# STATISTICAL METHODS AND ANALYSES FOR NEXT-GENERATION SEQUENCING DATA

**by**

**XIAOQING YU**

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Dissertation Advisor: Dr. Shuying Sun

Department of Epidemiology and Biostatistics

**CASE WESTERN RESERVE UNIVERSITY**

August, 2014

**CASE WESTERN RESERVE UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

We hereby approve the thesis/dissertation of

**Xiaoqing Yu**

candidate for the degree of **Ph.D.** *.

Committee Chair

**Xiaofeng Zhu**

Committee Member

**Shuying Sun**

Committee Member

**Robert Elston**

Committee Member

**Nathan Morris**

Committee Member

**Jing Li**

Date of Defense

**June 6, 2014**

*We also certify that written approval has been obtained

for any proprietary material contained therein.

Table of C ontents

List of Tables

List of Figures

## Acknowledgements

First of all, I would like to express my sincere gratitude to my research advisor Dr. Shuying Sun, for her continuous guidance and support throughout my Ph.D study and research work, for her patience, motivation, enthusiasm, and immense knowledge. Her advice on both research as well as on my career has been invaluable. I feel truly fortunate to have had tremendous support from Dr. Xiaofeng Zhu, who has helped me overcome many crisis situations. I also want to acknowledge Dr. Robert Elston and Dr. Nathan Morris for their thoughtful comments and constructive suggestions on my thesis research. I am grateful to Dr. Jing Li for serving on the committee and providing valuable advice.

My gratitude is also extended to the faculty, staff, and fellow students in the Department of Epidemiology and Biostatistics, especially Sudha Iyengar, Catherine Stein, Rob Igo, Ralph O'Brien, Omar De La Cruz Cabrera, Tao Feng, Xiangqing Sun, Alberto Santana, Yanina Natanzon, Heming Wang, and Victor Courtney. I also want to thank Case Comprehensive Cancer Center for the financial support.

I thank all my friends at CWRU for their sincere friendship and moral support, which have made me feel warm in the long winter of Cleveland.

Finally, I would like to dedicate this dissertation to my dear parents, for their unconditional love and support throughout my life.

# Statistical Methods and Analyses for Next-generation Sequencing Data

## Abstract

## by

**XIAOQING YU**

The advent of next-generation sequencing (NGS) technologies has significantly advanced sequence-based genomic research and biomedical applications. Although a wide range of statistical methods and tools have been subsequently developed to support the analysis of NGS data in different steps and aspects, challenges continue to arise due to multiple issues. The central theme of this dissertation is to address the challenges and issues in three aspects of NGS analyses: sequencing alignment, Single Nucleotide Polymorphism (SNP) detection, and differential methylation identification.

First, to investigate issues of low sequencing quality and repetitive reads in alignment, four commonly used alignment algorithms (SOAP2, Bowtie, BWA, and Novoalign) have been thoroughly reviewed and evaluated. The results show that the concordance among the algorithms is relatively low in reads with low sequencing quality, but can be substantially improved by trimming off low quality bases before alignment. As for aligning reads from repetitive regions, the simulation analysis shows that reads from repetitive regions tend to be aligned incorrectly, and suppressing reads with multiple hits can improve alignment accuracy significantly. Second, to address the challenges in SNP detection caused by low coverage, four SNP calling algorithms (SOAPsnp, Atlas-SNP2, SAMtools, and GATK) have been compared and evaluated in a low-coverage single-sample sequencing dataset.

Although the four algorithms have low agreement, GATK and Atlas-SNP2 show relatively higher calling rates and sensitivity than others programs. Third, a new hidden Markov model-based approach, HMM-DM, has been developed to identify differentially methylated regions (DMRs) in bisulfite sequencing data. This method well accounts for the large within group variation of methylation levels and can detect differential methylation in single-base resolution. It has been demonstrated to have superior performance compared with BSmooth, and its application has been illustrated using a real sequencing dataset. In the last part of this thesis, five DMR identification methods (methylKit, BSmooth, BiSeq, HMM-DM, and HMM-Fisher) have been systematically reviewed and compared using bisulfite sequencing datasets. All five methods show higher accuracy in the identification of simulated DMRs that are relatively long and have small within group variation. Compared with the three other methods, HMM-DM and HMM-Fisher yield relatively higher sensitivity and lower false positive rates, especially in DMRs with large within group variation. However, in the real data analysis, the five methods show low concordances, probably due to the different approaches they are taking when tackling the issues in DMR identification. Therefore, to guarantee a higher accuracy in validation and further analysis, users may choose the identified DMRs that are long and have small within group variation as a priority. In summary, this thesis has addressed several important questions in NGS studies through the development of new statistical methods and comprehensive bioinformatic analyses.

# CHAPTER 1: INTRODUCTION AND SPECIFIC AIMS

## 1.1 Introduction

Since 1977, the Sanger method (Sanger, et al., 1977) has been applied to many large-scale sequencing projects, and is considered to be the "gold standard" in terms of both read length and sequencing accuracy (Bonetta, 2006). However, low capacity and high cost have prohibited its application in sequence-based biomedical research. To meet the great demand for more efficient and cost-effective sequencing, high-throughput sequencing technologies have been developed from automated Sanger sequencing to next-generation sequencing (NGS) over the past decade. Currently, next-generation sequencing technologies, including Roche/454, Illumina, SOLiD, and Helicos, are able to produce giga base-pairs of data per machine day (Metzker, 2010) with affordable cost. This rapid development of a new sequencing technologies substantially extends the scale and resolution of many sequence-based genomic studies (Mardis, 2008), including the assembly of new genome (Flicek and Birney, 2009), variation discovery (Dalca and Brudno, 2010), quantitative analysis of the transcriptome (RNA-seq) (Mortazavi, et al., 2008), identification of regulatory protein binding sites (ChIP-seq) (Johnson, et al., 2007; Park, 2009), and the identification of genome-wide methylation patterns (methyl-seq) (Brunner, et al., 2009; Cokus, et al., 2008). In response to the influx of new sequencing methods, many statistical and computational tools have been developed to support different steps of NGS data analysis, including data preprocessing, alignment, variant identification, DNA methylation studies, RNA-seq and ChIP-seq analyses. However, challenges arise from different aspects, such as the enormous number of reads, high complexity of data, and

sequencing errors. In this chapter, I will address the challenges in three aspects of NGS analyses: sequencing alignment, SNP calling, and differential methylation identification.

## 1.2 Specific Aims

### Specific Aim 1: Investigating the performance of alignment algorithms using sequencing data with varying qualities and from repetitive regions

With the rapid growth of new sequencing technologies, many alignment programs have been developed. These programs serve as relatively efficient and accurate tools in aligning a large number of reads, and greatly extend the applications of sequencing technologies. However, new challenges for alignments have arisen from applying sequencing technologies to address different biological questions. For example, how do reads with various sequencing qualities affect alignment results? How to deal with the reads that can be mapped to multiple locations on a reference genome? In order to investigate these questions, I propose to use both real sequencing data and simulated data with the above issues to evaluate the performances of four commonly used algorithms: SOAP2 (Li, et al., 2009), Bowtie (Langmead, et al., 2009), BWA (Li and Durbin, 2010), and Novoalign (http://www.novocraft.com), as shown below.

In order to evaluate how different alignment programs perform for sequencing reads with low quality ends, the four alignment algorithms will be used to map two datasets with different sequencing qualities, before and after trimming off the low quality bases. The performance of the four aligners will be measured in terms of concordance between any pair of aligners.

In order to evaluate how different alignment algorithms perform on data containing reads generated from regions with more repetitive sequences, the four aligners will be applied to simulated datasets with repetitive reads and to evaluate their performances by calculating false alignment rates.

The research work related to the above aim 1 will be shown in Chapter 2.

**Specific Aim 2: Studying SNP calling algorithms for their performance on low coverage sequencing data**

Many SNP calling programs have been developed to identify Single Nucleotide Variants (SNVs) in next-generation sequencing data. However, low sequencing coverage presents challenges to accurate SNV identification, especially in single-sample data. Moreover, commonly used SNP calling programs usually include several metrics in their output files for each potential SNP. These metrics are highly correlated in complex patterns, making it extremely difficult to select SNPs for further experimental validation. To address this issue, I propose to study the performance of four SNP calling algorithms, SOAPsnp (Li, et al., 2009), Atlas-SNP2 (Shen, et al., 2010), SAMtools (Deng, et al., 2009), and GATK (DePristo, et al., 2011; McKenna, et al., 2010), using a low-coverage single-sample sequencing dataset.

First**,** the key metrics reported in the output files of these four algorithms will be studied and used as filtering criteria to filter out low quality SNVs. Second, the results from the four algorithms will be compared with different coverage cutoffs, calculating the empirical sensitivity and calling rates. Third, suggestions will be provided for efficient and accurate SNV calling using a single-sample low-coverage sequencing dataset.

The research work related to the above aim 2 will be shown in Chapter 3.

**Specific Aim 3: Identifying differential methylations using a hidden Markov model-based approach**

It is very important and challenging to develop statistical methods to identify differential methylation in bisulfite sequencing data. First, differential methylation patterns are known to be associated with cancer initiation and progression, and they may be used as biomarkers in diagnosis. Accurate identification of differential methylation patterns will assist the development of cancer diagnosis, prognosis, and therapeutics. Second, although bisulfite-treatment and NGS technologies have extended methylation studies to a whole-genome single-base resolution, challenges arise due to sequencing errors and unknown methylation patterns along each chromosome (Eckhardt, et al., 2006). Third, there are limited available statistical methods for methylation sequencing data and most of them are designed for a specific sequencing protocol and do not account for the within group variation of methylation levels. Therefore, a more efficient and generally applicable method for methylation sequencing data is needed. To meet this need, I propose a Bayesian approach based on a hidden Markov model (HMM-DM) to identify differential methylation regions between any two biological conditions.

The proposed statistical method will first use a hidden Markov model to identify differentially methylated sites accounting for the similar pattern of nearby CG sites along a chromosome and the within group variation, then summarize the identified sites into regions. In the HMM step, three hidden states will be set as hypermethylated in group 1, equally methylated in both groups, and hypomethylated in group 1. The observations will

be methylation ratios of samples from the two groups. The emission probabilities will be modeled using a Beta distribution for each state. All key parameters will be estimated from the data directly. The proposed HMM method will be compared with BSmooth using simulated bisulfite sequencing data, and its application will be illustrated through a real breast cancer dataset.

The research work related to the above aim 3 will be shown in Chapter 4.

**Specific Aim 4: Comparing statistical methods in DMR identification using bisulfite sequencing data**

Over the last several years, several methods have been developed to identify differentially methylated regions in bisulfite sequencing data. In this aim, I propose a comprehensive comparison analysis of five DMR identification methods: methylKit (Akalin, et al., 2012), BSmooth (Hansen, et al., 2012), BiSeq (Hebestreit, et al., 2013), HMM-DM (Yu and Sun, 2014), and HMM-Fisher (Sun and Yu, 2014).

First, the features of all five methods will be reviewed and summarized with respect to several analytical aspects. Second, the effect of parameter settings in DMR identification will be studied for these five approaches. Third, using a simulated dataset, their performance will be evaluated by calculating the sensitivity and false positive rates of identified differential methylated CG sites. Fourth, in a real bisulfite treated methylation sequencing dataset, the differentially methylated CG sites identified by all five methods will be compared. Different methods' performance of estimating methylation levels will be evaluated as well.

The research work related to the above aim 4 will be shown in Chapter 5.

## References


Akalin, A., *et al.* (2012) methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles, *Genome biology*, **13**, R87.

Bonetta, L. (2006) Genome sequencing in the fast lane, *Nature Methods*, **3**, 141-147.

Brunner, A.L., *et al.* (2009) Distinct DNA methylation patterns characterize differentiated human embryonic stem cells and developing human fetal liver, *Genome Research*, **19**, 1044-1056.

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

Dalca, A.V. and Brudno, M. (2010) Genome variation discovery with high-throughput sequencing data, *Briefings in Bioinformatics*, **11**, 3-14.

Deng, J., *et al.* (2009) Targeted bisulfite sequencing reveals changes in DNA methylation associated with nuclear reprogramming, *Nature Biotechnology*, **27**, 353-360.

DePristo, M.A., *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature Genetics*, **43**, 491-498.

Eckhardt, F., *et al.* (2006) DNA methylation profiling of human chromosomes 6, 20 and 22, *Nature Genetics*, **38**, 1378-1385.

Flicek, P. and Birney, E. (2009) Sense from sequence reads: methods for alignment and assembly, *Nature Methods*, **6**, S6-S12.

Hansen, K.D, Langmead, B. and Irizarry, R. (2012) BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions, *Genome biology*, **13**, R83.

Hebestreit, K., Dugas, M. and Klein, H.-U. (2013) Detection of significantly differentially methylated regions in targeted bisulfite sequencing data, *Bioinformatics*, **29**, 1647-1653.

Johnson, D.S., *et al.* (2007) Genome-Wide Mapping of in Vivo Protein-DNA Interactions, *Science*, **316**, 1497-1502.

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

Li, H. and Durbin, R. (2010) Fast and accurate long-read alignment with Burrows–Wheeler transform, *Bioinformatics*, **26**, 589-595.

Li, R., *et al.* (2009) SNP detection for massively parallel whole-genome resequencing, *Genome Research*, **19**, 1124-1132.

Li, R., *et al.* (2009) SOAP2: an improved ultrafast tool for short read alignment, *Bioinformatics*, **25**, 1966-1967.

Mardis, E.R. (2008) The impact of next-generation sequencing technology on genetics, *Trends in Genetics*, **24**, 133-141.

McKenna, A., Hanna, M. and Banks, E. (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data, *Genome Research*, **20**, 1297-1303.

Metzker, M.L. (2010) Sequencing technologies — the next generation, *Nature Reviews Genetics*, **11**, 31-46.

Mortazavi, A., *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq, *Nature Methods*, **5**, 621-628.

Park, P.J. (2009) ChIP-seq: advantages and challenges of a maturing technology, *Nature Reviews Genetics*, **10**, 669-680.

Sanger, F., Nicklen, S. and Coulson, A.R. (1977) DNA sequencing with chain-terminating inhibitors, *Proceedings of the National Academy of Sciences*, **74**, 5463-5467.

Shen, Y., *et al.* (2010) A SNP discovery method to assess variant allele probability from next-generation resequencing data, *Genome Research*, **20**, 273-280.

Sun, S. and Yu, X. (2014) HMM-Fisher: a hidden Markov Model-based method for identifying differential methylation, *Manuscript in preparation*.

Yu, X. and Sun, S. (2014) HMM-DM: identifying differentially methylated regions using a Hidden Markov model, *Manuscript submitted for publication*.

# CHAPTER 2: HOW DO ALIGNMENT PROGRAMS PERFORM ON SEQUENCING DATA WITH VARYING QUALITIES AND FROM REPETITIVE REGIONS?

## 2.1 Introduction

The great demand for efficient, inexpensive, and accurate sequencing has driven the development of high-throughput sequencing technologies from automated Sanger sequencing to next-generation sequencing (NGS) over the past several years. Currently, NGS technologies are capable of producing low-cost data on a gigabase-pair scale in a single run, which usually includes millions of sequencing reads. This ability makes the NGS technology a powerful platform for various biological applications, such as genetic variant detection by whole-genome or target region resequencing, mRNA and miRNA profiling, whole transcriptome sequencing, ChIP-seq, RIP-seq, and DNA methylation studies. The first step of nearly all these applications is to align sequencing reads onto a reference genome. Thus, in order to obtain any further genetic information from sequencing data, the requirement of fast and accurate alignment tools has to be a priority (Li and Homer, 2010).

In parallel with the rapid growth of new sequencing technologies, many alignment programs (Alkan, et al., 2009; Chen, et al., 2009; De Bona, et al., 2008; Hach, et al., 2010; Harris, et al., 2010; Jiang and Wong, 2008; Lam, et al., 2008; Langmead, et al., 2009; Li and Durbin, 2009; Li and Durbin, 2010; Li, et al., 2008; Li, et al., 2008; Li, et al., 2009; Lin, et al., 2008; Ma, et al., 2002; Ning, et al., 2001; Rumble, et al., 2009;

Schatz, 2009; Weese, et al., 2009) have been developed, including MAQ, Novoalign (www.novocraft.com), SOAP, Bowtie, and BWA. Among all these five aligners, MAQ is the only one that indexes the reads, while all other aligners build indexes on a reference genome. In terms of the indexing algorithms they adopt, MAQ and Novoalign are two alignment programs that build an index with a hash table. To identify inexact matches in short-read alignments, MAQ uses a split strategy while Novoalign adopts an alignment scoring system based on the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970). SOAP2 employs a similar split strategy as MAQ in identification of inexact matches. Instead of using a hash table, SOAP2 adopts the FM-index algorithm (Ferragina and Manzini, 2000) to build an index, which greatly reduces the alignment time for substrings with multiple identical copies. Bowtie and BWA are two other alignment programs developed based on the FM-index method that uses a backtracking strategy to search for inexact matches. These programs serve as relatively efficient and accurate tools for aligning a large number of reads, and greatly extend the scale and resolution of sequencing technology applications.

New challenges for alignments have arisen from applying sequencing technologies to address different biological questions. For example, how do reads with various sequencing qualities affect alignment results? How do they deal with the reads that can be mapped to multiple locations on a reference genome? In order to answer these questions, we select four commonly used aligners (SOAP2, Bowtie, BWA, and Novoalign), and conduct a systematic analysis to evaluate the performance of these programs. First, we review and compare the algorithms these alignment programs employ as well as their advantages with respect to the major options they provide. Then, we use two sets of real

Illumina sequencing data and two sets of simulated data to study how the different alignment programs perform on sequencing data with varying quality and from repetitive regions. The performance is measured in terms of 1) concordance between any pair of the aligners, and 2) accuracy in simulated read alignment. We demonstrate that, for sequencing data with reads that have relatively good quality or have had the low quality bases trimmed off, all four alignment programs perform similarly. Furthermore, we show that trimming off low quality ends markedly increases the number of aligned reads and improves the consistency among the different aligners as well, especially for low quality data. However, Novoalign is more sensitive to the improvement of data quality. As for aligning reads from repetitive regions, our simulated data show that reads from repetitive regions tend to be aligned incorrectly, and suppressing reads with multiple hits can improve alignment accuracy.

## 2.2 Methods

### 2.2.1 Reviewing the features of alignment programs

Hash Table and suffix tree are two major indexing algorithms that current alignment programs use. Hash Table indexing, which was first introduced into the field of alignment by BLAST (Altschul, et al., 1990), keeps the positions of k-mer query subsequence as keys, and then searches for the exact match of the keys in reference sequences. It consumes less space since it builds an index for positions of sequences instead of the sequences themselves. Among different suffix tree algorithms, FM-index is based on the Burrows-Wheeler transforms (BWT) (Burrows and Wheeler, 1994). BWT is a reversible permutation of characters in a text. It transforms the original character string

into a more compressed format, where the same characters are placed side by side as a cluster, rather than in a scatter pattern. Out of the four alignment programs we are interested in, Novoalign adopts a hash Table algorithm, while SOAP2, Bowtie, and BWA adopt the FM-index (Table 2-1).

To find inexact matches, alignment programs allow a certain number of mismatches using different strategies (Table 2-1). SOAP2 uses a split-read strategy to allow at most two mismatches. A read will be split into three fragments, such that the mismatches can exist, at most, in two of the three fragments at the same time. Bowtie uses a backtracking strategy to perform a depth-first search through the entire space, which stops when the first alignment that satisfies a specific criterion is found (Langmead, et al., 2009). Similar to Bowtie, BWA also adopts a backtracking strategy to search for inexact matches. However, the search in BWA is bounded by a lower limit of the number of mismatches in the reads. With this limit better estimated, BWA is able to define a smaller search space, and thus make the algorithm more efficient (Li and Durbin, 2009). Moreover, BWA provides a mapping quality score for each read to indicate the Phred-scaled probability of the alignment being incorrect. This mapping quality score incorporates base qualities, number of mismatches, and the repeat structure. The higher the mapping quality score, the more accurate an alignment is. A zero will be assigned if a read is aligned to at least two locations with equal probabilities. On the other hand, Novoalign first finds candidate alignment positions from the reference genome for each read, and calculates alignment scores for these positions using the Needleman-Wunsch algorithm, based on base qualities, the existence of gap, and ambiguous codes (Ns). This alignment score is $-10\log_{10}(q)$, where $q$ represents the probability of observing the read sequence given a

specific alignment location. This score corresponds to the parameter setting "-t" at the command line when running Novoalign, which finds the best alignment with the lower score and any other alignments with similar scores. Because of this alignment-score-based search algorithm, users cannot define the number of allowed mismatches in each alignment, but they can set up a threshold of alignment scores.

We also summarize the major options that the four alignment programs provide (Table 2-2). SOAP2, Bowtie, BWA, and Novoalign all allow pair-end alignments, enable the identification of the best alignment, and incorporate certain ways of trimming low quality bases (Table 2-2). There are some characteristics unique to certain aligners. For example, in BWA, the maximum number of allowed mismatches is sensitive to the length of reads. If less than 4% of *m*-long reads with 2% uniform base error rate have more than *k* mismatches, then the maximum number of allowed mismatches in these reads is set to be *k*. Thus, for our simulated data with 50-bp-long reads, *k*=3. For the real NGS data with 68-bp-long reads, *k*=4 (Table 2-2). This number may vary depending on the length of reads after trimming. Unlike the other programs, Novoalign does not allow users to define the number of allowed mismatches. However, this parameter can be set indirectly by defining the threshold of the alignment score. In practice, setting the threshold at '-t 60' will be approximately equivalent to allowing two mismatches at high quality base positions and maybe one mismatch at a low quality position.

### 2.2.2 Alignment performance evaluation

#### *2.2.2.1 Datasets*

In order to examine how the four selected alignment programs (i.e., SOAP2, Bowtie, BWA, and Novoalign) perform on real sequencing data with varying quality, we use two single-end Illumina sequencing datasets (S1 and S2). S1 and S2 are sequenced from human colon cancer samples. For each of these two samples, about 3000 exons selected from cancer related genes are captured and sequenced by the Illumina sequencer, with 7,406,247 and 5,398,566 68-bp-long single-end reads generated respectively. We process the reads with the ShortRead package inside of Bioconductor (http://www.bioconductor.org) to evaluate the quality for each single base. The plot of base qualities suggests that S1 has an overall better quality than S2. For example, S2 has more low quality bases at the 3' end. In particular, the last 10 bases have average quality score less than 20 (Figure 2-1).

In order to examine how the four selected alignment programs perform on sequencing data obtained from repetitive regions, we simulate two sets of data from human genome 18: 1) 138771 50-bp-long reads are generated from about 3000 exon regions from which the real datasets S1 and S2 are generated, and these 3000 exon regions do not have many repetitive regions; 2) 55018 50-bp-long reads are generated from 218 CpG islands. These 218 CpG islands are selected from 28226 CpG islands along the whole genome, by the criteria that each chosen CpG island must have at least 25% repetitive bases, and these repetitive bases must be at least 50bp in length. Note that the purpose of this work is not to simulate reads from all repetitive regions in a genome.

It is difficult to precisely define repetitive regions in a genome. Therefore, we simply choose to use some CpG islands that have some level of repetitive regions.

In order to mimic the situation that one or two single bases are mismatches due to pure sequencing errors or true novel single nucleotide variants (SNV), we design the following four scenarios in our simulation data:

1) randomly set one mismatch for each read and let all bases have high quality;
2) randomly set two mismatches for each read and let all bases have high quality;
3) randomly set one low quality mismatch for each read and let all other bases have high quality;
4) randomly set two low quality mismatches for each read and let all other bases have high quality.

1) and 2) are the cases that the mismatches are likely due to the existence of novel SNVs. 3) and 4) are the cases that the mismatches are likely due to pure sequencing errors. Low quality mistmatahces are assigned with Phred quality score ranging from 5-15; high quality bases are assigned with Phred quality score ranging from 30-40. Phred quality score is defined as $-10*\log_{10}(p)$, where $p$ is the base-calling error probability.

#### *2.2.2.2 Trimming, alignment, and evaluation*

In order to evaluate how different alignment programs perform in sequencing reads with low quality ends, we use the four alignment algorithms to map S1 and S2 before and after trimming off the low quality bases using BRAT trim (Harris, et al., 2010). In particular, we set the parameters of BRAT as trimming from both the 5' and 3' ends until

reaching a base with quality score higher than 20, and allowing at most two Ns in each read. The length of each trimmed read is at least 24 bases, and the majority of them are larger than 50 bases. We then perform alignments against the human genome 18 on trimmed and non-trimmed S1 and S2 data using SOAP2 (Li, et al., 2009), Bowtie (Langmead, et al., 2009), BWA (Li and Durbin, 2010), and Novoalign (www.novocraft.com), respectively. For the purpose of this study, we set the parameters in all four alignment programs as follows: 1) At most two mismatches are allowed in SOAP2, Bowtie, and BWA for each alignment. Owing to the different alignment searching algorithm that Novoalign uses, we set the parameter t at 60 to allow approximately up to two mismatches, and then choose the alignment reads with no more than two mismatches using the NM tag in the output. 2) Randomly report one alignment for each read, or only report reads with unique alignments. For each alignment result we calculate the percentage of aligned reads. The performance of the four alignment programs is measured in terms of concordance between any pair of aligners because no known truth for real sequencing data is available. In particular, for each pair of aligners, aligned reads are assigned into four classes as follows:

Class 1: a read is aligned to the same location by both aligners that we are comparing (e.g., aligner 1 and aligner 2);

Class 2: a read is aligned to different locations by both aligners;

Class 3: a read is only aligned by one of the two aligners (e.g., aligner 1);

Class 4: a read is only aligned by the other aligner in a comparison pair (e.g., aligner 2).

If two alignment algorithms perform similarly, there should be a relatively small number of reads in class 2, 3, and 4 as shown in Figure 2-2.

In order to evaluate how different alignment algorithms perform on data containing reads generated from regions with more repetitive sequences, we use two simulated datasets. One dataset is simulated from the 3000-exon regions that do not have a lot of repetitive bases and the other one is from 218 selected CpG islands that have many repetitive bases. For both simulated datasets, we align these reads using the four selected alignment programs. While aligning these simulated reads, all parameters are set the same as the ones used for the real NGS data, except that we allow one mismatch for those datasets with only one mismatch simulated. Because we know the position from which each simulated read is generated, the performance of the four alignment tools is measured in terms of the accuracy of simulated read alignment. We define a true alignment as a situation when a read is aligned back into the same position from which it was generated. In addition, a false alignment is defined as a read that is aligned to other positions rather than the one from which it was generated.

## 2.3 Results

### 2.3.1 Benchmark of aligners

To assess the speed of index building and read mapping in these four aligners, we use the non-trimmed S1 data, which has 7.4 million 68-bp-long single-end reads. We align these reads using the human genome 18 as a reference, with at most two mismatches allowed and one alignment randomly reported for each read (Table 2-3). Novoalign is extremely fast (4.02 min) at index building, while the other three take more

than one hour to finish the same job (Table 2-3). As for read mapping, SOAP2 and Bowtie have a similar number of reads mapped although SOAP2 takes 6 minutes less than Bowtie. BWA maps 76.12% of all reads, which is slightly more than SOAP2 and Bowtie, within 26.4 minutes. Novoalign, on the other hand, is much more time-consuming. It takes 62.9-minute CPU time to align 73.64% of the reads in single-end mode.

### 2.3.2 Aligners' performance on sequencing data with different qualities

For the dataset S1 that has relatively good quality, all four aligners generally show good concordance, without trimming off low quality bases. A similar number of reads is aligned by each aligner (Table 2-4). Over 95% of the reads are assigned into class 1 (i.e., more than 95% of reads are aligned to the same locations by both aligners) when comparing SOAP2, Bowtie, and BWA, pairwise, while Novoalign shows slightly less agreement (84-88%) with the other three aligners (Table 2-5). However, for the S2 dataset that has very low quality bases at many reads, the comparison results are quite different. In the non-trimmed dataset S2, when Novoalign is compared with any of the other three aligners, less than 50% of the reads are assigned to class 1 (i.e., less than 50% of the reads are aligned to the same locations by both aligners), but 15% of the reads are assigned to class 2 (i.e., 15% of reads are aligned to different locations by two aligners), and over 30% are assigned to classes 3 or 4 in total (i.e., about 30% of reads are aligned by only one of the two aligners), respectively (Table 2-6). That means for Novoalign and any other aligner (SOAP2, Bowtie, or BWA), only 50% of all aligned reads are mapped by both of them. This inconsistency of Novoalign's performance in different datasets might result from the fact that S2 has overall lower quality than the S1 dataset (Figure 2-

1). To further investigate the effect of sequencing quality, we trim both S1 and S2 datasets with BRAT trim, and then do alignment using the four aligners.

Performing trimming on NGS data not only cuts off the low quality bases from both ends, but also discards poor quality reads, and thus improves the reads' quality markedly. After trimming, 399,442 (5.4%) and 204,911 (3.8%) reads are discarded from S1 and S2 data, respectively. With slightly fewer reads available for alignment, however, the number of aligned reads is increased by 15-17% in the S1 data, and 34-42% in the S2 data, for all four-alignment programs. This apparent difference in the magnitude of increase indicates that trimming has a greater effect on the S2 dataset than on the S1 dataset. Another interesting observation is that Novoalign aligns 42% more reads in trimmed S2 than non-trimmed S2, while the increment in the other three aligners is only about 35%, suggesting that data quality improvement has a larger effect on Novoalign.

By trimming off the reads before alignment, we observe a substantial increase in the number of reads that fall into class 1 in all pair-wise comparisons, in both S2 and S1 datasets (Tables 2-5, 2-6, 2-7, and 2-8). That is, more reads are aligned to the same locations by the comparison pair. This increase indicates an improved concordance among the four aligners. Moreover, trimming appears to have a greater effect on S2, a dataset with lower quality, than on the S1 dataset. In the pair-wise comparisons between Novoalign and any of the other three aligners for the S2 dataset, the number of reads assigned to the first class increases almost 3 fold (1.2 million vs. 3.7 million), while the number of reads that are only aligned by the other aligners become markedly less (see class 2 of Table 2-8), compared to non-trimmed alignments (see class 2 of Table 2-6). On the other hand, in the S1 dataset, trimming only improves the agreement between

Novoalign and the other three aligners by 8-10% (Tables 2-5, 2-7). This differentiation in the magnitude of concordance improvement, along with the fact that performing trimming leads to a more significant improvement in reads' quality for S2 dataset, further indicates that Novoalign is more sensitive to changes in sequencing quality.

### 2.3.3 Aligners' performance on reads with multiple alignments

To evaluate these aligners in terms of their performance on reads with multiple alignments, we set the alignment parameters in two different ways: (1) randomly report one alignment for each read and (2) only report the read with a unique position (suppress reads that can be aligned to multiple locations). Compared to the former strategy, the latter discards around 4-10% of aligned reads from S1 and 2.5-8% from S2 (see Table 2-4).

In pair-wise comparisons among all four aligners, we find that in both the S1 and S2 datasets, suppressing multiple alignments decreases the number of reads aligned to different positions (class 2) in all comparison pairs, while the number of reads aligned to same positions (class 1) stays the same (Tables 2-5, 2-6, 2-7, 2-8). Reads with multiple alignments are more likely to be aligned to different locations by different aligners, due to the difference in alignment strategies these aligners employ, as well as the strategy of randomly choosing one alignment to report. Therefore, the number of reads assigned to class 2 in any comparison is reduced by suppressing multiple alignments. Next, we will use simulated data to investigate this further.

### 2.3.4 Aligners' performance on simulated data

In order to study the four aligners' performance on reads from repetitive regions, we use the two sets of simulated data as mentioned in the Dataset subsection. One dataset is simulated from 3000 exon regions that do not have many repetitive bases. The other dataset is from 218 selected CpG islands that have a lot of repetitive bases. In these two simulated datasets, no matter whether the mismatch positions are designed to have high or low quality, all four aligners show a lower false alignment rate in the dataset generated from 3000 exon regions (0.7-5%, see Table 2-9A, B) compared to the dataset generated from 218 CpG islands that have more repetitive regions (14-17%, see Table 2-10A, B). Since the reads from regions with repetitive bases have a much higher probability of being aligned onto multiple locations, we can predict that suppressing multiple hits can help diminish the false alignments caused by repetitive bases. As expected, the alignment accuracy in CpG island simulation data is substantially improved by suppressing multiple alignments (Table 2-10A, B).

By assigning mismatches with high quality, we mimic the true novel variants that are more likely to have better quality. By assigning mismatches with low quality, we mimic pure sequencing errors. In both cases, SOAP2, Bowtie, and BWA are found to have similar false alignment rates, no matter whether the alignment report is randomly reporting one alignment or suppressing reads with multiple hits (Tables 9 and 10). However, Novoalign exhibits higher false alignment rates compared to the other three aligners.

## 2.4 Discussion

Trimming off the low quality ends of reads improves their quality, and thus improves their alignment results. Although the number of reads available for alignments decreases after trimming, we still observe an increase in the number of successfully aligned reads, as well as in the concordance among aligners. S1, with a higher mean and a smaller deviation of base quality score, clearly has better quality than S2 (Figure 2-1). Thus, it is predictable that trimming has a greater effect on the S2 dataset than on the S1 dataset, which has been shown by our data analysis. Having a lower quality at the 3' end is a commonly observed problem in single-end sequencing data, especially in the early version of the Illumina sequencer. By trimming, which only takes a few minutes to process for a dataset with several million reads, users can benefit greatly. For example, more information can be extracted from the data since more reads will be aligned after trimming. With the improvement in alignment quality and quantity seen here, we recommend trimming prior to any alignment and downstream analysis, especially for poor quality data.

