TAMeBS: a sensitive bisulfite-sequencing read mapping tool for DNA methylation analysis

Ruimin Sun[1,2], Ye Tian[1] and Xin Chen[1]

[1]School of Physical and Mathematical Sciences,Nanyang Technological University, Singapore
[2]School of Biological Sciences,Nanyang Technological University, Singapore

Abstract

Cytosine methylation plays an important role in many biological regulation processes. The current gold-standard method for analyzing cytosine methylation is based on sodium bisulfite treatment and high-throughput sequencing technologies. In this paper we introduce a new tool called *TAMeBS* for cytosine methylation analysis using bisulfite sequencing data. It aims to align long bisulfite-treated DNA reads onto a reference genome sequence with high mapping efficiency and estimate the methylation status of each cytosine very accurately. Our approach builds on recent advances in alignment techniques, including bi-directional FM-index, approximate seeds, and the likelihood-ratio scoring matrix which was designed particularly for aligning bisulfite-treated DNA reads. We compared *TAMeBS* with several popular bisulfite-treated read mapping tools on both simulation and real data. Experimental results showed that *TAMeBS* could detect many more uniquely best mapped reads than other tested tools while achieving a good balance between sensitivity and precision. The source code of *TAMeBS* is freely available at `https://sourceforge.net/projects/tamebs/`.

1 Introduction

DNA methylation is one of the most characterized epigenetic modifications of genomes. In eukaryotes, it involves an addition of a methyl group to the 5th carbon residue of a cytosine (mC5). Methylation of cytosines acts as a key factor in

many essential biological processes, including embryonic growth, X chromosome inactivation, genomic imprinting, cancer development in mammals, regulation of gene expression, and transposon silencing in plant cells [8, 19]. Cytosine methylation levels vary significantly in different genomic contexts. Different from the dominant CG methylation in mammalian organisms [18, 10], cytosines can be methylated in all sequence contexts in plants [16]. Thus, studying methylation in plants often requires much more complex analysis than in animals and human.

To determine the genome-wide DNA methylation patterns, the current gold-standard method is based on sodium bisulfite treatment and high-throughput sequencing. Briefly, bisulfite treatment converts unmethylated cytosines into uracils, which are subsequently changed to thymines by DNA polymerase chain reaction (PCR). In contrast, the methylated cytosines remain unchanged after bisulfite treatment (see Figure 1). Such different reactions of methylated and unmethylated cytosines from bisulfite treatment enable us to determine the methylation states by comparing DNA sequences before and after bisulfite treatment [10]. Together with the rapidly-advancing next-generation sequencing (NGS) technologies, we are able to perform genome-wide methylation analysis at the single base-pair resolution at a very low cost. The technique that applies NGS to bisulfite treated DNA sequences is called bisulfite sequencing or BS-Seq for short, and the resulting sequencing reads are then called as BS reads.

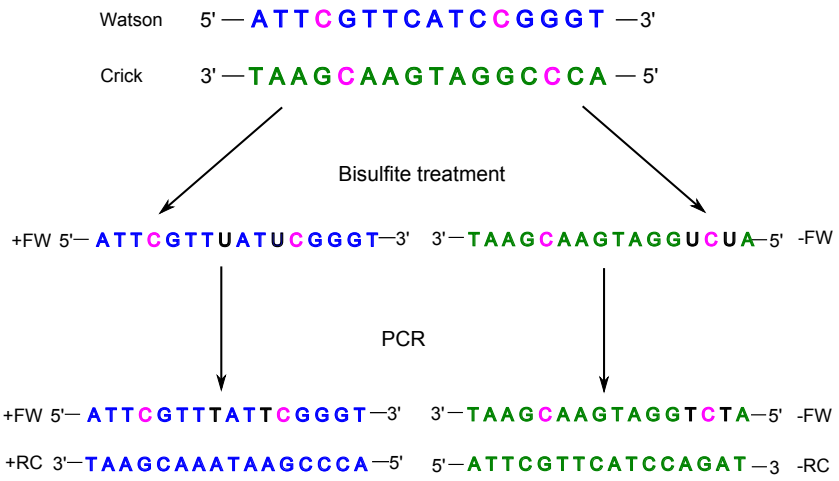

Figure 1: Sodium bisulfite treatment does not affect methylated Cs (in rosy pink) but unmethylated Cs. Thus, each C still left in the sequence after bisulfite treatment implies a cytosine methylation at its genomic location. Since two complementary DNA strands are not symmetric after bisulfite treatment, four different strand sequences may be produced after PCR amplification.

By applying the BS-Seq technique to genome-wide methylation analysis, the

first computational step is always to map a large number of BS reads to a reference genome sequence. While there are many excellent tools available for general sequence alignment tasks, they are often found not satisfactory or convenient when applying to BS read sequences. This is not surprising due to the different characterization of BS read sequences from the general genomic sequences. After the bisulfite treatment, C/T mapping becomes asymmetrical. That is, a base T in a BS read shall be allowed to match both bases C and T in the reference genome, but not vice versa. On the other hand, as two complementary DNA strands usually contain different distributions of mC5's, their converted strands after bisulfite treatment would no longer be complementary with each other. Moreover, there are two different PCR library protocols proposed for producing BS reads: directional protocol and non-directional protocol. The BS reads from directional protocols come only from the two bisulfite-converted DNA strands, i.e., $\pm FW$ strands. However, the BS reads generated by non-directional protocol could come from either $\pm FW$ strands or their reverse complement $\pm RC$ strands (see Figure 1).

