Two algorithmic problems in analyzing genetic and epigenetic variations

Sun, Ruimin

2015

Sun, R. (2015). Two algorithmic problems in analyzing genetic and epigenetic variations.
Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/65312>

<https://doi.org/10.32657/10356/65312>

**TWO ALGORITHMIC PROBLEMS IN ANALYZING
GENETIC AND EPIGENETIC VARIATIONS**

SUN RUIMIN

**SCHOOL OF PHYSICAL AND MATHEMATICAL
SCIENCES**

2015

**TWO ALGORITHMIC PROBLEMS IN ANALYZING
GENETIC AND EPIGENETIC VARIATIONS**

SUN RUIMIN

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy

2015

Acknowledgements

After about four and a half years, I am finally finishing the journey of my PhD studies. At the end of this long and unforgettable trip, I would like to express my sincere gratitude to the people giving me a lot of help at the beginning of my thesis.

First of all, I want to show my deepest respect and genuine thanks to my supervisor, Prof. CHEN Xin, for his advice, guidance, help and encouragement to my PhD studies. During these years, he teaches me lots of his research experiences and always inspires me to find ideas to solve research problems in bioinformatics. His diligent and rigorous attitudes to scientific research also motivate me. I also want to say thanks to Prof. ZHANG Lifeng, from School of Biological Sciences (SBS), for the financial support of my last year's PhD studies.

Secondly, I would like to express my thanks to all the co-authors of my research papers: Prof. TANG Kai (SBS), Prof. MU Yuguang (SBS), GAO Xiang, and HAN Nanyu, for their valuable biological experiments and precious suggestions in the work of SNP detection; Prof. ZHANG Luoxin (NUS) and WU Qiong, for their contributions to theoretical support of the SNP discovery from mass spectrometry data. I also wish to thank TIAN Ye for his help in statistical analysis of methylation data.

Thirdly, I wish to show my gratitude to all the examiners for examining my thesis and providing precious suggestions.

Last but not the least, I want to show my appreciation to my families and friends for their support and blessings. In particular, I would like to say thanks to my dear boyfriend, TIAN Ye, who goes with me during this long journey, for his concern, help and encouragement.

Contents

Summary	4
1 Introduction	5
1.1 Introduction	5
1.2 Basics of Genetics	7
1.2.1 DNA, genes, and chromosomes	7
1.2.2 RNA and gene expression	9
1.2.3 Single nucleotide polymorphism	11
1.2.4 Epigenetics	13
1.2.5 Cytosine methylation and hydroxymethylation	14
1.2.6 Brief introduction of next-generation sequencing techniques . .	17
2 SNP Detection Using Mass Spectrometry Data	23
2.1 Introduction	23
2.1.1 Sequencing with base-specific cleavage and MS	24
2.1.2 Detecting SNPs from mass spectra	25
2.1.3 Existing methods reviews	28
2.1.4 Our contribution	29

2.2	Preliminaries	30
2.2.1	<i>In-silico</i> predicted mass spectrum	30
2.2.2	Experimentally measured mass spectrum	32
2.2.3	Explanation of measured mass peaks	34
2.3	Algorithm in SNPMS	35
2.3.1	Discussion of algorithm	35
2.3.2	Detecting SNPs in close vicinity	39
2.4	Results	42
2.4.1	Results of simulated data	42
2.4.2	Results of biological data	46
2.5	Discussion and Improvement	50
3	DNA Methylation Analysis	55
3.1	Introduction	55
3.1.1	Computational challenges of aligning BS-Seq data	56
3.1.2	BS-Seq alignment methods reviews	59
3.2	Backgrounds	64
3.2.1	Suffix array	64
3.2.2	Burrows-Wheeler transform	66
3.2.3	FM index	68
3.2.4	Bi-directional BWT and FMD index	71
3.2.5	Seeds for alignment	76
3.3	Our Method: TAMEBS	82
3.3.1	Finding approximate seeds with bi-directional index	83
3.3.2	Extending seed hits	86

3.3.3	Methylation calling	90
3.4	Results	91
3.4.1	Simulation experiments	92
3.4.2	Biological experiments	99
3.4.3	Discussion of scoring matrixes	103
3.5	Conclusion	109
4	Conclusion	112
4.1	Summary	112
4.2	Technical Contributions	113
4.3	Future Work	114
	Reference	117
	My Publications	128

Summary

Single nucleotide polymorphism (SNP) is the most common type of genetic variations. Accurate detection of SNPs is crucial to many downstream studies. To detect SNPs, MALDI-TOF mass spectrometry combined with base-specific cleavage reactions has been employed in many experiments. A new SNP detecting algorithm is presented in the thesis, together with the performance evaluation of its implemented program called SnpMs. Results demonstrate that SnpMs has a high ability to detect SNP mutations accurately.

Cytosine methylation plays an important role in many biological regulation processes. The current golden standard method for analyzing cytosine methylation is BS-Seq. In this thesis, a new tool called TAmEBS is introduced to align BS-Seq reads and estimate the methylation status of each cytosine. Experimental results on both simulated and real data showed that TAmEBS could detect many more uniquely best mapped reads while achieving a good balance between sensitivity and precision.

Chapter 1

Introduction

1.1 Introduction

The story of life was explored at a visible level before 1665 when Robert Hooke discovered the fundamental component of organisms, cell. After debating for more than a century, the Cell Theory was finally formulated by Matthias Schleiden and Theodor Schwann in 1830s and completed by Robert Remak and Rudolf Virchow in 1850s. The Cell Theory consists of three tenets: firstly, all living organisms are composed of one or more cells; secondly, the cell is the most basic unit of life; and lastly, all cells arise from pre-existing, living cells through cell division.

The studies of cells were advanced by the discovery of genes and chromosomes in cell nuclei. From the late 19th century, a large number of experiments were carried out to figure out the truth that a living organism passes its traits to its offspring. Three types of molecules were discovered successively, proteins, DNA, and RNA. They were also proved to be main factors for regulating cell functions and transmitting information to new-born cells. In brief, DNA stores all the heritable information of an living organism,

and RNA transfers a part of information to different places in a cell where these small parts of information are used as templates to synthesize proteins. Proteins perform a variety of functions within living organisms, including catalyzing metabolic reactions, replicating DNA, responding to stimuli, and transporting molecules from one location to another. The studies of genes, heredity and variation in living organisms form the field of genetics.

The development of genetic research promotes the growth of the relevant analytical technologies. One of the most fundamental technologies to study genetics is DNA sequencing, a process of determining the sequence of nucleotides in DNA fragments. Over the recent decade, the cost of sequencing was lowered dramatically from $\sim \$0.75$ per base to $\sim \$0.1$ per million bases, while the amount of sequence data production was increased to millions of reads per run [41, 43, 49]. To utilize such huge amount of data to search for and analyze genetic patterns in the full genomes of living organisms, effective and efficient computational tools are therefore highly required.

In this thesis, I will start from the brief introduction of some basic concepts of genetics and several sequencing technologies. Then I will present two widely-discussed and well-studied genetic/epigenetic variations, that is, single-nucleotide polymorphisms (SNPs) and DNA methylation, as well as the fundamentally computational problems with respect to these two genetic variations. One problem is to accurately detect SNPs by using mass spectrometry data, while the other is to analyze DNA methylation states by aligning bisulfite sequencing data. Due to the special properties of these two types of data, general-purpose methods cannot be applied directly, and hence specific approaches have to be created.

In Chapter 2 and Chapter 3, I will discuss in details the above two computational problems, respectively. The corresponding biological backgrounds will be firstly in-

troduced, followed by the properties of the data produced by the respective biological techniques. Then, I will discuss the methods that we developed, together with the comparative experiments on both simulated and real biological datasets. Experimental results with respect to either problem demonstrate the high capability of the corresponding approach that we developed. The materials of these two chapters are based on our previously published papers [64] and [65].

At last, Chapter 4 concludes this thesis and discusses the brief backgrounds of the future research topics.

1.2 Basics of Genetics

1.2.1 DNA, genes, and chromosomes

The gate of science of genetics was opened by Gregor Johann Mendel, a scientist and Augustinian monk in 1860s. Mendel studied the heritable traits of garden peas and suggested the existence of a factor, termed as a gene later, that conveys traits from parents to offspring. In 1910s, Thomas Hunt Morgan demonstrated that genes are carried on chromosomes according to the observation of the birth of a white-eyed male mutant in his fly room. Encouraged by this observation, Morgan and his students proceeded to map genes to certain locations on chromosomes. In 1913, his student Alfred Sturtevant constructed the first genetic map of a chromosome showing the linear alignment of genes on the chromosome. However, it was still unknown which part of a chromosome contains the genes. Proteins were suspected to be the containers of genes, because proteins are the other main component of chromosomes besides DNA (or deoxyribonucleic acids). The exact location of genes was not confirmed until 1944, when

Oswald Theodore Avery, Colin McLeod and Maclyn MacCarty proved that DNA is the molecule coding for genes.

The structure of a DNA molecule was determined by James D. Watson and Francis Crick in 1953. A DNA molecule consists of two strands, spiraling as a double helix. Both strands of DNA are directional, running from 5' end to 3' end. The two strands of a DNA molecule run in opposite directions, which is termed as anti-parallel. Each strand is composed of a chain of four types of nucleotides, differentiated from each other by chemical bases – adenine (A), cytosine (C), guanine (G), and thymine(T). In other words, each DNA strand can be regarded as a sequence or chain written by these four nucleotide bases.

Each nucleotide in one DNA strand pairs with its specific partner nucleotide in the opposite strand with hydrogen bonds. It is summarized by the base pairing rules: A pairs with T, and C pairs with G. Accordingly, the nucleotide string of one strand can completely define the nucleotide string of the other, which implies the key of DNA replication. Briefly speaking, DNA replication duplicates itself by splitting its two strands and using each strand as a template for the synthesis of the new complementary strand (see Figure 1.1).

Genes are some segments of DNA and arranged linearly along DNA base pair sequences. A gene is actually the unit of inheritable information which can determine certain biological functions. Within cells, DNA is organized into a structure called chromosome. Typically, eukaryotic cells (cells with nuclei, such as animals, plants, and fungi) have linear chromosomes while prokaryotic cells (cells without defined nuclei, such as bacteria) have circular chromosomes. When a cell divides, DNA replication happens so that each daughter cell contains a complete set of chromosomes. In general, the full set of chromosomes in an organism is called the genome.

Figure 1.1: The scheme of DNA replication. The synthesis of children strands starts before the parent strands are completely split. Moreover, the process of DNA replication complies with the base pairing rules. This figure comes from en.wikipedia.org/wiki/DNA_replication.

1.2.2 RNA and gene expression

Genes can determine biological functions but they are not the final executors. A vast variety of functions within living organisms are performed by proteins, large and complex-structured molecules. A protein molecule usually consists of one or more long chains of amino acid residues, which fold into its active three-dimensional structures to carry out cellular functions. The amino acid sequence of a protein molecule is determined by DNA sequences of some genes, but not produced directly from these genes. It can be directly observed in eukaryotic cells, where DNA always resides within the nucleus whereas proteins are located in cytoplasm. In fact, an RNA (ribonucleic acid) exists to collect the genetic information from DNA inside the nucleus and convey the information to ribosome in cytoplasm. The ribosome binds to the RNA chain and uses it as a template to link amino acids together.

DNA, RNA and proteins constitute the three major macromolecules that are essential for all living organisms. Like DNA, RNA has a chain structure comprised by nucleotides. However, different from DNA, RNA is a single-stranded molecule and uses nucleotides A, C, G and U (uracil) to carry genetic information. Many viruses use RNA genomes directly to encode proteins. For cellular organisms, RNA is also essential to inheritance because it transmits genetic information from DNA to synthesize functional proteins.

Gene expression is the whole process in which genetic information on genes is used to produce biologically functional molecules. It starts from *transcription* that produces messenger RNA (mRNA) from DNA. Briefly speaking, one of the DNA strands of a gene is used as a template and an mRNA is synthesized from the 3' end of the template strand to the 5' end. The production of mRNA depends on the specific base pairing rules that the nucleotide A pairs with U. In prokaryotic cells, mRNA created from transcription is ready to produce proteins. However in eukaryotic cells, the product of transcription is only an initial transcript of RNA, known as precursor mRNA (or pre-mRNA). A series of modifications are required by a pre-mRNA to become a mature mRNA. The RNA splicing is a modification unique to eukaryotes, which selects the separated coding sequences (exons) on pre-mRNA and splices them together to form a mature mRNA. The mature mRNA can be exported to ribosomes in the cytoplasm from the nucleus.

mRNA is an intermediate agent that carries information for the synthesis of one or more proteins. Once mRNA arrives at ribosome, it acts as a template for synthesizing proteins according to genetic code (see Table 1.1). The code maps 64 nucleotide triplets, called codons, to 20 amino acids. Each codon corresponds to a binding site complementary to an anticodon triplet in transfer RNA (tRNA). tRNAs with the same

anticodon sequence carry the same type of amino acid. The ribosome then links amino acids together in order specified by codons in the coding region of mRNA. This process is called *translation*. During and after translation, the linear chain of amino acids folds into its characteristic and functional three-dimensional structure to carry out the related cellular functions.

The whole process of gene expression can be summarized by the **central dogma**, which states that DNA makes RNA and RNA makes protein. Figure 1.2 describes the basic process of gene expression.

1st	2nd								3rd		
	U		C		A		G				
U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U		
	UUC		UCC		UAC		UGC		C		
	UUA		UCA		UAA		Och		UGA	Opa	A
	UUG		UCG		CAG		Amb		UGG	Try	G
C	CUU	Leu	CCU	Pro	CAU	His	CGU	Arg	U		
	CUC		CCC		CAC		CGC		C		
	CUA		CCA		CAA		Gln		CGA	A	
	CUG		CCG		CAG				CGG	G	
A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U		
	AUC		ACC		AAC		AGC		C		
	AUA		ACA		AAA	Lys	AGA	Arg	A		
	AUG	Met	ACG		AAG		AGG		G		
G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U		
	GUC		GCC		GAC		GGC		C		
	GUA		GCA		GAA	Glu	GGA		A		
	GUG		GCG		GAG		GGG		G		

Table 1.1: The standard genetic code table. Amino acids written in red color correspond to stop codons.

1.2.3 Single nucleotide polymorphism

Single nucleotide polymorphism (or SNP for short) is the most common type of genetic variations occurring within a population. It involves the different nucleotides at the single position in a DNA sequence between individuals or paired chromosomes. For

Figure 1.2: Illustration of the basic process of gene expression. Figure is downloaded from <http://en.wikipedia.org/wiki/Genetics>.

instance, a DNA fragment of an individual has sequence CCGTTTGA, while in another individual, the DNA fragment at the same position is sequenced as CCGTCTGA. There is a SNP (T/C) at the 5th position of the fragment. For human genome, there are roughly 10 million SNPs, which means that every 300 nucleotides contain one SNP on average (learn.genetics.utah.edu/content/pharma/snips/).

SNPs change DNA sequence, so they can cause the differences in the expressed amino acid sequences when they occur within protein-coding regions of genes, and hence may give rise to distinct functional proteins. However, only a small number of SNPs are responsible for a variety of traits, such as appearance, disease susceptibility or response to drugs. A single SNP may cause a Mendelian disorder, such as sickle-cell anemia. In many cases, multiple SNPs work together to cause complex genetic disorders, for instance, heart disease and diabetes. Therefore, detecting known SNPs and discovering new SNPs are of great importance in biomedical research to study the genetic reasons of diseases and develop the corresponding genetic therapies.

SNPs attract a large amount of studies due to their genetic significance. Many techniques have been utilized to detect SNPs in sample genomes, such as the application of

mass spectrometers. Here, we developed an accurate method to detect SNPs from mass spectrometry data and the details can be found in Chapter 2.

1.2.4 Epigenetics

In the past few decades, an increasing number of genomic problems were discovered and studied. However, it turns out to be impossible to understand the mechanisms of cellular function and regulation through studying genomes merely. For example, it is quite difficult to explain within the genome scenario that cells in different tissues have various functions but share the same genetic information. Thus, it is reasonable to study the mechanisms that are irrelevant to the changes of DNA sequence but crucial to maintaining gene regulation and genetic stability. Epigenetics is exactly the study of the heritable changes in gene expression, but not in DNA sequence. Epigenetic modifications alter the active status of genes (turned on or turned off) and thereby result in different expression of functional proteins.

Epigenetic changes occur naturally and regularly throughout lifetime. However, it has been observed that epigenetic modifications can also be influenced by environments in vitro, such as diets, stresses, pollutions and ages. Epigenetic modifications make every single living organism unique. On the other hand, many studies show that some epigenetic changes can be passed on to offspring. Such process is also called epigenetic inheritance, which can have an impact on evolution. Besides, epigenetic modifications can have damaging effects. Abnormal epigenetic changes can cause incorrect expression and thus lead to severe diseases, such as cancers and other disorders (such as Angelman syndrome).

Several types of inheritance systems, including DNA methylation and hydroxymethy-

lation, non-coding RNA (ncRNA) associated silencing and histone modification, play a role in initiating and sustaining epigenetic modifications [10]. DNA methylation occurs at the nucleotide level, which adds a methyl group to nucleotide base A or C and modifies the active status of genes. DNA hydroxymethylation in animal genomes refers to an oxidation product of the methylated cytosines. It has been observed that hydroxymethylcytosines exist extensively in brain tissues and have a strong effect on brain development [55]. Histones are proteins around which DNA winds in a chromosome. Histone modifications can alter the way DNA wraps around it and thereby affect which gene is active to express. ncRNA is a functional RNA molecule that is transcribed from DNA but is not translated into a protein. ncRNAs primarily regulate gene expression and are involved in DNA methylation, histone modification and gene silencing. Figure 1.3 depicts the mechanisms of these epigenetic changes.

In this thesis, we focus the discussion on DNA methylation, especially cytosine methylation. In the subsequent sections and Chapter 3, several aspects of the study of cytosine methylation will be introduced, including its biological significance, relevant research techniques, specific challenges, currently available computational solutions and our proposed method.

1.2.5 Cytosine methylation and hydroxymethylation

DNA methylation refers to the addition of a methyl group (CH_3) onto the cytosine or adenine nucleotide. Methylation of cytosine (or cytosine methylation) occurs in almost all living organisms while methylation of adenine (or adenine methylation) is only found in prokaryotic organisms, such as bacteria. Eukaryotes, including plants, animals, and human beings, draw most attention of researchers. Thus the cytosine

Figure 1.3: Epigenetic changes. Epigenetic changes modify the genomes but do not change the nucleotide sequence. DNA methylation and histone modification are the two typical examples of epigenetic modifications. Their mechanisms are described in the picture. Figure is taken from <http://en.wikipedia.org/wiki/Epigenetics>.

methylation is so far the best-studied epigenetic modification. Specifically, cytosine methylation means that a methyl group is added at the fifth carbon residue of the cytosine ring, so methylated cytosines are usually called 5-methylcytosines (shortened as 5mC). Cytosine methylation acts as a key factor in many essential biological processes, including embryonic growth, X chromosome inactivation, genomic imprinting, cancer development in mammals, regulation of gene expression, and transposon silencing in plant cells [58, 18].

Methylated cytosines are not distributed randomly along the DNA sequence. In most cases, cytosine methylation occurs in a CpG dinucleotide context, where a nucleotide C is linked with a nucleotide G by phosphate along DNA sequence. Previous studies showed that more than 70% of all CpGs are methylated in human genome [11]. It is well noticed that promoters of many genes contain a special region having high fre-

quency of CpG dinucleotides. Such special genomic regions are known as CpG islands (CGI). Generally, cytosines in the CpG islands of promoters are unmethylated if the genes are expressed, whereas CpGs of the coding regions are mostly methylated [2]. Methylation of CpGs within the gene promoters can result in transcriptional silencing, a feature found in many types of human cancers.

Methylated cytosines can also be found in non-CpG contexts, including CHG and CHH sites (H refers to any nucleotide but G). For most vertebrates, non-CpG methylation can only be found in specific tissues, such as embryonic stem cells. In contrast, cytosine methylation of plant genomes occurs in both CpG and non-CpG contexts. In *Arabidopsis* and other flowering plants, the significance of non-CpG methylation has been shown in regulating gene expression on a genome-wide scale [68].

To detect cytosine methylation of DNA, sodium bisulfite treatment is generally employed as a gold standard method. In this treatment, sodium bisulfite dominates the conversion of unmethylated cytosine into uracil, but does not affect methylated cytosine. According to the changes introduced by bisulfite treatment, methylation patterns can be determined directly through comparison to the DNA sequence before bisulfite treatment or the reference DNA sequence [30].

Hydroxymethylation of cytosine is an oxidation process of methylated cytosines. It is mainly studied and discussed in animal genomes. Although a large body of experimental evidence suggests the critical importance of hydroxymethylcytosine (or 5hmC for short), its exact biological function still requires a lot of research. The existence of 5hmC may cause the failure of the detection of 5mC based on the standard bisulfite treatment, because hydroxymethylcytosines do not react to the chemical conversion reagent [50]. In other words, 5hmCs are not converted to uracils after the standard bisulfite treatment. Two solutions are available so far to distinguish between 5mC and

5hmC: one is oxidizing hydroxymethylcytosine to activate its reaction to bisulfite conversion; and the other employs the TET-assisted bisulfite sequencing which converts the methylated cytosines to bisulfite-sensitive residues [46].

However, in current research studies, the impact of hydroxymethylation is always ignored due to its unclear biological function and lower level of occurrence compared to methylation [46, 55]. Therefore, we do not consider hydroxymethylation in our work on analyzing cytosine methylation from bisulfite sequencing reads.

1.2.6 Brief introduction of next-generation sequencing techniques

DNA sequencing is the process of establishing the precise order of the four nucleotides - A, C, G, and T - within a DNA strand. The first most widely used DNA sequencing method is the Sanger sequencing that was developed by Frederick Sanger and his colleagues in 1977. Briefly speaking, Sanger sequencing copies a piece of cloned DNA with a DNA primer and stops the replication process by using one of the four modified dideoxynucleotides (ddATP, ddCTP, ddGTP, and ddTTP) in each of the four independent reactions. The resulting DNA fragments are heat denatured and separated by size using gel electrophoresis. Finally, the DNA sequence can be directly read according to the DNA bands visualized by auto-radiography or UV light. Figure 1.4 describes the schematics of Sanger sequencing.

