ACCURATE ALIGNMENT OF SEQUENCING READS FROM VARIOUS GENOMIC ORIGINS

LIM JING QUAN

NATIONAL UNIVERSITY OF SINGAPORE

2014

ACCURATE ALIGNMENT OF SEQUENCING READS FROM VARIOUS GENOMIC ORIGINS

LIM JING QUAN

(B.CompSc.(Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information that have been used in the thesis.

This thesis has not been submitted for any degree in any university previously.

Lim Jing Quan
18/July/2014

Acknowledgements

I thank my thesis supervisor Dr Sung Wing-Kin for his impeccable patience, selfless guidance and sharing of his invaluable knowledge over the course of my candidature. I am also glad to have Prof. Wong Lim Soon and Prof. Tan Kian Lee to be my thesis advisory committee members. I am thankful to Dr Wei Chia-Lin, Dr Li Guoliang, Dr Eleanor Wong and Dr Chandana Tennakoon for successful collaboration on some of the projects, which I have worked on and have eventually made up parts of this thesis. I would also like to thank Dr Teh Bin Tean, Dr Lim Weng Khong, Sanjanaa and Saranya from Duke-NUS graduate medical school for accommodating me while I was still working on this thesis.

The pursuit for knowledge over these years has not been a bed of roses for me. There was a point of time when I had wanted to quit my candidature. I am grateful that I have still managed to turn back, pull through and reach ‘this’ particular point of the thesis. To my comrades whom have made the lab an enjoyable place to work in, I thank you all in no particular order of favor or seniority: Sucheendra, Chuan Hock, Javad, Hugo Willy, Hoang, Zhizhuo, Xueliang, Chandana, Rikky, Gao Song, Peiyong, Ruijie, Narmada, Liu Bing, Difeng, Tsung Han, Benjamin G., Wang Yue, Michal, Wilson, Hufeng, Chern Han, Mengyuan, Kevin L., Alireza, Ramanathan and Ratul for inspiration and for contributing to the finishing of this thesis in various ways.

Finally, I would like to thank my family and Chu Ying for their patience. Once again, I thank all of you for keeping me aspired and hopeful towards the end of my candidature.

Contents

Summary

Sequencing technologies have revolutionized the study of genomes by generating high throughput data for various studies which are not cost-efficient when done with Sanger sequencing. The first step in analyzing these high throughput data is often to find the original location from which the data reads are sequenced from a reference genome. Moreover, references genomes can be very large (human genome ~3.2GB). This calls for better methodologies in aligning reads onto a reference genome.

In this thesis, we present three methodologies in producing accurate alignments of DNA-sequencing reads with bisulfite-induced nucleotide conversion, DNA-sequencing reads with mismatches and gaps, and RNA-sequencing reads with intronic spliced junctions.

Our first contribution is BatMeth; a fast, sensitive and accurate aligner for DNA-sequencing reads derived from sodium bisulfite treatment. BatMeth is designed to handle both base-space and color-space bisulfite-treated reads. Based on List-Filtering, Mismatch-Stage-Filtering, BatMeth was able to avoid examining spurious hits and improve the efficiency and specificity of our alignment. Our experiments also show that BatMeth can produce better methylation callings across samples of different bisulfite conversion rates.

BatAlign is our next contribution which can align DNA-sequencing reads in the presence of both mismatches and insert-delete (indel) accurately. Two novel

strategies called Reverse-Alignment and Deep-Scan are developed to enable the efficient reporting of accurate alignments for these reads. Reverse-Alignment starts the alignment of a read by looking for the most probable preliminary alignments incrementally. Deep-Scan refines the preliminary alignments by searching for a targeted subset of less probable alignments to better distinguish the best alignment from the rest. BatAlign was able to achieve competitive runtime efficiency with SIMD-enabled Smith-Waterman algorithm for the extension of seeds from a long read in our seed-and-extend strategy.

Our last contribution is BatRNA is designed to recover splice alignment of a RNA-sequencing read sensitively and efficiently. As RNA-sequencing datasets can have very varying mixture of exonic and spliced reads in them, BatAlign was introduced in BatRNA as a pre-mapping tool to draft up the possible spliced sites of the genome. After which, we filtrate the reads from the mappings of BatAlign to be mapped by BatRNA for possible spliced alignments of the reads. The resultant mappings from both BatAlign and BatRNA are considered for the final alignment of a read. Compared with other popular and recent RNA-sequencing aligners, BatRNA was able to produce very sensitive and accurate alignments in a dataset of mixed exonic and spliced reads, while maintaining competitive runtimes.

In summary, we have developed various methodologies to align reads on to a reference genome, sequenced from various genomic origins, accurately and sensitively.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Earth has been brimming with life for as long as we can remember. Being intellectually revolutionized agents, it is imminent for us to question and understand the ravels of life. As some may have known it as “The Code of Life”, DNA has been understood to be the determinant material that guides the molecular operations and propagation of organisms. Pioneers such as Charles Darwin and Gregor Mendel first studied the rules of such propagates between the years of 1856 and 1865.

In 1859, Charles Darwin published his theory of evolution with inspiring evidences in a book titled “On the Origin of Species” [1]. He showed that all species of life have descended from common ancestors and rejected competing explanations of species being transmuted from one and another. This scientific theory proposed a branching pattern of evolution for different species resulted from a process, which he has coined as Natural Selection. While the theory of Natural Selection was centered on the communal pressure for survival in an ecosystem, Gregor Mendel focused on the passing of phenotypes from parents to its offsprings of the same species. Mendel's experiments on plant hybridization led to the understandings on how the propagation of dominant and recessive phenotypes in a species was carried out in the form of inheritable materials [2], which we now call it as genes. It was not until 1940s that Darwin's theory of Natural Selection and Mendel's Law of Inheritance were combined to give rise to evolutionary biology.

DNA is made up of genes, which gave phenotypic traits to an organism. It was first isolated as a weak acid and was identified as the genetic material in 1944 by Oswald Avery, Colin MacLeod and Maclyn McCarty [3]. Within the next decade, science celebrated the ground-breaking discovery on the structure of DNA with the publication of three papers by Nature: one from James Watson and Francis Crick of Cambridge University that proposed the double helix sugar-phosphate backbone structure of the DNA [4], and two accompanying papers from Franklin Rosalind [5] and Maurice Wilkins [6] of King's College, London, who used X-ray diffraction images to support the helical structure of DNA.

After the DNA double helix structure was discovered, scientists moved on to investigate the contents of what it holds, in particular, the sequences of nucleotides that form genes. DNA was sequenced for the first time in early 1970s by Frederick Sanger [7], Walter Gilbert and Allan Maxam [8], and were published independently in 1977. Sanger sequencing was the first established method to sequence long stretches of DNA and had partially been used to produce the first draft of the human genome, known as the Human Genome Project (HGP), starting from 1990 and to its completion in 2003 with a working draft of the human genome [9].

Due to the influx of funding and talent into the field of genomics, huge advances in sequencing technologies were achieved and also gave rise to a new generation of sequencing technologies which we call second-generation sequencing (SGS) technologies.

With SGS technologies at the disposal of scientists, landmark projects were launched. After the HGP, scientists went on to sequence the genomic sequences of a wide variety of species from various clades such as mammal, nematode and insect. Some examples included humans of different ethnic groups and different strains of influenza viruses. Alongside with DNA sequencing projects, Human Encyclopedia of DNA Elements (ENCODE) project was also launched in 2003 to build a comprehensive list of functional elements of the human genome. ENCODE projects encompassed the studies of genetic elements that acted at the RNA level, protein level, and regulatory elements that control cellular functions[10]. As of 2012, ENCODE had claimed to have assigned biochemical functions for 80% of the human genome [11].

1.2 History of DNA Sequencing

1.2.1 First-Generation sequencing

DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. In 1977, the first whole DNA sequence was obtained, from the entire genome of bacteriophage Φ–X174, using chain-termination methods [12]. This sequencing method was developed in 1975 by Sanger [13] and followed independently by Maxam and Gilbert in 1977. The Maxam-Gilbert method was more laborious and hazardous to handle, as the chemicals used in the sequencing procedures were more radioactive than Sanger's method. Due to these reasons, Sanger sequencing became dominant and was representative of first-generation sequencing methods. Even till now, Sanger sequencing is still practiced due to the longer read-lengths, ~800 bases in average, that it can generate as compared to ~100 bases long reads from Illumina GA IIx machines [14]. Sample preparation for Sanger sequencing starts by generating randomly sized fragments from the same DNA fragment. The ends of these differently sized fragments are then labeled respectively with one of the four fluorescent dyes which substitutes for each of the four nucleotides of the DNA – adenine, cytosine, guanine and thymine. Next, the dye-ended fragments are ran across an agarose gel and will be separated by their lengths. Lastly, the sequence of the DNA sample is determined from the last base of the fragments as depicted by the order of their relative positions in the gel. Although, this method can be fully automated to sequence long stretches of DNA, it still took about 13 years and three billion dollars to produce the first working draft of the human genome for the HGP. The main drawback of Sanger sequencing is that the throughput of each run is too low to perform in-depth studies on the complex dynamics of the human genome.

1.2.2 Second-Generation sequencing

This wave of technologies aimed to offer numerous advantages over Sanger sequencing in the form of (1) shorter runtime (increasing sequencing speed); (2) higher throughput (sequencing more bases within shorter periods of time); (3) cheaper sequencing costs (less reagents were needed for the experiments) and (4) higher accuracy (enabling discovery of rare-occurring variants).

The second generation of sequencing (SGS) was first described by two publications in 2005 [15, 16]. The initial impacts that polony sequencing had brought about was the lower sequencing costs and the potential for scientists to capture the complex dynamics of the genome at high resolutions. A year later, two Cambridge scientists developed the Solexa 1G sequencer and it was able to produce a throughput of 1 giga-base in a single experimental run for the first time in history using reversible terminator chemistry [17]. In the same year of 2006, Agencourt was purchased by Applied Biosystems which introduced SOLiD sequencing [18] which too had the ability to sequence a genome as complex as the human genome. Other SGS technologies include Roche 454 pyrosequencing [19], IonTorrent semiconductor sequencing [20], DNA nanoball sequencing [21] and Heliscope single molecule sequencing [22]. With most SGS technologies, strands of identical DNA were anchored to a fixed location to be read by a sequential series of label-scan-wash cycles. Each of this cycle will yield a read-base and will no longer continue when the series of label-scan-wash cycles fall below a threshold of quality. Due to the high density of DNA that can be packed into a single sequencing template platform, the throughput from such technologies far exceeded of those of Sanger sequencing [14]. This has directly made quantification of transcripts, genome-wide methylation profiling and many other studies possible.

More cost-effective methods were also developed to compromise between the competing goals of genome-wide coverage and cost-effective targeted-coverage. An example will be “exome sequencing” whereby ~1% of the human protein-coding genome was targeted for sequencing [23, 24].

1.2.3 Third-Generation sequencing

Sanger sequencing and SGS technologies have by far revolutionized the field of genomics. However, there are still aspects of genome biology that are still beyond the capabilities of SGS technologies. The main shortcomings of SGS technologies are the long runtime (a few days), short read-lengths and potentially high sequence bias and/or sequencing errors. The large number of label-scan-wash cycles required to generate a read has to be synchronized and a lot of overhead has resulted before the next subsequent cycle can start. This caused the time needed to generate viable reads of long read-lengths to be long. It is also due to the fact that the label-scan-wash cycles have to be synchronized in-between cycles. This means that the yield of each step of the series of cycles will be <100% as the cycles can get "dephased" and out-of-synchronization which produce erroneous reads. As such, this causes an increase in sequencing errors as the read elongates during sequencing. This "dephasing" problem also caused the average read lengths generated by SGS technologies to be generally less than the lengths achieved by Sanger sequencing. Another source of read errors comes in the form of sequencing bias that results from PCR amplification [25] as a direct consequence from an intermediate step in SGS technologies. In view of these shortcomings of SGS technologies, single-molecule sequencing (SMS) technology is being developed for the scientific community.

Unlike sequencing-by-synthesis (SBS) technologies in SGS, SMS interrogates single molecules of DNA, using SBS too, in an asynchrony manner. In this manner, tens of thousands of reads can be sequenced within hours as compared to days as needed with SGS technologies. In addition, since molecules are interrogated individually, there is no need to amplify the DNA sample prior to sequencing by SGS technologies. This eliminates amplification bias or defects that may be introduced by PCR amplification. The nucleotides used in SGS technologies are usually 'color-coded' with a dye and this makes them different from the natural-occurring nucleotides that made up DNA. This chemical bias is further removed in SMS technologies and the reagent used to replace the dyed nucleotides is none other than DNA polymerase itself that is responsible for DNA synthesis.

The main idea in SMS comes from the tangible measurements that can be measured when the new DNA fragment is synthesized upon the template fragment. The measurements can then be interpreted as an ordered sequence of nucleotides. Some technologies of this new generation are based on but are not limited to the use of nanopores, tunneling currents during DNA synthesis, mass spectrometry, micro-fluidic chips and electron microscopes.

1.3 Motivation

1.3.1 Looking at the DNA with an intent

Little was known about the functions of DNA when it was identified as the genetic materials of organisms in 1944. It was also unclear on how DNA polymorphisms played a part in the molecular functions of the cells in an organism. It was not until 1956 when Vernon Ingram successfully associated a single amino acid substitution with Sickle Cell Disease [26]. Since then, scientists have moved forward with the intention to better understand diseases caused by genetic variations and to discover new ways to treat them [27]. Other common genetic diseases include Cystic Fibrosis, Glucose-6 Phosphate Dehydrogenase deficiency and Color Blindness.

Genetic diseases were first thought to be direct causal of mutations in the DNA. This thought could not be more wrong. The transcription of DNA to RNA and translation of RNA to proteins can also be affected by the products of its own processes. The study on the causal effects of the DNA and its products other than the changes in the underlying sequence is now termed as epigenetics [28]. The two epigenetic modifications to the genome are histone modifications and DNA methylation [29].

Regardless of genomic or epigenetic factors, the important challenge is to understand the mechanisms that control the expression of genes in a genome. By learning about these processes, we can uncover more ways to treat, cure or even prevent adverse phenotypes from a diseased genome.

1.4 General workflow on sequencing reads

Due to the limitations of technology, the sequenced reads is almost always shorter than genomes that are to be sequenced. As such, from the raw sequenced reads of a sample to downstream analysis in the dry-lab, the common step in most processing pipelines is to map the sequenced reads onto a reference genome. In the field of genomics, the two most front-line computational tasks are 1) mapping reads back onto a reference genome and 2) de novo or guided assembly of the genome with sequenced reads to produce a reference for sequenced reads to be mapped on. These tasks are generally the most computationally intensive tasks in the pipelines and the problem is made worse by the voluminous amount of data that needs to be processed (~600 Gb in a single run).

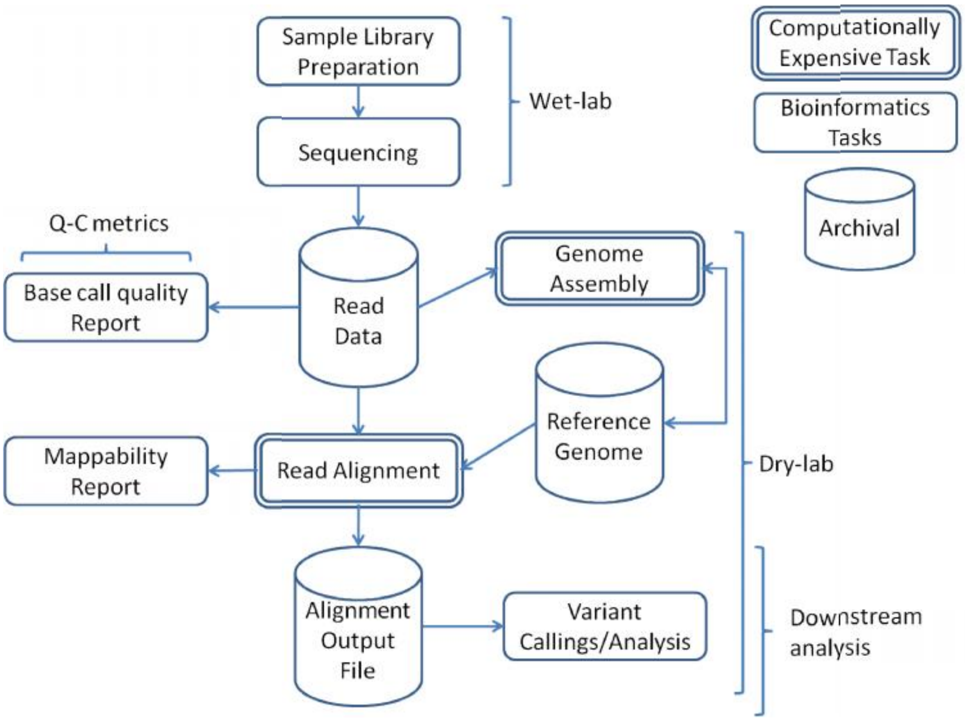


Figure 1.1. General workflow on sequencing reads

1.5 The mapping challenge

One can see the problem of mapping a read onto a reference genome as a computational problem of matching a short query string to a large database of reference text. The objective is to find the original location from where the read was supposed to originate from the reference genome. The challenges of alignment of SGS reads are composed of different error profiles of sequenced reads from different sequencing technologies, short read lengths (reads from SGS can be ~36 bases long), large reference length which the reads need to be mapped on and the voluminous data that are generated from SGS machines [30]. Since mapping the reads is a prelude to many downstream analyses such as and not restricted to variant-callings, quantification of rare transcripts and annotation of epigenetic factors on the DNA, it is important to map the sequenced reads with high sensitivity, specificity and speed.

Many scientists and classic software have tackled this problem. For instance, BLAST [31] (~50k citation count) and BLAT [32] have shown the demand and impact of bioinformatics in the understanding of genomic data. However, classic legacy software cannot handle SGS well as they are initially not designed for SGS reads. Therefore, new methods have to be developed to handle SGS reads. This thesis aims to report on new algorithms that we have developed to align SGS reads with high sensitivity, specificity and speed.

1.6 Contribution of thesis

The first contribution of this thesis was the development of BatMeth, which is a fast and efficient algorithm for the alignment of bisulfite-treated DNA sequencing reads back onto a genome allowing mismatches. BatMeth is based an exact algorithm, namely BatMis, for the alignment of reads onto a genome allowing mismatches. Preliminary hits to a read are not aligned but counted from the FM-index representation of the reference genome to which the reads are being mapped. By designing the appropriate heuristics, BatMeth has shown to be an improved aligner in our benchmarks. In addition, it was also shown to have less bias when mapping bisulfite treated samples across a wide range of bisulfite conversion rates on both Illumina base-space and SOLiD color space reads.

The second contribution was the development of BatAlign for the accurate alignments of DNA sequencing reads by allowing both mismatches and indels. The algorithm of BatAlign was designed to discriminate between polymorphisms and sequencing errors with high precision. The main novelty is that a long seed (75 bp) is used to find hits of increasing alignment cost incrementally, while allowing at most 5 mismatches and 1 gap at the same time, with an exact solution. The initial method for aligning short reads (~100 bases) was also extended to handle longer reads of (150-250 bases). I have designed all the other main algorithmic components of BatAlign. In addition, I have also designed and performed almost all the experiments found in the paper describing BatAlign (Guan Peiyong helped out by supplying me with a RSVsim-rearranged reference genome for one part of the experiments). This is a joint work with Dr Chandana Tennakoon. Dr Chandana developed the data structures used to pair reads efficiently and the function needed to calculate the mapping quality of resultant alignments. BatAlign was benchmarked on a wide class of simulated and real reads and have shown to be more accurate than other popular aligners in terms of mapping accuracy on published PCR-validated structural variation in hepatocellular carcinoma.

The third contribution was the development of BatRNA for the alignment of reads while allowing mismatches, indels and large intronic gaps to be present in a single read too. The main novelty of BatRNA is that it is a hybrid method between exon-first and split-and-extend approaches. This is also a joint work with Dr Chandana Tennakoon. Dr Chandana has designed the recursive step of the BatRNA algorithm to look for putative seeding alignments of the query read. However, this recursive step was biased towards introducing intronic gaps into the alignment of RNA-seq reads and is very computationally expensive to perform. I have performed the empirical analytical study on picking the appropriate seed-length and mismatch-number to be used with the recursive algorithm. I have also extended his algorithm to selectively realign, by using a deterministic quality filter, the preliminary alignments reported by BatAlign to undergo the recursive splicing algorithm to obtain the final spliced alignment of reads spanning across exon-exon boundaries. Benchmarks showed that BatRNA gives sensitive and accurate mappings in a mixed sample of exonic and spliced reads across varying read lengths.

In summary, we have developed three novel alignment algorithms on improved data structures for the efficient and accurate mappings of sequencing reads from various genomic contexts. BatMeth was published in *Genome Biology* (I as first author, Impact Factor: 10.5) and BatAlign (accepted and in press) will appear in *Nuclei Acid Research* (I as first author, Impact Factor: 8.808).

1.7 Organization of the thesis

The remaining contents of the thesis are organized as follows. Chapter 2 presents a preliminary of biological background and survey of SGS technologies required for the proper understandings of the thesis. Chapter 3 will present the survey of bisulfite-treated DNA-seq aligners, gapped DNA-seq aligners and spliced RNA-seq aligners in their respective subsections. Chapters 4-6 describe our algorithms for the improved alignment of bisulfite-treated DNA-seq reads, gapped DNA-seq reads and spliced RNA-seq reads respectively. Chapter 7, the last chapter, will conclude the thesis with a summary of all the presented work and a brief discussion on the possible future developments which still can be carried out on alignment algorithms.

Chapter 2

Basic Biology and Sequencing Technologies

2.1 Basic Biology

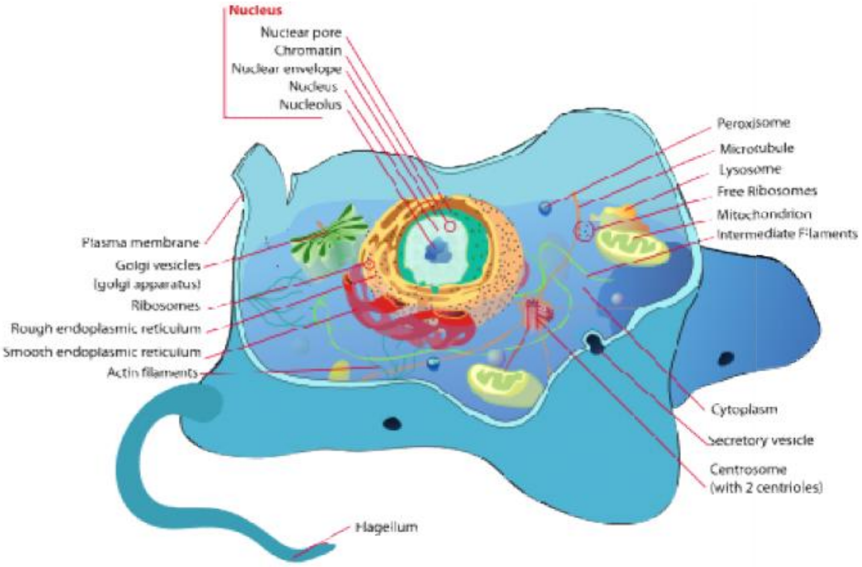


Figure 2.1. Schematic diagram of a typical animal cell

In this chapter, we present the background knowledge on molecular biology and describe some of the SGS technologies that are widely used today. We also describe the types of reads, which will be mapped onto the reference genome by aligners.

Cells are the building blocks of organisms and complex organisms and an adult human can be made up of approximately 300 trillions of cells. Cells are also referred to be the building blocks of life as they are essential to maintain the bodily functions of organisms. On a cellular level, a cell has a cell membrane, nucleus, golgi apparatus, cytoplasm and mitochondrion as drawn in Figure 2.1. On a macro-molecular scale, it typically contains carbohydrates, amino acids, lipids and nucleic acids. With the advancements made in the wet-lab experiments, studies can now be carried out to study the activities of the various macromolecules in the cells. For instance, genome-wide gene expression can be analyzed using high throughput methods such as RNA-seq data [33], spliced alignment tools [34-41] and transcripts-isoforms quantification tools [42-46]. Repetitive regions of the genome can be hard to study with SGS data, as alignment tools will not be able to report the putative original location of the read in the genome with high confidence for uniqueness. These repetitive regions include the telomeres and centromeres of the human genome and alternative methods such as florescence immuno-staining techniques can be used to study these repetitive genomic regions [47].