In recent years, a number of tools have been developed for aligning the BS-Seq read data. Based on their strategies to deal with the asymmetrical C/T mapping, these aligners can be divided into two broad categories: wild-card aligners and three-letter aligners [1]. Three-letter aligners, such as BS Seeker [4], Bismark [9], and BatMeth [15], choose to convert Cs in both reference genome and BS reads to Ts and then apply some standard alignment tools (see Supplementary Material, available at `https://sourceforge.net/projects/tamebs/`). In comparison, wild-card aligners such as Last [6] do not perform any C-to-T conversion. Instead, they treat Cs in genomic sequences differently—either replace each C in the reference genome with a wild-card letter to match both C and T in reads or define a scoring scheme for C/T mapping. It is worth noting that neither of the above asymmetrical C/T mapping strategy seems perfect. A wild-card aligner can achieve high genomic mapping coverage, but often introduce bias towards methylated Cs in methylation level estimation. The three-letter aligners can align reads with high accuracy, but may miss many uniquely mapped reads due to their reduced alphabet and sequence complexity. Besides, most of these aligners cannot achieve a good balance between sensitivity and accuracy when they are applied to map BS reads containing more mutations.

In this paper we present a new approach to aligning BS reads and estimating methylation state, which is implemented in our software tool called *TAMeBS*. See Figure 2 for a schematic illustration of *TAMeBS*. In brief, *TAMeBS* was built on the bi-directional FM-index data structure which was originally proposed in [12, 14] as well as the classical seed-and-extend sequence alignment scheme. In the initial seeding step, *TAMeBS* proceeds in the same way as a three-letter aligner in order to find as many hits as possible. In the subsequent extension step, it instead adopts the wild-card scheme together with a likelihood-ratio scoring matrix in order to

find sensitive alignments.With the above strategy, *TAMeBS* is capable of not only filtering out many ambiguous alignments from a common three-letter aligner, but also reducing the bias towards methylated cytosines typically incurred by a wild-card aligner. Experimental results show that *TAMeBS* could particularly recover many true alignments that other tools would otherwise have missed.

In the present study we consider only the BS reads produced by Illumina platform [3] using directional protocol. Since the sequencing errors present in Illumina reads are mostly substitutions, we ignore sequencing errors of insertions and deletions in read alignments. In the remaining of this paper, we first introduce the bi-directional FM-index data structure and show how to apply it to find the hits of approximate seeds. Then, we discuss in detail the results of both simulation and real biological experiments and the comparisons against four popular bisulfite-treated read mapping tools.

2 Method

2.1 Bi-directional FM-Index

Briefly, a suffix array (SA) implies all suffixes of a text sorted in lexicographical order. Thus, if a pattern string P occurs in a text T, it gives rise to an interval $[l(P), u(P)]$ in the suffix array of T (denoted as SA_T) such that

$$l(P) = \min \{k : P \text{ is the prefix of } T_{SA_T(k)}\},$$

$$u(P) = \max \{k : P \text{ is the prefix of } T_{SA_T(k)}\},$$

where $T_{SA_T(k)}$ is the suffix starting at $T[SA_T(k)]$ and $SA_T(k)$ is the original position of the kth smallest suffix in T. As suffix array requires a large amount of memory space, a compressible data structure based on Burrows-Wheeler transform (BWT) is often used in place of suffix array [2]. According to [5], the SA interval of pattern P can be computed from the BWT string of the text T very efficiently by performing backward search.

Backward search can deal with exact matching very well. Unfortunately, it becomes inconvenient to find approximate matches, especially when applying to double-strand DNA sequences. To improve the efficiency and flexibility of finding approximate matches, we need BWT-based backward search as well as forward search. For this purpose, bi-directional BWT was firstly introduced in [12] to allow matching to be conducted in both forward and backward directions. To map reads onto the reference genome T with errors allowed, bi-directional BWT uses two SA intervals for T and for the reverse of T, respectively. Further to the work of [12], H. Li proposed a so-called FMD-index [14], which is a single index structure constructed for both forward and reverse-complementary strands of DNA