In the better quality dataset S1, Novoalign performs similarly to SOAP2, Bowtie, and BWA, no matter which set of parameters we use. However, in the lower quality dataset S2, Novoalign shows patterns that are different from the other three aligners. For example, Novoalign aligns more reads than the others and shows a greater increase in the number of aligned reads after trimming (Table 2-5). This might be due to the differences in alignment algorithms between Novoalign and the others. As we have shown, in SOAP2, Bowtie, and BWA, the alignment strategy is restrained by the number of mismatches allowed. That means users can specify the number of mismatches they prefer

for any alignment process to obtain optimal results for their purpose. Unlike the other three aligners, Novoalign uses an alignment score as a criterion. This alignment score is calculated based upon the base qualities, the existence of gaps, and the ambiguous codes for the entire read. For Novoalign, setting the threshold of the alignment score "-t" at 60 in the command line ensures that only the alignments with an alignment score of no more than 60 are reported, which is approximately equivalent to allowing two mismatches in alignment. However, this is only the case when the quality of reads is within a reasonable range. When applying these aligners to poor quality datasets, such as S2, Novoalign may become more sensitive to the data quality, and therefore show quite different results as compared to SOAP2, Bowtie, and BWA. After trimming off the low quality ends, the quality of the reads is improved. Thus, the Novoalign results become more similar to the others.

Since the alignment results may be sensitive to the choice of the alignment score threshold, especially for the lower quality data S2, we explore the impact of the parameter "-t" in Novoalign by setting it at different values: default (set automatically based on read length, genome size and other factors), 60, 70, and 75. For both S1 and S2 datasets, 'default value' decreases the concordance of Novoalign with other aligners dramatically; using 70 and 75, the concordance of Novoalign and other aligners is similar to the one using 60. Therefore, we conclude that the pattern of lower concordance of Novoalign with others in a poor quality dataset is not due to improper parameter choice.

In addition to Novoalign, Bowtie also allows users to have the option of considering the quality of mismatches. It enables users to set the maximum permitted total of quality values at all mismatched positions throughout the entire alignment (i.e., the "-e" option

when setting parameters to run Bowtie). To investigate this parameter setting in Bowtie, we both allow 2 mismatches and set the parameter "–e" at 20, 40, 60, and 80, respectively (data not shown). For our datasets, when the "-e" parameter is set at 40, 60 and 80, there are nearly identical results as compared to the output from only setting the number of allowed mismatches at 2 (i.e., "-v 2"). But setting "–e" at 20 shows severe departures from the other three aligners. In our datasets, most reads have moderate to good quality scores. However, setting "–e" at 20 only allows extremely low quality mismatched positions, and therefore rules out the majority of reads with high quality mismatched positions.

Like trimming off the reads, suppressing multiple alignments also improves the consistency among the three aligners (Table 2-6). Out of the multiple locations of the reference genome that one read can be aligned onto, only one is true. Even though all aligners can choose one alignment for each read, based on a certain standard, there is no guarantee that the one they choose represents the true location. Thus, eliminating all reads having multiple alignments will help improve the accuracy of alignments and also the consistency among the four aligners. Our analysis resulting from the S1 and S2 datasets supports this conclusion. We design one simulated dataset that contains many repetitive bases. By eliminating reads with multiple alignments, the false alignment rate decreases to almost 0 for SOAP2, Bowtie, and BWA, and below 9% for Novoalign (Table 2-10).

In addition to the trimming and initial parameter setting of aligners, we also investigate the impact of filtering the alignments based on the mapping quality score provided in the output files of different aligners. Out of the four aligners, BWA and

Novoalign both report a mapping quality score for each alignment. For BWA, this score is approximately a Phred-scaled probability of the alignment being incorrect, which takes the values of 37, 25, and any value between 23 and 0. In general, a score of 37 means the read is aligned to a unique position with less than 2 mismatches; a score of 25 means the read is aligned to a unique position with 2 mismatches; a score between 23 and 0 means the read is aligned to multiple locations, such that a lower score means that the mapped location is less accurate (based on BWA source code). For Novoalign, the mapping quality score correlates with the probability of the alignment given the read and genome, and ranges from 0 to 150. Higher scores mean better alignment qualities. To explore the effect of quality score filtering, we checked the mapping quality scores in the untrimmed S1 and S2 data with one alignment reported randomly (Figure 2-3). The distribution of scores shows that both aligners yield alignments with high mapping quality scores. For Novoalign, the majority of reads have a mapping quality score of 150 (Figure 2-3A and B), which is the upper limit of the score. For BWA, the majority of reads have a score of 37 or 25 (Figure 2-3C and D), which means each of them is explicitly aligned to a unique position with 0 to 2 mismatches. A small fraction of reads have scores between 23 and 0. These reads are generally mapped to multiple locations in the reference genome. Therefore, quality score filtering wouldn't show much impact on the concordance among aligners in the real datasets. In addition, since SOAP2 and Bowtie do not have alignment quality scores in their respective output files, to ensure a relatively fair comparison, no alignment quality filters are used.

As for the mapability of those target regions that we used in our simulation data, we have checked their mapability using the "Duke uniqueness 35bp" method provided by the

UCSC genome browser for the 218 CpG islands and 3000 exon regions. This Duke method reports a mapability score between 0 and 1, with 1 representing a completely unique sequence. A score of 0.5, 0.3, 0.25, or 0 represents that the sequence occurs twice, three times, four times, or more than four times, respectively. For the 218 CpG islands, 80.09% are completely unique, which means all 35-bp sequences within these islands occur only once in the genome; while 19.91% are not completely unique, which means at least one 35-bp sequence within each of these islands occurs more than once in the genome. The median mapability score of all CpG islands is 1 and the mean is 0.9830. For the 3000 exon regions, 95.40% are completely unique and 4.60% are not completely unique. The median mapability score of all regions is 1 and the mean is 0.9930. Generally speaking, the 3000 exon simulation data has better mapabilty than the 218 CpG island data.

There are different ways to evaluate the current available programs. For example, Ruffalo et al. developed a simulation and evaluation suite to compare a few available aligners using only simulated data (Ruffalo, et al., 2011). In this chapter, we focus mainly on comparing them from two specific angles (i.e., using real reads with varying qualities and simulated reads from repetitive regions). Thus, there are a few limitations in our work. First, rather than from the whole human genome, both the real data and the simulated data are from part of it. Second, our sequencing datasets are only from the Illumina sequencer. Third, we mainly use single-end sequencing data without considering pair-end data. Fourth, there are many other great alignment algorithms (Alkan, et al., 2009; Chen, et al., 2009; De Bona, et al., 2008; Hach, et al., 2010; Jiang and Wong, 2008; Lam, et al., 2008; Li, et al., 2008; Lin, et al., 2008; Ma, et al., 2002; Rumble, et al., 2009;

Schatz, 2009; Weese, et al., 2009) that we did not compare. Although our work has these limitations, the approach we used is very general, and it can be applied to the pair-end whole genome real and simulated sequencing data as well as to data generated from other platforms. It can also be utilized with some minor modification, if necessary, to study the performance of other alignment programs.

## 2.5 Conclusion

In order to study how alignment programs perform on data with varying quality and from repetitive regions, we have evaluated the performances of four commonly used alignment programs—SOAP2, Bowtie, BWA, and Novoalign—on two real NGS datasets and two simulated datasets. Our results show that, for sequencing data with reads that have relatively good quality or have had the low quality bases trimmed off, all four alignment programs perform similarly. We have also demonstrated that trimming off low quality ends markedly increases the number of aligned reads and improves the consistency among different aligners, especially for low quality data. However, Novoalign is more sensitive to the improvement of data quality. Trimming off low quality ends increases the concordance between Novoalign and the others significantly. Therefore, the quality of sequencing data has a great impact on alignment results, and we highly recommend assessing sequencing quality first and then trimming off low quality bases if necessary. As for aligning reads from repetitive regions, our simulation data show that reads from repetitive regions tend to be aligned incorrectly, and suppressing reads with multiple hits can improve alignment accuracy.

## References


Alkan, C., *et al.* (2009) Personalized copy number and segmental duplication maps using next-generation sequencing, *Nature Genetics*, **41**, 1061-1067.

Altschul, S.F., *et al.* (1990) Basic local alignment search tool, *Journal of Molecular Biology*, **215**, 403-410.

Burrows, M. and Wheeler, D.J. (1994) A block-sorting lossless data compression algorithm. *Technical Report 124.* Digital Equipment Corporation, Palo Alto, CA.

Chen, Y., Souaiaia, T. and Chen, T. (2009) PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds, *Bioinformatics*, **25**, 2514-2521.

De Bona, F., *et al.* (2008) Optimal spliced alignments of short sequence reads, *BMC Bioinformatics*, **24**, i174-i180.

Ferragina, P. and Manzini, G. (2000) Opportunistic data structures with applications. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, pp. 390-398.

Hach, F., *et al.* (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping, *Nature Methods*, **7**, 576-577.

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

Jiang, H. and Wong, W.H. (2008) SeqMap: mapping massive amount of oligonucleotides to the genome, *Bioinformatics*, **24**, 2395-2396.

Lam, T.W., *et al.* (2008) Compressed indexing and local alignment of DNA, *Bioinformatics*, **24**, 791-797.

Langmead, B., *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome, *Genome biology*, **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 Durbin, R. (2010) Fast and accurate long-read alignment with Burrows–Wheeler transform, *Bioinformatics*, **26**, 589-595.

Li, H. and Homer, N. (2010) A survey of sequence alignment algorithms for next-generation sequencing, *Briefings in Bioinformatics*, **11**, 473-483.

Li, H., Ruan, J. and Durbin, R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores, *Genome Research*, **18**, 1851-1858.

Li, R., *et al.* (2008) SOAP: short oligonucleotide alignment program, *Bioinformatics*, **24**, 713-714.

Li, R., *et al.* (2009) SOAP2: an improved ultrafast tool for short read alignment, *Bioinformatics*, **25**, 1966-1967.

Lin, H., *et al.* (2008) ZOOM! Zillions of oligos mapped, *Bioinformatics*, **24**, 2431-2437.

Ma, B., Tromp, J. and Li, M. (2002) PatternHunter: faster and more sensitive homology search, *Bioinformatics*, **18**, 440-445.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology*, **48**, 443-453.

Ning, Z., Cox, A.J. and Mullikin, J.C. (2001) SSAHA: A Fast Search Method for Large DNA Databases, *Genome Research*, **11**, 1725-1729.

Ruffalo, M., LaFramboise, T. and Koyutürk, M. (2011) Comparative analysis of algorithms for next-generation sequencing read alignment, *Bioinformatics*, **27**, 2790-2796.

Rumble, S.M., *et al.* (2009) SHRiMP: Accurate Mapping of Short Color-space Reads, *PLoS Computational Biology*, **5**, e1000386.

Schatz, M.C. (2009) CloudBurst: highly sensitive read mapping with MapReduce, *Bioinformatics*, **25**, 1363-1369.

Weese, D., *et al.* (2009) RazerS—fast read mapping with sensitivity control, *Genome Research*, **19**, 1646-1654.

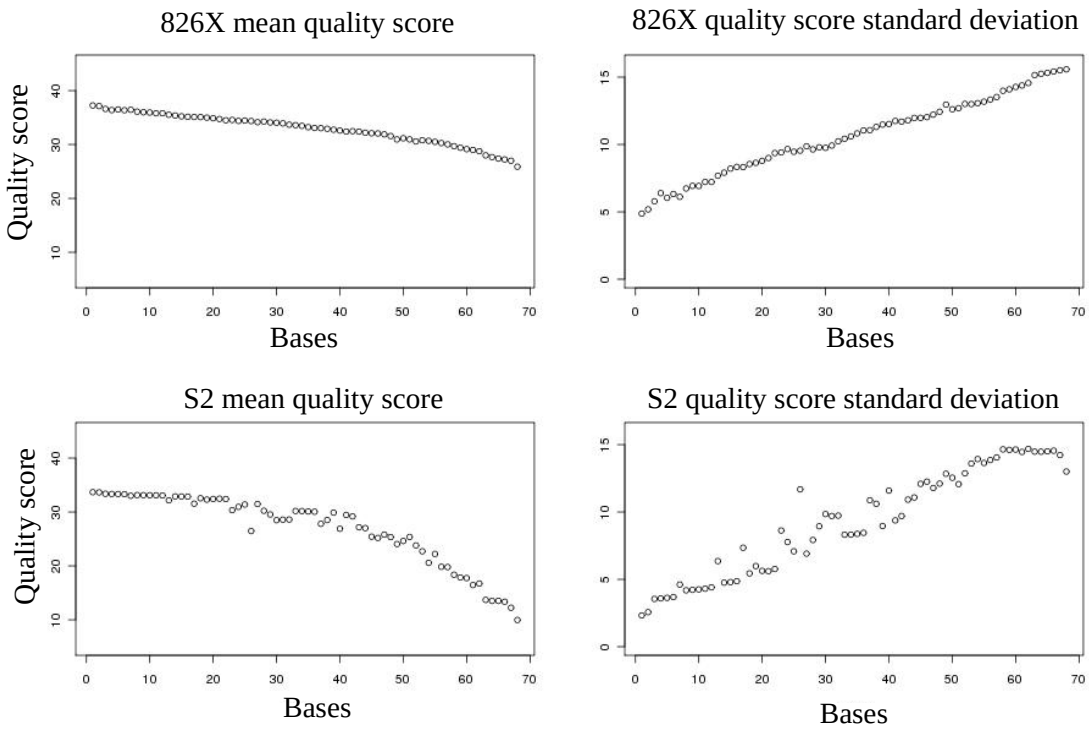



**Figure 2-1**

**Mean quality score and standard deviation for each base position in the S1 and S2 datasets.** Quality score is assessed in Illumina FASTQ format.

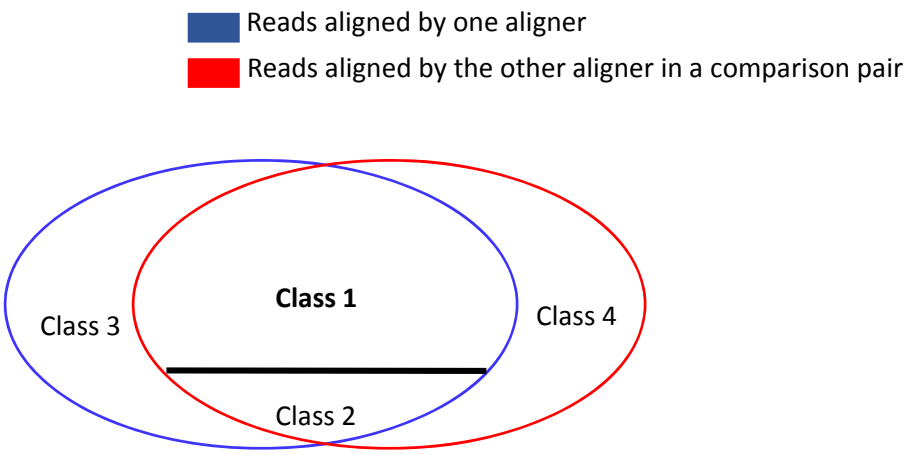



**Figure 2-2**

**The four classes to which all reads are assigned during a pair-wise comparison.** Class1 is a group of reads each of which is assigned to the same location by aligners 1 and 2; Class 2 is a group of reads each of which is assigned to a different location by aligner 1 and 2; Class 3 is a group of reads each of which is only aligned by aligner 1; Class 4 is a group of reads each of which is aligned only by aligner 2.

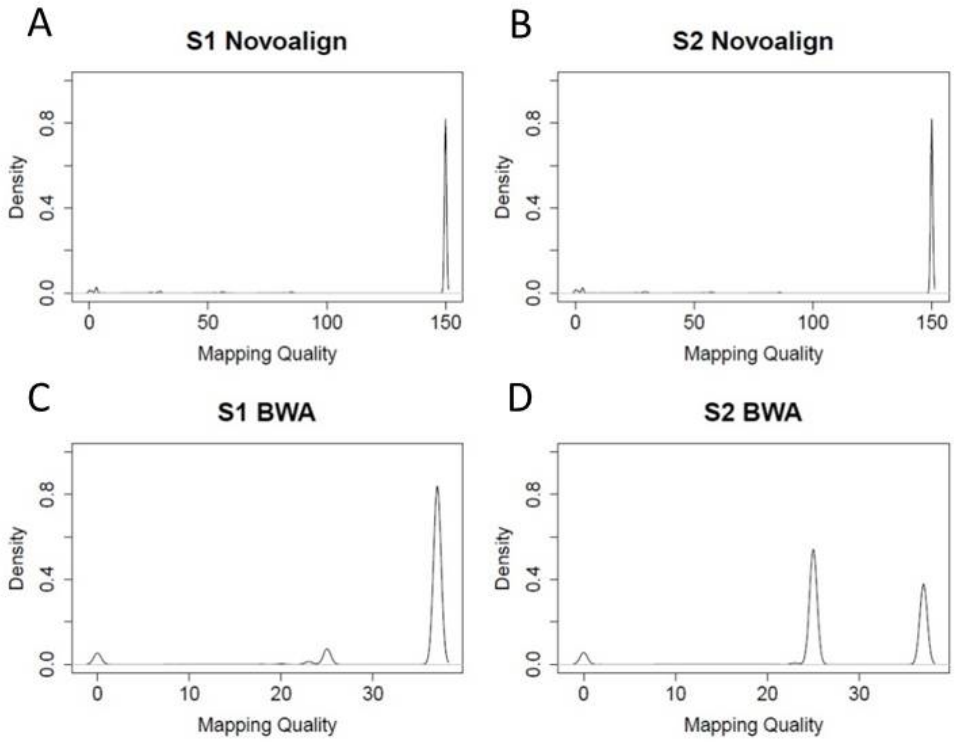



**Figure 2-3**

**Mapping quality scores reported in Novoalign and BWA.** Alignment is performed on the untrimmed S1 and S2 datasets, with one alignment randomly reported for each read.

**Table 2-1: Algorithms of four aligners: SOAP2, Bowtie, BWA, and Novoalign.**

| | **SOAP2 (2.20)*** | **Bowtie (0.12.3)** | **BWA (0.5.8C)** | **Novoalign (2.07.00)** |
|---|---|---|---|---|
| **Indexing** | FM-index | FM-index | FM-index | Hash table |
| **Inexact match** | Split read | Quality-aware backtracking | Backtracking | Alignment scoring |

*version of the program

**Table 2-2: Available options in SOAP2, Bowtie, BWA, and Novoalign**.

| | **SOAP2 (2.20)** | **Bowtie (0.12.3)** | **BWA (0.5.8C)** | **Novoalign (2.07.00)** |
|---|---|---|---|---|
| **Mismatch allowed** | exactly 0,1,2 | max in seed, 0-3<br>max in read | up to *k** | up to 8 or more in single end; |
| **Alignments reported per read** | random/all/none | up to any | up to any | random/all/none/ |
| **Gap alignment** | 1-3 bp gap | unavailable | available | up to 7 bp |
| **Pair-end reads** | available | available | available | available |
| **Best alignment** | minimal number of mismatch | minimal number of mismatch | minimal number of mismatch | highest alignment score |
| **Trim bases** | 3' end | 3' and 5' end | available | 3' end** |

*Given a read of length *m*, less than 4% of *m*-long reads with 2% uniform base error rate may have more than *k* mismatches. For *m*=15-37 bp, *k*=2; for *m*=38-64 bp, *k*=3; for *m*=64-92 bp, *k*=4; for *m*=92-123 bp, *k*=5; for *m*=124-156 bp, *k*=6.
**only available for single-end reads

**Table 2-3: Indexing and alignment time of four alignment programs.**

Index is built on the human genome 18 for each aligner. 7.4 million single-end reads are then mapped onto the human genome 18. The read length is 68 bp. At most two mismatches are allowed in all programs, and one alignment is randomly reported for each read. The CPU time in minutes on dual quad-core 2.66Ghz Xeon E5430 processor for index building and alignment processing, as well as percent of mapped reads, are shown in this table.

| **Programs** | **Index time (min)** | **Alignment time (min)** | **Reads aligned (%)** |
|---|---|---|---|
| **SOAP2 (2.20)** | 89.50 | 15.4 | 75.96 |
| **Bowtie (0.12.3)** | 192.00 | 21.2 | 75.71 |
| **BWA (0.5.8C)** | 101.50 | 26.4 | 76.12 |
| **Novoalign (2.07.00)** | 4.02 | 62.9 | 74.61 |

**Table 2-4: Percentage of reads aligned in S1 and S2 datasets by four aligners under different settings.**

| Aligners with different settings | | S1 | | S2 | |
|---|---|---|---|---|---|
| | | w/o trim | w/ trim | w/o trim | w/ trim |
| | | 7,406,247 | 7,006,805 | 5,398,566 | 5,193,655 |
| **Randomly report one alignment per read** | SOAP2 | 75.96% | 91.45% | 42.12% | 76.81% |
| | Bowtie | 75.71% | 91.36% | 41.83% | 76.67% |
| | BWA | 76.12% | 91.80% | 41.94% | 76.88% |
| | Novoalign | 73.64% | 91.60% | 34.50% | 76.94% |
| **Suppress reads w/ multiple alignments** | SOAP2 | 71.85% | 85.90% | 39.75% | 71.31% |
| | Bowtie | 68.82% | 81.90% | 38.89% | 68.63% |
| | BWA | 74.40% | 84.07% | 39.12% | 69.75% |
| | Novoalign | 69.67% | 86.09% | 32.63% | 71.63% |

**Table 2-5: Agreement among aligners in S1 non-trimmed data**.

| Comparison pair | | Class 1 | Class 2 | Class 3 | Class 4 |
|---|---|---|---|---|---|
| **Randomly report one alignment per read** | SOAP2 vs. Bowtie[1] (5,626,038)[2] | 96.25% | 3.41% | 0.34% | 0.002% |
| | SOAP2 vs.BWA (5,656,559) | 95.72% | 3.40% | 0.34% | 0.54% |
| | Bowtie vs. BWA (5,637,504) | 95.80% | 3.66% | 0.00002% | 0.54% |
| | SOAP2 vs. Novoalign (5,757,260) | 85.13% | 7.32% | 5.27% | 2.28% |
| | Bowtie vs. Novoalign (5,748,724) | 85.18% | 7.26% | 5.13% | 2.47% |
| | BWA vs. Novoalign (5,835,451) | 85.20% | 7.24% | 5.37% | 2.19% |
| **Suppress reads with multiple alignments** | SOAP2 vs. Bowtie (5,321,512) | 95.78% | 0.00002% | 4.22% | 0.003% |
| | SOAP2 vs.BWA (5,361,466) | 96.50% | 0.0005% | 2.75% | 0.75% |
| | Bowtie vs. BWA (5,213,871) | 97.76% | 0.00% | 0.0004% | 2.24% |
| | SOAP2 vs. Novoalign (5,447,206) | 88.14% | 4.27% | 5.28% | 2.31% |
| | Bowtie vs. Novoalign (5,432,410) | 84.72% | 4.08% | 5.02% | 6.18% |
| | BWA vs. Novoalign (5,458,788) | 85.92% | 4.11% | 5.48% | 4.49% |

1. Comparison pair in the format of aligner 1 vs. aligner 2.
2. Total number of reads aligned by either of these two aligners in a comparison pair.

## Table 2-6: Agreement among aligners in S2 non-trimmed data.

| Comparison pair | | Class1 | Class 2 | Class 3 | Class 4 |
|---|---|---|---|---|---|
| **Randomly report one alignment per read** | SOAP2 vs. Bowtie[1] (2,209,957)[2] | 95.69% | 3.62% | 0.69% | 0.003% |
| | SOAP2 vs.BWA (2,215,397) | 95.45% | 3.61% | 0.69% | 0.25% |
| | Bowtie vs. BWA (2,200,129) | 95.37% | 3.70% | 0.00% | 0.25% |
| | SOAP2 vs. Novoalign (2,436,379) | 49.58% | 15.40% | 25.72% | 9.26% |
| | Bowtie vs. Novoalign (2,424,001) | 49.81% | 15.38% | 25.53% | 9.46% |
| | BWA vs. Novoalign (2,428,458) | 49.68% | 15.44% | 25.48% | 9.40% |
| **Suppress reads with multiple alignments** | SOAP2 vs. Bowtie (2,085,316) | 97.84% | 0.00% | 2.15% | 0.007% |
| | SOAP2 vs.BWA (2,094,218) | 97.57% | 0.0008% | 1.99% | 0.43% |
| | Bowtie vs. BWA (2,052,464) | 99.94% | 0.00% | 0.0003% | 0.59% |
| | SOAP2 vs. Novoalign (2,303,060) | 51.37% | 13.36% | 25.71% | 9.46% |
| | Bowtie vs. Novoalign (2,283,644) | 50.93% | 13.35% | 25.07% | 10.65% |
| | BWA vs. Novoalign (2,292,171) | 50.86% | 13.33% | 25.35% | 10.46% |

1 Comparison pair in the format of aligner 1 vs. aligner 2.
2 Total number of reads aligned by either of these two aligners in a comparison pair.

## Table 2-7: Agreement among aligners in S1 trimmed data.

| Comparison pair | | Class 1 | Class 2 | Class 3 | Class 4 |
|---|---|---|---|---|---|
| **Randomly report one alignment per read** | SOAP2 vs. Bowtie[1] (6,409,534)[2] | 95.89% | 3.95% | 0.13% | 0.03% |
| | SOAP2 vs.BWA (6,440,873) | 95.42% | 3.92% | 0.13% | 0.52% |
| | Bowtie vs. BWA (6,432,433) | 95.30% | 4.21% | 0.00002% | 0.49% |
| | SOAP2 vs. Novoalign (6,430,033) | 94.62% | 4.84% | 0.13% | 0.35% |
| | Bowtie vs. Novoalign (6,422,084) | 94.77% | 4.84% | 0.07% | 0.33% |
| | BWA vs. Novoalign (6,435,917) | 94.83% | 4.84% | 0.30% | 0.05% |
| **Suppress reads with multiple alignments** | SOAP2 vs. Bowtie (6,020,802) | 95.29% | 0.0002% | 4.68% | 0.003% |
| | SOAP2 vs.BWA (6,068,512) | 96.26% | 0.0005% | 2.93% | 0.81% |
| | Bowtie vs. BWA (5,890,868) | 97.42% | 0.00% | 0.0004% | 2.58% |
| | SOAP2 vs. Novoalign (6,043,150) | 98.47% | 0.95% | 0.18% | 0.40% |
| | Bowtie vs. Novoalign (6,035,510) | 94.11% | 0.92% | 0.06% | 4.92% |
| | BWA vs. Novoalign (6,066,586) | 95.62% | 0.92% | 0.57% | 2.90% |

1. Comparison pair in the format of aligner 1 vs. aligner 2.
2. Total number of reads aligned by either of these two aligners in a comparison pair.

**Table 2-8: Agreement among aligners in S2 trimmed data.**

| Comparison pair | | Class 1 | Class 2 | Class 3 | Class 4 |
|---|---|---|---|---|---|
| **Randomly report one alignment per read** | SOAP2 vs. Bowtie[1] (3,890,070)[2] | 94.94% | 4.84% | 0.20% | 0.02% |
| | SOAP2 vs.BWA (3,900,529) | 94.69% | 4.82% | 0.20% | 0.29% |
| | Bowtie vs. BWA (3,892,602) | 94.77% | 4.96% | 0.0002% | 0.27% |
| | SOAP2 vs. Novoalign (3,909,055) | 93.84% | 5.32% | 0.34% | 0.50% |
| | Bowtie vs. Novoalign (3,901,709) | 94.50% | 5.30% | 0.15% | 0.50% |
| | BWA vs. Novoalign (3,908,656) | 93.96% | 5.30% | 0.33% | 0.41% |
| **Suppress reads with multiple alignments** | SOAP2 vs. Bowtie (3,611,489) | 96.20% | 0.0002% | 3.79% | 0.02% |
| | SOAP2 vs.BWA (3,636,423) | 96.42% | 0.0007% | 2.87% | 0.70% |
| | Bowtie vs. BWA (3,531,986) | 98.38% | 0.00% | 0.0007% | 1.62% |
| | SOAP2 vs. Novoalign (3,638,616) | 98.07% | 0.54% | 0.32% | 0.76% |
| | Bowtie vs. Novoalign(3,631,179) | 95.06% | 0.52% | 0.11% | 4.31% |
| | BWA vs. Novoalign (3,652,782) | 97.99% | 0.54% | 0.70% | 3.31% |

1 Comparison pair in the format of aligner 1 vs. aligner 2.
2 Total number of reads aligned by either of these two aligners in a comparison pair.

**Table 2-9: Percentage of aligned reads and the false alignment rate for 3000 exon simulation data.**

| **A. Mismatches with high quality (30-40)** | | | | | | |
|---|---|---|---|---|---|---|
| **Mismatch** | **Settings** | | **SOAP2** | **Bowtie** | **BWA** | **Novoalign** |
| **1** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 0.76 | 0.77 | 0.76 | 4.83 |
| | Suppress reads w/ multiple alignments | aligned (%) | 98.69 | 98.65 | 98.68 | 98.69 |
| | | False alignments (%) | 0 | 0 | 0 | 4.13 |
| **2** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 0.78 | 0.78 | 0.76 | 8.95 |
| | Suppress reads w/ multiple alignments | aligned (%) | 98.69 | 98.68 | 98.68 | 98.67 |
| | | False alignments (%) | 0 | 0 | 0 | 8.26 |
| **B. Mismatches with low quality (5-15)** | | | | | | |
| **Mismatch** | **Settings** | | **SOAP2** | **Bowtie** | **BWA** | **Novoalign** |
| **1** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 0.77 | 0.75 | 0.76 | 3.10 |
| | Suppress reads w/ multiple alignments | aligned (%) | 98.69 | 98.65 | 98.68 | 98.69 |
| | | False alignments (%) | 0 | 0 | 0 | 4.13 |
| **2** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 0.77 | 0.81 | 0.76 | 5.49 |
| | Suppress reads w/ multiple alignments | aligned (%) | 98.69 | 98.68 | 98.68 | 98.67 |
| | | False alignments (%) | 0.02 | 0 | 0 | 4.78 |

**Table 2-10: Percentage of aligned reads and the false alignment rate for 218 CpG island simulation data.**

| **A. Mismatches with high quality (30-40)** | | | | | | |
|---|---|---|---|---|---|---|
| **Mismatch** | **Settings** | | **SOAP2** | **Bowtie** | **BWA** | **Novoalign** |
| **1** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 13.80 | 13.84 | 13.80 | 17.25 |
| | Suppress reads w/ multiple alignments | aligned (%) | 84.26 | 84.26 | 84.26 | 84.34 |
| | | False alignments (%) | 0 | 0 | 0.01 | 4.09 |
| **2** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 13.90 | 13.98 | 13.91 | 20.77 |
| | Suppress reads w/ multiple alignments | aligned (%) | 84.39 | 84.22 | 84.39 | 84.23 |
| | | False alignments (%) | 0.21 | 0 | 0.02 | 8.20 |
| **B. Mismatches with low quality (5-15)** | | | | | | |
| **Mismatch** | **Settings** | | **SOAP2** | **Bowtie** | **BWA** | **Novoalign** |
| **1** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 13.79 | 13.83 | 13.80 | 15.93 |
| | Suppress reads w/ multiple alignments | aligned (%) | 84.26 | 84.26 | 84.26 | 84.34 |
| | | False alignments (%) | 0 | 0 | 0.001 | 2.42 |
| **2** | Randomly report one alignment | aligned (%) | 100 | 100 | 100 | 100 |
| | | False alignments (%) | 13.82 | 13.86 | 13.91 | 17.79 |
| | Suppress reads w/ multiple alignments | aligned (%) | 84.39 | 84.22 | 84.39 | 84.23 |
| | | False alignments (%) | 0.21 | 0 | 0.02 | 4.86 |

# CHAPTER 3: COMPARING SNP CALLING ALGORITHMS USING LOW-COVERAGE SEQEUCNING DATA

## 3.1 Introduction

SNPs, which make up over 90% of all human genetic variation (Collins, et al., 1998), contribute to phenotype differences and disease risk. Due to their high frequency and binary variation patterns, SNPs have been widely used as generic markers in disease association studies to identify genes associated with both monogenic (Jimenez-Sanchez, et al., 2001) and complex diseases, such as diabetes (Altshuler, et al., 2000; Palmer, et al., 2011; Wolford, et al., 2006; Zeggini, et al., 2005), autoimmune diseases (Arinami, et al., 2005; Ueda, et al., 2003; Vyshkina and Kalman, 2005), cancers (Bond and Levine, 2006; Kammerer, et al., 2005), and Alzheimer's disease (Corneveaux, et al., 2010; Kuwano, et al., 2006). SNPs also serve as popular mabout olecular markers in pharmacogenomic studies to understand inter-individual differences in response to treatments (Henningsson, et al., 2005; Higashi, et al., 2002). Therefore, it is essential to obtain accurate SNP information through advanced methods such as high throughput next-generation sequencing (NGS) technologies.

NGS technologies (e.g., the Solexa/Illumina sequencer, 454/Roche system, and SOLiD/ABI system) have been widely used in the last several years (Shendure, et al., 2004). A single sequencing run by an NGS platform can generate data in the gigabase-pair scale, which usually contains millions and even hundreds of millions of sequencing reads. This high throughput makes NGS technologies more suitable for Single Nucleotide Variant (SNV) identification compared to traditional technologies. However, challenges

are also present. To produce such an enormous amount of data, multiple sequencing procedures (e.g., template amplification, florescent intensity detection, and base calling) are involved in NGS technologies (Metzker, 2010). As a result, artifacts can be introduced by both systematic and random errors. These errors include mishandled templates, PCR amplification bias, and fluorescence noise. Since SNV detection relies on the identification of polymorphisms at the level of individual base pairs, any sequencing error can lead to an incorrect SNP identification. Furthermore, other genetic variants (e.g., copy number variation, insertion, deletion, inversion, and rearrangements) make accurate SNP calling even more difficult.