To first study the genomes using SGS high-throughput data, scientists have to use sequencing machines to 'read' out the genomic sequences of the prepared sample. Many of these SGS sequencing technologies come from Illumina, Life Technologies, Roche and Ion Torrent. Depending on the types of samples, methods of preparation for the experiments and the sequencing technologies being used, a wide range of analysis can be carried out. Figure 2.2 shows some of the analysis that can be carried out on SGS sequencing data. This phase is the first step in identifying and understanding the dynamics of macromolecules in a cell.

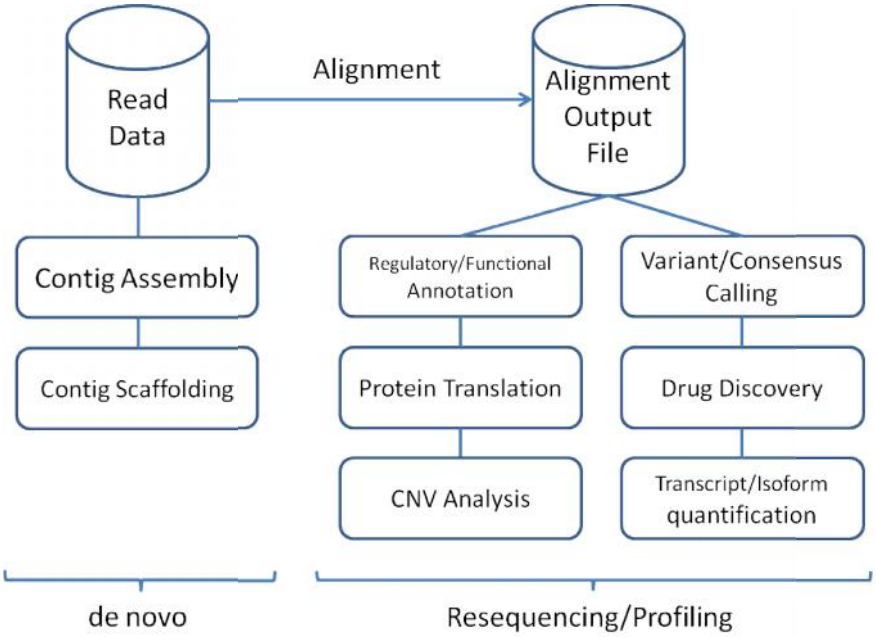


Figure 2.2. Two main types of genomic tasks and their respective downstream analysis. De novo tasks involve the manipulation of read data without a reference genome. Profiling tasks use the alignment of the read on a reference for analysis.

2.2 Central Dogma of Molecular Biology

The Central Dogma of Molecular Biology is one of the main principles in molecular biology. It states that the transfer of genetic information is from the gene sequences of the DNA to the proteins, which carries out various cellular functions. Although there are exceptions pointing against it, it is still widely accepted in the community of molecular biology. Figure 2.3 depicts the general passing of sequential information between genetic materials as stated by the general cases in the Central Dogma of Molecular Biology as formulated in 1970 [48].

The central dogma states that the DNA molecules encode all genetic information. The genetic information can be visualized as linear sequences of nucleotides in the cells. When cells grow and divide, genetic information is transmitted from the parent cell to the daughter cells by replication. This process creates a duplicate of the DNA molecule during the synthesis phase of the cell cycle. During the synthesis of messenger ribonucleic acid (mRNA), part of the original DNA sequence acts as a template for the mRNA sequence to be synthesized on. This process of synthesizing mRNA is known as transcription.

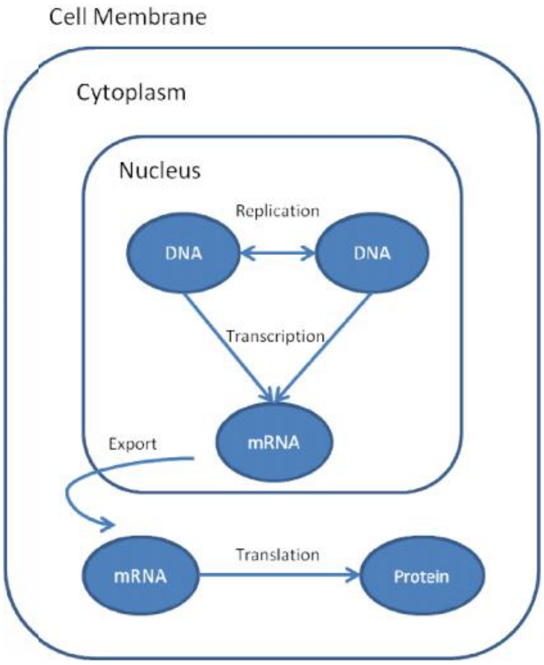


Figure 2.3. The general cases of the central dogma of molecular biology for eukaryotic cells

For eukaryotic cells, the mRNA molecules are then transported out of the nucleus, into the cytoplasm of the cell, where consecutive triplets of nucleotides are read as codons by protein complexes known as ribosomes. In the ribosome-mRNA complex, aminoacylated transfer-RNAs (tRNA) are recruited and used to link the amino acids that form protein polypeptide chains. The process of reading mRNA and forming protein complex is known as translation. This protein polypeptide chains will undergo post-translation modifications to a stable folded 3D structure that will contribute to its functions. These 3D protein structures will then drive the various cellular functions in organisms.

From the central dogma, DNA-DNA replication, DNA-RNA transcription and RNA-Protein translation are the main processes that describe the transfer of genetic information from one medium to the other. In addition, we can also see that there are several ways in which cells can be regulated. For instance, the amount of mRNA transcribed from the DNA, known as gene expression level, will be translated into varying concentrations of proteins which, in turn, will up/down-regulate transcription; affecting levels of gene expression and subsequently their corresponding protein concentration levels in the form of a feedback loop. Post-modifications to the proteins such as phosphorylation and acylation can also affect the functional properties of proteins. By mutating the DNA sequences, changing the levels of mRNA and protein abundances can lead to the onset of diseases such as Sickle-cell anemia and Cystic Fibrosis.

In Appendix A, we will describe replication, transcription and translation in detail.

2.3 Next Generation Sequencing Technologies

Chapter 1 gave a brief history of sequencing technologies and the motivation to uncover insights that genomic sequences can contain. In this section, we will briefly describe the computational challenges that these technologies have brought about and the main ideas behind some sequencing technologies that the thesis is focused on. Currently, sequencing technologies support sequencing materials from a wide range of starting materials, such as genomic DNA, PCR products, bacterial artificial genome (BAC) and complementary

DNA. Without loss of generality, we will describe the sequencing of genomic DNA in the following subsections by various sequencing technologies.

2.3.1 Roche/454 Sequencing

454 sequencing is arguably the first high throughput sequencing technology that was available to the market. This technology eradicated the need for DNA sample fragments to be cloned in bacterial hosts. By removing bacterial clonal copies of the DNA, we also remove any amplification bias, which may be introduced by the hosts into the DNA sample. Instead of in vivo cloning of the DNA sample using bacterial hosts, the amplification process is replaced by a more efficient in vitro DNA amplification method called emulsion PCR [49]. With emulsion PCR, fragmented DNA will attach to a streptavidin bead covered with adapter probes with bases complementary to that of the fragmented DNA. The ideal scenario will be that one fragment of DNA to ligase to one bead and be suspended in an emulsion so that individual beads will be trapped in micro-reactors for PCR-based amplification reactions. The whole emulsion of beads will be amplified in parallel to create millions of clonal copies of each DNA fragment on each bead. After amplification, the emulsion is removed from the mixture of beads as like removing the oil from an oil-and-water mixture. Finally, the beads will be loaded onto a picotiter plate prior to being sequenced by a sequencing machine [16].

The loaded picotiter plate will have hundreds of thousands of sequencing processes to be carried out in parallel, which directly obtain dramatic increase in sequencing throughput as compared to Sanger sequencing [50]. As sequencing takes place, a nucleotide is added one by one to the immobilized ligated template DNA on the bead. Whenever a complementary nucleotide is added to the template DNA, a chemiluminescent enzyme, present in the reaction mix, will produce a detectable light by releasing inorganic pyrophosphate [19, 51]. This is also why 454 sequencing is also known as pyrosequencing and SBS.

Since the detected light signal is directly proportional to the number of bases incorporated onto the template DNA in one sequencing cycle, pyro-sequenced reads will often wrongly sequence lengths of homo-polymeric nucleobases.

2.3.2 Ion Torrent Sequencing

Ion Torrent invented the first semiconductor-sequencing chip that was commercially available for the market. Similar to 454 sequencing, Ion Torrent clonally amplified DNA fragments by using emulsion PCR. After which, the beads with the amplified DNA materials will each sit inside a micro-well for PCR-based amplification reactions. The main difference between Ion Torrent sequencing and 454 pyro-sequencing is that the Ion Torrent's chip itself is the sequencing machine [52].

The sequencing of the DNA fragment starts by flooding the bead-loaded wells with one nucleotide after another sequentially. When the DNA fragment is extended by the incorporation of nucleotides, it releases hydrogen ions into the well and this changes the pH of the solution in the well. This chemical change of pH can be directly recorded by a sensor plate at the bottom of the well into voltage readings [53]. Since the chip directly detects the nucleotides, which are being synthesized onto the DNA template fragments in the wells, no external optical instruments are needed.

Although Ion Torrent sequencing is based on a different methodology from Roche-454 for sequencing genomic materials, its sequenced reads also have the problem of wrongly estimating lengths of homo-polymeric nucleobases [54]. This is due to the intensities of the produced voltages, being directly proportional to the number of bases incorporated, onto the template DNA in a single sequencing cycle.

2.3.3 Illumina/Solexa Sequencing

DNA molecules are first fragmented into varying lengths, through the use of a nebulizer, by sonication [55] or nebulization [56]. The subset of these randomly sized DNA fragments, with similar length, is then selected for sequencing. Illumina uses ‘bridge’ amplification reaction that occurs on the surface of the flow cell to sequence a DNA fragment [57]. The surface of the flow cell is coated with single stranded oligonucleotides as complementary probes that correspond to the priming adapters ligated to both ends of the DNA fragment. These single stranded oligonucleotides are bounded to the surface of the flow cell exposed to the reagents for polymerase-based extension. Priming occurs at the free end of the ligated fragments and ‘bridge’ over to a complementary oligonucleotide on the flow cell.

Repeated denaturation and extension result in localized amplification of single molecules in millions of unique locations spread across the flow cell. These unique locations are referred to as “cluster stations”. Figure 2.4 shows the bridge amplification of DNA fragments to obtain clusters of amplified DNA materials [58]. The flow cell, with millions of amplified clusters, is then loaded into the Solexa sequencer for sequential cycles of extension and imaging. The cycles of sequencing consists of the incorporation of fluorescent nucleotides and the imaging of the entire flow cell in a sequential synchronized manner. These images represent the respective base being synthesized at each individual location of the flow cell. Any laser signals, above background signal, will identify the physical location of a cluster. The fluorescent emission will then be used to identify the nucleotide base that was incorporated at that position. The cycle is then repeated, one base at a time, generating images that represents a single base extension in the cluster.

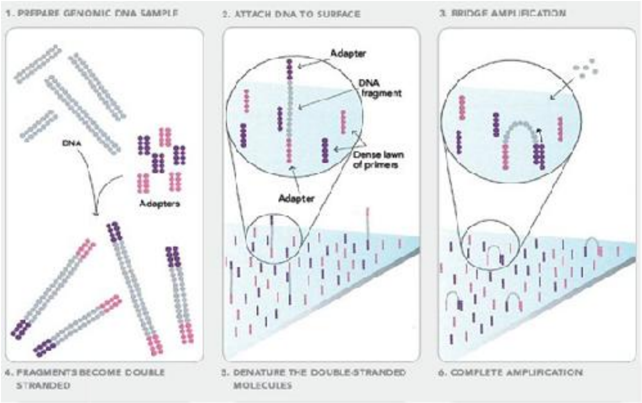

1. PREPARE GENOMIC DNA SAMPLE
2. ATTACH DNA TO SURFACE
3. BRIDGE AMPLIFICATION
DNA
Adapters
Adapter
DNA fragment
Dense lawn of primers
Adapter
4. FRAGMENTS BECOME DOUBLE STRANDED
5. DENATURE THE DOUBLE-STRANDED MOLECULES
6. COMPLETE AMPLIFICATION

Figure 2.4. Schematic diagram of bridge amplification forming cluster stations. Source: [58]

The result of sequencing a DNA fragment with Solexa will be a string of ATCG (‘N’ may be included as an ambiguous sequenced base), which is the read representation of a cluster on the flow cell.

2.3.4 ABI/SOLiD Sequencing

Similar to 454 and Ion Torrent sequencing, SOLiD also uses emulsion PCR that generates ‘bead’ clones. Each bead is then attached to the surface of a flow cells via 3’ modifications to the DNA strands. At this point, we will have a flow cell with millions of beads; each of a single genomic DNA template, with distinct adaptors on either ends being monitored simultaneously via sequential digital imaging.

Now, the interrogation of a nucleotide is no longer driven by polymerase but rather by ligating labeled oligonucleotides with the queried DNA template [15]. The technology is also radical, as each ligated oligonucleotide will degenerate the queried DNA strand at positions from 3 to 5, and one of the 16 specific dinucleotides at positions 1-2 from the 3’ end. Base positions from 6 till the 5’ end of the oligonucleotide are also degenerated and it will hold one of the four fluorescent dyes.

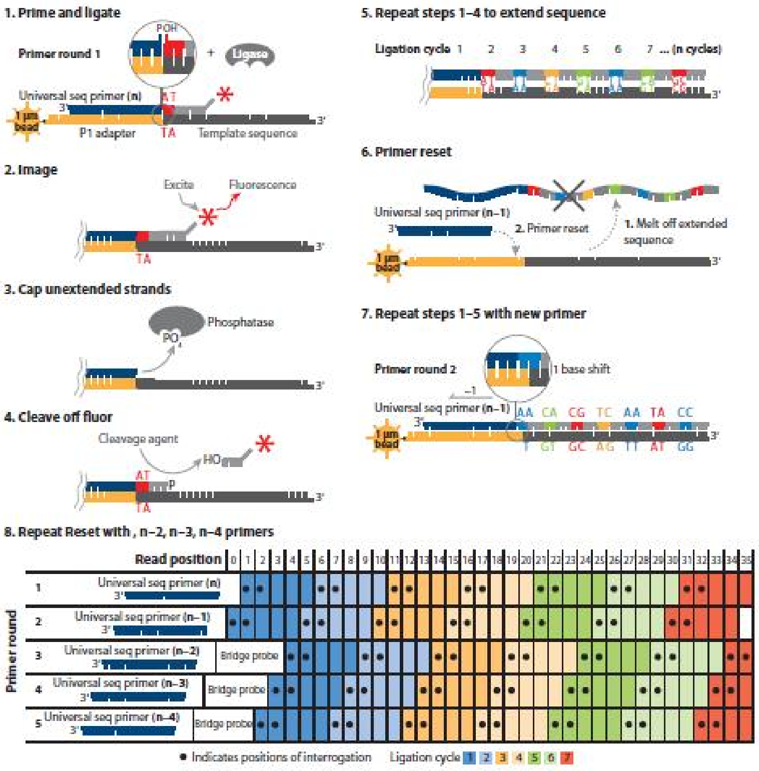

1. Prime and ligate
Primer round 1
POH
+
Ligase
Universal seq primer (n)
3'
AT
1 µm bead
P1 adapter
TA
Template sequence
3'
2. Image
Excite
Fluorescence
TA
3'
3. Cap unextended strands
Phosphatase
PO4
3'
4. Cleave off fluor
Cleavage agent
HO
AT
P
TA
3'
5. Repeat steps 1–4 to extend sequence
Ligation cycle 1 2 3 4 5 6 7 ... (n cycles)
3'
6. Primer reset
Universal seq primer (n–1)
3'
2. Primer reset
1. Melt off extended sequence
1 µm bead
3'
7. Repeat steps 1–5 with new primer
Primer round 2
1 base shift
–1
Universal seq primer (n–1)
3'
AA CA CG TC AA TA CC
T GT GC AG TT AT GG
1 µm bead
3'
8. Repeat Reset with , n–2, n–3, n–4 primers
Read position 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35
Primer round
1 Universal seq primer (n) 3'
2 Universal seq primer (n–1) 3'
3 Universal seq primer (n–2) 3' Bridge probe
4 Universal seq primer (n–3) 3' Bridge probe
5 Universal seq primer (n–4) 3' Bridge probe
● Indicates positions of interrogation
Ligation cycle 1 2 3 4 5 6 7

Figure 2.5. Workflow of ligase-mediated sequencing approach from ABi SOLiD. Source: [58]

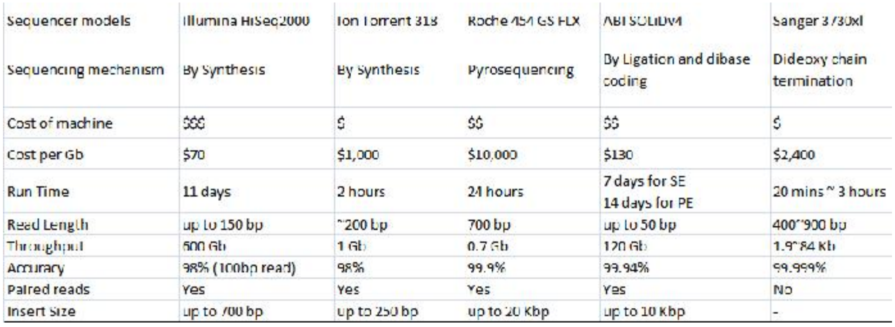
Figure 2.5 provides an overview of steps involved in sequencing DNA fragments using SOLiD sequencers [58]. The sequencers initially involve annealing a primer, hybridizing and ligating a mixture of fluorescent oligonucleotides of 8-mers whose 1st and 2nd 3' bases match that of the template. The unextended fragments are then capped with the same mixture of non-fluorescent probes. Following which, phosphatase treatment is applied to prevent out-of-phase ligation to allow timely detection of specific fluorescent dyes. After imaging, the dyes are removed via a two-step chemical cleavage of the three 5' bases, leaving behind a 5-base ligated probe, a 5' phosphate. These steps are repeated, this time, for querying the 6th and 7th bases. After ~10 cycles, a 'reset' of primers is initiated. The initial primer and all ligated parts of the template are melted and washed away. A new primer that is N-1 in length takes over and restarts the whole process of sequencing.

The method of sequencing by ligation used in SOLiD sequencers has been reported to have problem sequencing palindromic sequences [59].

2.3.5 Comparison

Having discussed some of the most popular sequencing technologies, we can infer that there is no best sequencer for all types of experiments. The type of sequencer used is largely dependent on the type of data, which the experimenters intend to collect and the budget allowed in these sequencing projects. Table 1 shows the specifications of some commercial sequencers, which are commonly used today [14, 60].

Table 2.1. Comparison between some commercialized sequencing platforms in the market.



Sequencer models	Illumina HiSeq2000	Ion Torrent 318	Roche 454 GS FLX	ABI SOLiDv4	Sanger 3730xl
Sequencing mechanism	By Synthesis	By Synthesis	Pyrosequencing	By Ligation and dibase coding	Dideoxy chain termination
Cost of machine	$$$	$	$$	$$	$
Cost per Gb	$70	$1,000	$10,000	$130	$2,400
Run Time	11 days	2 hours	24 hours	7 days for SE 14 days for PE	20 mins ~ 3 hours
Read Length	up to 150 bp	~200 bp	700 bp	up to 50 bp	400~900 bp
Throughput	600 Gb	1 Gb	0.7 Gb	120 Gb	1.9~84 Kb
Accuracy	98% (100bp read)	98%	99.9%	99.94%	99.999%
Paired reads	Yes	Yes	Yes	Yes	No
Insert Size	up to 700 bp	up to 250 bp	up to 20 Kbp	up to 10 Kbp	-

2.4 Origins and representations of sequenced data

Chapter 1 has outlined some of the challenges in aligning sequencing reads to a reference genome, which this thesis tackles. In this subsection, we will highlight some of the genomic materials, which are commonly being sequenced. In addition, we also describe the two main representations of sequenced reads. In doing so, we present an overview on how alignment-challenges can arise from various sequencing technologies and representations.

2.4.1 Whole-genome and targeted sequencing

High throughout sequencing technologies have successfully been used to sequence genome-wide data for the study of genomes in its entirety. Machines from Illumina and Roche have much higher throughput and shorter sequencing times as compared to Sanger sequencing. As such, they are almost always picked over Sanger sequencing to sequence whole genomes. The two main types of projects, which are performed with NGS, are de novo assemblies of whole genomes and the alignment of sequencing reads onto reference genomes. In the former type of projects, scientists construct a reference genome from the sequenced reads to allow the study of various disease-causing genomic features such as SNPs, indels, structural variants and epigenetic profiles to be performed on it. In the latter, the alignments of the sequenced reads onto a reference genome are used to uncover some of the mentioned disease-causing genomic features, which are of interest to the community. [61] showed that massively sequenced reads are able to reconstruct the mutational signatures on a genome-wide scale in gastric cancer samples.

However, whole-genome sequencing (WGS) is not preferred as the cost of reagents far exceeds the requirements of a study. For instance, it is not cost-efficient to use WGS to study only 10 single nucleotide positions (the human diploid genome has about 6G locations). Thus, targeted sequencing was developed to sequence only specific genomic regions-of-interest at much higher coverage. With high coverage, targeted sequencing can study mutations at higher resolution than with WGS but at lower cost. In addition to lower cost, the time used to sequence the sample is also reduced as compared to Sanger sequencing and WGS. [24] allowed the study of a rare Mendelian disease through targeted sequencing on a small population and the identification of the genes responsible for Millers syndrome. Examples of targeted sequencing are exome sequencing, amplicon sequencing and reduced representation bisulfite sequencing.

2.4.2 RNA-seq – mRNA

RNA-seq is used to create a profile of transcription levels of all genes in a genome called transcriptome [33]. The transcriptome is clinically important in genetic diagnosis as the functional consequences in a cell can be viewed as the abundance of transcript sequences in it. The transcriptome was first profiled from using Sanger sequencing on the DNA fragments that are complementary to mRNA fragments called Expressed Sequenced Tags (EST) [62]. However, due to the low throughput of Sanger sequencing, lowly transcribed genes may not be profiled. Since NGS can sequence genomic samples with high throughput, lowly expressed transcripts can also be profiled and be quantified directly to the number of mRNA fragments being sequenced. In addition, the cost of RNA-seq is also lower than EST sequencing as the required amount of RNA is less than what is needed for EST sequencing.

2.4.3 Epigenetic sequencing

Epigenetic is the study on the causal effects of the DNA and its products other than the changes in the underlying genomic sequences. The main types of epigenetic studies enabled by high throughput sequencing technologies are Chromatin-Immuno Precipitation sequencing (ChIP-seq) and methylation studies.

ChIP-seq is used to identify the protein-binding sites on the DNA [63]. This is important as it helps scientists to understand how DNA-protein interaction affects gene expression. ChIP-seq was preceded by ChIP-chip, which requires a pre-designed microarray. This made ChIP-chip susceptible to hybridization bias as microarrays come with a fixed number of probes. ChIP-seq lacks this form of bias as sequencing technologies can amplify all ChIP-enriched regions and can be applied to genome-wide discovery of transcription factors, structural proteins and DNA modifications.

DNA methylation is the process of adding a functional methyl group to the cytosine of the DNA [64]. Changes in DNA methylation influences the expression of genes in cells and also differentiated matured cells from embryonic stem cells. The methylation profiles, methylome, of differentiated cells from different tissues are vastly unique to one another. Knowing that sites of methylation in differentiated cells are specific and permanent, these cells are prevented to revert back to their pluri-potent state [65]. The current golden standard to produce a genome-wide methylome of a sample is to perform bisulfite treatment sequencing on the sample. Bisulfite treatment modifies the unmethylated cytosines to uracils and leaves methylated cytosines unchanged [66]. Upon subsequent PCR amplification of the bisulfite treated sample, uracils will be amplified as thymine on the + strand and adenine on the – strand; unmodified cytosines will be amplified as if DNA-DNA replication is taking place. By using NGS, each possible site of methylation can be surveyed at high resolution and gives a finer granularity of methylation rates at each site.