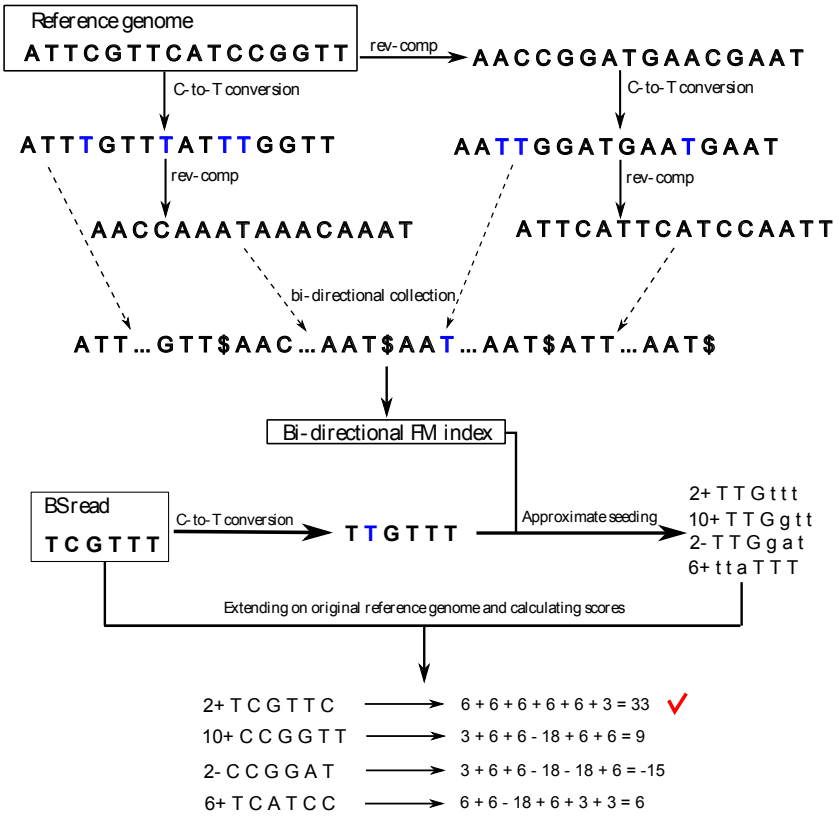

Figure 2: Schematic illustration of *TAMeBS*. Starting from the two complementary strands of a reference genome, we first convert all C's to T's and then further generate their respective reverse complementary strands. The four resulting strands are concatenated into a bi-directional collection, for which we build a bi-directional FM index. After this index-building process, for each BS read, we convert all C's to T's and then find its candidate mapping hits by using an approximate seeding strategy. As shown above, applying seed TTG gives rise to hits at position 2 and 10 on the forward converted strand and position 2 on the reverse-complementary converted strand. Another eligible hit is given by seeding with TTT. Afterwards, hit extensions are performed between the original reference genome sequence and the read sequence without any C-to-T conversions. The full alignments are scored with a likelihood-ratio scoring matrix (Table 1), and those achieving the highest score are finally reported.

sequences. FMD-index uses a *bi-interval* to accomplish the search and matching in two directions. Specifically, a bi-interval consists of three components. If we write the reverse complement of P as $\overline{P}$, then the bi-interval of P is defined as $[l(P), l(\overline{P}), s(P)]$, where $l(P)$ is the left endpoint of the SA interval of P and $s(P)$ is the length of this SA interval. We applied bi-intervals as well as the relevant algorithms in [14] (i.e., Algorithm 2 and 3) to find exact matchings with BS-Seq data.

Given a reference genome T, we construct a *bi-directional collection* by concatenating two C-to-T converted strands of T and their respective reverse complements into one string (see Figure 2), i.e.,

$$\tilde{T} = T^c_+ \circ \$ \circ \overline{T^c}_+ \circ \$ \circ T^c_- \circ \$ \circ \overline{T^c}_- \circ \$$$

where $\circ$ denotes the string concatenation and \$ is a sentinel symbol with the lexicographical order \$ < A < C < G < T. Moreover, T^c_+ and T^c_- represent the C-to-T converted Watson and Crick strands while $\overline{T^c}_+$ and $\overline{T^c}_-$ represent their reverse-complementary strands, respectively. With the FMD-index built on $\tilde{T}$, a bi-interval $[l(P), l(\overline{P}), s(P)]$ contains all the occurrence information of a BS read P on four bisulfite-treated strands of the reference genome T to facilitate fast read alignment.

2.2 Seeding

Seed-and-extension is a classical strategy to approach the sequence alignment problem. As a critical part of this strategy, how to choose seeds has attracted a lot of research attention. In this study, we apply approximate seeds [20] to detect the possible genomic locations of a BS read. In order to achieve higher sensitivity, we applied the C-to-T conversion on each BS read sequence P (see Figure 2). If P is a BS read generated with the directional protocol, we need to execute the mapping procedure (presented below) only once on P after the C-to-T conversion. If P is a BS read generated with the non-directional protocol, we need to execute the mapping procedure twice, one on P after the C-to-T conversion and the other on P after the G-to-A conversion.

2.2.1 Approximate seeds

Given a read P with l bp, a positive integer k, and the reference genome G, the general sequence alignment problem is to find out all the substrings of G that can match P with at most k mismatches (as mentioned earlier, we ignore indels here). To locate the approximate matches of P in G, we partition P into $m = \lfloor k/2 \rfloor + 1$ non-overlapped segments called seeds, as discussed in [20]. Then, we restrict the

length of each seed to be at most $\lceil l/m \rceil$ bp. By the pigeonhole principle, there will be at most $\lfloor k/2 \rfloor$ seeds containing two or more mismatches, if the read P is aligned to a genomic location with at most k mismatches. In other words, if P has an alignment within k mismatches, there must exist at least one seed being matched exactly or with one mismatch. Such a matching is thus called a *hit* of this seed. In general, the seeds with mismatches allowed in their hits are called *approximate seeds*, to distinguish from exact seeds.