In order to identify SNVs using NGS data, various SNP calling programs have been subsequently developed (Altmann, et al., 2011; Bansal, 2010; Cibulskis, et al., 2013; DePristo, et al., 2011; Edmonson, et al., 2011; Garrison and Marth, 2012; Goya, et al., 2010; Koboldt, et al., 2009; Li, et al., 2008; Li, et al., 2009; Martin, et al., 2010; Quinlan, et al., 2008; Rivas, et al., 2011; Shen, et al., 2010; Vallania, et al., 2010; Wei, et al., 2011). For a general survey on SNP calling programs, please check the review paper by Pabinger *et al*. (Pabinger, et al., 2013). These programs serve as useful tools to detect SNPs from high throughput sequencing data and greatly extend the scale and resolution of sequencing technology applications. Our preliminary work has shown that, for sequencing datasets that have high coverage and are of high quality, SNP calling programs can perform similarly (Adams, et al., 2012). However, when the coverage level is low in a sequencing dataset, it is challenging to accurately call SNVs (The Genomes Project, 2012). Moreover, commonly used SNP calling programs (e.g., SOAPsnp (Li, et al., 2009), Atlas-SNP2 (Shen, et al., 2010), SAMtools (Deng, et al., 2009), and GATK

(DePristo, et al., 2011; McKenna, et al., 2010)) all include different metrics for each potential SNP in their output files. These metrics are highly correlated in complex patterns, which make it challenging to select SNPs that are used for further experimental validation. In order to accurately detect SNPs from a low-coverage sequencing dataset, effective solutions have been in great demand. Some studies have shown that incorporating haplotype information and other pooled information can help in identifying SNPs in multiple-sample datasets (Li, et al., 2012; Li, et al., 2011; The Genomes Project, 2012). However, many pilot studies have a small sample size (e.g., one or two samples), so the multiple-sample methods cannot be applied. Although the difficulty of SNP calling using single-sample low-coverage sequencing data has been recognized, it is still unclear how well different SNP calling algorithms perform and how to choose reliable SNPs from their results.

In this chapter, we have conducted a systematic analysis using a single-sample low-coverage dataset to compare the performance of four commonly used SNP calling algorithms: SOAPsnp, Atlas-SNP2, SAMtools, and Unified Genotyper (UGT) in GATK. We have also explored the filtering choice based on the metrics reported in the output files of these algorithms. First, we improve the quality of the raw sequencing data by trimming off the low quality ends for reads in the data, then call SNVs using the four algorithms on these trimmed sequencing reads. We compare the SNV calling results from the four algorithms without using any post-output filters. Second, we explore the values of a few key metrics related to SNVs' quality in each algorithm and use them as the post-output filtering criteria to filter out low quality SNVs. Third, we choose several cutoff values for the coverage of called SNVs in order to increase the agreement among the four

algorithms. With the above analysis procedure, our goal is to offer insights for efficient and accurate SNV calling for single-sample low-coverage sequencing datasets.

## 3.2 Methods

### 3.2.1 Reviewing the key features of SNP calling algorithms

#### *3.2.1.1 Preprocessing steps of different SNP calling algorithms*

Alignment (i.e., mapping the reads back to a reference genome) is a fundamental and crucial step of any NGS data analysis, including SNP calling. In order to eliminate the possible sources of calling errors in the alignment results, almost all SNP calling algorithms incorporate certain processing steps, as shown in Table 3-1. In this section, we review these steps one by one.

1) In order to deal with duplicate reads that may be generated during PCR, Atlas-SNP2, SAMtools, and GATK remove all the reads with the same start location in the initial alignment, except the one that has the best alignment quality. In contrast, instead of removing the duplicate reads, SOAPsnp sets a penalty to reduce the impact of these duplications.
2) In order to deal with reads that are aligned to multiple locations on the genome, SOAPsnp only takes into account the uniquely aligned reads, i.e., reads with only one best hit (the alignment with the least number of mismatches). Atlas-SNP2, GATK, and SAMtools do not have a specific strategy to deal with the multiple-hit issue, instead these calling programs accept all hits that the alignment results provide.
3) In order to make sure the sequencing quality of each read reflects the true sequencing error rate, SOAPsnp, SAMtools, and GATK recalibrate the raw sequencing quality

scores generated by NGS platforms. Key factors, such as raw quality scores, sequencing cycles, and allele types, are all considered.

4) In order to deal with the presence of indels, both SAMtools and GATK include a realignment step to ensure accurate variant detection. In particular, GATK constructs the haplotype that could best represent the suspicious regions and realigns these regions appropriately according to this best haplotype. In contrast, SOAPsnp and Atlas-SNP2 do not utilize a specific indel realignment algorithm. SOAPsnp authors have conducted a simulation using a set of simulated data with 10,000 indels, and have shown that only 0.6% of reads containing indels are misaligned, and only 0.03% of those incorrect SNPs are retained in the final SNP calling output after routine processes, including pre-filtering and genotype determination.

#### *3.2.1.2 SNP calling*

In order to identify novel SNPs using sequencing reads and their quality scores, all four SNP calling programs apply the Bayesian method. SOAPsnp, SAMtools, and GATK-UGT compute the posterior probability for each possible genotype, and then choose the genotype with the highest probability ($P_H$) as the consensus genotype. A SNP is called at a specific position if its consensus genotype is different from the reference. As a result, for both SOAPsnp and SAMtools, a *phred*-like consensus quality score, representing the accuracy of the SNP calling, is calculated as $-10\log_{10}[1-P_H]$. Different from the other three algorithms, Atlas-SNP2 calculates the posterior probabilities for each variant allele instead of the genotype, and the genotype is determined afterwards according to the ratio of the number of reads covering the reference and the number of

reads covering the most likely variant. Depending on the Bayesian framework that each SNP calling program uses, different sets of metrics can be considered in SNP calling procedures (Table 3-2). Several common parameters are often considered by most calling programs (e.g., quality scores, sequencing cycles, and allele types). There are also some parameters specifically adopted by each algorithm. In particular, Atlas-SNP2 considers several unique metrics: 1) whether the allele is involved in a multi-nucleotide polymorphism (MNP) event; 2) whether the allele is a "swap-base", defined as the situation in which two adjacent mismatches invert their nucleotides with respective to the reference; 3) whether the allele passes the neighboring quality standard (NQS), which means that the quality score of the variant allele should be higher than 20, and the quality score of each of the five flanking bases on both sides should be higher than 15; and 4) whether the variant allele coverage is at least 3. SAMtools incorporates two unique metrics, base dependency and strand independency. The former accounts for the correlation between bases, while the latter assumes that reads from different strands are more likely to have independent error probabilities.

#### *3.2.1.3 Built-in filtering*

After obtaining the raw genotypes or variant alleles, several internal filters are used by Atlas-SNP2, SAMtools and GATK-UGT to further identify potential SNPs (Table 3-3). For example, Atlas-SNP2 allows users to set up a cutoff value for the posterior probability to get a customized list of potential variants among those putative variant alleles. The genotyping results are given in a variant call format (VCF) output file and several criteria are applied to determine the final genotypes:

(1) Both strands are required to be supported by variant alleles.

(2) Cutoff values for the percentage of variant reads are set to determine homozygous or heterozygous genotypes. In particular, at a specific locus, if less than 10% of the total reads support the variant allele, the genotype is determined to be a homozygous reference for this locus; if the percentage of variant reads is between 10% and 90%, a heterozygous genotype is assigned to this locus; if the percentage of variant reads is higher than 90%, this locus is determined as a homozygous variant.

(3) A binomial test is employed to estimate the genotype qualities, and gives a posterior probability to indicate how confident the algorithm is in calling this position as a variant.

Similar to Atlas-SNP2, SAMtools and UGT also produce SNP calling results in VCF output. Therefore, the internal filtering criteria of VCF are incorporated in GATK-UGT and SAMtools (e.g., the *phred*-scaled quality score for the variant allele must be higher than a certain value). Since the VCF also reports some additional information about the called SNPs, such as strand bias, quality by depth (coverage), mapping quality, read depth, and genotype quality that represents the quality of the called SNPs, users can further filter the called SNPs based on the cutoff values they choose for these metrics. Although SOAPsnp does not particularly use any internal filtering, it does provide several metrics in the output for each called SNP, e.g., consensus score, quality of best allele, quality of second best allele, and sequence depth. These metrics can be used as customized post-output filters.

### 3.2.2 Datasets

To study the performance of these different SNP calling tools in low-coverage data, we use a low-coverage (1-2X) whole-genome sequencing dataset from the pilot 1 of 1000 genome project: ERR000044. This dataset is sequenced from the sample #NA18550, with 6,333,357 45-bp-long reads generated. We first explore the sequencing quality by plotting the quality scores at each base using the software package FastQC (http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc). The sequence quality stays high at the beginning of the reads, and then drops quickly when reaching towards the 3' end of the reads (Figure 3-1).

### 3.2.3 SNP detection and comparison

There are four major steps in the overall workflow (Figure 3-2). First, before alignment, we trim off the low quality ends of reads using the trim function in the BRAT packageemploy. In particular, the BRAT trim function is set to cut from both the 5' and 3' ends until it reaches bases with quality scores higher than 20 (i.e., 1% error rate). This trim function allows at most two Ns in each read. Second, alignments are conducted by either SOAP2 (version 2.21) or BWA (version 0.6.2), using the human genome 18 as the reference. At most two mismatches are allowed for each read, and only the reads aligned to unique positions are reported in the output files. Third, SNPs are called on chromosomes 1 and 2. All SOAPsnp calls are performed on SOAP2 alignment results, since SOAP2 is the only input format SOAPsnp can take. Because Atlas-SNP2, SAMtools, and GATK-UGT all require alignment results in the SAM format, which can be generated by BWA but not SOAP2, these three are performed on BWA alignment

outputs. For the results of each SNP calling algorithm, we identify the dbSNPs and non-dbSNPs, using the dbSNP information (dbSNP build 130) downloaded from the UCSC Genome Browser (http://genome.ucsc.edu/). Finally, we compare the SNP calling results from the four algorithms. Since Atlas-SNP2 requires at least 3X coverage to detect a variant, for a fair comparison, we only use SNPs with at least 3X in each algorithm. All detected SNVs are assigned to the following classes:

I. Single nucleotide variants (SNV) identified by only one SNP calling algorithm.
II. SNVs identified by any two SNP calling algorithms.
III. SNVs identified by any three SNP calling algorithms.
IV. SNVs identified by all four SNP calling algorithms.

This procedure is first conducted without any post-output filters. Then we apply filters based on the key metrics in the output of each SNP calling algorithm (Table 3-4), with different coverage cutoff values.

## 3.3 Results

### 3.3.1 Alignment and the impact of trimming

In raw data, among the 6,333,357 single-end reads, about 70% are aligned against human genome 18 by SOAP2 and BWA. 110 to 400 non-dbSNPs (potentially novel SNVs) are detected in each of the four SNP calling algorithms on chromosomes 1 and 2 (Table 3-5A). Since trimming can remove low-quality bases and thus improve the alignment results (Yu, et al., 2012), we trim the data using the trim function of the BRAT package. This process not only cuts off the low quality bases from both ends, but also discards reads that are shorter than 24-bp after trimming. As a result, 6,000 (0.1%) reads

are removed. With slightly fewer reads (6,327,430) available, however, the number of aligned reads is increased by 100,000 (2%). Consequently, more SNPs are detected in trimmed data compared to raw data (Table 3-5B). Among the four algorithms, SOAPsnp calls more SNVs than the other three, in both raw and trimmed data. This is probably due to the fact that SOAPsnp has almost no internal filtering criterion after calling a SNV, meaning that it is not as stringent as the others. Although SOAP2 aligns slightly more reads than BWA, our previous study has shown that SOAP2 and BWA have similar alignment performance in trimmed data (Yu, et al., 2012). Therefore, the difference between SOAPsnp and the other three algorithms is less likely caused by alignment disagreements. When compared to SOAPsnp, Atlas-SNP2 calls significantly less SNVs than the other programs. The possible reasons are: 1) more stringent internal criteria are applied to determine SNVs, including the coverage for variant alleles on both strands and the percentage of variant reads; 2) the threshold for the posterior probability is set as ≥ 0.95. Since Atlas-SNP2 requires at least 3X coverage to call a SNV, we only report the called SNVs with ≥ 3X coverage in the other three algorithms. Without any coverage filtering (≥ 1X) in both raw and trimmed datasets, SOAPsnp calls about 4000 SNVs, which are dramatically more than the 2000 SNVs called by SMAtools and GATK. Since SNVs from raw and trimmed data show similar patterns, and trimmed data have more SNVs called, we use the trimmed data in further analysis.

### 3.3.2 Comparison without any filtering

In order to examine the agreement between the four algorithms, we compare both dbSNP and non-dbSNP results in trimmed data (see Figure 3-3). Overall, dbSNPs exhibit a better agreement than non-dbSNPs. This observation is consistent with our expectations.

Since the dbSNP positions are known and well studied, they are more likely to be called. However, in terms of the performance of the four algorithms, dbSNPs and non-dbSNPs show similar patterns. Figure 3-3 shows that GATK-UGT and SAMtools have better agreement compared to the other comparison pairs. This is probably due to one or more of the following reasons: 1) they are both Bayesian-based algorithms; 2) they incorporate similar information when determining the genotypes; and 3) they apply similar internal filters to the called SNVs. Because Atlas-SNP2 is more stringent than the other three calling programs, most of the SNVs called by Atlas-SNP2 are also called by at least one of the other programs. Different from Atlas-SNP2, there are 101 dbSNPs and 160 non-dbSNPs that are only called by SOAPsnp. In order to investigate the difference between these SNVs that are only called by SOAPsnp and those that are also called by at least one of the other three algorithms, we compare their key metrics from the SOAPsnp output: consensus score, quality of best allele, quality of second best allele, and sequencing depth. No obvious difference is discovered between the two types of SNVs. Most of the SNVs have a consensus score between 2 and 20, with only a few reaching the upper limit of 99. Moreover, most of the SNVs are covered by 3 to 10 reads in total.

### 3.3.3 Exploration of key metrics in four SNP calling algorithms

#### *3.3.3.1 Key metrics in SOAPsnp*

We have examined SOAPsnp's SNP calling quality in low-coverage data by checking the coverage and consensus scores for called dbSNPs and non-dbSNPs. We have found that low coverage is often associated with low consensus scores, while high coverage is often associated with high consensus scores. The consensus score in

SOAPsnp represents how confident the algorithm is in calling a SNV. A higher value corresponds to a higher confidence. Therefore, using the consensus score as a filter is necessary in order to have accurate SNP calling in SOAPsnp. We have checked the distribution of consensus scores in SOAPsnp results and have chosen filtering criteria based on this distribution. Table 3-6 shows that 91 SNVs have a consensus score < 5, indicating lower confidence. With a filtering criterion for consensus scores set at ≥ 5, 91 SNVs are removed and 877 SNVs are left in total.

#### *3.3.3.2 Key metrics in Atlas-SNP2*

Unlike SOAPsnp, Atlas-SNP2 provides a posterior probability for every potential SNV. It requires users to set a threshold for the posterior probability. With a low coverage, many potential SNVs reported by Atlas-SNP2 have low posterior probabilities. In our previous analysis, we use "posterior probability ≥ 0.95" as a criterion to call SNVs, resulting in a much smaller number of SNVs when compared to the other three calling programs. In order to investigate whether the posterior probability is a potential filter criterion, we set the cutoffs at ≥ 0.3 and then ≥ 0.1. With a lower threshold of 0.1, the number of SNVs called by Atlas-SNP2 increases from 448 to 539 (Table 3-7).

#### *3.3.3.3 Key metrics in GATK-UGT*

In the GATK-UGT output, there are several metrics associated with the quality of potential SNVs. We have checked a few important ones among them, which are "genotype quality", "QUAL", 'FisherStrand", "HaplotypeScore", "MappingQualityRankSumTest", and "ReadPosRankSumTest".

"Genotype quality" represents the quality of the called SNVs. It ranges from 0 to 99, with higher values corresponding to higher qualities. To better understand the calling quality of GATK-UGT in low-coverage data, we have checked the distribution of the genotype quality. In this low-coverage dataset, for dbSNPs, the genotype ranges from 4 to 99, and 80% of dbSNPs have a genotype quality lower than 30; while for non-dbSNPs, the genotype ranges from 2 to 99, and 70% of non-dbSNPs have a genotype quality lower than 30. Then, based on the distribution, we choose several different cutoff values for genotype quality, ≥ 5, 6, 7, 8, 9, and 10 (Table 3-8). With the cutoff set at ≥ 9, 53 SNVs (32 dbSNPs and 21 non-dbSNPs respectively) are removed, resulting in 676 remaining SNVs.

In the VCF output, there is a metric called "QUAL", a *phred*-scaled quality probability of the SNVs being a homozygous reference. A higher "QUAL" score indicates a higher confidence. In our dataset, all called SNVs have a QUAL value ≥ 30, which is a commonly used criterion for reliable SNP calling in GATK-UGT.

Another indicator of SNVs' quality is strand bias, which looks for the instance where the variant allele is disproportionately represented on one strand. In the GATK-UGT output, "FisherStrand" is a *phred*-scaled p-value using Fisher's Exact test to detect strand bias. A higher "FisherStrand" value represents a more pronounced bias, indicating a false positive. The commonly used criterion for reliable SNV calling is to remove any SNV with a "FisherStrand" value > 60. In our dataset, the "FisherStrand" value for all SNVs ranges from 0 to 25. Therefore, there is no need for filtering using "FisherStrand".

"HaplotypeScore" in the GATK-UGT output is a measure of how well the data from a 10-base window around the called SNV can be explained by at most two haplotypes. Usually, in the case of mismapped reads, there are more than two haplotypes around the SNV and this SNV is likely to be a false positive. A higher "HaplotypeScore" value represents a higher probability that the called SNV is artificial due to mismapping. In Table 3-9, we check the distribution of "HaplotypeScore" in dbSNPs and non-dbSNPs. The majority of SNVs have a low "HaplotypeScore" ($\leq 10$), indicating a generally good mapping in this dataset. Since the commonly used criterion for reliable SNVs calling is removing any SNV with a "HaplotypeScore" > 13, we use 13 as a filtering criterion, which removes 26 SNVs in total.

"MappingQualityRankSumTest" is a Wilcoxon rank test that tests the hypothesis that the reads carrying the variant allele have a consistently lower mapping quality than the reads with the reference allele. This metric is only available for the SNVs where both the variant allele and reference allele are supported by reads. In our dataset, there are 225 SNVs (97 dbSNPs and 126 non-dbSNPs) that have "MappingQualityRankSumTest" values, indicating that they have coverage for both the variant and reference allele. In these 225 SNVs, the "MappingQualityRankSumTest" value ranges from -7 to 2 for dbSNPs, and -5 to 2 for non-dbSNPs. The commonly used criterion for reliable SNVs calling removes any SNV with a "MappingQualityRankSumTest" value < -12.5. Since in our dataset all SNVs are > -12.5, there is no need to apply any filter on the "MappingQualityRankSumTest" values.

"ReadPosRankSumTest" is a Mann-Whitney Rank Sum Test that tests the hypothesis that instead of being randomly distributed over the read, the variant allele is

consistently found more often at the beginning or the end of a sequencing read. Similar to the "MappingQualityRankSumTest", this metric is also only available for the SNVs where both the variant allele and reference allele are supported by reads. In our dataset, for the SNVs that actually have the "ReadPosRankSumTest" report, their values range from -5 to 6. These values satisfy the common criterion that the "ReadPosRankSumTest" value is ≥ -20.

Based on the above exploration of the six key metrics in the GATK-UGT output, we set a series of filtering criteria for reliable SNP calling by GATK-UGT: "genotype quality" ≥ 9; "QUAL" ≥ 30; "FisherStrand" ≤ 60; "HaplotypeScore" ≤ 13; "MappingQualityRankSumTest" ≥ -12.5; "ReadPosRankSumTest" ≥ -20. As a result, 650 SNVs (out of 729 raw SNVs) pass the filtering, 427 dbSNPs and 223 non-dbSNPs. We will use this set of SNVs in a later analysis. Since "QUAL", "FisherStrand", "MappingQualityRankSumTest", and "ReadPosRankSumTest" values all satisfy the criteria in our dataset, we cannot remove any SNV by applying filtering on these four metrics. However, they are all important metrics that are related to SNP quality. Thus, we recommend that users filter raw SNP calling results based on their values.

#### *3.3.3.4 Key metrics in SAMtools*

Similar to GATK-UGT, SAMtools reports the VCF output. We have checked two important metrics in SAMtools results: "genotype quality" and "QUAL". In both dbSNPs and non-dbSNPs, the values of genotype quality range from 4 to 99. Setting different cutoff values for "genotype quality" does not filter out significantly more of the called SNP (Table 3-10). For "QUAL", all SNVs have a QUAL value ≥ 3, which is a commonly

used criterion for "QUAL" in SAMtools results. Therefore, for our dataset we do not apply any filter on SAMtools results and use the raw SNVs for a later analysis.

### 3.3.4 Comparison with filtering using key metrics and coverage

To compare the four algorithms under different coverage levels, we use the SNP calling results with filtering criteria applied in each calling program, and then add the filtering of coverage with several cutoff values, ≥ 4X, 5X, 6X, 7X, 8X, 9X, and 10X (Table 3-11). The number of SNVs called by each calling program decreases dramatically by more than 50% when the cutoff increases from 3X to 4X, and drops to about 15% at 10X. With 3X, SOAPsnp calls more SNVs than the other calling programs, while Atlas-SNPs calls the least. However, when the coverage cutoff increases, the number of SNVs called by each calling program becomes more similar, with SOAPsnp calling slightly more.

Table 3-11 shows the changing patterns of the number of SNVs as the coverage cutoff level increases. Although the numbers of SNVs identified by the different calling programs become more similar as the coverage cutoff increases, it is unclear whether the agreement of different calling programs and their performance will increase accordingly. In order to address this question, we have done further comparisons using the following two methods: Method 1 checks the agreement among different calling algorithms (see Table 3-12, Figures 3-4 and 3-5), and Method 2 calculates empirical positive calling rates and sensitivity (see Table 3-13). For both methods, we check dbSNPs and non-dbSNPs separately.

#### *3.3.4.1 Method 1: check the agreement among different calling programs*

For dbSNPs, using the original setting (≥ 3X), there are 592 unique dbSNPs called by the four algorithms, and 46.79% of them are common among all the calling programs. When increasing the cutoff of coverage to 4X, although the number of unique dbSNPs drops dramatically from 592 to 276, the percentage of agreements among the four calling programs remains similar (Table 3-12A). With a further increase of coverage cutoff values, the number of unique dbSNPs continually decreases, while the agreements stay similar (Table 3-12A). For each SNP calling program, we plot the agreements with other algorithms under different coverage cutoffs (Figure 3-4). For SOAPsnp, even though the number of called dbSNPs drops dramatically, the agreement with other calling programs does not change as much as the coverage cutoff increases. For Atlas-SNP2, the percentage of agreement with the other three calling programs decreases when the coverage cutoff increases. This is probably due to the fact that with a lower cutoff (≥ 3X), Atlas-SNP2 calls many fewer SNVs than the other calling programs. Therefore, compared to the other programs, the 277 agreeing dbSNPs form a larger portion of all the SNVs called by Atlas-SNP2. However, when the coverage cutoff increases, the number of dbSNPs called by Atlas-SNP2 is far more similar to the other algorithms, therefore the percentage of agreement in Atlas-SNP2 becomes smaller. Compared to SOAPsnp and Atlas-SNP2, GATK-UGT and SAMtools exhibit a higher agreement with other calling programs. 60-70% of their dbSNPs are called by all four programs, 20% are called by three programs, and about 10% are called by two programs (see Figure 3-4, bottom panel). Moreover, in both GATK-UGT and SAMtools, when the cutoff increases from 3X to 5X, the percentage of dbSNPs called by all four programs increases 3-4%.

For non-dbSNPs, the comparison results show similar patterns as for dbSNPs, but with a lower percentage of agreement (Table 3-12B). The number of unique non-dbSNPs called by the four algorithms drops from 402 to 211 when the coverage cutoff increases from 3X to 4X, and finally decreases to 79 when the coverage cutoff is 10X. The percentage of non-dbSNPs called by all four calling programs increases over the different coverage cutoffs, especially from 3X to 7X, while the percentage of non-dbSNPs only called by one algorithm decreases over the cutoffs, from 37.56% in 3X to 31.65% in 10X. For each calling program, we plot the agreement with other algorithms under different coverage cutoffs (Figure 3-5). Among the four calling algorithms, SOAPsnp shows the lowest percentage of agreements with the others. This low agreement is probably due to the fact that SOAPsnp always calls more SNVs than the other programs under all coverage levels we use. In all four calling programs, the percentage of agreement increases with the coverage cutoff value, especially from 3X to 7X, indicating that filtering the non-dbSNPs with a higher coverage threshold improves the agreement among the four algorithms.

#### *3.3.4.2 Method 2: calculate empirical positive calling rates and sensitivity*

For this comparison method, we choose the variants that are called by at least three calling programs as the "empirical truth", and then investigate the calling performance of each SNP calling program based on this empirical truth by calculating both the positive calling rate and the sensitivity. We then compare the four calling programs at different coverage levels using these rates. The positive calling rate and the sensitivity are calculated as Positive calling rate = A/(A+B), and Sensitivity = A/(A+C) as shown in Table 3-13. In these formulas, A is the number of SNVs identified as an empirical truth

(i.e., called by at least 3 calling programs) and also called by this calling program; B is the number of SNVs identified as an empirical truth, but not called by this calling program; and C is the number of SNVs called by this calling program, but is not an empirical truth.

The results of comparing the four SNP calling algorithms using the empirical positive calling rate and sensitivity are shown in Table 3-14 and Table 3-15 and are explained below.

1) For calling dbSNP positions, Table 3-14A (dbSNPs) shows that SOAPsnp has a relatively lower positive calling rate. This is because SOAPsnp tends to call more variants than the other three calling programs, suggesting a higher false positive rate. GATK has a relatively higher positive calling rate than the others at all different coverage levels for calling dbSNPs. Atlas-SNP2 and SAMtools tend to stay between SOAPsnp and GATK.
2) For calling non-dbSNP positions, similar to dbSNPs, Table 3-14B shows that SOAPsnp tends to call more false positive variants since it lacks stringent internal filtering criteria. Atlas-SNP2 shows the highest positive calling rate. This is probably because it is the most stringent calling program. GATK has a higher positive calling rate than SOAPsnp and SAMtools.
3) As far as the positive calling rate is concerned, Atlas-SNP2 and GATK perform better than SOAPsnp and SAMtools on both dbSNPs and nondbSNPs. With the change of coverage level, the comparison results are relatively stable.
4) For calling dbSNPs and non-dbSNPs, Table 3-15 shows that, with the exception of SAMtools, the other three programs all have very high sensitivity in calling SNVs.

Overall the sensitivity of all calling programs are pretty stable across the different coverage levels, except that Atlas-SNP2's sensitivity is a bit low at 3X coverage.

## 3.4 Discussion

Identifying a reliable list of SNPs is critical when analyzing NGS data. For data with high-coverage and/or multiple samples, previous studies have shown that different SNP calling algorithms have a good agreement between each other and have high true positive rates (Li, et al., 2012; Li, et al., 2011; The Genomes Project, 2012). However, for single-sample low-coverage data, it is difficult to call SNVs with high confidence. In order to provide insights into the choice of SNP calling program, we have compared the performance of four commonly used SNP calling algorithms using low coverage sequencing data.

### 3.4.1 The four SNP calling algorithms and post-output filtering

Out of the four algorithms, SOAPsnp calls many more SNVs compared to the others. This is probably because it has less internal filtering criteria. After applying the criterion that removes any SNVs with a consensus score lower than 5, the total number of SNVs called by SOAPsnp decreases and becomes more similar to the other algorithms. In the SOAPsnp output file, the consensus score is an important metric representing the quality of calling a SNP. Therefore, when processing low-coverage data, we recommend that users apply the consensus score as a post-output filter for SOAPsnp results.

Atlas-SNP2 is much more stringent compared to the other three algorithms. 97% of the SNVs called by Atlas-SNP2 are also called by at least one of the other three calling

programs. With a much lower threshold for the posterior probability, Atlas-SNP2 calls more SNVs but still fewer than the other algorithms. Since it has the lowest number of called SNVs, Atlas-SNP2 appears to have a higher positive calling rate and sensitivity when compared to the other calling programs (Tables 14 and 15). However, when using Atlas-SNP2 to deal with low-coverage dataset, users should be careful with the filtering settings. For example, in this study, we set the threshold for posterior probability at 0.1, which indicates a low confidence in calling a SNP. Because Atlas-SNP2 is much more stringent than the other programs, even with a low posterior probability, the called SNVs are still very likely to agree with other calling programs.

Compared to the above two algorithms, GATK-UGT and SAMtools call a moderate number of SNVs. When using the GATK-UGT package, applying the common criteria is necessary, including "Genotype quality", "QUAL", "MappingQualityRankSumTest", "FisherStrand", "HaplotypeScrore", and "ReadPosRankSumTest". With the SAMtools program, filtering out the SNVs with low genotype quality and low "QUAL" value can help improve the accuracy of SNP calling.

Filtering out the low quality SNVs is an important step before performing further analysis, especially for low-coverage data. When choosing the criteria for filtering, it is important not only to consider the commonly used standards, but also to take into account the characteristics of each specific dataset. For example, in our dataset, all the SNVs have little or no strand bias, have high "MappingQualityRankSumTest" scores, and have high "ReadPosRankSumTest" scores. Setting the threshold of genotype quality at 9 gives a similar number of SNVs compared to the other programs. Besides the key metrics that we have explored in the Results section, each algorithm provides additional information. For

instance, SOAPsnp reports the quality of the variant and reference alleles, the number of reads covering the variant and reference alleles, average copy number, and more. GATK-UGT and SAMtools both report their results in VCF, which can include many metrics. Users may check these metrics based on the characteristics of their own data if necessary, though we did not find these metrics to be very helpful (data not shown).

### 3.4.2 The impact of coverage

Coverage is an important factor to consider when assessing the quality of called SNVs. Without any coverage filtering (i.e., just using ≥ 1X coverage), the results of the four calling programs can be dramatically different. Usually, high coverage regions or bases tend to have higher calling qualities (e.g., higher consensus scores in SOAPsnp, higher posterior probabilities in Atlas-SNP2, and higher genotype qualities in SAMtools and UGT). Low coverage regions or bases tend to have lower SNP calling qualities. However, there is not a simple linear relationship between coverage and the genotype quality scores that are generated by different SNP calling programs.

Our results show that when increasing the coverage levels for each calling program, the number of identified SNVs drops dramatically in all calling programs. However, increasing sequencing coverage cutoffs does not necessarily lead to an increase in agreement among the different calling programs. In fact, our comparison results show that the impact of coverage on calling agreement is small, except that we see some agreement increase in non-dbSNPs when the coverage level changes from 3X to 7X (Figure 3-5). This may sound counterintuitive. However, this observation can be explained by the fact that the four programs use different statistical methods and

algorithms, which model different aspects of the sequencing information. These differences lead to the complex correlations of output metrics.

Filtering out many low-coverage SNVs may result in a sacrifice of missing novel SNVs. For example, the number of called SNVs in each calling program decreases by more than 50% when the coverage cutoff increases from 3X to 4X, and drops to 15% at 10X. Therefore, caution should be used when choosing coverage as a filtering criterion. Simply choosing the SNVs called with high coverage might not be sufficient. This is because, with a higher threshold of coverage, users may over-filter the results and miss novel SNPs related to the disease of interest.

### 3.4.3 Generalization of our results and decision making

In this chapter, we use a set of single-end data, which is one mate of a pair-end dataset. We have also conducted the same analysis using a different single-end sequencing dataset and have arrived at the same conclusions. Therefore, we only report the results from the first dataset we used. In addition, the results we report here are generated by analyzing chromosomes 1 and 2 together. We have also analyzed chromosomes 1 and 2 separately and get the same conclusions as when they are combined. Furthermore, the findings in this chapter are similar to the results reported by other researchers (O'Rawe, et al., 2013). Therefore, our comparison methods and results can be generally applied to low-coverage sequencing data. In addition, although this work mainly focuses on SNP calling in a single sample, our methods and conclusions can be easily applied to variant calling in multiple samples. In particular, the empirical-based

positive calling rate and sensitivity analysis can serve as an empirical standard for comparing algorithms in multiple-sample SNP calling.

Overall, the four calling programs have very low agreement amongst each other, with only roughly 35% ~ 45% for dbSNPs and 19% ~28% for non-dbSNPs. For very low coverage data, it might be wise to choose a concordance among two or more SNP calling program instead of just using one algorithm. However, this may result in a high false-negative rate, with many true SNVs being missed. In addition, choosing filtering cutoff values for coverage and different quality scores with high and low values may have the same advantages and disadvantages as choosing a single SNP calling program vs. using the concordance of two or more SNP calling programs. Therefore, as far as the experimental validation of novel SNVs is concerned, we recommend that users employ a comprehensive strategy in their validation plan. First, in order to obtain a high experimental validation rate, users may choose the SNVs that are called by more than one algorithm and with high metrics (e.g., coverage and quality scores) in the beginning of the validation process. Then, if the validation success rate is high, users may validate more low coverage SNVs called by multiple calling programs, or SNVs called by only one program but with high quality. This approach can both ensure an effective validation and avoid missing many true disease-contributing SNVs.

## 3.5 Conclusions

We have compared the performance of four SNP calling programs in a low-coverage single-sample sequencing dataset. It is important to filter out the SNVs of low quality using different metrics (e.g., quality scores and coverage). Our results show that the

concordance among these different calling algorithms is low, especially in non-dbSNPs. We also find that increasing the cutoff values of coverage has little effect on improving the concordance. Although this finding is consistent with pervious research results in low-coverage data (O'Rawe, et al., 2013), it seems to be very counterintuitive. The above finding is also different from our experience with high-coverage data (Adams, et al., 2012), in which increasing the coverage cutoffs improves the agreement among SNP callers. There may be many different reasons that explain this counterintuitive result. Based on our understanding of low coverage data, we list a few possible reasons here. First, in a dataset with generally low coverage, the SNPs with extremely high coverage are likely to be false positives, which may cause the low agreement in high cutoffs. Second, in addition to coverage, there are other unknown factors that may affect the accuracy in SNP calling, and these factors may introduce more noise in low-coverage data than in high-coverage data. Third, the four SNP calling programs employ different statistical methods and algorithms to incorporate coverage and different quality metrics.