In order to obtain sequence information for large-scale projects with lower cost and higher efficiency, the development of DNA sequencing technologies entered the era of high-throughput sequencing (or next-generation sequencing, short for NGS) in late 1990s. NGS technologies produce thousands or millions of sequences using parallel sequencing approaches to reduce the total cost. Many NGS techniques have been

Copyright M.W.King 1996

Figure 1.4: Sanger sequencing. Figure is obtained from dwb.unl.edu/Teacher/NSF/C08/C08Links/www.piopio.school.nz/nolmed.htm.

commercially developed and used since 2005 [41, 43, 49].

In our study, we mainly focus the analysis on sequencing data generated by the Genome Analyzer system of Illumina (Solexa). Figure 1.5 illustrates the three critical processes of sequencing DNA by Illumina Genome Analyzer [43, 49]. In the first sample preparation step, specific adapters are attached to both two ends of each DNA fragment, which form the sequencing library. The adapted library is amplified to generate the detectable sequencing features. In the subsequent step, the sequencing library is immobilized on the oligo-derivatized surface of a flow cell, a planar and fluidic device. The flow cell can create abundant primers on its inner surface. The immobilized se-

quencing library is then amplified on a solid support by Bridge-PCR (polymerase chain reaction). Basically, Bridge-PCR starts with forming a bridge structure by hybridizing an immobilized sequencing library fragment with a primer on the surface of a flow cell. Such bridge structured molecule then acts as a template to generate its complementary strand. Once the bridged double-strand DNA is created, a denaturing reagent is employed to free both strands. After repeated reagent flush cycles of denaturation, annealing, extension, and wash, multiple DNA copies or clusters are produced on each flow cell lane. In the last step, the Illumina Genome Analyzer utilizes a sequencing-by-synthesis approach to determine the DNA sequence of each cluster based on four fluorescent nucleotides. Such approach enables us to read one base each time along the DNA sequence from the image panel.

Compared to other NGS platforms, the Illumina Genome Analyzer can produce millions of reads in 36 - 300 bp length with less time and cost [56, 41]. Moreover, it generally creates few errors in a read and in most cases the errors are base substitutions. In our study of DNA methylation, we consider reads generated by the Illumina platform after the sample genome is treated by sodium bisulfite conversion.

Besides next-generation sequencing technologies, DNA sequences can be detected by other methods that utilize the physical or chemical properties of DNA molecules. One of these methods is based on mass spectrometry, especially matrix-assisted laser desorption ionization (MALDI) time-of-flight (TOF) mass spectrometry (MS) [9]. We applied the data from MALDI-TOF MS to detect SNPs in a sample DNA sequence. Figure 1.6 depicts the general processes of using MALDI-TOF MS to analyze biomolecules (such as DNA, proteins) or large organic molecules. The sample molecules mixed with some matrix material are immobilized on a metal surface. Then the molecules are ionized by a pulsed laser and accelerated in an electromagnetic field. During this step, the

ions will have the same amount of kinetic energy if they have the same charge. According to the classical electrodynamics, two particles with the same mass-to-charge ratio (denoted by m/z) move in the same path in a vacuum when subjected to the same electric and magnetic fields. Therefore, the smaller an ion is, or the higher an ion is charged, the faster it arrives at the detector. Once the ions reach the detector, a signal peak is generated, resulting in a spectrum at the end [45, 34]. Each signal peak implies a group of ions having the similar mass-to-charge ratios and hitting the detector within a time unit. Moreover, the height of a signal peak roughly indicates the number of ions arriving at the detector. Accordingly, the sequencing information of the sample DNA molecules can be deduced by comparing their experimental spectrometry with the theoretical spectrometry of the reference DNA sequence.

Figure 1.6: MALDI-TOF-MS. Figure is from [45]

Now I give a brief discussion on the mass unit in terms of mass-to-charge ratio (m/z) applied in mass spectrometry. Here, m refers to the mass number, measured on

the carbon-12 scale (i.e., a carbon-12 weighs 12 Da) and z is the charge number of an ion. So if a 2+ ion has the mass 100 Da, its mass-to-charge ratio is $m/z = 50$. For ionized molecules from MALDI-TOF mass spectrometer, they are ideally charged by one proton that has one positive electric charge. Molecules that gain multiple protons are rarely found [1]. Therefore, m/z is usually treated equally to Da.

Figure 1.5: Sequencing Approach of the Genome Analyzer system [43].

Chapter 2

SNP Detection Using Mass Spectrometry Data

2.1 Introduction

Single-nucleotide polymorphism (SNP) can be defined as a substitution of one single nucleotide for another at a specific genomic locus. It is among the most important genetic factors that contribute to human evolution, diseases and biological functions. Many applications such as clinical diagnosis and virus identification rely heavily on the accurate detection of SNPs in the sample sequences of interest.

Over the past thirty years, many different methods have been developed for SNP detection, including denaturing gradient gel electrophoresis (DGGE) [14], chemical or enzymatic cleavage at mismatches sites [48], single strand conformation polymorphism (SSCP) [53], denaturing high performance liquid chromatography (DHPLC) [52], hybridization to oligonucleotide arrays [6], matrix assisted laser desorption/ionization time-of-flight (MALDI-TOF) mass spectrometry (MS) [19, 28, 63], direct DNA se-

quencing [60, 16], and recently emerging next-generation sequencing (NGS) technologies [44, 37, 38]. While every existing method has certain limitations, the MALDI-TOF MS based approach compares favorably with others in terms of high-throughput, time- and cost-efficiency, and reproducibility [28, 63, 4, 12]. As discussed in [12], although NGS technologies have been rapidly developed in recent decades, the practical running cost is still very high due to its complex assay procedure compared to mass spectrometry base methods. Moreover, NGS technologies generally require very long time in sample preparation, which especially prevents their application in clinical microbiology. However, MALDI-TOF MS is able to analyze whole bacterial cells without sample preparation so that the time is dramatically shortened to get the results of the bacterial culture and further to control the spread of an epidemic with little delay [12]. In this chapter, the proposed approach to detect SNPs was inspired by the study of influenza A H1N1 virus using MALD-TOF MS conducted in the lab of Dr. Tang Kai.

2.1.1 Sequencing with base-specific cleavage and MS

The MALDI-TOF mass spectrometry-based approach for SNP detection proceeds with the following typical data acquisition procedure. Polymerase chain reaction (PCR) is first employed to amplify the target sample DNA sequence with some promoter tags incorporated to the 5' ends of primers. In experiments, the PCR primers carrying different promoter sequences may be used in order to produce the transcripts of both DNA strands in separate strand-specific reactions. Some experiments selected the T7 and SP6 promoter sites that are carried by each forward PCR primer and reverse PCR primer, respectively [19, 63]. In [28], the T3 promoter sequence was combined to the 5' end of the reverse primer. Different from the above research, a universal primer system

was used for PCR amplification in [22] to reduce the primer costs, because only one type of promoter tag (T7) was required. Using such universal primer system, the PCR product is then subjected to the shrimp alkaline phosphatase (SAP) treatment, which should degrade the unused dNTP. After the SAP treatment, the PCR product is *in vitro* transcribed with mutant T7 transcriptase to generate single-strand RNA transcripts.

In the next step, a single-strand RNA molecule is cleaved by a base-specific enzymatic reaction using RNase T1 (e.g. [19] and [28]) or RNase A (e.g. [22]) or both (e.g. [63]). RNase T1 cuts the RNA sequence exactly after every G, whereas RNase A cleaves specially after the pyrimidine residue, that is, C or U. In our study, we utilized RNase A combined with the non-cleavable dCTP and dTTP/dUTP nucleotides in two independent transcription experiments. One experiment substitutes rCTP by dCTP and the other uses dTTP or dUTP instead of rUTP. Due to the substitution of rNTPs by non-cleavable dNTPs during the transcription of either forward or reverse strand of the sample DNA sequence, the cleavage reactions can be performed specifically to each of four RNA bases.

Finally, MALDI-TOF MS is applied to the cleavage fragments giving rise to four base-specific mass spectra. We extract the list of signal peaks that correspond to masses and intensities [25] from each sample spectrum and utilize such information in the downstream detection of SNPs. All of the above experimental processes are summarized in Figure 2.1.

2.1.2 Detecting SNPs from mass spectra

Mass spectra corresponding to four base-specific cleavage reactions can be utilized to detect SNPs, because a single nucleotide substitution may lead to up to 10 mass

Figure 2.1: Schematic of sequencing with base-specific cleavage reaction and MALDI-TOF MS.

spectral changes [63]. We can use an example to illustrate the mass spectral changes caused by a SNP. Suppose that $W = AACAACGTGGCCAT$ is a wild-type DNA sequence, and an A/G SNP occurs at the forth A in a sample sequence S . That is, $S = AACACGTGGCCAT$. After the cleavage reaction specific to C on the forward RNA strand of S , a mass spectrum can be obtained, consisting of the masses of fragments $\{AAC, AGC, GTGGC, C, AT\}$. Comparing the sample spectrum with that of the wild-type genome W , which comprises the masses of fragments $\{AAC(\times 2), GTGGC, C, AT\}$,

the signal peak corresponding to the mass of *AAC* turns to be shorter and an additional signal peak corresponding to the mass of *AGC* appears. In other words, two changes exist in the mass spectrum specific to cleaving after C. Similarly, in the cleavage reaction specific to U on the forward RNA strand (equivalent to cleaving after T on the forward DNA strand), two changes can be observed. One is the disappearance of a signal peak corresponding to the mass of the wild-type fragment *AACAACGT(U)* and the other is the appearance of a new signal peak corresponding to the mass of the sample fragment *AACAGCGT(U)* resulting from the SNP. When we cleave the reverse RNA strand specific to C, it is equivalent to cut the forward DNA strand after every G. We can thereby achieve the sample mass spectrum corresponding to the fragments {*AACAG, CG, TG, G, CCAT*}. Under the same cleavage reaction, the wild-type mass spectrum contains the masses of the fragments {*AACAACG, TG, G, CCAT*}. Obviously, the wild-type signal peak with respect to *AACAACG* is missing while two additional signal peaks with respect to *AACAG* and *CG* appear in the sample spectrum, which shows us three changes. In the last case that cleavage reaction is performed specific to A (i.e., cleaving after each U on the reverse RNA strand), three similar changes can also be observed: the disappearance of the wild-type signal peak of fragment *CGTGGCCA*, the appearance of a new signal peak with respect to the sample fragment *GCGTGGCCA*, and the reduction of signal intensity corresponding to fragment A.

Accordingly, given a reference DNA sequence, we may generate its theoretical mass spectra by performing *in-silico* base-specific cleavage reactions and mass spectrometry analysis. In our study, we assume that the sample sequence differs from the reference sequence by only a few SNP mutations. Thus, these SNP mutations can be implied from the discrepancies between the experimentally measured mass spectra of the sam-

ple sequence and the *in-silico* predicted mass spectra of the reference sequence. The major discrepancies that can be utilized for reliable SNP detection are the appearance of unexpected signal peaks and the disappearance of expected signal peaks in the measured mass spectra. In particular, we call a peak in the measured mass spectra the *additional peak* if it appears in one of the four measured mass spectra but cannot be found in the predicted mass spectrum with respect to the same cut base.

2.1.3 Existing methods reviews

To detect SNP mutations, visual interpretation of mass spectra is often employed [19, 28, 63], which is very labor-intensive and time-consuming. To facilitate the automatic detection of SNP mutations from the mass spectrometry data, two software packages have been previously developed. A brief introduction is given below to each tool.

RNaseCut

RNaseCut [28] is freely available at <http://www.vetmed.uni-muenchen.de/gen/forschung.html>. It computes all the possible mutation candidates that are able to interpret a different mass peak in the measured mass spectrometry. However, there is no further automatic step to make confirmation of true mutations, thus manual validation is still needed.

MassARRAYTM

The second existing software package is the proprietary MassARRAYTM SNP Discovery software package from Sequenom, Inc. This software basically implemented Böcker's algorithm [4], which is discussed in the next section. Compared to RNaseCut,

it goes a step further after all possible mutation candidates are found out. In order to determine the true mutation SNPs, it applies a scoring and thresholding procedure to evaluate each candidate mutation. Although the software package provides a fully automatic process for SNP detection, it is difficult to obtain this commercial software at low expense, and hence only a few labs are using this software, as far as we know.

2.1.4 Our contribution

In this chapter, we present a new algorithm for accurate detection of SNP mutations from mass spectrometry data. Compared to Böcker’s algorithm, it is a more effective way to integrate the information in four complementary base-specific mass spectra. As mentioned above, Böcker’s algorithm employs a two-step procedure which first generates all mutation candidates and then scores them. In the first step, the additional peaks in the measured mass spectra are examined independently rather than collectively. As a consequence, a large number of spurious mutations are produced as candidates. These spurious mutations will inevitably confound the scoring analysis in the second step, making the true mutations less likely to be detected. In contrast, our algorithm adopts an iterative and progressive procedure. It repeatedly identifies SNP mutations that have most likely occurred in the sample sequence, while at the same time it progressively updates the reference sequence by correcting these mutations. As a result, the earlier a mutation is detected, the more likely it is true. Moreover, the mutations detected earlier may largely determine the mutations that would be detected later, thereby avoiding many spurious mutations to be evaluated.

Our new algorithm has been implemented in a program called SNPMs. See Figure 2.2 for the schematic outline of its data acquisition and analysis. To assess the

Figure 2.2: Schematic of SNPMs.

performance of SNPMs as a tool to detect SNPs, we carried out several comparative experiments on both simulated and real biological datasets. The test results clearly show that SNPMs outperforms RNaseCut, the only alternative and publicly available program to date. In particular, SNPMs can successfully detect eight out of ten true SNP mutations that have occurred in the coding region of gene Hemagglutinin (HA) from our collaborator's lab sample of the influenza A H1N1 virus strain WSN/33. There is no comparative evaluation with Böcker's algorithm in this study, because we were not able to obtain a copy of the proprietary MassARRAY™ SNP Discovery software package for experiments.

2.2 Preliminaries

2.2.1 *In-silico* predicted mass spectrum

To detect SNP mutations from mass spectrometry data, a reference DNA sequence is required. Before we predict the four complementary mass spectra with respect to

different cleavage reactions, it is worth noting that each peak in the mass spectrometry indicates the mass and intensity of a cluster of DNA fragments generated from some base specific cleavage reaction. Ideally, we assume that each peak has a sufficient high intensity value if it is generated by the DNA fragments; while the signal peaks corresponding to noises have very low intensity values.

In order to calculate the *in-silico* predicted mass value of a DNA sequence f , we define the *base composition* of f to be a map $comp : \Sigma \rightarrow \mathbb{N}$, where $\mathbb{N}$ is the set of non-negative integers. In the particular case of DNA, $comp$ actually counts the numbers of A, C, G, and T in the sequence f , respectively. That is, if f contains i As, j Cs, k Gs, and l Ts, where $i, j, k, l \in \mathbb{N}$, then $comp(A) = i$, $comp(C) = j$, $comp(G) = k$, $comp(T) = l$. Specially, we denote the base composition of f to be $comp = A_i C_j G_k T_l$. Moreover, it should be noted that two DNA sequences with different orders of nucleotides can have the same base composition. For example, DNA sequence $GCCACATG$ and sequence $CACGGTAC$ have the same base composition of $A_2 C_3 G_2 T_1$.

Making use of the concept of base composition, the *in-silico* predicted mass spectrometry can be constructed. Consider a cleavage reaction with respect to the cut base x . If a cleavage fragment f has the base composition of $A_i C_j G_k T_l$, then we can compute its *in-silico* predicted mass value $m_x(f)$ as the following

$$m_x(f) = i \cdot m(A) + j \cdot m(C) + k \cdot m(G) + l \cdot m(T) + m_0$$

where $m(\cdot)$ is the mass value of the respective base (given $m(A) \approx 313.06$ Da, $m(C) \approx 289.05$ Da, $m(G) \approx 329.05$ Da and $m(T) \approx 304.05$ Da), and m_0 is an experiment-specific mass intermediate. For instance, if the endonuclease RNase A is used in the cleavage reaction, we have $m_0 = 18$ which accounts for an H at the 5' terminus and an OH at the

3' phosphate. If the endonuclease RNase T1 is instead used, then we shall have $m_0 = 0$ because a terminal 2',3'-cyclic phosphate is usually generated as a hydrolysis intermediate which leads to a loss of water. Accordingly, the four complementary *in-silico* predicted mass spectra of a reference DNA sequence can be achieved by computing the *in-silico* predicted mass values of all different cleavage fragments resulting from the corresponding base-specific cleavage reactions.

2.2.2 Experimentally measured mass spectrum

MALDI-TOF mass spectrometry is one of the most useful techniques for determining the mass of biomolecules. In our experiments of SNP detection, it is applied to the products of a cleavage reaction, resulting in a sample spectrum that correlates mass and signal intensity of the cleavage fragments [25]. The sample spectrum is then analyzed to extract a list of signal peaks whose attributes include mass, relative intensity, and signal-to-noise ratio. The above mass spectrometry assay is applied to the cleavage reactions specific to all four bases, resulting in four complementary mass spectra.

There is a limited mass range in which a cleavage fragment can be reliably detected by current MALDI-TOF MS. A typical mass range is from 1,000 Da to 10,000 Da so that the cleavage fragments of length only from 3 bases to approximately 30 bases can be detected. Longer cleavage fragments tend to have their signals lost due to poor detection efficiency, while fragments shorter than 4 bases fall in the mass range where matrix peaks dominate.

An experimentally measured mass spectrum typically contains a mixture of peaks that represent signals and noises respectively. In the current implementation of SNPMs, we take a simple thresholding approach to pick signals from noises. A mass peak is

picked as signal when its signal-to-noise ratio exceeds a user-defined threshold (the default is 20). A robust peak picking method, such as the one in [8], can be used, which is expected to further improve the accuracy of SNPMs to detect SNPs.

Ideally, every peak in a measured mass spectrum shall have at least one cleavage fragment to generate it. In other words, we shall find a cleavage fragment whose *in-silico* predicted mass value is equal to the measured mass value of each peak (within an instrument-specific mass tolerance). In practical experiments, however, the measured mass spectrum usually includes a number of signal peaks unrelated to the sample DNA sequence, because of the impossibility of perfect experimental conditions. Therefore, it is always necessary to calibrate the experimentally measured mass spectrum. One basic calibration method, which is known as internal calibration, adds the standard molecules with known masses into the sample and obtains a mass spectrum of the mixture through MALDI-TOF. The mass peaks of the standard molecules are firstly identified and employed to calibrate the whole spectrum. The mass spectrum after internal calibration can be highly accurate, but the sample mass spectrometry peaks might be suppressed by this approach [21, 67].

In our study, we have no standard molecules mixed with the cleavage fragments of the sample DNA sequence. In this case, a MALDI-TOF mass spectrometer may have a constant mass shift across all the peaks in a mass spectrum. In SNPMs, we firstly infer the most possible base compositions whose *in-silico* predicted mass values approximate each measured mass value. Then we estimate this constant mass shift as the average difference between the measured mass values and their closest *in-silico* predicted mass values inferred previously. We use the estimate value to calibrate the measured mass values of peaks. After this mass calibration, we delete from the mass spectrum those peaks that still could not be generated by any cleavage fragment.

In the description below, we use $\mathcal{M}_\Sigma$ to denote the set of signal peaks from the four complementary mass spectra after peak calling and mass calibration. The mass value and signal-to-noise ratio of a peak p can be retrieved by using the functions $m(p)$ and $r(p)$, respectively.

2.2.3 Explanation of measured mass peaks

We say a cleavage fragment f can *explain (interpret or yield)* a measured mass peak p with respect to the same cut base x if the *in-silico* predicted mass value of f is equal to the measured mass value of p up to a small precision (e.g., $\pm 0.01\%$ for a reflection TOF instrument). Furthermore, we say a reference sequence s can *explain (or interpret)* a measured mass peak p if there exists a cleavage fragment in s that can explain p (with respect to the same cut base).

Given a reference sequence s and four complementary measured mass spectra $\mathcal{M}_\Sigma$ (generated by an unknown sample sequence), let $\mathcal{M}_\Sigma(s)$ be the maximum-cardinality subset of $\mathcal{M}_\Sigma$ in which every mass peak can be yielded only by a unique cleavage fragment of s . For instance, if $s := AACAACT$ and $\mathcal{M}_\Sigma := \{m_A(CGA), m_A(CT), m_C(AAC), m_C(GAC), m_G(AACG), m_G(ACT)\}$ (corresponding to the unknown sequence $AACGACT$), then $\mathcal{M}_\Sigma(s) := \{m_A(CT)\}$. In this example, we assume that only the cleavage fragments of length from two bases to four bases can be detected. Therefore, there is no mass peak with respect to the cut base T . Observing the measured mass spectra, only $m_A(CT)$ and $m_C(AAC)$ can be explained by cleavage fragments in s . However, $m_C(AAC)$ can be yielded by either AAC at position 0 or AAC at position 3, so $m_C(AAC)$ cannot be included in $\mathcal{M}_\Sigma(s)$.

With this subset $\mathcal{M}_\Sigma(s)$, we next define a score that reflects how well the reference

sequence s can explain the measured mass spectra $\mathcal{M}_\Sigma$. That is,

$$r(s, \mathcal{M}_\Sigma) = \sum_{p \in \mathcal{M}_\Sigma(s)} r(p),$$

where $r(p)$ is the signal-to-noise ratio value of a measured mass peak p retrieved from the sample spectrum. Note that the higher the score $r(s, \mathcal{M}_\Sigma)$ is, the better the reference sequence s would explain the measured mass spectra $\mathcal{M}_\Sigma$. This score plays an important role in the algorithm in our software package SNPMS.

