BIOINFORMATICS APPLICATIONS NOTE Vol. 27 no. 7 2011, pages 1011–1012
doi:10.1093/bioinformatics/btr046

Genome analysis

Advance Access publication January 28, 2011

SHRiMP2: Sensitive yet Practical Short Read Mapping

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

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

Associate Editor: Alfonso Valencia

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

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

Received on September 26, 2010; revised on December 23, 2010; accepted on January 24, 2011

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 previous versions. SHRiMP2 supports Fasta and Fastq input, SAM output, Illumina/Solexa, Roche/454 and AB/SOLiD reads, a paired mapping mode, parameters for miRNA mapping, and parallel computation.

2 METHODS

SHRiMP2 indexes 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 [similar to other read mappers, e.g. Langmead *et al.* (2009); Li and Durbin (2009); Wu and Nacu (2010)] resulted in a dramatic speed increase and further allowed us to add a paired mapping mode and utilize multi-threaded computation. For more details on the methods described below, see the original SHRiMP paper (Rumble *et al.*, 2009), as well as the supplement.

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) k-mer. 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 RAM, and lookups are performed while running through the read set. The index of a genome of length n with k seeds of weight w takes $k \times (4^w \times 12 + n \times 4)$ bytes. With the default parameters ($k=4$, $w=12$), the index of the human genome (hg19) takes 48 GB. SHRiMP2 provides tools to break a genome into pieces that fit in a target RAM size. The overhead introduced by splitting is insignificant: as demonstrated in the supplementary Material, using one node with 16 GB and a 4-way split of hg19 versus one node with 32 GB and a 2-way split results in $\sim$2% slowdown.

Projecting the Reads: 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 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 two diagonals, and a CML is generated for every such window. This process is analogous to q-gram filters (Rasmussen *et al.*, 2006). The CML generation step is one of the major differences between BFAST and SHRiMP: while BFAST uses a larger number of long seeds, and generates CMLs based on a single seed match, SHRiMP2 (and the original SHRiMP) requires multiple seed matches between the read and the reference. This allows for the effective use of seeds with smaller weight and length, and improves sensitivity.

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 within a specified range of the first. A 'rescue' mode is available to re-map pairs with anomalous spacing.

Smith–Waterman Alignment: The CMLs are eventually investigated by the Smith–Waterman (SW) string matching algorithm (Smith and Waterman, 1981), Similar to the original version of SHRiMP, SHRiMP2 supports full alignment (with indels) of both letter space and color-space data. For SOLiD reads we align the genome to the four possible 'translations' of the read, thus allowing for sequencing errors (see Homer *et al.*, 2009; Rumble *et al.*, 2009).

SHRiMP2 uses a caching heuristic to speed up the alignment of reads from repetitive regions: after alignment, we compute a hash of the target region, and store it together with the score. Before starting a SW, we first

*To whom correspondence should be addressed.