In order to provide an empirical standard for choosing a SNP calling program, we have calculated the empirical positive calling rate and sensitivity for each calling algorithm under different cutoffs of coverage. We have found that dbSNPs have generally higher rates compared to non-dbSNPs, suggesting lower quality in called non-dbSNPs in low-coverage sequencing data. Moreover, among the four calling programs, GATK and Atlas-SNP2 show a relatively higher positive calling rate and sensitivity when compared to the others, and GATK tends to call more SNVs than Atlas-SNP2. Therefore, if users intend to use only one calling program, we recommend GATK. However, in

order to increase the overall accuracy, we recommend users employ more than one SNP calling algorithm.

## References


Adams, M.D., *et al.* (2012) Global mutational profiling of formalin-fixed human colon cancers from a pathology archive, *Modern Pathology*, **25**, 1599-1608.

Altmann, A., *et al.* (2011) vipR: variant identification in pooled DNA using R, *Bioinformatics*, **27**, i77-i84.

Altshuler, D., *et al.* (2000) The common PPARr Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes, *Nature Genetics*, **26**, 76-80.

Arinami, T., *et al.* (2005) Genomewide High-Density SNP Linkage Analysis of 236 Japanese Families Supports the Existence of Schizophrenia Susceptibility Loci on Chromosomes 1p, 14q, and 20p, *American journal of human genetics*, **77**, 937-944.

Bansal, V. (2010) A statistical method for the detection of variants from next-generation resequencing of DNA pools, *Bioinformatics*, **26**, i318-i324.

Bond, G.L. and Levine, A.J. (2006) A single nucleotide polymorphism in the p53 pathway interacts with gender, environmental stresses and tumor genetics to influence cancer in humans, *Oncogene*, **26**, 1317-1323.

Cibulskis, K., *et al.* (2013) Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples, *Nature Biotechnology*, **31**, 213-219.

Collins, F.S., Brooks, L.D. and Chakravarti, A. (1998) A DNA Polymorphism Discovery Resource for Research on Human Genetic Variation, *Genome Research*, **8**, 1229-1231.

Corneveaux, J.J., *et al.* (2010) Association of CR1, CLU and PICALM with Alzheimer's disease in a cohort of clinically characterized and neuropathologically verified individuals, *Human Molecular Genetics*, **19**, 3295–3301.

Deng, J., *et al.* (2009) Targeted bisulfite sequencing reveals changes in DNA methylation associated with nuclear reprogramming, *Nature Biotechnology*, **27**, 353-360.

DePristo, M.A., *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature Genetics*, **43**, 491-498.

Edmonson, M.N., *et al.* (2011) Bambino: a variant detector and alignment viewer for next-generation sequencing data in the SAM/BAM format, *Bioinformatics*, **27**, 865-866.

Garrison, E. and Marth, G. (2012) Haplotype-based variant detection from short-read sequencing, *arXiv preprint arXiv:1207.3907v2 [q-bio.GN]*.

Goya, R., *et al.* (2010) SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors, *Bioinformatics*, **26**, 730-736.

Henningsson, A., *et al.* (2005) Association of CYP2C8, CYP3A4, CYP3A5, and ABCB1 Polymorphisms with the Pharmacokinetics of Paclitaxel, *Clinical Cancer Research*, **11**, 8097-8104.

Higashi, M.K., *et al.* (2002) Association Between CYP2C9 Genetic Variants and Anticoagulation-Related Outcomes During Warfarin Therapy, *JAMA: The Journal of the American Medical Association*, **287**, 1690-1698.

Jimenez-Sanchez, G., Childs, B. and Valle, D. (2001) Human disease genes, *Nature*, **409**, 853-855.

Kammerer, S., *et al.* (2005) Association of the NuMA region on chromosome 11q13 with breast cancer susceptibility, *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 2004-2009.

Koboldt, D.C., *et al.* (2009) VarScan: variant detection in massively parallel sequencing of individual and pooled samples, *Bioinformatics*, **25**, 2283-2285.

Kuwano, R., *et al.* (2006) Dynamin-binding protein gene on chromosome 10q is associated with late-onset Alzheimer's disease, *Human Molecular Genetics*, **15**, 2170-2182.

Li, H., Ruan, J. and Durbin, R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores, *Genome Research*, **18**, 1851-1858.

Li, R., *et al.* (2009) SNP detection for massively parallel whole-genome resequencing, *Genome Research*, **19**, 1124-1132.

Li, Y., *et al.* (2012) Single Nucleotide Polymorphism (SNP) Detection and Genotype Calling from Massively Parallel Sequencing (MPS) Data, *Statistics in Bioscience*, **5**, 3-25.

Li, Y., *et al.* (2011) Low-coverage sequencing: Implications for design of complex trait association studies, *Genome Research*, **21**, 940-951.

Martin, E.R., *et al.* (2010) SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies, *Bioinformatics*, **26**, 2803-2810.

McKenna, A., Hanna, M. and Banks, E. (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data, *Genome Research*, **20**, 1297-1303.

Metzker, M.L. (2010) Sequencing technologies -- the next generation, *Anglais*, **11**, 31-46.

O'Rawe, J., *et al.* (2013) Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing, *Genome Medicine*, **5**, 28.

Pabinger, S., *et al.* (2013) A survey of tools for variant analysis of next-generation genome sequencing data, *Briefings in Bioinformatics advanced online publication, doi: 10.1093/bib/bbs086*.

Palmer, N.D., *et al.* (2011) Resequencing and Analysis of Variation in the TCF7L2 Gene in African Americans Suggests That SNP rs7903146 Is the Causal Diabetes Susceptibility Variant, *Diabetes*, **60**, 662-668.

Quinlan, A.R., *et al.* (2008) Pyrobayes: an improved base caller for SNP discovery in pyrosequences, *Nature Methods*, **5**, 179-181.

Rivas, M.A., *et al.* (2011) Deep resequencing of GWAS loci identifies independent rare variants associated with inflammatory bowel disease, *Nature Genetics*, **43**, 1066-1073.

Shen, Y., *et al.* (2010) A SNP discovery method to assess variant allele probability from next-generation resequencing data, *Genome Research*, **20**, 273-280.

Shendure, J., *et al.* (2004) Advanced sequencing technologies: methods and goals, *Nature Reviews Genetics*, **5**, 335-344.

The Genomes Project, C. (2012) An integrated map of genetic variation from 1,092 human genomes, *Nature*, **491**, 56-65.

Ueda, H., *et al.* (2003) Association of the T-cell regulatory gene CTLA4 with susceptibility to autoimmune disease, *Nature*, **423**, 506-511.

Vallania, F.L.M., *et al.* (2010) High-throughput discovery of rare insertions and deletions in large cohorts, *Genome Research*, **20**, 1711-1718.

Vyshkina, T. and Kalman, B. (2005) Haplotypes within genes of β-chemokines in 17q11 are associated with multiple sclerosis: a second phase study, *Human Genetics*, **118**, 67-75.

Wei, Z., *et al.* (2011) SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data, *Nucleic Acids Research*, **39**, e132.

Wolford, J.K., *et al.* (2006) Variants in the gene encoding aldose reductase (AKR1B1) and diabetic nephropathy in American Indians, *Diabetic Medicine*, **23**, 367-376.

Yu, X., *et al.* (2012) How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?, *BioData Mining*, **5**, 6.

Zeggini, E., *et al.* (2005) Largescale studies of the association between variation at the TNF/LTA locus and susceptibility to type 2 diabetes, *Diabetologia*, **48**, 2013-2017.

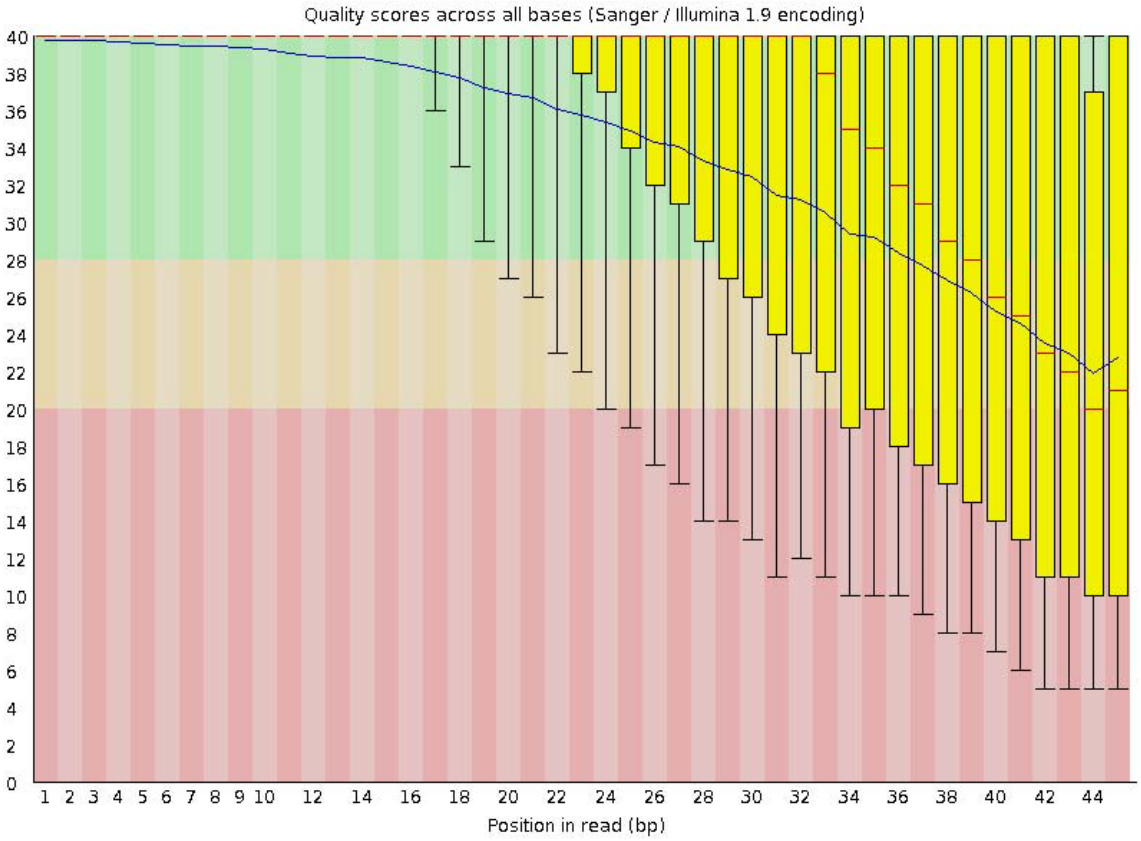



**Figure 3-1**

**Box plots for sequencing quality score (generated by FastQC).** The blue line represents the mean quality score for each base. Red lines represent medians. Yellow boxes represent 25$^{th}$ to 75$^{th}$ percentiles. The upper and lower whiskers represent 10 and 90 percentiles, respectively.

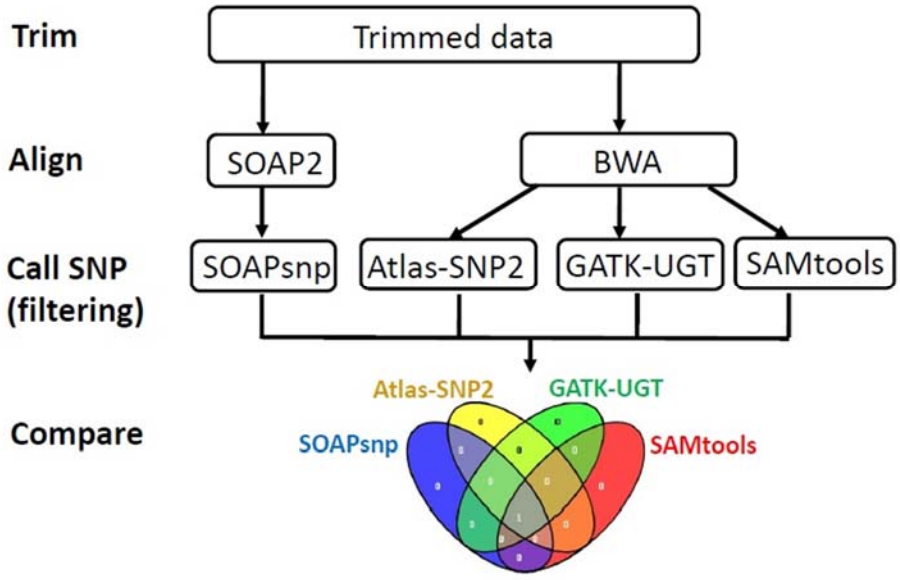



**Figure 3-2**

**The overall workflow of comparing the four SNP calling algorithms.**

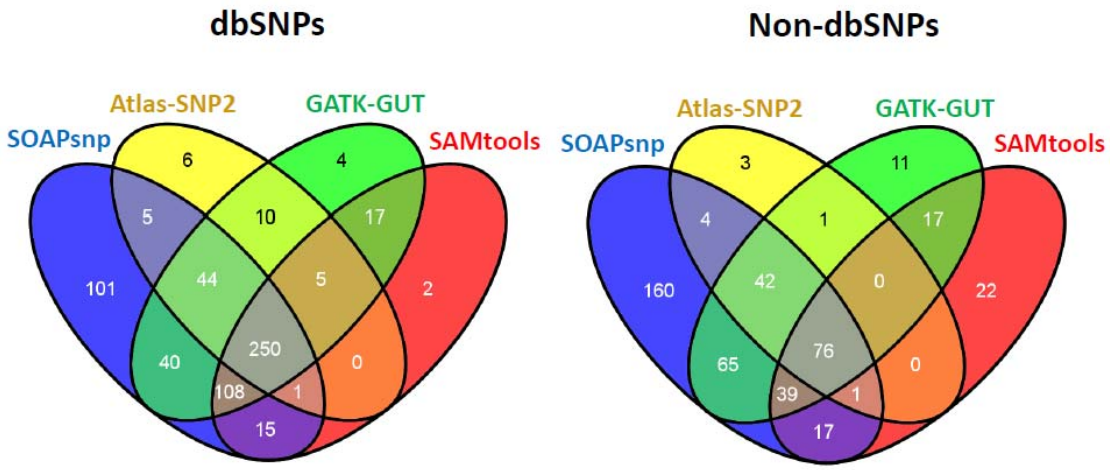



**Figure 3-3**

**The comparison results of trimmed data without any post-output filters.** All SNVs require ≥ 3X coverage.

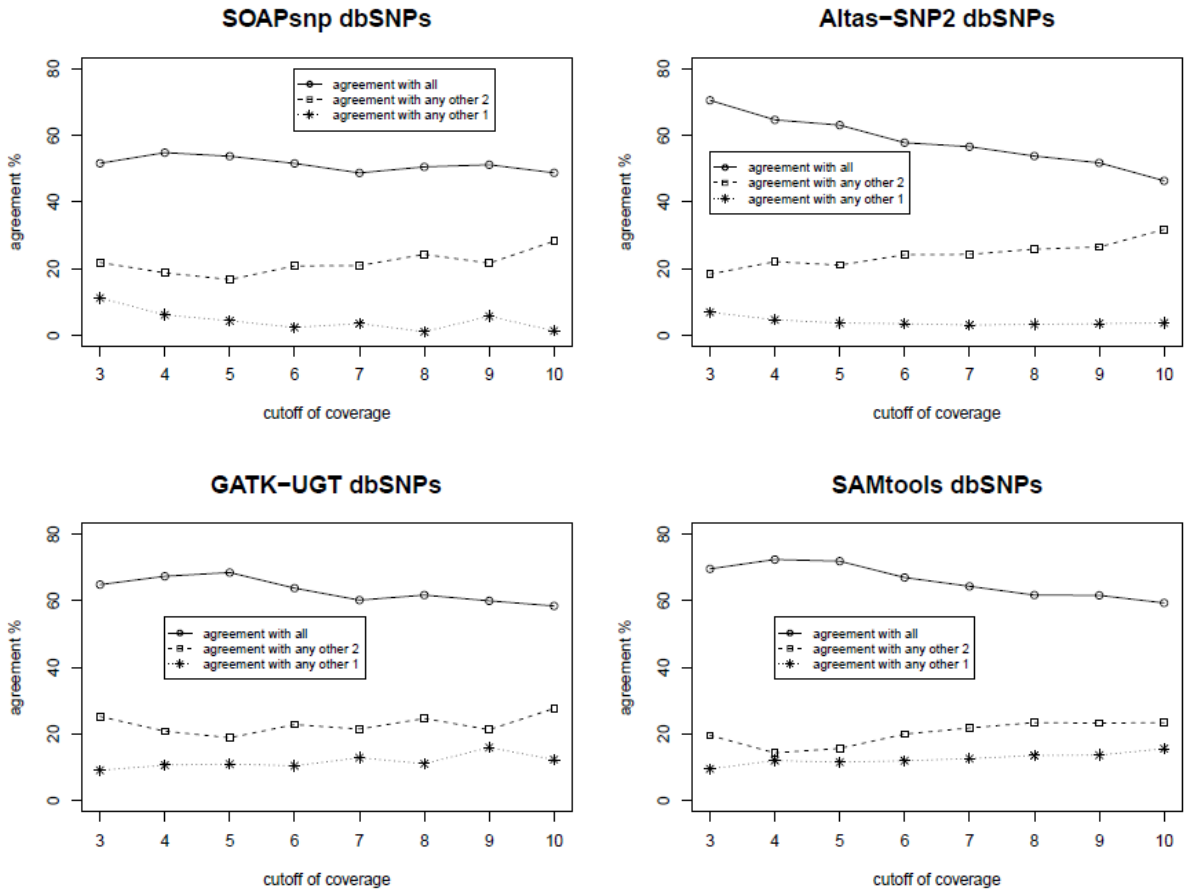



**Figure 3-4**

**The agreement of dbSNPs with different coverage cutoffs in each of the four algorithms.**

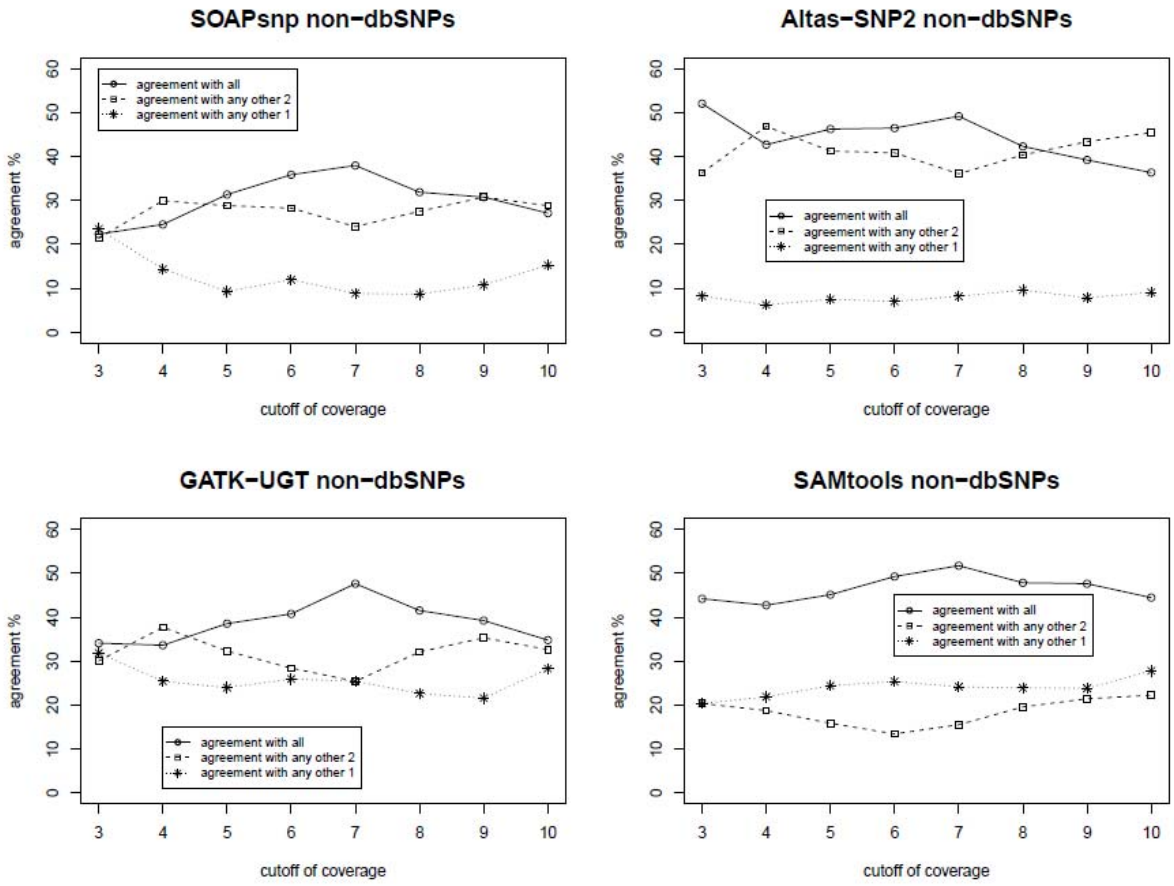



**Figure 3-5**

**The agreement of non-dbSNPs with different coverage cutoffs in each of the four algorithms.**

**Table 3-1. Preprocessing steps in each of the four algorithms.**

| | SOAPsnp | Atlas-SNP2 | SAMtools | GATK-UGT |
|---|---|---|---|---|
| **Version** | 1.03 | 1.2 | 1.1.18 | 1.6 |
| **Format of aligned reads** | SOAP output | SAM/BAM | BAM | SAM/BAM |
| **Duplicate reads** | Penalty | Remove using Atlas-SNP-mapper | Removed | Remove using picard (http://picard.sourceforge.net/) |
| **Reads with multiple-hit** | Remove | Keep all hits | Keep all hits | Keep all hits |
| **Quality recalibration** | Yes | No | Yes | Yes |
| **Realignment** | No | No | Yes | Yes |

**Table 3-2. Metrics considered in calling SNPs by each of the four algorithms.**

| | SOAPsnp | Atlas-SNP2 | SAMtools | GATK-UGT |
|---|---|---|---|---|
| **Quality score** | Recalibrated | Raw | Recalibrated | Recalibrated |
| **Machine cycle** | Yes | Yes | No | Yes |
| **Allele type** | Yes | No | No | Yes |
| **Duplication level** | Penalty in quality score | No | No | No |
| **Swap-base** | No | Yes | No | No |
| **MNP events** | No | Yes | No | No |
| **NQS** | No | Yes | No | No |
| **Coverage variation** | No | Yes | No | No |
| **Base dependency** | Yes | No | Yes | No |
| **Strand independency** | No | NO | Yes | No |

**Table 3-3. Criteria for calling a SNP in each of the four algorithms.**

| | SOAPsnp | Atlas-SNP2 | SAMtools | GATK -UGT |
|---|---|---|---|---|
| **Quality score** | No | Yes | Yes | Yes |
| **Strand bias** | No | Both strands must be covered by variant allele | Yes | Yes |
| **Coverage limits** | No | variant allele coverage ≥ 3<br>upper limits for coverage | Yes | No |
| **Variant reads percentage** | No | Heterozygous: ≥ 10%<br>Homozygous variant: ≥ 90% | No | No |
| **SNP Location** | No | No | No | No |

**Table 3-4. Key metrics in each of the four algorithms.**

| | Metrics |
|---|---|
| **SOAPsnp** | Consensus score [0, 99] |
| **Atlas-SNP2** | Posterior Probability |
| **SAMtools** | Genotype quality [0,99], QUAL |
| **GATK-UGT** | Genotype quality [0,99], QUAL, FisherStrand, HaplotypeScore, MappingQualityRankSumTest, ReadPosRankSumTest |

**Table 3-5. Number of SNVs called by four algorithms using raw and trimmed data.**

A. In raw data

| | ≥ 3X* | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **SOAPsnp** | 940 | 545 | 395 |
| **Atlas-SNP2** | 432 | 315 | 117 |
| **SAMtools** | 532 | 376 | 156 |
| **GATK-UGT** | 669 | 444 | 225 |

B. In trimmed data

| | ≥ 3X* | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **SOAPsnp** | 968 | 564 | 404 |
| **Atlas-SNP2** | 448 | 321 | 127 |
| **SAMtools** | 570 | 398 | 172 |
| **GATK-UGT** | 729 | 478 | 251 |

* Atlas-SNP2 requires at least 3X to call a SNV. For the other three algorithms, we choose the called SNVs with ≥ 3X coverage.

**Table 3-6. Number of SNVs called by the SOAPsnp with different cutoffs of consensus score.**

| Cutoffs | SNVs | dbSNPs | Non-dbSNPs | SNVs ≥ * |
|---|---|---|---|---|
| **=0** | 41 | 10 | 31 | 968 |
| **=1** | 8 | 2 | 6 | 927 |
| **=2** | 11 | 5 | 6 | 919 |
| **=3** | 6 | 2 | 4 | 908 |
| **=4** | 25 | 8 | 17 | 902 |
| **=5** | 261 | 179 | 82 | 877 |
| **=6** | 125 | 101 | 24 | 616 |
| **=7** | 21 | 13 | 8 | 491 |
| **=8** | 58 | 36 | 22 | 470 |
| **=9** | 24 | 12 | 12 | 412 |
| **=10** | 13 | 4 | 9 | 388 |

* number of SNVs that have consensus score ≥ the cutoff values

**Table 3-7. Number of SNVs called by Atlas-SNP2 with different cutoffs of the posterior probability.**

| Posterior probability | SNVs | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **≥ 0.95 (original setting)** | 448 | 321 | 127 |
| **≥ 0.3** | 476 | 342 | 134 |
| **≥ 0.1** | 539 | 393 | 146 |

**Table 3-8. Number of SNVs called by GATK-UGT with different cutoffs of genotype quality.**

| Cutoffs | SNVs | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **≥ 0** | 729 | 478 | 251 |
| **≥ 5** | 724 | 476 | 248 |
| **≥ 6** | 723 | 476 | 247 |
| **≥ 7** | 681 | 450 | 231 |
| **≥ 8** | 681 | 450 | 231 |
| **≥ 9** | 676 | 446 | 230 |
| **≥ 10** | 476 | 217 | 259 |

**Table 3-9. Number of SNVs called by GATK-UGT with different cutoffs of HaplotypeScore.**

| Cutoffs | SNVs | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **=0** | 613 | 419 | 194 |
| **≥ 1** | 638 | 431 | 207 |
| **≥ 2** | 653 | 437 | 216 |
| **≥ 5** | 680 | 448 | 232 |
| **≥ 10** | 693 | 453 | 240 |
| **≥ 13** | 703 | 459 | 244 |
| **≥ 20** | 707 | 462 | 245 |
| **≥ 30** | 718 | 468 | 250 |
| **all** | 729 | 478 | 251 |

**Table 3-10. Number of SNVs called by SAMtools with different cutoffs of genotype quality.**

| Cutoffs | SNVs | dbSNPs | Non-dbSNPs |
|---|---|---|---|
| **≥ 4 (all)** | 570 | 398 | 172 |
| **≥ 5** | 567 | 397 | 170 |
| **≥ 6** | 565 | 396 | 169 |
| **≥ 7** | 564 | 395 | 169 |
| **≥ 8** | 563 | 395 | 168 |
| **≥ 9** | 559 | 393 | 166 |
| **≥ 10** | 558 | 393 | 165 |

**Table 3-11. Number of SNVs called by each of the four algorithms with different coverage cutoffs.**

| Coverage cutoffs | SOAPsnp | Atlas-SNP2 | GATK-UGT | SAMtools |
|---|---|---|---|---|
| **≥ 3X** | 877 (537, 340) | 539 (393, 146) | 650 (427, 223) | 570 (398, 172) |
| **≥ 4X** | 397 (230, 167) | 291 (195, 96) | 309 (187, 122) | 270 (174, 96) |
| **≥ 5X** | 280 (162, 118 ) | 218 (138, 80) | 223 (127, 96) | 203 (121, 82) |
| **≥ 6X** | 222 (130, 92) | 187 (116, 71) | 186 (105, 81) | 167 (100, 67) |
| **≥ 7X** | 194 (115, 79) | 160 (99, 61) | 156 (93, 63) | 145 (87, 58) |
| **≥ 8X** | 168 (99, 69) | 145 (93, 52) | 134 (81, 53) | 127 (81, 46) |
| **≥ 9X** | 153 (88, 65) | 138 (87, 51) | 126 (75, 51) | 115 (73, 42) |
| **≥ 10X** | 137 (78, 59) | 126 (82, 44) | 111 (65, 46) | 100 (64, 36) |

**Table 3-12. Comparing four algorithms using different coverage cutoffs.**

The total number of dbSNPs called by four algorithms; the number (percentage) of dbSNPs called by only one of the four algorithms; the number (percentage) of dbSNPs called by any two algorithms; the number (percentage) of dbSNPs called by any three algorithms; the number (percentage) of dbSNPs called by four algorithms. (A.) dbSNPs. (B.) non-dbSNPs.

**A. dbSNPs**

| Coverage cutoffs | Total | By 1 | By 2 | By 3 | By 4 |
|---|---|---|---|---|---|
| ≥ **3X** | 592 | 108 (18.24%) | 82 (13.85%) | 125 (21.11%) | 277 (46.79%) |
| ≥ **4X** | 276 | 68 (24.64%) | 32 (11.59%) | 50 (18.12%) | 126 (45.65%) |
| ≥ **5X** | 201 | 61 (30.35%) | 20 (9.95%) | 33 (16.42%) | 87 (43.28%) |
| ≥ **6X** | 169 | 54 (31.95%) | 15 (8.88%) | 33 (19.53%) | 67 (39.64%) |
| ≥ **7X** | 153 | 53 (34.64%) | 15 (9.80%) | 29 (18.95%) | 56 (36.60%) |
| ≥ **8X** | 134 | 43 (32.09%) | 12 (8.96%) | 29 (21.64%) | 50 (37.31%) |
| ≥ **9X** | 123 | 38 (30.89%) | 15 (12.20%) | 25 (20.33%) | 45 (36.59%) |
| ≥ **10X** | 110 | 34 (30.91%) | 11 (10.00%) | 27 (24.55%) | 38 (34.55%) |

**B. non-dbSNPs**

| Coverage cutoffs | Total | By 1 | By 2 | By 3 | By 4 |
|---|---|---|---|---|---|
| ≥ **3X** | 402 | 151 (37.56%) | 99 (24.63%) | 76 (18.91%) | 76 (18.91%) |
| ≥ **4X** | 211 | 76 (36.02%) | 41 (19.43%) | 53 (25.12%) | 41 (19.43%) |
| ≥ **5X** | 161 | 57 (35.04%) | 30 (18.63%) | 37 (22.98%) | 37 (22.98%) |
| ≥ **6X** | 127 | 38 (29.92%) | 27 (21.26%) | 29 (22.83%) | 33 (25.98%) |
| ≥ **7X** | 106 | 33 (31.13%) | 21 (19.81%) | 22 (20.75%) | 30 (28.30%) |
| ≥ **8X** | 93 | 32 (34.41%) | 17 (18.28%) | 22 (23.66%) | 22 (23.66%) |
| ≥ **9X** | 87 | 28 (32.18%) | 16 (18.39%) | 23 (26.44%) | 20 (22.99%) |
| ≥ **10X** | 79 | 25 (31.65%) | 18 (22.78%) | 20 (25.32%) | 16 (20.25%) |

**Table 3-13. Positive calling rate and sensitivity.**

For a specific calling program (e.g., SOAPsnp), A is the number of SNVs identified as an empirical truth (i.e., called by at least 3 calling programs) and also called by this calling program; B is the number of SNVs identified as an empirical truth, but not called by this calling program; C is the number of SNVs called by this calling program, but is not an empirical truth. Positive calling rate is calculated as A/(A+B); sensitivity is calculated as A/(A+C).

| | | **Empirical truth** | |
|---|---|---|---|
| | | SNV | Not SNV |
| **Program's calling results** | called as SNV | A | B |
| | called as Reference (i.e., not SNV) | C | -- |

**Table 3-14. Positive calling rates of the four algorithms with different coverage cutoffs.** (A.) dbSNPs. (B.) non-dbSNPs.