2.3 Algorithm in SNPMS

To detect SNPs from the four complementary base-specific mass spectra with high accuracy, we devised an iterative greedy algorithm. Its main idea is to repeatedly identify the optimal potential SNP mutations while progressively updating the reference sequence by correcting these SNP mutations, until no more potential SNP mutations can be found. When the execution of the algorithm terminates, a list of SNP mutations that might most possibly occur in the sample DNA sequence is reported. The algorithm is summarized in Algorithm 1, which is discussed in more detail in the following section.

2.3.1 Discussion of algorithm

The algorithm begins with an initialization procedure (line 1 to line 4). In this step, we first find all the cleavage fragments in the reference sequence s that are necessary for s to explain some peaks in the mass spectra $\mathcal{M}_\Sigma$. Precisely, after these cleavage fragments are attached by the specific cut bases at their both ends, they are able to explain peaks in the mass spectra subset $\mathcal{M}_\Sigma(s)$. The bases of these fragments are then

Algorithm 1 SNPMs($s, \mathcal{M}_\Sigma$)

Input: A reference sequence s and four complementary mass spectra $\mathcal{M}_\Sigma$ of an unknown sample sequence

Output: A list Δ of potential SNP mutations that might have taken place in the sample sequence

```
1:  $\Delta \leftarrow \text{null}$ 
2: Calculate  $\mathcal{M}_\Sigma(s)$ .
3: Fix bases in  $s$  needed to explain peaks of  $\mathcal{M}_\Sigma(s)$ .
4:  $\mathcal{M}_\Sigma \leftarrow \mathcal{M}_\Sigma \setminus \mathcal{M}_\Sigma(s)$ .
5: repeat
6:    $\delta \leftarrow \text{null}$ 
7:    $r(\delta) \leftarrow 0$ 
8:   for each permissible base substitution  $\delta'$  do
9:     Apply base substitution  $\delta'$  to  $s$  and get  $s'$ 
10:    Calculate  $\mathcal{M}_\Sigma(s')$ 
11:     $r(\delta') = r(s', \mathcal{M}_\Sigma) = \sum_{p \in \mathcal{M}_\Sigma(s')} r(p)$ 
12:    if  $r(\delta') > r(\delta)$  then
13:       $\delta \leftarrow \delta'$ 
14:       $r(\delta) \leftarrow r(\delta')$ 
15:    end if
16:  end for
17:  if  $\delta \neq \text{null}$  then
18:    Add  $\delta$  to the set  $\Delta$ .
19:    Update  $s$  by applying  $\delta$  to it.
20:    Calculate  $\mathcal{M}_\Sigma(s)$ .
21:    Fix bases in  $s$  needed to explain peaks of  $\mathcal{M}_\Sigma(s)$ .
22:     $\mathcal{M}_\Sigma \leftarrow \mathcal{M}_\Sigma \setminus \mathcal{M}_\Sigma(s)$ .
23:  end if
24: until  $\delta == \text{null}$ 
25: return  $\Delta$ 
```

labeled as being in the *fixed* status, simply indicating that they will not be subject to any further modification. Meanwhile, we update the measured mass spectra $\mathcal{M}_\Sigma$ by deleting those mass peaks of $\mathcal{M}_\Sigma(s)$ from $\mathcal{M}_\Sigma$, that is, $\mathcal{M}_\Sigma := \mathcal{M}_\Sigma \setminus \mathcal{M}_\Sigma(s)$. This update can be performed because the reference sequence s does not need any SNP mutation to explain any mass peak of $\mathcal{M}_\Sigma(s)$.

After initialization, an iterative greedy procedure is then invoked (line 5 to line 24). At each iteration, we first identify an optimal *potential* SNP mutation from all the *permissible* base substitutions that could be made to the reference sequence s . Here, a base substitution is *permissible* if it can be applied to a base of s that is not yet labeled as being in the fixed status. For each permissible base substitution δ , we calculate a score $r(\delta)$ as

$$r(\delta) = r(s', \mathcal{M}_\Sigma) = \sum_{p \in \mathcal{M}_\Sigma(s')} r(p),$$

where s' is the reference sequence s after the base substitution δ is applied to it. As we can see, this score can offer a rough estimate on how much a base substitution could aid in the explanation of the mass peaks in $\mathcal{M}_\Sigma$. Therefore, a reasonable choice of the optimal potential SNP mutation is the base substitution with the highest score $r(\delta)$.

Ideally, only the true SNP mutations could achieve the highest scores. However, in practical experiments, it might be observed that more than one permissible base substitutions achieve the same highest score. Such cases may be resulted from the limited mass range of current MALDI-TOF mass spectrometers. It is possible that the true base substitution leads to a cleavage fragment with either too small or too large mass value. For such case, we currently select the base substitution that is firstly detected to be the optimal potential SNP mutation.

Once the optimal potential SNP mutation is chosen, we apply it to s to obtain a

new reference sequence (still denoted as s). Then, like what we have already done in the initialization step, find all the fragments in the new reference sequence s that are necessary for s to explain some peaks in $\mathcal{M}_\Sigma(s)$ and label their bases as being in the fixed status. Meanwhile, we update the mass spectra $\mathcal{M}_\Sigma$ by deleting those mass peaks of $\mathcal{M}_\Sigma(s)$ from $\mathcal{M}_\Sigma$, that is, $\mathcal{M}_\Sigma := \mathcal{M}_\Sigma \setminus \mathcal{M}_\Sigma(s)$. The above procedure is iterated until no more potential SNP mutation can be found. At that time, the reference sequence s can no longer explain any mass peaks in $\mathcal{M}_\Sigma$ (if it is still not empty), even after a single base substitution is applied to s .

Note that the iterative procedure can always converge to have δ equal to null. Every time when a permissible base substitution achieves the highest score, the reference sequence is updated by applying this optimal potential SNP mutation and fixing the corresponding bases that are required to uniquely explain the corresponding mass peaks. It implies that we have no chance to select the same optimal mutation at different iterative steps. Therefore, the iterative procedure always stop at the moment either when all bases of the reference sequence are fixed or when all the remained permissible base substitutions fail to achieve non-zero scores.

The consuming time of whole iterative procedure depends heavily on the number of SNPs in the sample sequence. The fewer SNPs exist in a sample sequence, the more bases can be fixed in the initialization step, and hence the less time it requires to select the optimal potential SNP mutations. In contrast, the more SNPs occur in a sample sequence, the more permissible base substitutions have to be checked and scored, and therefore, the more time the whole procedure costs. The relationship between running time and the number of SNPs are proved by our simulation experiments. The software SNPMS runs on a personal computer with processor Pentium(R) 4 CPU 3.20GHz. When the sample sequence contains 5 SNPs, the average running time is 18.14s. When the

number of SNPs in a sample sequence increases to 10, the average running time of SnpMs grows to 44.98s.

Finally, the entire execution of the algorithm terminates with a list of potential SNP mutations reported (line 25). The last reference sequence s may be returned as the putative sample DNA sequence t that we might have used for the experimental data acquisition.

We implemented the above algorithm in a program called SnpMs using the C++ programming language. It is freely available at <http://www1.spms.ntu.edu.sg/~chenxin/SnpMs>.

2.3.2 Detecting SNPs in close vicinity

It becomes increasingly challenging to detect SNPs when they occur in close vicinity, especially when they are inside the same cleavage fragment. The solution provided in [4] is to increase the sequence variation cost, that is, to increase the number of mutations permitted in a cleavage fragment to interpret an observed mass peak. However, it will inevitably introduce a large number of spurious SNPs required to be evaluated in the later stage of their algorithm, which may adversely prevent the true SNPs from being detected.

In our algorithm presented above, a SNP can be detected only when it is the only sequence variation in a cleavage fragment. In other words, we will not use a cleavage fragment with two or more SNPs to explain an observed mass peak during each iteration of the algorithm execution.

Fortunately, our algorithm employs an iterative and progressive procedure which still allows us to detect SNPs in close vicinity, even when they occur inside the same

cleavage fragment. To illustrate this by an example, let the reference sequence be $s := GCACGAG$ and the unknown sample sequence be $t := GCTTGAG$. Thus, the four complementary mass spectra measured for the sample sequence is

$$\mathcal{M}_\Sigma = \{m_A(GCTTG), m_C(TTGAG), m_G(CTT), m_T(GAG)\}.$$

Here, we suppose that the cleavage fragments with less than three bases cannot be detected by MALDI-TOF mass spectrometer. Compared to the reference sequence s , there are two adjacent SNP mutations that occurred in t : one is the base substitution A/T at position 3 and the other is the base substitution C/T at position 4 (when we count the positions starting from 1).

According to [4], these two SNP mutations are not *independent*. Specially, if two SNP mutations δ_1 and δ_2 are independent with each other, the sum of the changes of base compositions resulting from δ_1 and the changes of base compositions resulting from δ_2 includes all the changes of base compositions resulting from both of them with respect to all cut bases. In this example, with respect to the cut base $x = T$, the *in-silico* base compositions of the reference sequence s should be the set $C_{0,T} = \{A_2C_2G_3\}$. If $\delta_1 = T$ is applied to position 3, the sequence is updated to be $s_1 := GCTCGAC$ and the resulting set of base compositions shall be $C_{1,T} = \{C_1G_1, A_1C_1G_2\}$. Similarly, if $\delta_2 = T$ is applied to position 4, s is changed to $s_2 := GCATGAG$ and the set of base compositions of s_2 is $C_{2,T} = \{A_1C_1G_1, A_1G_2\}$. When both SNP mutations are applied in the reference sequence s , the base compositions corresponding to the sample sequence t are contained in set $C_{1,2,T} = \{C_1G_1, A_1G_2\}$. Here $C_{1,2,T}$ means the set of base compositions with respect to cut base T after δ_1 and δ_2 are both applied to s . Apparently, $C_{1,2,T} \subseteq C_{1,T} \cup C_{2,T}$, which implies that δ_1 and δ_2 are independent with respect to cut base

T. However, when we consider cut base $x = A$, they turn to be dependent. In this case, the sets of base compositions corresponding to s , s_1 , s_2 , and t shall be $C_{0,A} = \{C_1G_1, G_1\}$, $C_{1,A} = \{C_2G_2T_1, G_1\}$, $C_{2,A} = \{C_1G_1, G_1T_1, G_1\}$, and $C_{1,2,A} = \{C_1G_2T_2, G_1\}$, respectively. It is obvious that $C_{1,2,A} \not\subseteq C_{1,A} \cup C_{2,A}$ because neither $C_{1,A}$ nor $C_{2,A}$ contains the base composition of $C_1G_2T_2$. Similarly, we can prove that these two SNP mutations are not independent with respect to the cut base $x = C$ or G , either.

If we make use of the Böcker's algorithm [4], we have to increase the sequence variation cost to two mutations to explain each mass peak in $\mathcal{M}_\Sigma$. As a consequence, to explain the mass peak $m_A(GCTTG)$, the base substitution A/T at position 3 can be treated as a candidate SNP mutation and the base substitution C/T at either position 2 or position 4 might be the other candidate SNP mutation. These two spurious SNP candidates cannot be distinguished until the scoring step is performed.

However, because these two SNP mutations are independent with respect to the cut base $x = T$, it permits our algorithm to detect both SNP mutations one by one, without the need of increasing the sequence variation cost to two mutations as in Böcker's algorithm [4]. To be specific, in the first iteration of our algorithm, we may find the base substitution C/T at position 4 as the optimal potential SNP mutation to explain the measured mass peak $m_T(GAG)$. At the beginning of the second iteration, we thus have both the reference sequence s and the mass spectra $\mathcal{M}_\Sigma$ updated as follows

$$s := GCAT\textcolor{blue}{T}GAG$$

and

$$\mathcal{M}_\Sigma = \{m_A(GCTTG), m_C(TTGAG), m_G(CTT)\}.$$

Then, the base substitution A/T at position 3 shall be identified as the new optimal

potential SNP mutation as it can explain all the mass peaks in $\mathcal{M}_\Sigma$. At the end of the second iteration, we have the new reference sequence

$$s := GCTT\textcolor{blue}{G}AG$$

and the empty set $\mathcal{M}_\Sigma$. As it can be seen, our algorithm has successfully detected the two SNP mutations without inducing any spurious SNPs.

2.4 Results

As mentioned in the introduction, there are two software tools for SNP discovery using base specific cleavage and mass spectrometry in the literature. The first one is called RNaseCut, which can be freely downloaded. The second one is the proprietary MassARRAYTM SNP Discovery software package from Sequenon, Inc. Its algorithmic details were presented in the reference [4]. Unfortunately, we were not able to obtain a copy for our experiments in this study.

2.4.1 Results of simulated data

We carried out several tests on simulated data to assess the effectiveness of our iterative algorithm for SNP detection. In the first test dataset, we randomly generate a DNA sequence containing 653 bases and use this sequence as the reference sequence. Then we simulate a sample sequence by adding five random SNP mutations in the reference sequence. Furthermore, the mass spectra of the sample sequence with respect to four base-specific cleavage reactions are simulated through the *in-silico* computation (refer to Section 2.2.1). Due to the mass range limit of MALDI-TOF mass spectrometer, only

the mass peaks that correspond to cleavage fragments of at least 3 bases are included in the mass spectra. After the test dataset is simulated, both SnpMs and RNaseCut will take the reference sequence and the simulated experimental mass spectra as input for SNP detection.

Their detection results are then validated with the true SNP mutations using the following three performance measures – sensitivity, precision and F-measure. They are defined as

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

and

$$F\text{-measure} = \frac{2 \times Sensitivity \times Precision}{Sensitivity + Precision}$$

where TP represents the number of true positives, FN the number of false negatives, and FP the number of false positives. In detail, both softwares report the possible SNP mutations that they can detect, together with the location of each candidate SNP according to the reference sequence. If some true SNP mutation does occur at a location outputted by a software, regardless of the substitution bases, then we say the software report one true positive result. In contrast, if there is no true SNP mutation at a reported position, the result shall be defined as a false positive. Furthermore, if the location of a true SNP mutation is not detected by a software, the software will have a false negative. Therefore, the sensitivity score evaluates the percentage of true SNP mutations a software can detect, while the precision score reflects the percentage of detected mutations that are true. Moreover, the F-measure score, which is the harmonic mean of sensitivity and precision, can be used to evaluate the overall performance of a SNP detection

software. In other words, the higher F-measure score a software obtains, the better it performs for detecting SNP mutations.

Finally, we generated 100 random data instances as above, and computed the means and variances of the respective performance measures. The experimental results for the above test dataset are summarized in Table 2.1. It is easy to see that SNPMs achieves a lower average sensitivity score than RNaseCut (0.78 vs 0.91). However, the average precision score of RNaseCut is only ~ 0.06 , significantly lower than 0.81 of SNPMs. Such low precision score achieved by RNaseCut is attributed to its strategy of reporting all possible base substitutions, which contain a large number of spurious SNP mutations. Putting them together, SNPMs still outperforms RNaseCut significantly in terms of the average F-measure score (0.79 vs 0.11).

Software	sensitivity (%)	Precision (%)	F-measure (%)
SNPMs	78.20(3.89)	81.00(3.87)	79.36(3.76)
RNaseCut	91.40(2.10)	5.96(0.03)	11.15(0.11)

Table 2.1: Performance evaluation on the simulated dataset where a randomly generated sample sequence contains five random SNP mutations. Note that the value in parentheses after each mean score represents the variance of the corresponding measure.

To assess the detection performance of SNPMs on a more challenging dataset, the second test dataset was generated in the same way as the first dataset except that 10 random SNP mutations rather than 5 are added into every instance of the sample sequence. It is not surprising that all the performance scores of both SNPMs and RNaseCut slightly dropped, as seen in Table 2.2. However, its average F-measure score is still much higher than that of RNaseCut (0.74 vs 0.11).

For a more comprehensive comparison, we generated another two test datasets. Compared to the previous two datasets, the only difference is that a real biological

Software	sensitivity (%)	Precision (%)	F-measure (%)
SNPMs	70.70(1.95)	77.18(2.02)	73.55(1.83)
RNaseCut	87.10(1.53)	5.83(0.02)	10.89(0.06)

Table 2.2: Performance evaluation on the simulated dataset where a randomly generated sample sequence contains ten random SNP mutations.

sequence (one fragment of gene Hemagglutinin in the influenza A H1N1 viral strain WSN/33; see the next section) was used as the reference sequence instead of a randomly generated one. The simulation results of these two datasets are summarized in Table 2.3. The performance behaviors of both SNPMs and RNaseCut are consistent with their performances in the experiments on the first two datasets. RNaseCut achieves slightly higher sensitivity scores than SNPMs in both datasets (0.93 vs 0.78 and 0.88 vs 0.69), but its precision scores are extremely low across all the experiments, which are always lower than 0.06. This special performance of RNaseCut should be attributed to the fact that it aims only to find all possible SNP mutations that are able to explain a differing mass peak in the measured mass spectra without any further attempt to identify which mutations are really true SNP mutations. As a result, RNaseCut has a much worse performance than SNPMs in terms of the F-measure score that evaluates the overall capability of accurately detecting SNP mutations.

#SNPs	Software	sensitivity (%)	Precision (%)	F-measure (%)
5	SNPMs	78.00(4.12)	81.38(4.23)	79.35(3.50)
	RNaseCut	93.40(1.68)	5.01(0.02)	9.48(0.06)
10	SNPMs	69.10(3.20)	76.17(2.61)	72.13(2.76)
	RNaseCut	88.00(1.36)	5.51(0.02)	10.33(0.07)

Table 2.3: Performance evaluation on the simulated datasets where either 5 or 10 SNPs are randomly added into a real sample sequence.

2.4.2 Results of biological data

Influenza A H1N1 virus was the most common cause of human influenza in recent years, especially responsible for the flu pandemic in 2009. In our experiments, the influenza A H1N1 viral strain WSN/33 was used and the comparative analysis was mainly focused on the hemagglutinin (HA) gene. The reference sequence that we used was CY009604, taken from NCBI dataset (<http://www.ncbi.nlm.nih.gov/genomes/FLU/Database/multiple.cgi>). Due to natural accumulated mutations, it is commonly expected that the WSN HA gene samples kept in the lab would have base differences from the reference sequence in the dataset.

Hemagglutinin (HA) is an elongated trimeric transmembrane glycoprotein, which can be found on the surface of the influenza viruses. It plays a central role in the viral infection process, because it is responsible for binding the virus to cells on the membranes and causing the fusion of host endosome membrane with the viral membrane. Thus, hemagglutinin is a primary target of neutralizing antibodies. The HA gene used in our study is about 1750 bp in length. In experiments performed in Dr. Tang Kai's lab, four pairs of PCR primers were designed to amplify four (overlapping) fragments from the HA gene sequence and then performed a separate comparative analysis for each fragment. Below we report the experimental results for the fragment which has incurred the largest number of base mutations (among the four amplified fragments).

We (Dr. Gao Xiang, from Dr. Tang Kai's lab) performed the base-specific cleavage and MALDI-TOF assay to the sample fragment under examination. The resulting four complimentary base-specific mass spectra were then input into our algorithm SNPMs for automatic SNP mutation detection. The reference sequence is the corresponding DNA sequence segment in gene CY009604 from position 410 to position 920, plus a

26-bp PCR primer added at the 5' end. Finally SNP-Ms predicted a total of 18 SNP mutations, and they are summarized in Table 2.4.

#	mutation	position	peaks (mass/SNR)	remarks
1	C / T	88	1907.30 / 302.51	true positive
2	A / G	462	2363.16 / 151.14	partially true positive
3	G / A	102	2942.15 / 147.00	true positive
4	T / C	292	1633.06 / 79.90 2925.21 / 45.85	true positive
5	C / T	12	1578.22 / 107.78	in T7 promoter region
6	C / A	235	2252.35 / 80.62 1978.06 / 24.11	false positive
7	A / C	147	3287.60 / 97.97	different base change
8	A / G	247	3271.63 / 81.71	partially true positive
9	G / T	197	2002.10 / 76.36	false positive
10	C / A	354	2832.13 / 76.77	partially true positive
11	C / T	142	2910.49 / 34.18	false positive
12	G / T	201	1328.03 / 34.26	false positive
13	T / G	269	1673.05 / 32.72	partially true positive
14	A / G	265	4219.86 / 28.23 2965.14 / 20.36	true positive
15	T / C	4	1601.21 / 30.21	in T7 promoter region
16	A / C	107	1985.22 / 30.25	false positive
17	A / T	77	1689.04 / 26.71	different base change
18	G / A	55	1649.01 / 21.09	false positive

Table 2.4: SNP detection results of SNP-Ms on a sample sequence from the influenza A H1N1 strain WSN/33. The # denotes the serial number of a detected SNP. SNR is short for signal-to-noise ratio.

To validate the above prediction, the influenza A H1N1 viral strain WSN/33 sample was sent for direct Sanger sequencing. The direct sequencing revealed ten SNP mutations that have occurred in the sample sequence. In the following, we consider these the mutations as ‘ground truth’ to evaluate the predictive performance of SNP-Ms.

As we can see in Table 2.4, SNP-Ms was able to correctly detect four of the ten true mutations. They are mutations 1, 3, 4 and 14 (i.e., ranked the 1st, 3rd, 4th and 14th) in

the output of SNPMS. All these mutations are supported by strong signal peaks in the measured mass spectra. For example, the detection of mutation 1 is due to the mass peak with relative intensity 76.07% and signal-to-noise ratio 302.53 (see Figure 2.3). The detection of mutation 4 is due to two mass peaks with signal-to-noise ratios of 79.90 and 45.85, respectively.

Figure 2.3: The mass peaks that support mutation 1, which is a true positive. Its supporting peak has very high relative intensity 76.07% and signal-to-noise ratio 302.51.

For another four true mutations, SNPMS can actually determine their correct base substitutions, but it only fails to unambiguously localize them. They are mutations 2, 8, 10 and 13 in the output of SNPMS. These mutations cannot be unambiguously localized because there exist multiple occurrences of a mutated base in the respective cleavage fragment but no signal peaks exist in the measured mass spectra that allow to pinpoint which occurrence has actually mutated. Take mutation 2 as an example, where a base substitution A/G is needed in the cleavage fragment AGAGGA (with respect to the cut base T) to explain a measured peak of mass 2363.16 Da. However, no matter which of

the three bases of A in the cleavage fragment is mutated, the same fragment mass value would be obtained. In this case, the true mutation actually occurs at the third base A, but SNPMs reported it instead at the first base A. Following this observation, we indicate such putative mutations as *partially true positives* in Table 2.4.

