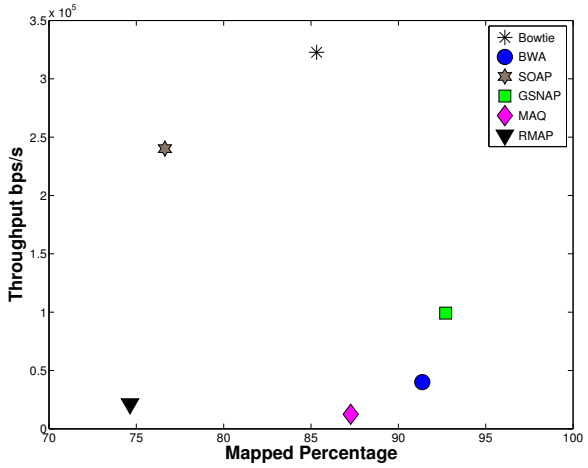

Figure 1: Mapping 16 million reads of length 70 extracted from the Human genome. Each tool was allowed to use its own default options.

wgsim[3]. In particular, the reads were generated with 0.09% SNP mutation rate, 0.01% indel mutation rate, 2% uniform sequencing base error rate, and with a maximum insert size of 500, which are the same parameters used in [7].

We evaluate the performance of the tools by considering two important factors: the mapping percentage and the throughput. The mapping percentage is the percentage of reads each tool maps while the throughput is the number of mapped base pairs per second. In addition, we report the percentage of the suboptimal (*subopt*) hits and the ambiguously mapped reads (*amb*). *subopt* is the percentage of mapped locations violating the mapping criteria while *amb* is the percentage of reads mapped to more than one location with the same number of mismatches.

First, we present the effect of the default options. The results for this experiment are given in Figure 1. As stated previously, tools try to use the options that yield a good performance while maintaining a good output quality. For instance, Bowtie achieves a throughput of around $3 \cdot 10^5$bps/s at the expense of mapping only 85% of the reads. On the other hand, BWA maps 91% of the reads at the expense of having a throughput of $0.4 \cdot 10^5$bps/s. However, using only the default options to build our conclusions is misleading. Indeed, further experiments show that BWA obtains a high throughput when allowed to use the same options as Bowtie. Moreover, BWA achieves a higher throughput than Bowtie in other experiments. Therefore, it is important to use the same options to truly understand how the tools' behave.

In the remaining experiments, unless otherwise stated, the number of mismatches in the seed and in the whole read are fixed to 2 and 5, respectively, while the quality threshold is 100. The minimum and maximum insert sizes allowed are 0 and 500, respectively. In addition, the splicing, SNPs, and gapped alignment options are disabled. For the number of reported hits, tools are only allowed to report one location. The default values are used for the remaining options.

The experiments are divided as follows:

3.1 Mapping options

Quality threshold is one of the two main metrics used for mismatch tolerance. The other main metric is the explicit specification of the number of mismatches. To compare fairly between the

[3]https://github.com/lh3/wgsim

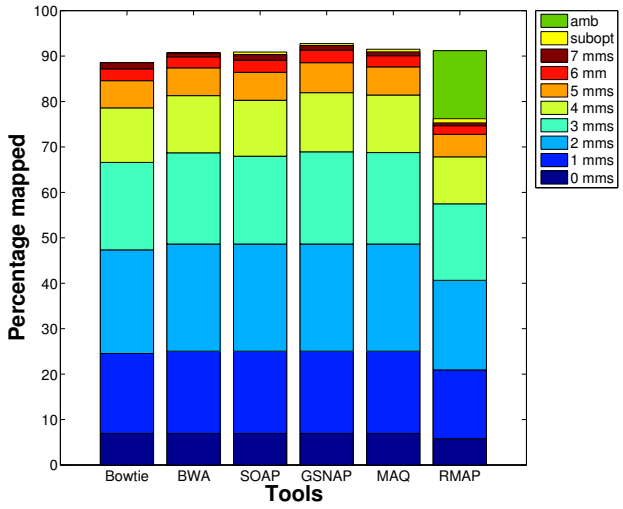

Figure 2: Mapping 1 million reads of length 125 while allowing up to 7 mismatches and a quality threshold of 140. The *subopt* is 0.6% for SOAP and MAQ and 0.45 for GSNAP.

