BIOINFORMATICS APPLICATIONS NOTE
Vol. 27 no. 17 2011, pages 2435–2436
doi:10.1093/bioinformatics/btr394

Sequence analysis
Advance Access publication June 30, 2011

MethylCoder: software pipeline for bisulfite-treated sequences

Brent Pedersen*, Tzung-Fu Hsieh, Christian Ibarra and Robert L. Fischer
Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720, USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: MethylCoder is a software program that generates per-base methylation data given a set of bisulfite-treated reads. It provides the option to use either of two existing short-read aligners, each with different strengths. It accounts for soft-masked alignments and overlapping paired-end reads. MethylCoder outputs data in text and binary formats in addition to the final alignment in SAM format, so that common high-throughput sequencing tools can be used on the resulting output. It is more flexible than existing software and competitive in terms of speed and memory use.

Availability: MethylCoder requires only a python interpreter and a C compiler to run. Extensive documentation and the full source code are available under the MIT license at: https://github.com/brentp/methylcode.

Contact: bpederse@gmail.com

Received on April 15, 2011; revised on June 6, 2011; accepted on June 24, 2011

1 INTRODUCTION

Whole-genome bisulfite sequencing (BS-Seq) is used to determine the methylation levels at each cytosine. In DNA that has been treated with bisulfite, unmethylated cytosines are converted to uracil and then to thymine following PCR amplification. Methylated cytosines are not converted; so an epigenetic mark is visible in the genetic code.

In order to determine nucleotide-resolution methylation levels, bisulfite-treated reads must be re-aligned to the reference genome. Since some of the cytosines (C) in each read may have been converted to thymines (T), direct alignment is not possible because conversions will appear as mismatches. Our approach is to artificially (*in silico*) convert remaining Cs to Ts in the reads and the reference genome. A short-read aligner can then be used to map the converted reads to the converted genome. For each alignment, the original read and reference sequence can be recovered. A reference C that is aligned to a T in a read can be considered as converted, or unmethylated.

Previous software for aligning BS-Seq data used a variety of methods (see Huss, 2010 for a more complete list). Bisulfite sequence mapping program (Xi and Li, 2009) uses a bit mask (that treats Cs and Ts identically) and hashing to map bisulfite-treated reads. Earlier tools such as CokusAlignment (Cokus *et al.*, 2008) use strategies that are slower than current aligners. BS-Seeker (Chen *et al.*, 2010) and Bismark (Kreuger and Andrews, 2011) use a similar strategy to the one described here. BS-Seeker is memory intensive, and does not support paired-end reads. Both Bismark and BS-Seeker are limited to the bowtie aligner and do not support color space reads. Bisulfite-treated reads analysis tool (BRAT; Harris *et al.*, 2009) also uses a hashing approach and is the only other aligner that avoids double-counting overlapping paired-end reads.

We introduce MethylCoder, a fast, memory-efficient BS-Seq pipeline. It supports both paired- and single-end reads in color space or nucleotide formats. MethylCoder provides a single entry point and common output formats for the bowtie (Langmead *et al.*, 2009) and genomic short-read nucleotide alignment program (GSNAP) (Wu and Nacu, 2010) aligners. Each of these aligners has different strengths; GSNAP has no limitation on the size of the reference, but does not consider quality information with the reads. Bowtie can only map to references <4 Gb in total length, but considers quality and can map color space reads. Utilizing these short-read aligners, while providing access to their arguments, ensures that MethylCoder is flexible and accessible.

As with aligners for genomic DNA, there are many viable choices available for BS-Seq mapping. We introduce a resource to benchmark the time, memory use and number of mapped reads for the BS-Seq aligners mentioned above.

2 IMPLEMENTATION

2.1 Preparing the reads and reference

For bowtie, MethylCoder automatically creates a new reference FASTA with two sequences for each sequence in the original FASTA: the first has the Cs converted to Ts and the headers are prefixed with ‘f’ for forward; the second is reverse complemented, then C-to-T converted and the headers are prefixed with ‘r’ for reverse (this step is not necessary for GSNAP, since it handles the potential C-to-T mappings internally). MethylCoder then calls the aligner to do the indexing on the new FASTA. For bowtie, if the reads are in color space, then the index is automatically created in color space. The indexing steps are performed only once; in subsequent runs, MethylCoder will check the timestamp of the index relative to the reference and only re-index if the reference has been updated. This timestamp checking is performed throughout the MethylCoder pipeline to allow the analysis to be re-run without performing unnecessary steps.

If bowtie is specified as the aligner, MethylCoder converts all cytosines to thymine (C-T) in the FASTQ or FASTA reads and saves to a new file. During the conversion, the read files are indexed for random access by creating a custom disk-based index using a BerkeleyDB (Olson *et al.*, 1999) key-value store. Given a header, this store returns the integer location in the file where that record occurs. Since the original file and the C-T converted have the same number of characters per record, a single index works for the original and the C-to-T converted files. This is used later to recover both the

*To whom correspondence should be addressed.

original read and the converted read given only the header in the alignment output without keeping all reads in memory.