For the remaining two true mutations, SNPMs can still detect mutations at their (exact or nearby) positions but with different base changes. For example, there is a true mutation A/G at position 146, but SNPMs instead detected a mutation A/C at position 147 (i.e., mutation 7 in the output). In another case, there is a true mutation A/C at position 77, but SNPMs detected a mutation A/T at the same position (which is the mutation 17). In Table 2.4, we indicate such putative mutations as “different base changes”.

Among the eight putative mutations that are considered as false positives, two are located inside the T7 promoter regions. Indeed, most of the signal peaks that were used to support these false positive mutations are quite weak. For example, the detection of mutation 11 is due to the mass peak that has very low relative intensity 13.25% and signal-to-noise ratio 34.18 (see Figure 2.4). Mutation 6 is a noticeable exception, for which we have not found any good explanation.

As we can see from the above discussions, SNPMs has achieved the satisfactory SNP detection results for the influenza A H1N1 viral strain WSN/33 sample that we experimented. For comparison, we also ran the program RNaseCut on the same biological dataset. It reported 1377 potential mutations. Apparently, there are too many false positives to be useful for any downstream analysis.

Figure 2.4: The mass peaks that support mutation 11, which is a false positive. Its supporting peak has very low relative intensity 13.25% and signal-to-noise ratio 34.18.

2.5 Discussion and Improvement

The accurate detection of SNPs is crucial for the success of many downstream analyses such as the clinical diagnosis, virus identification, genetic mapping and association studies. One method that can offer high-throughput, reproducibility and low cost for SNP detection is based on nucleotide-specific enzymatic or chemical fragmentation followed by MALDI-TOF MS. To use this method for the automatic detection of SNP mutations, a challenging computational problem thus arises on how to integrate the information in the four complementary base-specific mass spectra in an efficient and effective way.

In this chapter, we presented an iterative and progressive algorithm. It works mainly by repeatedly identifying the SNP mutations that have potentially occurred in the sample sequence while progressively updating the reference sequence by correcting these

mutations. Unlike the algorithm in [4], it still allows to detect SNPs in close vicinity without increasing the sequence variation cost. We implemented the proposed algorithm in a program called SNPMS. Comparative evaluation has been carried out on both simulated and real biological datasets, and the results demonstrated the high ability of SNPMS to accurately detect SNP mutations. In particular, it achieved significantly higher precision scores than RNaseCut, the only alternative and publicly available program to date.

However, the algorithm in SNPMS has some limitations, which are discussed below. The first limitation is attributed to the base fixing strategy in the initialization stage of our algorithm. It labels the bases as being in the fixed status if the cleavage fragments that they reside can uniquely explain some measured mass peaks in $\mathcal{M}_\Sigma(s)$ with respect to the same cut bases. In most cases, this strategy is quite helpful to reduce the complexity of examining candidate SNP mutations because it excludes those positions where SNP mutations has little probability to occur. However, there does exist few chances that the locations of the true SNP mutations are fixed at the beginning and result in the increasing false negatives. To illustrate this special case, let the reference sequence be $s := AACGATCGAT$. We also suppose that the unknown sample sequence t contains one SNP mutation A/T at position 2 and one SNP mutation T/A at position 6. In addition, suppose that we are also given the measured mass spectra of t , which is $\mathcal{M}_\Sigma = \{m_A(TCG), m_C(GAA), m_C(GAT), m_G(ATC), m_G(AAC), m_T(CGAACGA)\}$. When we execute the initialization step of the algorithm in SNPMS, we have

$$\mathcal{M}_\Sigma(s) = \{m_A(TCG), m_G(ATC), m_G(AAC)\}.$$

After we fix the bases according to the measured mass peaks in $\mathcal{M}_\Sigma(s)$, we notice that

all bases of s are fixed except for the last two bases. At the beginning of the second iteration, $\mathcal{M}_\Sigma = \{m_C(GAA), m_C(GAT), m_T(CGAACGA)\}$. Then the base substitution T/A at the last position shall be identified to be the optimal potential SNP mutation because it can explain two mass peaks with respect to the cut base C. To this point, the algorithm terminates and falsely reports the potential SNP mutation T/A at the last position of s .

In order to improve the accuracy, two approaches might solve the first problem. One is a naive idea that cancels the initial labels of being in the fixed status and restart the algorithm after updating the measured mass spectra by deleting those mass peaks in $\mathcal{M}_\Sigma(s)$. That is, we free all the bases in s and resume the algorithm with the input mass spectra $\mathcal{M}_\Sigma = \{m_C(GAA), m_C(GAT), m_T(CGAACGA)\}$. Thereby, the base substitution T/A at position 6 shall be first identified to be the optimal potential SNP mutation since it is able to explain two mass peaks $m_C(GAA)$ and $m_C(GAT)$. Applying the updated reference sequence $s := AACGAACGAT$ and the updated mass spectra $\mathcal{M}_\Sigma = \{m_T(CGAACGA)\}$, the base substitution A/T at position 2 can be detected as another potential SNP mutation which explains the mass peak with respect to the cut base T. Therefore, we are able to detect all the true SNP mutations and obtain the sample sequence $t := ATCGAACGAT$. The other solution makes use of the mass value of each signal peak in the measured mass spectra. Furthermore, we define a new set $\mathcal{M}_\Sigma(s, p)$ containing all measured mass peaks in $\mathcal{M}_\Sigma(s)$ whose mass values exceed a threshold value p . This new set may be applied in the initialization stage by substituting $\mathcal{M}_\Sigma(s)$ in order to avoid fixing the bases that might be true SNP mutations. This approach is feasible because the signal peaks with larger mass values implies longer cleavage fragments that can hardly contain SNP mutations if they are explained by unique cleavage fragments in the reference sequence.

The second limitation is caused by the limit mass range that can be reliably detected by the current MALDI-TOF mass spectrometer. Theoretically, any SNP mutation can be precisely detected based on the ten signal changes resulting from the base substitution. However, the cleavage fragments with too few or too many bases cannot be reliably detected, which may result in spurious SNP mutations that achieve the same highest scores in our algorithm. The current version of SNPMs simply report the first-met permissible base substitution without any further examination. It is not surprising at all that the prediction strategy might produce more false positives. However, it is too difficult to solve this problem by improving the algorithm merely. Instead, we should expect the improvement of the MALDI-TOF MS technology.

The third limitation should be attributed to the property of input data. One of the most important step in the algorithm is to infer the base composition of a cleavage fragment according to the corresponding mass value measured by MALDI-TOF mass spectrometer. Although we are given the reference mass values of all four DNA nucleotides, some mass values can be obtained by more than one base composition. For instance, the base compositions of G_5T_8 and C_5G_8 have the same mass value of 4081.25 Da. Furthermore, even though the mass peak corresponds to a unique base composition, the sequence order of bases in the corresponding cleavage fragment may not be unique. As a result, a potential SNP mutation cannot be located with high confidence, especially when only one measured mass peak supports this SNP mutation. For example, if a SNP A/G is known occurring in the cleavage fragment *ACCGAG* with respect to the cut base T, and if the SNP is supported by the unique mass peak that has the mass value of $m_T(ACCGAG)$, we cannot tell which base A is substituted by G in the sample sequence. As mentioned previously, we cannot solve the difficulty by simply improving algorithms unless the precision of the techniques is improved.

The accuracy of SNP detection also highly depends on the extraction of the signals of interest. To make improvement, some preprocessing steps can be employed before performing our algorithm. In theory, an observed mass spectrum comprises baseline, true signal and noise. Accordingly, the preprocessing steps that aim at increasing the signal-to-noise ratio shall be beneficial to accurately detect SNPs. One of the common strategies is called deisotoping. During this step, the isotopic clusters of mass peaks are identified and replaced by monoisotopic peaks. The intensity of a monoisotopic peak is usually the sum of the intensities of all peaks within the corresponding isotopic cluster [57]. Another way to increase the signal-to-noise ratio is to identify and remove the baselines from the signal peaks. A number of popular algorithms for baseline correction have been reviewed in [23], where the authors also proposed a systematic procedure to evaluate and choose the optimal baseline correction algorithm for a specific analysis. These two types of preprocessing methods can both help improve the accuracy of detecting SNPs from mass spectrometry data.

To conclude the discussion in this chapter, detecting SNPs based on nucleotide-specific cleavage reactions and MALDI-TOF MS is high-throughput, low cost and reproducible, but it can be only used to analyze short genomes due to its limited mass range. So, this method is more suitable for studies in virus. Our study was motivated by the requirement of real biological experiments, hence we only considered the detection of SNPs. However, the algorithm we proposed can be also applied to the detection of base insertions and deletions. We would add such functions into the program SNPMS if it is necessary in the future.

Chapter 3

DNA Methylation Analysis

3.1 Introduction

DNA methylation is one of the most characteristic epigenetic modifications. In eukaryotes, it is always referred to cytosine methylation, denoted by mC5 or mC. Cytosine methylation plays an important role in many essential biological processes related to gene expression, as introduced in Section 1.2.5. We know that methylated Cs at different genomic locations influence different biological functions. Moreover, the primary genomic contexts where mCs are located vary among different types of eukaryotic organisms. For mammalian genomes, mCs are dominantly located in CpG or CG contexts while in plant genomes mCs can be found in both CG and non-CG contexts. Furthermore, it is also worth noting that the number of non-CG contexts is much larger than the number of CG contexts in the genome of an eukaryotic organism. Therefore, analyzing cytosine methylation of plant genomes becomes particularly more complex than analyzing cytosine methylation of mammal genomes, because not only CG contexts but also non-CG contexts have to be considered.

To determine the genome-wide methylation status, the current gold standard method is based on bisulfite sequencing (BS-Seq), which combines the bisulfite treatment and the next-generation sequencing (NGS) technologies. In brief, unmethylated Cs are converted into Us after bisulfite treatment and subsequently into Ts during the PCR amplification process. In contrast, methylated Cs keep unchanged (see Figure 3.1). Accordingly, the methylation status of each cytosine can be determined by simply comparing the DNA sequences before and after bisulfite treatment [30]. Together with the rapidly-advancing NGS techniques, we are able to analyze the cytosine methylation in genome-wide scale at the single base-pair resolution with a very low cost. At last, a large number of BS-Seq reads (or BS reads) are produced by either directional or non-directional protocols, depending on different PCR strategies, and give rise to the first computational problem – mapping these reads onto a reference genome.

3.1.1 Computational challenges of aligning BS-Seq data

Given a BS read P , a reference genome sequence G , and a positive integer k , the mapping problem is to find a substring of G that has at most k differences with P . In our study, we only consider mismatches since DNA reads generated from Illumina platform scarcely contain insertions or deletions.

While there are many excellent tools available for general sequence alignment tasks, we note that they are not satisfactory or convenient when applying to BS read sequences. It is not surprising when we notice the differences of BS reads from the general genomic sequencing reads. Firstly, after bisulfite treatment, C/T mappings are not symmetrical. In read sequences, a T can be matched to both C and T in the genomic sequence, because the base T might be resulted from bisulfite conversion of an

unmethylated C in the sample genome. However, it is not allowed to be the other way round, that is, the genome-to-read mapping T-to-C should be treated as a mismatch. On the other hand, since the two complementary DNA strands usually contain different distributions of methylated Cs, their corresponding converted strands after bisulfite treatment are no longer complementary with each other. It means that we have to consider both strands when aligning a BS read onto the reference genome.

Besides, depending on different amplification strategies, there exist two types of PCR protocols: directional protocol and non-directional protocol (see Figure 3.1). BS reads from non-directional protocol are generated from both the bisulfite converted DNA strands ($\pm$ FW strands) and their reverse complement stands ($\pm$ RC strands). In some specific non-directional protocol library, each BS read is labeled by a tag showing its direction, i.e., being generated from FW strands or RC strands. Comparatively, only those two converted strands ($\pm$ FW strands) are utilized to produce BS reads in directional protocol. Therefore, different mapping approaches are required to handle BS reads from different protocol libraries.

All of the above characterizations increase the challenges that we face when applying general purpose sequence alignment tools to BS reads alignment. It might be possible to map BS reads with a general purpose aligner if more differences are allowed in a read or if some preprocesses are performed on the input BS reads. The first idea cannot work since it increases too much cost that few aligners can afford. However, the other idea is more feasible and has been employed by many alignment tools specific to mapping BS reads, which is discussed in the next section.

Figure 3.1: Two primary stages in bisulfite sequencing: bisulfite treatment and next generation sequencing (NGS). BS reads can be produced from two types of PCR protocols: directional and non-directional protocols, whose main difference is the origins of reads. Non-directional protocols generate reads from $\pm$ FM and $\pm$ RC strands while directional protocols only utilize +FM and -FM strands.

3.1.2 BS-Seq alignment methods reviews

In recent few years, a number of alignment tools specific for mapping BS reads have been developed. In general, these tools can be divided into two broad categories: three-letter aligners and wild-card aligners [3]. The main differences between such two types of aligners are their strategies to deal with the asymmetrical C/T mappings. Three-letter aligners choose to convert all Cs in both reads and reference genome to Ts and then apply some general purpose alignment tools. In contrast, wild-card aligners do not have such C-to-T conversion. They either convert Cs only in reference genome to a wild-card letter Y, allowing it to match both a C and T in read sequences, or define a special scoring scheme for C/T mapping. Some specific aligners belonging to each of these two categories are reviewed in the remaining part of this section. Overall, it shall be noted that none of these aligners can perform best in all cases. Three-letter aligners usually select output alignments from the unique mappings reported by their built-in general purpose alignment tools. In this way, they can achieve higher accuracy. However, as they reduce the alphabet and sequencing complexity from four to three after converting Cs to Ts, a lot of true unique alignments might be missed. There is a high probability that a true unique alignment has multiple occurrences in reference genome after C-to-T conversion. In contrast, a wild-card aligner can be expected to achieve high genomic mapping coverage, but often introduce bias towards methylated Cs in methylation level estimation. Besides, most of these aligners cannot achieve a good balance between sensitivity and accuracy, especially when they are applied to mapping BS reads containing more mutations.

- **BS Seeker [7]**

BS Seeker is able to map BS reads from both directional and non-directional

protocols. To align BS reads from non-directional protocol, BS Seeker starts with C-to-T conversion on both FW reads (i.e. BS reads generated from $\pm$ FW strands) and the two complementary strands of the reference genome. It then utilizes Bowtie [33] to map the converted reads to the two converted strands of the reference genome using only three letters, A, G, and T. Similarly when mapping RC reads (BS reads generated from $\pm$ RC strands), G-to-A conversion is firstly performed on RC read sequences and both strands of the reference genome before Bowtie is applied. In total, Bowtie has to be executed twice if the tag of a read is available; otherwise, four runs of Bowtie are required because a read without a tag is treated as both a FW read and RC read. BS Seeker keeps only the unique alignments for the post-processing steps, where a read T (or A) is allowed to match a genomic C (or G), while a read C (or G) that aligns to a genomic T (or A) is considered as a mismatch. Aligning BS reads from directional protocol is much simpler because only FW reads exist and only two runs of Bowtie are needed to map the C-to-T converted reads onto the two C-to-T converted strands of the reference genome.

BS Seeker is designed for aligning short BS reads with at most three mismatches inside each read. To run BS Seeker, users need to decide the *effective read length* in alignment process that allows at most three mismatches. In other words, if we desire to use BS Seeker to align BS reads on the reference genome with at most three mismatches per read, the alignment can be completed in full read length. However, if we want to achieve the alignment of each BS read that contains more than three mismatches, we have to select a segment of the read starting from the 5' end and map this segment to the reference genome with at most three

mismatches, while the rest part of the read is discarded. Accordingly, BS Seeker cannot achieve satisfactory accuracy or efficiency when it is used to map longer BS reads (say, 100bp long reads) with more than three mismatches allowed in each alignment.

- **Bismark [29]**

Bismark uses a similar approach to that in BS Seeker to align bisulfite sequencing reads. For each BS read, Bismark generates two converted versions based on C-to-T conversion and G-to-A conversion. Correspondingly, it also converts all Cs in both complementary strands of the reference genome to Ts in order to align the C-to-T converted reads; and similarly it performs the G-to-A conversion on both reference strands for mapping the G-to-A converted reads. Therefore, Bismark can handle the alignment task for BS reads from either directional or non-directional protocol. Moreover, Bismark utilizes four parallel runs of Bowtie to align C-to-T converted reads to two C-to-T converted strands of the reference genome and align G-to-A converted reads to two G-to-A converted reference genomic strands.

However, different from BS Seeker, Bismark can complete aligning BS reads in full read length due to its improved strategy. Users can define the maximum number of mismatches (at most three) within the alignment of the prefix of a read, whose length is also self defined. Candidate mapping hits are found by running Bowtie that aligns the prefix to the reference genome according to the pre-setting mismatch number. These hits from four parallel processes are then extended to alignments of the whole read and the unique alignment containing the fewest mismatches is kept to infer the methylation states.

Bismark can map reads in less CPU time due to parallel implementation. Comparing with other three-letter BS read aligners, Bismark can align more reads to the reference genome with comparable high accuracy. Moreover, it can handle both single-end and paired-end BS read alignments. However, Bismark cannot avoid losing a lot of useful mapping information either, because it reduces the alignment complexity like all aligners in three-letter category.

- **BatMeth** [39]

Before mapping BS reads to reference genome, BatMeth uses the same C-to-T conversion on both Watson and Crick strands of the reference genome and then builds index for the two converted strands based on Burrows-Wheeler transform (BWT). For each BS read with eligible *complexity*, all Cs are converted to Ts first. The complexity of a read is the differential entropy estimated from the counts of the four DNA bases in a read. BatMeth discards the BS reads whose differential entropy is less than 0.25. BatMeth then applies a short read mapper, BatMis Aligner [66], to map the converted reads in four ways onto the converted genomes and create four lists to count the four alignment hits. Specifically, BatMeth aligns the C-to-T converted read and the C-to-T converted reverse complement of the read onto the two converted reference genome strands, respectively. To get rid of the spurious hits, BatMeth filter the lists whose hits exceed the cutoff. For those remaining hits, the numbers of mismatches are calculated regardless of the C/T mappings caused by bisulfite treatment. Finally, BatMeth reports the locations of the unique hits that contain the fewest mismatches.

BatMeth is the fastest method to align BS reads as far as we know. Moreover, it can achieve the highest accuracy if at most five mismatches are allowed in an

alignment hit. However, both our experiments and previous studies [31] show that it fails to handle the alignment tasks that a BS read is allowed to match the reference sequence with more than five mismatches. In such cases, BatMeth cannot even align more than half of the BS reads.

- **Last[15]**

Last is a representative wild card aligner in terms of using a special scoring matrix. It first constructs an index based on each of the reference genome strands according to an *adaptive seed pattern*, where a seed pattern indicates the alphabet used at each position. Last applies a seed pattern, 111111110, where 1 represents a matching with respect to the alphabet $\{a, c = t, g\}$ and 0 means free of matching. In other words, the first eight positions should be exact matches that allows C-to-T mapping, whereas the 9th position can have a mismatch. Moreover, a seed is eligible to be extended in the following steps if and only if it satisfies both of the two conditions: i) it is a prefix of the seed pattern or a repeat of the seed pattern; ii) its frequency in the reference genome is less than a threshold value. Then Last finds all possible alignments of a BS read by using the eligible adaptive seeds. For each read, two runs of Last are required with respect to mapping a read onto both strands of the reference genome. Finally, a mapping score of each possible alignment is computed, which measures the probability that the alignment can be true. In other words, the higher score does an alignment achieve, the more likely does the read have an occurrence at the position referred to by the alignment. The alignment achieving the highest score will be reported, if it is a unique one.

The most highlighted characteristics of Last is its capability of mapping a BS read without any limitation of mismatch number. Moreover, the scoring scheme

specific to C/T mappings allows Last to distinguish the alignments containing the same number of mismatches. Therefore, compared to the above three-letter aligners, Last can be expected to report more alignment information, as we have seen in practical experiments. However, Last usually costs much more running time than the other BS read aligners when it is used to map longer BS reads. We believe that the use of adaptive seeds is responsible for the higher cost, because the seeding speed depends heavily on the read length.

3.2 Backgrounds

3.2.1 Suffix array

Let an alphabet be $\Sigma = \{a, c, g, t\}$ since we only consider the DNA sequences. Specially, we consider the four characters a, c, g, t equivalent to A, C, G, T , respectively. Given a text or string $T \in \Sigma^*$, the suffix of T starting from position i , denoted as T_i , is defined to be a substring of T locating between position i and the last position of T (including both ends), where $0 \leq i \leq |T| - 1$. While the prefix of T ending at position i is the substring of T from the first position of T to the position i , denoted by $T_{0,i}$.

The suffix array of T , denoted by SA_T , is defined to be an array of integers providing the starting positions of suffixes of T that are sorted in lexicographical order. In other words, the i th entry of SA_T ($SA_T[i]$) is the starting position of the i th smallest suffix of T , where $0 \leq i \leq |T| - 1$.

Given a text T , the construction of its suffix array always starts from collecting all of its suffixes. Generally, we use a specific sentinel symbol $\$$ to mark the end of the text T . Moreover, the sentinel symbol is unique in T and is lexicographically smaller

than any symbol (or character) in the given alphabet Σ . After sorting these suffixes in an increasing order, the suffix array of T can be obtained, which comprises the starting positions of the sorted suffixes. Figure 3.2 uses an example to illustrate the construction of a suffix array.

$T = a a t g c a t t g \$$

Figure 3.2: Example of constructing a suffix array. Given T with length 10 bp, all of its suffixes are listed on the left side. The blue numbers indicate the starting positions of the suffixes. These suffixes are then sorted lexicographically. The suffix array of T is comprised by the starting positions (in blue) of the sorted suffixes. So $SA_T[0] = 9$, and $SA_T[6] = 3$.