**A. dbSNPs**

| **Coverage cutoffs** | **SOAPsnp** | **Atlas-SNP2** | **GATK-UGT** | **SAMtools** |
|---|---|---|---|---|
| **≥ 3X** | 0.734 | 0.888 | 0.902 | 0.892 |
| **≥ 4X** | 0.735 | 0.867 | 0.882 | 0.868 |
| **≥ 5X** | 0.704 | 0.841 | 0.874 | 0.876 |
| **≥ 6X** | 0.723 | 0.819 | 0.867 | 0.870 |
| **≥ 7X** | 0.696 | 0.808 | 0.817 | 0.862 |
| **≥ 8X** | 0.747 | 0.796 | 0.864 | 0.852 |
| **≥ 9X** | 0.727 | 0.782 | 0.813 | 0.849 |
| **≥ 10X** | 0.769 | 0.780 | 0.862 | 0.828 |

**B. non-dbSNPs**

| Coverage cutoffs | SOAPsnp | Atlas-SNP2 | GATK-UGT | SAMtools |
|---|---|---|---|---|
| ≥ 3X | 0.438 | 0.863 | 0.628 | 0.628 |
| ≥ 4X | 0.545 | 0.896 | 0.713 | 0.615 |
| ≥ 5X | 0.602 | 0.875 | 0.708 | 0.610 |
| ≥ 6X | 0.641 | 0.873 | 0.691 | 0.627 |
| ≥ 7X | 0.620 | 0.852 | 0.730 | 0.672 |
| ≥ 8X | 0.594 | 0.827 | 0.736 | 0.674 |
| ≥ 9X | 0.615 | 0.824 | 0.745 | 0.690 |
| ≥ 10X | 0.559 | 0.818 | 0.674 | 0.667 |

**Table 3-15. Sensitivity of the four algorithms with different coverage cutoffs.** (A.) dbSNPs. (B.) non-dbSNPs.

**A. dbSNPs**

| Coverage cutoffs | SOAPsnp | Atlas-SNP2 | GATK-UGT | SAMtools |
|---|---|---|---|---|
| **≥ 3X** | 0.980 | 0.868 | 0.958 | 0.883 |
| **≥ 4X** | 0.960 | 0.960 | 0.938 | 0.858 |
| **≥ 5X** | 0.950 | 0.967 | 0.925 | 0.883 |
| **≥ 6X** | 0.940 | 0.950 | 0.910 | 0.870 |
| **≥ 7X** | 0.941 | 0.941 | 0.894 | 0.882 |
| **≥ 8X** | 0.937 | 0.937 | 0.886 | 0.873 |
| **≥ 9X** | 0.914 | 0.971 | 0.871 | 0.886 |
| **≥ 10X** | 0.923 | 0.985 | 0.862 | 0.815 |

**B. non-dbSNPs**

| Coverage cutoffs | SOAPsnp | Atlas-SNP2 | GATK-UGT | SAMtools |
|---|---|---|---|---|
| **≥ 3X** | 0.912 | 0.546 | 0.837 | 0.570 |
| **≥ 4X** | 0.968 | 0.915 | 0.926 | 0.628 |
| **≥ 5X** | 0.959 | 0.946 | 0.919 | 0.676 |
| **≥ 6X** | 0.952 | 1.000 | 0.903 | 0.677 |
| **≥ 7X** | 0.942 | 1.000 | 0.885 | 0.750 |
| **≥ 8X** | 0.932 | 0.977 | 0.886 | 0.705 |
| **≥ 9X** | 0.930 | 0.977 | 0.884 | 0.674 |
| **≥ 10X** | 0.917 | 1.000 | 0.861 | 0.667 |

# CHAPTER 4: IDENTIFYING DIFFERENTIAL METHYLATION USING A HIDDEN MARKOV MODEL

## 4.1 Introduction

DNA methylation is an epigenetic process that adds a methyl group to the 5' position of cytosine at a CG dinucleotide (i.e., a cytosine is located next to a guanine nucleotide). Differentially methylated regions (DMRs) can serve as novel biomarkers for disease diagnosis and potential targets for demethylation drug development in cancer studies (Strathdee and Brown, 2002; Wei, et al., 2003). Therefore, in recent studies, identification of DMRs has received more and more attention. To identify differential methylation patterns between any two groups of samples, it is essential to obtain methylation signals at the single CG site level. A commonly used technology that measures methylation at the single CG site level is high-throughput bisulfite sequencing, which combines bisulfite treatment with next-generation sequencing to provide single base, quantitative methylation signals. First, sodium bisulfite treatment specifically converts unmethylated cytosine to uracil (later read as thymine), leaving the methylated cytosine unaffected (Krueger, et al., 2012). Then this change is measured by next-generation sequencing, such as whole-genome bisulfite sequencing (WGBS) and targeted bisulfite sequencing (Meissner, et al., 2008). DNA sequencing reads may be subsequently mapped via bisulfite-conversion-aware aligners, such as BRAT (Harris, et al., 2010), Bismark (Krueger and Andrews, 2011), BS Seeker (Chen, et al., 2010), BISMA (Rohde, et al., 2010), SAAP-RRBS (Sun, et al., 2012), BSMAP (Xi and Li, 2009), PASS-bis (Campagna, et al., 2013), and RRBSMAP (Xi, et al., 2012). For each sequenced CG site,

these aligners generate the total number of cytosine (C) and thymine (T) residuals aligned to each position along genomic DNA sequences, and the methylation signal of a specific CG site is calculated as C/(C+T). With genome-wide methylation signals measured at the single CG level, the detection and analysis of DMRs with fine resolution become possible.

As bisulfite sequencing technologies have been widely used, a few computational tools for DMR identification have been developed. The common approach taken by DNA methylation studies is to perform Fisher's exact test (Bock, et al., 2012; Gu, et al., 2010; Li, et al., 2010; Lister, et al., 2011; Stockwell, et al., 2014.), or to set arbitrary cutoffs for differential methylation in large sliding windows (Laurent, et al., 2010; Lister, et al., 2009). These methods are fairly straightforward, but they fail to take into account two important features of DNA methylation. First, the methylation levels of neighbor CG sites are spatially correlated and can change within hundreds of base pairs (bps) (Eckhardt, et al., 2006). Second, there is a large variation of methylation signals across samples within the same biological group, especially in cancer samples. Recently, a few statistical algorithms accounting for some of these features have been published. For example, methylKit, an R package for the analysis of RRBS data (Akalin, et al., 2012), considers the spatial dependence of methylation in multiple hypothesis testing. It first models the methylation differences between two groups for each CG site using a logistic regression, and then corrects the multiple hypothesis testing with a sliding linear model (Wang, et al., 2011). In addition, BSmooth, a pipeline to detect differentially methylated regions in whole-genome bisulfite sequencing data, has been developed by Hansen *et al*. (Hansen, et al., 2012). BSmooth accounts for the spatial correlation by smoothing the

methylation signals via a local-likelihood estimation for each sample, and tests for group difference using a modified *t* test for each CG. Adjacent CG sites with absolute *t*-statistics above a certain threshold are defined as a DMR. Moreover, Biseq has been developed to identify DMRs in targeted bisulfite sequencing data (e.g., RRBS) (Hebestreit, et al., 2013). Instead of processing all sequenced CG sites, Biseq constrains the analysis to CG sites within CG clusters, which are defined as target regions enriched with frequently covered CG sites. Group differences are tested within these target regions using smoothed methylation signals, and later the significant target regions are trimmed to obtain differentially methylated regions.

Although the above algorithms can handle the common issues in DNA methylation to some extent, they have certain limitations. For example, most of them are primarily designed for either whole-genome BS only or targeted BS data only, but not for both data types. In addition, the default parameter settings may not be suitable for a specific dataset, while the parameters defined by users can largely influence the accuracy of analysis results. In particular, the length of identified DMRs is sensitive to the choice of smoothing window size, and thus a wider window usually lowers the sensitivity for small DMRs. Moreover, most of these algorithms are designed for testing differences between normal and cancer specimens, and the variation across samples at a single CG site is not well accounted for. However, the cross-sample variation of methylation level is usually large in cancer samples. Therefore, when comparing different cancer samples or tissues, it is difficult for these methods to handle the DM CG sites with large within group variation.

To address the above challenges in DMR identification, we propose a hidden Markov model-based method (HMM-DM) that can detect DMRs from bisulfite sequencing data generated based on different protocols. The HMM-DM approach first identifies differentially methylated CG sites accounting for spatial correlation across CG sites and variation across samples, and then summarizes adjacent DM CG sites into DM regions. The main methodological contributions of HMM-DM are: 1) it can robustly identify DMRs with various lengths and DM singletons; 2) methylation variation across samples in the same group is well accounted for; and 3) it is suitable for both whole genome and targeted bisulfite methylation sequencing data. We demonstrate the advantages of HMM-DM by applying it to simulated data and comparing with BSmooth. We also apply HMM-DM to a published breast cancer dataset and report our findings.

## 4.2 Methods

### 4.2.1 Real methylation data

To train and test our model, we use publicly available DNA methylation sequencing data (*GSE27003*) (Sun, et al., 2011) generated using the Reduced Representation Bisulfite Sequencing (RRBS) protocol (Gu, et al., 2010; Gu, et al., 2011) from eight breast cancer cell lines, including four estrogen receptor positive (ER+) and four negative (ER-) samples. We then use the software package BRAT (Harris, et al., 2010) to trim off bases with low quality from both ends of the reads and to align reads afterwards. Methylation levels are obtained for all CG sites in eight samples using the BRAT acgt-count function. After removing CG sites with extremely low methylation coverage, 77,822 CG sites from chromosome 1 are used for further analysis.

### 4.2.2 Simulation data

Because methylation patterns in real samples are complex and difficult to simulate, all DMRs are simulated based on methylation levels and variation statuses of the "control group" of a real dataset. In particular, we take the first 10,000 CG sites of the four ER+ samples from the data described earlier as a control group, and the same 10,000 CG sites of the four ER- samples as a test group. In order to mimic the true methylation signals and variation in real sequencing data, the methylation levels of the test group are simulated using the control group as a background. Specifically, DMRs in the test group are obtained by adding differential methylation signals with various lengths and intensities to the background. Simulated DMRs are generated this way to preserve the natural changes in methylation patterns across CG sites and the variation patterns among samples. First, CG sites are categorized into four region types based on their methylation levels and variation statuses in the control group: H, High-methylation regions with small between sample variation; L, low-methylation regions with small between sample variation; M-H, High-methylation regions with large between sample variation; and M-L, low-methylation regions with large between sample variation. This step generates 2459 regions. Second, from the regions generated above, we randomly choose 80 DMRs with various methylation statuses and sizes (1 - 76 CG sites) to create methylation differences. These DMRs cover a total of 929 differentially methylated CG sites. Third, methylation levels for the test group in these DMRs are sampled from uniform distributions. Since the region types are defined based on the control group, to create a contrast we simulate test samples with lower methylation levels for H and M-H DMRs and with higher methylation levels for L and M-L DMRs. In addition, to ensure a true difference in

DMRs with larger variation and/or smaller size, larger contrasts are created between the two groups for H-M and H-L DMRs and DMRs with ≤ 3 CG sites.

### 4.2.3 Hidden Markov model

A hidden Markov model consists of hidden states and observed data. The sequence of hidden states is modeled by a Markov process, where the probability of the present state only depends on the previous state and all other preceding states are irrelevant. To identify differentially methylated regions, we design a first-order hidden Markov model assuming that the hidden differential methylation state at each CG site depends on the differential methylation state of the immediately preceding CG site. To build a hidden Markov chain that integrates information from all samples in two groups, we first define observations and hidden states.

#### *4.2.3.1 Observations*

Observations are a *P*x*L* matrix of observed methylation signals/ratios for sample *p* at CG site *i*,

$$O = \{o_1, o_2, ..., o_p, ..., o_P\}, \text{ with } O_p = \{o_{p,1}, o_{p,2}, ..., o_{p,L}\}.$$

Each observation ranges from 0 to 1, since the methylation signal at each CG site is calculated as the ratio of the number of reads with a methylated C to the total number of reads covering this CG site.

#### *4.2.3.2 Hidden States*

We use $h_i$ to denote the hidden differential methylation state at the $i^{th}$ CG site (Figure 4-1). There are three possible hidden states for each CG, Hyper (hypermethylated), EM (equally methylated), and Hypo (hypomethylated). EM corresponds to the situation where all samples from two different groups have similar methylation levels. Hyper represents the situation where samples in group 1 generally have higher methylation levels than samples in group 2, while Hypo represents the reverse pattern. The initial probabilities of three hidden states are:

$$P(h_1 = Hyper) = P(h_1 = EM) = P(h_1 = Hypo) = 1/3 .$$

With observations and hidden states established, the probability of the observed data is derived as

$$\begin{aligned} P\left(o_{1,1},....,o_{L,P}\right) &= P\left(o_{1,1},....,o_{L,P} \mid h_1,...,h_L\right) \cdot P(h_1,...,h_L) \\ &= \left[\prod_{p=1}^{P}\prod_{i=1}^{L} P\left(o_{pi} \mid h_i\right)\right] \cdot \left[P\left(h_1\right)\prod_{i=2}^{L} P(h_i \mid h_{i-1})\right] \cdot \end{aligned}$$

The other key features of HMM are described below.

#### *4.2.3.3 Transition probabilities*

The transition probability describes the probability of transferring from one differential methylation state to another between any two consecutive CG sites:

$$P\left(h_i \mid h_{i-1}\right) = t_{h_{i-1},h_i} \text{ , where } i=2,\ldots,L.$$

We use a vector $\theta^t = \{t_{j,k}\} = \{t_{1,1}, t_{1,2}, t_{1,3}, t_{2,1}, t_{2,2}, t_{2,3}, t_{3,1}, t_{3,2}, t_{3,3}\}$ to denote the transition probabilities between two states, where *j* and *k* are hidden states of two consecutive CG sites, respectively (Table 4-1).

#### *4.2.3.4 Transition probabilities*

Emission probabilities model the probability of observing methylation level at a CG site given a differential methylation state. For a given CG site, if there are similar methylation levels between two groups (EM), we consider the two groups as P independent samples and assume that their methylation levels follow the same Beta distribution. Alternatively, if there is differential methylation between two groups, then ($o_{1i}$,…, $o_{Mi}$) from group1 are M independent samples whose methylation levels follow a Beta distribution with a specific shape, while ($o_{M+1,i}$,…, $o_{Pi}$) from group 2 are P − M independent samples follow a Beta distribution with a different shape. Therefore, the distribution of $o_{pi}$conditional on $h_i$ is

$$o_{p,i} \mid h_i, \theta_i^e \sim \begin{cases} \begin{cases} Beta(a_{i1}, 1) & for\ h_i = \text{Hyper}, p = 1, 2, ..., \text{M} \\ Beta(1, a_{i2}) & for\ h_i = \text{Hyper}, p = \text{M}+1, \text{M}+2, ..., \text{P} \end{cases} \\ Beta(a_{i3,} a_{i4}) \quad for\ h_i = \text{EM}, p = 1, 2, ..., \text{P} \\ \begin{cases} Beta(1, a_{i5}) & for\ h_i = \text{Hypo}, p = 1, 2, ..., \text{M} \\ Beta(a_{i6}, 1) & for\ h_i = \text{Hypo}, p = \text{M}+1, \text{M}+2, ..., \text{P} \end{cases} \end{cases},$$

where $\theta_i^e = (a_{i1}, a_{i2}, a_{i3}, a_{i4}, a_{i5}, a_{i6})$ at CG site *i*.

### 4.2.4 Estimating parameters

To estimate the parameters of HMM-DM, we use Bayesian methods by giving priors to unknown parameters and derive their conditional probabilities (posterior probabilities). All parameters are then sampled from their conditional probabilities.

#### *4.2.4.1 Transition probability parameters $\theta^t$*

We count the numbers of transitions $y_{j,k}$ in the inferred states that fall into each category in Table 4-1, where $\sum_{j=1}^{3}\sum_{k=1}^{3} y_{j,k} = L-1$. For example, $y_{1,2} = 100$ means that 100 CG sites change their differential methylation state from "Hyper" to "EM". Let each $(y_{j,1}, y_{j,2}, y_{j,3})$ follow a multinomial distribution,

$$\begin{cases} (y_{1,1}, y_{1,2}, y_{1,3}) \sim MultiNomial(y_{1,1}+y_{1,2}+y_{1,3},\ t_{1,1},\ t_{1,2},\ t_{1,3}) \\ (y_{2,1}, y_{2,2}, y_{2,3}) \sim MultiNomial(y_{2,1}+y_{2,2}+y_{2,3},\ t_{2,1},\ t_{2,2},\ t_{2,3}) \\ (y_{3,1}, y_{3,2}, y_{3,3}) \sim MultiNomial(y_{3,1}+y_{3,2}+y_{3,3},\ t_{3,1},\ t_{3,2},\ t_{3,3}) \end{cases} \quad (1).$$

Let the prior of each $(t_{j,1}, t_{j,2}, t_{j,3})$ follow a Dirichlet distribution,

$$\begin{cases} (t_{1,1}, t_{1,2}, t_{1,3}) \sim Dirichlet(\alpha_1, \alpha_2, \alpha_3) \\ (t_{2,1}, t_{2,2}, t_{2,3}) \sim Dirichlet(\alpha_4, \alpha_5, \alpha_6) \\ (t_{3,1}, t_{3,2}, t_{3,3}) \sim Dirichlet(\alpha_7, \alpha_8, \alpha_9) \end{cases} \quad (2).$$

Thus, from (1) and (2), the posterior probabilities of $\theta^t$ become,

$$\begin{cases} (t_{1,1}, t_{1,2}, t_{1,3} \mid y_{1,1}, y_{1,2}, y_{1,3}) \sim Dirichlet(\alpha_1+y_{1,1},\ \alpha_2+y_{1,2},\ \alpha_3+y_{1,3}) \\ (t_{2,1}, t_{2,2}, t_{2,3} \mid y_{2,1}, y_{2,2}, y_{2,3}) \sim Dirichlet(\alpha_4+y_{2,1},\ \alpha_5+y_{2,2},\ \alpha_6+y_{2,3}) \\ (t_{3,1}, t_{3,2}, t_{3,3} \mid y_{3,1}, y_{3,2}, y_{3,3}) \sim Dirichlet(\alpha_7+y_{3,1},\ \alpha_8+y_{3,2},\ \alpha_9+y_{3,3}) \end{cases},$$

which allows us to estimate the transition probabilities easily by sampling directly from Dirichelet distributions.

#### *4.2.4.2 Emission probability parameters $\theta^e$*

The prior distribution of $\theta_i^e$ at the $i^{\text{th}}$ CG site is modeled as shown below,

$$
\begin{aligned}
&a_{i1} \sim Uniform\left[u_{1,} v_1\right] \quad for\ h_i = \text{Hyper}, p = 1,2,...,\text{M} \\
&a_{i2} \sim Uniform\left[u_{2,} v_2\right] \quad for\ h_i = \text{Hyper}, p = \text{M}+1,\text{M}+2,...,\text{P} \\
&\varphi_i = a_{i3}/(a_{i3}+a_{i4}) \sim Beta(a_0, b_0) \\
&\gamma_i = a_{i3}+a_{i4} \sim Uniform(m,n) \qquad for\ h_i = EM, p = 1,2,...,\text{P} \\
&a_{i5} \sim Uniform\left[u_{3,} v_3\right] \quad for\ h_i = \text{Hypo}, p = 1,2,...,\text{M} \\
&a_{i6} \sim Uniform\left[u_{4,} v_4\right] \quad for\ h_i = \text{Hypo}, p = \text{M}+1,\text{M}+2,...,\text{P}
\end{aligned}
$$

Instead of using fixed values, we assign hyperpriors to the parameters of the distribution of $\theta_i^e$. For EM states, all hyperpriors are set to ensure no limitation on the shape of the emission Beta distribution, which allows us to model EM states with various methylation levels. For Hypo and Hyper states, $u$ and $v$ are set to limit the shape of the Beta distribution within a certain range, such that the two groups have different methylation levels. In particular, for Hyper states, where the samples in group 1 have higher methylation levels than group 2, $u_1$ and $v_1$ will be set to ensure a relatively higher mean in $Beta(a_{i1},1)$, and $u_2$, $v_2$ will be set to ensure a relatively lower mean in $Beta(1,a_{i2})$. A similar strategy is applied to Hypo states to ensure a lower mean for $Beta(1,a_{i5})$ and a higher mean for $Beta(a_{i6},1)$. With the above prior distributions, the conditional probabilities of $\theta_i^e$ given observed data and hidden states can be derived. Slice sampling (Neal, 2003) is then used to sample $\theta_i^e$ from the conditional probabilities.

### 4.2.5 Estimating differential methylation states

The differential methylation states of HMM are estimated by a Markov Chain Monte Carlo (MCMC) method. In particular, for a given CG site *i*, the Gibbs sampler (Gelfand and Smith, 1990) is used to sample the three hidden states from their conditional probability distribution:

$$P\left(h_i = k \mid O, h_1, ..., h_{i-1}, h_{i+1}, ..., h_L, \theta^e, \theta^t\right) \propto P\left(O, h_1, ..., h_{i-1}, h_i = k, h_{i+1}, ..., h_L, \theta^e, \theta^t\right)$$

$$\propto \left[\prod_{p=1}^{\mathrm{P}} P\left(O_{pi} \mid h_i = k, \theta^e\right)\right] \cdot \left[P\left(h_1\right) \cdot \prod_{j=2}^{L} P\left(h_j \mid h_{j-1}, \theta^t\right)\right] \cdot P(\theta^t) \cdot P(\theta^t)$$

$$\propto \begin{cases} \left[\prod_{p=1}^{\mathrm{P}} P\left(O_{pi} \mid h_i = k, \theta_1^e\right)\right] \cdot \left[P\left(h_1 = k\right) \cdot P\left(h_2 \mid h_1 = k, \theta^t\right)\right] & i = 1 \\ \left[\prod_{p=1}^{\mathrm{P}} P\left(O_{pi} \mid h_i = k, \theta_i^e\right)\right] \cdot \left[P\left(h_i = k \mid h_{i-1}, \theta^t\right) \cdot P\left(h_{i+1} \mid h_i = k, \theta^t\right)\right] & 1 < i < L \\ \left[\prod_{p=1}^{\mathrm{P}} P\left(O_{pi} \mid h_i = k, \theta_L^e\right)\right] \cdot \left[P\left(h_L = k \mid h_{L-1}, \theta^t\right)\right] & i = L \end{cases},$$

where *k*= “hyper”, “hypo”, or “EM”.

For each CG site, a posterior probability is provided for each of three possible states, showing the probability of the CG site being in one of the states. The state with highest posterior probability is assigned as the state of CG site *i*. To call a Hyper or Hypo CG site, we require the difference of mean methylation levels in two groups (mean difference) to be larger than a certain threshold (e.g., 0.3). This setting is to make sure the identified differentially methylated CG sites are biologically meaningful, that is, there is a measureable difference between the two groups.

### 4.2.6 Identifying differentially methylated regions

Each CG site is estimated by our hidden Markov model as equally methylated, hypermethylated, or hypomethylated. The hypermethylated and hypomethylated CG sites are called as differentially methylated (DM). We then summarize these DM CG sites into either single sites or regions with at least two CG sites. The adjacent DM CG sites are grouped into regions considering their differential methylation state, distance, and sequencing coverage. Only CG sites with the same states can be included in the same region. Therefore, this method generates two types of differentially methylated regions, hypermethylated regions and hypomethylated regions.

## 4.3 Results

### 4.3.1 Stimulation data

We apply the HMM-DM method to the simulated dataset described in the Methods section. With the cutoff of posterior probability set as $\geq 0.8$, we obtain 1068 identified DM CG sites, yielding a sensitivity of 97.74% and a specificity of 98.24%. Out of the 80 selected DMRs, 68 are completely identified, 7 are partially identified, and 5 singletons are not identified. An example of an identified DMR is illustrated in Figure 4-2. In this region, although the test group has a generally lower methylation level than the control group, large variation across the four control samples makes it difficult to identify by traditional methods. But HMM-DM successfully identifies all 15 CG sites within this 150 bp-long region as hypomethylated with posterior probabilities $\geq 0.9$, suggesting a high-confidence in calling DMRs that have large within group variation.

To illustrate the efficiency of HMM-DM in identifying differentially methylated CG sites, we first evaluate its overall performance under different cutoffs of the posterior probabilities. Since a higher posterior probability corresponds to a higher level of confidence, we expect that applying a relatively higher cutoff of posterior probability can filter out false positives and CG sites with weak DM signals. As shown in Table 4-2, the false positive rate decreases from 3.29% to 1.77% with the cutoff changing from 0.4 to 0.8. Over 90% of the identified true positive CG sites have posterior probabilities higher than 0.95 (data not shown). When we filter the results with different cutoffs, the sensitivity stays as high as 99% and drops slightly to 92% when only DM CG sites with posterior probability $\geq 0.95$ are considered as positives. This indicates that HMM-DM has a generally high sensitivity and accuracy in identifying DM CG sites.

We then further examine HMM-DM's sensitivity in detecting DMRs with different lengths (number of CG sites included in each region) and different within group variation (Figure 4-3). The 80 designed DMRs are separated into three categories based on their sizes: long DMRs with > 20 CG sites, median DMRs with 3 – 20 CG sites, and short DMRs with $\leq 2$ CG sites. For the CG sites within each of the 10 long DMRs, almost 100% of the designed DM CG sites are identified with high posterior probabilities. Only 1.8% of the CG sites are filtered out when the cutoff is set as high as $\geq 0.95$ (Figure 4-3, purple). For the median length DMRs, the sensitivity is 99.45% without any filtering. This number drops slightly for different cutoffs of posterior probability, but is still higher than 90% when the cutoff is increased to 0.95 (Figure 4-3, orange). As for the short DMRs, although HMM-DM shows a lower sensitivity than in the longer regions, over 80% of DM CG sites are identified with a posterior probability cutoff of $\geq 0.8$ (Figure 4-3,

blue), suggesting that HMM-DM is capable of detecting DMRs even when the differential signal occurs in rather small clusters. A similar pattern is shown in the analysis of DMRs with different variation levels: both small-variation DMRs (H and L regions) and large-variation DMRs (M-H and M-L regions) show high sensitivity for all cutoff values, with a slightly lower sensitivity in large-variation DMRs when the cutoff increases to 0.9 (Figure 4-3, green and red). All the above results indicate that although the variation levels and DMR sizes may influence HMM-DM to some extent, our method can accurately identify DM CG sites that occur in small clusters in the presence of large within group variation.

In order to further demonstrate our method, we compare it with the most commonly used and cited method BSmooth (Hansen, et al., 2012) using the simulated dataset. The parameters for the smoothing step of BSmooth are set to be comparable to HMM-DM: the minimum number of methylation loci in a smoothing window is set as 1, the minimum length of a smoothing window is set as 5, and the maximum gap between two methylation loci (before the smoothing is broken across the gap) is set as 100 bp. In the modified *t*-test step, all the 10,000 simulated CG sites are tested for methylation differences; the variance is estimated for the control group. Any CG with a modified *t*-statistics beyond a certain threshold is identified as a differentially methylated CG.

For all different *t*-statistic thresholds, the sensitivity and FPR are calculated for BSmooth (Table 4-3). Comparing HMM-DM with BSmooth, we find that HMM-DM achieves higher sensitivity than BSmooth and it has a much smaller false positive rate. In particular, in Table 4-4 we compare the HMM-DM results with posterior probability $\geq$ 0.8 to the BSmooth results with the *t*-statistic threshold 2.5. The HMM-DM result yields

a sensitivity of 97.74% (908 true positives) and a false positive rate of 1.77% (160 false positives), while the *t*-statistic threshold of 2.5 in BSmooth yields a much smaller sensitivity of 79.47% (729 true positives), and a much higher false positive rate of 10.12% (926 false positives). In addition, HMM-DM is more accurate in detecting DMRs with shorter length and larger within group variation. For the DMRs with 3-20 CG sites and with no more than 2 CG sites, HMM-DM achieves a sensitivity of 97.26%, and 81.48% respectively, while BSmooth identifies 76.64%, and 77.77% respectively. For the DMRs with relatively larger variation (M-H and M-L regions), HMM-DM detects 93.93% of the CG sites and BSmooth detects only 42.14%. In summary, HMM-DM is more powerful than BSmooth in identifying differentially methylated CG sites, without sacrificing the specificity.

### 4.3.2 Breast cancer data

To illustrate the application of our method, we apply HMM-DM to detect the differentially methylated CG sites between ER+ and ER- groups in a breast cancer sequencing dataset (Sun, et al., 2011). In chromosome 1, a total of 77,822 CG sites have been considered. To ensure that the detected DM CG sites have biological meaning rather than statistical significance alone, only CG sites with a mean difference (between the two groups) $\geq 0.3$ are identified as DM. CG sites in which ER- has higher methylation level compared to ER+ are defined as hypermethylated, and CG sites in which ER- has lower methylation level are defined as hypomethylated. We identify 2326 differentially methylated CG sites, forming 898 DM regions. The median length of these DMRs is 8 bp (minimum is 1 bp and maximum is 305 bp). 76.91% (1789) of the detected DM CG sites are hypermethylated in ER- group, while 23.09% (537) are hypomethylated. In addition,

all identified DM CG sites are categorized by their variation status within ER- and ER+ groups, respectively. The CG sites in which the 4 samples in one group all have methylation levels $\leq 0.4$, or all have methylation level $\geq 0.6$, are classified as small-variation; otherwise, the CG sites are classified as large-variation. The majority of identified DM CG sites have large variation either in one group (67%) or in both groups (31%), and only 2% of DM CG sites have small variation in both groups, suggesting HMM-DM is capable of identifying differentially methylated CG sites with various degrees of within group variation. Moreover, out of the 2326 DM CG sites detected by HMM-DM, there are 1577 CG sites covering a total of 236 genes either in the gene body (1296 CG sites) or in the promoter region (343 CG sites). Table 4-5 lists the top 10 genes that include the most DM CG sites. Majority of these genes show higher methylation levels in the ER- group, a breast cancer type that is more difficult to treat. In particular, there are five genes located in the 1p36 tumor suppressor region, suggesting a possible mechanism for the severity of the ER- condition.

## 4.4 Discussion

Our method has the following advantages. First, HMM-DM is not limited to any specific bisulfite sequencing protocol. It is suitable for detecting DMRs using data generated from both whole-genome bisulfite sequencing and targeted bisulfite sequencing. Second, HMM-DM has a finer resolution compared to BSmooth. Since it is a CG site-based approach, the changes of differential methylation pattern over short distances can be well captured. Therefore, HMM-DM allows users to fully benefit from the single-CG resolution of methylation measurement provided by the bisulfite sequencing technologies. Third, variation within the same biological group is taken account of. Beta distributions

can model methylation levels with different variation easily with different shape parameters. This property is particularly beneficial when dealing with cancer samples, where the between-sample variation is usually large. Fourth, HMM-DM has simple parameter settings for users, because all key parameters (e.g., the priors for transition and emission probabilities) are estimated from the data directly. After obtaining the raw results, users can choose the desired thresholds for the posterior probability and mean difference between groups for further filtering.

Our method also has certain limitation. For example, the current method doesn't directly incorporate coverage in the model. However, this limitation can be made up by performing quality control on coverage when preparing the input data. For example, before applying the HMM-DM to a dataset, CG sites with extremely low coverage are removed.

Distance between CG sites is considered in our model. We break the hidden Markov chain if the distance between two CG sites is too long. By considering the physical distance this way, we ensure the estimation of a CG's methylation status will not be influenced by CG sites that are far away. Moreover, when summarizing DM CG sites into DM regions, only DM CG sites that are close to each other (i.e., within 100 bp) are grouped into one region.

Our simulation data preserve the features of real bisulfite sequencing data. The simulated DMRs are not chosen by random cut, but based on the natural blocks of methylation status in real data. In real bisulfite sequencing data, adjacent CG sites tend to have similar methylation level and similar differential methylation status. It is relatively

less frequent to observe dramatic changes within hundreds of base pairs. Therefore, it is only proper to follow the natural change of methylation status, instead of creating differentially methylated blocks by arbitrary settings. As for the choice of simulating DMRs in the test group, besides uniform distributions we have also simulated another set of data using beta distributions. Since HMM-DM performs similarly in both settings, we only report results from the uniform distribution in this paper.

## 4.5 Conclusion

In this paper we have developed a hidden Markov model-based approach HMM-DM to detect DMRs from bisulfite sequencing data generated based on different protocols, such as whole-genome bisulfite sequencing and reduced representative bisulfite sequencing. The HMM-DM method is illustrated using both simulated and real datasets from breast cancer samples. The application to simulated data shows an increased power compared to the currently most commonly cited method BSmooth, especially in DMRs that are short and have large within group variation.

## References


Akalin, A., *et al.* (2012) methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles, *Genome biology*, **13**, R87.

Bock, C., *et al.* (2012) DNA Methylation Dynamics during In Vivo Differentiation of Blood and Skin Stem Cells, *Molecular Cell*, **47**, 633-647.

Campagna, D., *et al.* (2013) PASS-bis: a bisulfite aligner suitable for whole methylome analysis of Illumina and SOLiD reads, *Bioinformatics*, **29**, 268-270.

Chen, P., Cokus, S. and Pellegrini, M. (2010) BS Seeker: precise mapping for bisulfite sequencing, *BMC Bioinformatics*, **11**, 203.

Eckhardt, F., *et al.* (2006) DNA methylation profiling of human chromosomes 6, 20 and 22, *Nature Genetics*, **38**, 1378-1385.

Gelfand, A. and Smith, A. (1990) Sampling-Based Approaches to Calculating Marginal Densities, *Journal of the American Statistical Association*, **85**, 398-409.

Gu, H., *et al.* (2010) Genome-scale DNA methylation mapping of clinical samples at single-nucleotide resolution, *Nature Methods*, **7**, 133-136.

Gu, H., *et al.* (2011) Preparation of reduced representation bisulfite sequencing libraries for genome-scale DNA methylation profiling, *Nature Protocols*, **6**, 468-481.

Hansen, K.D, Langmead, B. and Irizarry, R. (2012) BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions, *Genome biology*, **13**, R83.

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

Hebestreit, K., Dugas, M. and Klein, H.-U. (2013) Detection of significantly differentially methylated regions in targeted bisulfite sequencing data, *Bioinformatics*, **29**, 1647-1653.

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

Krueger, F., *et al.* (2012) DNA methylome analysis using short bisulfite sequencing data, *Nature Methods*, **9**, 145 - 151.

Laurent, L., *et al.* (2010) Dynamic changes in the human methylome during differentiation, *Genome Research*, **20**, 320-331.

Li, Y., *et al.* (2010) The DNA methylome of human peripheral blood mononuclear cells, *PLoS biology*, **8**, e1000533.

Lister, R., *et al.* (2009) Human DNA methylomes at base resolution show widespread epigenomic differences, *Nature*, **462**, 315-322.

Lister, R., *et al.* (2011) Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells, *Nature*, **471**, 68-73.

Meissner, A., *et al.* (2008) Genome-scale DNA methylation maps of pluripotent and differentiated cells, *Nature*, **454**, 766-770.

Neal, R.M. (2003) Slice sampling, *The Annals of Statistics*, **31**, 705-767.