2.4.4 Base-space and color-space reads

Sequenced reads can be stored in various representations. The two most distinctive representations are base-space and color-space reads.

Base-space reads are stored as a string of characters consisting of “ACGTN”. This sequence of characters usually represents and can directly translate to the genomic sequence, which was being scanned by the sequencing machines. The character ‘N’ is to denote an ambiguous base in the read, which the sequencing machine cannot represent with any of the usual "ACGT" nucleotide characters with high confidence. Illumina/Solexa, Roche/454 and Life Technologies/Ion Torrent produce base-space reads.

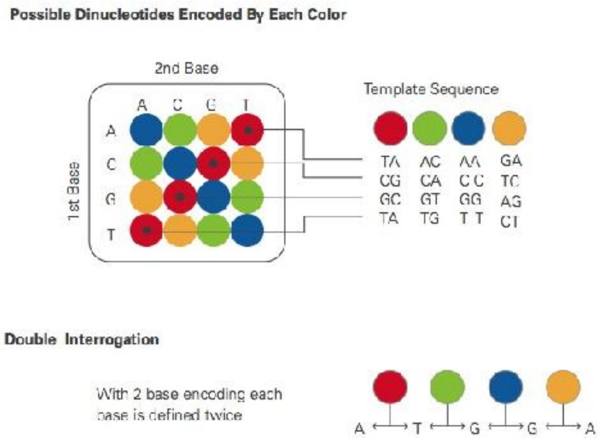


Figure 2.6. 2-base encoding scheme used by SOLiD sequencers. Source: [58]

For SOLiD, genomic bases are interrogated dinucleotide-ly, each color dye represents two adjacent genomic bases. Figure 2.6 shows how the 16 combinations of di-nucleotides are encoded by 4 color codes. An example color read will be "T000123100122331100". The first character, T, is the last base of the sequencing primer used and the numeric part of the string represents the transitions of genomic bases during the interrogation of the dinucleotides. To obtain the reverse-complement of a color read, we can simply reverse the numerical portion of the read and change the terminal base from 'T' to 'G'.

In addition, SNPs in a color read can be easily identified by two adjacent color mismatches in a read after being aligned to a reference genome. However, it is also this strength that is turned into a weakness for SNP-rich and bisulfite-treated data. Each base-letter mismatch will be represented by two adjacent color mismatches instead of a one-letter mismatch in other technologies. As there will more color-space mismatches in a color-space read than base-space mismatches in an equivalent base-space read, computational time dramatically increases when we map color reads with a higher mismatch number than with base-space reads.

2.4.5 Computational representation of data

Computers run on software that are compiled and executed as a string of 1s and 0s on the hardware level. Behind layers of abstraction, the data that we store in a computer are ordered strings of 1s and 0s. The smallest unit of storage in a computer is a bit, which can represent either a 1 or a 0. Data is often stored in pre-defined data structures, which are 4 bytes long (1 byte = 8 bits). A 4-bytes long data structure, holding 32 bits, can effectively express a large range of numbers. If we are to use units of 4-bytes structures to store each DNA nucleotide then this will put a lot of bits to waste. As DNA is comprised of only 4 unique characters, a 2-bit data structure is enough to represent a nucleotide uniquely with the 3 remaining nucleotides. In the literature, 2-bit encoding is used extensively to optimize the usage of space of genomic data.

In this thesis, we discuss mainly on aligning a read onto a reference genome with high accuracy and efficiency. To achieve this, we will build a 1-time index of our reference genome. The reference genome is represented as an FM-index [67]; an opportunistic data structure based on BWT [68] to optimize both space and time complexity in our alignment algorithms. It supports linear time complexity query operations, in terms of the length of the queried read. In the following chapter, we will review on the other alignments algorithms and the indexing techniques that they have employed in their respective methods.

Chapter 3

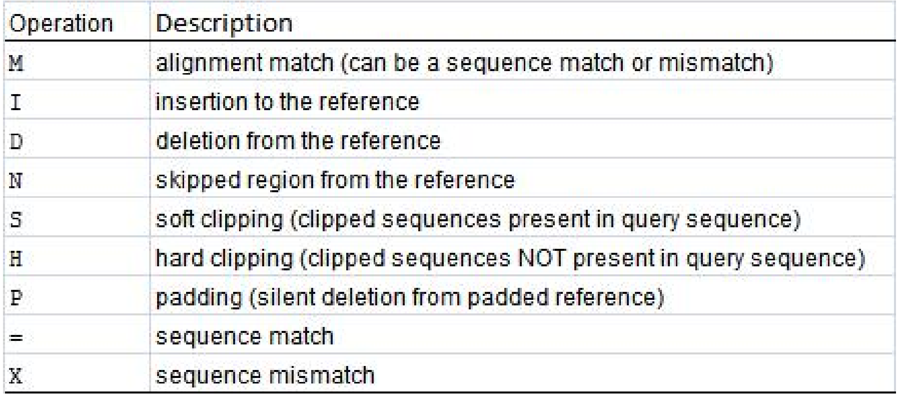
Survey of Alignment Methods

Alignment of genomic sequences has been tackled since the advent of sequencing technologies. Pioneering works, in the field of sequence alignment, such as Smith-Waterman [69] and Needleman-Wunsch [70] have guided the development of genomics through its infancy. However, as technologies advance, the amount of data and the types of sequencing reads, which can be generated, has increased dramatically. Therefore it is essential for methods to be able to align high volume of reads from various wet-lab/dry-lab origins accurately and efficiently. Hence, a salvo of alignment methods was designed to handle these reads.

3.1 Basics of Genomic Alignments

The aim of all genomic alignment algorithms is to map each query read to a reference genomic location in which it was originally sequenced from. Given a reference genome T and a read R, the alignment algorithm may report a list of putative genomic locations where R was sequenced from the reference genome. These putative locations are called hits or mappings. For each reported location on the reference genome that represents R, the aligner can also report the sequence of text-edit operations that transforms R into T. The text-edit operations are often stored as a string of characters as CIGAR string in a SAM formatted alignment file [71]. CIGAR strings are made up of the alphabet {M, I, D, N, S, H, P, =, X} and the details of each text-edit operation are described in Table 3.1.

Table 3.1. The possible text-edit operations that can be represented by a CIGAR for the alignment of a query string onto a reference text



Operation	Description
M	alignment match (can be a sequence match or mismatch)
I	insertion to the reference
D	deletion from the reference
N	skipped region from the reference
S	soft clipping (clipped sequences present in query sequence)
H	hard clipping (clipped sequences NOT present in query sequence)
P	padding (silent deletion from padded reference)
=	sequence match
X	sequence mismatch

Given that R can be transformed into T with a sequence of text-edit operations, we would ideally want to find the location in T such that the number of edit operations needed to transform R into T[loc .. loc + |R| + |gap|] is minimized. Naively, an aligner would want to align R onto T perfectly, with no mismatches or gaps between R and T. However, in the presence of polymorphisms and sequencing errors, it is uncommon for R to be mapped onto T perfectly. As such, a scoring function and scoring matrix can be designed to account for demerits from reference-read mismatches, opening gaps and extending gaps between the alignments of R and T; this is also known as the affine gap penalty function [72]. Thereafter, the best-scoring hit can be chosen from a list of preliminary candidate hits by using the affine gap penalty scores.

Aside from alignment score, an important measure of accuracy of an alignment is the Phred-scaled [73] mapping quality score or mapQ [74]. MapQ is defined by to -10log$_{10}$ Probability{mapping position is wrong}. If an alignment is deemed to be wrong, Probability{mapping position is wrong} = 1, then its mapQ will be assigned 0. An alignment with mapQ=0 is also deemed as ambiguous. As the mapQ increases, the likelihood of the aligned read being sequenced from the reported location increases too. Due to the variations that mapQ calculation functions can be designed, a higher mapQ does not always means better alignments.

3.2 Bisulfite-treated DNA-seq aligners

The methylation state of the whole genome is termed as the methylome. By using bisulfite (BS) conversion of genomic DNA, we can study the methylome at base-pair resolution. The resultant BS-converted DNA is sequenced with NGS (BS-seq). BS-seq is then mapped to a reference genome and the single nucleotide resolution methylome can be obtained. Although NGS has advanced the study of the methylome, there are still various challenges to infer the methylome accurately from NGS data. In this subsection, we will review on these challenges and the developed approaches, which are used to analyze BS-seq data.

3.2.1 Challenges in aligning BS-seq reads

Bisulfite treatment of a DNA fragment causes unmethylated and methylated cytosines to change to uracils and remain as cytosines respectively [66]. As uracils behave as a thymine, unmethylated cytosines will be amplified as adenine upon subsequent PCR amplification for the complementary DNA strand. As a reference genome will not contain any information of unmethylation, an allowance of mismatches has to be given when aligning unmethylated cytosines against the reference genome. This will inevitably reduce mapping efficiency.

Since the methylated state of a base in the sequenced read can only be inferred by comparing it to its corresponding mapped base on the reference genome, accurate alignment of BS-seq reads is critical in correctly deriving the methylome. Thus, special consideration has to be made for induced BS-mismatches when aligning a read onto a reference genome, discriminating them from sequencing errors and polymorphisms. In addition, cytosine methylation is not symmetrical on both strands of the DNA and candidate alignments on each strand must be examined. If the BS-reads are from a directional library, then only the DNA fragments from the top (Watson) and bottom (Crick) strands will be sequenced. However, if the BS-reads are from a non-directional library, all four possible orientations of the DNA fragments (Watson-forward/reverse and Crick-forward/reverse) can be sequenced. Non-directional libraries will require aligners to align a BS-read in all of its four possible strand orientations before the best alignment can be picked for the construction of the methylome. Figure 3.1 shows the possible BS-induced conversions that can take place on cytosines of bidirectional library after bisulfite treatment.

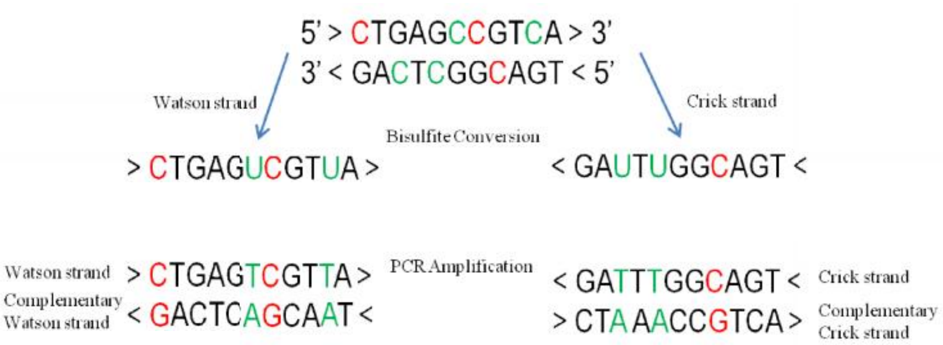


Figure 3.1. PCR amplification of bisulfite treated genomic DNA. The original strands of the DNA undergo bisulfite conversion with unmethylated-C changing to U and methylated-C remaining unchanged after the treatment. Methylated (Red) and Unmethylated (Green).

Apart from allowing a higher number of mismatches when aligning a BS-treated read onto a reference genome than with an untreated DNA read, an even higher number of mismatches needs to be allowed in aligning a BS-treated color read. This is so as a color base in a color read is called from the consensus of two adjacent nucleotides, a BS-conversion on one nucleotide will result in two adjacent color mismatches. Thus, it is computationally more expensive to align BS color-space reads than BS letter-space reads. In Chapter 4, we will describe the problem of BS-seq alignment and our solution to it in more details.

3.2.2 BS-aligner for Base-space reads

Base-space reads sequenced by Illumina sequencing technologies will represent bisulfite-converted bases as nucleotides mismatches on the reference genome. Generally, alignments of BS-reads are classified into two main types: Methylation-aware and Methylation-unbiased. After the mapping locations of the BS-reads are obtained, cytosines on the reference genome will be compared with the letter bases that are mapped to it. After which, the state of methylation on each possible site of methylation can be calculated.

3.2.3 BS-aligner for Color-space reads

Each color base in a color read depicts the transition from one letter base to the next adjacent base. Based on the terminal letter base of a color read, a color read can be converted into letter-space, one base at a time, by using the color-to-base transition matrix described in Chapter 2. However, if any of the color bases are erroneous, then this naïve conversion from color-space to letter-space will not be suitable. By using the color-to-base transition matrix on a mismatched color base, cascading base-letter mismatches will be introduced after the mismatched color base. The color error will be carried over throughout the remaining part of read when the conversion takes place. Due to this problem, in-silico conversion of cytosine to thymine is not advisable and unbiased methylation mapping is often opted out in these alignment tasks.

As each color base is interrogated from two nucleotides of a read, a letter-mismatch in the sequenced read will introduce two adjacent color-base mismatches into a color read. Hence, the same number of BS-induced conversion in a read will usually need to be aligned at a higher mismatches setting when in color-space than in base-space. However, with prior knowledge of methylation in various genomic contexts, we can apply in-silico bisulfite conversion to the reference genome in hope to reduce the number of mismatches needed to scan a BS-read against a reference genome. For most of the eukaryotic genomes, less than 5% of the methylation happens in non-CpG context [75]. Given this information, we can prepare an in-silico conversion of cytosine to thymine in non-CpG context of the reference genome prior to having BS-reads mapping on it. In general, this is not an unbiased approach to do bisulfite mapping but it does reduce the required number of mismatches that an aligner needs to scan a color-space BS-read against the reference genome. Semi methylation-aware mapping can be incorporated into an unbiased aligner to improve mapping sensitivity by remapping reads, which cannot be mapped unbiasedly. With the right set of heuristics, this 2-phase alignment strategy can improve mapping sensitivity and accuracy without much impedance on its speed [76].

3.2.4 Methylation-aware mapping

In methylation-aware aligners, cytosine in a BS-read is assumed to be sequenced from a methylated cytosine, whereas, a thymine is assumed to be either from an unmethylated cytosine or thymine. These assumptions encapsulate all possible combinations of methylation status that cytosines and thymines can have in a BS-read. For example, if a BS-read is to be sequenced with 10 cytosines and thymines, then a methylation-aware aligner will try to map 2^{10} possible representations of this BS-read onto the reference genome; permuting between cytosine and thymine at those 10 positions. Due to the amount of search-space that the aligner need to search through, it is able to produce the highest possible sensitivity in mapping color BS-reads.

At the expense of high mapping sensitivity, comes with speed slowdown and overestimation of methylation levels. In methylation-aware alignment, the reference genome is used as it is; assuming full methylation throughout the genome. As such, methylated sequences will map to the original genome more easily than sequences of lower methylation rates. This directly causes an overestimation of methylation levels from mappings reported by methylation-aware aligners.

An example of methylation-aware aligner is SOCS-B [77]. SOCS-B starts the alignment of a color read by first converting it to base-space. Four translations are computed, starting from all four possible nucleotides as the terminal base instead of the terminal primer base provided by the original color read. The substrings of the translated reads are enumerated in ternary to form a partial hash over positions represented by a cytosine or thymine. The mapping algorithm is based on an iterative version of the Rabin-Karp algorithm and generates candidate genomic locations of the partial hash. SOCS-B then uses dynamic programming and base qualities to compute the most probable methylation state for each cytosine. The optimal alignment should have the least number of color-space mismatches with respect to the reference genome.

3.2.5 Unbiased-Methylation mapping

Unbiased-methylation aligners convert cytosines in the BS-reads and reference genome to thymines prior to alignment and methylation-aware aligners do not. This conversion assumes that the BS-reads are fully BS-converted due to unmethylation throughout the experimental data. Since both the reference and read has all its cytosines being changed to thymines, the methylation state of either the reference or the read will not affect the alignment of such an *in-silico* converted read. The conversion will also not incur any biased estimation of methylation state to a BS-read after alignment. Since the BS-read and genome now assume same states of methylation, using a DNA-seq aligner will already enable us to align the converted error-free BS-read onto the converted reference genome with an exact match.

In the aspect of unbiased mapping, this type of alignment will have some shortcomings. Due to the in-silico conversion of cytosines to thymines, the alphabet size of the data gets reduce from 4 to 3; the complexity of the data is now greatly reduced. Thus, it has now become harder to map such an *in-silico* converted BS-read onto a similarly converted genome unambiguously. Hence, unbiased-methylation aligners generally yield lower mapping efficiency than methylation-aware aligners.

Some of the BS-aligners that fall into this category are Bismark [78], BRAT [79, 80] and BS-Seeker [81]. Bismark and BS-Seeker are based on Bowtie as a pre-mapping tool to align BS-reads for preliminary candidate mapping locations. These two methods prepare in-silico fully converted references prior to alignment and will have Bowtie map BS-read onto it. In addition, Bismark synchronizes the threads of Bowtie to consider methylation level of each cytosine from each read and this slows down the overall runtime of the program. BS-Seeker outputs the preliminary alignments of each thread into separate files and post-process these alignments but it takes up additional storage prior to the consideration of methylation levels for each read. BRAT-BW implements an FM-index alignment routine from scratch to avoid the problem of synchronization and large temporary storage from using an auxiliary program as a pre-mapping tool. BRAT-BW also guarantees to find all alignments if there is at most one mismatch in a prefix of length 32-64 bp (user-defined) of the read.

3.2.6 Semi Methylation-aware mapping

Due to the differences of methylation levels in different genomic contexts, in-silico conversion of the reference genome can be done to allow for both methylation-aware mapping and unbiased methylation alignment to take place at different parts of the reference genome with the same reference index. A semi methylation-aware mapping approach to profile the human methylome is to do unbiased mapping in CpG context and methylation-aware mapping in non-CpG context. This approach is used to improve mapping sensitivity by utilizing prior knowledge of expected methylation levels in different genomic contexts as studied in [82]. If such an aligner were to be used to map BS-reads from flowering plants to a corresponding reference genome, the aligner would probably do unbiased mapping in non-CHH (H = A, C and T) context, and methylation-aware mappings in CG and CHG contexts. The gain in mapping sensitivity comes at the expense of similar but milder drawbacks seen by methylation-aware aligners.

An example of an aligner that depends on such a mapping strategy is RMAP [83] and PASS-bis [84]. RMAP uses wildcard matching for positions represented by thymines and, thus, only maps unbiasedly in CpG genomic context; otherwise, it performs biased mapping in non-CpG genomic context. PASS-bis can map both base-space and color-space reads. While it does map base-space reads unbiasedly, it does not do so for color-space reads. Due to the fact that PASS-bis converts a color-space read to base-space read prior to mapping, the base-space read could be mis-represented by the reference due to cascading errors due to this type of naïve conversion. In order to maximize the mappability of each color read, PASS-bis performs a secondary phase of mapping based on the combinatorial assortment of genomic C-T conversions that we have already discussed as methylation-aware mapping. As this second phase of mapping is slow, it is implemented as an option in PASS-bis and even if it is used, it will only be activated when the read fails to map onto the fully in-silico converted reference genomes.

3.2.7 Comparison of BS-Seq Aligners

In the previous section, we have reviewed on three approaches, which are used to align BS-seq reads and the two types of representations that can represent BS-seq reads. Below, we summarize the details of the different BS-seq alignment methods for the analysis of methylome in Table 3.2.

Table 3.2. Methods for the alignment of Bisulfite-seq data and their performance measures

Method	Bismark	BRAT-BW	BSMAP	BS-Seeker	PASS-bis	RMAP-BS	SOCS-B	B-SOLANA
Reference	[78]	[80]	[85]	[81]	[84]	[83]	[77]	[86]
Mapping strategy	Bowtie	FM-index	SOAP	Bowtie	PASS	Positional weight matrix matching	Robin-Karp algorithm	Bowtie
Read-space	Letter	Letter	Letter	Letter	Letter/Color	Letter	Color	Color
Paired-end mode	Y	Y	Y	N	Y	N	N	N
Methylation-aware mapping	Unbiased	Unbiased	Biased	Unbiased	Semi	Semi	Biased	Biased
Best Alignment criteria	Lowest number of non BS-mismatches	Lowest number of mismatches OR non BS-mismatches	Lowest number of mismatches	Lowest number of mismatches	Lowest number of non BS-mismatches	Lowest number of mismatches	Lowest number of non BS-mismatches	Lowest number of mismatches
Output	a,b,c,d	a,c	a,b	a,b	a,b	a	a,b	a,b
Advantages	Speed	Speed	-	Speed	Sensitive	-	Full methylation-aware	Speed
Disadvantages	-	-	Speed	-	-	Speed and semi-biased mapping	Speed	-

a. Alignment output.
b. Methylation calls output.
c. Methylation caller.
d. Summary of methylation level.

3.3 Gapped DNA-seq aligners

In numerous studies of Mendelian diseases, periodic sequences were found to be mutagenic and contexts of indels in the human coding sequences were investigated for possible onsets of diseases [87, 88]. In the case of cancers, differential mutational studies were also carried out to identify somatic differences between normal and diseased tissues. For instance, the identification of aberrant integrations of hepatitis B virus into the genomes of its host-tissue will increase the chances for the onset of malignant hepatoma [89]. All these studies would not have been possible without the advent of gapped DNA-seq aligners.

In order to improve the space-time alignment efficiency of the voluminous data brought about by NGS, aligners relied on various indexing strategies. Indexing approaches can be divided into two main groups based on either the reference genome or query reads are indexed. Methods such as BWA [90, 91], Bowtie [92, 93], SOAP [94, 95], Novoalign [96], Stampy [97], PASS [98], CUSHAW [99, 100], SRmapper [101], SeqAlto [102] index the reference genome. On the other hand, Eland [103], RMAP [104], MAQ [74], SHRiMP [105, 106] and ZOOM [107] index the query reads and map them back onto the genomic reference sequences.

In general, gapped DNA-seq aligners can be classified in more than one way. In this thesis, we classified aligners based on their indexing strategies: hash based, suffix-trie based and merge-sort based approaches. With the context of this thesis in mind, we will not describe the details of the merge-sort based approach: SliderI/II. Readers who wish to understand how merge-sort was applied to the alignment of genomic data can refer to [108, 109].

Gapped aligners mostly involve finding a list of preliminary candidate mapping locations by aligning a substring of the read onto the reference with a technique called seeding. After which, a secondary step takes place by locally aligning the full-length reads against each of the preliminary candidate locations, also known as the extension phase, before they are written to the output file. The secondary step is computationally expensive and is the main reason why gapped alignment was avoided by pioneering mismatch-only aligners for short (~36 bp) reads. There are also works revolving around hardware acceleration to improve the efficiencies of local alignment [110, 111]. In general, seed-and-extend strategy dominates the field of aligning NGS reads.

3.3.1 Challenges in Gapped Alignment

During the early development of alignment algorithms for NGS reads, a number of pioneering aligners [74, 90, 92, 104] were developed. These aligners map a query read onto a reference genome within a number of mismatches only, which is also known as ungapped alignment. Due to the limitations of past sequencing technologies, the lengths of reads were short and ranged from ~25 bp to ~36 bp long. The short read-lengths slowed down most algorithms due to its redundant representations in the large reference genome and made gapped alignment infeasible. However, due to advancements in sequencing technologies, read-lengths now can reach as long as ~100 bp and ~250 bp from Illumina GAIIx/HiSeq and MiSeq machines respectively. The lengthened reads have now made gapped alignment more tractable.

As reads get longer, they will become likely to contain more SNPs, indels and structural variations in them than shorter reads. Ungapped alignment will not be sufficient to align them and gapped alignment becomes critically important in aligning these longer reads. In Chapter 5, we will describe the problem of gapped RNA-seq alignment and our solution to it in more details.

3.3.2 Hash/Seed based Approaches

Hash based approaches stemmed from the first hash-based algorithm, BLAST [31], and followed the seed-and-extend paradigm. Since the publication of the BLAST paper, many developments have been made to its original seeding idea to handle more genomic features that are present in NGS reads. In the following subsections, we will report on the different seeding methodologies found in the literature.