To align a read within k mismatches, $(\lfloor k/2 \rfloor + 1)$ approximate seeds can guarantee the full mapping sensitivity, as can $(k + 1)$ exact seeds which are obtained from an equally-spaced partition of the read sequence. However, due to their dramatic difference in length, approximate seeds can achieve mapping specificity significantly higher than exact seeds. It implies that, unless k is very small relative to l, we shall always expect a significantly lower number of the total hits from $(\lfloor k/2 \rfloor + 1)$ approximate seeds than from $(k + 1)$ exact seeds. This in turn reduces the number of hit extensions significantly and the total mapping time as well. We define $r = k/l$ and call it *the mutation rate*. Thus, the larger the mutation rate r, the higher the mapping specificity that approximate seeds improve over exact seeds. This property is particularly useful for fast alignment of BS reads. In this case, both reads and the reference genome are of the reduced alphabet and reduced sequence complexity due to the C-to-T conversion, thereby increasing the number of seed hits. Our experiments on the real BS read data of *Arabidopsis thaliana* show that up to 19 times less hits were generated with approximate seeds than with exact seeds for subsequent extensions (see Supplementary Material).

2.2.2 Seeding with bi-directional index

We denote the seeds of P as $P^{(i)}$, $i = 0, \ldots, m - 1$, ordered according to their starting positions in P, where $m = \lfloor k/2 \rfloor + 1$. For each seed $P^{(i)}$, we conduct bi-directional tests to find its hits (i.e., 1-approximation matchings) in a reference genome G. The forward and backward tests are described in Algorithm 1 and Algorithm 2 in Supplementary Material, respectively. They are developed actually based on the same observation that, for any hit alignment of $P^{(i)}$, either the first half $P^{(i)}[0, \lfloor l_i/2 \rfloor]$ or the second half $P^{(i)}[\lfloor l_i/2 \rfloor, l_i - 1]$ of $P^{(i)}$ shall be exactly matched, where l_i is the length of $P^{(i)}$. They also work in a quite similar way. For instance, the forward test starts by searching for exact matchings in the forward direction, from which we would obtain one of the following three possible outcomes:

1. Exact matchings of the whole seed $P^{(i)}$ are returned.

2. The exact matching process halts before the middle position $\lfloor l_i/2 \rfloor$ is reached.

3. The exact matching process passed the middle position $\lfloor l_i/2 \rfloor$ but fails to reach the right end.

The outcome 1) implies that we have found all the hits of $P^{(i)}$ which are the exact matchings. When it occurs, we proceed to the hit extension step directly without bothering to search for other hits that involve mismatches. In this way it will significantly speed up the read alignment process. However, the mapping sensitivity might be sacrificed, but not much, because there is little chance that the seed $P^{(i)}$ aligns to the true genomic location with exactly one mismatch and, at the same time, to some other genomic location without any mismatch. If the outcome 2) occurs, we stop the current test in the forward direction and then proceed to the test in the backward direction. In case of the outcome 3), we resume the exact matching process at the middle position in order to find the hits of $P^{(i)}$ with one mismatch occurring in the second half. The exact matching process is then recursively branched to accommodate a mismatch at each subsequent position. A branching process is terminated once two mismatches are met. For each branching process finally reaching the right end, we would obtain a bi-interval that gives rise to a set of hits of $P^{(i)}$.

Our method for finding the hits of approximate seeds as described above is different from the one implemented in Masai [20]. Unlike ours, the method in Masai essentially conducts approximate matching of seeds only in one direction. With bi-directional tests we require an exact matching of either the first half or the second half of the seed, which provides highly efficient filtering of spurious hits and thus reduces the total seeding time, as already argued in [13].

2.3 Hit extension

In this step, we aim to extend the seed hits into full alignments of reads. Different from the previous seeding process, our hit extensions are performed between the original read sequences and the reference genome sequence without any C-to-T conversion (see Figure 2). It would enable us to penalize a mapping of a genomic T against a read C which shall be considered as a mismatch. To further detect sensitive alignments, we use log likelihood ratios to score alignments in the same way as many traditional sequence alignment methods have done. In [6], a statistical model is proposed particularly for aligning BS reads to a reference genome sequence, from which a log likelihood-ratio scoring matrix is thus estimated (see Table 1). We also used this matrix in our tool *TAMeBS* to score alignments.

Table 1: The likelihood-ratio scoring matrix (from [6]) used in *TAMeBS*.

		Genome			
		A	C	G	T
Read	A	6	−18	−18	−18
	C	−18	6	−18	−18
	G	−18	−18	6	−18
	T	−18	3	−18	3

2.4 Implementation details

We developed a software tool called *TAMeBS*, implemented in C++ language, to align BS reads by making use of bi-directional FM-index, approximate seeds, and the likelihood-ratio scoring matrix as discussed above. We further extended *TAMeBS* to estimate methylation distributions from the aligned BS read data.

There are three components implemented in *TAMeBS*: bi-directional FM-index building, seed-and-extension read mapping, and methylation calling. In the index building component, *TAMeBS* constructs the suffix array as well as BWT of a bi-directional collection $\tilde{T}$ using a modified version of SA-IS algorithm [17]. In order to trade off between the memory space and running time, only part of suffix array is stored via sampling. In our implementation, one entry of the suffix array would be retained every 32 entries. Other entry elements will be computed out using the BWT whenever needed.