Rohde, C., *et al.* (2010) BISMA - Fast and accurate bisulfite sequencing data analysis of individual clones from unique and repetitive sequences, *BMC Bioinformatics*, **11**, 230.

Stockwell, P.A., *et al.* (2014.) DMAP: Differential Methylation Analysis Package for RRBS and WGBS data, *Bioinformatics advanced online publication*, doi:10.1093/bioinformatics/btu1126.

Strathdee, G. and Brown, R. (2002) Aberrant DNA methylation in cancer: potential clinical interventions, *Expert Reviews in Molecular Medicine*, **4**, 1-17.

Sun, Z., *et al.* (2011) Integrated Analysis of Gene Expression, CpG Island Methylation, and Gene Copy Number in Breast Cancer Cells by Deep Sequencing, *PLoS ONE*, **6**, e17490.

Sun, Z., *et al.* (2012) SAAP-RRBS: streamlined analysis and annotation pipeline for reduced representation bisulfite sequencing, *Bioinformatics*, **28**, 2180-2181.

Wang, H.-Q., Tuominen, L. and Tsai, C.-J. (2011) SLIM: a sliding linear model for estimating the proportion of true null hypotheses in datasets with dependence structures, *Bioinformatics*, **27**, 225-231.

Wei, S., Brown, R. and Huang, T. (2003) Aberrant DNA methylation in ovarian cancer: is there an epigenetic predisposition to drug response?, *Annals of the New York Academy of Sciences*, **983**, 243-250.

Xi, Y., *et al.* (2012) RRBSMAP: A Fast, Accurate and User-friendly Alignment Tool for Reduced Representation Bisulfite Sequencing, *Bioinformatics*, **28**, 430-432.

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

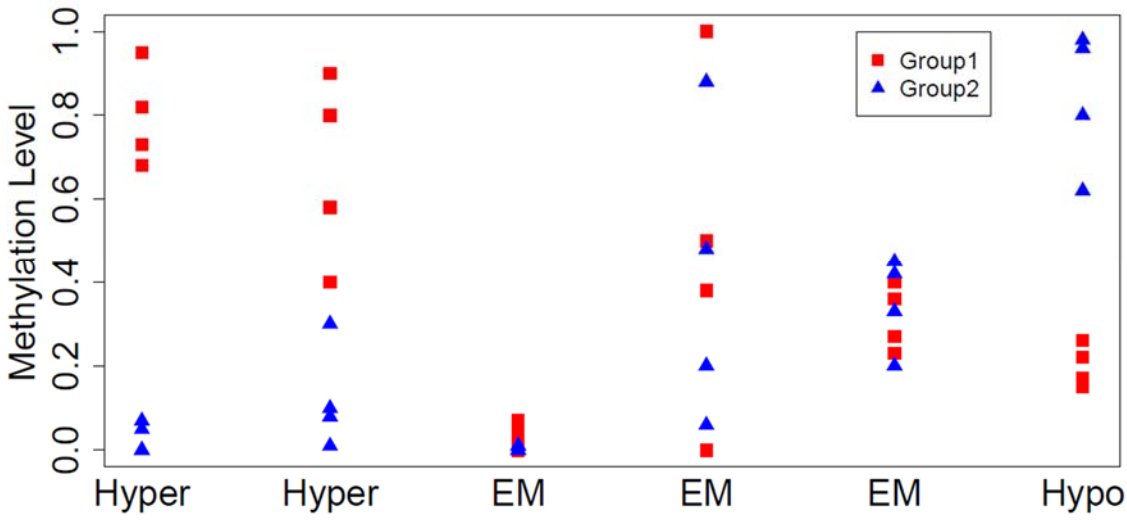



**Figure 4-1**

**An example of the hidden Markov model.** Observations are the methylation levels of six CG sites in groups 1 and 2, represented by red rectangles and blue triangles, respectively. The hidden states of six CG sites are denoted as “Hyper”, “EM”, and “Hypo”.

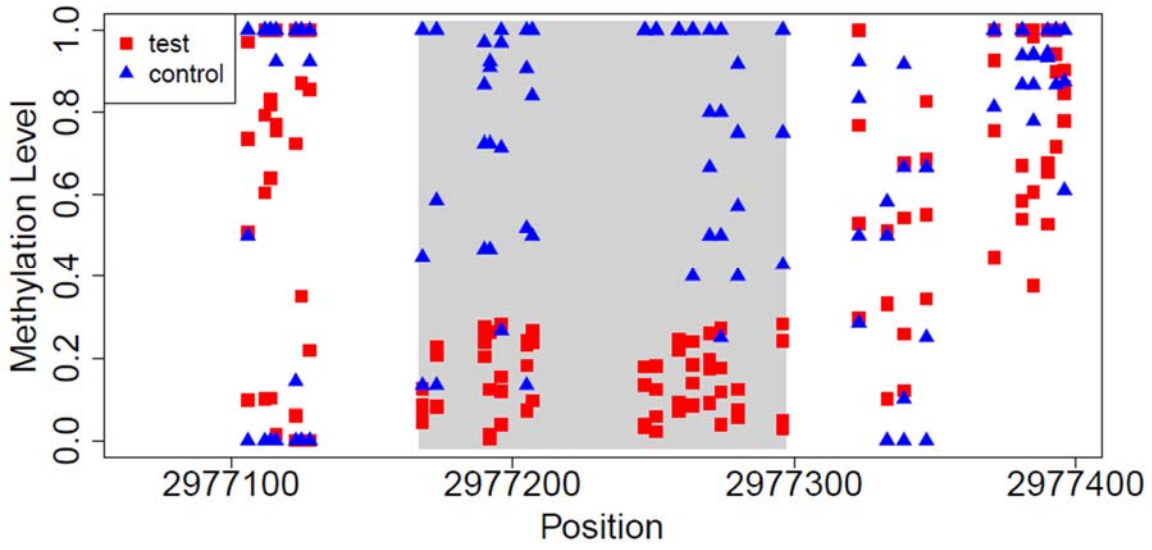



**Figure 4-2**

**A typical DMR.** The methylation levels of control and test groups are represented in red rectangles and blue triangles, respectively. The shaded box represents a simulated DMR identified by HMM-DM. All CG sites within this DMR are identified as hypo (hypomethylated in the test group) and the background CG sites are identified as EM (Equally methylated).

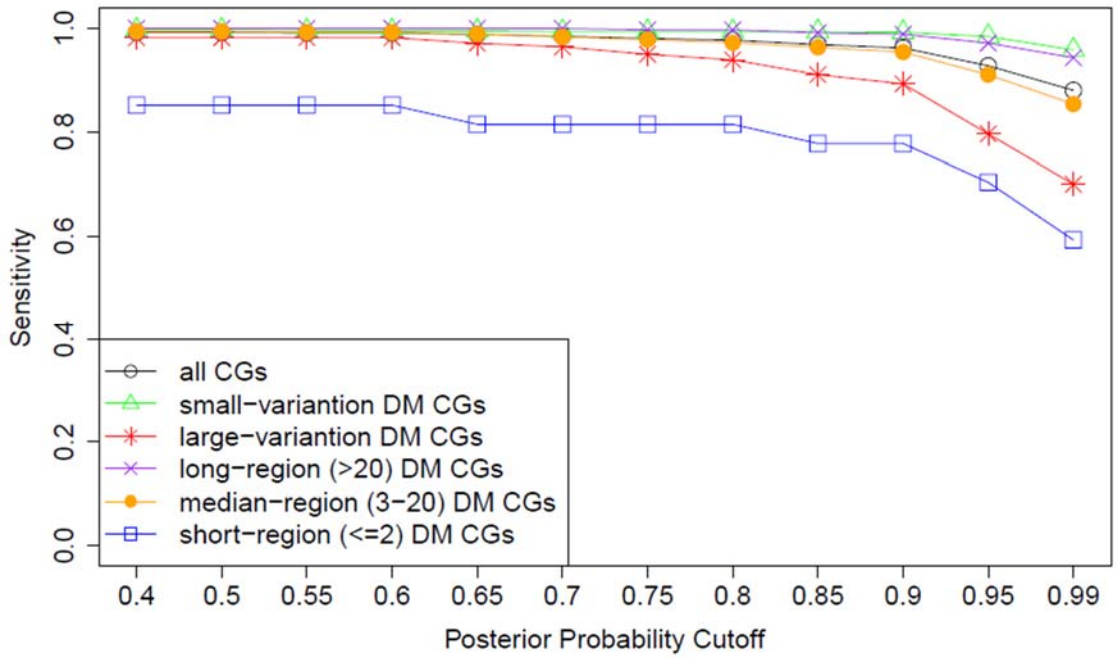



**Figure 4-3**

**Sensitivity of HMM-DM.** Shown are the overall sensitivity (black line) and the sensitivity to detect DMRs of different sizes and different within group variation (colored lines), under different posterior probability cutoffs. The larger the DMR size, and the lower the variation level, the higher the probability of identifying a CG site.

**Table 4-1. Transition probabilities between two adjacent states $h_{j-1}$ and $h_j$.** For any $j$=1,2,3, $t_{j,1} + t_{j,2} + t_{j,3} = 1$.

| $h_j = k$ / $h_{j-1} = j$ | 1 (Hyper) | 2 (EM) | 3 (Hypo) |
|---|---|---|---|
| **1 (Hyper)** | $t_{1,1}$ | $t_{1,2}$ | $t_{1,3}$ |
| **2 (EM)** | $t_{2,1}$ | $t_{2,2}$ | $t_{2,3}$ |
| **3 (Hypo)** | $t_{3,1}$ | $t_{3,2}$ | $t_{3,3}$ |

**Table 4-2. Sensitivity and FPR (%) of HMM-DM with different cutoffs of posterior probability.**

| **Cutoffs** | **0.4** | **0.5** | **0.55** | **0.6** | **0.65** | **0.7** |
|---|---|---|---|---|---|---|
| **Sensitivity** | 99.25 | 99.25 | 99.14 | 99.14 | 98.82 | 98.49 |
| **FPR** | 3.29 | 3.26 | 2.92 | 2.75 | 2.45 | 2.29 |
| **Cutoffs** | **0.75** | **0.8** | **0.85** | **0.9** | **0.95** | **0.99** |
| **Sensitivity** | 98.06 | 97.74 | 96.88 | 96.23 | 92.79 | 88.05 |
| **FPR** | 1.91 | 1.77 | 1.48 | 1.40 | 1.05 | 0.78 |

**Table 4-3. Sensitivity and FPR (%) of BSmooth with different cutoffs of posterior probability.**

| **Thresholds** | **1.5** | **1.6** | **1.8** | **2** | **2.5** | **3** |
|---|---|---|---|---|---|---|
| **Sensitivity** | 94.29 | 91.47 | 85.79 | 82.18 | 78.47 | 72.47 |
| **FPR** | 18.60 | 17.48 | 15.53 | 14.25 | 10.21 | 8.21 |
| **Thresholds** | **3.5** | **4** | **4.6** | **6** | **12** | |
| **Sensitivity** | 70.72 | 68.78 | 66.63 | 62.40 | 50.00 | |
| **FPR** | 6.93 | 5.99 | 5.14 | 4.20 | 3.33 | |

**Table 4-4. Comparing the performance of HMM-DM and BSmooth.**

| | HMM-DM | BSmooth |
|---|---|---|
| **Sensitivity , all DMRs** | 97.74% | 78.47% |
| **FPR, all DMRs** | 1.77% | 10.21% |
| **Sensitivity, DMR > 20** | 99.72% | 81.35% |
| **Sensitivity, DMR** | 97.26% | 76.64% |
| **Sensitivity, DMR ≤ 2** | 81.48% | 77.77% |
| **Sensitivity, small-variation DMR** | 99.38% | 89.98% |
| **Sensitivity, large-variation DMR** | 93.93% | 42.14% |

Shown are comparison results of HMM-DM with posterior probability ≥ 0.8 and BSmooth with modified *t*-statistics threshold of 2.5. Seven metrics are considered: the sensitivity and FPR for all simulated DMRs, sensitivity for DMRs with > 20 CG sites, DMRs with 3-20 CG sites, DMRs with ≤ 2 CG sites, as well as sensitivity for DMRs with small and large variation (see the first column).

**Table 4-5. Top 10 genes that include the most identified DM CG sites.**

| Location | Gene name | CG sites in gene body | CG sites in Promoter |
|---|---|---|---|
| 1p36.32 | AJAP1 | 67 (67/-) | 39 (39/-) |
| 1p34.3 | GRIK3 | 57 (37/20) | 44 (44/-) |
| 1p36.31 | CAMTA1 | 43 (25/18) | - |
| 1p36.23-p33 | PRDM16 | 33 (23/10) | - |
| 1p21 | LOC100129620 | 25 (25/-) | - |
| 1p21-22 | NTRK1 | 24 (21/3) | - |
| 1p13.2 | C1orf183 | 18 (18/-) | 5 (5/-) |
| 1p36.3 | GABRD | 21 (19/2) | - |
| 1p36.33 | RNF223 | 20 (20/-) | - |
| 1q42.13 | OBSCN | 19 (18/1) | - |

Shown are the location and name of the top 10 genes that include the most identified DM CG sites, and the number of CG sites in the gene body and promoter region. Number of hyper-methylated and hypo-methylated CG sites are shown in brackets, separated by a slash (/). "-" indicates that no hyper or hypo DM CG sites are identified in gene bodies or promoter regions.

# CHAPTER 5: COMPARING STATISTICAL METHODS FOR DIFFERENTIAL METHYLATION IDENTIFICATION USING BISULFITE SEQUENCING DATA

## 5.1 Introduction

DNA methylation is an important epigenetic modification that plays a key role in regulating gene expression (Baylin and Bestor, 2002; Gopalakrishnan, et al., 2008; Law and Jacobsen, 2010; Suzuki and Bird, 2008). Differential methylation patterns are usually observed between diseased and normal samples, tissues and specimens, and individuals from a population. A wide range of methylation studies have shown that some genomic regions are differentially methylated between normal and diseased specimens, as well as between different disease conditions (Eckhardt, et al., 2006; Hansen, et al., 2011; Irizarry, et al., 2009). Therefore, differentially methylated regions (DMRs) have been used as novel biomarkers for early detection and drug target identification of complex diseases such as cancers (Guzman, et al., 2012; Strathdee and Brown, 2002; Wei, et al., 2003).

Identifying differential methylation between two groups requires us to obtain methylation signals at each CG site for some genomic regions or the whole genome. At each CG site, bisulfite treatment can convert unmethylated cytosine to uracil (later read as thymine) while leaving the methylated cytosine unchanged (Krueger, et al., 2012). Therefore, bisulfite treatment combined with next-generation sequencing is a preferred method to measure methylation at single-base resolution. There are two

main types of bisulfite sequencing technologies, whole-genome bisulfite sequencing (WGBS) and reduced representation bisulfite sequencing (RRBS) (Meissner, et al., 2008). The former technique is comprehensive yet costly (Laird, 2003), while the latter is cost-effective but it only sequences the regions of genome with high GC contents. These technologies have been widely used to investigate DNA methylation patterns in human genomes (Hansen, et al., 2011; Lister, et al., 2009; Lister, et al., 2011; Sun, et al., 2011). Since bisulfite sequencing technologies can generate tremendous amount of data with complex biological features, great efforts have been made to process and analyze such large datasets. For example, to deal with the asymmetric mapping issues in bisulfite-converted reads, several alignment tools have been developed, including BSMAP (Xi and Li, 2009), BRAT (Harris, et al., 2010), BS Seeker (Chen, et al., 2010), BISMA (Rohde, et al., 2010), SAAP-RRBS (Sun, et al., 2012), Bismark (Krueger and Andrews, 2011), PASS-bis (Campagna, et al., 2013), and RRBSMAP (Xi, et al., 2012). Moreover, there are a few packages developed for the quality assessment of bisulfite sequencing data (Akalin, et al., 2012; Sun, et al., 2013; Sun, et al., 2012).

In addition to the alignment and quality assessing tools, several computational methods for DMR identification have been developed (Akalin, et al., 2012; Hansen, et al., 2012; Hebestreit, et al., 2013; Stockwell, et al., 2014.). For example, methylKit (Akalin, et al., 2012), an R package for the analysis of BS data, models the differential methylation between groups using a logistic regression. A sliding linear model (Wang, et al., 2011) is applied for converting p-values to q-values to correct for multiple testing. Another method, BSmooth (Hansen, et al., 2012), is a pipeline to

analyze WGBS data. It first smoothes the methylation level via a local likelihood estimation for each sample, and then tests for group differences using a modified *t*-test. In addition, BiSeq (Hebestreit, et al., 2013) identifies DMR in targeted BS data only, so that it constrains the analysis to CG clusters. A hierarchical testing procedure is then applied to test for DMRs within clusters and control the given false discovery rate (FDR). Moreover, two hidden Markov model (HMM)-based methods HMM-DM (Yu and Sun, 2014) and HMM-Fisher (Sun and Yu, 2014) have been recently developed by our group. The former estimates differential methylation status between two groups directly; while the latter estimates the methylation states as F (Fully methylated), P (Partly methylated), and N (Not methylated) for each sample with a HMM, and then tests for group differences using Fisher's exact test.

This chapter aims to systematically review and compare five DMR identification methods: methyKit, BSmooth, BiSeq, HMM-DM, and HMM-Fisher. Starting from the bisulfite sequencing data, these methods consist of six distinctive analysis aspects as shown in Figure 5-1. It is important to know that these analyses need not be performed in the particular order listed in Figure 5-1, and can be performed in different steps and in different ways. This chapter is organized as follows. We first review all five methods based on the six analysis aspects. Then we use simulated data to evaluate the performance of these five methods. Finally, we apply all five methods to real bisulfite treated methylation sequencing data and report the results of each method.

## 5.2 Methods

This section includes two parts. In Part I (subsection 5.2.1), we review the five DMR identification methods from six aspects: aligned bisulfite sequencing data format, quality control, smoothing, modeling, testing and defining DMRs, and further analysis. We summarize and compare their features. In Part II (subsection 5.2.2), we introduce the simulated and real datasets used to examine the methods, and describe the workflow of the comparison analysis.

### 5.2.1 Overview of DMR identification methods

Table 5-1 summarizes the main algorithms and basic functions used in each of the six analysis aspects for all five methods. In this section, we summarize these steps one by one.

#### *5.2.1.1 Aligned bisulfite sequencing data*

Bisulfite sequencing provides high throughput methylation data at the single base level. There are mainly two types of bisulfite sequencing protocols: whole-genome bisulfite sequencing (WGBS) that measures methylation levels for an entire genome, and targeted bisulfite sequencing (e.g., reduced representative bisulfite sequencing (Gu, et al., 2010; Gu, et al., 2011)) that reduces the complexity of the genome by sequencing the CG enriched regions using restriction enzymes and DNA fragment size selection. Among the five statistical methods developed for DMR identification, BSmooth is designed for methylation data from the WGBS protocol only and BiSeq is for target BS only, while methylKit, HMM-DM, and HMM-Fisher are not limited

to any specific protocol. For all these methods, bisulfite sequencing data need to be preprocessed by alignment tools to determine methylation signals. As for the format of input data, methylKit takes the total number of reads and the percent of methylated reads at each CG site, while the other four methods take the total number of reads and number of methylated reads at each CG site.

#### *5.2.1.2 Quality control*

Systematic sequencing errors and base-calling errors can affect the identification of DMRs and downstream analysis. Therefore, it is critical to perform quality control on raw methylation ratio data. As an important indicator of methylation data quality, coverage is commonly considered in the quality control step by most of the methods. Quality control need not be done only at the beginning of the analysis**.** In every step of DMR identification, quality controls have been conducted in various formats and in different degrees, which are summarized below for each method.

1) Before differential methylation detection, methylKit recommends users filter out CG sites with relatively high coverage to remove potential PCR bias. In addition, to avoid bias introduced by a systematically more sequenced sample, methylKit can normalize sequencing coverage among samples.
2) In BSmooth, the quality control step is performed in the modeling part. CG sites with low coverage or no coverage are removed from modeling and testing. The threshold for low coverage can be defined by users based on their own data.
3) BiSeq is a DMR identification method designed for targeted BS data. Therefore, it constrains the analysis to CG sites within CG clusters, which are regions with

higher coverage and higher density of CG sites. These clusters are detected using a three-step strategy. First, CG sites that are covered in the majority of samples (e.g., at least 75%) are defined as frequently covered CG sites. Second, it detects clusters within which the frequently covered CG sites are close to each other (e.g., at most 100 bp apart). Third, it retains only regions with a minimum number (e.g., 20) of frequently covered CG sites within the clusters.

4) For HMM-DM and HMM-Fisher, CG sites that are covered in only a minority of samples are removed; CG sites that are covered in the majority of samples but with only very low coverage are also removed.

#### *5.2.1.3 Smoothing*

Methylation levels of adjacent CG sites in a chromosome region tend to be similar (Eckhardt, et al., 2006). Therefore, a smoothing algorithm that borrows information from neighbors is appropriate in this context (Jaffe, et al., 2012). It not only reduces the required coverage, but also estimates methylation levels for the CG sites that are not covered by sequencing reads to avoid missing values. In addition, the falsely sequenced CG sites usually have low coverage and their methylation levels are dramatically different from their nearby sites; smoothing the methylation level can correct these sequencing errors to some extent, but it may introduce some bias.

To account for the spatial correlation of methylation levels, both BiSeq and BSmooth smooth the raw methylation data before detecting differential methylation. In particular, the raw methylation level is smoothed via a local likelihood function weighted on coverage and distance. For each CG site, let the methylation level $y$ =

$m/n$, where $m$ is the number of methylated reads and $n$ is the total number of reads. Then $m$ is modeled with a binomial distribution B($n$, $y$). Within a window of size $h$ around CG site $l$, CG sites are weighted by kernel functions, such that CG sites close to the CG site $l$ and with a high coverage are given high weight on the estimation of the methylation level at $l$. Despite that BSmooth and BiSeq use a similar algorithm to smooth methylation data, they are different as summarized below:

1) Since higher coverage gets higher weight, unusually high coverage can introduce bias into smoothing. Therefore, before the CG clusters are set up for smoothing, BiSeq limits the coverage, e.g., to the 90% quantile of all CG sites.
2) BSmooth performs smoothing on the entire chromosome for each sample, while BiSeq estimates each pre-defined CG cluster separately.
3) In BSmooth, the smoothing window size ($h$) defined by users is the minimum size and the actual bandwidth is enlarged until at least $h$ CG sites are included within the window. Therefore, the smoothing degree in BSmooth can be different for each sample and for each region. On the other hand, in BiSeq, $h$ defined by users is a fixed window size, such that the intensity of smoothing is uniform for each sample.
4) BSmooth employs a local logistic regression for smoothing, which can lead to the problem of extrapolation. For example, when there is a long region (L bp) without reads covered, the methylation level of a CG site can be predicted by covered CG sites that are L bp away, resulting in an over-estimated methylation level of 0 or 1.

It is important to note that the size of the smoothing window has a large impact on the smoothing step. On one hand, large bandwidth may lead to over-smoothing issues, such that real signals are smoothed away while false signals are introduced. On the other hand, small bandwidth may not do much smoothing at all. Therefore, proper smoothing window size is critical and should be chosen specifically for each dataset and each genomic region.

Other than smoothing, different ways are used to consider spatial correlation of methylation data. For example, HMM-Fisher and HMM-DM employ first-order Markov models. For a given CG site, these two models borrow methylation information from the previous CG site (see *Modeling*). In addition, to avoid inappropriate correlation between CG sites that are far away, the Markov chain in HMM-DM is broken if the distance between two consecutive CG sites is too large.

#### *5.2.1.4 Modeling*

Depending on the type of information that is modeled, the five methods can be grouped into three categories: modeling methylation levels, modeling methylation categories, and modeling differential methylation states directly.

*Modeling methylation levels*

In order to identify differential methylation between groups, methylKit, BSmooth, and BiSeq first model the methylation level for each CG site and then test for group differences. Detailed models and features of these methods are summarized below:

1) In methylKit, methylation level $y_i$ for sample $i$=1,..,n is modeled by a logistic regression: $\log\left(\frac{y_i}{1-y_i}\right) = \beta_0 + \beta_1 * x_i$, where $x_i = 1$ for the test group and $x_i = 0$ for the control group; $\beta_0$ is the log odds of the control group, and $\beta_1$ is the log odds ratio between the test and control groups. This logistic regression framework can be generalized to more than two groups or data types, with covariates incorporated into the model.
2) In BSmooth, the methylation level $y_{ij}$ for sample $i$ at location $j$ is assumed to be a sample-specific smooth function of genome location $l_j$, $f_i(l_j)$. Then a modified $t$-statistics $t(l_j)$ is formed, with the location-dependent standard deviation floored to the 75$^{th}$ percentile and smoothed using a running mean with a window size of 101. This $t(l_j)$ statistic is later used to test for methylation difference between groups.
3) Instead of estimating methylation levels for all CG sites, BiSeq constrains the analysis to CG sites within clusters defined in the quality control step. Within each cluster, the methylation level at each CG site, $y$, is modeled by a beta distribution with $E(y) = \mu$ and $Var(y) = \mu(1-\mu)/(1+\varphi)$, and the mean of methylation $y_j$ at position $j$ is modeled with a beta regression. Then the group difference is tested using the Wald test.

*Modeling methylation categories*

Instead of modeling methylation levels as continuous values, HMM-Fisher models methylation levels as categorical data for each sample separately. In the HMM step, hidden states $h$ are estimated as N (Not-methylated), P (Partly-

methylated), and F (Fully-methylated). At each CG site $j$, the transition between current state $h_j$ and the next state $h_{j+1}$ is allowed; and staying in the same state has a higher probability, while the transition between N and F has a lower probability. The emission probabilities (the probabilities of observing methylation levels $O_j$ given the hidden state of this CG site $h_j$) are modeled by the truncated normal distributions. Based on the biological meanings of the hidden states, the means of truncated normal distributions for N, P, and F states are set as 0, 0.5, and 1 respectively. Therefore, for each sample, the methylation state at each CG site is estimated as N, P, or F.

*Modeling differential methylation states*

All the four methods described above choose to model methylation levels first, and detect methylation differences afterward. Different from these methods, HMM-DM directly models the differential methylation between groups. Therefore, the hidden states are defined as Hyper (hypermethylated in the test group), Hypo (hypomethylated in the test group), and EM (equally methylated in both groups). The transition probabilities are estimated from the data using dirichlet distributions. As for the emission probabilities, HMM-DM uses beta distributions. For Hyper and Hypo states, at a given CG site $i$, two beta distributions with different means are used to model methylation levels of the control and test groups separately, to ensure differential methylation between the two groups. For the EM state, all samples from both control and test groups are modeled using the same distribution that assumes no differences between groups. Beta distributions are used for each CG site separately and all parameters are estimated from the data. Thus, the result of HMM-DM is the differential methylation status for each CG site – hyermethylated in the test group,

hypomethylated in the test group, or no differences between groups. Hypermethylated and hypomethylated CG sites with relatively large mean differences (e.g., ≥ 0.3) are defined as differentially methylated CG sites. Setting a relatively large mean difference is to ensure biological meaning of the identified CG sites.

#### *5.2.1.5 Testing and defining DMRs*

Depending on the testing strategies employed, the five methods can be grouped into three categories: controlling FDR, not controlling FDR, and not test-based.

*Controlling FDR*

For the analysis of either WGBS or RRBS data, the number of CG sites can go up to millions. Therefore, it is important to deal with the multiple testing issue (Storey, 2002; Storey and Tibshirani, 2003) in detecting differential methylation (Bock, 2012). Two methods, methylKit and BiSeq incorporate multiple testing corrections during their analysis. For methylKit, a sliding linear model (Wang, et al., 2011) is used to correct the p-values obtained from the logistic regression model to q-values. CG sites with associated q-values below a certain threshold and having large mean differences are defined as differentially methylated sites. As for BiSeq, a much more complex algorithm, two-step hierarchical testing (Benjamini and Heller, 2007) is used to correct for multiple testing. The first step is to detect CG clusters containing at least one differentially methylated location and to control a size-weighted FDR (WFDR) on clusters. To control the WFDR, the weighted Benjamini–Hochberg method (Benjamini and Hochberg, 1997) is applied on p-values of clusters, which are calculated from the p-values (Wald test in the Modeling step) for CG sites within the

clusters. Clusters with small p-values are selected and considered as candidates for DMRs. In the second step, the equally methylated CG sites within the selected CG clusters are removed and a location-wise FDR is controlled (Benjamini and Hochberg, 1997; Benjamini, et al., 2006). Therefore, the result of this hierarchical testing is a list of differentially methylated CG sites within clusters. The adjacent differentially methylated CG sites that locate within the same cluster and have the same direction of methylation differences are defined as one DMR. Thus, CG sites within a DMR are all hypermethylated or all hypomethylated.

*Not controlling FDR*

The other two test-based methods, BSmooth and HMM-Fisher, do not control FDR in their analysis. In BSmooth, after getting the statistics $t(l_j)$, DMRs are defined as groups of consecutive CG sites for which all $|t(l_j)| > c$, where $c$ is a positive cutoff selected based on the marginal empirical distribution of $t(l_j)$. In addition, CG sites with a large distance (e.g., 300 bp) are not allowed to be in the same DMR. In HMM-Fisher, the categorical data obtained from the HMM step is used to test for a group difference by Fisher's exact test. To better incorporate the information in neighboring CG sites and thus reduce the impact of small sample size and sequencing error, consecutive CG sites are combined if their distance is small (e.g., < 100 bp) and when the sample size is very small. CG sites with large mean differences and small p-values are then identified as differentially methylated CG sites. Finally, these identified CG sites are pooled into DMRs based on their p-values, distance, and the mean difference between groups.

*Not test-based*

HMM-DM is not a test-based method. The output results of hidden Markov models are the estimated differential methylation status of CG sites – hyper, EM, and hypo. To identify DMRs, the differentially methylated CG sites (hyper and hypo) detected from hidden Markov model are formed into regions based on their status, distance, posterior probability, and the mean difference between groups.

#### *5.2.1.6 Further analysis*

To understand the biological impact of differential methylation, the identified CG sites or regions should be put into genomic context for further analysis. All of the five methods provide tools for differential methylation visualizations and/or annotations.

Visual representation of data can be very useful for the interpretation of DMRs. Visualization tools for differential methylation can be divided into three types: 1) plots of the methylation levels for all samples with identified DMRs (BSmooth, BiSeq, HMM-Fisher, and HMM-DM); 2) summary statistics for DMRs, e.g., number of hyper- and hypo-methylation events per chromosome (methylKit); and 3) web-based genome browsers (UCSC Genome Browser and Integrated Genome Viewer) that allow users to browse methylation data along with its annotation (methylKit and BiSeq).

Annotating differential methylation regions can help to predict their functional impact and to find potential disease-related events for further analysis. Most methods

can annotate identified DMRs with CpG islands, CpG island shores, genes, and promoter regions (methylKit, BiSeq, HMM-DM, and HMM-Fisher). In addition, users may be interested in specific genomic regions that are related to certain diseases. Therefore, annotation can also be performed for customer-supplied regions in methylKit, HMM-Fisher, and HMM-DM.

#### *5.2.1.7 Summarizing key features*

Table 5-2 summarizes the key features of the five methods we have reviewed above. BSmooth and BiSeq are designed for specific BS protocol only, while the other three can be applied to both WGBS and target sequencing data. All five methods have corresponding R packages or pipelines available online. Coverage is considered to be an important indicator for sequencing quality and is used as a common criterion in the quality control step. Spatial correlation, the key characteristic of DNA methylation, is considered in most methods by borrowing information from neighboring CG sites. Two test-based methods, methylKit and BiSeq, intend to correct for multiple testing issues in DMR identification; DMR visualizations and genomic annotation tools are available in all five methods.

### 5.2.2 Datasets and comparison analysis

#### *5.2.2.1 Real methylation sequencing data*

To compare the five methods, we use publicly available DNA methylation sequencing data (*GSE27003*) (Sun, et al., 2011) generated using the Reduced Representation Bisulfite Sequencing (RRBS) protocol (Gu, et al., 2010; Gu, et al.,

2011) from eight breast cancer cell lines, including four estrogen receptor positive (ER+) and four negative (ER-) samples. We then use the software package BRAT (Harris, et al., 2010) to trim off bases with low quality from both ends of the reads and to align reads afterwards. Methylation levels are obtained for all CG sites in eight samples using the BRAT acgt-count function. After removing CG sites with extremely low methylation coverage, 77,822 CG sites from chromosome 1 are used for further analysis.

#### *5.2.2.2 Simulation data*

To mimic the complex DNA methylation patterns, all DMRs are simulated based on methylation levels and variation status of the "control group" of a real dataset. In particular, we take the first 10,000 CG sites of the four ER+ samples from the data described earlier as a control group, and the same 10,000 CG sites of the four ER- samples as a test group. For the test group, the methylation levels are simulated using the control group as a background. Specifically, DMRs in the test group are obtained by adding differential methylation signals with various lengths and intensities to the background. Simulated DMRs are generated this way to preserve the natural changes in methylation patterns across CG sites and the variation patterns among samples. The specific simulation procedure is explained as follows:

First, CG sites are categorized into five methylation classes based on their methylation level and variation status in the control group:

1) H (high methylation) where the methylation levels of all four control samples are ≥ 0.6, such that the within group variation is relatively small;

2) L (low methylation) where the methylation levels of all four control samples are ≤ 0.4, such that the within group variation is relatively small;

3) M (median methylation) where the mean of four control samples is within the range of (0.4, 0.6);

4) M-H (median-high methylation) where the mean is ≥ 0.6 but the variation across the four samples is relatively large compared to class H;

5) M-L (median-low methylation) where the mean is ≤ 0.4 but the variation across the four samples is relatively large compared to class L.

Second, based on the methylation classes, consecutive CG sites of the same class are grouped together, generating four types of regions: two types with small variation, H region and L region; and two types with large variation, M-H region and M-L region. The defined regions are further fine-tuned such that M class CG sites are allowed in M-H and M-L regions with low frequencies. This step generates 2459 methylation regions.