3.3.2.1 Seeds

The most primitive type of a seed is a contiguous substring of the query read. Seed is also termed as k-mer and is usually referred to a specific n-tuple of nucleotides or amino acid sequences. Pioneering aligner for NGS reads, such as BLAST, uses 11-mers (for DNA sequences) to seed the alignment query. Subsequently, BLAT [32], MegaBLAST [112] and YAHA [113] were developed to use 11-mer, 28-mer and 15-mer respectively.

By using a seed instead of the original query string for alignment, we can increase the sensitivity of the method. As a seed is much shorter than the original query read, it would have a higher chance of finding an exact representation of itself in the same reference genome. However, due to the reduced informational content in the seed (trimmed from the query read), it is now less unique and can be spuriously represented by many regions of the reference genome identically. In a seeding approach, the first task is to identify all possible locations to which the original read may be aligned to. Next, an extension step is to be performed on the seed against the candidate locations to report the alignments in order of decreasing alignment scores to the user.

3.3.2.2 Mismatch-seeds

If the correct alignment of a query read lies in an SNP-dense region of the DNA, a k-mer seed might miss it and, worse, other seeds may report a false-positive hit to the user. To resolve this shortcoming, mismatches are allowed in a seed to avoid missing the correct alignment during the initial seeding-phase of the alignment.

To the best of my knowledge, RMAP [104] is the first method to use mismatch-seed in the alignment of sequenced reads. RMAP uses a different set of seeds to achieve full sensitivity of k-mismatches reads through the use of k+1 seeds [114]. According to the pigeonhole principle, if we were to partition a k-mismatch read into k+1 equal adjacent and non-overlapping seeds, then at least one of the k+1 seeds can be represented exactly in the reference text. RMAP first identifies putative locations in the reference genome where the seeds can be matched exactly. Exact matching is preferred as it can be executed more efficiently than approximate matching and only the unseeded portions of the reads need to be realigned during the extension-phase of the alignment. The disadvantage of this seeding approach becomes obvious when k is large and each mismatch-seed is small. Ultra-short seeds will return too many spurious candidate locations for the extension-phase to work with and will take a great hit on efficiency.

3.3.2.3 Spaced-Seeds

By using contiguous bases as seeds, we are faced with two conflicting performance factors which aligners are designed to improve on: Speed and Sensitivity. In a seed-and-extend paradigm, an aligner would want to minimize the number of preliminary candidate locations, as it is computationally expensive to perform local realignment on all of them. With this in mind, better filtration methods were designed such that the seed-phase of the alignment will return a minimal set of candidate locations and preferably with one of the seeds representing the correct location of the query read. As such, spaced-seeds were developed as a filtration technique to achieve a balance between these two conflicting performance factors [115] in aligners.

A spaced-seed can be specified using a sequence of 1's and 0's. Some of the positions in a spaced-seed will be sampled and some will not be sampled, inherently allowing mismatches between itself and the reference sequence. A query performed with a spaced-seed will skip the sampling of bases between the reference and the underlying read-sequence that are marked by a 0 in the template spaced-seed. For instance, the use of spaced seed in PatternHunter showed that a template '111010010100110111' can be ~50% more sensitive than BLAST's default 11-mer seed for two sequences of 70% similarity [116].

Pioneering aligner based on the use of spaced-seed for filtration is Eland [103]. It used six spaced-seeds to span the entire query read. The scanning of the six spaced-seeds ensured that a two-mismatch query read (with respect to the reference), regardless of the mismatches' positions in the read, will be represented by at least one of the seeds. MAQ extended the idea of 6-template-2-mismatches from PatternHunter to guarantee recovery of k-mismatches hit of a query read. However, to provide full sensitivity, MAQ required $\binom{2k}{k}$ spaced-seeds to guarantee full sensitivity of k-mismatches mappings. Due to the large number of seeds that need to be scanned, MAQ guarantees full sensitivity by only using the spaced-seed seeding approach on the first 28 bp segment of the read and allowing at most two mismatches in this 28 bp segment. Usually the first k-bases from the 5' end of a sequenced read is selected to seed a query as it is empirically shown to contain less sequencing errors [117]. Once the spaced-seed returns a partial match, the seeding match is then fully extended to the full length of the input query read.

For the design of a minimum set of spaced-seeds to achieve certain sensitivity requirement and memory usage on a given read length, readers can refer to ZOOM! [107] for more details.

3.3.2.4 q-gram Filter

The primitive approach to recover indels from a short read is to anchor parts of the query read onto the reference in the seeding-phase and the indels to be recovered in the extension-phase by using SW-algorithm. Small indels of 1-3 bp [94] can be detected using this seed-and-extend approach.

In the previous seeding approaches, the candidate hits from one long seed will undergo the extension-phase of alignment. A q-gram is similar to a contiguous seed but by using multiple substrings in the filtration step, the extension-phase is only initiated on a cluster of localized seeds that shares t matching q-grams instead of partial matches from a single long contiguous seed. The q-gram filter is based on the observation that if the query string has at most k mismatches and gaps, then both the query string and the reference of length w will shares at least t = (w+1) – (k+1)q common q-grams [118, 119]. Based on q-gram filter, SHRiMP [105, 106] and RazerS [120, 121] are able to build an index which innately allowing gaps during the seeding-phase of an alignment. A more recent variant of q-gram filter can be seem in MASAI [122] where a set of multiple seeds can be searched simultaneously on a reference index to speed up alignment by 11.9x as compared to RaserS3.

3.3.3 Prefix/Suffix trie based approaches

The seed-phase and extension-phase of alignment correspond to the exact string matching problem and inexact string matching problem respectively for aligners based on trie. For these aligners to find an exact match to substrings of the query reads, they have to build an index of the reference genome using data structures such as FM-index [67], suffix array [123] and suffix tree [124]. The advantage of searching a query string against a trie-based index is that identical substrings of a reference genome need only to be searched once, as identical substrings collapse onto a single traversal path in trie-index.

The first use of trie in aligners are mainly based on suffix tree and can be traced back to MUMer [125] and OASIS [126]. However, the disadvantage of using suffix trees becomes obvious when the search index is huge. An immediate improvement on suffix tree, with respect to space-efficiency, is brought about by the development of suffix array (SA) based on Farach's [127] optimal linear time suffix tree construction algorithm. However, the sizes of the index built, based on suffix tree and suffix array still require more or equal memory as compared to the reference itself. Due to the fact that the reference genomes can be very large and is best to reside entirely in the working physical memory of the computer during alignment, the development of genomic index-building algorithms was motivated towards building a more space-efficient index. As such, reference indices based on enhanced suffix array (ESA) [128] and FM-index which required only space within or even less than the size of the reference were developed.

Vmatch [128] and Segemehl [129] are based on ESA which consist of an SA and auxiliary arrays. Theoretically, ESA is able to store each nucleotide using 6.25 bytes. Since ESA is a succinct representation of the suffix tree, ESA allows exact queries at the same time complexity offered by suffix tree while requiring lower space requirement than SA. A further improvement on space-efficiency is achieved through the development of FM-index that compresses full-text reference using Burrows-Wheeler transform [68]. The inventors of the FM-index also observed that the descendants of a node in a prefix trie could be located in constant time by performing an opportunistic backwards search on the structure itself, which allowed it to have the same time complexity of performing exact matching operations with that of a trie. Some pioneering genomic aligners that use FM-index are Bowtie, BWA and SOAP2. The FM-index is the most used trie-based index due to its minute memory footprint. GEM [130] is shown to be the fastest aligner by our benchmarks in the later chapter of this thesis and is also based on the FM-index.

3.3.3.1 Inexact Matching using Trie <stop>

Aligners can be based on different trie-related data structures but all of them can be translated into one another succinctly. Trie is excellent in finding exact matches as all identical copies of substrings from the reference are collapsed into a single traversal path but it is not ideal for performing inexact matches. Inexact matching is performed on trie by introducing mismatches and/or gaps, during the traversal of the trie, when the alignment progresses in a depth-first traversal on the data structure. The mismatches make the search space grow exponentially and affect alignment speed dramatically.

In order to perform inexact matching on trie efficiently, algorithms are designed to only explore a small portion of the initial search space. With the added constraints of a pruned search space, aligners hope to achieve speedups with minimal impact on sensitivity and accuracy of alignments in search/trie-traversal alignment algorithms.

MUMmer, Vmatch, CUSHAW2 and YAHA anchor alignment with exact matches and join these exact matched segments with gapped local alignment. In addition, Segemehl tries to align the longest exact prefix of each suffix and also introduces mismatches at certain positions of the query read to reduce false alignments.

OASIS and BWT-SW searches substrings of the reference by a depth-first traversal on the trie and align these substrings with the query strings by dynamic programming. BWA-SW extends from BWT-SW [131] by representing the query string as a directed words graph (DAWG) which enables it to deploy pruning heuristics to speed up alignments of query strings.

As dynamic programming using SW/NW-algorithms is much slower than a linear-time exact matching between the query string and the reference BWT-index, it is avoided as much as possible in Bowtie and BWA. Instead of realigning the short substrings of the reference with the query string, the query and substrings of the reference are only being realigned if they are within a number of mismatches. As BWA and Bowtie align a query read by the traversal of an FM-index, it can prune some branches in the search space off so this will result in fewer number of text-edit operations between the query read and reference genome for each read. BWA further speeds up gapped alignment by performing a banded-SW algorithm and employing MegaBLAST's X-Dropoff heuristic in the extension-phase of its seeding alignments.

Bowtie2 samples a set of 22-mer seeds from the query string using exact matching. The seeds are extended to their full length by dynamic programming in order of their frequencies of occurrences in the reference genome as indicated by their suffix array intervals. The prioritized seeds are realigned using hardware-accelerated versions of SW/NW-algorithms with Streaming SIMD Extensions 2 (SSE2) hardware instructions.

GEM uses region-based filtration technique to speed up exhaustive alignment of its query string. This technique identifies non-overlapping regions that are also non-repetitive (not masked by RepeatMasker) for the extension-phase. The seeds are extended using Myer's fast bit-vector algorithm. GEM can align up to several times faster than Bowtie2 and BWA as the filtration technique greatly reduced the number of candidate reference positions to be realigned.

3.3.4 Hardware acceleration of seed-extension

During implementation of the algorithms, source codes are written in a rather sequential manner but they do not need to be interpreted and executed in a sequential manner. By exploiting the features of modern hardware and application programming interface, performance of sequential programs can be improved. Many aligners are able to achieve decent speedups by introducing elements of concurrency into their algorithms. Three main exploits that current aligners have in them are multi-threading capabilities on multi-core system, Single-Input-Multiple-Data (SIMD) instructions and Graphics Processing Units (GPUs) accelerations.

Since the introduction of multiple-core central processing units (CPUs) in bench-top personal computers, coders have tried to fully utilize the availability of computational power on these processors by having multiple threads of their single program run in parallel on a single computer. Since the memory-overheard incurred by an addition thread of operation in genomic alignment is small, many users of genomic applications prefer using a shared-memory policy within a single execution process such as CUSHAW2.

In genomic applications, due to the alphabet size of data being handled, 2-bit encoding is often used, and many indexing and alignment operations can be seen as bit-based operations. In the extension-phase of alignment, the binary bits that represent the query string and reference text can be fetched into the registers of the CPUs such that a single instruction can manipulate more information per bit than it would normally have. A common application of SIMD acceleration [110, 111] is on the SW/NW-algorithm routine used in the extension-phase of aligners such as Bowtie2, SHRiMP and Novoalign.

GPUs are also gaining popularity in genomic applications. CPUs are designed with multiple computational cores for serial processing while general purpose GPUs consists of thousands of smaller computational cores which are designed for massive parallel processing of users-customized code [132]. More than often, the alignment of a genomic DNA fragment is independent of the alignment of other fragments and executing them in parallel is possible without affecting the end-results of each individual alignment. By using GPUs over CPUs in genomic alignment, SOAP3 [133] was able to achieve tens of times speedup over SOAP2 [95].

3.3.5 Comparison of Gapped DNA-Seq Aligners

In the previous section, we have reviewed on two indexing strategies and two mapping approaches for aligning gapped reads. In Table 3.3, we summarize the details of different gapped alignment methods together with a short description to each of them.

Table 3.3. Methods for gapped alignment and their respective main indexing/mapping strategies

Methods	Index		Type of Mapping Approach		Description	Reference
	Reference	Read	Hash-based	BWT-based		
BFAST	X		X		Uses empirical derived seed template for mapping fixed read lengths and genome sizes	[134]
BLASR	X			X	Maps PacBio reads with successive refinements to the local alignments of the seed locations	[135]
Bowtie	X			X	Bowtie1/2 aim at fast and sensitive mappings of reads. Version 2 targets longer reads and can do gapped alignment too	[92, 93]
BWA	X			X	BWA-short targets short reads of ~100bp with low (~3%) error rate. BWA-SW targets longer reads up to 10kbp with higher error rate	[90, 91]
CloudBurst		X	X		Uses Hadoop MapReduce framework to do alignment in the CLOUD	[136]
CUSHAW2	X			X	CUSHAW1 is targeted for CUDA-enabled GPUs. CUSHAW2(-GPU) is targeted for long read alignment for CPUs (GPUs).	[99, 100, 137]
Eland		X	X		First NGS short read aligner. Allows up to two mismatches in an alignment	[103]
GEM	X			X	Based on adaptive region based filtration technique for sensitive and extremely fast alignment efficiency	[130]
GNUMAP	X		X		Targets accurate gapped alignment of Illumina reads	[138]

Hobbes	X		X		Reports multiple putative mappings fast	[139]
MAQ		X	X		First program to use posterior mapping score to disambiguate multiple candidate mappings	[74]
Masai	X			X	Uses approximate seeds to speed up alignments	[122]
MOM	X		X		Identifies the maximal length match within the short read.	[140]
Mosaik	X		X		Uses banded SW-algorithm for extending seed locations	[141]
mrFAST	X		X		Uses cache oblivious memory technique to minimize memory miss-transfers to speed up gapped alignments of letter-space reads. mrsFAST is ungapped version of mrFAST. drFAST is designed for color-space reads.	[142-144]
Novoalign	X		X		High sensitivity and specificity alignments. Uses base qualities in all steps of alignments and output good calibrated posterior mapping quality scores	[96]
PASS	X		X		Alignment of words are pre-computed from the hashed index of the genome	[98]
PerM	X		X		Uses periodic seeds to quickly find alignments of up to four mismatches with full sensitivity	[145]
ProbeMatch	X		X		Uses gapped q-grams and q-grams of various pattern to identify seeding locations from a reference	[146]

RazerS		X	X		No restriction on read length. Seeds can be designed with predictable tradeoff between sensitivity and speed	[120, 121]
REAL	X		X		Targeted at fast, accurate and sensitive mappings of single-end reads	[147]
RMAP		X	X		Can map reads with an arbitrary numbers of mismatches	[104]
SeqAlto	X		X		Uses adaptive seeding approach to terminate alignment when alignment reaches certain confidence for reporting	[102]
SeqMap		X	X		Can align up to a mixture of 5 mismatches and gaps between the reference and the read	[148]
SHRiMP		X	X		Aims at accurate mapping of color-space reads. Version 2 index the reference instead of the reads	[105, 106]
Slider			Merge-sort		Reduces the percentage of base call error mismatches in an alignment; produces high SNP discovery rate	[108, 109]
SOAP2	X			X	Fast and accurate alignments on a wide range of read lengths. Improved version of SOAP1. SOAP3 is akin to GPU-enabled SOAP2.	[94, 95, 133]
SRmapper	X		X		Small memory footprint. ~2.5GB for human genome.	[101]
SSAHA2	X		X		Fast alignments for reads of small number of variants	[149]
Stampy	X		X		High sensitivity of reads with high percentages of variants in them. Very slow but can be sped up by using BWA as a pre-mapping tool	[97]

Subread	X		X	Uses novel 'seed-and-vote' paradigm to perform fast alignments	[150]
YAHA	X		X	Recover optimal breakpoints of alignments for structural variation detection	[113]
ZOOM		X	X	100% sensitivity of reads between 15-240bp with reasonable number of mismatches and gaps.	[107]

3.4 RNA-seq aligners <stop>

RNA, together with DNA and proteins, is one of the three major macromolecules that are needed for life. Pre-mRNA is synthesized from the DNA in a process called transcription and is matured by having its introns removed in eukaryotic cells [151]. In mammalian genomes, alternate splicing adds onto the genomic diversity by generating isoforms of the same gene [152]. The disruption in the synthesis of mRNA isoforms can cause genetic diseases [153, 154].

Since it is motivating to produce a map of genes together with their expression level on the genome-wide scale across various cell types, it is critical to annotate a transcriptome efficiently and accurately. The prevalent method for producing a genome-wide gene-map requires the costly and low-throughput method of applying capillary sequencing on cDNAs or expressed sequence tag (EST) fragments [62]. Due to the usage of low-throughput sequencing, the true diversity brought about by alternate splicing cannot be studied in depth without the advent of high throughput data. Alternatives to capillary sequencing of ESTs are tiling arrays and splice-aware microarrays. Tiling arrays are able to interrogate large transcribed regions but at limited resolution [155]. As for SJ-aware microarrays, they are fabricated with probes that hybridize to known RNA sequences and will not be suitable to quantify expression levels of novel or unrepresented genes [156, 157].

Due to the advent of NGS technologies, we are able to sequence cDNA sequences derived from RNA fragments [33]. This gave rise to high throughput sequencing of RNA fragments which we now know as RNA-seq. Methods such as Exonerate [158] and BLAT [32] which are designed for the alignment of capillary sequencing technologies are now unable to map voluminous NGS data within competitive timings. In order to improve the space-time efficiency in the alignment of RNA-seq data, computational tools have to be developed to deliver unprecedented speed for the alignment of RNA-seq reads.

Akin to the analysis of DNA-seq datasets, the first step of analyzing RNA-seq datasets is to align the RNA-seq reads back onto a reference genome or transcriptome. Given the myriad of aligners developed in the recent years, we are able to group these aligners into two main groups: Unspliced and spliced aligners.

3.4.1 Challenges in RNA-seq Alignment

The goal of RNA-seq alignment is the resolution of the gene-map together with the entire set of splice junctions annotated in it for different types of cells. Although the main challenge in gapped DNA-seq alignment is similar to RNA-seq alignment, the task of RNA-seq alignment is tougher as reads now need to be split into smaller k-mer for identification of small-exons (<20 bp) too. Shorter read fragments are harder to map unambiguously and will be more computationally expensive to resolve during the extension phase of the alignment. In addition, accurate detection of split junction without prior knowledge of splice signals is still an open problem especially in lowly transcribed regions. To make matters worse, canonical splice signals are ubiquitous in both transcribed and non-transcribed regions.

The presence of unexpressed genomic sequences, which are similar to concatenated sequences of multiple exons also, pose problems to the accurate alignment of RNA-seq reads. These regions, which are known as pseudogenes [159], are not transcribed from the genomic DNA into mRNA sequences and should not have RNA-seq reads mapping to it. However, due to the case whereby multi-exon spanning reads may map to these regions, without splicing, will pose a great challenge to exon-first method. Seed-and-extend methods will also face problem in determining if an unspliced full alignment of a RNA-seq read should actually be spliced or not.

In Chapter 6, we will describe the problem of spliced RNA-seq alignment and our proposed solution to it in detail.

3.4.2 Unspliced/Annotation-guided Aligners

The unspliced aligners are mostly as described in the previous section on gapped aligners. In the aspect of aligning RNA-seq data, unspliced aligners are mostly used to align RNA-seq reads to the assembled transcriptome without having the need to allow for large intronic gaps during alignment. Due to the use of the assembled transcriptome, unspliced aligners are also known as annotation-based aligners in the literature. Unspliced aligners are used when de novo detection of splice junctions is not needed and are great for mappings reads against a well annotated transcriptome for quantification studies [160-162]. Some examples of unspliced aligners are ERANGE [160] and RNA-MATE [163].

ERANGE begins by mapping reads onto the DNA reference genome. Reads that cannot map onto the DNA reference will be mapped again onto a known transcriptome. Highly reliable mappings from the previous 2-step alignment will be used to calculate the Reads Per Kilobase per Million mapped reads (RPKM) of putative transcripts. Lastly, the assignment of leftover ambiguous mappings to the current transcriptome will be based on the previously calculated RPKM as a form of weightage.

RNA-MATE is developed for aligning color-space RNA-seq reads. It follows a similar 2-step alignment strategy used in ERANGE. However, it is largely based on a recursive methodology. A read is truncated if it fails to map to a known transcriptome or DNA reference. The process of truncation is repeated until the truncated read length reaches a certain lower threshold or can be mapped using the 2-step alignment strategy. RNA-

MATE also provides an option to use ambiguous mappings from the alignment step for the quantification of expression transcripts. RNA-MATE is now superseded by X-MATE [164].

3.4.3 Spliced Aligner

Unlike unspliced aligners, spliced aligners align RNA-seq reads, consisting of adjacent exons and introns in it, back onto a genomic DNA reference genome. The transcripts represented by NCBI Reference Sequence Database (RefSeq) [165, 166] are downloaded and used as an oracle set for BEERS [167] to simulate RNA-seq reads from. On 76 bp, 100 bp and 120 bp of 2 millions simulated reads each, it was observed that there is 17.8%, 22.4% and 25.5% of reads spanning across two or more exons respectively in a single read. With increasing read lengths generated by improving sequencing technologies, it has become more important for aligner to handle spliced alignments more efficiently and accurately. Spliced aligners can generally be classified into two categories based on their method of detecting splice junctions - Exon-first and Seed-and-Extend. We will also describe a learning-based approach that is a sub-class of spliced aligners here.

3.4.3.1 Exon-first Approaches

Aligners categorized as exon-first approaches map the original RNA-seq reads onto a DNA reference in an ungapped fashion first. This initial alignment step will align reads that do not span across exon-exon junctions successfully. Hence, they are named "exon-first" approaches. This step essentially quantifies transcript abundance using only exonic reads and does not account for exon-exon boundaries in the reads. The mapped exonic reads are used as a guide to guide the detection of splice junction in the latter extension step. TopHat [40] and G-Mo.R-Se [168] to incorporate the mappings of the exonic reads to guide the alignment of non-exonic reads. The downside of this approach is that sufficient coverage is needed from the exonic mappings before it can be used to align non-exonic reads confidently.

The unmapped reads from the initial alignment step are now split into shorter fragments and aligned independently. Since the fragments are now shorter, they stand a better chance of aligning exactly onto the DNA reference lessening the chance of spanning across exon-exon boundaries. However, more computational power needs to be spent on realigning the many alignments that may be returned from the shorter seeded read-length to their full read-length. Although, extending the seed alignments can be slow, exon-first aligners can be very efficient as minority of reads would need this computationally expensive step.

Some examples of exon-first aligners are GEM (splice alignment module) [130], TopHat1/2 [40, 41], MapSplice [36], SpliceMap [39], SOAPsplice [169] and PASSion [37].

3.4.3.2 Seed-and-Extend Approaches

This class of spliced aligners begins aligning reads onto a DNA reference by splitting them up into smaller fragments first. The candidate alignments of these fragments are then used to localize the actual alignment of the original read. By merging initial seeding alignments with local realignment, the seeding alignments of the split fragments can be extended towards one and another to the original full read length. Some methods of this approach are QPALMA [38], GSNAP [170], Supersplat [171] and STAR [34].

Recently, seed-and-extend strategy is also extended to consider a read as a concatenation of multiple smaller read fragments by using multiple seeds in the alignment of the reads. These methods include CRAC [172], OLego [35] and Subjunc [150].

For both exon-first and seed-and-extend approaches, it is possible to check for flanking donor-recipient dinucleotides near the intron boundaries with known canonical splice sites to increase the reliability of detecting novel splice junctions and recover short-overhangs.

3.4.3.3 Learning-based Approaches

One of the earliest RNA-seq spliced aligner is QPALMA [38]. It is based on a learning algorithm, support vector machines (SVM) [173], to learn how splice junctions are positioned on a reference genome by training with a known set of spliced mappings. However, the performance of this strategy relies heavily on the completeness of the underlying training dataset for efficient and accurate performance. PALMapper [174] succeeded QPALMA, which is a combination of QPALMA and the short read alignment tool GenomeMapper [175]. PALMapper improved on the speed by using a banded semi-global and spliced alignment algorithm of GenomeMapper to align the RNA-seq reads while taking advantage of base quality information and the predictions of splice junctions from the SVM algorithm.

