BIOINFORMATICS

Vol. 00 no. 00 2010
Pages 1–3

SHRiMP2: Sensitive yet Practical Short Read Mapping

Matei David[1]*, Misko Dzamba[2], Dan Lister[2],Lucian Ilie[4], and Michael Brudno[2,3]

[1]Department of Computer Science, Princeton University
[2]Department of Computer Science, and [3]Donnelly Centre, University of Toronto
[4]Department of Computer Science, University of Western Ontario

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Summary: We report on a major update (version 2) of the original SHort Read Mapping Program (SHRiMP). SHRiMP2 primarily targets mapping sensitivity, and is able to achieve high accuracy at a very reasonable speed. SHRiMP2 supports both letter space and color space (AB/SOLiD) reads, enables for direct alignment of paired reads and uses parallel computation to fully utilize multi-core architectures.
Availability: SHRiMP2 executables and source code are freely available at: http://compbio.cs.toronto.edu/shrimp/
Contact: shrimp@cs.toronto.edu

1 INTRODUCTION

High Throughput Sequencing (HTS) machines produce datasets of 50–200 million reads of 32–400 base pairs (bp) per run. The first step in the analysis of HTS datasets is mapping the reads to a reference genome, which is followed by specialized processing tools that aim to identify signals (e.g. genomic variants, or high coverage peaks) from the mappings. Mapping HTS reads to a large reference genome is a non-trivial computational task, and the various read mapping programs that have been developed in recent years target different speed-accuracy trade-offs. Programs that primarily target speed are typically based on (near-)exact string matching methods, whereas programs that primarily target sensitivity (alignment of reads with high polymorphism, or to a distant reference) are often based on projections with spaced seeds. For a recent survey of the current read mapping programs, see (Li and Homer, 2010).

Here we report on a major update (version 2) of the SHort Read Mapping Program (SHRiMP) (Rumble *et al.*, 2009). SHRiMP2 primarily targets mapping accuracy, enabling the alignment of reads with extensive polymorphism and sequencing errors, while featuring a significant speedup over the previous versions. SHRiMP2 supports fasta and fastq input formats, SAM output format, mapping of Illumina/Solexa, Roche/454 and AB/SOLiD reads, a paired mapping mode, and parallel computation.

2 METHODS

SHRiMP2 works by indexing the genome using multiple spaced seeds, projecting each read to identify candidate mapping locations (CMLs), and ultimately investigating these CMLs with the Smith-Waterman algorithm. A major difference between the original SHRiMP and SHRiMP2 is that the former indexed the reads; switching to a genome index resulted in a dramatic speed increase and further allowed us to add a paired mapping mode and to utilize multi-threaded computation without affecting sensitivity. For more details on the methods described below, as well as for used below see the original SHRiMP paper (Rumble *et al.*, 2009).

Genome Index. SHRiMP2 starts by projecting the reference genome using several spaced seeds (Ilie and Ilie, 2007). Each seed is applied at each genome location, obtaining a (spaced) kmer. For every seed and every k-mer, the genome index contains a list of locations where that k-mer can be found using that seed. Ubiquitous k-mers (with very long lists) are discarded, as they do not help identify CMLs.

RAM Usage. The genome index is loaded in memory, and lookups are performed while running through the read set. To ensure good performance the entire genome index must fit in RAM. For each seed of weight w we construct an index that consists of a table with 4^w entries (if $w > 12$, SHRiMP2 provides the option to hash the resulting k-mers before indexing them, which can be thought of as using weight $w = 12$). Each entry of the table contains a pointer to a list of locations of the seed on the genome. An index is constructed for every seed, so indexing a genome of length n with k seeds of weight w requires: $k \cdot (4^w \cdot 12 + n \cdot 4)$ bytes. SHRiMP2 provides tools to break the genome into pieces that fit in a target RAM size, as well as to integrate the resulting mappings.

Projecting the Reads. With the genome index loaded in memory, several threads are used to map the reads in parallel. Each read is projected using the spaced seeds, and the genome locations where those k-mers appear are looked up in the index. These k-mers are the matching diagonals in the matrix where the genome is laid out on the x axis and the read on the y axis.