Third, from the regions generated above, we randomly choose 80 DMRs with various methylation statuses and sizes (1 - 76 CG sites) to create methylation differences. These DMRs cover 929 differentially methylated CG sites. Then, methylation levels for the test group in these DMRs are sampled from uniform distributions (Table 5-3). Since the region types are defined based on the control group, to create a contrast we simulate test samples with lower methylation levels for H and M-H DMRs and with higher methylation levels for L and M-L DMRs. In

addition, to ensure a true difference in DMRs with larger variation and/or smaller size, we use more stringent uniform distributions for H-M and H-L DMRs and DMRs with ≤ 3 CG sites (Table 5-3).

#### *5.2.2.3 Comparison analysis*

All five methods are compared by exploring the effect of parameter settings on the DMR identification results. We first use the default settings for each method, and then modify the settings based on the features of each method and the characteristics of the dataset. In order to compare the performance of the five methods, we then analyze their results using both simulated and real data.

For the simulated data, sensitivity and false positive rates are calculated for different cutoffs of statistics in each method, and the ROC curves are plotted accordingly. Moreover, the simulated DMRs are separated into classes based on their length and within group variation. In particular, the 80 simulated DMRs are separated into three classes based on their size: long DMRs with > 20 CG sites, median DMRs with 3 – 20 CG sites, and short DMRs with ≤ 2 CG sites. As for the variation levels, the 80 DMRs are grouped into two categories based on their within group variation: small-variation DMRs (H and L regions) and large-variation DMRs (M-H and M-L regions). Then the sensitivity for each class of DMRs are calculated and compared between methods.

For the real data, we compare the differentially methylated CG sites identified by each method and draw Venn diagrams to visualize the results. In addition, for the three methods that involve the estimation of methylation levels, we evaluate the effect

of their estimation by plotting the mean differences between groups for identified DM CG sites. Finally, to investigate the effect of coverage in estimation for the three methods, we plot their estimated mean differences vs. their raw mean differences for CG sites with different coverage cutoffs.

## 5.3 Results

### 5.3.1 Simulation data

#### *5.3.1.1 Default and modified Settings*

We first apply all methods to the simulated dataset with their default parameter settings (column 2 of Table 5-4) and cutoffs of statistics (column 2 of Table 5-5).

1) For MethylKit, the coverage of sequencing reads is normalized between samples to avoid bias introduced by systematically more sequenced sample; CG sites with q-statistics $\leq 0.01$ are considered to be differentially methylated sites.
2) For BSmooth, the minimum number of methylation loci in a smoothing window is set as 70; the minimum length of a smoothing window is set as 5; and the maximum gap between two methylation loci (i.e., before the smoothing is broken across the gap) is set as $10^8$ bp. In the modified *t*-test step, the variance is estimated for the control group. Any CG site with a statistics beyond 2 is identified as a differentially methylated CG.
3) For BiSeq, the analysis is first constrained to CG clusters with at least 20 CG sites, where the distance between any two CG sites within a cluster is $\leq 100$ bp, and then methylation levels of the CG sites within these clusters are smoothed

with a window of 80 bp. To define a differentially methylated region, the cluster-wise FPR is set at 0.1, and the CG-wise FPR is set at 0.05.

4) For HMM-DM, the differentially methylated CG sites are defined as the DM CG sites with posterior probabilities > 0.4.

5) For HMM-Fisher, the cutoff of p-value is set as 0.05 to identify DM CG sites.

Table 5-6A shows the number of identified DM CG sites, number of true positive (sensitivity), and number of false positive (false positive rate) in the five methods with default settings. MethylKit, HMM-DM, and HMM-Fisher all yield high sensitivity, while BSmooth and BiSeq show much lower sensitivity and low false positive rates. These differences are due to the cluster pattern of simulation data and the different degrees of spatial correlation each method incorporated. The simulation data are generated based on the real breast cancer dataset, where the CG sites form into relatively small clusters. In methylKit, the methylation level is estimated for each CG site separately; for HMM-DM and HMM-Fisher, the state of each CG site only depends on the previous one CG. Therefore, the estimated methylation levels or estimated DM patterns in these three methods are not heavily influenced by neighboring CG sites. However, in BSmooth and BiSeq, the smoothing windows are much larger (at least 70 CG sites for BSmooth and 80 bp for BiSeq); especially in BSmooth, the smoothing is only broken when the two consecutive CG sites are more than $10^8$ bp away. Therefore, the shorter differentially methylated regions can be easily underestimated. In addition, BiSeq constrains the analysis to CG clusters with relatively long length and high CG content, such that smaller clusters are left out from the testing for differential methylation.

For the purpose of a fair comparison, we modify the settings of BSmooth and BiSeq to be similar to the other methods (column 3 of Table 5-4). In particular, all clusters are used for analysis in BiSeq, and the smoothing window size is set to be much smaller (at least 5 CG sites in BSmooth and 25 bp in BiSeq). The parameter settings for methylKit, HMM-DM and HMM-Fisher stay the same as the default. For each method, sensitivity and false positive rates are calculated for different cutoffs of statistics. We then choose the cutoffs that yield relatively higher sensitivity and relatively lower false positive rate (column 3 in Table 5-5) and show their results in Table 5-6B. With the modified settings and the chosen cutoffs, all five methods identify a similar number of DM CG sites. In particular, although the sensitivity of methylKit only drops by 10% compared to the default settings (Table 5-6A), the number of false positive significantly decreases from 914 to 387. In BSmooth, both sensitivity and false positive rate are increased with the modified settings. Moreover, the number of DM CG sites called by BiSeq significantly increases from 491 (Table 5-6A) to 1234 (Table 5-6B), yielding a much higher sensitivity. In summary, the five methods perform better with the modified settings as shown in Table 5-6B. Therefore, we use the modified settings in all further analysis.

#### *5.3.1.2 Method comparison*

To compare the performance of the five approaches, we also show their ROC curves with modified settings in Figure 5-2. Because two FPR levels have to be chosen for BiSeq, we plot three ROC curves each with a fixed cluster-wise FPR (q) and different CG-wise FPRs (q2). In general, HMM-DM and HMM-Fisher achieve higher sensitivity than the others for false positive rates lower than 5%. Out of the

three q values chosen for BiSeq, q = 0.9 yields the highest sensitivity by sacrificing the false positive rate. MethylKit can achieve a sensitivity as high as 95% but with a false positive rate of almost 10%. Among all approaches, BSmooth shows the lowest sensitivity and the highest false positive rate. This is because BSmooth is more sensitive to the length than to the intensity of the differential methylation signal. Therefore, long regions that are only slightly different between the two groups (e.g., mean difference $\leq 0.05$) are ranked much higher than smaller regions with strong differential methylation signals.

Then for each approach we choose the "optimal" cutoff that shows relatively higher sensitivity and relatively lower false positive rate than other cutoffs in the ROC curve analysis (Figure 5-2, circle on each curve). We then compare the results of these "optimal" cutoffs in detail. The overall sensitivity and false positive rate of "optimal" cutoffs are shown in Table 5-6B. Among all methods, HMM-DM and HMM-Fisher achieve the highest sensitivity with the lowest FPR; while BSmooth yields the lowest sensitivity of 66.15% and the highest FPR of 5.13%. Table 5-7 depicts their sensitivity in DMRs with different lengths and variation levels. HMM-DM shows high sensitivity in all five classes of DMRs especially in DMRs with large variation; while BSmooth has the lowest sensitivity among all methods. Compared with the other DMRs, the DMRs with $\leq 2$ CG sites have much lower sensitivity in all five approaches. This can be explained by the fact that almost all approaches incorporate spatial correlation when identifying DM CG sites and regions, therefore small regions with one or two CG sites are more likely to be weighted out by their neighboring background CG sites. In particular, HMM-Fisher shows a relatively

lower sensitivity (66.67%) for small DMRs. This is probably because HMM-Fisher combines the neighboring CG sites in Fisher's exact test step, such that the signal of a single DM CG is very likely to be balanced out by the neighboring background CG sites. As for the variation types, all methods work well in regions with small within group variation in both groups, which is a relatively easy situation to identify DMRs. However, for the regions with large within group variation, BSmooth shows a much lower sensitivity of 10.35% compared to other methods and other situations.

### 5.3.2 Breast cancer data

We also compare the five approaches using the real breast cancer dataset mentioned in the Methods section. In chromosome 1, a total of 77,822 CG sites is considered. To ensure that the identified DM CG sites have biological meaning rather than statistical significance alone, only CG sites with mean differences $\geq 0.3$ are identified as DM. For methylKit, BSmooth, and Biseq, the mean difference is the difference of the estimated methylation levels; for HMM-DM and HMM-Fisher that do not estimate methylation levels directly, the mean difference is the difference of raw methylation levels. In addition, DM CG sites in which ER- has higher methylation level compared to ER+ are defined as hypermethylated, and DM CG sites in which ER- has lower methylation level are defined as hypomethylated. With the default settings (column 2 of Table 5-1) and default cutoff of statistics (column 2 of Table 5-2), the five approaches show dramatically different results. Then we use the modified settings (column 3 of Table 5-1) for further analysis. Posterior probability $> 0.4$ in HMM-DM and $p \leq 0.05$ in HMM-Fisher are used to define DM CG sites. The cutoff in BSmooth ($-1.8 \leq q \leq 1.8$) is chosen based on the plot of q-

statistics following the instruction of the BiSeq user manual. The cutoffs in methylKit ($q < 10^{-14}$) and BiSeq ($q = 0.5$, $q_2 = 0.99$) are chosen such that these two methods can get a similar number of DM CG sites as the others. Table 5-8 shows the number of DM CG sites (hyper- and hypomethylated) by each method, where the majority of DM sites have higher methylation in the ER- than the ER+ group. All methods identify around 2000 DM sites, except that BiSeq only identifies 766 DM sites.

Figure 5-3 shows the Venn diagrams comparing all approaches. Because BiSeq identifies significantly fewer DM CG sites than the others, we first compare the other four methods without BiSeq (Figure 5-3A). In total, 4752 DM CG sites are detected, with 12.96% detected by all four, 15.63% by any three, 19.97% by any two, and 51.44% by only one method. We then add BiSeq to the comparison (Figure 5-3B). The number of DM CG sites shared by all methods decreases from 616 to 387, while the percentages of DM CG sites identified by any two or only one method stay similar. In both the four-method and five-method comparisons, the methods show low concordance. This is probably because the five methods address differential methylation identification from different angles and employ different algorithms.

MethylKit, BSmooth, and BiSeq all use the estimated methylation levels to test for differential methylation, while HMM-DM and HMM-Fisher do not estimate the methylation level for each CG site. To investigate the effect of estimation, we plot the absolute value of the raw mean differences for the DM CG sites identified by each method in Figure 5-4. In HMM-DM and HMM-Fisher, all DM CG sites show mean difference $\geq 0.3$ since DM CG sites are defined based on the magnitude of the raw mean differences. For the other three methods, DM CG sites are required to have an

estimated mean difference ≥ 0.3. Therefore, this plot examines the agreement between estimated and raw mean differences for the identified DM CG sites. Both BSmooth and BiSeq smooth the methylation levels using local likelihood estimation incorporating the information of distance, coverage, and neighboring CG sites. BiSeq shows a similar pattern as HMM-DM and HMM-Fisher. Only 4.18% of identified the DM CG sites have mean differences less than 0.3, suggesting a good agreement between estimated and raw mean differences. However, 17.77% of DM CG sites in BSmooth have raw mean difference < 0.3 while their estimated mean differences are actually ≥ 0.3. The difference between BSmooth and Biseq may be because BSmooth has a larger smoothing effect, even though the smoothing window size is comparable in these two methods. In BiSeq the smoothing window size is fixed at 25 bp, while in BSmooth the given window size is a minimum size and can be enlarged to any number as long as the consecutive CG sites are within 100 bp (column 3 of Table 5-1). Among all the methods, the estimated mean differences of methylKit are most different from the raw mean differences. While all DM CG sites by methylKit have estimated mean differences ≥ 0.3, 32.79% of them show raw mean differences < 0.3. This is probably because methylKit estimates the methylation level for each CG separately, with only the coverage incorporated. In addition, this finding also suggests that even though the DM CG sites in methylKit are identified on the basis of statistical significance, a large percentage of them may not be real differential methylation signals.

Coverage is a factor that all three methods methylKit, BSmooth, and BiSeq consider when they estimate or smooth the methylation levels. CG sites with higher

coverage are usually given higher weight in the estimation. To check the effect of coverage in estimation for these three methods, we plot their estimated mean differences vs. their raw mean differences for CG sites with different coverages in Figure 5-5. As we expect, CG sites with higher coverage (≥ 30 X) shows a better agreement between estimated and raw values in all three methods. When comparing the three methods, the estimation of BiSeq has the best agreement with the raw data, while methylKit shows the lowest concordance between estimated and raw mean difference. This observation is consistent with our previous finding obtained based on how well the estimates can represent the raw data: BiSeq > BSmooth > methylKit.

## 5.4 Discussion

For the breast cancer data, BiSeq identifies many fewer DM CG sites than the other methods. In fact, the majority of CG sites fail the cluster-wise FDR control. Even with a large cluster-wise FDR of 0.9, only 2,596 out of the 77,822 CG sites are available for further analysis. This is probably because that FDR control can lead to a low sensitivity or a high false negative rate under certain circumstances (Pawitan, et al., 2005). There are at least two factors determining the FDR characteristics of a DMR detection study: (1) the proportion of truly differentially methylated CG sites and (2) the sample size. To guarantee a small FDR and a high sensitivity, there needs to be a large percentage of CG sites that are truly differentially methylated, as well as a large sample size. However, in the breast cancer data, less than 5% CG sites are identified as DM by the other methods, suggesting that only a small proportion is truly differentially methylated. Moreover, the sample size of the data is relatively small, with four samples in each group. Therefore, for a dataset with a higher

percentage of true DM or a larger sample size, BiSeq may yield a high sensitivity when a small FDR is controlled, as in the simulated data and dataset used in the BiSeq paper (Hebestreit, et al., 2013) .

To explore the effect of parameter settings in HMM-DM and HMM-Fisher, we also modify their parameters as we do with the other methods. In HMM-DM, key parameters, such as the prior for transition and emission probabilities, are estimated from the data directly. There are only two parameters that might need to be changed: (1) the number of CG sites to break the Markov chain and (2) the direchlet prior for transition probabilities. Similarly, in HMM-Fisher, there are only two parameters that might need to be modified: (1) the standard deviation of the truncated normal distribution of emission probabilities for the three states and (2) the dirichelet prior for the transition probabilities. Different settings of these parameters are applied to the two methods, and similar results are obtained. Therefore, we only report the results of the default settings in this chapter.

When comparing the five methods using simulated data, ROC curves are plotted with the y-axis ranging from 0.5 to 1. This is because all the five methods have sensitivity much higher than 0.5. Therefore, although the traditional ROC curves usually have a y-axis of 0 to 1, we use a smaller range to zoom in for better illustration.

As for the real breast cancer data analysis, the five methods show low concordance in the identified DM sites. This is probably due to several reasons. First, methylation sequencing is still a relatively new research area. Different resources

from both biological and technological aspects may contribute to its complexity. Second, to identify differential methylation, each method approaches the question from different angles and has its own features. Therefore, to choose a proper method for identifying DMRs in a specific dataset, we suggest that users select a method based on the characteristics of the data and the advantages of each method. In addition, for the purpose of validation and further analysis, users may first select the identified DMRs that are relatively long and have small within group variation to guarantee a high accuracy, and then move on to shorter regions and DM sites with larger within group variation.

## 5.5 Conclusion

In this chapter, we have provided a comprehensive comparison analysis of methods available for the identification of differential methylation in bisulfite sequencing data. First, it is important to explore the effect of parameter settings on the accuracy and efficiency of DMR identification. The simulation data analysis shows that the modified parameter settings can yield higher sensitivity and/or lower false positive rates, especially for methylKit, BSmooth, and BiSeq. Second, to compare the five methods, we have evaluated their performances in simulated DMRs with different length and within group variation. All five methods can better identify DMRs that are relatively long and have small within group variation. Among all methods, HMM-DM and HMM-Fisher exhibit relatively high sensitivity and low false positive rates, especially in DMRs with large within group variation. Third, we have compared the five methods using a real breast cancer dataset; however, a low concordance is observed. We have also investigated the effect of methylation

estimation. Our results show that among the three methods that involve methylation estimation, BiSeq can best present the raw methylation signals. Therefore, in view of the above findings, we recommend that users choose DMR identification methods based on the characteristics of the data and the different advantages that each method has. We also recommend that, when validating and further analyzing the identified DMRs, users choose long DMRs that have small within group variation as a priority.

## References


Akalin, A.*, et al.* (2012) methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles, *Genome biology*, **13**, R87.

Baylin, S. and Bestor, T.H. (2002) Altered methylation patterns in cancer cell genomes: Cause or consequence?, *Cancer Cell*, **1**, 299-305.

Benjamini, Y. and Heller, R. (2007) False Discovery Rates for Spatial Signals, *Journal of the American Statistical Association*, **102**, 1272-1281.

Benjamini, Y. and Hochberg, Y. (1997) Multiple Hypotheses Testing with Weights, *Scandinavian Journal of Statistics*, **24**, 407-418.

Benjamini, Y., Krieger, A.M. and Yekutieli, D. (2006) Adaptive linear step-up procedures that control the false discovery rate, *Biometrika*, **93**, 491-507.

Bock, C. (2012) Analysing and interpreting DNA methylation data, *Nature Reviews Genetics*, **13**, 705-719.

Campagna, D.*, et al.* (2013) PASS-bis: a bisulfite aligner suitable for whole methylome analysis of Illumina and SOLiD reads, *Bioinformatics*, **29**, 268-270.

Chen, P.Y., Cokus, S.J. and Pellegrini, M. (2010) BS Seeker: precise mapping for bisulfite sequencing, *BMC Bioinformatics*, **11**, 203.

Eckhardt, F.*, et al.* (2006) DNA methylation profiling of human chromosomes 6, 20 and 22, *Nature Genetics*, **38**, 1378-1385.

Gopalakrishnan, S., Van Emburgh, B.O. and Robertson, K.D. (2008) DNA methylation in development and human disease, *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, **647**, 30-38.

Gu, H.*, et al.* (2010) Genome-scale DNA methylation mapping of clinical samples at single-nucleotide resolution, *Nature Methods*, **7**, 133-136.

Gu, H.*, et al.* (2011) Preparation of reduced representation bisulfite sequencing libraries for genome-scale DNA methylation profiling, *Nature Protocols*, **6**, 468-481.

Guzman, L., *et al.* (2012) Analysis of aberrant methylation on promoter sequences of tumor suppressor genes and total DNA in sputum samples: a promising tool for early detection of COPD and lung cancer in smokers, *Diagnostic Pathology*, **7**, 87.

Hansen, K.D., *et al.* (2011) Increased methylation variation in epigenetic domains across cancer types, *Nature Genetics*, **43**, 768-775.

Hansen, K.D, Langmead, B. and Irizarry, R. (2012) BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions, *Genome biology*, **13**, R83.

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

Hebestreit, K., Dugas, M. and Klein, H.-U. (2013) Detection of significantly differentially methylated regions in targeted bisulfite sequencing data, *Bioinformatics*, **29**, 1647-1653.

Irizarry, R.A., *et al.* (2009) The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores, *Nature Genetics*, **41**, 178-186.

Jaffe, A.E., *et al.* (2012) Bump hunting to identify differentially methylated regions in epigenetic epidemiology studies, *International Journal of Epidemiology*, **41**, 200-209.

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

Krueger, F., *et al.* (2012) DNA methylome analysis using short bisulfite sequencing data, *Nature Methods*, **9**, 145 - 151.

Laird, P.W. (2003) The power and the promise of DNA methylation markers, *Nature Reviews Cancer*, **3**, 253-266.

Law, J.A. and Jacobsen, S.E. (2010) Establishing, maintaining and modifying DNA methylation patterns in plants and animals, *Nature Reviews Genetics*, **11**, 204-220.

Lister, R., *et al.* (2009) Human DNA methylomes at base resolution show widespread epigenomic differences, *Nature*, **462**, 315-322.

Lister, R., *et al.* (2011) Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells, *Nature*, **471**, 68-73.

Meissner, A., *et al.* (2008) Genome-scale DNA methylation maps of pluripotent and differentiated cells, *Nature*, **454**, 766-770.

Pawitan, Y., *et al.* (2005) False discovery rate, sensitivity and sample size for microarray studies, *Bioinformatics*, **21**, 3017-3024.

Rohde, C., *et al.* (2010) BISMA - Fast and accurate bisulfite sequencing data analysis of individual clones from unique and repetitive sequences, *BMC Bioinformatics*, **11**, 230.

Stockwell, P.A., *et al.* (2014.) DMAP: Differential Methylation Analysis Package for RRBS and WGBS data, *Bioinformatics advanced online publication*, doi:10.1093/bioinformatics/btu1126.

Storey, J.D. (2002) A direct approach to false discovery rates, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**, 479-498.

Storey, J.D. and Tibshirani, R. (2003) Statistical significance for genomewide studies, *Proceedings of the National Academy of Sciences*, **100**, 9440-9445.

Strathdee, G. and Brown, R. (2002) Aberrant DNA methylation in cancer: potential clinical interventions, *Expert Reviews in Molecular Medicine*, **4**, 1-17.

Sun, S., Noviski, A. and Yu, X. (2013) MethyQA: a pipeline for bisulfite-treated methylation sequencing quality assessment, *BMC Bioinformatics*, **14**, 259.

Sun, S. and Yu, X. (2014) HMM-Fisher: a hidden Markov Model-based method for identifying differential methylation, *Manuscript in preparation*.

Sun, Z., *et al.* (2011) Integrated Analysis of Gene Expression, CpG Island Methylation, and Gene Copy Number in Breast Cancer Cells by Deep Sequencing, *PLoS ONE*, **6**, e17490.

Sun, Z., *et al.* (2012) SAAP-RRBS: streamlined analysis and annotation pipeline for reduced representation bisulfite sequencing, *Bioinformatics*, **28**, 2180-2181.

Suzuki, M. and Bird, A. (2008) DNA methylation landscapes: provocative insights from epigenomics, *Nature Reviews Genetics*, **9**, 465 - 476.

Wang, H.-Q., Tuominen, L. and Tsai, C.-J. (2011) SLIM: a sliding linear model for estimating the proportion of true null hypotheses in datasets with dependence structures, *Bioinformatics*, **27**, 225-231.

Wei, S., Brown, R. and Huang, T. (2003) Aberrant DNA methylation in ovarian cancer: is there an epigenetic predisposition to drug response?, *Annals of the New York Academy of Sciences*, **983**, 243-250.

Xi, Y., *et al.* (2012) RRBSMAP: A Fast, Accurate and User-friendly Alignment Tool for Reduced Representation Bisulfite Sequencing, *Bioinformatics*, **28**, 430-432.

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

Yu, X. and Sun, S. (2014) HMM-DM: identifying differentially methylated regions using a Hidden Markov model, *Manuscript submitted for publication*.

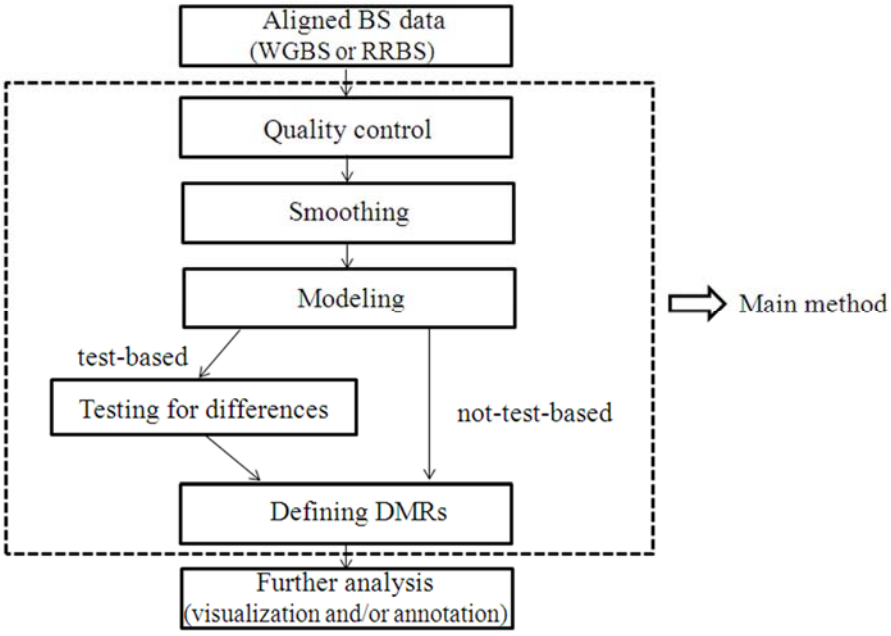



**Figure 5-1**

**Six analysis aspects of DMR identification methods.**

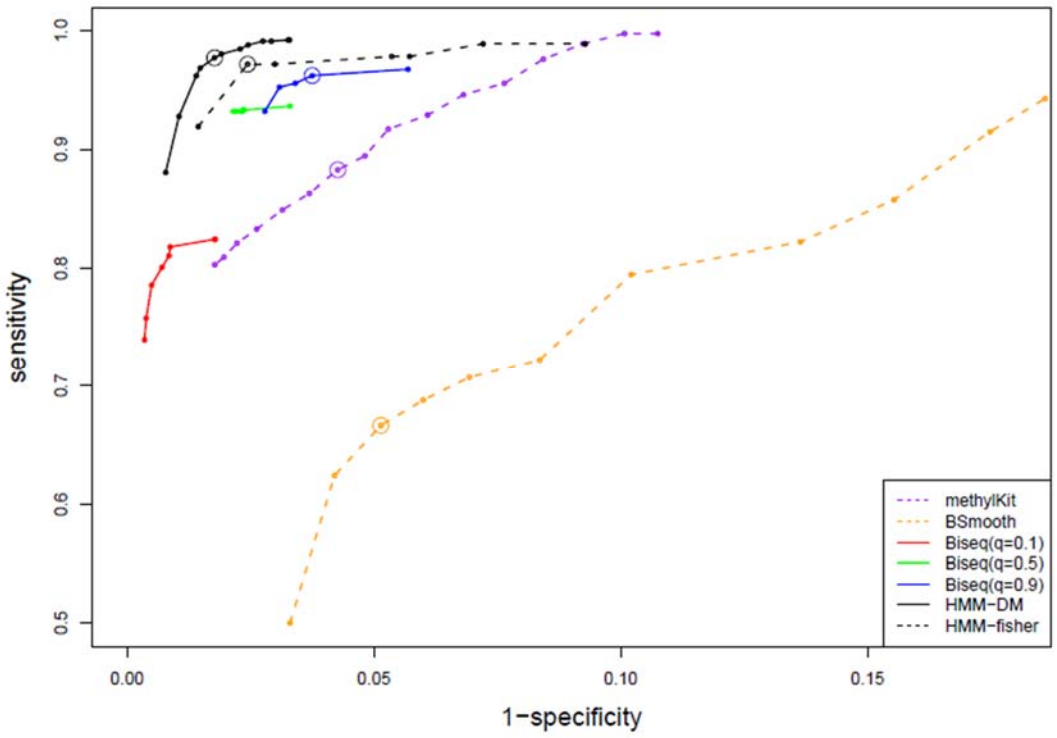
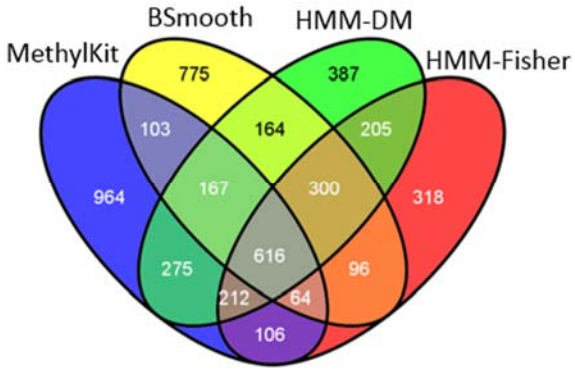


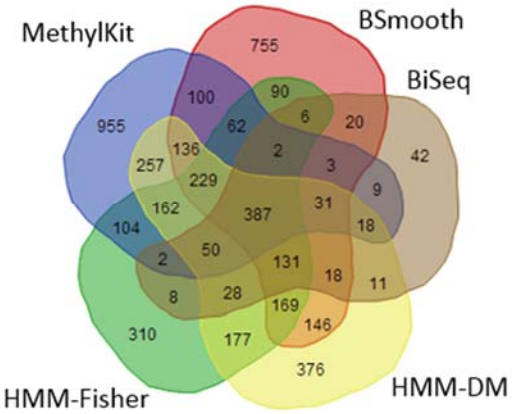

**Figure 5-2**

**ROC curves for differentially methylated CG sites identified by the five methods**. Shown are ROC curves for different q-value thresholds (methylKit; purple dashed line), different *t*-statistics (BSmooth; orange dashed line), different q (cluster-wise FPR) and q2 (CG-wise FPR) values (BiSeq; colored solid line), different posterior probability cutoffs (HMM-DM; black solid line), and different p-value thresholds (HMM-Fisher; black dashed line). Each ROC curve for BiSeq is generated from a chosen q value with different q2 values. The circle on each curve shows the "optimal" cutoff that shows relatively higher sensitivity and relatively lower false positive rate than other cutoffs. For BiSeq, the "optimal" cutoff is found with q = 0.9.

A.




B.




| Total | 4752 |
|---|---|
| **By all 4** | 616 (12.96%) |
| **By any 3** | 743 (15.63%) |
| **By any 2** | 743 (19.97%) |
| **By only 1** | 2444 (51.44%) |

| Total | 4794 |
|---|---|
| **By all 5** | 387 (8.07%) |
| **By any 4** | 443 (9.24%) |
| **By any 3** | 604 (12.60%) |
| **By any 2** | 922 (19.23%) |
| **By only 1** | 2438 (50.86%) |

**Figure 5-3**

**Comparing the DM CG sites identified by all five approaches.** (A) Comparing all methods except BiSeq. Shown is the Venn diagram of the comparison results and number (percentage) of DM CG sites identified by all four, any three, any two, and only one method. (B) Comparing all five methods. Shown is the Venn diagram of the comparison results and number (percentage) of DM CG sites identified by all five, any four, any three, any two, and only one method.

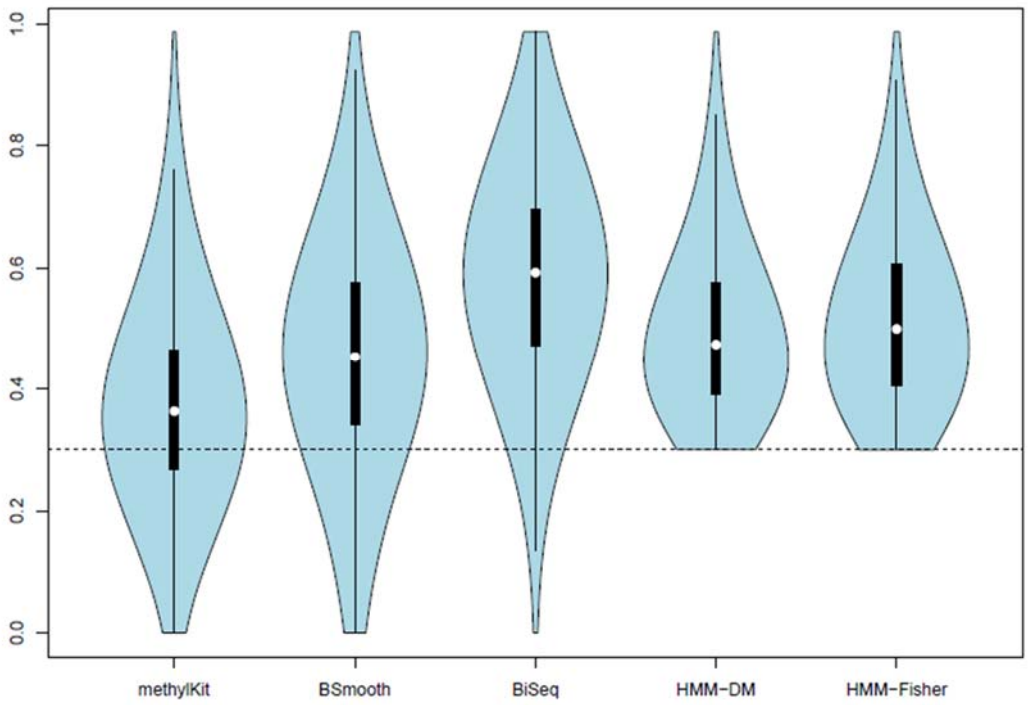



**Figure 5-4**

**Absolute values of raw mean differences for DM CG sites identified by all methods**. Each violin shows the distribution of the raw mean differences for the DM CG sites identified by each method, with width proportional to the number of CG sites. For each CG, mean difference is calculated as the mean methylation level in the ER+ group minus the mean methylation level in the ER- group. The dashed line indicates raw mean difference of 0.3. The percentage of DM CG sites with raw mean difference < 0.3 is 32.79% for methylKit, 17.77% for BSmooth, 4.18% for BiSeq, and 0% for HMM-DM and HMM-Fisher.