Also based on a learning approach is HMMSplicer [176] which is a tool developed for the discovery of novel and known splice junctions. HMMSplicer trains a hidden markov chain model (HMM) by using the halves of aligned reads that initially cannot align onto the genomic reference genome. From the trained HMM model, the method tries to find the splice junctions within the other halves of these aligned reads and match the remaining portion of the read to the downstream of the spliced sites. As HMMSplicer trains using data from the input itself, it is also capable of detecting novel splice junctions.

Since the objective of using machine-learning approach in RNA-seq alignment is for the accurate discovery of known splice junctions, learning-based approaches can also be regarded as a subset of spliced aligners.

Most aligners assume a known gene model for the sequenced reads and can be biased towards the detection of canonical (~98.7% of the splicing junctions in mammalian sample [177]) and semi-canonical junctions . Non-canonical junctions such as splicing of exons that does not lie on the same RNA transcript (trans-splicing [178]) may not be detected. However, learning-based approached can learn from sample-specific data and train a sample-specific model for unbiased detection of splicing junctions without relying on annotations of known splicing motifs. As such, this approach might be more suitable for de novo discovery of splicing junctions of less studied organisms.

3.4.4 Comparison of RNA-seq Aligners

In the previous section, we have reviewed on two main mapping strategies, possible usage of known canonical splicing signals and annotated intronic gaps for guided RNA-seq alignment. We summarize and characterize different RNA-seq alignment methods for the analysis of transcriptome in Table 3.4.

Table 3.4. Methods for RNA-seq alignment and their respective mapping strategies and usage of annotations for spliced alignments

Methods	Mapping Strategy		Use of Annotations		Splice junction Model		Description	Reference
	Exon-first	Seed-Extend	Yes	No	Biased	Unbiased		
ABmapper	X			X	X		k-mer from both ends of seeds are searched against Suffix Array index and extended towards each. Still essentially a exon-first approach as seeds is extended for exonic mapping first.	[179]
CRAC		X		X		X	Uses k-mer profiling to detect candidate mutations, indels, splicing and chimeric junctions	[172]
GEM-rna-mapper	X		X	X		X	Based on GEM. (unpublished)	[130]
GSNAP		X		X	X		Detection of novel splice junctions is based on a probabilistic model implemented as a maximum entropy model on user-specified known splice junctions.[180]	[170]
HMMsplicer	X			L	X		Uses half-read mapping to train a HMM to detect most probable splice position	[176]
MapNext		X	X	X	X		Using un-annotated mode, searches paired k-mer in a hash table within 10kbp and with the same strand	[181]
MapSplice	X			X		X	Sensitive for exonic reads	[36]

OLego	X		X	L	X		Targeted at finding small exon and good specificity on exonic reads. Based on logistic regression for detecting splice junctions	[35]
OSA		X	X	X	X		Trims poor-quality 3' ends of reads and improves alignment speed	[182]
PALMapper		X	L		X		Combined QPALMA and GenomeMapper	[174]
PASSion	X			X	X		Use of pattern growth algorithm and splicing signals to detect both novel and known splicing junctions	[37]
PASTA	-			L	X		Similar to seed-eXtend strategy, it uses patterned alignments of 2 subreads split at various points for spliced mapping	[183]
QPALMA		X	L		X		Used in silico spliced reads from annotated genome to train a 'weighted degree' kernel with SVMs	[38]
RNASEQR		X	X			X	Reduces false identifications of SNVs near splice junctions	[184]
RUM	X		X	X	X		Combination of Bowtie (exonic) and BLAT (spliced) are used to align reads to both the transcriptome and genome	[167]
SeqSaw		X		X		X	Based on SeqMap [148]. High specificity in detecting splice junctions	[185]
SOAPsplice	X			X	X		Use two filtration strategies to produce low false positive rates	[169]

SpliceMap	X			X	X		50bp reads cannot be extended for more than 40bp and residual overhang must be >10bp	[39]
SplitSeek		X		X		X	Suitable for detecting novel splicing junctions and chimeric transcripts	[186]
STAR		X	X	X		X	Ultrafast aligner that can discover non-canonical junctions and fusion junctions	[34]
Subread/Subjunc		X		X		X	Uses a seed-and-vote strategy on sub-reads for alignment	[150]
Supersplat		X		X		X	Finds every possible splice junction by mapping different 2-chunk reads for alignment	[171]
TopHat 1/2	X		X	X	X		Construct exon islands with exonic reads to determine localize final splice junctions. TopHat2 can handle indels	[40, 41]
TrueSight	X			X	X		Takes all possible splice junctions of a transcriptome from the aligning reads and learn a regression model to find best assignments for them	[187]
X-Mate	X			X		X	Upgraded version of RNA-Mate [163]. Designed for color-space reads but can align base-space reads too	[164]

L is for machine-based learning.

Chapter 4

Bisulfite Sequencing Reads Alignment

4.1 Introduction

DNA methylation modifies the nucleotide cytosine by the addition of methyl groups to its C5 carbon residue by DNA methyltransferases [188]. This modification can be inherited through cell division and it plays an important role in many biological processes, such as heterochromatin and transcriptional silencing [189, 190], imprinting genes [191], inactivating the X chromosome [192] and silencing of repetitive DNA components in healthy and diseased (including cancerous) cells [193, 194]. Methylation analysis can also be used to diagnose pre-natal Down's syndrome [195]. Thus, the genome-wide methylation profiles of different tissues are important to understand the complex nature and effects of DNA methylation.

In the past decade, quantum leaps have been made in the development of sequencing technologies by vendors such as Illumina-Solexa and Applied BioSystems (AB)-SOLiD.

These can generate millions of short reads at a lower cost compared to traditional Sanger methods [65, 196-199]. Bisulfite (BS) treatment converts unmethylated cytosines (Cs) to uracils (which are then amplified by PCR as thymine (T) without affecting the other nucleotide bases and methylated cytosines [200]. Next-generation sequencing coupled with bisulfite treatment enables us to produce a methylome of a genome at single base resolution and low cost.

4.2 Related Work

One important step in calling methylation of a genome is to map BS reads. Mapping of BS reads is different from that of ChIP-Seq and RNA-Seq data since the non-methylated Cs are converted to Ts by BS treatment and subsequent PCR. The BS reads are difficult to map to the reference genome due to the high number of mismatches between the converted Ts and the original Cs. For mapping Illumina BS reads, the pioneering published methods are BSMAP [85] and RMAP [83]. BSMAP aligns a BS read to the reference genome by first enumerating all C-to-T combinations within a user-defined length k seed of the reads; then, through hashing, BSMAP aligns the seeds onto the genome and putative alignments are extended and validated with the original reads. After this step, BSMAP can output an unambiguous hit for each read, if available. BRAT [79] uses a similar strategy as BSMAP. It converts the reference genome into a TA reference and a CG reference (each converted reference uses one bit per base). Using a 36-mer hash table, BRAT aligns the first 36 bases of every read and its 1-neighbors on the two converted references to identify possible alignments. RMAP uses layered seeds as a bit-mask to select a subset of the bases in the reads and constructs a hash table to index all the reads. However, these seed-hash-based approaches are slow.

Subsequently, several methods were proposed to map BS reads onto the converted genomes. MethylCoder [201] surfaced as a BS read mapper that uses GSNAP [170] to do a primary mapping of *in silico* converted reads (that is, all Cs in the reads are converted to Ts) onto a converted reference genome (that is, all Cs in the genome are converted to Ts). Those reads that fail to map onto the converted genome will be remapped again in their original forms onto the original reference. BS-Seeker [81] and Bismark [78] use a similar conversion strategy as BSMAP except that they align the reads with Bowtie [92] and unique hits are found by a seed-then-extend methodology. (Note that every tool has its own uniqueness criterion. A tool will denote a read to have a unique hit if it finds exactly one occurrence of the read in the reference genome.) Both methods trade accuracy for efficiency.

AB-SOLiD color reads are different from Illumina reads since they encode every pair of bases with four different colors. (For more details on this sequencing technology and how it differs from sequencing by synthesis, see [18, 58, 202, 203].) Unlike BS mapping of Illumina reads onto converted genomes, mapping BS color reads onto converted genomes produces many mismatches when the regions are highly methylated [204]. This also causes a dramatic decrease in the unique mapping rate and unbiased measurements of hypomethylation sites. In addition, a single color error in a read will lead to incorrect conversions throughout the rest of the read (Figure 4.1a,b). Although *in silico* conversion of Cs to Ts guarantees unbiased alignments in base space, this is not preferred for color reads.

SOCS-B [205] and B-SOLANA [86] were developed to map BS color reads. SOCS-B splits a color read into four parts and tries to get hits for any combination of two parts via an iterative Rabin-Karp approach [206]. SOCS-B uses a dynamic programming (DP)

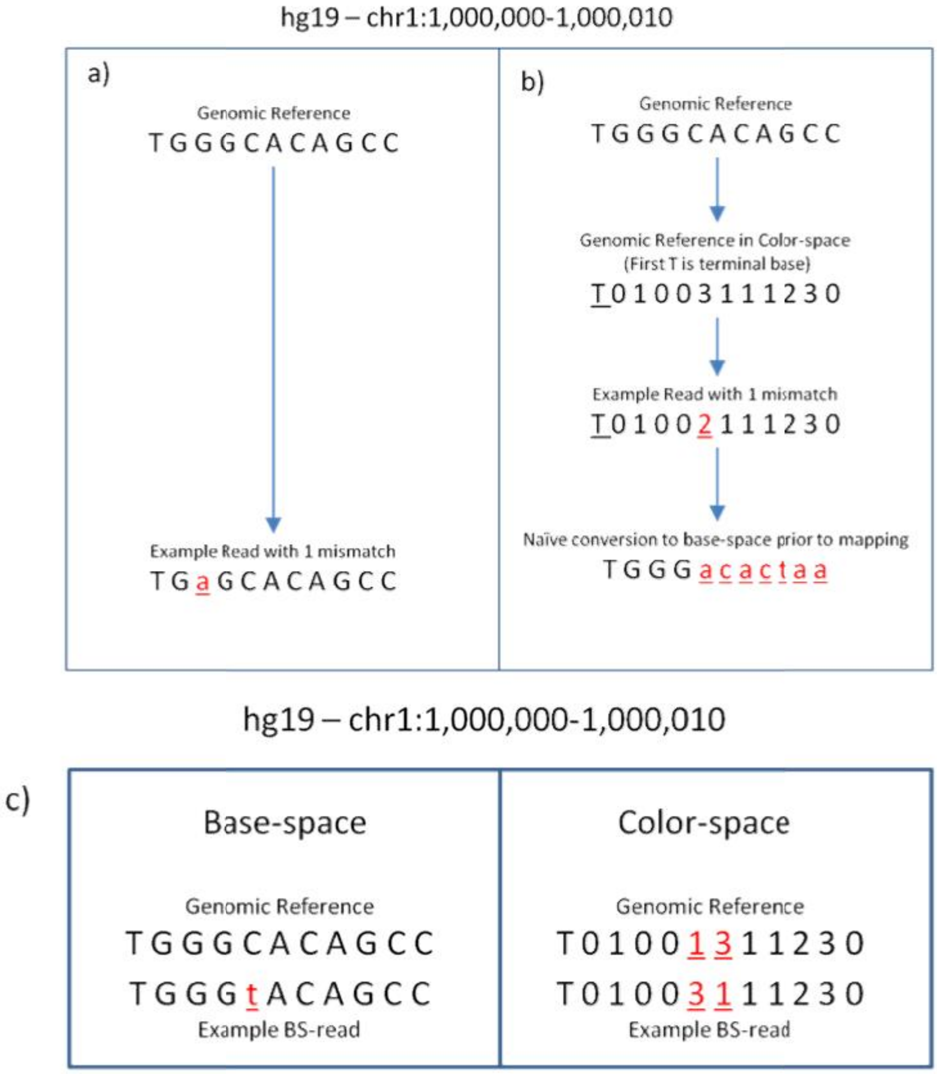


Figure 4.1. (a,b) Base call error simulation in Illumina and SOLiD reads reflecting one mismatch with respect to the reference from which they are simulated in their respective base- and color-space. (b) A naïve conversion of color read to base space, for the purpose of mapping against the base space reference, is not recommended as a single color base error will introduce cascading mismatches in base space. (c) A BS conversion in base space will introduce two adjacent mismatches in its equivalent representation in color space.

approach to convert an aligned read to the aligned portion of the reference genome. The conversion starts with all possible four nucleotides as the pseudo-terminal base (rather than just the terminal base from the read). Subsequently, the sub-strings of the four translations are used to generate partial hashing seeds that are then mapped onto the hashed reference genome. However, the running time of SOCS-B is long and the unique mapping rate is too low to be practical. B-SOLANA improves speed and unique mapping rate by aligning against both fully converted and non-CpG converted references simultaneously with Bowtie. The final hits are determined by checking their number of mismatches.

A recent *Nature* review paper [204] reported that Bismark and BS-Seeker are the most recent published methods for mapping BS base reads whereas B-SOLANA is the most recent published method for mapping BS color reads. This review also highlighted the main challenges to develop methods that can map reads unbiasedly and to improve unique mapping rates for mapping color reads.

4.3 Results

BatMeth (Basic Alignment Tool for Methylation) was developed by us to address the issues of efficiency and accuracy on mapping BS reads from Illumina and BS color reads from SOLiD. Unlike existing algorithms, BatMeth does not map the BS reads in the initial stage. Instead, BatMeth counts the number of hits of the BS reads to remove spurious orientations of a read. This idea has significantly sped up the mapping process and has also reduced the number of false positives. When dealing with color reads, BatMeth reduced bias on hypomethylation measurements with high initial mismatch scanning. BatMeth also employed a DP conversion step for the color reads to account for BS mismatch accurately and an incremental processing step to produce higher unique mapping rates and speed (refer to the Materials and methods section for details).

4.3.1 Evaluated programs and performance measures

In order to evaluate the performance of our pipeline, we have tested the following programs: BSMAP, BS-Seeker, and Bismark for base-space mapping; and SOCS-B and B-SOLANA for color-space mapping. BS-Seeker and Bismark only output unique hits for each read. BSMAP, SOCS-B and B-SOLANA will output at most one hit per read, with a flag to indicate if a hit is unique. Some reads can map to multiple genomic locations and since a read can only come from one origin, retaining such non-unique mappings will affect the accuracy of downstream analysis such as unbiased methylation site calls. To avoid the problem of wrong methylation calls, all six programs were thus compared with their unique mapping rates.

All our experiments were run on a server equipped with an Intel Xeon E7450 @ 2.40GHz and 128 GB of RAM. We allowed the same mismatch number and CPU threads on all the compared programs in our experiments. Other parameters were kept at default (see Section 1 of Additional file 1 for the choice of parameters used).

We have compared the performance of BatMeth with recent stable versions of BSMAP (2.4.2), BS-Seeker, Bismark (0.5.4), SOCS-B (2.1.1) and B-SOLANA (1.0) using both simulated and real data sets (BS-Seeker, Bismark and B-SOLANA used Bowtie 0.12.7 in our experiments). With simulated Illumina and SOLiD reads, BatMeth (default mode) recovered the highest number of hits, has the lowest noise rate and is the fastest among the compared programs. BatMeth is also able to produce better unbiased results than the other programs by comparing the detected methylation levels in different genomic contexts over simulated data sets (Illumina and SOLiD reads) of different methylation levels. With a paired-end library, we show the specificity of our Illumina results by counting the pairs of concordant paired reads that fall within the expected insert size of the library. With a directional library, we indicate the specificity of our results with direction-specific information. In summary, BatMeth is an improved BS mapper in terms of speed, recovery rate and accuracy, and, in particular, has addressed the main challenges of mapping color reads identified in [204].

We have not included RMAP in our comparisons as it only performs biased mapping in a non-CpG context. MethylCoder was also not included because a newer variant of it, namely B-SOLANA, has been released (MethylCoder's release notes mention that it is now deprecated due to the release of B-SOLANA). BRAT was considered impractical as it only considers one base error in the first 36 bp of a read and therefore was not included in our experiments.

Below, we define 'recovery' to be the portion of the unique hits recovered by the programs. We also define 'accuracy' to be the portion of the recovered hits that are correct. All recorded timings are wall clock times. A 'hit' is a genomic location to which a read is aligned. Lastly, due to sequencing errors and BS mismatches, we allow k (>0) mismatches when mapping a BS read onto a reference. A genomic location is deemed to be unique for a read if it is the only location with the lowest number of mismatches with respect to the read.

4.3.2 Evaluation on the simulated Illumina data

We generated 1 million reads, each 75 bp long, which were randomly simulated from the human genome hg19 using the simulator found in RMAP-bs [207]. The data set was built by allowing a maximum of three mismatches per read. Each C in the simulated read, regardless of its context, was BS converted at a uniform rate of 97%. We benchmarked BatMeth and the other methods, BSMAP, BS-Seeker and Bismark, on this data set (see Section 1.1 of Additional file 1 for parameters used). Since the original coordinates in the simulated reads are known, we can evaluate the accuracy of all the programs by comparing their outputs with the original coordinates. We mapped the reads onto the reference allowing at most three mismatches. BatMeth recovered the most number of true positives and the lowest number of false positives and is the fastest program, as shown in Figure 4.2a.

We further illustrate that BatMeth can achieve better unbiased methylation calls than the best published method, Bismark, by replicating the experimental settings of Figure 4.2b in [204]. We used the same simulator, Sherman [208], the same number of reads (1 million), the same length of read (75 bases) and the same reference genome (NCBI37) for this comparison. We used Sherman to simulate 11 sets of data, from 0% to 100% of BS conversion in increments of 10%. Sherman emulates BS conversion by converting all Cs regardless of their genomic context with a uniform distribution. No non-BS mismatches were allowed in the reads, during the scanning phase, for both BatMeth and Bismark. The results produced by Bismark show exactly the same trends as the graph that was presented in [204]. Table 4.1 presents the performance of BatMeth and Bismark in terms of mapping efficiency, detected methylation levels in different genomic contexts from various *in silico* methylation rates in different contexts (CG, CHG and CHH genomic contexts, where H stands for base A/C/T only). BatMeth has an average of approximately 1.1% better mapping efficiency and about twice the accuracy as Bismark in estimating methylation levels of Cs from different genomic contexts with different initial methylation levels.

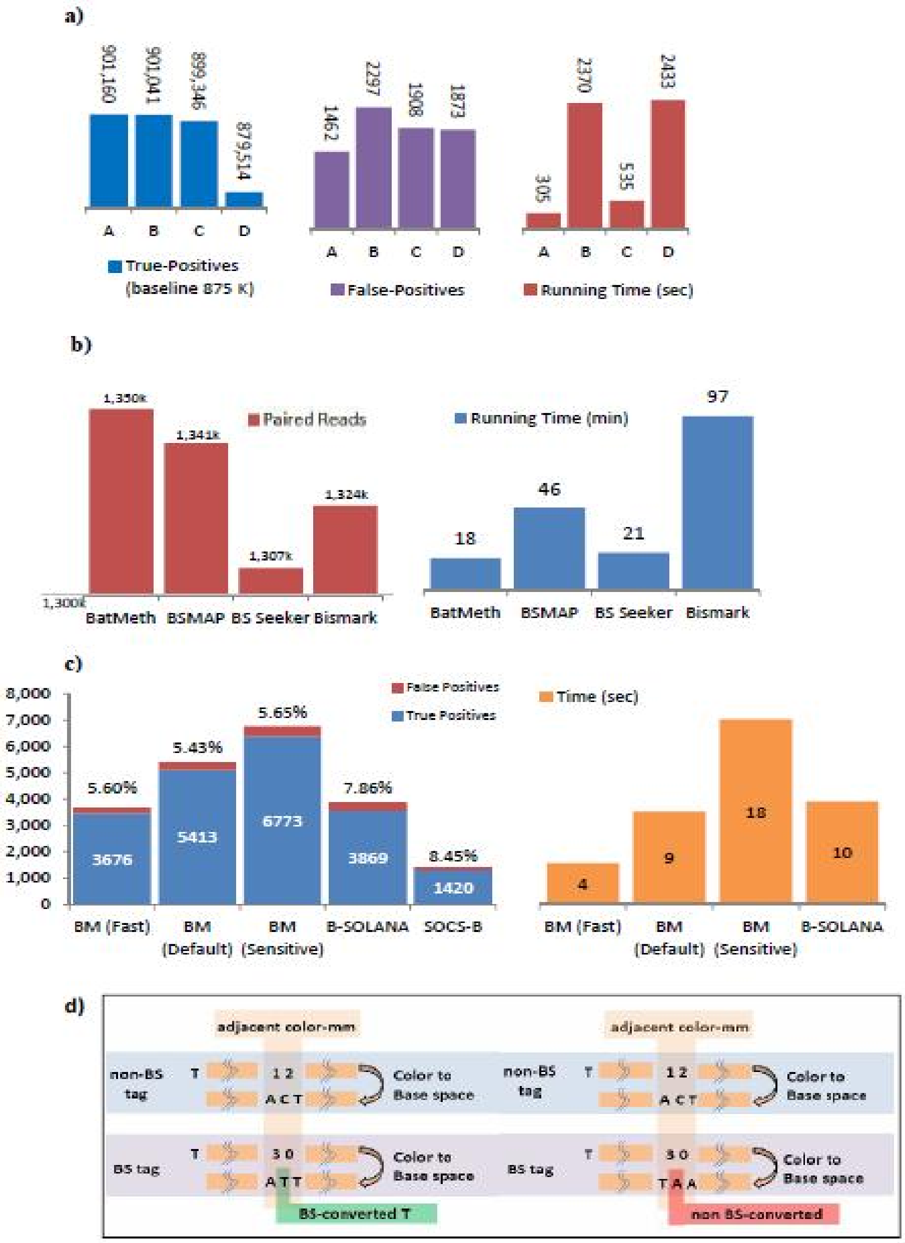


Figure 4.2. Benchmarking of programs on various simulated and real data sets (a) Benchmark results of BatMeth and other methods on the simulated reads: A, BatMeth; B, BSMAP; C, BS-Seeker; D, Bismark. The timings do not include index/table building time for BatMeth, BS-Seeker, and Bismark. These three programs only involve a one-time index-building procedure but BSMAP rebuilds its seed-table upon every start of a mapping procedure. (b) Insert lengths of uniquely mapped paired reads and the running times for the compared programs. (c) Benchmark results on simulated SOLiD reads. Values above the bars are the percentage of false positives in the result sets. The numbers inside the bars are the number of hits returned by the respective mappers. The graph on the right shows the running time. SOCS-B took approximately 16,500 seconds and is not included in this figure. (d) BS and non-BS induced (SNP) adjacent color mismatches.

Table 4.1. Comparison of mapping efficiencies and estimation of methylation levels in various genomic contexts

BatMeth (%)				Bismark (%)				Oracle BS rate (%)
Mapping efficiency	CG	CHG	CHH	Mapping efficiency	CG	CHG	CHH	
94.2	0.0	0.0	0.0	91.1	0.0	0.0	0.0	0.0
94.0	10.0	10.0	10.0	92.1	10.0	10.0	10.0	10.0
93.9	20.0	20.0	20.0	92.4	20.0	20.1	20.0	20.0
93.8	30.0	30.0	30.0	92.5	29.9	30.0	30.0	30.0
93.6	39.9	40.0	40.0	92.5	40.0	40.0	40.0	40.0
93.5	50.0	50.0	50.0	92.6	50.0	50.0	50.0	50.0
93.4	60.0	60.0	60.0	92.6	60.0	60.1	60.0	60.0
93.2	70.0	70.0	70.0	92.7	70.0	70.0	70.0	70.0
93.0	79.9	80.0	80.0	92.6	79.9	80.0	80.0	80.0
92.8	90.0	90.0	90.0	92.6	90.1	90.0	90.0	90.0
92.6	100	100.0	100.0	92.6	100.0	100.0	100.0	100.0

Methylation levels in various genomic contexts, such as CG, CHG and CHH (H is A/C/T only), are called by BatMeth and Bismark and validated against the oracle BS rate used in Sherman.