Generating Candidate Mapping Locations (CMLs). Given a length and a score, the list of matching diagonals is scanned for genomic windows of the given length where an alignment with the given score (between the read and the genome) can be constructed from diagonals alone. A CML is generated for every such window. This process is analogous to q-gram filters (Rasmussen *et al.*, 2006).

Paired Mapping Mode. In this mode, the reads in every pair are analyzed and mapped together: a CML for one is analyzed only if a CML for the other exists such that the size of the genomic insert between the two falls within a specified allowable range.

*to whom correspondence should be addressed

© Oxford University Press 2010.

		(0,0)		(2,0)		(4,0)		(0,3)		(2,3)		(4,3)	
	Program	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
50 paired	SHRiMP	98.7	93.6	98.8	92.8	98.6	87.3	98.6	90.9	98.0	83.9	97.0	68.3
	SHRiMP-f	99.1	93.8	99.1	86.8	98.5	58.9	98.7	82.3	98.0	58.8	96.3	30.2
	BFAST	97.9	92.3	96.5	85.4	93.7	63.1	95.9	78.7	94.2	61.5	90.0	38.6
	Bowtie	95.8	45.4	95.2	1.0	100.0	0.0	96.8	1.2	100.0	0.0	0.0	0.0
	BWA	86.0	60.4	78.6	4.7	39.1	0.0	82.6	3.8	65.5	0.2	0.0	0.0
50 unpaired	SHRiMP	99.2	88.0	98.2	86.5	95.9	69.4	97.5	85.0	94.4	66.8	87.0	24.4
	SHRiMP-f	99.5	88.1	98.0	78.1	92.8	26.9	97.6	76.2	92.9	30.6	83.8	4.9
	BFAST	98.9	88.1	96.7	79.7	90.9	48.7	96.3	76.6	92.4	51.2	84.2	23.4
	Bowtie	63.1	63.1	0.3	0.3	0.0	0.0	2.1	2.1	0.0	0.0	0.0	0.0
	BWA	91.7	76.7	73.6	2.0	32.2	0.0	87.2	5.3	75.9	0.1	0.0	0.0
75 paired	SHRiMP	98.0	94.8	98.2	94.8	98.3	94.7	97.8	94.5	98.2	94.4	98.0	92.8
	SHRiMP-f	98.6	95.5	98.8	94.9	98.8	89.1	98.7	94.9	98.7	90.5	98.2	75.8
	BFAST	98.5	96.0	98.2	95.6	97.6	93.6	97.8	94.9	97.4	93.8	96.6	88.9
	Bowtie	97.4	18.3	97.8	0.2	0.0	0.0	100.0	0.2	0.0	0.0	0.0	0.0
	BWA	91.3	59.2	87.4	8.0	43.5	0.1	87.6	6.8	69.9	0.3	18.2	0.0
75 unpaired	SHRiMP	98.9	91.4	98.6	91.2	98.3	90.2	98.4	91.2	97.8	89.7	96.8	76.7
	SHRiMP-f	99.3	91.7	99.0	90.7	98.4	75.0	98.9	90.9	98.0	76.0	95.4	29.1
	BFAST	99.2	92.3	98.6	91.5	97.4	88.3	98.3	91.6	97.4	88.9	95.4	79.1
	Bowtie	40.9	40.9	0.1	0.1	0.0	0.0	0.8	0.8	0.0	0.0	0.0	0.0
	BWA	94.8	76.2	92.5	6.3	60.0	0.0	94.1	9.5	85.6	0.1	0.0	0.0

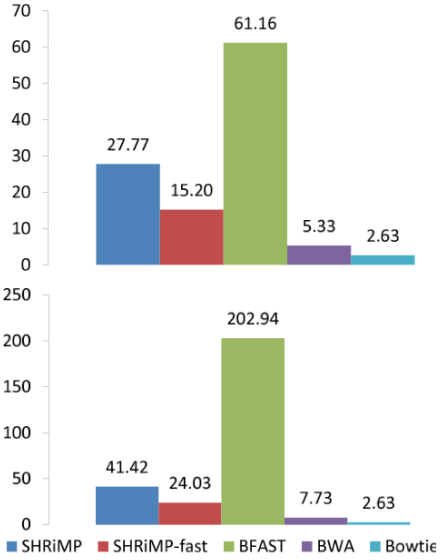

Fig. 1. **A** Precision and recall in 4 regimes, by polymorphism. (A, B): reads containing A SNPs, one indel of size B, and errors. **B** Running times (in minutes) of each tool on $6 \cdot 10^6$ unpaired 50bp reads (top) and paired 75bp reads (bottom), while utilizing 6 cores of an 8 core 3Ghz Xeon machine with 16Gb RAM.