The suffix array of a text T can be used to locate every occurrence of a pattern string P in T . If P is a substring of T , it must be a prefix of some suffix of T and hence corresponds to an entry of the suffix array of T . It is also easy to note that the suffixes beginning with the same prefix are grouped together after sorting. Thus, searching the occurrences of the pattern string P is equivalent to searching an interval from the suffix array of T , whose entries indicate the locations of the occurrences of P . Taking the example in Figure 3.2, a pattern string $P = at$ corresponds to an interval $[2, 3]$ of SA_T that implies the locations of P in T , position 1 and position 5. Formally, such interval

is named as a **suffix array interval** and denoted by $[l(P), u(P)]$, where

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

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

and $0 \leq k \leq |T| - 1$.

Applying suffix array intervals, it can be time-efficient to search and locate the occurrences of a pattern string. However, a large amount of memory space is required to store suffix array. Let $|T| = n$, then the DNA sequence T requires $O(n \log |\Sigma|)$ bits, while the suffix array of T occupies $O(n \log n)$ space in bits. Considering the human genome with $n \approx 3 \times 10^9$ and $|\Sigma| = 4$, suffix array needs ~ 16 times more space than the original genomic sequence. To avoid the high demand of memory space, a compressible data structure based on Burrows-Wheeler transform (BWT) is often used in place of suffix array.

3.2.2 Burrows-Wheeler transform

Burrows-Wheeler transform (BWT) is a reversible permutation of the text (or string) characters, which is a compressible string [5]. Making the BWT of a given string T requires two steps:

1. Generate all rotational permutations of T to form a conceptual matrix;
2. Sort the permutations according to lexicographical order, i.e., $\$ < a < c < g < t$.

Sorting the permutations is equivalent to sorting the rows of the original conceptual matrix, and hence a new matrix is obtained after these two steps. We use $\mathcal{M}$ to denote

the transformed matrix. Then the BW transform of string T , denoted by B_T , is the string comprising the characters in the last column of the transformed matrix $\mathcal{M}$. As shown in Figure 3.3, the BW transform of the DNA sequence $T = \text{aatgcattg}\$$ is $B_T = \text{g\$acgtttaa}$.

Figure 3.3: Illustration of BWT. After sorting all the rotations of T , the BWT of T is the last column in the blue frame. That is, $B_T = \text{g\$acgtttaa}$. The green numbers in front of the first column of the transformed matrix are the indexes of the rows.

As we mentioned at the beginning, BWT is a reversible transformation. Let F be the first column of matrix $\mathcal{M}$ and L be the last column. Since each row of $\mathcal{M}$ is a cyclic shift of T , $L[i]$ always proceeds $F[i]$ in the original text T for all $0 \leq i \leq |T| - 1$, except for the i th row ending with the sentinel symbol $\$$. Moreover, if $L[i]$ is the r th occurrence of a character x in the last column L and $F[j]$ is the r th occurrence of x in the first column F , then $L[j]$ immediately proceeds $L[i]$ in the original text. In the previous example (Figure 3.3), $L[4]$ and $F[6]$ are the second occurrences of g in the last and first columns, respectively. When we compare row 4 and row 6 with the original string T , we note that these two g 's actually correspond to the same position in T . Accordingly, $L[6] = t$ locates one position before $L[4]$ in T due to the equivalence of $L[4]$ and $F[6]$. The above observation is known as the **last-to-first mapping** (or **LF mapping**) and is proved in [5]. LF mapping gives rise to the algorithm (see Algorithm 2) that reserves

the BW transformed string back to the original text.

Algorithm 2 *ReverseBWT*

Input: A BWT string B , $|B| = n$.

Output: The original string T .

```

1: Construct  $F$  by sorting characters in  $B$  lexicographically
2:  $T[n - 1] \leftarrow \$$ 
3:  $x \leftarrow B[0]$ 
4:  $k \leftarrow n - 2$ 
5: while  $x \neq \$$  do
6:    $T[k] \leftarrow x$ 
7:    $r_i \leftarrow$  the rank of  $x$  in  $B$ 
8:    $j \leftarrow$  position of the  $r_i$ th  $x$  in  $F$ 
9:    $x \leftarrow B[j]$ 
10:   $k \leftarrow k - 1$ 
11: end while
12: return  $T$ 

```

3.2.3 FM index

It isn't hard to observe that sorting the cyclic rotations of T is equivalent to sorting the suffixes of T , as shown in Figure 3.4 . This strong relation between the matrix $\mathcal{M}$ and suffix array can be summarized by Equation 3.1. In other words, the i th character in B_T is exactly the character located at one position before the i th smallest suffix in T , unless the i th smallest suffix is T itself. It provides a way to construct BWT of a string from the corresponding suffix array.

$$B_T[i] = \begin{cases} T[SA_T[i] - 1], & \text{if } SA_T[i] > 0, \\ \$, & \text{if } SA_T[i] = 0. \end{cases} \quad (3.1)$$

Constructing BWT from suffix array is simple and direct, however, it is not obvious the other way round. In order to achieve the suffix array from BWT, the full-text minute-

$T = \text{a a t g c a t t g \$}$

		i	SA _T [i]		B _T [i]
9	\$ a a t g c a t t g	0	9	\$ a a t g c a t t	g
8	g \$ a a t g c a t t	1	0	a a t g c a t t g	\$
7	t g \$ a a t g c a t	2	1	a t g c a t t g \$	a
6	t t g \$ a a t g c a	3	5	a t t g \$ a a t g	c
5	a t t g \$ a a t g c	4	4	c a t t g \$ a a t	g
4	c a t t g \$ a a t g	5	8	g \$ a a t g c a t	t
3	g c a t t g \$ a a t	6	3	g c a t t g \$ a a	t
2	t g c a t t g \$ a a	7	7	t g \$ a a t g c a	t
1	a t g c a t t g \$ a	8	2	t g c a t t g \$ a	a
0	a a t g c a t t g \$	9	6	t t g \$ a a t g c	a

Figure 3.4: Relation between suffix array and BWT. On the left side, each row is a permutation of the characters of T and begins with a suffix of T . After sorting these suffixes together with the remaining parts of their permutations, the obtained matrix is exactly the transformed matrix $\mathcal{M}$. From $\mathcal{M}$, both the suffix array (in blue) and the BWT (in rose) of T can be implied.

space (FM) index can be used, which is created according to the LF mapping [13]. The LF mapping is actually a one-to-one map from a character in the last column $L[i]$ to a character in the first column $F[j]$. In general, this mapping is represented as $j = LF(i)$. From Section 3.2.2, the occurrence rank of a character is essential to figure out the format of LF mapping. In fact, LF mapping consists of two functions, denoted by $C(x)$ and $Occ(x, k)$ [13]. $C(x)$ is the number of characters in text $L = B_T$ that are lexicographically smaller than x . For example, $B_T = g\$acgtttaa$ and $C(c) = 4$. $Occ(x, k)$ represents the number of occurrences of x in B_T before position k (note that the occurrence at position k is not counted). Specially, $Occ(x, 0) = 0$. For the same example above, $Occ(c, 3) = 0$ and $Occ(c, 4) = 1$. The LF mapping is therefore defined as, $LF(i) = C(L[i]) + Occ(L[i], i)$. By using this mapping recursively, the suffix array can be obtained from the BWT structure (see Algorithm 3).

The FM index is the combination of BWT string B_T , $C(x)$ and $Occ(x, k)$, where

Algorithm 3 *BWT2SA*

Input: A BWT string B , $|B| = n$, an integer i , $0 \leq i < n$.

Output: The i th entry of suffix array.

```
1: if  $i == 0$  then
2:   return  $n - 1$ 
3: else
4:   if  $B[i] == \$$  then
5:     return 0
6:   end if
7: end if
8:  $j \leftarrow i, k \leftarrow 0$ 
9:  $x \leftarrow B[i]$ 
10: while  $x \neq \$$  do
11:    $j \leftarrow C(x) + Occ(x, j)$ 
12:    $x \leftarrow B[j]$ 
13:    $k \leftarrow k + 1$ 
14: end while
15: return  $k$ 
```

$x \in \Sigma \cup \{\$ \}$ and $0 \leq k \leq |B_T| - 1$. Note that $Occ()$ requires $O(4 \times |B_T| \log |B_T|)$ memory space to store the number of occurrences of each character at each position. In practice, a genome sequence is very long, which results in a large demand of memory to store all values of $Occ()$. To reduce the memory requirement, we skip some positions and only record $Occ(x, k)$ of each character if k is a multiple of 32. When the values of $Occ()$ at the skipped positions are needed, we can simply calculate them by counting.

The FM index is usually applied instead of suffix array due to its compressible structure. To locate a pattern sequence within a reference string, it is equivalent to find a suffix array interval, which can be completed by applying the FM index. Given the BWT of text T , B_T , a character x , and the suffix array interval of a pattern sequence P

$[l(P), u(P)]$, then the suffix array interval of xP is

$$\begin{aligned} l(xP) &= C(x) + Occ(x, l(P)), \\ u(xP) &= C(x) + Occ(x, u(P) + 1). \end{aligned} \tag{3.2}$$

We can easily prove that xP is a substring of T if and only if $l(xP) \leq u(xP)$. Therefore, to find the number of occurrences of a pattern sequence, we simply need to recursively calculate the suffix array intervals with Equation 3.2 starting from the last base, which is also named as *backward search*. In order to locate a pattern sequence, we can retrieve its original positions based on the suffix array. In practice, to improve the retrieving efficiency, a part of suffix array entries are usually stored in memory and the others are computed using an algorithm similar to Algorithm 3. We select the entries that are multiples of 32 to be stored in our work, together with the corresponding positions in BWT.

3.2.4 Bi-directional BWT and FMD index

Backward search by using the FM index can deal with exact matching very well. Unfortunately, it is not sufficient to find exact matchings merely, because sequencing errors and genetic variations may result in alignments containing mismatches. In real bioinformatic studies, we have to consider approximate or inexact alignments. The FM index can be used to find the approximate alignments for DNA reads, but it is not convenient.

One of the most famous applications is Bowtie, which utilizes a backtracking algorithm (see Figure 3.5) to align short reads containing mismatches [33]. Note that Bowtie has to check every possible substitution at every base between the last base of a read and the base where exact matching fails. Although several approaches are de-

veloped to improve efficiency, Bowtie cannot avoid excessive backtracking when more mismatches are allowed in an alignment. Another well-known aligner using FM index to find approximate alignments is called BWA, which searches all possible approximate matches for short reads by using a recursive algorithm [36]. BWA runs faster than Bowtie when aligning DNA reads with shorter length ($<80\text{bp}$), but spends much more time when the read length increases. Considering the current Illumina platform, the reads are always longer than 80 bp and hence the alignments should tolerate more mismatches per read. Furthermore, the reads should be mapped to both two complementary strands of the reference genome in order to achieve comprehensive alignment results. Therefore, neither Bowtie nor BWA is efficient enough to handle these issues.

To improve the efficiency and flexibility of finding approximate matches, a BWT based index structure that enables either backward or forward search becomes favorable. For this purpose, bi-directional BWT was firstly introduced in [32] to allow matching to be conducted in both forward and backward directions. To map reads onto the reference genome T with mismatches allowed, bi-directional BWT uses two suffix array intervals for T and the reverse of T , respectively. Further to the work of [32], a so-called FMD-index was proposed in [35]. The FMD index is a single index constructed for both forward and reverse complementary strands of DNA sequences. In our work, we focus on mapping the bisulfite treated reads to reference genome, thus the construction process of FMD index is introduced in the scenario of BS-Seq alignment in the remaining text of this section.

Given a reference genome sequence T , we firstly create four strands: the C-to-T converted Watson strand (T_+^C), the C-to-T converted Crick strand (T_-^C), the reverse complementary strand of T_+^C ($\overline{T_+^C}$) and the reverse complementary strand of T_-^C ($\overline{T_-^C}$). From now on, we use $\overline{X}$ to represent the reverse complement of string X . Subsequently,

Figure 3.5: Inexact alignment with Bowtie. The pattern string $ggtat$ contains a g/a mismatch at position 3. Bowtie begins with exact matching and fails to reach the left end. It then resumes the exact matching process by substituting the base that corresponds to the last available interval. Repeating such backtracking search, Bowtie finally finds the valid alignment that contains one mismatch. The figure comes from [33].

a *bi-directional collection* of T is constructed by concatenating the above four strands into one string, i.e.,

$$\tilde{T} = T_+^C \circ \$ \circ \overline{T_+^C} \circ \$ \circ T_-^C \circ \$ \circ \overline{T_-^C} \circ \$,$$

where $\circ$ denotes the string concatenation. Note that four sentinel symbols $\$$ exist in a bi-directional collection. We specially define their orders according to their locations in $\tilde{T}$. So if $\tilde{T}[i] = \tilde{T}[j] = \$$, $\tilde{T}[i] < \tilde{T}[j]$ if and only if $i < j$. Furthermore, the reverse

complement of \$ is itself.

Once a bi-directional collection $\tilde{T}$ is built, we take the BW transformation of $\tilde{T}$. Accordingly, given a read P , a corresponding suffix array interval can be calculated. The FMD index uses a *bi-interval* to accomplish the search and matching in two directions. Specifically, a bi-interval consists of three components and denoted by $[l(P), l(\bar{P}), s(P)]$, where $l(P)$ is the left endpoint of the suffix array interval of P and $s(P)$ is the length of this suffix array interval. Let us start from two lemmas to show how to calculate the bi-intervals of xP and Px ($x \in \Sigma \cup \{\$ \}$) when the bi-interval of P is given. These two lemmas are from [32, 35].

Lemma 1 *Given a bi-directional collection $\tilde{T}$, for any pattern string P , the size of its suffix array interval is equal to that of $\bar{P}$. That is, $s(P) = s(\bar{P})$.*

It is easy to observe that the occurrences of a pattern and its reverse complement within the four strands must be the same. From the BW transformed matrix, we can also observe that

Lemma 2 $\sum_x s(Px) = s(P) = \sum_x s(xP)$, $x \in \Sigma \cup \{\$ \}$.

When the bi-interval of P is known as $[l(P), l(\bar{P}), s(P)]$, $l(xP)$ can be obtained directly from backward search as the normal FM index. We note that $l(Px) \leq l(P)$ and $l(Px) < l(Py)$ for $x < y$, $x, y \in \Sigma \cup \{\$ \}$. So we just need to count the number of characters that are located after P and are smaller than x . Hence, $l(Px)$ can be calculated by adding the count to $l(P)$. By applying the above two lemmas, computing $l(\bar{xP})$ is equivalent to counting within the interval both the sentinel symbols and the characters in the BWT string that are larger than x . Hence, the bi-interval of xP can be obtained. As for computing the bi-interval of Px , we simply reverse the first two components of

the bi-interval of $\overline{Px} = \overline{x} \circ \overline{P}$. These computing processes are summarized in Algorithm 4 and Algorithm 5, which follows the algorithms used in [35].

Algorithm 4 *FMDBackwardSearch*($x, [l, k, s]$)

Input: BWT string of $\tilde{T}$, B , bi-interval of P , $[l, k, s]$, a character x

Output: The bi-interval of xP

```

1:  $l_x \leftarrow C(x) + Occ(x, l)$ 
2:  $s_x \leftarrow Occ(x, l + s) - Occ(x, l)$ 
3:  $k_x \leftarrow 0$ 
4: for  $i \leftarrow l$  to  $l + s - 1$  do
5:    $b \leftarrow B[i]$ 
6:   if  $b == \$$  then
7:      $k_x \leftarrow k_x + 1$ 
8:   else
9:     if  $b > x$  then
10:       $k_x \leftarrow k_x + 1$ 
11:    end if
12:  end if
13: end for
14: return  $[l_x, k_x, s_x]$ 

```

As we mentioned at the beginning, we select the FMD index in order to find approximate matches for a BS read. In our study, we applied the classical seed-and-extension strategy to approach the sequence alignment problem. As a critical part of this strategy, how to choose seeds has attracted a lot of research attentions. In the next section, we discussed four ways to choose seeds and select approximate seeds for our algorithm. Specially, we developed bi-directional tests based on FMD index to select an eligible approximate seed.

Algorithm 5 *FMDForwardSearch*($x, [l, k, s]$)

Input: BWT string of $\tilde{T}$, B , bi-interval of P , $[l, k, s]$, a character x

Output: The bi-interval of Px

```

1:  $[k_x, l_x, s_x] \leftarrow FMDBackwardSearch(\overline{x}, [k, l, s])$ 
2: return  $[l_x, k_x, s_x]$ 

```

3.2.5 Seeds for alignment

Given a read P with l bp, a positive integer k , and the reference genome G , the general sequence alignment problem is to find out all the substrings of G that can match P with at most k mismatches. We do not consider insertions and deletions in the alignment problem, because we only deal with aligning reads produced by Illumina platform, where the sequencing errors present in reads are mostly substitutions.

One classical approach to approximately aligning a read is known as seed-and-extension. To apply this strategy, the appropriate selection of seeding methods is essential to identify the correct alignments after the extension steps. Below, we discuss four popular types of seeds: exact seeds, spaced seeds, adaptive seeds and approximate seeds.

Exact seeds

Exact seeding is the most widely used approach. To locate the approximate matches of P in G , we equally partition P into $k + 1$ non-overlapped segments called seeds. By the pigeonhole principle, there will be at most k seeds containing one or more mismatches, if the read P is aligned to a genomic location with at most k mismatches. In other words, if P has an alignment within k mismatches, there must exist at least one seed being matched exactly. Such a matching is thus called a hit of this exact seed.

Exact seeds are very useful when the alignments are permitted to contain few mismatches. For clarity, we define $r = k/l$ and call it the *mutation rate*. In terms of mutation rate, exact seeds enable to improve the efficiency of read mapping when $r < 3\%$ in general cases. When r increases, the length of each seed decreases. Accordingly, it results in a dramatic increase of the number of subsequent hit extension steps and the

total mapping time as well.

Spaced seeds

Spaced seeds are developed and used in DNA comparison in order to increase the sensitivity and decrease the computational time. Spaced seeds allow fixed “don’t care” positions and are usually modeled by 0-1 strings, where 1 represents a matching and 0 means free of matching. The concept of spaced seeds was formally proposed by B. Ma et al. [42]. They also provided theoretical support for the benefits of alignment using spaced seeds.

We utilize an example to illustrate the efficiency of mapping with spaced seeds. Suppose that a DNA sequence r has 100 bases with 70% similarity to the sample sequence. If a consecutive exact seed has a length of 12 bp, according to a lemma in [42], the expected number of hits shall be 1.23. If a length-31 spaced seed with 19 “don’t care” positions is considered, the corresponding expected number of hits is 0.97. Hence, with the same number of exact matchings, the spaced seeds need less hit extension steps than the consecutive exact seeds in theory. Experimental results in the previous researches [26, 42, 27] also showed that spaced seeds can help reduce the computing time and achieve higher sensitivity than consecutive exact seeds.

The spaced seeds avoid a decrease in sensitivity when seed length increases, which exact seeds suffer from. However, the selection of an optimal spaced seed pattern for a practical alignment task is much more difficult, especially when little knowledge of a dataset is given.

Adaptive seeds

Adaptive seeding is an alternative solution to approach approximate alignment problems. Adaptive seeds are defined based on a frequency threshold f instead of the maximum number of mismatches allowed in an alignment. Given a read P , a segment of P is an adaptive seed if it has at most f matches in the reference genome G [27]. Adaptive seeds are proposed according to the observation that sequencing errors tend less likely to occur at the bases near the 5' end.

To find adaptive seeds, we need to find the shortest seed starting at every position of the read P such that the seed has at most f hits in G . The seed finding strategy ensures that the number of subsequent extending steps is upper bounded no matter how many mismatches are allowed in the alignment problems. It implies that the running time of read mapping with adaptive seeds only largely depends on the read length and the threshold f . It also tells the main limitation of adaptive seeds that this approach may spend more time when aligning longer reads. Our experiment results also show that the running time of Last that adopts adaptive seeds is longer than most of the other softwares when mapping 100bp reads with at most three mismatches.

Approximate seeds

Approximate seeds were proposed and defined formally in [62]. Generally, a given read P is firstly partitioned into $m \leq k + 1$ non-overlapped equal-length seeds. From the pigeonhole principle, at least one seed exists such that it can be matched to a substring of G with at most $\lfloor k/m \rfloor$ mismatches. The seeds with at most $\lfloor k/m \rfloor$ mismatches allowed in their hits are thus called *approximate seeds*. In our study, we apply approximate seeds to detect the candidate genomic locations of a BS read. Specially, we partition a

read into $m = \lfloor k/2 \rfloor + 1$ seeds and try to find a seed being matched exactly or with one mismatch. We call the corresponding hits as the 1-approximation hits.

Like spaced seeds, approximate seeds can increase the seed length and sensitivity simultaneously. Compared with the selection of spaced seeds, it is much easier to select approximate seeds because we are not bothered by seed patterns any more.

Applying approximate seeds can achieve significantly higher mapping efficiency than applying exact seeds, because the increasing seed length gives rise to a dramatic decrease of seed hits, especially in the alignment problems where k is large relative to l . In other words, the larger the mutation rate r , the higher the mapping specificity that approximate seeds improve over exact seeds. This property is particularly useful for fast alignment of BS reads, where the alphabet and the sequence complexity are reduced. Our experiments on the real BS read data of *Arabidopsis thaliana* show that up to 19 times less hits were generated with approximate seeds than with exact seeds for subsequent extensions when at most five mismatches are allowed in an alignment (see Table 3.1).

Data accession	Type of Seeds	No. of Seed hits (M)	H_e/H_a
ERR046546	exact	37001.22	19.99
	approximate	1851.40	
ERR046551	exact	12575.78	18.94
	approximate	664.01	
SRR771521	exact	26239.92	43.36
	approximate	605.10	

Table 3.1: Seed hits comparison when $k = 5$. Our experiment used the whole data in dataset ERR046546 (~ 25 M) and the first 10 million reads in each dataset of ERR046551 and SRR771521. The last column shows the times of the seed hits produced by using exact seeds over those by using approximate seeds, where H_e and H_a represent the number of seed hits generated by using exact seeds and approximate seeds, respectively.