4.3.3 Evaluation on the real Illumina data

We downloaded about 850 million reads sequenced by Illumina Genome Analyzer II (Gene Expression Omnibus (GEO) accession number GSE19418) [209] on H9 embryonic stem cells. Since BSMAP is not efficient enough to handle the full data set, 2 million paired-end reads were randomly extracted from one of the runs in GSE19418 for comparative analysis with BSMAP. Reads were observed to have a lot of Ns near the 3' end and were trimmed down to 51 bp before being mapped onto hg19 with at most two mismatches per read (see Section 1.2 of Additional file 1 for parameters used).

For this sample data set, BatMeth mapped 1,518,591 (75.93%) reads uniquely compared to 1,511,385 (75.57%) by BSMAP, 1,474,880 (73.74%) by BS-Seeker and 1,498,451 (74.92%) by Bismark. Out of all the hits reported by BatMeth, 1,505,190, 1,464,417 and

1,481,251 mapped loci were also reported by BSMAP, BS-Seeker and Bismark, respectively. BatMeth found 13,401, 54,174 and 37,340 extra hits when compared to BSMAP, BS-Seeker and Bismark, respectively. BSMAP, BS-Seeker and Bismark also found 6,195, 10,463 and 17,220 extra hits, respectively, when compared to our result set.

Next, we mapped the two reads of every paired-end read independently to investigate the mapping accuracy of the compared programs. Since the insert size of this set of paired-end reads is approximately 300 bp, a pair of partner reads can be expected to be mapped correctly with a high probability if they are mapped concordantly within a nominal distance of 1,000 bp. The high number of such pairable reads (Figure 4.2b) indicates that BatMeth is accurate. Figure 4.2b also shows that BatMeth is fast.

Table 4.2. Comparison of speed and unique mapping rates on three lanes of human BS data

	Number of	Unique mapping (%) [a]		Running time (minutes) [a]	
Read file	reads	BatMeth	BS-Seeker	BatMeth	BS-Seeker
SRR019048	15,331,851	37.4	37.2	30	87
SRR019501	7,217,883	44.7	44.5	16	41
SRR019597	5,943,586	58.2	58.1	13	37

[a]Threshold of two mismatches used.

We have also downloaded approximately 28.5 million reads sequenced by Illumina Genome Analyzer II on the human H1 embryonic cell line (GEO accession numbers SRR019048, SRR019501 and SRR019597) [81]. We only compared BatMeth with BS-Seeker since BSMAP and Bismark are too slow (see Section 1.3 of Additional file 1 on parameters used). Furthermore, Krueger and Andrews [78] mention that Bismark is both slower and less likely to report unique hits than BS-Seeker. Table 4.2 shows the unique mapping rates and running times of BatMeth and BS-Seeker. In summary, BatMeth achieved the best mappability rate, lowest estimated false positive rate and was the fastest on real Illumina data.

4.3.4 Evaluation on the simulated SOLiD data

We generated 10,000 simulated reads, each having 51 color bases, that were randomly extracted from chromosome 1 of UCSC hg19 using the simulator from RMAP-bs [207]. RMAP-bs was used to convert the Cs in the reads, regardless of its context, to Ts at a uniform rate of 97% to simulate BS conversions. In addition, for each read, zero to two non-BS base mismatches were introduced with equal chance before the read was converted to color space. Lastly, sequencing errors were added at a uniform rate of 5% to the reads.

The simulated color reads were mapped using BatMeth, SOCS-B and B-SOLANA allowing resultant unique hits to have at most three mismatches. Precisely, BatMeth and SOCS-B allowed at most three non-BS mismatches while B-SOLANA did not discount BS mismatches (see Section 1.4 of Additional file 1 for parameters used). Figure 4.2c summarizes the results of the three programs together with the verification against the oracle set. BatMeth gave many more correct hits and fewer wrong hits than both SOCS-B and B-SOLANA. BatMeth can be made to offer a flexible tradeoff between unique mapping rates and speed. In the 'default' mode, BatMeth was found to be more sensitive (approximately 15%) and faster (approximately 10%) than the most recent published B-SOLANA. In the 'sensitive' mode, BatMeth was found to be more sensitive (approximately 29%) and slower (approximately two times) than B-SOLANA. In addition to producing approximately 15% to 29% more correct hits, BatMeth had a precision of 94.5% while that of B-SOLANA and SOCS-B was 92.1% and 91.5%, respectively. These statistics show that BatMeth is an accurate mapper for color reads.

To illustrate that BatMeth can achieve better unbiased methylation calls for color reads than the best published method, B-SOLANA, we replicated the experimental settings of Figure 4.2c in [204] to compare the two programs; we used the same simulator (Sherman), the same number of reads (1 million), the same length of read (75 bp) and the same reference genome (NCBI37) for this comparison. We used Sherman to simulate 11 sets of data, from 0% to 100% of BS conversion at increments of 10%. Sherman emulates BS conversion by converting all Cs regardless of their genomic context with a uniform distribution. Default parameters were used for BatMeth and B-SOLANA. The graph produced by us for B-SOLANA shows the same trends as that presented in [204]. We further broke down the graphs as well as those in Figures 4.3a (BatMeth) and 3b (B-SOLANA), which show rates of methylation calling for various *in silico* methylation rates (0% to 100% at divisions of 10% of BS conversion) in different contexts (CG, CHG and CHH genomic contexts, where H stands for base A/C/T only) of the genomes, into separate series of data. Subsequently, we did a direct comparison between BatMeth and B-SOLANA to show that BatMeth is better than B-SOLANA in all contexts of methylation calling, namely, CG (Figure 4.3c), CHG (Figure 4.3d), CHH (Figure 4.3e) and non-unique mapping rates (Figure 4.3f). To be exact, BatMeth was approximately 0.7%, 0.7% and 2.2% more accurate than B-SOLANA in the methylation callings of the CG, CHG and CHH sites, respectively, and had an average of approximately 9.2% more non-unique mappings than B-SOLANA on the tested data sets.

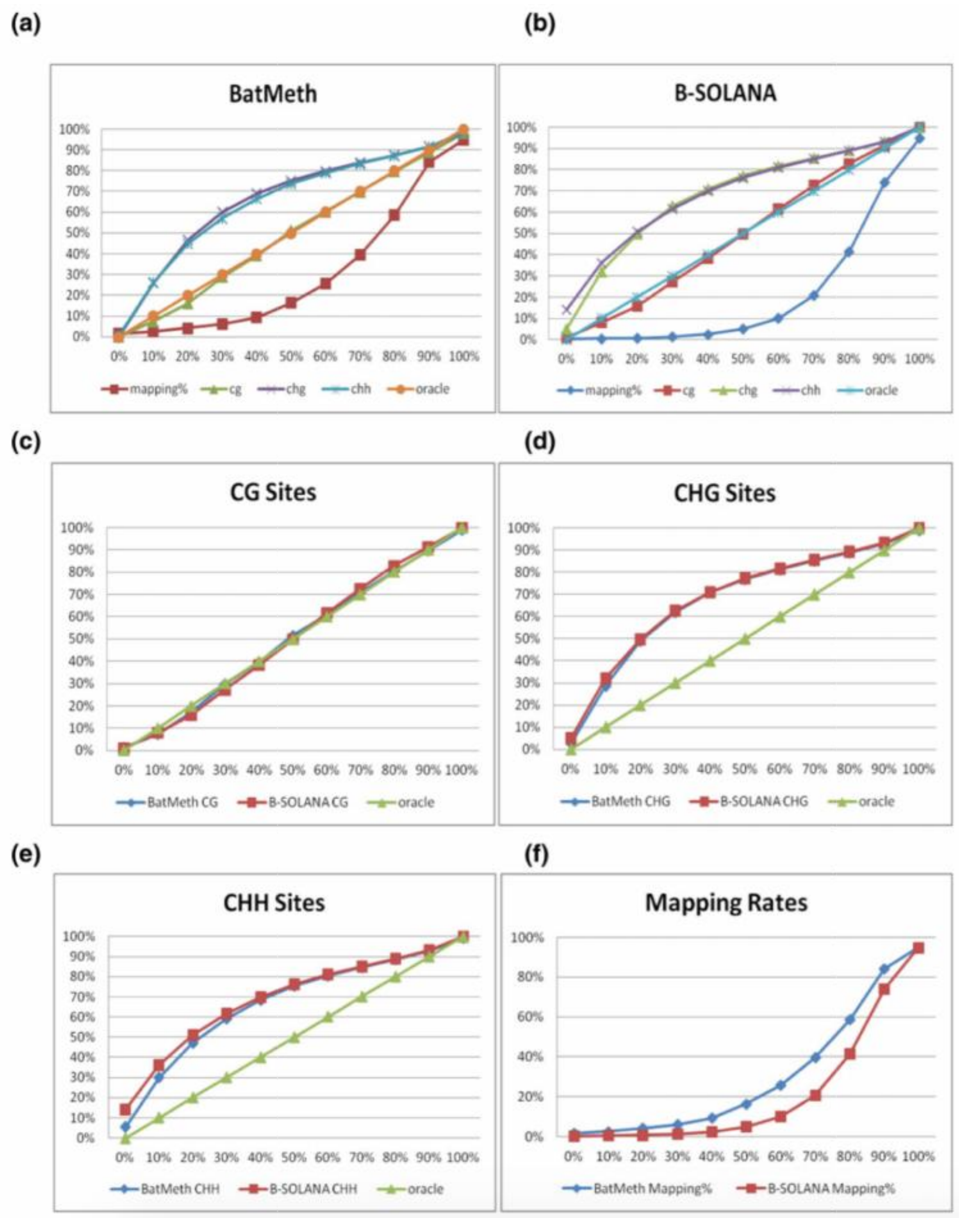


Figure 4.3. A total of 106, 75 bp long reads were simulated from human (NCBI37) genomes. Eleven data sets with different rates of BS conversion, 0% to 100% at increments of 10% (context is indicated), were created and aligned to the NCBI37 genome. (a-e) The x-axis represents the detected methylation conversion percentage. The y-axis represents the simulated methylation conversion percentage. (f) The x-axis represents the mapping efficiency of the programs. The y-axis represents the simulated methylation conversion percentage of the data set that the program is mapping. (a,b) The mapping statistics for various genomic contexts and mapping efficiency with data sets at different rates of BS conversion for BatMeth and B-SOLANA, respectively. (c-e) Comparison of the methylated levels detected by BatMeth and B-SOLANA in the context of genomic CG, CHG and CHH, respectively. (f) Comparison of mapping efficiencies of BatMeth and B-SOLANA across data sets with the described various methylation levels.

4.3.5 Evaluation on the real SOLiD data

We downloaded about 495 million reads sequenced by AB SOLiD system 3.0 (Sequence Read Archive (SRA) accession number SRX062398) [199] on colorectal cancer. Since SOCS-B is not efficient enough to handle the full data set, 100,000 reads were randomly extracted from SRR204026 to evaluate BatMeth against SOCS-B and B-SOLANA. The mismatch threshold used was 3 (see Section 1.5 of Additional file 1 for parameters used).

Table 4.3. Unique mapping rates and speed on 100,000 real color reads

SRR204026	Unique mapping (%)[a]	Estimated noise (%)[b]	Timing
BatMeth (fast)	39.6	0.47	77 s
BatMeth (default)	45.8	0.94	247 s
BatMeth (sensitive)	52.1	1.75	521 s
B-SOLANA[c]	37.4	2.06	130 s
SOCS-B[d]	28.3	4.55	~71 h

[a] We tabulated the unique mapping rates of the 100,000 reads. [b]The error rates are estimated from the number of reverse-strand mappings as stated by Equation 2 in Materials and methods. [c]Note that 3.26% of B-SOLANA's resultant reads are double-counted as B-SOLANA reported two hits for them. One of the two hits is assumed to be correct for the estimation of the noise rate of B-SOLANA. [d]Reverse-strand mapping is allowed by enabling G-A transitions in SOCS-B. BatMeth fast, default, and sensitive modes were run with -n0-N3, -n0-N4, -n0-N5 as parameters, respectively.

Table 4.3 compares the unique mapping rates and running times between BatMeth, SOCS-B and B-SOLANA. Note that BatMeth always has a higher unique mapping rate (from 39.6% to 52.1%; from fast to sensitive mode) than the next best method, B-SOLANA with 37.4%. At the same time, BatMeth maintained low rates of noise (from 0.47% to 1.75%; from fast to sensitive mode). Hence, it is still more specific than the other programs. In terms of running time, BatMeth fast mode is approximately 1.7 times faster and BatMeth sensitive mode is approximately 4 times slower than B-SOLANA. It was also observed that 3.26% of the resultant hits from B-SOLANA are duplicated; some of the reads were given two hit locations as B-SOLANA traded speed for checking the uniqueness of hits.

Based on the experiments performed, BatMeth's memory usage peaked at 9.3 GB (approximately 17 seconds of load time) for Illumina reads and 18.8 GB (approximately 35 seconds of load time) for color reads while BSMAP and BS-Seeker peaked at 9+ GB and Bismark peaked at 12 GB. SOCS-B peaked at 7+ GB and B-SOLANA peaked at 12 GB. Parameters used for all experiments are recorded in Additional file 1. In summary, the experiments in this section show that BatMeth is the fastest among all the compared programs. Furthermore, BatMeth also has the highest recovery rate of unique hits (exclusive of false positives) and the best accuracy among all the compared programs.

4.4 Materials and Methods

4.4.1 Methods for base reads

4.4.1.1 Problem definition and overview of the method

The problem of mapping BS reads is defined as follows. A BS treatment mismatch is defined as a mismatch where the aligned position is a T in the read and the corresponding position in the reference genome is a C. Given a set of BS reads, our task is to map each BS read onto the reference genome location, which minimizes the number of non-BS mismatches.

The algorithm of BatMeth is as follows. BatMeth starts off by preparing the converted genome and does a one-time indexing on it. Next, low complexity BS reads are discarded; otherwise, we obtain counts of the hits for BS reads and discard the hits according to list filtering. After this, each of the retained hits will be checked for BS mismatches by ignoring C to T conversions caused by the BS treatment. BatMeth reports the unique hit with the lowest non-BS mismatches for each read. Figure 4.4a outlines the algorithm and we discuss the novel components that aid BatMeth to gain speed and accuracy below.

4.4.1.2 Converted genome

Similar to BS-Seeker and Bismark, we prepare a converted reference genome with all Cs converted to Ts. Since the plus and minus strands are not complementary after Cs are converted to Ts, we have to create two converted references where one is for the plus strand and the other is for the minus strand. Burrows-Wheeler transform (BWT) indexing of the two new converted references is done before the mapping.

4.4.1.3 Low complexity BS reads

BatMeth does not map BS reads with low complexity. The complexity of the raw read is computed as Shannon's entropy, and raw BS reads with a differential entropy $H < 0.25$ are discarded. In BatMeth, differential entropy is estimated from the discrete entropy of the histogram of A/C/G/T in a read. Depending on the design of the wet-lab experiment, the amount of reads being discarded by this entropy cutoff varies. In our experiments on Illumina reads, approximately 0.5% of the reads were discarded.

4.4.1.4 Counting hits of BS reads and list filtering

For those reads that pass the complexity filter, we first convert all Cs to Ts and map them against the converted genomes. In contrast to existing methods, BatMeth does not obtain the best or second best hits (for example, BS-Seeker and Bismark) from each possible orientation of a converted read and reports the lowest-mismatch locus to be the resultant hit for a read. In the case of hyper-methylation, the correct hit may not be the best or second best hit as it might contain more mismatches. Thus, this approach will miss some correct solutions. BatMeth also does not enumerate all hits like BSMAP, which is slow. Instead of mapping the reads directly, BatMeth counts the number of hits where the read or its reverse complement can occur on the two converted genomes using an in-house short read mapper, BatMis Aligner [210]. Table 4.4 shows the four ways of aligning the converted reads onto the converted genomes, which yield four counts of hits.

Table 4.4. Possible ways to map a BS read onto the converted genome

	Reference (C T)	RC reference (C T)
Read (C T)	Count 1	Count 2
RC Read (C T)	Count 3	Count 4

RC, reverse-complement.

Out of the four counts on the four lists, only one list contains the true hit. List filtering aims to filter away those spurious lists of hits (represented by the counts) that are unlikely to contain the true hit. Note that a read can appear to be repetitive on one strand but unique on the opposite strand of the DNA. Hence, if a list has many hits (by default the cutoff is set to be 40 hits) with the same number of mismatches, we discard such a list since it is likely to be spuriously reported for one strand of the reference genome. Another reason for rejecting such lists is that they may contain hits that may be of the same mismatch number as the hit that is unique on the opposite strand, rendering all hits as ambiguous.

Apart from improving the uniqueness of the putative resultant hit among all reported hits of a BS read, filtering also reduces the number of candidate hits that need to be checked.

a)

```
1.  Prepare the converted Reference Indexes for both plus and minus strands.
2.  For each input read do
3.      Prepare the plus and minus conversions of the read
4.      Count the number of hits using 4 possible ways to map the converted reads on the Converted Genome
5.      Using List Filtering, we filter the lists whose number of hits > cutoff
6.      For each hit in the unfiltered lists, compute the number of mismatches ignoring the BS-treatment mismatches.
7.      If the least mismatch hit is unique then
8.          Report its location.
9.      Else
10.         Report it as non-unique.
11.     EndIf
12. EndFor
```

b)

```
1.  Prepare 4 Reference Indexes for the two fully-converted color genomes and the two non-CpG converted color genomes.
2.  For every read do
3.      Count the number of hits for 2 possible ways to map the read and its reverse on the fully-converted color genomes
4.      Apply List Filtering on the counts obtained from Step 3.
5.      Apply Mismatch Stage Filtering to the unfiltered list from Step 4.
6.      Apply Conversion of Bisulfite Color reads to Base reads to the hits from Step 5.
7.      Determine the Color Mismatch Counts for the hits on the ordered hits from Step 6.
8.      If the least mismatch hit is unique then
9.          Report it. Goto Step 14.
10.     ElseIf the least mismatch hit is non-unique
11.         Reported it as non-unique. Goto Step 14.
12.     ElseIf no hits found on fully-converted color genomes then
13.         Repeat Steps 3 to 14 with non-CpG-converted color genomes
14.     EndIf
15. EndFor
```

Figure 4.4. Outline of the mapping procedure. (a) Mapping procedure on Illumina BS base reads. (b) Mapping procedure on SOLiD color-space BS reads.

This improves the efficiency of the algorithm. For example, consider the simulated BS-converted read 'ATATATATGTGTATATATATATATATATATATGTGTATATATATGTGTGTATA TATATATA TATATATGTATATAT' being mapped onto the converted hg19 genomes as discussed earlier. We obtained four counts of 1, 0, 40 and 40 hits by mapping the converted reads onto the converted genomes. The last two lists are filtered away since they have too many hits, leaving us to check only one hit instead of 81 for BS mismatches. Since the data are simulated, the unfiltered hit is found to be the correct unique hit for this read, which the other mappers cannot find.

Table 4.5. Cutoffs for list filtering on simulated reads from the Results section

List size	Mismatch counting in seconds[a]	Correct hit	Wrong hit	Total hit
20	136	901,164	1,516	902,680
40	165	901,160	1,462	902,622
60	191	901,165	1,454	902,619
100	279	901,166	1,448	902,614
200	475	901,166	1,447	902,613
500	1,197	901,167	1,450*	902,617
1,000	2,942	901,167	1,450*	902,617

Asterisks indicate increased false-positives produced with large list filtering cutoffs.

Table 4.5 shows the effect of using list filtering on the same set of simulated data from the evaluation on the simulated Illumina reads. We ran BatMeth with different cutoffs for list filtering and we can see that the time taken increased linearly with increasing cutoffs for list filtering while sensitivity and accuracy dropped. With large cutoffs such as ≥500 (marked by asterisks in Table 4.5), the number of wrong hits increased while sensitivity still continued to drop. Thus, we have chosen a cutoff of 40 for a balance of speed, sensitivity and accuracy. (Disabling list filtering will cause BatMeth to check through all the reported candidate locations for a read and will slow BatMeth down by approximately 20-fold, as shown in Table 4.5.)

4.4.2 Methods for color reads

4.4.2.1 Overview of the method

Due to the di-nucleotide encoding and sequencing errors in SOLiD color reads, a naïve conversion from color space to base space is hardly possible without errors. As a color error in a read will introduce cascading base-space errors, we cannot use the method described in 'Methods for base reads' above to map BS color reads. This section describes how we aim to map each BS color read uniquely to the reference genome while minimizing the number of non-BS treatment mismatches.

The algorithm of BatMeth is as follows. BatMeth starts by preparing the converted genome and non-CpG converted genome and does a one-time BWT indexing on them. For every color read, we do a 'counting hits of BS color reads' for it on the references and discard the list of hits according to list filtering. After applying mismatch stage filtering, the unfiltered hits are converted to base space as described in 'Conversion of bisulfite color reads to base reads' below to allow for the checking of BS mismatches. The color mismatch count for the retained hits is then determined and the unique locus with the lowest mismatch count reported; otherwise, no hits will be reported for this read. We have also utilized additional heuristics, such as fast mapping onto two indexes and handling hypo- and/or hyper-methylation sites to speed up and improve the accuracy of BatMeth, which we discuss below. All the components, namely list filtering, mismatch stage filtering, conversion of BS color reads to base reads, color mismatch count, fast mapping onto two indexes and handling hypo- and/or hyper methylation sites differ from existing methods. Figure 4.4b outlines the algorithm and shows how the components are assembled for SOLiD color-space BS read mapping.

4.4.2.2 Non-CpG converted genome

The reference genome and its reverse-complement were first prepared by converting all its Cs to Ts as described in the base reads mapping procedures; then, the two converted genomes are encoded into color space. These two genomes are called fully converted color genomes. In addition, the reference genome and its reverse-complement are similarly converted except that the Cs in CpG are left unchanged. We call these the non-CpG converted color genomes. Finally, the BWT indexes for these four color genomes are generated.

In the algorithm, the BS color reads will be mapped to the fully converted color genomes to identify unique hits first; if this fails, we will try to map the reads onto the non-CpG converted color genomes and BatMeth will label which reference a hit is from.

The reason for using the non-CpG converted genome is that the conversion step for BS color reads is different from that for Illumina. In Illumina reads, the C-to-T mismatches between the raw BS reads and the reference genome are eliminated by converting all Cs to Ts in both the reads and the reference genomes. However, we cannot make such a conversion in BS color reads as we do not know the actual nucleotides in the reads. Based on biological knowledge, we know that CpG sites are expected to be more methylated [211]. Hence, such conversion reduces the number of mismatches when the color reads are mapped onto the reference genome in color space. This aids in gaining coverage in regions with high CpG content. Thus, BatMeth maps BS reads to both hyper- and hypo-methylation sites.

4.4.2.3 *Counting hits of BS color reads and list filtering*

Unlike sequencing by Illumina, SOLiD only sequences reads from the original BS-treated DNA strands. During PCR amplification, both strands of the DNA are amplified but only the original forward strands are sequenced. Subsequently, during the sequencing phase, reverse-complement reads are non-existent as a specific 5' ligated P1 adaptor is used. As such, matches to the reverse-complement of the BS-converted reference genome are invalid.

In other words, although a BS color read has four possible orientations to map on the non-CpG converted color genomes (or the fully converted color genomes), only two orientations are valid as opposed to the four orientations in the pipeline on Illumina reads (Table 4.6). As opposed to the mapping of Illumina reads, it is not preferred to do a naïve conversion of color reads to base space prior to mapping. Figure 4.1a shows that a single base call error in an Illumina read will introduce one mismatch with respect to the reference. However, Figure 4.1b shows that a single base color call error in a color read will introduce cascading base mismatches instead of just one color mismatch if we are to map the color read as it is onto the reference in color space.

Table 4.6. Possible ways to map a BS color read onto the converted color genome

	Reference (C T)	RC reference (C T)
Read	Count 1	Invalid
RC read	Invalid	Count 4

RC, reverse-complement.

Thus, we will need to do a primary map onto a converted genome with a higher mismatch parameter (by default, 4) than what we usually use for Illumina BS reads as a BS mismatch will introduce two adjacent color mismatches (see Figure 4.1c for an example of BS-induced adjacent color mismatches). Similar to mapping Illumina reads, we count the number of possible hits from the two valid orientations. Then, the list filtering step is applied to filter the lists with too many hits (by default, more than 10). (Note that this property also helps us to estimate the noise rate; we discuss this further in 'Noise estimation in color reads' below.)