2.2 Alignment and tabulation

MethylCoder includes a few default arguments to the aligners. Other arguments for the aligner can be specified explicitly. For example, we find it best to only consider uniquely mapped reads and each aligner has arguments that specify this.

Upon completion of the alignment, the output is saved to a SAM file. If bowtie was specified as the aligner, MethylCoder can attempt to align any remaining, unmapped reads in their original form to the original genome (neither has *in silico* C-to-T conversion). This recovers some highly methylated alignments that would otherwise map to multiple places in the converted genome.

Following the alignment by GSNAP, methylation is tabulated by extracting the reference sequence to which the read is aligned and comparing it with the aligned sequence reported in the SAM file. If the alignment is to the reverse strand or to the right end of a paired-end read, MethylCoder counts the occurrence of a reference G aligning to a read A (not methylated) or to a read G (methylated). Otherwise, the count is of reference C to read T (not methylated) and to a read C (methylated).

For alignments by bowtie, the process is similar except that MethylCoder must recover the original, non-C-to-T converted read from the disk-based index and do the tabulation relative to the original reference. For each cytosine in the reference to which that read maps, we tabulate the number of Cs and number of Ts that align to it. During this step, MethylCoder avoids double-counting overlapping paired-end reads (which are common due to the read-prep) and disregards parts of the alignment that are soft-masked. Checking each base in every read is computationally intensive so we have implemented this part of MethylCoder in the C programming language to improve performance.

Because of the *in silico* C-to-T conversion, bowtie can align a read C to a genomic T without 'seeing' the mismatch. For each alignment, we check for this case, discount that conversion and increment the number of mismatches. If that new mismatch count exceeds the number allowed in the arguments to MethylCoder, the alignment is discarded. This improves accuracy of the alignments.

2.3 Output

During the tabulation, MethylCoder creates a new SAM file with the reference specified as one of the original chromosomes—not the f/r prefixed references output by the aligner. In addition, the read sequence is reverse complemented and the quality reversed if the read had been aligned to the reversed reference. This SAM file is then directly usable in tools such as samtools (Li *et al.*, 2009).

The C, T summations are reported per reference cytosine in simple text and binary formats. Finally, MethylCoder prints a methylation summary for each chromosome and for the entire reference. A file that documents the exact command used to run MethylCoder is created and the SAM file and the per-base methylation files contain that information in the header. These aid in performing reproducible research.

2.4 Examples and benchmarking

We demonstrate the use of MethylCoder in a full comparative analysis between *Arabidopsis thaliana* endosperm and embryo tissues. The example includes downloading the reads and reference, trimming the reads, running MethylCoder and performing the downstream comparative analysis. After tabulating the methylation within each gene, we find overrepresented gene ontology terms in those genes that are differentially methylated between the two tissue types. The code to reproduce the example, along with a detailed explanation, is available here: https://github.com/brentp/methylcode/blob/master/example/example.rst.

In addition, we have created a benchmarking resource that tracks memory usage, processing time and number of reads mapped for various BS-Seq software packages. Each implementation has its strengths; this will serve as a resource for researchers in making their decision. The resource, available here: https://github.com/brentp /methylcode/tree/master/bench includes the commands to download the software and sequence data, run the benchmark and create a table of the results.

3 CONCLUSIONS

MethylCoder is a novel tool for mapping BS-Seq reads because it allows fast and memory-efficient mapping of single or paired-end reads in FASTQ or FASTA format in both color and nucleotide space—something that no other BS-Seq software allows. It is also unique in correcting for soft-masked reads and overlapping paired-end reads. It provides a common invocation to two short-read aligners. MethylCoder also outputs a simple per-base summary of methylation and a SAM file that can be used by common high-throughput sequencing visualization and analysis software. We have created a benchmark comparing current implementations as a resource for researchers doing BS-Seq. The results show that MethylCoder offers the choice of speed via bowtie, or sensitivity via GSNAP.

Funding: National Institute of Health grant R01-GM069415 to R.L.F.

Conflict of Interest: none declared.

REFERENCES

Chen,P. *et al.* (2010) BS Seeker: precise mapping for bisulfite sequencing. *BMC Bioinformatics*, **11**, 203.

Cokus,S.J. *et al.* (2008) Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning. *Nature*, **452**, 215–219.

Harris,E.Y. *et al.* (2010) BRAT: bisulfite-treated reads analysis tool. *Bioinformatics*, **26**, 572–573.

Huss,M. (2010) Introduction into the analysis of high-throughput-sequencing based epigenome data. *Brief Bioinform,* **11**, 512–523.

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

Kreuger,F. and Andrews,S. (2011) Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics*, **27**, 1571–1572.

Li,H. *et al.* (2009) The Sequence alignment/map (SAM) format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Olson,M. *et al*. (1999) Berkeley DB. *Proc. of the 1999 Summer Usenix Technical Conf.*, June 1999.

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

Xi,Y. and Li,W. (2009) BSMAP: whole genome bisulfite sequence MAPping program. *BMC Bioinformatics*, **10**, 232.