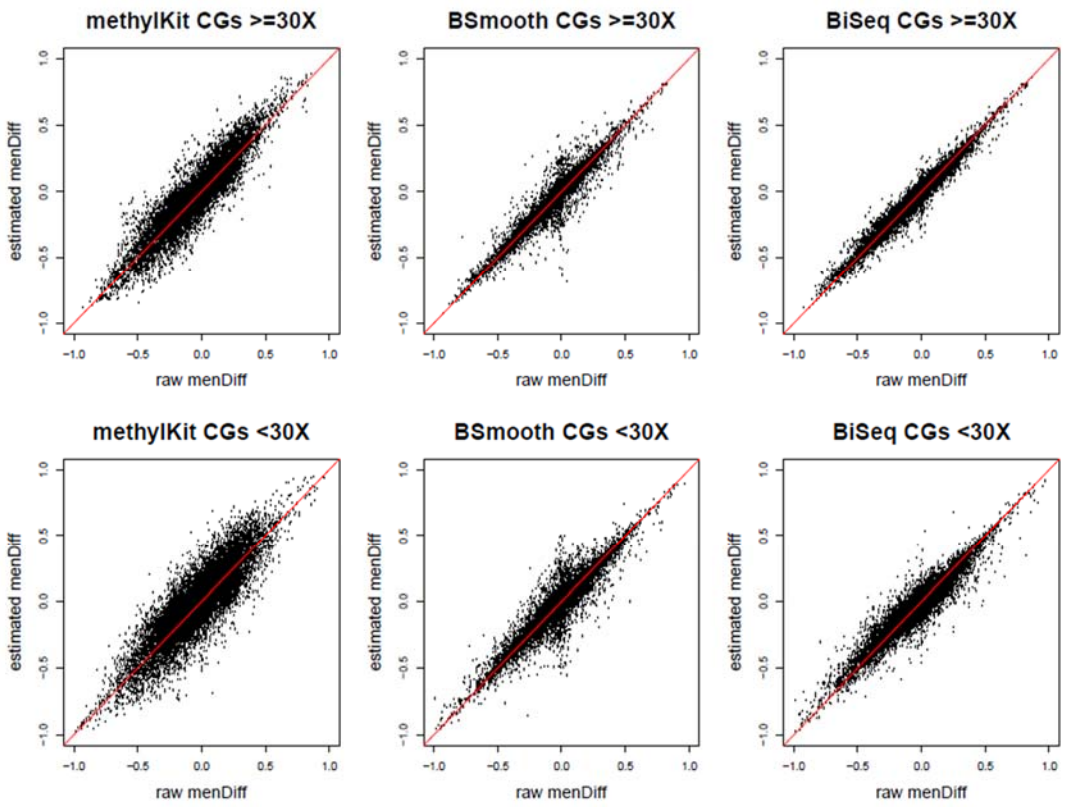



**Figure 5-5**

**Plots of estimated mean differences vs. raw mean differences for CG sites with different coverage.**

**Table 5-1. Algorithms and functions in each analysis aspect for the five methods.**

| | **MethylKit** | **BSmooth** | **BiSeq** | **HMM-Fisher** | **HMM-DM** |
|---|---|---|---|---|---|
| **Quality control and preprocessing** | Coverage normalization | Removing low coverage | Constraining on CpG cluster | Removing low coverage | Removing low coverage |
| **Smoothing** | No smoothing involved | Locally weighted logistic regression | 1) Limiting the high coverage<br>2) Weighted local likelihood | One first order hidden Markov model for each sample | First order hidden Markov model for both groups |
| **Modeling** | Modeling methylation level using logistic regression | 1) Quality control<br>2) Modeling methylation level using linear regression | Modeling methylation level using beta regression | Modeling methylation category using HMM | Modeling DM status using HMM |
| **Testing** | Sliding linear model to correct p-value | Modified *t*-test | 1) Cluster-wise testing<br>2) CG-wise testing | Fisher's exact test | No testing involved |
| **Defining DMRs** | Single-CG level | Region-level | Region-level | 1) Single-CG level<br>2) Summarize into regions | 1) Single-CG level<br>2) Summarize into regions |
| **Further analysis** | Annotation and visualization | Visualization | Annotation and visualization | Annotation and visualization | Annotation and visualization |

**Table 5-2. Key features in DMR identification methods.**

| | MethylKit | BSmooth | BiSeq | HMM-Fisher | HMM-DM |
|---|---|---|---|---|---|
| Data type | WGBS<br>Targeted BS | WGBS | Targeted BS | WGBS<br>Targeted BS | WGBS<br>Targeted BS |
| R package/code | package methylKit | Bioconductor package bsseq | Bioconductor package biseq | Pipeline HMM-Fisher | Pipeline HMM-DM |
| Limit high coverage | √ | × | √ | × | × |
| Remove low coverage | √ | √ | √ | √ | √ |
| Spatial correlation | × | √ | √ | √ | √ |
| Multiple testing correction | √ | × | √ | × | Not applicable |
| DMRs visualization | √ | √ | √ | √ | √ |
| Genomic annotation | √ | × | √ | √ | √ |

√: the method has a specific feature

×: the method does not have a specific feature

**Table 5-3. Uniform distributions that are used to simulate the test samples in DMRs.**

| | > 3 CG sites | ≤ 3 CG sites |
|---|---|---|
| H DMRs | Uniform (0, 0.4) | Uniform (0, 0.2) |
| L DMRs | Uniform (0.6, 1) | Uniform (0.8, 1) |
| M-H DMRs | Uniform (0, 0.3) | Uniform (0, 0.2) |
| M-L DMRs | Uniform (0.7, 1) | Uniform (0.8, 1) |

**Table 5-4. The default and modified settings of the five methods.**

| | Default settings | Modified settings |
|---|---|---|
| MethylKit | Normalizing read coverage | Same as the default |
| BSmooth | Smooth window ≥ 70 CG/1000 bp, distance ≤ $10^8$ bp | Smooth window ≥ 5 CG/25 bp, distance ≤ 100bp |
| BiSeq | Cluster ≥ 20 CG sites, distance ≤ 100 bp,<br>smooth window = 80 bp | Cluster ≥ 1 CG, distance ≥ 100 bp,<br>smooth window = 25 bp |
| HMM-DM | Partition = 200 CG, transition prior = dirichlet (10, 10, 10) | Same as the default |
| HMM-Fisher | Transition prior = dirichlet (1,1,1), the standard deviation of the emission distribution is 0.12, 0.15, and 0.13 for N, P, and F states respectively | Same as the default |

**Table 5-5. The cutoff statistics for default and modified settings using simulated data.**

| | Cutoff for default settings | Cutoff for modified settings |
|---|---|---|
| MethylKit | $q < 0.01$ | $q < 10^{-10}$ |
| BSmooth | $-2 \leq q \leq 2$ | $-4.6 \leq q \leq 4.6$ |
| BiSeq | q (cluster-wise FPR) =0.1<br>$q_2$ (CG-wise FPR) =0.05 | q = 0.9<br>$q_2$= 0.1 |
| HMM-DM | Posterior probability > 0.4 | Posterior probability > 0.8 |
| HMM-Fisher | $p \leq 0.05$ | $p \leq 0.03$ |

**Table 5-6. Results of the five methods from the simulated dataset.**

**A.** Default parameter settings and default cutoff of output statistics

| | **Called DM** | **True positive (sensitivity)** | **False positive (FP rate)** |
|---|---|---|---|
| MethylKit | 1841 | 927 (99.89%) | 914 (10.08%) |
| Bsmooth | 500 | 460 (49.52%) | 40 (0.44%) |
| BiSeq | 491 | 435 (46.82%) | 56 (0.63%) |
| HMM-DM | 1220 | 922 (99.25%) | 298 (3.29%) |
| HMM-Fisher | 1174 | 903 (97.20%) | 271 (2.99%) |

**B.** Modified settings and "optimal" cutoff of output statistics

| | **Called DM** | **True positive (sensitivity)** | **False positive (FP rate)** |
|---|---|---|---|
| MethylKit | 1207 | 820 (88.27%) | 387 (4.27%) |
| Bsmooth | 1085 | 619 (66.13%) | 466 (5.13%) |
| BiSeq | 1234 | 894 (96.23%) | 340 (3.75%) |
| HMM-DM | 1206 | 908 (97.74%) | 298 (1.77%) |
| HMM-Fisher | 1124 | 903 (97.20%) | 221 (2.44%) |

"Optimal" cutoff: the cutoff statistic for each method that shows relatively higher sensitivity and relatively lower false positive rate than other cutoffs. The "optimal" cutoffs are: $q \leq 10^{-7}$ for methylKit, $-4.6 \leq t \leq 4.6$ for BSmooth, $q = 0.9$ and $q2 = 0.1$ for BiSeq, posterior probability $\geq 0.8$ for HMM-DM, and $p \leq 0.03$ for HMM-Fisher.

**Table 5-7. Sensitivity of the five approaches in DMRs with different lengths and variation levels.**

| | methylKit | BSmooth | BiSeq | HMM-DM | HMM-Fisher |
|---|---|---|---|---|---|
| DMRs >20 (414 CG) | 93.22% | 72.59% | 99.72% | 99.72% | 99.03% |
| DMRs 3-20 (488 CG) | 85.95% | 63.14% | 94.71% | 97.26% | 97.34% |
| DMRs ≤ 2 (27 CG) | 70.37% | 59.26% | 81.48% | 81.48% | 66.67% |
| Small variation DMRs (649 CG) | 90.79% | 86.89% | 99.38% | 99.38% | 99.23% |
| Large variation DMRs (280 CG) | 82.50% | 10.35% | 88.93% | 93.93% | 92.50% |

Shown are comparison results of five approaches with their "optimal" cutoff values. Sensitivity is calculated for DMRs with > 20 CG sites, DMRs with 3-20 CG sites, DMRs with ≤ 2 CG sites, as well as DMRs with small and large within group variation (see the first column). The number of DM CG sites within each region type is shown in parenthesis in column 1.

**Table 5-8. The number of DM, hypermethylated, and hypomethylated CG sites identified by each method.**

| | Called DM | Hypermethylated | Hypomethylated |
|---|---|---|---|
| MethylKit | 2507 | 1722 | 785 |
| Bsmooth | 2285 | 1612 | 673 |
| BiSeq | 766 | 633 | 133 |
| HMM-DM | 2326 | 1789 | 537 |
| HMM-Fisher | 1917 | 1513 | 404 |

# CHAPTER 6: DISCUSSION AND FUTURE WORK

Next-generation sequencing technologies have revolutionized genomic and genetic research. However, there are multiple issues that add complexity to next-generation sequencing analysis. In this dissertation, I have addressed several of these issues, which can be summarized as the following four topics: 1) aligning sequencing reads with varying sequencing quality and reads from repetitive regions; 2) identifying SNPs in low sequencing coverage data; 3) developing more accurate and efficient methods for the identification of DMRs in bisulfite sequencing data; and 4) evaluating different DMR identification methods with bisulfite sequencing data. In this chapter, I briefly review the main results and achievements of each chapter and discuss directions for future work.

In Chapter 2, I have evaluated the performance of four commonly used alignment programs — SOAP2, Bowtie, BWA, and Novoalign — on data with varying quality and from repetitive regions. The results show that, for sequencing data with reads that have relatively good quality or that have had low quality bases trimmed off, all four alignment programs perform similarly. In addition, trimming off low quality ends markedly increases the number of aligned reads and improves the consistency among different aligners as well, especially for low quality data. However, Novoalign is more sensitive to the improvement of data quality. Trimming off low quality ends significantly increases the concordance between Novoalign and other aligners. As for aligning reads from repetitive regions, the simulation data show that reads from repetitive regions tend to be aligned incorrectly, and suppressing reads with multiple hits can improve alignment accuracy. Besides the above discoveries, this research work can be extended in the following repects. First, in addition

to the single-end sequencing reads, conducting a systematic comparison to pair-end sequencing data can expand the scope of this study. Second, the current simulation data study mainly focuses on the issue of repetitive reads. Simulating sequencing error patterns and repetitive reads at the same time may help to study the interplay of these two issues.

Chapter 3 is a comprehensive study that evaluated the performance of four SNP calling algorithms (SOAPsnp, Atlas-SNP2, SAMtools, and GATK) using low-coverage single-sample sequencing data. Without any post-output filtering, SOAPsnp calls more SNVs than the other programs since it has fewer internal filtering criteria. Atlas-SNP2 has stringent internal filtering criteria; thus it reports the least number of SNVs. When comparing the four algorithms using different coverage cutoff values, the results indicate that: 1) the overall agreement of the four calling algorithms is low, especially in non-dbSNPs; 2) the agreement of the four algorithms is similar when using different coverage cutoffs, except that the non-dbSNPs agreement level tends to increase slightly with increasing coverage; and 3) overall, GATK and Atlas-SNP2 have a relatively higher positive calling rate and sensitivity, but GATK calls more SNVs. Therefore, if users intend to use only one calling program, GATK may be a good choice. However, in order to increase the overall accuracy, it is better to employ more than one SNP calling algorithms and a comprehensive strategy in their validation plan. Users may first take the SNVs identified by at least two algorithms and with high coverage for validation, then move on to the low-coverage SNVs identified by multiple algorithms or SNVs called by one method but with high quality.

Chapter 4 is the introduction of a hidden Markov model-based approach HMM-DM, which is about identifying differentially methylated regions using bisulfite sequencing data.

The proposed statistical method uses a HMM to account for the sequencing errors and the spatial correlation between CG sites along the genome. The methylation levels of both groups under three differential methylation states are modeled using Beta distributions, which well account for the within group variation in DNA methylation. The performance of this HMM-DM method is evaluated based on a simulated dataset where DMRs of different length and within group variation are generated. The evaluation results show that HMM-DM performs better than BSmooth, the most commonly used and cited DMR identification method, especially in DMRs that are short and have large within group variation. This study has demonstrated several advantages of HMM-DM too. First, it is designed to be suitable for detecting DMRs using data generated from both whole-genome bisulfite sequencing and targeted bisulfite sequencing. It also can be applied to any epigenetic regions of biological interest. Second, HMM-DM is developed for methylation sequencing data with single-base-resolution. Thus, methylation changes over short distances (e.g., even a few bps) can be well captured. Third, with the Bayesian approach, parameters in the model are estimated from the data with given prior distributions, which makes this method more efficient in capturing the real methylation patterns. In addition to the above advantages, the HMM-DM method can be improved by incorporating the sequencing coverage into the hidden Markov model. This change may better correct sequencing errors, since the falsely sequenced CG sites usually have low coverage and their methylation levels may be dramatically different from nearby sites.

The final part of this dissertation (Chapter 5) includes a review and comparison of five DMR identification methods using both simulated and real bisulfite sequencing data. In the simulated dataset, major findings are: 1) Parameter settings can largely affect the

accuracy of DMR identification; 2) All five methods show higher accuracy in the identification of simulated DMRs that are long and have small within group variation; 3) HMM-DM and HMM-Fisher yield relatively higher sensitivity and lower false positive rate than others, especially in DMRs with large within group variation. For the real sequencing data analysis, the five methods show low concordances, probably due to the different approaches they are using when tackling the issues in DMR identification. In addition, among the three methods (methylKit, BSmooth, and BiSeq) that involve methylation estimation, BiSeq can best present the raw methylation signals. Therefore, users may select DMR identification methods based on the characteristics of their data and the advantages of each method. To guarantee a higher accuracy in validation and further analysis, users may choose the identified DMRs that are longer and have smaller within group variation as a priority. Currently, the datasets used in this study are generated from targeted bisulfite sequencing and the parameter settings in all methods are modified to adapt to the data. Since some methods are developed for both targeted and whole genome sequencing data, it is still worthwhile to examine the performance of all methods in a whole genome sequencing dataset.

## Bibliography


Adams, M.D., *et al.* (2012) Global mutational profiling of formalin-fixed human colon cancers from a pathology archive, *Modern Pathology*, **25**, 1599-1608.

Akalin, A., *et al.* (2012) methylKit: a comprehensive R package for the analysis of genome-wide DNA methylation profiles, *Genome biology*, **13**, R87.

Alkan, C., *et al.* (2009) Personalized copy number and segmental duplication maps using next-generation sequencing, *Nature Genetics*, **41**, 1061-1067.

Altmann, A., *et al.* (2011) vipR: variant identification in pooled DNA using R, *Bioinformatics*, **27**, i77-i84.

Altschul, S.F., *et al.* (1990) Basic local alignment search tool, *Journal of Molecular Biology*, **215**, 403-410.

Altshuler, D., *et al.* (2000) The common PPARr Pro12Ala polymorphism is associated with decreased risk of type 2 diabetes, *Nature Genetics*, **26**, 76-80.

Arinami, T., *et al.* (2005) Genomewide High-Density SNP Linkage Analysis of 236 Japanese Families Supports the Existence of Schizophrenia Susceptibility Loci on Chromosomes 1p, 14q, and 20p, *American journal of human genetics*, **77**, 937-944.

Bansal, V. (2010) A statistical method for the detection of variants from next-generation resequencing of DNA pools, *Bioinformatics*, **26**, i318-i324.

Baylin, S. and Bestor, T.H. (2002) Altered methylation patterns in cancer cell genomes: Cause or consequence?, *Cancer Cell*, **1**, 299-305.

Benjamini, Y. and Heller, R. (2007) False Discovery Rates for Spatial Signals, *Journal of the American Statistical Association*, **102**, 1272-1281.

Benjamini, Y. and Hochberg, Y. (1997) Multiple Hypotheses Testing with Weights, *Scandinavian Journal of Statistics*, **24**, 407-418.

Benjamini, Y., Krieger, A.M. and Yekutieli, D. (2006) Adaptive linear step-up procedures that control the false discovery rate, *Biometrika*, **93**, 491-507.

Bock, C. (2012) Analysing and interpreting DNA methylation data, *Nature Reviews Genetics*, **13**, 705-719.

Bock, C.*, et al.* (2012) DNA Methylation Dynamics during In Vivo Differentiation of Blood and Skin Stem Cells, *Molecular Cell*, **47**, 633-647.

Bond, G.L. and Levine, A.J. (2006) A single nucleotide polymorphism in the p53 pathway interacts with gender, environmental stresses and tumor genetics to influence cancer in humans, *Oncogene*, **26**, 1317-1323.

Bonetta, L. (2006) Genome sequencing in the fast lane, *Nature Methods*, **3**, 141-147.

Brunner, A.L.*, et al.* (2009) Distinct DNA methylation patterns characterize differentiated human embryonic stem cells and developing human fetal liver, *Genome Research*, **19**, 1044-1056.

Burrows, M. and Wheeler, D.J. (1994) A block-sorting lossless data compression algorithm. *Technical Report 124.* Digital Equipment Corporation, Palo Alto, CA.

Campagna, D.*, et al.* (2013) PASS-bis: a bisulfite aligner suitable for whole methylome analysis of Illumina and SOLiD reads, *Bioinformatics*, **29**, 268-270.

Chen, P., Cokus, S. and Pellegrini, M. (2010) BS Seeker: precise mapping for bisulfite sequencing, *BMC Bioinformatics*, **11**, 203.

Chen, Y., Souaiaia, T. and Chen, T. (2009) PerM: efficient mapping of short sequencing reads with periodic full sensitive spaced seeds, *Bioinformatics*, **25**, 2514-2521.

Cibulskis, K.*, et al.* (2013) Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples, *Nature Biotechnology*, **31**, 213-219.

Collins, F.S., Brooks, L.D. and Chakravarti, A. (1998) A DNA Polymorphism Discovery Resource for Research on Human Genetic Variation, *Genome Research*, **8**, 1229-1231.

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

Corneveaux, J.J.*, et al.* (2010) Association of CR1, CLU and PICALM with Alzheimer's disease in a cohort of clinically characterized and neuropathologically verified individuals, *Human Molecular Genetics*, **19**, 3295–3301.

Dalca, A.V. and Brudno, M. (2010) Genome variation discovery with high-throughput sequencing data, *Briefings in Bioinformatics*, **11**, 3-14.

De Bona, F.*, et al.* (2008) Optimal spliced alignments of short sequence reads, *BMC Bioinformatics*, **24**, i174-i180.

Deng, J.*, et al.* (2009) Targeted bisulfite sequencing reveals changes in DNA methylation associated with nuclear reprogramming, *Nature Biotechnology*, **27**, 353-360.

DePristo, M.A.*, et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data, *Nature Genetics*, **43**, 491-498.

Eckhardt, F.*, et al.* (2006) DNA methylation profiling of human chromosomes 6, 20 and 22, *Nature Genetics*, **38**, 1378-1385.

Edmonson, M.N.*, et al.* (2011) Bambino: a variant detector and alignment viewer for next-generation sequencing data in the SAM/BAM format, *Bioinformatics*, **27**, 865-866.

Ferragina, P. and Manzini, G. (2000) Opportunistic data structures with applications. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, pp. 390-398.

Flicek, P. and Birney, E. (2009) Sense from sequence reads: methods for alignment and assembly, *Nature Methods*, **6**, S6-S12.

Hach, F.*, et al.* (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping, *Nature Methods*, **7**, 576-577.

Hansen, K., Langmead, B. and Irizarry, R. (2012) BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions, *Genome biology*, **13**, R83.

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

Hebestreit, K., Dugas, M. and Klein, H.-U. (2013) Detection of significantly differentially methylated regions in targeted bisulfite sequencing data, *Bioinformatics*, **29**, 1647-1653.

Henningsson, A.*, et al.* (2005) Association of CYP2C8, CYP3A4, CYP3A5, and ABCB1 Polymorphisms with the Pharmacokinetics of Paclitaxel, *Clinical Cancer Research*, **11**, 8097-8104.

Higashi, M.K., *et al.* (2002) Association Between CYP2C9 Genetic Variants and Anticoagulation-Related Outcomes During Warfarin Therapy, *JAMA: The Journal of the American Medical Association*, **287**, 1690-1698.

Garrison, E. and Marth, G. (2012) Haplotype-based variant detection from short-read sequencing, *arXiv preprint arXiv:1207.3907v2 [q-bio.GN]*.

Gelfand, A. and Smith, A. (1990) Sampling-Based Approaches to Calculating Marginal Densities, *Journal of the American Statistical Association*, **85**, 398-409.

Gopalakrishnan, S., Van Emburgh, B.O. and Robertson, K.D. (2008) DNA methylation in development and human disease, *Mutation Research/Fundamental and Molecular Mechanisms of Mutagenesis*, **647**, 30-38.

Goya, R., *et al.* (2010) SNVMix: predicting single nucleotide variants from next-generation sequencing of tumors, *Bioinformatics*, **26**, 730-736.

Gu, H., *et al.* (2010) Genome-scale DNA methylation mapping of clinical samples at single-nucleotide resolution, *Nature Methods*, **7**, 133-136.

Gu, H., *et al.* (2011) Preparation of reduced representation bisulfite sequencing libraries

Guzman, L., *et al.* (2012) Analysis of aberrant methylation on promoter sequences of tumor suppressor genes and total DNA in sputum samples: a promising tool for early detection of COPD and lung cancer in smokers, *Diagnostic Pathology*, **7**, 87.

Hansen, K.D., *et al.* (2011) Increased methylation variation in epigenetic domains across cancer types, *Nature Genetics*, **43**, 768-775.

Hansen, K.D, Langmead, B. and Irizarry, R. (2012) BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions, *Genome biology*, **13**, R83.

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

Hebestreit, K., Dugas, M. and Klein, H.-U. (2013) Detection of significantly differentially methylated regions in targeted bisulfite sequencing data, *Bioinformatics*, **29**, 1647-1653.

Irizarry, R.A., *et al.* (2009) The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific CpG island shores, *Nature Genetics*, **41**, 178-186.

Jaffe, A.E., *et al.* (2012) Bump hunting to identify differentially methylated regions in epigenetic epidemiology studies, *International Journal of Epidemiology*, **41**, 200-209.

Jiang, H. and Wong, W.H. (2008) SeqMap: mapping massive amount of oligonucleotides to the genome, *Bioinformatics*, **24**, 2395-2396.

Jimenez-Sanchez, G., Childs, B. and Valle, D. (2001) Human disease genes, *Nature*, **409**, 853-855.

Johnson, D.S., *et al.* (2007) Genome-Wide Mapping of in Vivo Protein-DNA Interactions, *Science*, **316**, 1497-1502.

Kammerer, S., *et al.* (2005) Association of the NuMA region on chromosome 11q13 with breast cancer susceptibility, *Proceedings of the National Academy of Sciences of the United States of America*, **102**, 2004-2009.

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

Krueger, F., *et al.* (2012) DNA methylome analysis using short bisulfite sequencing data, *Nature Methods*, **9**, 145 - 151.

Koboldt, D.C., *et al.* (2009) VarScan: variant detection in massively parallel sequencing of individual and pooled samples, *Bioinformatics*, **25**, 2283-2285.

Kuwano, R., *et al.* (2006) Dynamin-binding protein gene on chromosome 10q is associated with late-onset Alzheimer's disease, *Human Molecular Genetics*, **15**, 2170-2182.

Laird, P.W. (2003) The power and the promise of DNA methylation markers, *Nature Reviews Cancer*, **3**, 253-266.

Lam, T.W., *et al.* (2008) Compressed indexing and local alignment of DNA, *Bioinformatics*, **24**, 791-797.

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

Laurent, L., *et al.* (2010) Dynamic changes in the human methylome during differentiation, *Genome Research*, **20**, 320-331.

Law, J.A. and Jacobsen, S.E. (2010) Establishing, maintaining and modifying DNA methylation patterns in plants and animals, *Nature Reviews Genetics*, **11**, 204-220.

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

Li, H. and Durbin, R. (2010) Fast and accurate long-read alignment with Burrows–Wheeler transform, *Bioinformatics*, **26**, 589-595.

Li, H. and Homer, N. (2010) A survey of sequence alignment algorithms for next-generation sequencing, *Briefings in Bioinformatics*, **11**, 473-483.

Li, H., Ruan, J. and Durbin, R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores, *Genome Research*, **18**, 1851-1858.

Li, R., *et al.* (2008) SOAP: short oligonucleotide alignment program, *Bioinformatics*, **24**, 713-714.

Li, R., *et al.* (2009) SNP detection for massively parallel whole-genome resequencing, *Genome Research*, **19**, 1124-1132.

Li, R., *et al.* (2009) SOAP2: an improved ultrafast tool for short read alignment, *Bioinformatics*, **25**, 1966-1967.

Li, Y., *et al.* (2010) The DNA methylome of human peripheral blood mononuclear cells, *PLoS biology*, **8**, e1000533.

Li, Y., *et al.* (2011) Low-coverage sequencing: Implications for design of complex trait association studies, *Genome Research*, **21**, 940-951.

Li, Y., *et al.* (2012) Single Nucleotide Polymorphism (SNP) Detection and Genotype Calling from Massively Parallel Sequencing (MPS) Data, *Statistics in Bioscience*, **5**, 3-25.

Lin, H., *et al.* (2008) ZOOM! Zillions of oligos mapped, *Bioinformatics*, **24**, 2431-2437.

Lister, R., *et al.* (2009) Human DNA methylomes at base resolution show widespread epigenomic differences, *Nature*, **462**, 315-322.

Lister, R., *et al.* (2011) Hotspots of aberrant epigenomic reprogramming in human induced pluripotent stem cells, *Nature*, **471**, 68-73.

Ma, B., Tromp, J. and Li, M. (2002) PatternHunter: faster and more sensitive homology search, *Bioinformatics*, **18**, 440-445.

Mardis, E.R. (2008) The impact of next-generation sequencing technology on genetics, *Trends in Genetics*, **24**, 133-141.

Martin, E.R., *et al.* (2010) SeqEM: an adaptive genotype-calling approach for next-generation sequencing studies, *Bioinformatics*, **26**, 2803-2810.

Meissner, A., *et al.* (2008) Genome-scale DNA methylation maps of pluripotent and differentiated cells, *Nature*, **454**, 766-770.

Metzker, M.L. (2010) Sequencing technologies -- the next generation, *Anglais*, **11**, 31-46.

McKenna, A., Hanna, M. and Banks, E. (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data, *Genome Research*, **20**, 1297-1303.

Mortazavi, A., *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq, *Nature Methods*, **5**, 621-628.

Neal, R.M. (2003) Slice sampling, *The Annals of Statistics*, **31**, 705-767.

Needleman, S.B. and Wunsch, C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins, *Journal of Molecular Biology*, **48**, 443-453.

Ning, Z., Cox, A.J. and Mullikin, J.C. (2001) SSAHA: A Fast Search Method for Large DNA Databases, *Genome Research*, **11**, 1725-1729.

O'Rawe, J., *et al.* (2013) Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing, *Genome Medicine*, **5**, 28.

Pabinger, S., *et al.* (2013) A survey of tools for variant analysis of next-generation genome sequencing data, *Briefings in Bioinformatics advanced online publication, doi: 10.1093/bib/bbs086*.

Palmer, N.D., *et al.* (2011) Resequencing and Analysis of Variation in the TCF7L2 Gene in African Americans Suggests That SNP rs7903146 Is the Causal Diabetes Susceptibility Variant, *Diabetes*, **60**, 662-668.

Park, P.J. (2009) ChIP-seq: advantages and challenges of a maturing technology, *Nature Reviews Genetics*, **10**, 669-680.

Pawitan, Y., *et al.* (2005) False discovery rate, sensitivity and sample size for microarray studies, *Bioinformatics*, **21**, 3017-3024.

Quinlan, A.R., *et al.* (2008) Pyrobayes: an improved base caller for SNP discovery in pyrosequences, *Nature Methods*, **5**, 179-181.

Rivas, M.A., *et al.* (2011) Deep resequencing of GWAS loci identifies independent rare variants associated with inflammatory bowel disease, *Nature Genetics*, **43**, 1066-1073.

Rohde, C., *et al.* (2010) BISMA - Fast and accurate bisulfite sequencing data analysis of individual clones from unique and repetitive sequences, *BMC Bioinformatics*, **11**, 230.

Ruffalo, M., LaFramboise, T. and Koyutürk, M. (2011) Comparative analysis of algorithms for next-generation sequencing read alignment, *Bioinformatics*, **27**, 2790-2796.

Rumble, S.M., *et al.* (2009) SHRiMP: Accurate Mapping of Short Color-space Reads, *PLoS Computational Biology*, **5**, e1000386.

Sanger, F., Nicklen, S. and Coulson, A.R. (1977) DNA sequencing with chain-terminating inhibitors, *Proceedings of the National Academy of Sciences*, **74**, 5463-5467.

Schatz, M.C. (2009) CloudBurst: highly sensitive read mapping with MapReduce, *Bioinformatics*, **25**, 1363-1369.

Shendure, J., *et al.* (2004) Advanced sequencing technologies: methods and goals, *Nature Reviews Genetics*, **5**, 335-344.

Shen, Y., *et al.* (2010) A SNP discovery method to assess variant allele probability from next-generation resequencing data, *Genome Research*, **20**, 273-280.

Stockwell, P.A., *et al.* (2014.) DMAP: Differential Methylation Analysis Package for RRBS and WGBS data, *Bioinformatics advanced online publication*, doi:10.1093/bioinformatics/btu1126.

Storey, J.D. (2002) A direct approach to false discovery rates, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**, 479-498.

Storey, J.D. and Tibshirani, R. (2003) Statistical significance for genomewide studies, *Proceedings of the National Academy of Sciences*, **100**, 9440-9445.

Strathdee, G. and Brown, R. (2002) Aberrant DNA methylation in cancer: potential clinical interventions, *Expert Reviews in Molecular Medicine*, **4**, 1-17.

Sun, S., Noviski, A. and Yu, X. (2013) MethyQA: a pipeline for bisulfite-treated methylation sequencing quality assessment, *BMC Bioinformatics*, **14**, 259.

Sun, S. and Yu, X. (2014) HMM-Fisher: a hidden Markov Model-based method for identifying differential methylation, *Manuscript in preparation*.

Sun, Z.*, et al.* (2011) Integrated Analysis of Gene Expression, CpG Island Methylation, and Gene Copy Number in Breast Cancer Cells by Deep Sequencing, *PLoS ONE*, **6**, e17490.

Sun, Z.*, et al.* (2012) SAAP-RRBS: streamlined analysis and annotation pipeline for reduced representation bisulfite sequencing, *Bioinformatics*, **28**, 2180-2181.

Suzuki, M. and Bird, A. (2008) DNA methylation landscapes: provocative insights from epigenomics, *Nature Reviews Genetics*, **9**, 465 - 476.

The Genomes Project, C. (2012) An integrated map of genetic variation from 1,092 human genomes, *Nature*, **491**, 56-65.

Ueda, H.*, et al.* (2003) Association of the T-cell regulatory gene CTLA4 with susceptibility to autoimmune disease, *Nature*, **423**, 506-511.

Vallania, F.L.M.*, et al.* (2010) High-throughput discovery of rare insertions and deletions in large cohorts, *Genome Research*, **20**, 1711-1718.

Vyshkina, T. and Kalman, B. (2005) Haplotypes within genes of β-chemokines in 17q11 are associated with multiple sclerosis: a second phase study, *Human Genetics*, **118**, 67-75.

Wang, H.-Q., Tuominen, L. and Tsai, C.-J. (2011) SLIM: a sliding linear model for estimating the proportion of true null hypotheses in datasets with dependence structures, *Bioinformatics*, **27**, 225-231.

Weese, D., *et al.* (2009) RazerS—fast read mapping with sensitivity control, *Genome Research*, **19**, 1646-1654.

Wei, S., Brown, R. and Huang, T. (2003) Aberrant DNA methylation in ovarian cancer: is there an epigenetic predisposition to drug response?, *Annals of the New York Academy of Sciences*, **983**, 243-250.

Wei, Z., *et al.* (2011) SNVer: a statistical tool for variant calling in analysis of pooled or individual next-generation sequencing data, *Nucleic Acids Research*, **39**, e132.

Wolford, J.K., *et al.* (2006) Variants in the gene encoding aldose reductase (AKR1B1) and diabetic nephropathy in American Indians, *Diabetic Medicine*, **23**, 367-376.

Xi, Y., *et al.* (2012) RRBSMAP: A Fast, Accurate and User-friendly Alignment Tool for Reduced Representation Bisulfite Sequencing, *Bioinformatics*, **28**, 430-432.

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

Yu, X., *et al.* (2012) How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?, *BioData Mining*, **5**, 6.

Yu, X. and Sun, S. (2014) HMM-DM: identifying differentially methylated regions using a Hidden Markov model, *Manuscript submitted for publication*.

Zeggini, E., *et al.* (2005) Largescale studies of the association between variation at the TNF/LTA locus and susceptibility to type 2 diabetes, *Diabetologia*, **48**, 2013-2017.