tools, a relationship between the two metrics should be found, which is the main target of this experiment. In this experiment, a synthetic dataset with the same quality score for each base is used instead of real dataset. The different base quality scores in real data cause Bowtie and MAQ to allow more mismatches than the other tools. For instance, when allowing a quality threshold of 70 and 5 mismatches for the remaining tools, Bowtie and MAQ map reads with up to 10 mismatches while the other tools are limited to 5. The results for this experiment are not shown due to the space limitation. Nevertheless, in the following, we show how the tools map the reads within the same maximum number of mismatches using synthetic data.

For synthetic data, quality thresholds of 60, 80, 100, 120, and 140 should correspond to 3, 4, 5, 6, and 7 mismatches. To assess our conclusion, we designed an experiment where all tools were allowed a maximum of 7 mismatches while using a quality threshold of 140. Figure 2 shows that the tools map the reads with the same maximum number of mismatches. Moreover, the mapping rates are similar. However, there is a difference in the mapping rates for some of the tools due to the default options restricting the search space and the report of *subopt* hits. For instance, 0.6% of reported hits for MAQ and SOAP are suboptimal while Bowtie's default options limit the allowed number of backtracks to find mismatches. For RMAP, it does not report locations for the *amb* reads.Therefore, RMAP has the least mapping percentage. On the other hand, GSNAP maps 92% of the reads even though it reports *subopt* hits. This is due to being a non-seed based tool, hence, more mismatches are allowed to be found in the first few base pairs.

Number of mismatches Not only does the number of mismatches affect the percentage of mapped reads, but also affects the throughput. In particular, the mapping percentage increases nonlinearly with the number of mismatches. Figure 3 shows the effect of the number of mismatches in more details. There is a 20% increase in the percentage of mapped reads when allowing 3 mismatches instead of 2. On the other hand, there is less than 0.7% increase when allowing 7 mismatches instead of 6. In addition, the percentage of suboptimal hits decreases for large number of mismatches. For instance, SOAP reports 21% of suboptimal hits when allowing 2 mismatches while it is reduced to 1% when allowing 6 mismatches. From the throughput point of view, the tools behave differently. For instance, Bowtie, MAQ, and RMAP are able to maintain almost the same throughput while it increases for SOAP and GSNAP and drecreases for BWA. The degradation in BWA performance is due to exceeding the default number of mismatches leading