Smith-Waterman Alignment. The CMLs for each read (or pair) are investigated by the standard Smith-Waterman (SW) string matching algorithm (Smith and Waterman, 1981), which helps obtain high sensitivity to indels. For SOLiD reads we align the genome to the four possible "translations" of the read, thus allowing for sequencing errors (see Rumble *et al.* (2009); Homer *et al.* (2009)). SHRiMP2 also uses a caching heuristic to speed up the alignment of reads from repetitive regions: after alignment, we compute a cryptographic hash of the target region, and store it together with the score. Before starting a SW, we first check if an identical region has already been aligned, and if so just reuse the score.

3 RESULTS & DISCUSSION

We compared SHRiMP2 to three other leading read mapping programs: BFAST (Homer *et al.*, 2009), BOWTIE (Langmead *et al.*, 2009), and BWA (Li and Durbin, 2009). We generated 2 datasets, each containing 6,000,000 paired color-space reads, of 50 and 75bp, respectively, simulated from the *Ciona.savignyi* genome (180Mb). The reads contain variants (SNPs and indels), as well as sequencing errors distributed according to typical (non-uniform) error profiles of the SOLiD machine (4% average per-color error rate).

A read (pair) is mapped "uniquely" if of all mappings for that read (pair), the one with the highest score is unique. A uniquely mapped read (pair), is mapped "correctly" if the mapping with the highest score is $\pm 5bp$ of the location where the read (or both reads in the pair) was simulated from. We define recall as the fraction of all reads (pairs) that are mapped correctly and precision as the fraction of all uniquely mapped reads (pairs) that are mapped correctly. In Figure 1A we present precision and recall of each algorithm, and in Figure 1B we demonstrate the runtimes for two of the datasets.

Of all the other short read mapping programs, we found that BFAST is the only one directly comparable to SHRiMP2 in providing high sensitivity even for highly polymorphic reads, practical speed, and wealth of features. In our tests, SHRiMP2 achieves similar or better sensitivity for all polymorphism classes, at about one fourth the running time of BFAST. While we include BOWTIE and BWA in the comparison, these programs primarily target speed, and do not match the sensitivity of SHRiMP2 or BFAST for highly polymorphic reads. We also include SHRiMP2 in "fast" mode (SHRiMP-f; seeds of weight 16 vs. 12, SW score thresholds 60/68 vs. 50/55). At these parameters SHRiMP2 is one order of magnitude slower than BOWTIE and BWA, but achieves significantly better sensitivity for various polymorphism classes.

ACKNOWLEDGEMENTS

Funding: SHRiMP development is suppored by MITACS, CIHR, and Life Technologies research grants to MB. Matei David is supported by NSF grant CCF-0832797.

REFERENCES

Homer, N., Merriman, B., and Nelson, S. F. (2009). Bfast: An alignment tool for large scale genome resequencing. *PLoS ONE*, **4**(11), e7767.

Ilie, L. and Ilie, S. (2007). Multiple spaced seeds for homology search. *Bioinformatics*, **23**(22), 2969–2977.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome Biology*, **10**(3), R25.

Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, **25**(14), 1754–1760.

Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*.

Rasmussen, K. R., Stoye, J., and Myers, E. W. (2006). Efficient q-gram filters for finding all e-matches over a given length. *J. of Computational Biology*, **13**, 296–308.

Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., and Brudno, M. (2009). Shrimp: Accurate mapping of short color-space reads. *PLoS Comput Biol*, **5**(5), e1000386.

Smith, T. F. and Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**(1), 195 – 197.