4.4.2.4 Conversion of bisulfite color reads to base reads

After the color BS reads are aligned to the reference genome, we can convert the color BS reads to their most-likely nucleotide equivalent representation. In the context of BS mapping, we discount all the mismatches caused by BS conversions.

We use a DP formulation as presented in [90] to convert color reads to base reads except that the costs for BS-induced mismatches have to be zeroed when the reference is C and the read is T. This conversion is optimal and we use the converted base read to check against the putative genomic locations from list filtering to interrogate all mismatches in the read to determine if they are caused by BS conversion, base call error or SNP.

4.4.2.5 Color mismatch count

After converting each color read to its base-space equivalent representation, we can calculate the number of base mismatches that are actually caused by BS treatment in the color read. Figure 4.2d shows two different types of adjacent color mismatches that are caused by BS conversion (left) and non-BS conversion (right). For BS-induced adjacent mismatches, we assign a mismatch cost of 0 to the hit. For non-BS-induced adjacent mismatches, we assign a mismatch cost of 1 to the hit.

To be precise, we consider a color read as C[1..L], where L is the read length, and let B[1..L-1] be the converted base read computed from the DP described previously and mm[i] as a mismatch at position i of C, which is computed using Equation 4.1. The mismatch count of C is computed as mm[1]+…+mm[L-1], where:

$$mm[i] = \begin{cases} 1, if\ C[i]\ and\ C[i+1] \text{ are color mismatches, } B[i] \text{ is non-BS mismatch} \\ 0, otherwise \end{cases} \tag{4.1}$$

4.4.2.6 Mismatch stage filtering

We have developed a set of heuristics to improve the rate of finding a unique hit among the set of candidate hits. First, we sort and group the initial hits by their number of color mismatches; then, we try to find a unique hit with the minimum non-BS-mismatch count within each group of hits.

As the bound of color mismatches is known, we can apply a linear time bucket sort to order all the candidate hits according to their mismatch counts. The group of initial mapping loci with the lowest mismatch number is recounted for their number of base mismatches using the converted read in base space obtained from the previously discussed DP formulation. If a unique lowest base mismatch hit exists among them, we report this location as unique for this read. Otherwise, we proceed to recount the base mismatches for the group of mapping loci with the next highest color mismatch count. We continue this procedure until a unique hit is found or until there are no more color-space mismatch groups to be examined. A unique hit must be unique and also minimizes the base mismatch counts among all previously checked hits in the previous groups.

Mismatch stage filtering enables us to check less candidate hits, which speeds up the algorithm. It also improves the unique mapping rate as there are less ambiguous hits within a smaller group of candidate hits.

When the above components are applied, the mapping rates on SOLiD data improve progressively as seen below. By using Equation 1 to count color mismatches, BatMeth was able to increase the number of unique mappings by approximately 9% and by employing mismatch stage filtering, unique mapping rate is approximately increased by another 3%. With this increase in unique mappings of approximately 12%, BatMeth had an estimated noise level of approximately 1% as based on Equation 2 while B-SOLANA and SOCS-B had estimated noise levels of approximately 2.06% and 4.55%, respectively, on the same set of 100,000 reads. These statistics agree with the results on the simulated data and indicate that BatMeth is capable of producing low-noise results.

4.4.2.7 Fast mapping onto two indexes

As mentioned in the 'Non-CpG converted genome' section above, we map BS color reads onto four converted references, two of which have their Cs converted to Ts at non-CpG sites and the other two have all their Cs converted to Ts. It was observed that mappings on both non-CpG converted and fully converted references highly coincide with each other with an approximately 95.2% overlap. Due to this observation, we try to map onto the fully converted reference first to give us a mapping to regions of hypo-methylation status. If there are no mappings found on the fully converted references, then BatMeth maps the same read again onto the non-CpG converted references, which biases hyper-methylation sites. This allows the simultaneous interrogation of canonical CpG hyper-methylation sites with reduced biased mapping on the fully converted genome. BatMeth also labels each hit with the type of converted references it was mapped to. Overall, this approach can save time by skipping some scanning of the non-CpG-converted references.

4.4.2.8 Handling hypo- and/or hyper-methylation sites

With prior knowledge of the methylation characteristics of the organism to be analyzed, different *in silico* conversions to the reference can be done and the best alignments can be determined from the combined set of results of different mapping runs. BatMeth uses two types of converted genomes to reduce mapping biases to both hyper- and hypo-methylation sets. Since the two sets of hits from the two genomes coincide to a large extent, we can save time by scanning a read on one genome with a much lower mismatch number than on the other genome.

BatMeth allows users to choose the mismatch number they want to scan on each of the two types of genomes. We now introduce M1 and M2 (capped at 5) as the mismatch numbers used in the scans against the fully converted and non-CpG-converted genomes, respectively. For the best sensitivity, BatMeth scans at M1 = M2 = 5 for both hyper- and hypo-methylation sites. For the highest speed, BatMeth scans at [M1 = 0, M2 = 3] and [M1 = 3, M2 = 0], which will perform biased mapping to hyper- and hypo-methylation at CpG sites, respectively. Figure 4.2c shows the results of running the various modes of BatMeth (fast, default and sensitive) on a set of 10,000 simulated color reads.

4.4.2.9 Noise estimation in color reads

To estimate noise rates, we map the real reads in their two possible orientations onto the genome. If a hit is found for a read from the original strands of the genome, we try to map the same read onto the complement strand of the genome too. If a lower mismatch hit can be found from the complement strand of the genome, then we mark the result for this read as noise. We use the proportion of marked reverse-complement unique mappings to estimate the noise level, given by Equation 4.2:

$$err = \frac{\#\ of\ reverse-complement\ mappings}{\#\ of\ mappings}$$

$$\mathrm{err} = \frac{\#\mathrm{ofreverse{-}complementmappings}}{\#\mathrm{ofmappings}} \qquad (4.2)$$

4.4.2.10 Handling ambiguous bases

For base reads, non-A/C/G/T bases are replaced by A so they will not affect the callings of methylation sites. Similarly, color reads with non-A/C/G/T bases are replaced with 0. Non-A/C/G/T bases on the reference genome are converted to A to avoid affecting downstream methylation callers. We have avoided converting them to random nucleotides as it may produce false hits in regions containing ambiguous bases. We mapped 1 million 75 bp reads and have seen reads being mapped to poly-N regions. This can be mostly attributed to the reduced alphabet size, from four to three, due to BS conversions.

4.5 Discussion

DNA methylation is an important biological process. Mapping the BS reads from next-generation sequencing has enabled us to study DNA methylation at single-base resolution. Our proposed method aims to develop efficient and accurate methods to map BS reads.

This study employed three methods to evaluate the performance of BS read mapping methods. The first method measured the ratio of correct and wrong unique unambiguous mappings. This method only applies to simulated data when the actual locations of the reads are known. For real data, the number of unambiguous mappings alone may not be a good criterion to evaluate accuracy (we can map more reads at a higher mismatch number, which results in lower specificity). The second method evaluated the accuracy using the number of reads that were mapped in consistent pairs, and can only be employed when paired-end read information is available. The third method used the directionality of the mapped reads from SOLiD sequencing. For the SOLiD reads, we mapped reads unbiasedly onto both forward and reverse directions of our reference genome. From the unambiguous mappings, we estimated the error rate of our unique mappings from the proportion of reverse direction unique mappings in the result sets. All these measures were used on different sets of simulated and real data and they suggest that BatMeth produces high quality mapping results.

For future work, our team will be working on more time-efficient data structures to better streamline our algorithm.

4.6 Conclusions

We report a novel, efficient and accurate general-purpose BS sequence mapping program. BatMeth can be deployed for the analysis of genome-wide BS sequencing using either base reads or color reads. It allows asymmetric BS conversion to be detected by labeling the corresponding reference genome with the hit. The components discussed in the Materials and methods section, such as list filtering, mismatch stage filtering, fast mapping onto two indexes, handling hypo- and hyper-methylation sites and other heuristics have offered increased speed and mappability of reads. In addition, BatMeth reduces biased detection of multiple CpG heterogeneous and CpH methylation across the whole reference by mapping onto both fully converted and non-CpG references and then labeling the reference to which the hits are from to aid biologists to discriminate each hit easily. Users can also choose to bias against either reference with varying mismatch scans. In assessing the uniqueness of a hit for BS color reads, BatMeth considers both strands of the DNA simultaneously while B-SOLANA considers both DNA strands separately. Hence, BatMeth has a stronger uniqueness criterion for hits as B-SOLANA may produce two hits for a read, one hit for each separate DNA strand. Lastly, BatMeth uses an optimal DP algorithm to convert the color read to base space to check for non BS mismatches.

Chapter 5

Gapped Alignment Problem

5.1 Introduction

Aligning sequencing reads to a reference genome is usually the first step in most of the genomic analysis. However, it is harder to align sequencing reads, which span across genomic variations back onto a reference genome as the whole-reads do not represent the reference genome exactly. As such, the sensitivity and accuracy of calling structural-variations (SV) can be affected. This motivated us to study the alignment of short reads which are associated not only with SV but also with single nucleotide variants (SNV) and insert-delete (Indel) variants.

Alignment tools were initially developed to align short reads allowing mismatches only. A number of such methods have been proposed, including SOAP [94], RMAP [104], Bowtie [92], PerM [145] and BatMis [210]. Although they are generally fast, they will miss capturing the wide spectrum of non-SNVs, which have been shown to represent 7-8% of human polymorphisms [212]. As increasing evidences show that indels are involved in a wide range of diseases [213], mismatch aligners are unsuitable to be used in the studies of such biologically important events.

Gapped aligners were proposed to align reads which span across indels. Existing gapped alignment methods mostly use the seed-and-extend approach by first aligning a part of the read to obtain preliminary seeding locations for the queried read. Different gapped aligners use different seeding techniques, including contiguous exact match seeds (BLAT [32], MegaBLAST , SeqAlto [102], YAHA [113], BWA-Mem [214]), mismatch-seeds (RMAP [104], Stampy [97]), spaced-seeds (Eland [103], PatternHunter [115], MAQ [74], ZOOM [107]) and q-gram filters (RazerS [120] , SHRiMP [105], MASAI [122]). Next, the seed locations are extended to the full length of the read, allowing for gaps, and these alignments are reported to the users.

5.2 Related Work

Current gapped aligners generally offer reasonable efficiency and accuracy. However, they assume that the parts of each read used to obtain the preliminary seeding locations to have a small number of mismatches in them. This assumption will bias the alignments. Our analysis shows that the majority of the reads with incorrect alignments are (1) reads whose seeds have many mismatches or (2) reads rescued by incorrect pairing the paired reads. (1) is an unfavorable consequence of the seed-and-extend approach. When there are indels or too many mismatches in the read, the aligners will misalign the read due to incorrect seeding of candidate locations. (2) is due to biased pairing methodologies that over-rely on the estimated/given nominal insert sizes of the paired-end libraries. With existing approaches, paired reads that span over a SV (i.e. the aligned locations of the two reads are not within the expected insert-size) can be misaligned to other genomic locations where they can be concordantly aligned instead. This bias will affect alignment and adversely impact variant-calling performance.

Although misalignments of reads are low in general, it is important to resolve them, as they are most likely to contain variants. As such, our variant of seed-and-extend gapped aligner, BatAlign [215], was developed to offer accurate alignments of reads spanning across SNVs, indels and SVs. Unlike existing seeding strategies, BatAlign allows high mismatches and a gap in the seeding regions of the read. It utilizes two strategies called *Reverse-alignment* and *Deep-scan* to find confident seed locations for reads. It also performs unbiased mapping of paired reads to avoid misaligning SV-spanning reads.

The organization of this article is as follows. We first describe the simulation of data and how the performances of aligners are being compared with one another. Next, we describe the routines being implemented in BatAlign.

In Discussion and Results, we first touch on the inadequacies on current seed-and-extend methodologies, then we compare the performance of BatAlign with some published methods over a wide range of datasets: ART-simulated datasets, indel-aberrant datasets, simulated paired-end datasets, RSV-simulated SV-datasets, and real datasets. The Overall, the results show that BatAlign had the highest F-measure for aligning reads, which contain variants or span across genomic breakpoints among the compared methods.

5.3 Results

Mapping biases, that occur in genomic regions with strong homology to other genomic locations [216], contribute to erroneous callings of SNVs, indels and structural variations. This problem should be given more attention as misalignments by a particular aligner tends to be recurrent among reads that share similar genomic contexts. As such, we are strongly motivated to study the alignment performance of officially published methods on reads with and without spanning variants. Based on our study, we developed BatAlign for accurate, sensitive and efficient alignment of NGS reads.

5.3.1 Simulation study showing that existing methods have difficulties mapping reads with high mismatches or located near structural variations

The alignment of reads in the presence of SNVs, and/or SVs still remain challenging despite developments already made by published aligners. This section intends to study the alignment accuracy of existing published methods using simulated reads that span across genomic regions with a high number of SNVs or near SVs.

Simulated reads with k mismatches can be mapped with less than k mismatches: Mismatches (like SNPs) can cause misalignments of reads to homologous genomic regions, especially when reads are sequenced from highly polymorphic regions. We simulated reads (see *Simulation of data*) to study the effects of mismatches in producing misalignments. For each read, we reported the lowest-mismatch unique hits (using BatMis [210], an exact k-mismatch alignment algorithm). We then compared the number of mismatches at which the reads were simulated with (we call this value *A*) and mapped at (we call this value *B*). Interestingly, if *A*=*B*, the respective alignments from BatMis were mapped correctly. However, when *A*≠*B*, the mappings were wrong, as it must be so due to being aligned to a location different from where it was simulated.

We should note that with the increase of simulated mismatches in a read, the occurrences of it being misaligned with a lesser number of mismatches also increase; statistically, this is true as mismatches act as wild cards in string-matching problems. From the mappings of BatMis, the rates of misalignment for reads simulated with 1 to 5 mismatches increased from 0.3% to 0.9% respectively. This result implies that it is unwise to always pick the lowest-mismatch hit as it might misrepresent the original location of a read.

To further investigate the impact of high-mismatch reads on the performance of the current published methods, we procured two groups of reads from the current set of simulated reads. The first and second group consisted of k-mismatch reads that can be mapped uniquely by k-mismatch and less than k-mismatch respectively. On the first group of reads, all the compared published methods have an average sensitivity of ~90% and the specificity approaches 100%. However, on the second group of reads, both sensitivity and specificity never exceeded 2% (see Figure 5.1). This highlights the difficulty faced by current methods on mapping high-mismatch reads and will later be resolved by using BatAlign's *Deep-Scan*.

Mate-pair information can falsely disambiguate alignments: Mate-pair information is useful in aligning two individually repetitive mate-paired reads unambiguously to the locality of each other in the reference genome. An ideal aligner should be able to align concordant and discordant paired-reads without bias, i.e., same rates of specificity while maintaining high sensitivity on mapping these two types of reads. (A concordant paired-read is a pair of reads that are sequenced from the vicinity of each other, within the expected wet-lab insert-size, with respect to the reference genome; otherwise, it is a discordant paired-read.) However, if mate-pair information is used too aggressively, an aligner might wrongly align a pair of discordant read-pair concordantly onto the reference genome.

We have studied the impact of mate-pair information on alignment performance by aligning two types of simulated paired-reads (see *Simulation of data*). The first set consists of paired-reads that were simulated with a mean insert-size of 500 bp (s.d. of 50

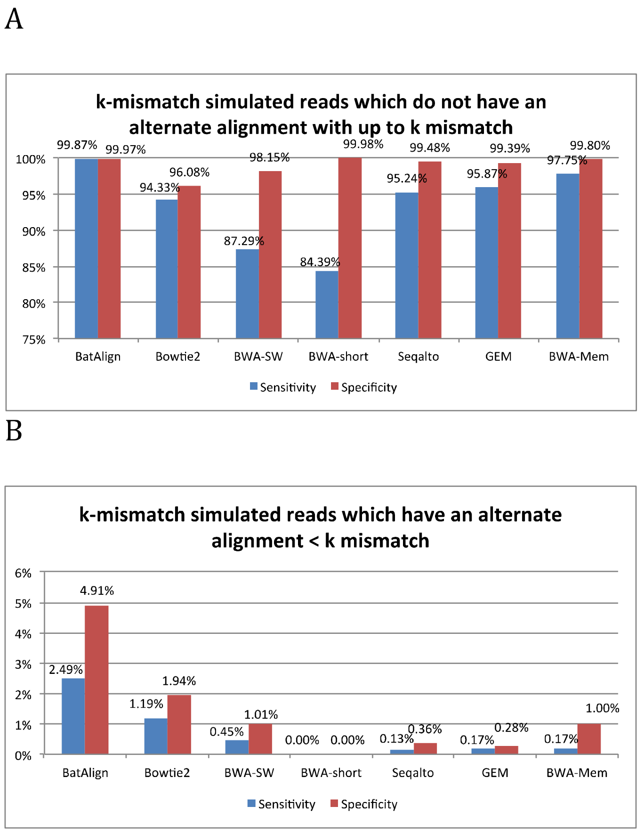
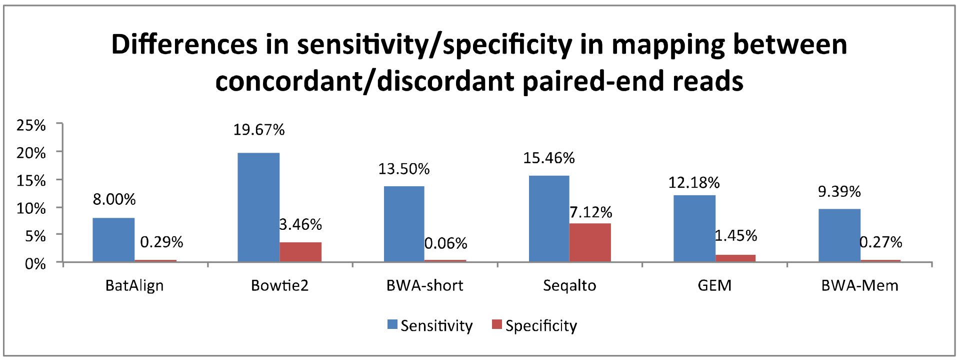


Figure 5.1. A) The sensitivity and specificity of compared methods on k-mismatch reads which can be mapped uniquely with k-mismatch. B) shows similar statistics to A) by mapping k-mismatch reads which have alternate unique alignment of ≤ k-mismatch.

bp) and the other set consists of paired-reads simulated with each end of the paired-reads from different chromosomes. Figure 5.2 reports on the differences in mapping sensitivity and specificity of each published method between these two sets of reads. An ideal aligner should exhibit minimal performance shift between these two types of reads. However, we observed that the alignment performance of the compared methods varied greatly from one another between these two types of paired-reads. The estimated bias, between mapping concordant and discordant read-pairs, in terms of sensitivity and specificity ranged from ~9.4% to ~20% and ~0.1% to ~7.1% respectively among the compared methods.

Figure 5.2. The differences in sensitivity and specificity between mapping paired-end datasets with simulated concordant and discordant paired-end information.



5.3.1.1 *Evaluation on ART-simulated reads*

To evaluate the performance of BatAlign on aligning reads, we compared it with 6 other officially published methods. We used ART [217] to simulate three datasets of 75bp, 100bp and 250bp read-lengths. Then, the reads in these datasets were aligned using the different methods. Figure 5.3 depicts the ROC plots on the ART-simulated datasets. Validation on the alignments showed that BatAlign has a better performance than the other compared methods in terms of both sensitivity and specificity over a large range of mapQ on the 75/100/250 bp datasets. We also cross-compared the methods as described in *Compared methods and method of cross-comparison* and presented their respective sensitivity and accuracy in Table 5.1. As shown in Table 5.1, BatAlign consistently outperformed the other compared methods in terms of sensitivity and specificity on simulated reads of various read-lengths.

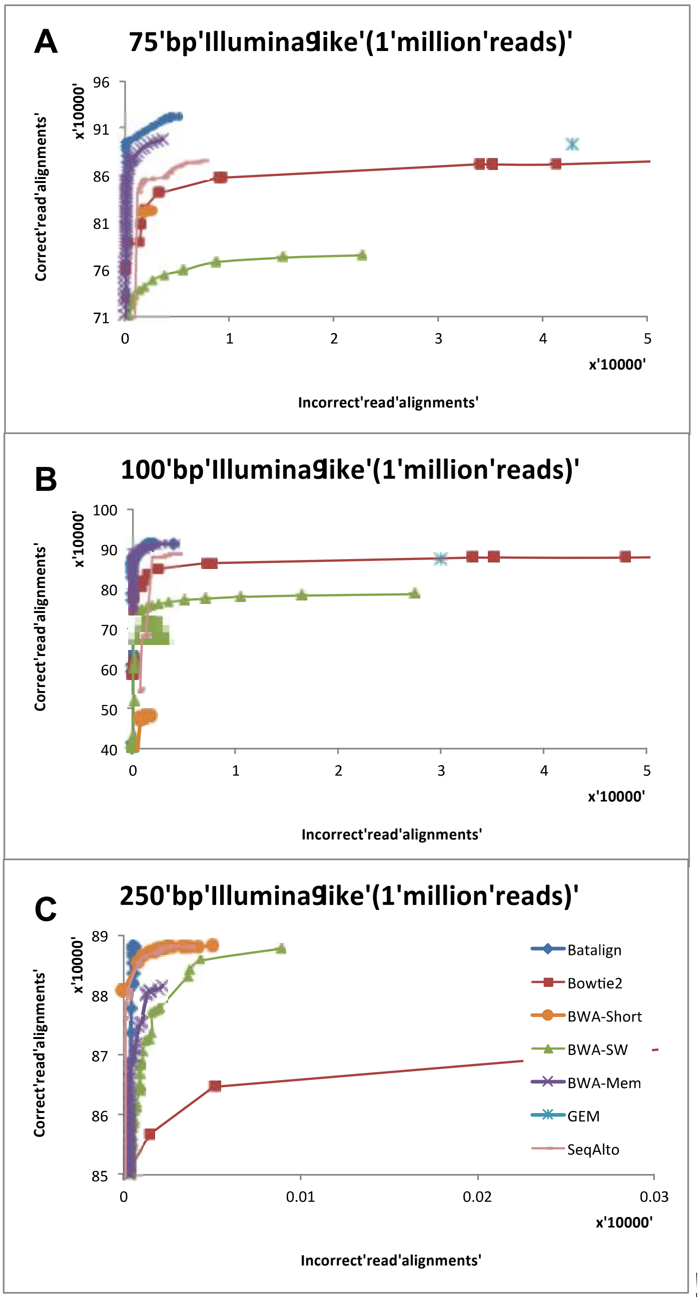


Figure 5.3. Sensitivity and accuracy for aligning simulated reads from ART. Cumulative counts of correct and wrong alignments from high to low mapping quality for simulated Illumina-like (A) 75 bp and (B) 100 bp (C) 250bp data- sets.

Table 5.1. Cross-comparison of sensitivity at similar specificity and vice versa for simulated datasets of 75/100/250 bp.

Simulated Data	Bat-Align		Bowtie2		BWA-SW		SeqAlto		BWA-short		BWA-Mem	
	SEN	ACC	SEN	ACC	SEN	ACC	SEN	ACC	SEN	ACC	SEN	ACC
75bp	91.0	99.998	84.1	99.987	74.9	*97.168*	85.5	99.862	*82.2*	99.944	89.5	99.998
100bp	*91.1*	100.0	83.7	99.992	75.8	*96.644*	87.7	99.786	47.8	-	90.2	99.999
250bp	*88.8*	99.999	85.1	99.891	86.1	*99.996*	88.2	99.999	88.5	100.0	87.1	99.998

SEN is sensitivity

ACC is specificity

Similar to the experiments performed in GEM's paper, we also validated the top 10 hits reported by each method. The complete breakdown of this validation by the first (or best) alignment, as ordered by their respective aligner, can be found in Table 5.2. From Table 5.2, we can see that BatAlign has reported the most number of correct hits as top-ranked hits in our simulated data on simulated data of various read-lengths.

5.3.1.2 *Evaluation on simulated Indel-aberrant reads*

The reads generated by ART have less than 0.01% probability of containing an indel. Therefore, ART-simulated datasets only show the performance of the methods on reads containing mismatches and SNVs. We used ART to spike in either inserts or deletions into each dataset at a rate of 0.1% to further gauge the performance of BatAlign on indel-aberrant data.