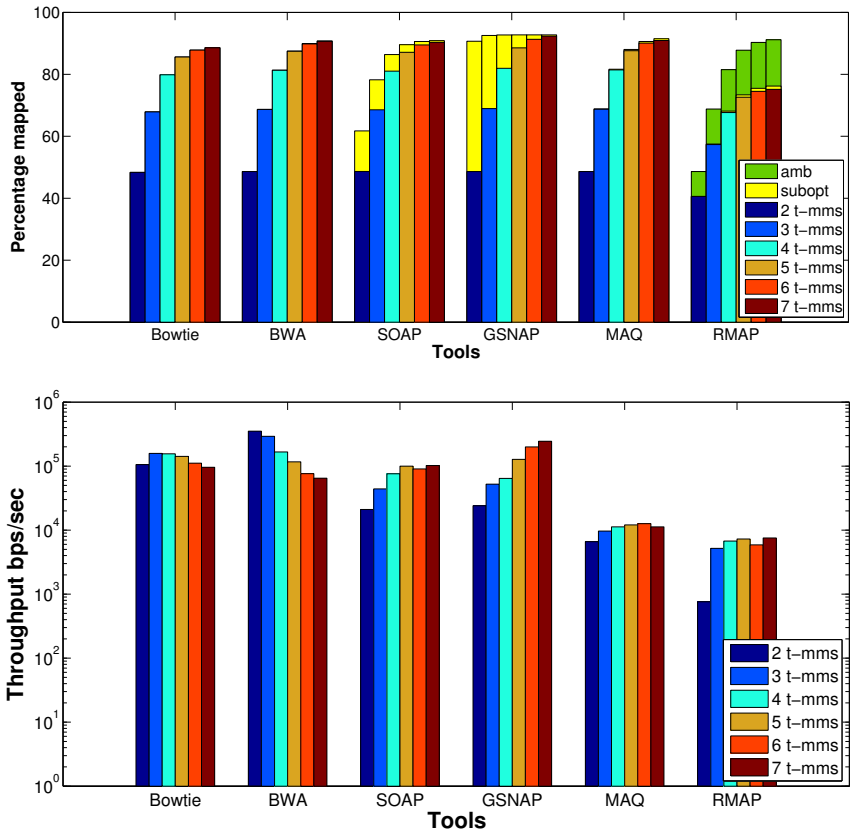

Figure 3: Comparing the different tools while changing the total mismatches from 2 to 7. T-mms stands for the maximum allowed mismatches. A data set of 1 million reads of length 125 extracted from the Human genome was used in this experiment.

to excessive backtracking to find mismatch locations.

Seed length Theoretically, when fixing the number of allowed mismatches in the seed and in the whole read, changing the seed length affects the mapping results. Specifically, a shorter seed allows more mismatches in the remaining part of the read to be found. Therefore, the percentage of mapped reads would increase even though the throughput would decrease. On the other hand, having a longer seed would result in pruning some parts of the search tree as soon as possible leading to throughput improvement. The aim of this experiment is to study this trade off. As shown in the results given in Figure 4, the tools behave as expected. However, there are some exceptions. For instance, when increasing the seed length from 32 to 36 the percentage of mapped reads for SOAP and Bowtie decreases, however the throughput is not affected. In addition, there is a 0.8% increase in the percentage of mapped reads for Bowtie when increasing the seed length from 28 to 32. This behavior is due to the backtracking property that stops once a certain limit is reached. Therefore, as a result of having less erroneous characters in the seed part, Bowtie can continue more in the depth first search without exceeding the backtracking limit.

3.2 Input properties

Read length Longer reads tend to have more mismatches beside requiring more time to be fully mapped [16]. In general, for a fixed number of mismatches, increasing the read length decreases the percentage of mapped reads. Therefore, the aim of this experiment is to understand the read length effect. The results in Figure 5 show that the mapping percentage decreases with the increase in the read length while the reporting of suboptimal hits increases. As an example, 95% of Fangs' output for read length 500 is suboptimal compared to 12% of its output for read length 200. This is due to the increase of the erroneous bases with the increase of the read length. Therefore, it becomes harder to map the reads with the specified mapping criteria. In addition, Bowtie and BWA were the only short sequence mapping tools that managed to map long reads. In particular, the max read length was 128 for MAQ, 300 for RMAP, and 200 for GSNAP while SOAP took more than 24 hours to map read length 300 and did not finish. From the throughput point of view, tools do not maintain the same behavior. For instance, the throughput of Bowtie and SOAP decreases for long read lengths. This is due to the backtracking property and the split strategy [8] used by Bowtie and SOAP, respectively, to find inexact matches. On the other hand, even thought the throughput of BWA and GSNAP increase with the read length, it starts to decrease for read length 500 and 200, respectively. GSNAP uses a faster mapping algorithm depending on the read length and the number of mismatches which is triggered in our case for read length ≥ 125. Therefore, the throughput increases. However, due to the work needed to generate and combine position lists to create candidate mapping regions, it starts to decrease for read length 200.

Paired-end Mapping paired-end reads affects the performance of the tools due to the added constraint of mapping both ends within a maximum insert size. However, this effect is not well understood as most of the tools were not tested on paired-end reads. Hence, we focus on understanding the effect of paired-end reads in this experiment. The results in Figure 6 (ungapped bars) show that the throughput decreases for all of the tools when mapping paired-end reads, except for BWA which was able to maintain almost the same throughput while MAQ had a small increase. Even though all of the algorithms work by finding mapping locations for each end alone and then try to find the best pair, GSNAP was the only tool to face a drop by 90% in the throughput.