A

		(0,0)		(1,0)		(2,0)		(4,0)		(0,3)		(1,3)		(2,3)		(4,3)	
	Program	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.	Prec.	Recl.
50 paired	SHRiMP	99.7	96.6	99.6	96.4	99.6	95.7	99.3	89.3	99.3	93.5	99.3	90.6	98.6	85.7	97.6	69.7
	BFAST	95.4	93.8	94.3	91.6	92.6	86.2	87.0	63.5	91.6	78.8	89.3	71.8	86.8	61.9	80.7	38.8
	BWA	91.1	65.2	85.4	27.7	64.7	5.4	17.7	0.3	62.0	4.4	49.2	1.5	29.6	0.4	11.9	0.1
	Bowtie	97.5	46.6	97.5	11.1	96.9	1.0	0.0	0.0	97.1	1.3	100	0.2	100	0.0	0.0	0.0
75 paired	SHRiMP	99.6	97.5	99.6	97.2	99.6	97.3	99.6	96.9	99.3	96.6	99.5	96.9	99.4	96.5	99.2	94.5
	BFAST	97.4	97.1	97.1	96.8	96.8	96.5	95.9	94.5	96.4	96.0	96.0	95.5	95.9	94.8	94.1	89.5
	BWA	93.2	62.3	86.5	30.2	68.2	8.8	14.7	0.4	65.0	7.5	41.5	2.2	22.4	0.6	11.7	0.1
	Bowtie	98.1	18.1	98.4	2.6	96.2	0.1	100	0.0	97.1	0.5	100	0.0	0.0	0.0	0.0	0.0
50 single	SHRiMP	99.7	93.3	98.9	92.6	98.0	91.1	94.8	72.5	97.0	89.5	95.3	83.5	93.0	69.6	83.4	25.6
	BFAST	98.9	93.0	97.9	90.5	96.2	83.7	87.7	50.7	95.2	80.4	92.8	68.7	89.0	53.5	78.0	24.6
	BWA	95.3	79.7	93.0	33.7	71.8	2.1	15.2	0.0	89.5	5.6	83.7	1.1	61.9	0.1	0.0	0.0
	Bowtie	95.2	65.5	92.1	15.7	49.1	0.3	2.5	0.0	92.1	2.2	85.4	0.4	36.8	0.0	0.0	0.0
75 single	SHRiMP	99.7	96.0	99.6	95.8	99.4	95.6	98.9	94.4	99.2	95.5	98.8	94.9	98.5	93.7	97.2	79.7
	BFAST	99.3	96.0	99.1	95.6	98.8	95.1	97.4	91.6	98.5	95.1	98.0	94.1	97.4	92.1	94.3	81.6
	BWA	97.5	78.2	97.0	38.0	95.1	6.5	56.4	0.0	96.7	9.4	94.6	1.2	90.4	0.2	100	0.0
	Bowtie	97.4	42.0	96.2	6.0	75.7	0.1	0.0	0.0	95.8	0.8	96.3	0.1	100	0.0	0.0	0.0

B

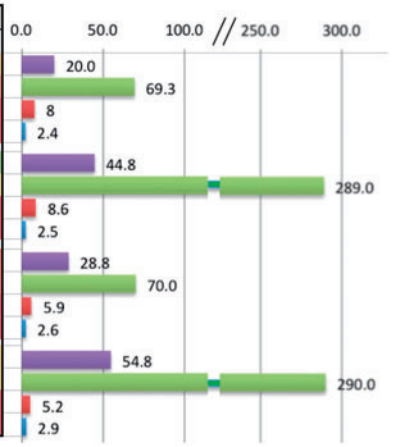

Fig. 1. (**A**) Precision and recall by amount of variation for 4 datasets, by polymorphism: (number of SNPs, Indel size). (X, Y): reads/read pairs containing X SNPs, where the larges indel is of size Y, as well as errors. (**B**) Running times (in min) of each tool on 6×10^6 reads from each dataset while utilizing an 8 core 3.0 GHz Intel Xeon machine with 16 GB RAM.

check if an identical region has already been aligned, and if so just reuse the score.

3 RESULTS AND 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 75 bp, respectively, simulated from the human chromosome 1. 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). We mapped both datasets as both paired and single-end reads.

A read (pair) is mapped 'uniquely' if the mapping with the highest score is unique. This mapping is 'correct' if it is within 10 bp 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 each tool on 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, with a running time that is 2–5 times faster than 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 evaluated the speed of SHRiMP2 on real AB SOLiD data. We estimate that a 30× coverage of hg19 by unpaired 50 bp color space reads can be mapped by 20 nodes, each with 8 cores and 16 GB of RAM, in 3 days. For details, see the Supplementary Material.

Funding: SHRiMP development is suppored by MITACS, CIHR, and Life Technologies research grants to M.B. NSF grant (CCF-0832797) to M. David.

Conflict of Interest: none declared.

REFERENCES

Homer,N. *et al*. (2009) Bfast: an alignment tool for large scale genome resequencing. *PLoS ONE*, **4**, e7767.

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

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

Rasmussen,K.R. *et al*. (2006) Efficient q-gram filters for finding all e-matches over a given length. *J. Computat. Biol.*, **13**, 296–308.

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

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

Wu,T.D. and Nacu,S. (2010) Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, **26**, 873–881.