In the read mapping component, we align BS reads to the reference genome one after another. For each read, we first convert all Cs to Ts if it contains any. Then we find all the 1-approximation hits of its seeds by using the bi-directional tests as described in Section B. For each hit found, we extend it to a full alignment of the read up to k mismatches. In order to improve sensitivity as well as accuracy, the likelihood-ratio matrix shown in Table 1 is used to score alignments where no C-to-T conversion is made in both read and genome sequences. For the sake of faster alignment, if a seed has too many hits (exceeding a preset threshold B), all these hits will be thrown away and thus excluded from further extension. The threshold value of B should depend on the seed length. The longer a seed, the smaller the threshold value of B. In *TAMeBS*, B is empirically set to be $\lceil 2000/length(seed) \rceil$ (note that the seed length varies with the mutation rate r). We say an alignment is *best* if it contains at most k mismatches while achieving the highest mapping score. And, an alignment is called *uniquely best* if it is the only best alignment. *TAMeBS* can report uniquely best, any best and all best alignments of a read depending on users' choices. By default, the uniquely best alignments are reported.

In the methylation calling component, *TAMeBS* infers cytosine methylation

from the read alignment results and produces two files. The first tabular file contains the methylation status of every base C in the reference genome, and the second file summarizes the distributions of methylated cytosines in different genomic contexts.

3 Results and Discussion

Our experiments below use *Arabidopsis thaliana* (*A.thaliana*) as the reference genome. It comprises five chromosomes with a total of about 119 million base pairs. The genome sequence was downloaded from the NCBI database at `http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi?taxid=3702`.

All experiments were run on a Linux server with processor Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHz and RAM 32GB. We compared the performance of *TAMeBS* with four popular BS read alignment tools, BS Seeker, Bismark, Last and BatMeth. For methylation estimation, we chose to compare it with Bismark only, as Bismark was previously shown capable of achieving very high quantitative accuracy [11]. All tested tools applied their own default parameter settings except for those error-related settings. Specially for BS Seeker and Bismark, we tried different values for their error-related parameters and used the best results for performance comparison (Supplementary Material).

3.1 Evaluation metrics

Note that all the tested tools report uniquely best alignments by default. Previous studies [4, 15] used *mapping efficiency* for performance evaluation, which refers to the percentage of reads that can be uniquely mapped by a tool. On simulated datasets, we know where each read originates from in the reference genome. In this case, a read is considered *correctly aligned* if it is aligned to its original genomic location. We hence define *sensitivity* as the percentage of total reads that are correctly aligned and *precision* as the percentage of aligned reads that are correctly aligned. A single evaluation metric, called *F-measure*, is defined as the harmonic mean of sensitivity and precision. That is,

$$F\text{-}measure = \frac{2 * precision * sensitivity}{precision + sensitivity}.$$

The F-measure is intended to evaluate the overall performance of a tool. In general, the higher the F-measure score, the better the alignment performance.

3.2 Evaluation on simulated data

We simulated three datasets, each of which contains 11 million 100-bp reads. All reads were randomly generated from the reference genome *A.thaliana*. The three datasets contain three, five, and seven mismatches per read, respectively (*i.e.*, with the mutation rate r of 3%, 5%, and 7%, respectively). In order to comprehensively evaluate mapping capabilities, each dataset comprises 11 subsets generated from the reference genome with different proportions of methylated Cs, ranging from 0% to 100% by increment of 10%. All subsets are of the same size, *i.e.*, each containing one million reads. We sampled mC5's uniformly according to a given methylation percentage, regardless of their genomic contexts. Furthermore, we followed [6] to set the bisulfite conversion rate as 99%.

The detailed experimental results of the three datasets for five BS read mapping tools are summarized in Table 2. All the tested tools performed very well at the low mutation rate $r = 3\%$. Their mapping efficiencies and F-measures can generally achieve up to 95% and 97%, respectively, while *TAMeBS* achieved the highest. At the medium mutation rate $r = 5\%$, Bismark and BS Seeker failed to obtain satisfactory results as their mapping efficiencies dropped dramatically below 80%. We believe that it is mainly due to their built-in aligner Bowtie, which allows to map efficiently only reads with a limited number ($\leq$3) of mutations at the 5' end. Again *TAMeBS* achieved the highest mapping efficiency and F-measure (95.78% and 97.64%), improving over the second best aligner BatMeth by 2.57% and 1.15%, respectively. At the high mutation rate $r = 7\%$, the mapping efficiency of BatMeth dropped below 24%, which means that BatMeth failed to align more than 76% reads. In contrast, *TAMeBS* and Last remained the high mapping efficiency as well as the high F-measure score. Compared with Last, *TAMeBS* uniquely mapped 3.2% more reads with an improved F-measure score of 1.48%. Considering the CPU time used by each mapping tool, *TAMeBS* ran comparatively fast with other tools at both the low and medium mutation rates, but several times slower at the high mutation rate. We expected this relatively low time efficiency of *TAMeBS*, as it was aimed mainly at achieving high mapping efficiency and accuracy (in terms of F-measure) for the subsequent accurate methylation estimation analysis.