Compared with adaptive seeds, aligning reads with approximate seeds requires less

running time when not too many mismatches (< 5) are permitted in an alignment. In addition, mapping with approximate seeds can be expected to achieve more correct alignments than mapping with adaptive seeds. To illustrate the higher accuracy of aligning with approximate seeds compared to aligning with the adaptive seeds, we use an example where the reference genome sequence is $G = \text{acatggcgtaccgcagtgcatgaca}$ and the read sequence is $P = \text{atctca}$. It can be observe that P has a match in G at position 9 containing two mismatches, c/t and g/t at position 1 and 3 in P , respectively. Accordingly, we test the mapping correctness with these two types of seeds by allowing at most two mismatches in an alignment. We first align P to G by using adaptive seeds with $f = 2$. Starting from the first base of P , we check the frequency of each substring and we have at with a frequency of 2. After checking all the remaining positions of P , no eligible adaptive seeds can be obtained any more, because those substrings have either zero occurrence in G or more than $f = 2$ occurrences. Therefore, only one adaptive seed at is available for the subsequent extending. However, the two hits of seed at in G fails to give rise to the correct alignment, both of which map the read P at the cost of 3 mismatches. Next, we apply approximate seeds to mapping P . We partition P into two seeds, i.e., atc and tca , and find the hits containing at most one mismatch for each seed. Seed atc has three 1-approximation hits at positions 2, 9, and 19 in G , respectively. We extend every hit of seed atc and find the hit at position 9 giving rise to an alignment having two mismatches. On the other hand, seed tca also report three 1-approximation hits at positions 12, 17, and 23 in G , among which the hit at position 12 implies the same alignment at position 9 with two mismatches. It is obvious that the correct alignment is found by using approximate seeds. Moreover, these advantages of applying approximate seeds are also shown by our experimental results presented in the later Results section.

Figure 3.6: Schematic illustration of *TAMEBS*. Starting from the two complementary strands of a reference genome, we first convert all C's to T's and then further generate their respective reverse complementary strands. The four resulting strands are concatenated into a bi-directional collection, for which we build a bi-directional FM index. After this index-building process, for each BS read, we convert all C's to T's and then find its candidate mapping hits by using an approximate seeding strategy. As shown above, applying seed TTG gives rise to hits at position 2 and 10 on the forward converted strand and position 2 on the reverse-complementary converted strand. Another eligible hit is given by seeding with TTT. Afterwards, hit extensions are performed between the original reference genome sequence and the read sequence without any C-to-T conversions. The full alignments are scored with a likelihood-ratio scoring matrix (Table 3.2), and those achieving the highest score are finally reported.

3.3 Our Method: TAMeBS

We developed a software tool called TAMeBS, implemented in C++ language, to align BS reads by making use of bi-directional FM index (FMD index), approximate seeds and the likelihood-ratio scoring matrix. We further extended TAMeBS to estimate cytosine methylation distributions from the aligned BS reads. Our work can be found in [65] and the software can be freely downloaded at <http://sourceforge.net/projects/tamebs/>.

There are three components implemented in TAMeBS: bi-directional FM index building, seed-and-extension read mapping, and methylation calling. Figure 3.6 illustrates the schematics of our method. In the index building component, TAMeBS constructs the suffix array as well as the BWT of a bi-directional collection $\tilde{T}$ using a modified version of SA-IS algorithm [51]. In order to trade off between the memory space and running time, only a part of suffix array is stored via sampling. In our implementation, one entry of the suffix array would be retained every 32 entries. Other entry elements will be computed using the BWT whenever needed.

In the read mapping component, we align BS reads to the reference genome one after another. For each read, we first convert all Cs to Ts if it contains any. Then we find all the 1-approximation hits of its seeds by using the bi-directional tests as described in Section 3.3.1. For each hit found, we extend it to a full alignment of the read up to k mismatches. In order to improve sensitivity as well as accuracy, the likelihood-ratio matrix introduced in Section 3.3.2 is used to score alignments where no C-to-T conversion is made in both read and genome sequences. For the sake of faster alignment, if a seed has too many hits (exceeding a preset threshold B), all these hits will be discarded and thus excluded from further extension. The threshold value of B

should depend on the seed length. The longer a seed, the smaller the threshold value of B . In TAMeBS, B is empirically set to be $\lceil 2000/\text{length}(\text{seed}) \rceil$ (note that the seed length varies with the mutation rate r). We say an alignment is *best* if it contains at most k mismatches while achieving the highest mapping score. And an alignment is called *uniquely best* if it is the only best alignment. TAMeBS can report uniquely best, any best and all best alignments of a read depending on users' choices. By default, the uniquely best alignments are reported.

In the methylation calling component, the methylation status of each cytosine and the overall distributions are inferred from the read alignment results. The details are discussed in Section 3.3.3.

3.3.1 Finding approximate seeds with bi-directional index

We denote the seeds of P as $P^{(i)}$, $i = 0, \dots, m - 1$, ordered according to their starting positions in P , where $m = \lfloor k/2 \rfloor + 1$. For each seed $P^{(i)}$, we conduct bi-directional tests to find its hits (i.e., 1-approximation matchings) in a reference genome G . The forward and backward tests are described in Algorithm 6 and Algorithm 7, respectively. They are developed based on the observation that for any hit alignment of $P^{(i)}$, either the first half $P^{(i)}[0, \lfloor l_i/2 \rfloor]$ or the second half $P^{(i)}[\lfloor l_i/2 \rfloor + 1, l_i - 1]$ of $P^{(i)}$ shall be exactly matched, where l_i is the length of $P^{(i)}$. They also work in quite a similar way. For instance, the forward test starts by searching for exact matchings in the forward direction, from which we would obtain one of the following three possible outcomes:

1. Exact matchings of the whole seed $P^{(i)}$ are returned.
2. The exact matching process halts before the middle position $\lfloor l_i/2 \rfloor$ is reached.

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

The outcome 1 implies that we have found all the hits of $P^{(i)}$ which are the exact matchings. When it occurs, we proceed to the hit extension step directly without bothering to search for other hits that involve mismatches. In this way it will significantly speed up the read alignment process. However, the mapping sensitivity might be sacrificed, but not much, because there is little chance that the seed $P^{(i)}$ aligns to the true genomic location with exactly one mismatch and, at the same time, to some other genomic locations without any mismatch, especially when the seed is longer (say, >30 bp). If the outcome 2 occurs, it implies that at least one mismatch exist within the current half. So we stop the current test in the forward direction and then proceed to the test in the backward direction. In case of the outcome 3, we resume the exact matching process at the middle position in order to find the hits of $P^{(i)}$ with one mismatch occurring in the second half. The exact matching process is then recursively branched to accommodate a mismatch at each subsequent position, described in Algorithm 8. A branching process is terminated once two mismatches are met. For each branching process finally reaching the right end, we would obtain a bi-interval that gives rise to a set of hits of $P^{(i)}$.

Note that the two functions *ForwardExactSearch*(P) and *BackwardExactSearch*(P) in Algorithms 6 to 8 are used to find exact matchings of a pattern string P with bi-directional FM index. *ForwardExactSearch*(P) starts from the first base of P and recursively apply Algorithm 5 to calculate the bi-intervals in forward direction. The search will stop and report the current bi-interval together with the stopping position if it reaches the rightmost base of P or an invalid bi-interval of size zero is obtained.

BackwardExactSearch(P) works in a similar way to *ForwardExactSearch(P)*, except that it searches from the last base of P in backward direction by recursively using Algorithm 4. These two functions are described in Algorithm 9 and 10.

Algorithm 6 *ForwardApproximateSearch*

Input: A seed $P^{(i)}$

Output: A set $\mathcal{S}$ of bi-intervals of seed hits

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2:  $b \leftarrow \lfloor l_i/2 \rfloor$ 
3:  $[h, I] \leftarrow \text{ForwardExactSearch}(P^{(i)})$ 
4: if  $h == l_i$  then
5:    $\mathcal{S} \leftarrow \mathcal{S} \cup \{I\}$ 
6: else
7:   if  $h > b$  then
8:      $\mathcal{S} \leftarrow \mathcal{S} \cup \text{MismatChecking}(b, l_i - 1)$ 
9:   end if
10: end if
11: return  $\mathcal{S}$ 

```

Algorithm 7 *BackwardApproximateSearch*

Input: A seed $P^{(i)}$

Output: A set $\mathcal{S}$ of bi-intervals of seed hits

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2:  $b \leftarrow \lfloor l_i/2 \rfloor$ 
3:  $[h, I] \leftarrow \text{BackwardExactSearch}(P^{(i)})$ 
4: if  $h == -1$  then
5:    $\mathcal{S} \leftarrow \mathcal{S} \cup \{I\}$ 
6: else
7:   if  $h < b$  then
8:      $\mathcal{S} \leftarrow \mathcal{S} \cup \text{MismatChecking}(b, 0)$ 
9:   end if
10: end if
11: return  $\mathcal{S}$ 

```

Our method for finding the hits of approximate seeds as described above is different from the one implemented in Masai [62]. Unlike ours, the method in Masai essentially conducts approximate matching of seeds only in one direction. With bi-directional

tests we require an exact matching of either the first half or the second half of the seed, which provides highly efficient filtering of spurious hits and thus reduces the total seeding time, as already argued in [33].

3.3.2 Extending seed hits

In this step, we aim to extend the seed hits into full alignments of reads. Different from the previous seeding process, our hit extension is performed between the original read sequences and the reference genome sequence without any C-to-T conversion (see Figure 3.6). It would enable us to penalize a mapping of a genomic T against a read C which shall be considered as a mismatch. To further detect sensitive alignments, we use log likelihood ratios to score alignments in the same way as many traditional sequence alignment methods have done. Following the process used in the study of Last [15], the classical scoring matrixes is defined by

$$S_{xy} = T \ln \left(\frac{A_{xy}}{P_x G_y} \right),$$

where A_{xy} is the probability that base x of read sequences is aligned to base y of the reference genome sequence in a true alignment; P_x is the probability of an occurrence of base x in read sequences and G_y is the probability of base y occurring in the reference genome. T is an arbitrary scale factor.

In the case of aligning BS reads, we note that a fraction of read Cs are converted to Ts after bisulfite treatment. Let F denote the conversion fraction and suppose that all read Cs have the same chance to be converted to Ts (regardless of genomic contexts).

Algorithm 8 *MismatchChecking*

Input: Integers b and e , the bi-interval of $P^{(i)}[0, b - 1]$ if $b < e$, otherwise that of $P^{(i)}[b + 1, l_i]$.

Output: A set of bi-intervals of $P^{(i)}[0, l_i]$, S , that imply alignments with one mismatch between position b and e .

```
1:  $i \leftarrow b$ 
2: if  $b \geq e$  then
3:   for  $i \geq e$  do
4:      $I \leftarrow \text{BackwardExactSearch}(P^{(i)}[i + 1, l_i])$ 
5:     for  $x \in \Sigma \ \& \ x \neq P^{(i)}[i]$  do
6:        $I_i \leftarrow \text{FMDBackwardSearch}(P^{(i)}[i + 1, l_i], x)$ 
7:       if  $I_i$  has positive size then
8:          $[h, I_i] \leftarrow \text{BackwardExactSearch}(P^{(i)})$ 
9:         if  $h < 0$  then
10:           $S \leftarrow S \cup \{I_i\}$ 
11:        end if
12:      end if
13:    end for
14:  end for
15: else
16:   for  $i < e$  do
17:      $I \leftarrow \text{ForwardExactSearch}(P^{(i)}[0, i - 1])$ 
18:     for  $x \in \Sigma \ \& \ x \neq P^{(i)}[i]$  do
19:        $I_i \leftarrow \text{FMDForwardSearch}(P^{(i)}[0, i - 1], x)$ 
20:       if  $I_i$  has positive size then
21:          $[h, I_i] \leftarrow \text{ForwardExactSearch}(P^{(i)})$ 
22:         if  $h \geq l_i$  then
23:           $S \leftarrow S \cup \{I_i\}$ 
24:        end if
25:      end if
26:    end for
27:  end for
28: end if
29: return  $S$ 
```

Algorithm 9 *ForwardExactSearch*

Input: A pattern sequence P , BWT string of $\tilde{T}$

Output: A bi-interval I and the stopping position h

```
1:  $x \leftarrow P[0]$ 
2:  $I \leftarrow [C(x), C(\bar{x}), Occ(x, |\tilde{T}|)]$ 
3: for  $h \leftarrow 1$  to  $|P| - 1$  do
4:    $x = P[h]$ 
5:    $I \leftarrow FMDForwardSearch(x, I)$ 
6:   if  $s_I == 0$  then
7:     Stop
8:   end if
9: end for
10: return  $[h, I]$ 
```

Algorithm 10 *BackwardExactSearch*

Input: A pattern sequence P , BWT string of $\tilde{T}$

Output: A bi-interval I and the stopping position h

```
1:  $l_p = |P|$ 
2:  $x \leftarrow P[l_p - 1]$ 
3:  $I \leftarrow [C(x), C(\bar{x}), Occ(x, |\tilde{T}|)]$ 
4: for  $h \leftarrow l_p - 2$  to  $0$  do
5:    $x = P[h]$ 
6:    $I \leftarrow FMDBackwardSearch(x, I)$ 
7:   if  $s_I == 0$  then
8:     Stop
9:   end if
10: end for
11: return  $[h, I]$ 
```

Then A_{xy} and P_x should be modified when $x = c$ or t . We have

$$\begin{aligned} P'_c &= (1 - F) \cdot P_c, & A'_{cy} &= (1 - F) \cdot A_{cy} \\ P'_t &= P_t + F \cdot P_c, & A'_{ty} &= A_{ty} + F \cdot A_{cy} \end{aligned}$$

Furthermore, P_x and G_y are assumed to be approximately 1/4. It implies that the four DNA bases are evenly distributed in both reference genome sequence and read sequences. Moreover, the identity of alignments is assumed to be 99% and hence

$$A_{xy} = \begin{cases} 0.99/4 & \text{if } x = y, \\ 0.01/12 & \text{if } x \neq y. \end{cases}$$

Taking the above assumptions together with $F \approx 1$ and $T = 10/\ln 10$, we have the scoring matrix specific to alignments of BS reads in Table 3.2. We also used the same scoring matrix as the default one in our tool TAMeBS.

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

Table 3.2: The likelihood-ratio scoring matrix (from [15]) used in *TAMeBS*.

However, the previous assumptions cannot fit the practical scenario. For instance, according to the reference genome *Arabidopsis thaliana* (*A.thaliana*) from NCBI, the proportions of the four DNA bases are not identical, which are estimated as

$$n_a : n_c : n_g : n_t = 3 : 5 : 5 : 3.$$

Moreover, methylated Cs have different distributions in different organisms, as we introduced earlier. In mammals, methylated Cs can be rarely found in non-CG contexts. In contrast, a small proportion of Cs in non-CG context can be methylated in plant cells. For individuals of *A.thaliana*, the ratios of methylated Cs in different contexts (i.e., CG, CHG and CHH) are about 55%, 23% and 22% on average [40]. Therefore, the scoring matrix designed for aligning BS reads should consider these practical issues so that the mapping results can be more accurate.

We tried several settings of the scoring matrixes depending on the real situation. The experimental results are presented in the Results section. Overall, with these more practical scoring schemes, much more reads can be aligned to the reference genome. Such improvement is expected to provide more useful information for the downstream analysis.

3.3.3 Methylation calling

In order to infer the methylation status of a cytosine in reference genome, we have to check the reported alignments that cover its position. If a genome C is mapped by a read C in an alignment, we say that the genome C has a *methylated call*; whereas if it is mapped by a read T in an alignment, it has an *unmethylated call*.

The methylation status of a genome C can be indicated by two measures. If only one allele is considered, a cytosine is either methylated or unmethylated. In this case, an *absolute methylation level* is used to measure the methylation status of a base C, which is either 1 or 0. However, if a sample consists of multiple cells in an experiment, an *average methylation level* of a genome C can be calculated as the number of methylated calls divided by the total number of methylated and unmethylated calls. Therefore, an

average methylation level has a range between 0 and 1.

Our software, TAMEBS, measures cytosine states with both absolute and average methylation levels. It produces two files in the methylation calling step. The first tabular file contains the average methylation level of every base C in the reference genome, together with the corresponding numbers of methylated calls and unmethylated calls. The second file summarizes the percentages of methylated cytosines in different genomic contexts. TAMEBS utilizes both absolute and average methylation levels to infer the percentages. With the absolute methylation levels, TAMEBS counts all methylated calls in each genomic context regardless of the repeated genomic positions, which is similar to the approach of Bismark. In contrast, when we use average methylation levels to make estimation, each genomic position is counted once if it is covered by an alignment. In such a case, a base C in the reference genome sequence is considered methylated if its average methylation level exceeds 0.5.

3.4 Results

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

All experiments were run on a Linux server with processor Intel(R) Xeon(R) CPU E5-2650 @ 2.00GHz and RAM 32GB. We compared the performance of TAMEBS with four popular BS read alignment tools, BS Seeker, Bismark, Last and BatMeth. For methylation estimation, we compare it with Bismark only, as Bismark was previously shown capable of achieving very high accuracy [31]. All tested tools were applied

using their default parameter settings except for those error-related settings. Specially for BS Seeker and Bismark, we tried different values for their error-related parameters and used the best results for performance comparison.

3.4.1 Simulation experiments

Experiment setup

Totally we conducted two simulation experiments to evaluate the alignment performance. In the first experiment, we simulated three datasets, each of which contains 11 million 100-bp reads. All reads were randomly generated from the reference genome *A.thaliana*. The three datasets contain three, five, and seven mismatches per read, respectively (*i.e.*, with the mutation rate r of 3%, 5%, and 7%, respectively). In order to comprehensively evaluate mapping capabilities, each dataset comprises 11 subsets generated from the reference genome with different proportions of methylated Cs, ranging from 0% to 100% by increment of 10%. All subsets are of the same size, *i.e.*, each containing one million reads. We sampled methylated Cs uniformly according to a given methylation percentage, regardless of their genomic contexts. Furthermore, we followed [15] to set the bisulfite conversion rate as 99%.

We noticed that the proportion of methylated cytosines in real plant genomes is estimated between 5% and 25% [18]. In order to compare the alignment performance of these tools under more realistic setup, we designed the second simulation experiment. We simulated eight datasets by using the BS read simulator Sherman (<http://www.bioinformatics.bbsrc.ac.uk/projects/sherman/>). Each dataset contained one million 100-bp reads generated from *A.thaliana* with bisulfite conversion rate of 90%. In Sherman, the bisulfite conversion rate is defined as the percentage of Cs

converted to Ts regardless of their genomic contexts. Thus, the proportion of methylated Cs in this simulation setup was 10%. In addition, these eight datasets contained 0 to 7 SNPs per read, respectively.

To evaluate the performance of a mapping tool in methylation estimation, we used Sherman to simulate a dataset of BS reads from the reference genome *A.thaliana* under a more realistic scenario. This large-scale dataset consisted of four subsets, each of which contained one million 100bp-reads at a fixed mutation rate (0, 1, 3, and 5 SNPs per read for four subsets, respectively). Moreover, we set the bisulfite conversion rates for cytosines in CG and non-CG contexts as 40% and 80%, respectively, based on a previously reported distribution of methylated cytosines in *A.thaliana* [40].

Evaluation metrics

Note that all the tested tools report uniquely best alignments by default. Previous studies [7, 39] used *mapping efficiency* for performance evaluation, which refers to the percentage of reads that can be uniquely mapped by a tool. On simulated datasets, we know where each read originates in the reference genome. In this case, a read is considered *correctly aligned* if it is aligned to its original genomic location or *wrongly aligned* if it is aligned to a different location. We hence define *sensitivity* as the percentage of total reads that are correctly aligned and *precision* as the percentage of aligned reads that are correctly aligned. A single evaluation metric, called *F-measure*, is defined as the harmonic mean of sensitivity and precision. That is,

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

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

To compare and evaluate the performance in methylation calling, we calculate the percentage of methylated Cs in each of the three genomic contexts using the absolute methylation level from the alignment results reported by a tested tool (either TAMeBS or Bismark). We then show how close it is from the true methylation percentage in the simulation study. Besides the overall methylation percentages, we are also interested in the absolute numbers of methylated and unmethylated cytosines in genome called by a mapping tool.

Alignment results comparison and discussion

The detailed results of the first experiment for the five BS read mapping tools are summarized in Table 3.3. All the tested tools performed very well at the low mutation rate $r = 3\%$. Their mapping efficiencies and F-measures can generally achieve up to 92% and 96%, respectively, while TAMeBS achieved the highest (95.85% and 97.76%).

At the medium mutation rate $r = 5\%$, Bismark and BS Seeker failed to obtain satisfactory results as their mapping efficiencies dropped dramatically below 80%. We believe that it is mainly due to their built-in aligner Bowtie, which allows to map efficiently only reads with a limited number (≤ 3) of mutations at the 5' end (see Section 3.1.2 for details). Again TAMeBS achieved the highest mapping efficiency and F-measure (95.78% and 97.64%), improving over the second best aligner BatMeth by 2.57% and 1.15%, respectively. We also noticed that BatMeth found the alignments of the BS reads that allow at most 5 mismatches with very high accuracy in terms of precision (100%). This is because BatMeth always discards all alignments without any further checking if they match multiple substrings of the reference genome. In con-

trast, the other four software tools try to select an alignment hit that is both unique and the best in terms of quality (Bismark and BS Seeker) or mapping score (Last and TAMEBS). It is not surprising that such selection might sacrifice some accuracy since some reads have their uniquely best alignments at the wrong genomic locations.

We specially took a test on TAMEBS for its accuracy of selecting the uniquely best alignments. For the sake of convenient discussion, we call an alignment having multiple mapping hits in the reference genome as an *ambiguous* alignment. We extracted the reads having ambiguous alignments from the first two data sets via exhaustive checking. By comparing the alignment results reported by TAMEBS to their true genomic locations, we found that 13.36% and 21.68% of the ambiguous alignments in the first two datasets ($r = 3\%$ and $r = 5\%$) were reported by TAMEBS with accuracies of 82.03% and 83.86%, respectively. BatMeth did not report any ambiguous alignment at all, as what we expected.