Genome type To capture the effect of the genome type, we designed an experiment in which the Human, the Zebrafish, and the Lancelet genomes were used as reference genomes. The sizes of these genomes are 3.1Gbps, 1.5Gbps, and 0.9Gbps, respectively. Theoretically, for genome indexing based tools (i.e, Bowtie, BWA, SOAP, and GSNAP), the throughput is expected to increase with the decrease in the genome size. However, The results in Figure 7 show that some tools do not act as expected. For instance, SOAP's throughput decreases significantly for the

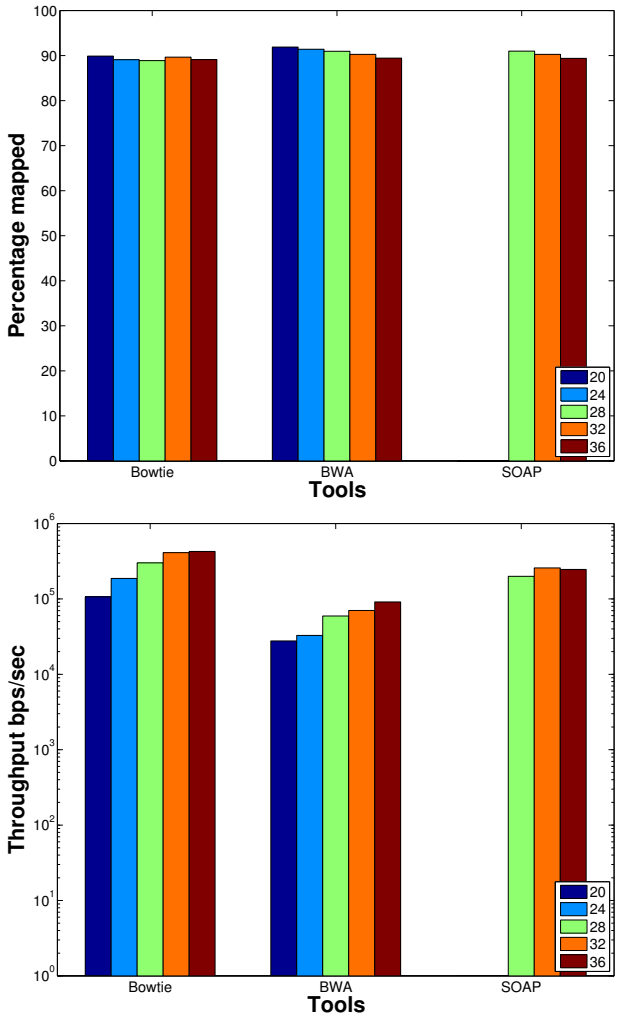

Figure 4: The effect of changing the seed length on the BWT based tools. The tools were used to map 16 million reads of length 70 bps on the Human genome. SOAP does not support seed length < 28.

Zebrafish genome while GSNAP did not finish its run on the same genome albeit running for two days. The reason for this behavior is the large repetition rate in the Zebrafish genome. For instance, when mapping 1 million randomly generated reads, around 600 reads were mapped to more than 100,000 locations in comparison to the Lancelet with the maximum number of locations is around 10,000 for only 1 read. Hence, for GSNAP, the large repetition rates lead to generating long genomic position lists; resulting in slowing down GSNAP significantly. Another interesting result is the ability of most of the tools to map more than 96% of the reads for the Zebrafish data set compared to around 91% for the Human and 89% for the Lancelet. The large mapping percentage is also due to the large repetition. Hence, due to synthetically generating the reads, large number of reads would be generated from the repeated regions, thus, even with a mismatch, the probability of finding a mapping location would increase.