We noticed that the proportion of methylated cytosines in real plant genomes is estimated between 5% and 25% [8]. In order to compare the alignment performances of these tools under more realistic setup, we simulated another eight datasets by using the BS read simulator Sherman (`http://www.bioinformatics.bbsrc.ac.uk/projects/sherman/`). Each dataset contained one million 100-bp reads generated from *A.thaliana* with bisulfite conversion rate of 90%. In Sherman, the bisulfite conversion rate is defined as the percentage of C's converted to T's regardless of their genomic contexts. Thus, the proportion of methylated C's

Table 2: Average performances of five BS read mapping tools on three simulated data sets. MapEff, Sens, prec, and F-ms represent the mapping efficiency, sensitivity, precision and F-measure, respectively. When the mutation rate increased to 5%, Bismark and BS Seeker cannot uniquely align over 80% reads. When the mutation rate further increased to 7%, only *TAMeBS* and Last successfully mapped more than 90% reads. In comparison, BatMeth mapped less than 24% reads.

r	Tools	MapEff (%)	Sens (%)	Prec (%)	F-ms (%)	CPU time (m:s)
3%	*TAMeBS*	95.85	95.74	99.88	97.76	03 : 58
	Last	92.42	92.36	99.93	96.00	06 : 57
	BatMeth	94.24	94.23	100.00	97.03	01 : 09
	Bismark	95.50	95.46	99.90	97.63	05 : 24
	BS Seeker	95.66	95.54	99.87	97.65	09 : 43
5%	*TAMeBS*	95.78	95.58	99.79	97.64	08 : 02
	Last	92.19	92.11	99.91	95.85	07 : 24
	BatMeth	93.21	93.22	100.00	96.49	04 : 01
	Bismark	79.51	79.29	99.72	88.34	04 : 38
	BS Seeker	79.74	39.64	49.71	44.11	08 : 00
7%	*TAMeBS*	95.11	94.80	99.67	97.17	19 : 38
	Last	91.91	91.82	99.90	95.69	07 : 12
	BatMeth	23.49	23.47	99.78	38.01	04 : 12
	Bismark	30.48	30.27	99.33	46.40	01 : 57
	BS Seeker	75.33	37.24	49.44	37.24	14 : 16

in this simulation setup was 10%. In addition, these eight datasets contained 0 to 7 SNPs per read, respectively. Figure 3 (or Figure S1 in Supplementary Material) depicts the mapping performance of each tool. Similar to the above simulation results, *TAMeBS* achieved the highest sensitivity and mapping efficiency on almost all test datasets. It also offered the best balance between sensitivity and precision among all the compared mapping tools. Although having the highest precision on the datasets with less than six SNPs per read, BatMeth achieved the sensitivity significantly lower than *TAMeBS* and thus the worse mapping performance in terms of F-measure. We further compared the output alignments of *TAMeBS* with those of Last and BatMeth. We found that more than 99.97% of the read alignments output by BatMeth were also found by *TAMeBS* on the first six datasets. And, approximately 98.62% of the alignments output by Last were also found by *TAMeBS*. These experimental results clearly show that *TAMeBS* can achieve very high mapping efficiency and sensitivity at a small cost of precision.

3.3 Evaluation on biological data

To evaluate our tool on real biological data, we downloaded about 25M paired-end reads from the NCBI Sequence Read Archive (SRA). The SRA accession number is ERR046546, and the reads were sequenced from the *A. thaliana* genome by