At the high mutation rate $r = 7\%$, the mapping efficiency of BatMeth dropped below 24%, which means that BatMeth failed to align more than 76% reads. Such performance agreed with the results reported in [31]. In contrast, TAMEBS and Last maintained high mapping efficiency as well as high F-measure score. Compared with Last, TAMEBS uniquely mapped 3.2% more reads with an improved F-measure score of 1.48%. Considering the CPU time used by each mapping tool, TAMEBS ran comparatively fast with other tools at both the low and medium mutation rates, but unfortunately several times slower at the high mutation rate. We expected this relatively low time efficiency of TAMEBS, as it was aimed mainly at achieving high mapping efficiency and accuracy (in terms of F-measure) for the subsequent accurate methylation estimation analysis.

The results of the second experiment is depicted in Figure 3.7. Similar to the first

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

Table 3.3: Alignment performances of five BS read mapping tools on three simulated data sets. MapEff, Sens, prec, and F-ms represent the mapping efficiency, sensitivity, precision and F-measure, respectively. When the mutation rate increased to 5%, Bismark and BS Seeker cannot uniquely align over 80% reads. When the mutation rate further increased to 7%, only *TAMeBS* and Last successfully mapped more than 90% reads. In comparison, BatMeth mapped less than 24% reads.

simulation results, *TAMeBS* achieved the highest sensitivity and mapping efficiency on almost all test datasets. It also offered the best balance between sensitivity and precision among all the compared mapping tools. Note that the precision of *TAMeBS* was lower than a few other tools as a result of the attempt to map more BS reads onto the reference genome. Although having the highest precision on the datasets with less than six SNPs per read, BatMeth achieved the sensitivity significantly lower than *TAMeBS* and thus the worse mapping performance in terms of F-measure. We further compared the output alignments of *TAMeBS* with those of Last and BatMeth (see Table 3.4 and Table 3.5). We found that more than 99.97% of the read alignments output by BatMeth were also found by *TAMeBS* on the first six datasets. And, approximately 98.62% of

the alignments output by Last were also found by TAMEBS. These experimental results clearly show that TAMEBS can achieve very high mapping efficiency and sensitivity at a small cost of precision.

r	0	1%	2%	3%	4%	5%	Ave
$A_T \cap A_B(K)$	957.39	952.64	947.71	942.11	937.80	932.6	945.07
$A_T \setminus A_B(K)$	0.54	5.49	10.45	15.07	19.38	23.84	12.46
$A_B \setminus A_T$	0	0	105	508	153	709	245.8

Table 3.4: Comparison of the alignments reported by TAMEBS and BatMeth. r is the mutation rate, ranging from 0 to 5%, due to the failure alignment of BatMeth when $r \geq 6\%$. A_T and A_B represent the alignments reported by TAMEBS and BatMeth, respectively. $A_T \cap A_B$ counts the number of alignments found by both softwares. $A_T \setminus A_B$ ($A_B \setminus A_T$) counts the number of alignments found by TAMEBS (BatMeth) but missed by BatMeth (TAMEBS).

r	0	1%	2%	3%	4%	5%	6%	7%	Ave
$A_T \cap A_L(K)$	948.40	947.75	947.04	945.53	944.72	942.08	939.77	929.73	943.13
$A_T \setminus A_L(K)$	9.54	10.38	11.12	11.65	12.64	14.37	16.45	19.51	13.20
$A_B \setminus A_L(K)$	0.268	0.498	0.619	1.063	0.903	1.493	1.937	8.560	1917.6

Table 3.5: Comparison of the alignments reported by TAMEBS and Last. r is the mutation rate, ranging from 0 to 7%. A_L represent the alignments reported by Last. The other representations have the similar meaning to those in Table 3.4.

Results of methylation calling

In the experiment simulated to evaluate the performance of a mapping tool in methylation calling and estimation, we obtained the results shown in Table 3.6. In particular, TAMEBS reported the same methylation percentages as Bismark in all genomic contexts. It is worth noting that, as a previous study has demonstrated, Bismark could generally achieve very high quantitative accuracy in estimating methylation percentages [31]. Thus, TAMEBS is able to perform accurate methylation estimation as well.

(a)

(b)

(c)

(d)

Figure 3.7: Simulation results of the eight datasets generated by Sherman. Only *TAMEBS* and *Last* obtained good enough mapping performance on all eight sets of data. For the other four tools, we presented their mapping results only when their mapping efficiency were over 50%.

We further compared the absolute numbers of cytosines in the genome called by these two tools. It shall be noted that each C in the genome has to be called as methy-

lated, or called as unmethylated, or not called by a tool at all. TAMEBS successfully called about 76% cytosines in the reference genome, whereas Bismark called only 48% cytosines. It means that there were more than half of cytosines in the genome for which Bismark did not find any read alignment (i.e. those Cs are not called as methylated or unmethylated by Bismark). Moreover, we found that 83% of cytosines called by Bismark were also called by TAMEBS. To be specific, TAMEBS called about 17 million more cytosines than Bismark (among a total of about 42 million cytosines in the genome). This superior performance of TAMEBS shall be mainly attributed to its high mapping efficiency (95.8%) and F-measure (97.8%) in the previous alignment procedure. We believe that a high methylation calling rate of genomic cytosines is vital to many downstream applications such as the genome-wide detection of differentially methylated regions [17].

Tools	<i>TAMEBS</i>	Bismark($n = 3, l = 44$)
CPU/elapsed time(s)	1806/1820	942/1882
Mapping Efficiency (%)	95.8	59.3
Sensitivity(%)	95.7	59.3
Precision(%)	99.9	99.9
F-measure(%)	97.8	74.4
mCG/mCHG/mCHH (%)	59/20/20	59/20/20

Table 3.6: The mapping and mC estimation performance of TAMEBS and Bismark. Note that Bismark achieved the best results with parameter setting $n = 3, l = 44$.

3.4.2 Biological experiments

Datasets

To evaluate our tool on real biological data, we downloaded about 25M paired-end reads from the NCBI Sequence Read Archive (SRA). The SRA accession number is

ERR046546, and the reads were sequenced from the *A. thaliana* genome by using Illumina Genome Analyzer IIX. As the current implementation of TAMeBS takes only single-end reads as input, we chose to align the first read of each pair in our experiment. According to our observation, the first base is 'N' for most reads. Thus, we cut the first base off from each read. At the end, we extracted 10 million 100-bp single-end reads to construct a test dataset. Besides, we also performed the same process on the whole dataset and obtained 25M single-end reads to be tested as well.

Evaluation metrics

We adopted mapping efficiency to measure the performance of a mapping tool. It can be expected that the accuracy of each tested aligner is hardly evaluated, because the original genomic locations are unknown for the reads in this biological dataset. In order to further evaluate the mapping sensitivity, we performed exhaustive search for all the reads that can be uniquely mapped onto the reference genome within 3 mismatches. The mapping sensitivity is thus defined as the percentage of those uniquely mapped reads that would be returned by a mapping tool.

Results of real experiments

The mapping efficiency and running time of each mapping tool is summarized in Table 3.7. We note that BatMeth consumed the least CPU time, but reported the fewest uniquely best alignments (<38%). With the parameter setting $k = 5$, TAMeBS achieved not only higher mapping efficiency than Last (57.05% vs 56.96%), which obtained the second best performance, but also higher mapping speed. With $k = 7$, TAMeBS achieved even higher mapping efficiency (57.48%) at the expense of long CPU time (>2h).

Table 3.9 presents the mapping sensitivity achieved by each mapping tool. TAMEBS (with parameter setting $k = 3$) achieved the highest mapping sensitivity at 99.80%. It can be indicated that TAMEBS found almost all the uniquely best alignments within 3 mismatches. We conducted experiments on the whole 25M single-end reads as well and obtained the similar results (see Table 3.8 and Table 3.10).

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

Table 3.7: Mapping 10M 100-bp single-end reads extracted from ERR046546 to the *A. thaliana* genome.

Software	Mapping Efficiency(%)	CPU Time(h:m:s)
TAMEBS($k = 3$)	55.62	01 : 59 : 22
TAMEBS($k = 5$)	57.04	02 : 56 : 48
BatMeth($n = 3$)	37.75	00 : 46 : 09
BatMeth($n = 5$)	37.01	02 : 00 : 14
BS Seeker	15.25	02 : 34 : 01
Last	56.98	02 : 36 : 45
Bismark($n = 3, l = 36$)	54.62	01 : 37 : 42

Table 3.8: Alignment Results for ERR046546, ~ 25.2M reads

Methylation estimation results

We ran TAMEBS and Bismark to infer the methylation status from their alignment results reported on the two real datasets (containing 10M and 25.2M single-end reads, respectively). The estimation results are summarized in Table 3.11 and Table 3.12.

As we have no idea about the true methylation status, we cannot use any metrics to evaluate the performance of each tool. But we can still observe that these two software tools reported quite similar results on both the smaller dataset and the larger dataset.

Software	TAMeBS($k = 3$)	BatMeth($n = 3$)	BS Seeker	Last	Bismark
Sensitivity(%)	99.80	79.94	97.57	98.97	98.76

Table 3.9: Mapping sensitivity achieved by each tool to find the percentage of unique best alignments within three mismatches. BS Seeker and Bismark used the same parameter settings as in the above table. Among the total 10M tested reads, about 4.7M reads has unique alignments within three mismatches.

Software	TAMeBS($k = 3$)	BatMeth($n = 3$)	BS Seeker	Last	Bismark
Sensitivity(%)	99.79	79.82	97.59	99.00	98.76

Table 3.10: Mapping sensitivity achieved by each software tool for ERR046546, ~ 25.2M reads

Software	Total mC(%)	CpG(%)	CHG(%)	CHH(%)
TAMeBS($k = 3$)	9.88	31.02	10.63	2.89
TAMeBS($k = 5$)	9.95	31.29	10.70	2.96
Bismark	9.58	30.0	10.3	2.8

Table 3.11: Methylation calling results of TAMeBS and Bismark for ERR046546 (10M 100-bp reads). Total mC, CpG, CHG, and CHH denote the percentage of methylated Cs called by a mapping tool in all cytosines, CpG contexts, CHG contexts and CHH contexts, respectively.

Software	Total mC(%)	CpG(%)	CHG(%)	CHH(%)
TAMeBS($k = 3$)	9.9	31.0	10.6	2.9
TAMeBS($k = 5$)	9.9	31.2	10.7	3.0
Bismark	9.6	29.9	10.3	2.8

Table 3.12: Methylation estimation for ERR046546, ~ 25.2 M reads. Notations have similar meanings to those in Table 3.11.

3.4.3 Discussion of scoring matrixes

In this section, we discuss five different likelihood-ratio scoring matrixes besides the default one introduced previously. These matrixes are defined based on either the different distributions of four bases in reference genome sequences or the different methylation levels of cytosines in different genomic contexts. Among them, one matrix can be utilized for all eukaryotic genomes, three matrixes are developed for plants and the other is for human genomes. We give a detailed discussion of each matrix with experimental results in the following parts.

Matrix A

Similar to the calculation of the default scoring matrix, we assume that the probability that a base occurs in either reference genome sequence or the read sequences is $1/4$. Moreover, we employ the same definition of the probability A_{xy} that base x of a read sequence is aligned to base y of the reference genome sequence in a true alignment, that is,

$$A_{xy} = \begin{cases} 0.99/4 & \text{if } x = y, \\ 0.01/12 & \text{if } x \neq y. \end{cases}$$

In this case, the only different assumption is the bisulfite conversion fraction F , which is set to be 90%. According to the definition of scores, we have the first scoring matrix S_A in Table 3.13.

		Genome			
		A	C	G	T
Read	A	6	-19	-19	-19
	C	-19	6	-19	-19
	G	-19	-19	6	-19
	T	-19	3	-19	3

Table 3.13: The likelihood-ratio scoring matrix S_A .

To evaluate the alignment performance of our software TAMEBS using matrix S_A , we used Sherman to simulate one million 100bp bisulfite sequencing reads with 90% of Cs converted to Ts from the reference genome *A.thaliana*. Furthermore, each read does not contain any mutation. Let $D_{A.th}$ denote this simulation dataset. We first applied TAMEBS with the default scoring matrix to mapping dataset $D_{A.th}$ and then compared the result with that produced by TAMEBS with scoring matrix S_A . Finally, we observed that the scoring matrix S_A gave rise to exactly the same performance as the default scoring matrix (see Table 3.17).

Matrix B

As we mentioned previously, the distribution of the four bases in the reference genome *A.thaliana* is

$$G_a = G_t = 3/16, G_c = G_g = 5/16.$$

Accordingly, we defined another scoring matrix under the same assumptions for the scoring matrix S_A , except for the probability of G_y . Thus, we achieved the second scoring matrix (S_B), specific to the species *A.thaliana*, in Table 3.14.

		Genome			
		A	C	G	T
Read	A	7	-20	-20	-18
	C	-18	5	-20	-18
	G	-18	-20	5	-18
	T	-18	2	-20	5

Table 3.14: The likelihood-ratio scoring matrix S_B .

Using scoring matrix S_B , we ran TAMEBS on the simulated dataset $D_{A.th}$ and evaluated the corresponding alignment results. From Table 3.17, we can observe that more alignments were found by using the scoring matrix S_B than using the default one (in terms of mapping efficiency, 96.81% vs 95.79%). However, the accuracy, in terms of F-measure, dropped by 0.14% when the scoring matrix S_B was employed because of the decreased precision (99.32% vs 100%). Compared to the mapping efficiency that was increased by more than 1%, the less loss of accuracy (in terms of F-measure) and precision should be tolerable. Furthermore, we believe that the more alignments detected, the more benefits in the downstream analysis.

Matrix C

In this case, we assume that the probabilities P_x and G_y have the same values, i.e., 1/4. We also employ the same definition of A_{xy} in the above cases. At this time, we consider the different distributions of methylated Cs in CG and non-CG contexts. Specially for species of *A.thaliana*, the proportion of methylated Cs in CG context is around 60% on average, while about 20% of Cs in non-CG context (i.e., CHG or CHH) are methylated. Therefore, we define the bisulfite conversion fraction F as follows,

$$F = \begin{cases} 40\% & \text{if } x = c \text{ or } t, \text{ and } x \text{ is followed by } g, \\ 80\% & \text{otherwise.} \end{cases}$$

Given these assumptions, we have a scoring matrix, denoted by S_C , in Table 3.15.

		Genome			
		A	C	G	T
Read	A	6	-19	-19	-19
	C	-19	6	-19	-19
	G	-19	-19	6	-19
	T	-19	1(2)	-19	5(3)

Table 3.15: The likelihood-ratio scoring matrix S_C . The values in parentheses represent the scores with respect to non-CG contexts.

Again, we applied the scoring matrix S_C and executed TAMEBS on the simulated dataset $D_{A.th}$. The results are shown in Table 3.17. Much more alignments were reported, resulting in the mapping efficiency of 96.91%. Meanwhile, the precision decreased to 99.24%, which led to less F-measure (97.69%). However, we believe that such little loss of accuracy can be ignored if the increase of mapping efficiency is weighed more.

Matrix D

To make our discussion more practical, we consider both the different distribution of bases in the reference genome sequence and the different bisulfite conversion fractions of Cs in distinct genomic contexts. In other words, the probability G_y is the same as that in the definition of the scoring matrix S_B , while the bisulfite conversion fraction F is defined in the same way as the scoring matrix S_C . The probabilities P_x and A_{xy} are not changed. We can get a new scoring matrix S_D in Table 3.16.

Employing the scoring matrix S_D , we aligned the BS reads in the simulated dataset $D_{A.th}$ by running TAMEBS and obtained the alignment results summarized in Table 3.17. It is worth noting that the alignment performance was equally good with those

		Genome			
		A	C	G	T
Read	A	7	-20	-20	-18
	C	-18	5	-20	-18
	G	-18	-20	5	-18
	T	-18	0(2)	-20	6(5)

Table 3.16: The likelihood-ratio scoring matrix S_D . The values in parentheses represent the scores with respect to non-CG contexts.

achieved by using the scoring matrix S_A and the default matrix. Comparing the performances of TAMeBS using these four scoring matrixes, we are able to conclude that changing either the distribution of four bases in reference genome sequence or the bisulfite conversion fraction in distinct sequence contexts is more meaningful, whereas changing both of them contributes little to the accuracy of alignments.

Matrix	MapEff (%)	Sens (%)	Prec (%)	F-ms (%)
S_0	95.79	95.79	100.00	97.85
S_A	95.79	95.79	100.00	97.85
S_B	96.81	96.15	99.32	97.71
S_C	96.91	96.18	99.24	97.69
S_D	95.79	95.79	100.00	97.85

Table 3.17: Comparison of the alignment performances of TAMeBS using different scoring matrixes. S_0 is the default scoring matrix that is same with Last.

Matrix H

We specially construct a scoring matrix for human genome where methylated Cs dominantly locate in CG contexts. Accordingly, we specially define the bisulfite conversion fraction F to be

$$F = \begin{cases} 20\% & \text{if } x = c \text{ or } t, \text{ and } x \text{ is followed by } g, \\ 99\% & \text{otherwise.} \end{cases}$$

It corresponds to the case where almost all methylated Cs can be found in CG contexts and the proportion of methylated Cs in CG contexts is 80% [54]. As for the values of P_x , G_y and A_{xy} , they are defined in the same way as in the default scoring matrix. Therefore, we can achieve the scoring matrix for human genome in Table 3.18, denoted by S_H .

		Genome			
		A	C	G	T
Read	A	6	-19	-19	-19
	C	-19	6	-19	-19
	G	-19	-19	6	-19
	T	-19	-2(3)	-19	5(3)

Table 3.18: The likelihood-ratio scoring matrix S_H . The values in parentheses represent the scores with respect to non-CG contexts.

To evaluate the alignment performance of TAMeBS with different scoring matrixes specific to human genomes, we simulated a new dataset, denoted as S_H , by using the software package DNemulator [15]. DNemulator can simulate various cytosine methylation rates and bisulfite conversion. Moreover, it simulates DNA sequencing errors and polymorphisms based on real sequencing data. So DNemulator can be used as a satisfactory tool to simulate BS reads for human genomes, because the simulated reads can be treated as those from real sequencing. It can be downloaded at www.cbrc.jp/dnemulator/.

To simulate the dataset D_H , we used the human chromosome 22 (chr22) as the reference sequence. We also assigned polymorphisms in the chr22 based on the real frequencies obtained from 'snp138Common.txt' in the UCSC Genome Browser (<http://genome.ucsc.edu/>). We generated one million BS reads with 100bp length, and mapped these reads by running TAMeBS, employing either the default scoring matrix or S_H . In particular, the parameter k in TAMeBS was set to be 3, which means that we

tried to find all alignments with at most 3 mismatches. The results of these two runs are summarized in Table 3.19. It is worth noting that scores of both the mapping efficiency and the accuracy (in terms of F-measure) were improved by utilizing the scoring matrix S_H , compared to the results obtained by using the default scoring matrix. Therefore, we can expect better performance of read mapping if we modify the definition of bisulfite conversion fraction F for different eukaryotic organisms.

Matrix	MapEff (%)	Sens (%)	Prec (%)	F-ms (%)
S_0	93.28	93.15	99.86	96.39
S_H	93.66	93.40	99.73	96.46

Table 3.19: Alignment results of TAMEBS using different scoring matrixes to map reads from dataset D_H . S_0 is the default scoring matrix.

3.5 Conclusion

In this chapter, we introduced a new bisulfite sequencing read mapping tool called TAMEBS for DNA methylation analysis. It aims to align long bisulfite treated reads onto a reference genome sequence with high mapping efficiency and sensitivity so that the methylation status of each genomic cytosine can be accurately estimated. To this end, we build TAMEBS on several recent advances in sequence alignment techniques, including bi-directional FM-index, approximate seeds, and likelihood-ratio scoring matrix which is designed particularly for aligning bisulfite treated DNA reads.

In both simulated and real data experiments, TAMEBS demonstrated strong ability to detect more uniquely mapped reads than other tested tools while retaining a good balance between mapping sensitivity and precision. Moreover, TAMEBS achieved comparably high accuracy in methylation percentage estimation with the existing mapping tool Bismark. However, TAMEBS could determine the methylation status for many

more cytosines in the genome than Bismark. It is a feature that many subsequent analyses shall find beneficial.

We designed five additional scoring matrix for TAMeBS to deal with practical alignment problems. The results of the experiments showed the improvements in both mapping efficiency and accuracy brought by most of these scoring matrixes. These results also inspire us to develop an approach to the automatic selection of appropriate scoring schemes for various datasets.

Considering the implementation of TAMeBS, its running time increases substantially when the number of mismatches per read is more than five. When a read contains more mismatches, the corresponding approximate seeds become shorter and the resulting number of candidate alignments turns to be larger. The more candidate alignments are produced, the more time are required to score them and select the best one. In addition, TAMeBS usually requires more memory space than other mapping tools as a result of the application of bi-directional index structure. The bi-directional FM-index together with the reference genome sequence totally occupies $5n + \frac{5}{32}n \log n$ space, where n represents the length of the reference genome. When the number of mismatches per read increases, the required memory for recording candidate alignments grows accordingly. Although it shall not be a big issue that prevents TAMeBS from running, code optimization is still required to reduce the computing cost.

Moreover, TAMeBS can only align single-end BS reads generated from directional protocols. To align single-end BS reads from non-directional protocols, we have to construct an extra file to store the reverse complements of the BS reads and run TAMeBS twice on the original read file and their reverse complement sequence, respectively. On the other hand, paired-end reads can improve the accuracy of alignment, so the implementation of mapping paired-end reads is also on the future schedule.

Another limitation of TAMeBS is that insertions and deletions (indels) are not considered in the read alignment procedure. However, indels have low chances to occur in a sample genome.