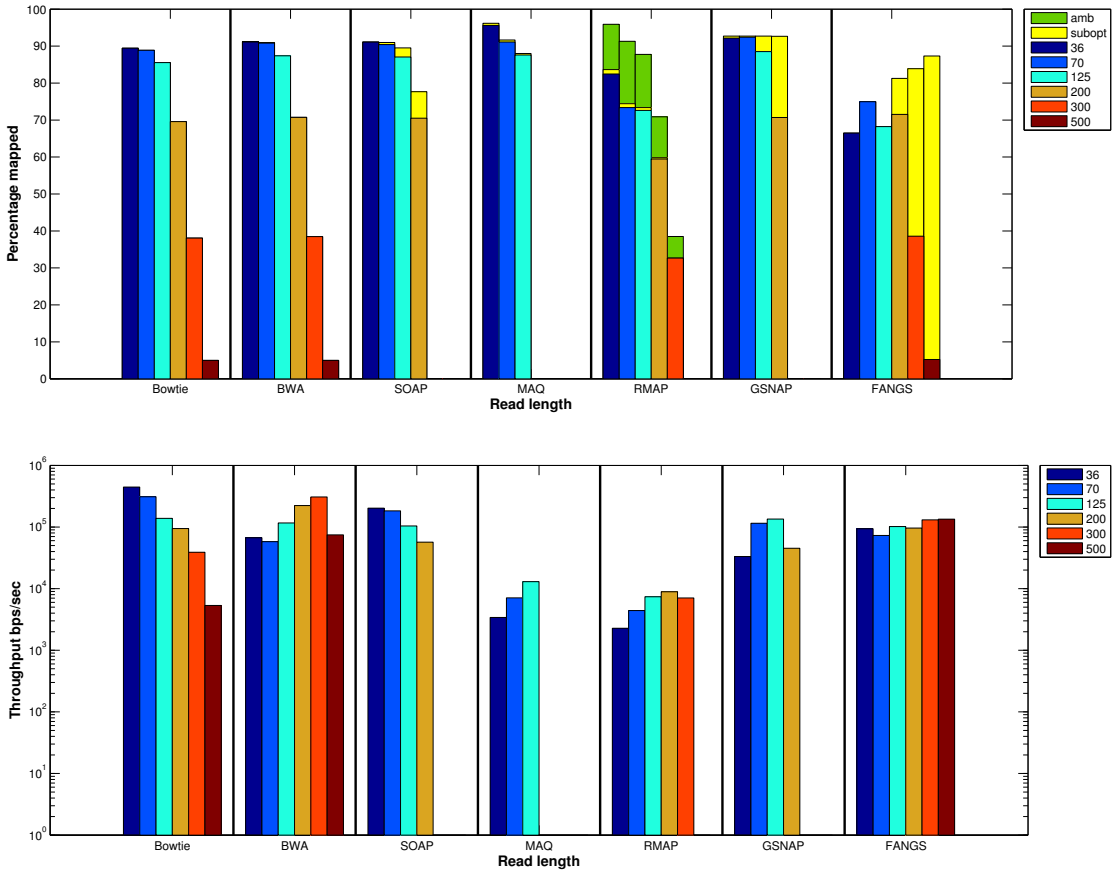

Figure 5: The effect of changing the read length from 36 to 500. The reads were generated from the Human genome. RMAP and MAQ are slower than the other tools. Therefore, 1 million reads were used to test MAQ and RMAP while 16 million reads were used for the remaining ones.