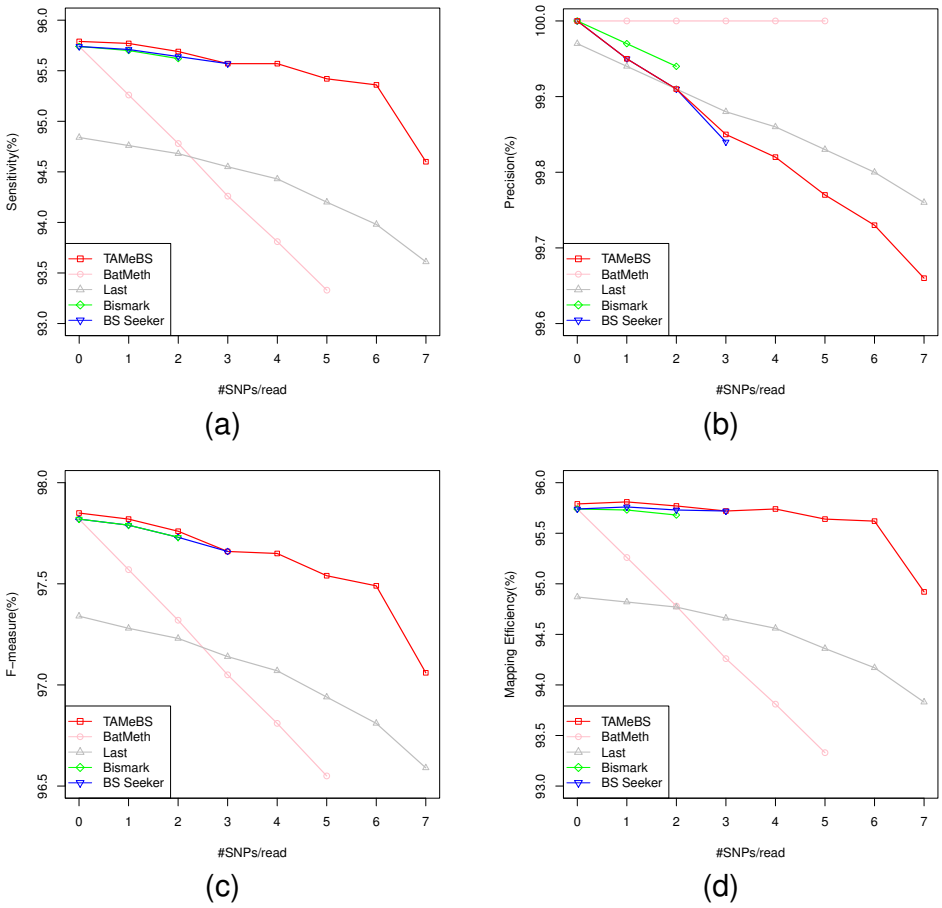

Figure 3: Simulation results of the eight datasets generated by Sherman. Only *TAMeBS* and Last obtained good enough mapping performance on all eight sets of data. For the other four tools, we presented their mapping results only when their mapping efficiency were over 50%.

using Illumina Genome Analyzer IIx. As the current implementation of *TAMeBS* takes only single-end reads as input, we chose to align the first read of each pair in our first experiment. According to our observation, the first base is 'N' for most reads. Thus, we cut the first base off from each read. At the end, we extracted 10 millions 100-bp single-end reads to construct a test dataset. The mapping efficiency and running time of each mapping tool is summarized in Table 3. We note that BatMeth consumed the least CPU time, but reported the fewest uniquely best alignments (<38%). With the parameter setting $k = 5$, *TAMeBS* achieved not only higher mapping efficiency than Last (57.05% vs 56.96%) but also higher mapping speed. With $k = 7$, *TAMeBS* achieved even higher mapping efficiency (57.48%) at the expense of long CPU time (>2h).

Table 3: Mapping 10M 100-bp single-end reads extracted from ERR046546 to the *A. thaliana* genome.

Software	Mapping Efficiency(%)	CPU Time(h:m:s)
TAMeBS($k = 3$)	55.64	00 : 50 : 10
TAMeBS($k = 5$)	57.05	01 : 19 : 13
TAMeBS($k = 7$)	57.48	02 : 27 : 14
BatMeth($n = 3$)	37.81	00 : 26 : 03
BatMeth($n = 5$)	37.05	00 : 53 : 15
BS Seeker($e = 50, m = 3$)	53.43	01 : 00 : 57
Last	56.96	01 : 36 : 11
Bismark($n = 3, l = 36$)	54.62	00 : 41 : 35

For the reads in this biological dataset, their original genomic locations are unknown. In order to further evaluate the mapping sensitivity, we performed exhaustive search for all the reads that can be uniquely mapped onto the reference genome within 3 mismatches. The mapping sensitivity is thus defined as the percentage of those uniquely mapped reads that would be returned by a mapping tool. *TAMeBS* with $k = 3$ achieved the highest mapping sensitivity at 99.80%, indicating that it found almost all the unique best alignments within 3 mismatches (see Supplementary Material Table S3). We conducted experiments on the whole 25M single-end reads as well and obtained the similar results. All these experimental results are presented in Supplement Material.

3.4 Methylation estimation

To evaluate the performance of a mapping tool in methylation estimation, we calculate *the methylation percentage* as the number of methylated calls divided by the total number of methylated and unmethylated calls from the read alignment result and then show how close it is from the *true* methylation percentage in the simulation study. Besides the overall methylation percentage, we are also interested in the absolute numbers of methylated and unmethylated cytosines in genome called by a mapping tool. A cytosine's methylation status is not called when there is no read aligned to it.

Again, we used Sherman to simulate a dataset of BS reads from the reference genome *A.thaliana* under a more realistic scenario. This large-scale dataset consisted of four subsets, each of which contained one million 100bp-reads at a fixed mutation rate (0, 1, 3, and 5 SNPs per read for four subsets, respectively). Moreover, we set the bisulfite conversion rates for cytosines in CG and non-CG contexts based on a previously reported distribution of methylated cytosines in *A.thaliana*, which are 60% and 20%, respectively [16]. The performance of *TAMeBS* and

Bismark in estimating the genome-wide methylation percentage along with the mapping performance can be found in Supplementary Material. In particular, *TAMeBS* reported the same methylation percentages as Bismark in all genomic contexts. It is worth noting that, as a previous study has demonstrated, Bismark could generally achieve very high quantitative accuracy in estimating methylation percentages [11]. Thus, *TAMeBS* is able to perform accurate methylation estimation as well.

We further compared the absolute numbers of cytosines in genome called by these two tools as methylated or unmethylated. *TAMeBS* successfully called about 76% cytosines in the reference genome, whereas Bismark called only 48% cytosines. It means that there were more than half of cytosines in the genome for which Bismark did not find any read alignment. Moreover, we found that 83% of cytosines called by Bismark were also called by *TAMeBS*. To be specific, *TAMeBS* called about 17 million more cytosines than Bismark (among a total of about 42 million cytosines in the genome). This superior performance of *TAMeBS* shall be mainly attributed to its high mapping efficiency (95.8%) and F-measure (97.8%) in the previous alignment procedure. We believe that a high methylation calling rate of cytosines in genome is vital to many other applications such as the genome-wide detection of differentially methylated regions [7].