Chapter 4

Conclusion

4.1 Summary

In this thesis, we have mainly addressed two algorithmic problems related to analyzing genetic/epigenetic variations. The first one is to detect single nucleotide polymorphisms in a sample DNA sequence using base-specific cleavage reaction and MALDI-TOF mass spectrometry. To solve this problem, a reference sequence is required to generate the *in-silico* predicted complementary mass spectra with respect to four cut bases. Comparing the measured mass spectra of the sample sequence with the predicted mass spectra, we find the minimal set of potential SNP mutations that can explain the discrepancies between the measured and the predicted mass spectra. It is a challenging computational problem to integrate the information in the four complementary base-specific mass spectra in an efficient and effective way. Several software packages exist to facilitate the automatic detection of SNPs from mass spectrometry data. However, they cannot provide satisfactory SNP detection results, because they either have no attempt to tell the true SNP mutations or have low efficiency to detect SNP mutations in

close vicinity.

The second problem we addressed is to align bisulfite sequencing (BS) DNA reads onto a reference genome sequence. The alignment problem with respect to BS reads is challenging due to several special reasons. The most important reason is the asymmetrical C/T mapping, because a part of Ts in the read sequences are attributed to the bisulfite conversion of the unmethylated Cs. Moreover, the different distributions of methylated Cs in the two complementary DNA strands also increase the difficulties and complexities of mapping BS reads onto the reference genome. There exist two classes of tools developed for aligning BS read data: three-letter aligners and wild-card aligners. However, neither of them can achieve a good balance between accuracy and sensitivity.

4.2 Technical Contributions

SNPMS

We have developed an iterative and progressive algorithm to accurately detect SNP mutations from mass spectrometry data. It works mainly by repeatedly identifying the SNP mutations that have potentially occurred in the sample sequence while progressively updating the reference sequence by correcting these mutations. We implemented the algorithm in a program called SNPMS, which is freely available at <http://www1.spms.ntu.edu.sg/~chenxin/SnpMs>. Comparative evaluation on both simulated and real biological datasets demonstrated the strong ability of our software to accurately detect SNP mutations.

TAMeBS

We have also developed a new approach to aligning BS reads and estimating methylation states. Our approach is based on several recent advances in sequence alignment techniques, including bi-directional FM-index, approximate seeds and likelihood-ratio scoring matrix. We implemented our algorithm in a software package called TAMeBS. Its source code can be freely downloaded from <http://sourceforge.net/projects/tamebs/>. Compared to the other tested software packages on both simulated and real data, TAMeBS showed ability to detect more uniquely mapped reads while achieving a better balance between sensitivity and precision. Moreover, TAMeBS can also perform methylation calling and estimate methylation states of cytosines from the alignment results. Experiment results also showed the strong ability of TAMeBS to accurately estimate methylation percentages and determine methylation status for many more cytosines.

4.3 Future Work

Besides the modification of the proposed algorithms to complement their function and improve their efficiencies, we will take one step further to identify differentially methylated regions between sample groups.

Most studies in cytosine methylation focus on the CpG sequence contexts in mammalian genomes, due to the unknown biological functions of methylated cytosines in non-CpG contexts. According to many previous works, a single methylated CpG may affect gene expression regulation and hence play a critical role at the beginning and during the development of many diseases [3]. However, it is not convincing to merely analyze methylation states of single CpG sites, because the existence of SNPs and se-

quencing errors might lead to defective or even faulty analyses of single CpG sites [47]. Therefore, profound analyses are required to understand the biological functions of methylation.

It has been shown that functions of methylation are associated with genomic regions, such as the promoter regions, gene bodies, etc. [24]. Most studies in analyzing methylation in mammals pay more attention to CpG islands (CGIs). A CpG island is usually referred to as the region of DNA that has a high frequency of CpG sites. Specially in human genomes, more than 70% of promoters have high CpG concentrations, which implies higher frequencies of CpG sites within the 3 thousand-base neighborhoods around the transcriptional start sites (TSSs) [61]. Compared to the tissue-specific methylation in gene bodies, most CGIs of the promoters are unmethylated. Methylated CpG sites in the gene promoters may lead to long-term silencing, which has been observed in various human cancers, such as lung and ovarian cancers. Moreover, methylation in gene bodies is also found to be the cause of cancer gene mutations in both somatic cells and the germline [24].

All of the above facts lead to the methylation analyses in the regional scale so that the more detailed maps between methylation and diseases can be uncovered. Thus, the differentially methylated regions (DMRs) between sample groups (e.g., normal tissues vs cancer tissues) need to be identified in whole genome. A DNA region can be considered as a DMR if the DNA methylation levels within the region are consistently or statistical-significantly different between sample groups.

A number of research works in identifying DMRs have been published, but few with respect to utilizing whole-genome bisulfite sequencing (WGBS) data. Although WGBS enables a straight-forward quantification of methylation at very high resolution, the expensive cost and computationally complex data prevent it from wide application [46].

To the best of our knowledge, BSmooth [17] firstly provided a solution to detecting DMRs from WGBS data. After aligning BS reads to the reference genome, BSmooth applies a local-likelihood smoother to estimate the methylation level in a genomic region for a single sample and then forms a statistic comprising the mean differences and standard errors of the estimates to identify DMRs. BiSeq was proposed in [20] several months after BSmooth. Although BiSeq aims to analyze data generated from reduced representation bisulfite sequencing (RRBS), it can also be applied to analyzing WGBS data. Briefly, BiSeq starts from defining CpG clusters, which is followed by local smoothing of the methylation data within each CpG cluster. The DMRs are identified by using a hierarchical testing procedure. Another approach to detecting DMRs from WGBS data is implemented in the software package called Bisulfighter [59]. Different from the previous two methods, Bisulfighter uses a framework based on hidden Markov models (HMMs) and the expectation-maximization (EM) algorithm to identify DMRs. However, it is not surprising that none of these methods can comprehensively detect DMRs from WGBS data. The smoothing based methods (i.e., BSmooth and BiSeq) usually assume that the methylation level varies smoothly in a genomic region, resulting in the failure to detect single CpG sites that have significantly differential methylation status between sample groups. In contrast, Bisulfighter is designed to identify DMRs with various sizes, but it can only be applicable when there is no biological replicate. However, it is not recommended when experiments are performed without biological replicates, because the experimental results are insufficient to reveal the relationships between methylation and diseases [46]. Thus, we are motivated to approach the identification of DMRs from WGBS data in our future study.

Bibliography

- [1] Mass spectrometry analysis of oligonucleotide syntheses. Technical report, Integrated DNA Technologies, 2005, 2010, 2011.
- [2] Adrian P. Bird. CpG-rich islands and the function of DNA methylation. *Nature*, 321(6067):209–213, May 1986.
- [3] Christoph Bock. Analysing and interpreting DNA methylation data. *Nature Reviews Genetics*, 13(10):705–719, October 2012.
- [4] Sebastian Böcker. SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry. *Bioinformatics*, 19(1):i44, 2003.
- [5] Michael Burrows and David J. Wheeler. A block sorting lossless data compression algorithm. Technical report, Digital System Research Center, 130 Lytton Avenue, Palo Alto, CA, 1994.
- [6] Mark Chee, Robert Yang, Earl Hubbell, Anthony Berno, Xiaohua C. Huang, David Stern, Jim Winkler, David J. Lockhart, Macdonald S. Morris, and Stephen P. A. Fodor. Accessing genetic information with high-density DNA arrays. *Science*, 274(5287):610–614, October 1996.

- [7] Pao-Yang Chen, Shawn J Cokus, and Matteo Pellegrini. BS Seeker: precise mapping for bisulfite sequencing. *BMC Bioinformatics*, 11:203, April 2010.
- [8] Pan Du, Warren A. Kibbe, and Simon M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, August 2006.
- [9] John R. Edwards, Hameer Ruparel, and Jingyue Ju. Mass-spectromethy DNA sequencing. *Mutation Research*, 578(1-2):2–12, June 2005.
- [10] Gerda Egger, Gangning Liang, Ana Aparicio, and Perter A. Jones. Epigenetics in human disease and prospects for epigenetic therapy. *Nature*, 429(6990):457–463, May 2004.
- [11] Melanie Ehrlich, Miguel A. Gama-Sosa, Lan-Hsiang Huang, Rose Marie Midgett, Kenneth C. Kuo, Roy A. McCune, and Charles Gehrke. Amount and distribution of 5-methylcytosine in human DNA from different types of tissues of cells. *Nucleic Acids Research*, 10(8):2709–2721, April 1982.
- [12] S. Emonet, H. N. Shah, A. Cherkaoui, and J. Schrenzel. Application and use of various mass spectrometry methods in clinical microbiology. *Clinical Microbiology and Infection*, 16(11):1604–1613, Nov 2010.
- [13] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *FOCS'00 Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 390, Washington, DC, USA, 2000. IEEE Computer Society.

- [14] S. G. Fischer and L. S. Lerman. DNA fragments differing by single base-pair substitutions are separated in denaturing gradient gels: correspondence with melting theory. *Proceedings of the National Academy of Sciences*, 80(6):1579–1583, March 1983.
- [15] Martin C. Frith, Royta Mori, and Kiyoshi Asai. A mostly traditional approach improves alignment of bisulfite-converted DNA. *Nucleic Acids Research*, 40(13):e100, March 2012.
- [16] Richard A. Gibbs, Phi-Nga Nguyen, Lincoln J. McBride, Sandy M. Koepf, and C. Thomas Caskey. Identification of mutations leading to the Lesch-Nyhan syndrome by automated direct DNA sequencing of in vitro amplified cDNA. *Proceedings of the National Academy of Sciences*, 86(6):1919–1923, March 1989.
- [17] Kasper D Hansen, Benjamin Langmead, and Rafael A Irizarry. BSmooth: from whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biology*, 13:R83, October 2012.
- [18] Thomas J Hardcastle. High-throughput sequencing of cytosine methylation in plant DNA. *Plant Methods*, 9(1):16–24, June 2013.
- [19] Ralf Hartmer, Niels Storm, Sebastian Boecker, Charles P. Rodi, Franz Hillenkamp, Christian Jurinke, and Dirk van den Boom. RNase T1 mediated base-specific cleavage and MALDI-TOF MS for high-throughput comparative sequence analysis. *Nucleic Acids Research*, 31(9):e47, May 2003.
- [20] Katja Hebestreit, Martin Dugas, and Hans-Ulrich Klein. Detection of significantly differentially methylated regions in targeted bisulfite sequencing data. *Bioinformatics*, 29(13):1647–1653, July 2013.

- [21] Karin Hjerno and Peter Hojrup. Calibration of Matrix-Assisted Laser Desorption/Ionization Time-of-Flight peptide mass fingerprinting spectra. *Methods in Molecular Biology*, 367:49–60, 2007.
- [22] Christiane Honisch, Anu Raghunathan, Charles R. Cantor, Bernhard O. Palsson, and Dirk van den Boom. High-throughput mutation detection underlying adaptive evolution of *Escherichia coli*-K12. *Genome Research*, 14(12):2495–2502, December 2004.
- [23] Kristian hovde Liland, Trygve Almoy, and Bjorn-Helge Mevik. Optimal choice of baseline correction for multivariate calibration of spectra. *Applied Spectroscopy*, 64(9):212–220, Sep 2010.
- [24] Peter A Jones. Functions of DNA methylation: islands, start sites, gene bodies and beyond. *Nature Reviews Genetics*, 13(7):484–492, May 2012.
- [25] Michael Karas and Franz Hillenkamp. Laser desorption ionization of proteins with molecular masses exceeding 10000 daltons. *Analytical Chemistry*, 60(20):2299–2301, October 1988.
- [26] W. James Kent and Alan M. Zahler. Conservation, regulation, syntenic, and introns in a large-scale *C. briggsae* - *C. elegans* genomic alignment. *Genome Research*, 10(8):1115–1125, August 2000.
- [27] Szymon M. Kielbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C. Frith. Adaptive seeds tame genomic sequence comparison. *Genome Research*, 21(3):487–493, January 2011.

- [28] Stefan Krebs, Ivica Medugorac, Doris Seichter, and Martin Forster. RNasecut: a MALDI mass spectrometry-based method for SNP discovery. *Nucleic Acids Research*, 31(7):e37, April 2003.
- [29] Felix Krueger and Simon Andrews. Bismark: a flexible aligner and methylation caller for bisulfite-seq applications. *Bioinformatics*, 27(11):1571–1572, June 2011.
- [30] Felix Krueger, Benjamin Kreck, Andre Franke, and Simon Andrews. DNA methylation analysis using short bisulfite sequencing data. *Nature Methods*, 9(2):145–151, January 2012.
- [31] Govindarajan Kunde-Ramamoorthy, Cristian Coarfa, Eleonora Laritsky, Noah J. Kessler, R. Alan Harris, Mingchu Xu, Rui Chen, Lanlan Shen, Aleksandar Milosavljevic, and Robert A. Waterland. Comparison and quantitative verification of mapping algorithms for whole-genome bisulfite sequencing. *Nucleic Acids Research*, 42(6):e43, January 2014.
- [32] T.W. Lam, Ruiqiang Li, Alan Tam, Simon Wong, Edward Wu, and S.M. Yiu. High throughput short read alignment via bi-directional BWT. In *2009 IEEE International Conference on Bioinformatics and Biomedicine*, pages 31–36, Washington, DC, USA, November 2009. IEEE.
- [33] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, March 2009.

- [34] J. Kathleen Lewis, Jing Wei, and Gary Siuzdak. Matrix-assisted laser desorption/ionization mass spectrometry in peptide and protein analysis. *Encyclopedia of Analytical Chemistry*, pages 5880–5894, September 2006.
- [35] Heng Li. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. *Bioinformatics*, 28(14):1838–1844, July 2012.
- [36] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, May 2009.
- [37] Heng Li, Jue Ruan, and Richard Durbin. Mapping short DNA sequencing reads and calling variances using mapping quality scores. *Genome Research*, 18(11):1851–1858, August 2008.
- [38] Ruiqiang Li, Yingrui Li, Xiaodong Fang, Huanming Yang, Jian Wang, Karsten Kristiansen, and Jun Wang. SNP detection for massively parallel whole-genome resequencing. *Genome Research*, 19(6):1124–1132, June 2009.
- [39] Jing-Quan Lim, Chandana Tennakoon, Guoliang Li, Eleanor Wong, Yijun Ruan, Chia-Lin Wei, and Wing-Kin Sung. BatMeth: improved mapper for bisulfite sequencing reads on DNA methylation. *Genome Biology*, 13(10):R82, October 2012.
- [40] Ryan Lister, Ronan C. O’Malley, Julian Tonti-Filippini, Brian D. Gregory, Charles C. Berry, A. Harvey Millar, and Joseph R. Ecker. Highly integrated single-base resolution maps of the epigenome in arabisopsis. *Cell*, 133(3):523–536, August 2008.

- [41] Lin Liu, Yinhu Li, Siliang Li, Ni Hu, Yimin He, Ray Pong, Danni Lin, Lihua Lu, and Maggie Law. Comparison of next-generation sequencing systems. *Journal of Biomedicine and Biotechnology*, 2012(2012):1–11, 2012.
- [42] Bin Ma, John Tromp, and Ming Li. PatternHunter: faster and more sensitive homology search. *Bioinformatics*, 18(3), 2002.
- [43] Elaine R. Mardis. Next-Generation DNA sequencing methods. *Annual Review of Genomics and Human Genetics*, 9:387–402, 2008.
- [44] Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun He Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L. I. Alenquer, Thomas P. Jarvie, Kshama B. Jirage, Jong-Bum Kim, James R. Knight, Jana R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod B. Makhigani, Keith E. McDade, Michael P. McKenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan Fredrik Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, September 2005.
- [45] M. H. Maurer and W. Kuschinsky. *Handbook of Neurochemistry and Molecular Neurobiology*, chapter 7.1, pages 737–769. Springer US, January 2007.

- [46] Klaas Mensaert, Simon Denil, Geert Trooskers, Wim Van Criekinge, Oliver Thas, and Tim De Meyer. Next-Generation technologies and data analytical approaches for epigenomics. *Environmental and molecular mutagenesis*, 55(3):155–170, April 2014.
- [47] Karin B Michels, Alexandra M Binder, Sarah Dedeurwaerder, Charles B Epstein, John M Greally, Ivo Gut, E Andres Houseman, Benedetta Izzi, Karl T Kelsey, Alexander Meissner, Aleksandar Milosavljevic, Kimberly D Siegmund, Christoph Bock, and Rafael A Irizarry. Recommendations for the design and analysis of epigenome-wide association studies. *Nature Methods*, 10(10):949–955, October 2013.
- [48] R. M. Myers, Z. Larin, and T. Maniatis. Detection of single base substitutions by ribonuclease cleavage at mismatches in RNA:DNA duplexes. *Science*, 230(4731):1242–1246, December 1985.
- [49] Samuel Myllykangas, Jason Buenrostro, and Hannlee P. Ji. *Overview of sequencing technology platforms*, chapter 2, pages 11–25. Springer US, September 2012.
- [50] Colm Nestor, Alexey Ruzov, Richard R. Meehan, and Donncha S. Dunican. Enzymatic approaches and bisulfite sequencing cannot distinguish between 5-methylcytosine and 5-hydroxymethylcytosine. *Biotechniques*, 48(4):317–319, April 2010.
- [51] Ge Nong, Sen Zhang, and Wai Hong Chan. Two efficient algorithms for linear time suffix array construction. *Computers, IEEE Transactions*, 60(10):1471–1484, October 2011.

- [52] Michael C. O'Donovan, Peter J. Oefner, Stacy C. Roberts, Jehannine Austin, Bastiaan Hoogendoorn, Carol Guy, Graham Speight, Meena Upadhyaya, Steve S. Sommer, and Peter McGuffin. Blind analysis of denaturing high-performance liquid chromatography as a tool for mutation detection. *Genomics*, 52(1):44–49, August 1998.
- [53] M. Orita, H. Iwahana, H. Kanazawa, K. Hayashi, and T. Sekiya. Detection of polymorphisms of human DNA by gel electrophoresis as single-strand conformation polymorphisms. *Proceedings of the National Academy of Sciences*, 86(8):2766–2770, April 1989.
- [54] Mattia Pelizzola and Joseph R. Ecker. The DNA methylome. *FEBS Letters*, 585(13):1994–2000, July 2011.
- [55] Gerd P Pfeifer, Swati Kadam, and Seung-Gi Jin. 5-hydroxymethylcytosine and its potential roles in development and cancer. *Epigenetics Chromatin*, 6(1):10, May 2013.
- [56] Michael A Quail, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and Yong Gu. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequences. *BMC Genomics*, 13(1):341, July 2012.
- [57] Beata Reiz, Attila Kertesz-Farkas, Sandor Pongor, and Michael P. Myers. Data preprocessing and filtering in mass spectrometry based proteomics. *Current Bioinformatics*, 7(2):212–220, June 2012.

- [58] Keith D. Robertson. DNA methylation and human disease. *Nature Reviews Genetics*, 6(8):597–610, August 2005.
- [59] Yutaka Saito, Junko Tsuji, and Toutai Mituyama. Bisulfighter: accurate detection of methylated cytosines and differentially methylated regions. *Nucleic Acids Research*, 42(6):e45, April 2014.
- [60] F. Sanger, S. Nicklen, and A.R. Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, December 1977.
- [61] Serge Saxonov, Paul Berg, and Douglas L. Brutlag. A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters. *Proceedings of the National Academy of Sciences*, 103(5):1412–1417, January 2006.
- [62] Enrico Siragusa, David Weese, and Knut Reinert. Fast and accurate read mapping with approximate seeds and multiple backtracking. *Nucleic Acids Research*, 41(7):e78, January 2013.
- [63] Patrick Stanssens, Marc Zabeau, Geert Meersseman, Gwen Remes, Yannick Gansemans, Niels Storm, Ralf Hartmer, Christiane Honisch, Charles P. Rodi, Sebastian Böcker, and Dirk van den Boom. High-throughput MALDI-TOF discovery of genomic sequence polymorphisms. *Genome Research*, 14(1):126–133, January 2004.
- [64] Ruimin Sun, Xiang Gao, Nanyu Han, Qiong Wu, Yuguang Mu, Kai Tang, and Xin Chen. Accurate detection of SNPs using base-specific cleavage and mass

- spectrometry. In *BIBM'12*, pages 1–4, Philadelphia, PA, US, October 2012. IEEE Computer Society.
- [65] Ruimin Sun, Ye Tian, and Xin Chen. TAMEBS: a sensitive bisulfite-sequencing read mapping tool for DNA methylation analysis. In *BIBM'14*, Belfast, UK, November 2014. IEEE Computer Society.
- [66] Chandana Tennakoon, Rikky W. Purbojati, and Wing-Kin Sung. BatMis: a fast algorithm for k-mismatch mapping. *Bioinformatics*, 28(16):2122–2128, August 2012.
- [67] Bosong Xiang and Mindy Prado. An accurate and clean calibration method for MALDI-MS. *Journal of Biomolecular Techniques*, 21(3):116–119, September 2010.
- [68] Xiaoyu Zhang, Junshi Yazaki, Ambika Sundaresan, Shawn Cokus, Simon W.-L. Chan, Huaming Chen, Ian R. Henderson, Paul Shinn, Matteo Pellegrini, and Steve E. Jacobsen. Genome-wide high-resolution mapping and functional analysis of DNA methylation in Arabidopsis. *Cell*, 126(22):1189–1201, September 2006.

My Publications

1. Ruimin Sun, Ye Tian and Xin Chen, TAMeBS:a sensitive bisulfite-sequencing read mapping tool for DNA methylation analysis. *Accepted by IEEE International Conference on Bioinformatics and Biomedicine 2014.*
2. Ruimin Sun, Xiang Gao, Nanyu Han, Qiong Wu, Yuguang Mu, Kai Tang and Xin Chen, Accurate detection of SNPs using base-specific cleavage and mass spectrometry. *Published in Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on 4-7 October. Pages: 1-4.*
3. Xin Chen, Qiong Wu, Ruimin Sun and Luoxin Zhang, Two combinatorial optimization problems for SNP discovery using base-specific cleavage and mass spectrometry. *BMC Systems Biology*, 6(2):S5, 2012.
4. Xin Chen, Ruimin Sun and Jiadong Yu, Approximating the double-cut-and-join distance between unsigned genomes. *BMC Bioinformatics*, 12(9):S17, 2011.