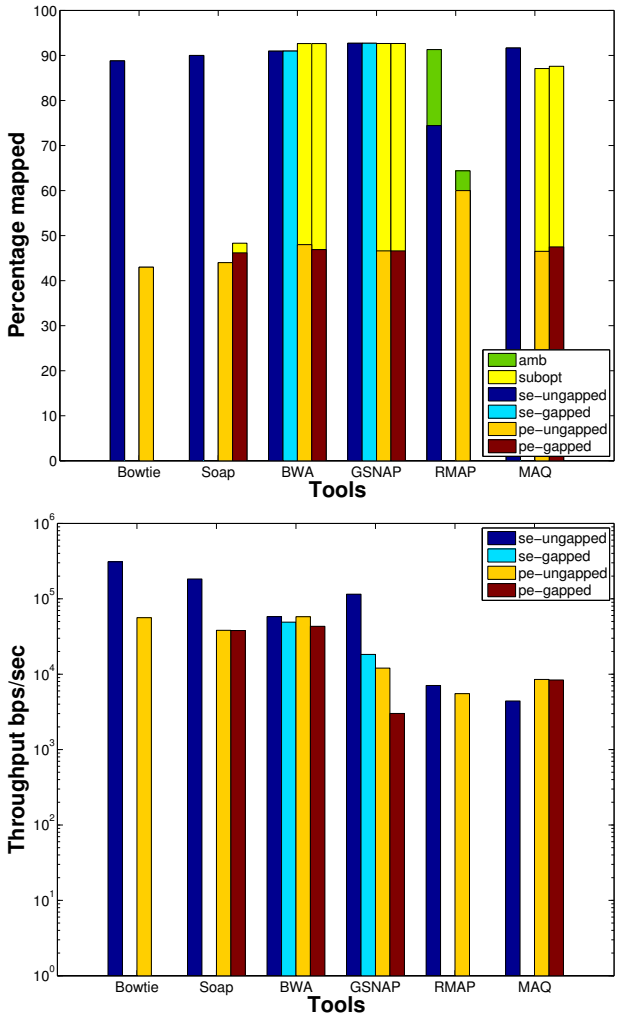

Figure 6: The effect of mapping paired end reads of length 70 to the Human genome. 1 million reads were used to test RMAP and MAQ while 16 million reads were used to test the other tools. SE and PE refer to single end and paired-end, respectively. *subopt* is only provided for PE due to exceeding the allowed insert size.

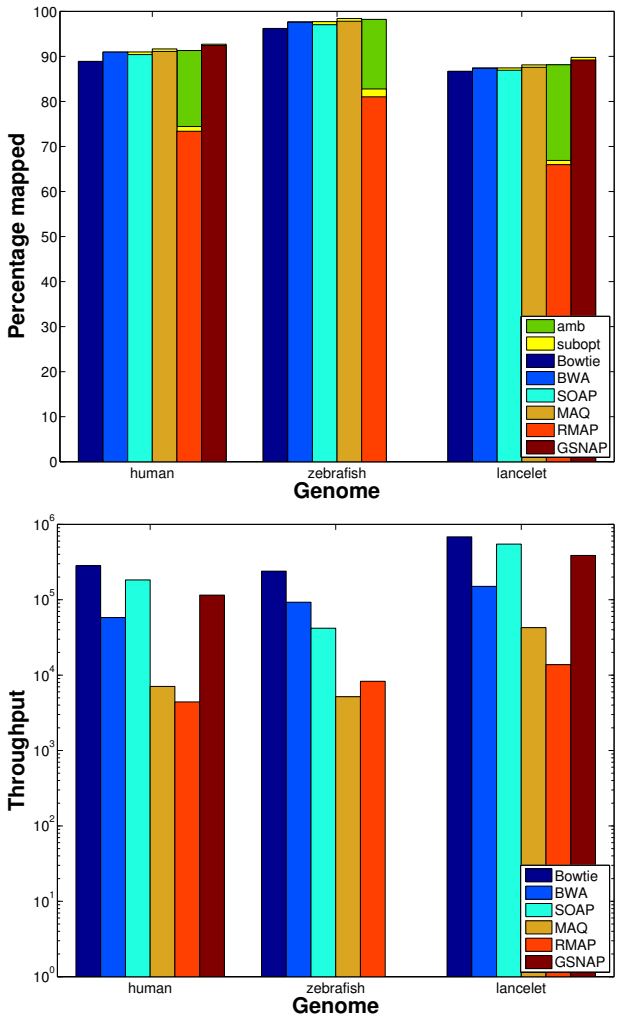

Figure 7: Effect of changing the genome type. 16 million reads of length 70bps were generated from the Human, Zebrafish, and Lancelet genomes for this test. However, only 1 million reads were used for MAQ and RMAP.

3.3 Algorithmic features

Gapped alignment is supported by most of the tools for paired-end reads, however, GSNAP and BWA are the only tools to support it for single end reads. In general, gapped alignment should improve the mapping percentage albeit decreasing the throughput. We designed an experiment to understand the effect of gapped alignment. Tools were used to map synthetically generated reads of length 70 to the Human genome while allowing one gap of length 3. The results in Figure 6 show that the mapping percentage increases by 4% for SOAP in case of gapped alignment, while there is no change for BWA and GSNAP. However, there is a drop of 15% and 75% in the throughput for BWA and GSNAP, respectively. The decrease for GSNAP is due to the overhead added to the algorithm to find pairs of candidate regions that co-localize within a maximum allowed gap size. Then, it tries to find a crossover between the two regions without exceeding the maximum number of mismatches leading to a significant decrease in the throughput.