4 Conclusion

In this paper we introduced a new BS read mapping tool called *TAMeBS* for DNA methylation analysis. It aims to align long bisulfite-treated reads onto a reference genome sequence with high mapping efficiency and sensitivity so that the methylation status of each genomic cytosine can be accurately estimated. To this end, we built *TAMeBS* on several recent advances in sequence alignment techniques, including bi-directional FM-index, approximate seeds, and the likelihood-ratio scoring matrix which is designed particularly for aligning bisulfite-treated DNA reads. In both simulated and real data experiments, *TAMeBS* demonstrated its strong ability to detect more uniquely mapped reads than other tested tools while retaining a good balance between mapping sensitivity and precision. Moreover, *TAMeBS* achieved comparably high accuracy in methylation percentage estimation with the existing mapping tool Bismark. However, *TAMeBS* could determine the methylation status for many more cytosines in genome than Bismark. It is a feature that many subsequent analyses shall find beneficial.

We noticed that *TAMeBS* required much more running time and memory than other mapping tools as the mutation rate increased. Although it shall not be a big issue that prevents *TAMeBS* from running, we are currently working on code optimization in order to reduce its computing cost.

Acknowledgment

We would like to thank the anonymous referees for their valuable comments. This work was partially supported by the Singapore Ministry of Education Academic Research Fund (MOE2012-T2-1-055) and the Singapore National Medical Research Council grant (CBRG11nov091).

Reference

References

[1] Christoph Bock. Analysing and interpreting DNA methlation data. *Nat Rev Genet*, (10):705–719, October 2012.

[2] M. Burrows and D. J. Wheer. A block-sorting lossless data compression algorithm. Technical report, Digital Systems Research Center, 130 Lytton Avenue, Palo Alto, California, 1994.

[3] Aniruddha Chatterjee, Peter A Stockwell, Euan J Rodger, and Ian M Morison. Comparison of alignment software for genome-wide bisulphite sequence data. *Nucl Acids Res*, 40(10):e79, May 2012.

[4] Pao-Yang Chen, Shawn J Cokus, and Matteo Pellegrini. Bs seeker: precise mapping for bisulfite sequencing. *BMC Bioinformatics*, 11:203, April 2010.

[5] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *FOCS'00 Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 390. IEEE Computer Society Washington, DC, USA, 2000.

[6] Martin C. Frith, Ryota Mori, and Kiyoshi Asai. A mostly traditional approach improves alignment of bisulfite-converted DNA. *Nucl Acids Res*, 40(13):e100, March 2012.

[7] Kasper D Hansen, Benjamin Langmead, and Rafael A Irizarry. Bsmooth: from whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biology*, 13:R83, October 2012.

[8] Tomas J Hardcastle. High-throughput sequencing of cytosine methylation in plant DNA. *Plant Methods*, page 16, June 2013.

[9] Felix Krueger and Simon R. Andrews. Bismark: A flexible aligner and methylation caller for bisulfite-seq applications. *Bioinformatics*, 27(11):1571–1572, April 2011.

[10] Felix Krueger, Benjamin Kreck, Andre Franke, and Simon R Andrews. DNA methylome analysis using short bisulfite sequencing data. *Nat Methods*, 9(2):145–151, January 2012.

[11] Covindarajan Kunde-Ramamoorthy, Cristian Coarfa, Eleonora Laritsky, Noah J. Kessler, R. Alan Harris, Mingchu Xu, Rui Chen, Lanlan Shen, Aleksandar Milosavljevic, and Robert A. Waterland. Comparison and quantitative verification of mapping algorithms for whole-genome bisulfite sequencing. *Nucl Acids Res*, (6):e43, January 2014.

[12] T. W. Lam, R. Li, A. Tam, S. Wong, E. Wu, and S. M. Yiu. High throughput short read alignment via bi-directional BWT. In *BIBM'09*, pages 31–36, November 2009.

[13] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, (3):R25, March 2009.

[14] Heng Li. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics*, 28(14):1838–1844, July 2012.

[15] Jing-Quan Lim, Chandana Tennakoon, Guoliang Li, Eleanor Wong, Yijun Ruan, Chia-Lin Wei, and Wing-Kin Sung. BatMeth: improved mapper for bisulfite sequencing reads on DNA methylation. *Genome Biol*, 13(10):R82, October 2012.

[16] Ryan Lister, Ronan C. O'Malley, Julian Toni-Filippini, Brian D. Gregory, Charles C. Berry, A. Harvey Millar, and Joseph R. Ecker. Highly integrated single-base resolution maps of the epigenome in Arabidopsis. *Cell*, (3):523–536, August 2008.

[17] Ge Nong, Sen Zhang, and Wai Hong Chan. Two efficient algorithms for linear suffix array construction. *Computers, IEEE Transactions*, (10):1471–1484, October 2011.

[18] Mattia Pelizzola and Joseph R. Ecker. The DNA methylome. *FEBS Letters*, 585(13):1994–2000, July 2010.

[19] Keith D. Robertson. DNA methylation and human disease. *Nat Rev Genet*, 6(8):597–610, August 2005.

[20] Enrico Siragusa, David Weese, and Knut Reinert. Fast and accurate read mapping with approximate seeds and multiple backtracking. *Nucl Acids Res*, 41(7):e78, January 2013.