3.4 Scalability

In this experiment, we tested the multithreading behavior. In addition, pMap[4] was used to run multiple instances of each tool on a number of processors on a single node to test the effect of using multiprocesses. pMap is an open-source MPI-based tool that enables parallelization of existing short sequence mapping tools by partitioning the reads and distributing the work among the different processors. A single node was used in the multiprocesses experiment to understand the effect of a good implementation of multithreading on the different tools. The results for both experiments are given in Figure 8. We can observe from the multithreading results that the tools had almost a linear speedup up to 4 threads. However, when increasing to 8 threads, Bowtie was the only tool to achieve 8x speedup. In addition, BWA had a similar speedup in both multithreading and multiprocesses. For the multiprocesses experiment, concerning the multithreading non-supporting tools, FANGS achieved almost a 6x speedup while there was a small improvement for MAQ and RMAP. For the remaining tools, most of them were able to maintain more than a 5x speedup for 8 processors, however this is less than a linear speedup. One reason for this degradation is the overhead of the distribution and merging steps required by distributed memory systems. As expected, we can notice that multithreading provides almost a linear speedup, however, it is limited by the number of cores.

In general, using multiprocesses provides more degrees of freedom by parallelizing tools that do not support multithreading and by making use of the available computational resources.

4 Conclusion

Next generation sequencing technology has led to the development of many sequence mapping tools. Choosing the best tool among them has become a challenging task. In this work, we provided a set of benchmarking tests that extensively analyze the performance of the different tools. Each of the benchmarking test stresses a different aspect. Furthermore, we showed that comparing between the tools from only one perspective leads to unfair comparison. In addition, using only the default options is not sufficient for a thorough comparison. The benchmarking tests were applied on the latest sequence mapping tools, namely, Bowtie, BWA, SOAP, MAQ, RMAP, GSNAP, and FANGS. We showed that each tool has its own strengths and weaknesses. Therefore, it is important to define each user's need to choose the most suitable tool. For instance, GSNAP is better than the other tools for large number of mismatches while its performance degrades significantly when mapping paired-end reads. Bowtie and BWA do not report any suboptimal hits, thus, leading to a more confidence in the mapped reads. In addition, changing the genome

[4] http://bmi.osu.edu/hpc/software/pmap/pmap.html

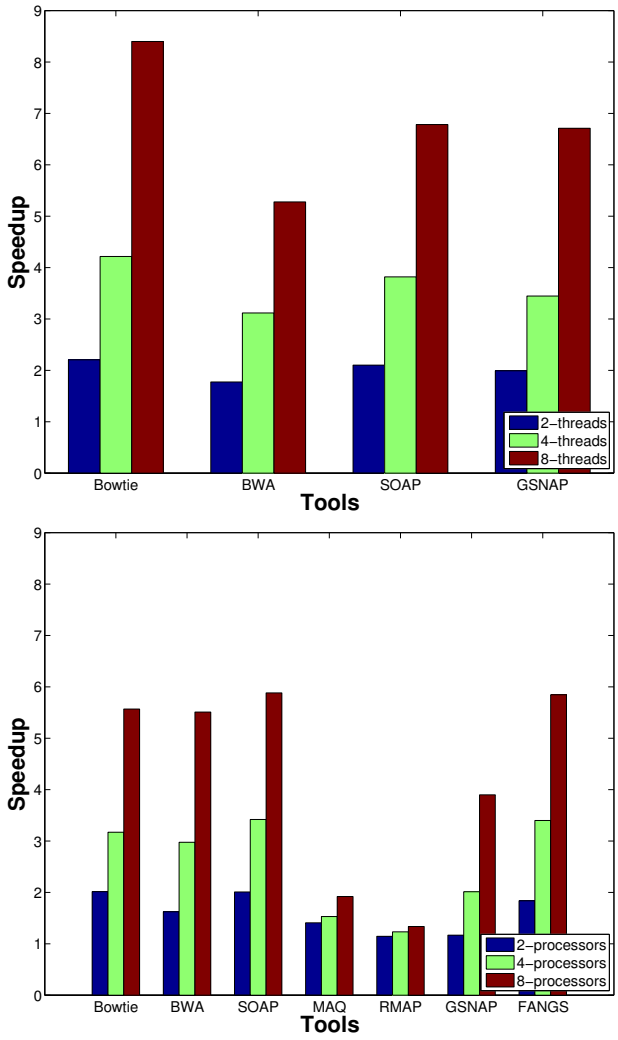

Figure 8: The speedup when mapping 16 million reads of length 125 extracted from the Human genome while allowing multithreading (the upper figure) or using multiprocesses (the lower figure).

type does not affect their performance. However, BWA scales better for longer reads while Bowtie gives the maximum speedup in multithreading mode. RMAP is superior than the other tools regarding the mapping percentage of paired end reads while SOAP performs better when allowing gapped alignment.

Acknowledgment

Funding: This work was supported in parts by the DOE grant DE-FC02-06ER2775; by the NSF grants CNS-0643969, OCI-0904809, and OCI-0904802.

References

[1] P. Flicek and E. Birney, "Sense from sequence reads: methods for alignment and assembly," *Nature Methods*, vol. 6, no. 11s, pp. S6–S12, October 2009.

[2] S. J. Cokus, S. Feng, X. Zhang, Z. Chen, B. Merriman, C. D. Haudenschild, S. Pradhan, S. F. Nelson, M. Pellegrini, and S. E. Jacobsen, "Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning," *Nature*, vol. 452, no. 7184, pp. 215–219, February 2008.

[3] H. Li, J. Ruan, and R. Durbin, "Mapping short dna sequencing reads and calling variants using mapping quality scores." *Genome research*, vol. 18, no. 11, pp. 1851–1858, November 2008.

[4] A. D. Smith, Z. Xuan, and M. Q. Zhang, "Using quality scores and longer reads improves accuracy of solexa read mapping." *BMC bioinformatics*, vol. 9, no. 1, pp. 128+, February 2008.

[5] T. Wu and S. Nacu, "Fast and SNP-tolerant detection of complex variants and splicing in short reads," *Bioinformatics*, vol. 7, no. 26, pp. 873–881, 2010.

[6] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome." *Genome biology*, vol. 10, no. 3, pp. R25+, 2009.

[7] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows-wheeler transform." *Bioinformatics (Oxford, England)*, vol. 25, no. 14, pp. 1754–1760, July 2009.

[8] R. Li, C. Yu, Y. Li, T. W. Lam, S. M. Yiu, K. Kristiansen, and J. Wang, "SOAP2: an improved ultrafast tool for short read alignment." *Bioinformatics (Oxford, England)*, vol. 25, no. 15, pp. 1966–1967, August 2009.

[9] S. Misra, R. Narayanan, S. Lin, and A. Choudhary, "FANGS: high speed sequence mapping for next generation sequencers," in *Proc. of the 2010 ACM Symposium on Applied Computing (SAC'10)*, 2010, pp. 1539–1546.

[10] S. M. Rumble, P. Lacroute, A. V. Dalca, M. Fiume, A. Sidow, and M. Brudno, "Shrimp: Accurate mapping of short color-space reads," *PLoS Comput Biol*, vol. 5, no. 5, pp. e1 000 386+, May 2009.

[11] S. Bao, R. Jiang, W. Kwan, B. Wang, X. Ma, and Y.-Q. Q. Song, "Evaluation of next-generation sequencing software in mapping and assembly." *Journal of human genetics*, vol. 56, pp. 406–414, Apr. 2011.

[12] M. Holtgrewe, A. K. Emde, D. Weese, and K. Reinert, "A novel and well-defined benchmarking method for second generation read mapping," *BMC Bioinformatics*, vol. 12, no. 1, pp. 210+, 2011.

[13] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in Bioinformatics*, vol. 11, no. 5, pp. 473–483, September 2010.

[14] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Digital Systems Research Center, Tech. Rep. 124, 1994.

[15] P. Ferragina and G. Manzini, "Opportunistic data structures with applications," in *Proc. of the 41st Annual Symposium on Foundations of Computer Science*, 2000.

[16] J. Schroder, H. Schroder, S. J. Puglisi, R. Sinha, and B. Schmidt, "Shrec: a short-read error correction method." *Bioinformatics*, vol. 25, no. 17, pp. 2157–2163, 2009.