Since BatAlign allowed one gap in the seed region, BatAlign can seed locations for an indel-read with high accuracy and without bias for mismatch-stricken locations which will cause indel-reads to be misaligned. On this read class, BatAlign achieved the highest F-measure of 92.0% and 91.9% on the 'delete' and 'insert' respectively. BWA-MEM

Table 5.2. Number of first (or best) alignment reported by various methods on simulated 100bp dataset.

Rank					75bp dataset						
					#Correct hits						Sum of
Aligner	1	2	3	4	5	6	7	8	9	10	correct hits
BatAlign	933560	11604	732	622	745	335	146	116	90	68	948018
Bowtie2	866410	10873	4095	1541	551	314	192	142	112	77	884307
BWA-SW	786309	2	0	0	0	0	0	0	0	0	786311
BWA-Mem	897581	-	-	-	-	-	-	-	-	-	897581
GEM	893162	13603	5243	2231	1074	688	555	375	323	272	917526
BWA-Short	834519	10008	3535	1226	408	160	83	52	43	26	850060
Seqalto	885208	5692	1712	723	306	164	102	63	33	32	894035

Rank					100bp dataset						
					#Correct hits						Sum of
Aligner	1	2	3	4	5	6	7	8	9	10	correct hits
BatAlign	924272	7599	941	728	851	332	182	108	101	63	935177
Bowtie2	866310	8685	2833	948	254	110	69	42	23	5	879279
BWA-SW	794661	7	0	0	0	0	0	0	0	0	794668
BWA-Mem	912562	-	-	-	-	-	-	-	-	-	912562
GEM	875333	10327	3533	1445	638	377	283	193	163	178	892470
BWA-Short	484558	5207	1747	662	211	112	79	48	20	13	492657
Seqalto	890336	1821	515	194	91	44	38	11	3	8	893061

Rank					250bp dataset						
					#Correct hits						Sum of
Aligner	1	2	3	4	5	6	7	8	9	10	correct hits
BatAlign	894799	6394	732	824	225	175	107	98	53	50	903457
Bowtie2	892350	6245	1269	346	184	131	83	62	49	34	900753
BWA-SW	894395	1	0	0	0	0	0	0	0	0	894396
BWA-Mem	881193	-	-	-		-	-	-	-	-	881193
GEM	895450	5999	1203	319	201	116	92	79	50	46	903555
BWA-Short	894658	5615	800	67	0	0	0	0	0	0	901140
Seqalto	894661	3775	296	77	41	33	14	9	9	9	898924

also performed well on this read-class with F-measure of 90.8% due to the stitching of multiple maximal exact matching read-segments into the final alignment of a read.

The results from aligning on ART-simulated and Indel-aberrant datasets showed that BatAlign has better performance than the other methods on aligning a general ART-simulated dataset of reads containing a mixture of mismatches and indels. Thus, BatAlign can be used to identify a broad spectrum of short-range intra-chromosomal variants, in the presence of sequencing errors. We will discuss the performance of all compared methods in identifying long-range intra/inter-chromosomal variants in the next section.

5.3.1.3 Evaluation on concordant- and discordant-paired reads

Another issue of existing methods is that they may over-aggressively assume paired-reads to be concordant on the reference genome. In this section, we present the results on mapping concordant (emulating a normal genome) and discordant (emulating large deletions and structural variations in a diseased genome) simulated reads using the paired-end mapping mode available in the compared methods.

On the concordant paired-end dataset, BatAlign, Bowtie2, BWA-SW, GEM, BWA-Short, SeqAlto and BWA-Mem reported sensitivities with their corresponding specificities of 98.4% (99.891%), 91.1% (92.601%), 93.0% (98.711%), 97.4% (98.183%), 60.2% (99.702%), 96.2% (99.881%) and 98.2% (99.834%) respectively. When mapping concordant paired-reads, almost all the compared methods have similar accuracy. In contrast, with the discordant paired-end dataset, the sensitivity dropped for all the compared programs. Despite the drop in alignment performance, BatAlign still reported the highest sensitivity and specificity. On the discordant paired-end dataset, at mapQ > 0, the sensitivity of the compared methods with their corresponding specificity for BatAlign, Bowtie2, BWA-SW, GEM, BWA-Short, SeqAlto and BWA-Mem are 90.4% (99.571%), 71.5% (96.058%), -(-), 85.2% (96.735%), 46.7% (99.646%), 80.7% (92.759%) and 88.8% (99.565%) respectively. In general, BatAlign has the highest F-measure of 99.1% and 94.8% for the concordant and discordant paired-end datasets.

The mapping performance on these two datasets from the compared methods is shown in Figure 5.4. The alignment performances on each of the two datasets, from the same method, were joined together by a line. One can infer the robustness of paired-end mapping mode of a method from the interpolation of the line that joins the paired data points of the corresponding method in Figure 5.4. Thus, Figure 5.4 graphically shows how biased a method can be when aligning with mate-pair information. Overall, BatAlign was observed to have the smallest fluctuations in its F-measure, by only ~0.6%, between the two datasets.

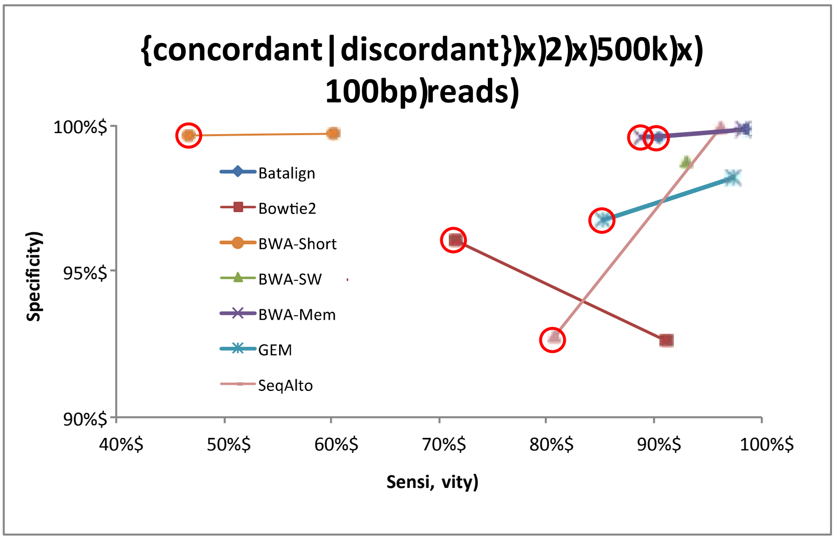


Figure 5.4. Sensitivity and specificity on mapping of concordant and discordant datasets using paired-end mapping mode of various methods. Data points circled in red depicts mapping performance on discordant dataset.

BWA-SW was unable to complete the alignment of 2 x 500k x 100bp discordantly paired reads and is plotted as a single data point.

From Figure 5.4, an interesting trend of results was observed for the compared programs excluding BatAlign. The initial observation was that methods, which had lower specificity on the concordant-paired dataset, would generally suffer a smaller drop in specificity on the discordant-paired dataset. For instance, Bowtie2 used to have a specificity of 92.601% on the concordant set but the specificity improved to 96.058% on the discordant set. The inverse of the initial observation on the results was also true. SeqAlto used to have the highest specificity of 99.881% on the concordant set but its specificity suffered the largest drop of 7.122% to 92.759% on the discordant paired-end dataset. These fluctuations in specificity are due to the aggressiveness of the pairing algorithms in the various methods to map paired-end reads close to each other on the reference genome.

The results in this subsection were obtained from running datasets of 100 bp long. Experiments were also done using 75 bp and 250 bp datasets and the trend of results were consistent among all three datasets.

5.3.1.4 Evaluation on reads from a RSVsim rearranged genome

Recent developments in SV-callers have revolved around the usage of soft-clipped reads [218, 219] and spanning read-pairs [220]. By extracting soft-clipped aligned reads and spanning read-pairs (read-pairs which were aligned outside of the expected range of insert-size or/and with abnormal orientations, an SV can be inferred from such signatures using an SV-calling algorithm.

Due to efficiency reasons and insufficient soft-clipped alignments (due to the local-realignment strategy and alignment scores used) from some of the compared aligners, we have decided to use BreakDancer to call our putative SVs back from the respective sets of alignments to gauge their performance on recovering the SVs. In addition, Table 5.3 shows the robustness of the alignments if the samples were to be down-sized to rates of 50% and 25% from the original simulated coverage depth of 30X. For the various down-sampled datasets, the SVs called out from BatAlign's alignments have higher F-measures consistently.

Table 5.3. F-measures of SV-callings against oracle information from Bioconductor's RSVsim package at various down-sampled rates of the dataset from an original depth of 30X.

Method	F-measures of SV-calling (at various down-sampling rates)		
	25%	50%	100%
BatAlign	**79.87%**	**83.73%**	**89.08%**
Bowtie2	75.99%	78.56%	76.21%
BWA	3.02%	7.87%	15.84%
BWA-SW	73.39%	78.88%	83.68%
GEM	75.17%	80.08%	79.09%
SeqAlto	76.11%	83.34%	88.63%
BWA-Mem	72.68%	70.87%	80.94%

5.3.2 Evaluation on real reads

We have downloaded 2 x 76 bp (SRA accession DRR000614, Sample: NA18943), 2 x 101 bp (SRA accession SRR315803, Sample: NGCII082 Mononuclear blood) and 2 x 150 bp (SRA accession ERR057562, Sample: ERS054071) paired-end datasets. The sequencing platform used for the downloaded datasets was Illumina Genome Analyzer IIx for the 76/101 bp dataset and Illumina MiSeq for the 150 bp dataset. We evaluated alignment performance on real data by performing single-end read mapping on paired-end datasets. Subsequently, we used the concordance and discordance mapping rates from the alignments to estimate the correct and wrong alignment rates (See *Comparison of alignment performance using ROC graphs* for more details on the definition of concordance and discordance). In order to minimize error in estimating alignment performance by using concordance information from the alignments, we have only used sequencing data from non-cancerous origins. Figure 5.5 depicts the ROC plots on the real datasets. Similar to our results on simulated data, BatAlign has reported more concordant and less discordant alignments on the tested real datasets over a large range of mapQ scores.

To verify if better mapping can improve variant calling, we apply different variant callers to the alignments of reads from some real dataset. First, we inspected the discordant read-pairs spanning validated SVs across various down-sampled rates of a real dataset (Accession: ERP001196, Patient Sample: 46T, Read Format: 2x90 bp, Nominal Insert-size: 170 bp). This patient sample was chosen as it contained a higher number of PCR-validated genomic rearrangements compared to the other samples in [221]. In Table 5.4A, we reported the number of SVs being supported by the alignments from the respective aligners. Across different down-sampled rates of the dataset, BatAlign was able to recall

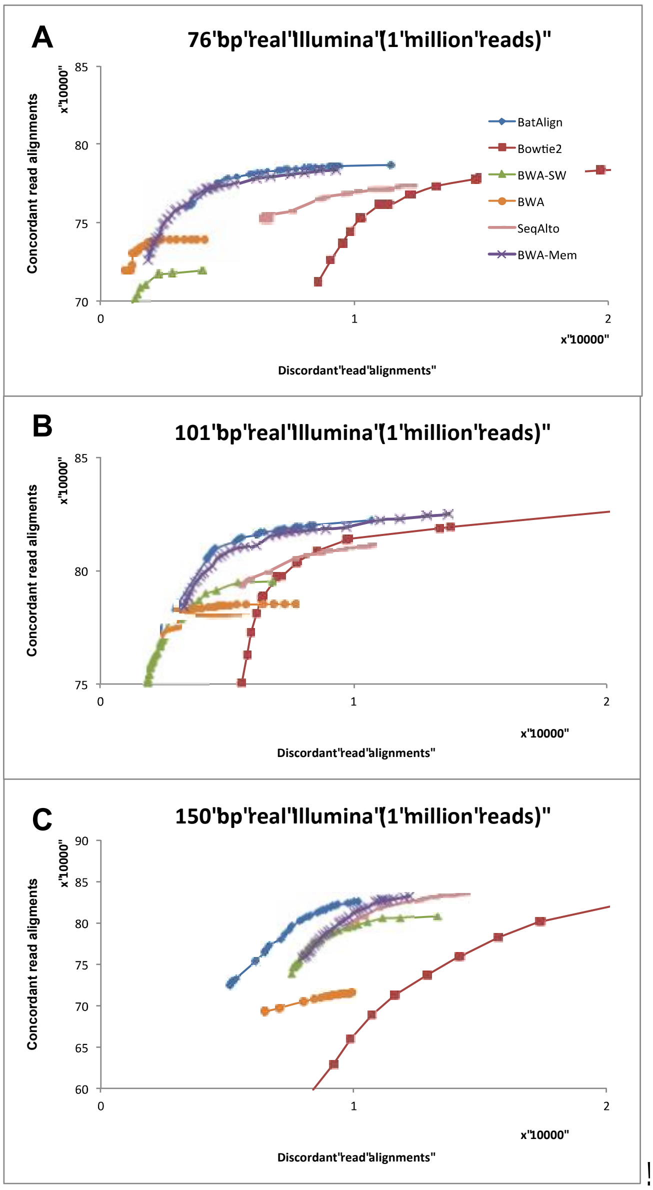


Figure 5.5. Concordance and discordance rates of alignments on real reads. Cumulative counts of concordant and discordant alignments from high to low mapping quality for real sequencing reads (A) 76 bp and (B) 101 bp (C) 150bp data-sets.

the most number of PCR-validated SVs across all of them. Table 5.4B reports on the number of candidate SVs being called out by BreakDancer, when compared with the number of validated SVs, we can estimate specificity of the alignments, which spanned outside of the expected sequenced insert-size. As compared to the second-best method, BatAlign produced 73.3% less callings but was still able to recall 4.3% more PCR-validated SVs. From this, we can infer that BatAlign is both sensitive and specific on aligning SV-spanning reads.

Table 5.4A. Comparison on the number of SVs recalled across various sub-sampled data of published and validated SVs of Patient 46T through manual counting of supporting real-pairs.

Methods	Intersect with published 46T data (at various ‘x’ down-sampling rates) - PE170-insert_size					
	1x	2x	3x	4x	5x	6x
BatAlign	**121**	**114**	**100**	**83**	**68**	**59**
Bowtie2	98	71	64	62	57	55
BWA	78	64	62	59	57	54
BWA-SW	116	104	81	71	65	58
GEM	80	66	64	63	60	55
SeqAlto	106	77	65	62	59	56
BWA-Mem	116	103	82	68	66	58

Table 5.4B. Total number of putative SVs called from across various sub-sampled data of Patient 46T.

Methods	Number of SVs called by Breakdancer across down-sampled rates					
	1x (base)	2x	3x	4x	5x	6x
BatAlign	39376	13105	7758	5180	3703	2800
Bowtie2	103721	29114	20133	15272	12332	10399
BWA	45184	9016	5129	3284	2330	1739
BWA-SW	52380	34729	22040	14931	10853	8259
GEM	16768	6449	3271	1969	1315	962
SeqAlto	56801	16760	10147	6831	4939	3829
BWA-MEM	68227	15062	9184	6135	4344	3237

The library used had ~20X in sequencing depth. A total of 126 validated SVs [221] were used for this comparison.

5.3.3 Evaluation on running times

Up till now, we reported on the F-measures (simulated data), concordance/discordance and variant-calling performance (real data) of the compared methods. BatAlign was generally observed to have the highest performance on these mentioned measures among the compared methods. BatAlign was developed to focus primarily on reporting accurate alignments and is also reasonably efficient. Table 5.5 shows that the relative runtimes and speed factors among the programs.

Table 5.5. Comparison of running times across all compared programs on 1 million reads from SRR315803.

Program	Runtime (seconds)	Speedup Factor
BatAlign	583	1.2
Bowtie2	459	1.5
BWA-Short	598	1.1
BWA-SW	639	1.1
GEM	**214**	**3.2**
SeqAlto	677	1.0
BWA-Mem	219	3.1

*Fastest speed up in bold.

5.4 Methods

5.4.1 Methods of experiments

5.4.1.1 Compared methods

We have used the following gapped alignment tools for comparison: BatAlign, Bowtie2 (2.0.6), BWA-Short, BWA-SW (0.6.1-r104), GEM (3rd release), SeqAlto (0.5-r123) and BWA-Mem (0.7.5a). These aligners are widely used and feature a wide range of mapping techniques. For each tool, the reference genome was indexed with default indexing parameters. hg19 reference genome was used for all experiments in this article. All experiments were run on a Linux workstation equipped with Intel X5680 (3.33 GHz) processor and 16GB RAM.

5.4.1.2 Simulation of data

We generated four classes of simulated data. The first class mimicked Illumina-like reads, the second class has one indel in each of its reads, the third class is 'paired' reads and the last class is from an RSV-rearranged [222] genome. The first class of reads was generated by ART (Huang et al. 2011) from hg19 (excluding non-chromosomal sequences). We have chosen ART for our study since the substitution errors were simulated according to empirical, position-dependent distribution of base quality scores; it also simulates insertion and deletion errors directly from empirical distributions obtained from the training data from the 1000 genomes project [223]. Empirical read quality score distributions were provided for read lengths 75 bp, 100 bp and 250 bp (these are the longest read lengths made available by ART). We have capped the number of mismatches and indels (SNVs or base-call errors or gaps) in the simulated reads at 7%.

The second class of reads was used to demonstrate the performance of BatAlign on aligning reads with indels. The average density of an indel is one in ~7.2kbp [224] so we simulated indels with ART at a much higher rate of 0.1% into 2 datasets (one each for insert- and delete-type of gap).

The third class of reads was used to demonstrate the efficacy of mate-pair information on the paired-end mapping mode of the compared programs. 6 sets of 1 million reads were created. Each set consisted of 2 x 500k x (75/100/250) bp x (concordant/discordant) reads. The first set consisted of concordant paired-end reads with a mean insert size of 500 bp and a standard deviation of 50 bp. The second set consisted of discordant paired-end reads, where the 'left' and 'right' ends of the paired reads were simulated from chromosome 1 and chromosome 2 of hg19 respectively. This class of reads was used to demonstrate the robustness of BatAlign when aligning reads with mate-pair information in the presence of genomic structural variations.

The fourth class of reads was used to gauge the performance of aligners on SV-spanning reads. A total of 3,760 SVs of insertions, deletions, duplications, inversions and translocations were simulated using the RSVsim package in the Bioconductor [225]. Reads were simulated from the rearranged genome to a depth of 30X and aligned to the hg19 reference genome. The resulting alignments were subsequently applied with BreakDancer [220] to call out putative SVs and were validated against the oracle information from the simulator.

As simulated data comes with oracle information, we have used the F-measure to gauge the performance of the methods: We define sensitivity (SEN) =TP/(TP+FN), accuracy (ACC) =TP/(TP+FP) where TP, FP and FN are true-positives, false-positives and false-negatives, respectively; F-measure = 2(SEN*ACC)/(SEN+ACC). As we do not have true-negatives in our simulated experiments, accuracy will be used interchangeably with specificity.

5.4.1.3 Comparison of alignment performance by stratifying against all reported mapQ scores

As the original locations of simulated reads were known, we have assessed the sensitivity and accuracy of each method using simulated reads in this section. For each method and each dataset, we discarded mappings with mapQ = 0 for all methods as they were deemed ambiguous. Then, we recorded the cumulative number of correct and wrong alignments by their respective decreasing mapQ and plotted these results in a form similar to a ROC curve; the corresponding cumulative number of correct and wrong alignments at a particular mapQ cut-off will be the respective x-axis and y-axis values for a single data point on the ROC curve. In addition, due to the inability to align some indels to their exact locations and the presence of soft-clippings, an alignment will be considered as a correct mapping if the simulator on the same strand within 50 bp of the position simulated the leftmost position.

For real datasets, to address the lack of oracle information, we have mapped the paired-end reads as single-end reads and calculated the fraction of reads that were mapped concordantly. We consider a pair of reads to be concordant if they have the correct orientation and maps within 1,000 bp of each other with a mapQ > 10. (The distance 1000 is chosen since Illumina GA II machines normally cannot sequence paired-end reads from DNA fragments of size longer than 1000bp.) If both ends of the paired-end reads are mapped but are not located within a distance of 1000bp to each other, they will be marked as discordant mappings. To plot the full spectrum of concordance/discordance in our experiments on real dataset for the ROCs, we recorded the number of concordant and discordant alignments stratified by the mapQ score of the 'head' read. We must also emphasize that although the rate of concordant mappings was taken as a performance measure for aligning real reads, it can only give a lower bound of performance when used on mapping datasets of expectedly high paired-end concordance rates. Mapped reads with unmapped mate/pair-read will not be considered as they only form a minimal portion of the mappings and there is no oracle data to readily verify the correctness of their alignments.

5.4.1.4 Method of cross-comparison

It was noted that GEM is the only method among the 6 compared methods that does not calculate a mapQ for its alignments. We run the default modes of the compared programs unless otherwise stated. We have also adopted the performance measure "first correct" (or best) alignments from GEM's paper into our experiments to make sure our comparisons were extensive and correct.

In this paper, we have compared the full spectrum of mappings by stratifying alignments by their reported mapping quality scores. However, it is hard to compare the absolute differences in performance between methods as the calculation of mapQ of an alignment differs from one method to another. To resolve this problem and to present the relative differences in the performances numerically between the different methods, we will have to pick a baseline performance indicator. For instance, we can compare the different rates of sensitivity of the methods at similar rates of specificity while using the program with the best specificity as a baseline performance indicator for sensitivity. In general, more false-positive mappings will come with increasing sensitivity. Hence, we picked and compared the sensitivity and specificity of the various methods as described to remove bias due to the calculation of mapQ.

5.4.2 Our proposed solution: BatAlign = (Reverse-alignment + Deep-scan) + Unbiased mapping of paired reads

To align a read, existing approaches first find putative hits of short seeds from the query read. These putative partial hits are usually exact or 1-mismatch occurrences with respect to the reference genome. When there are high mismatches and/or indels in the read, it is likely that the seeded locations do not represent the original location of the queried read. To address the problem of missing hits from using low edit-distance short seeds, BatAlign uses high edit-distance in a long-seed (5-mismatches, 1-gap, and 75 bp) instead to search for a global base-call-quality-aware least-cost hit in the reference genome. To find the least-cost hits, BatAlign uses *Reverse-alignment* to enumerate putative candidate hits in increasing order of alignment cost (i.e. increasing number of mismatches and gaps). Since the hit having a minimum number of mismatches may not be correct (as shown in the Discussion section), *Deep-scan* was developed to selectively scan deeper into the search space of putative hits even after the least-cost hit has been found. The alignments of all candidate hits reported by *Reverse-alignment* and *Deep-scan* will be extended to their original full read-length. Then, base-call-quality-aware scores for these hits are computed. For the single-end mode, BatAlign will report the hits in the order of this quality-aware score to the users. For the paired-end mode, BatAlign will align both reads in the paired-read independently as if they were from a single-end sequencing experiment. Next, BatAlign will report the alignments for the paired-reads, which best represent the estimated insert-size of the prepared library.

5.4.3 Details of algorithms in BatAlign

5.4.3.1 Problem definition and overview of the method

The problem of mapping genomic reads is defined ideally as follows: Given a set of genomic reads, find the origin of each read in the reference genome, along with their correct alignments. However, in practice, this problem cannot always be solved and we have to resort to finding the most likely point of origin and alignment for each read.

The outline of BatAlign algorithm is as follows. As a pre-processing step, a one-time indexing of the reference genome is done. Next, it will start scanning for the most probable hits of the read in the reference by using *Reverse-alignment*. *Deep-scan* is then applied to scan and pick the most probable hit of the read from the reference genome. BatAlign then calculates a mapQ score for this hit and reports it. Below, we will discuss the novel components that aid BatAlign to gain accuracy and efficiency.

5.4.3.2 Reverse-alignment

Seed-based aligners search for candidate hits of its seeds; then, these hits are extended and the best alignment is selected based on a set of pre-defined criterion. In contrast, *Reverse-alignment* does the opposite by searching for the best possible hits in the reference first.

With a set of match/mismatch/gap scores assigned, we pre-compute the combination and the order of matches/mismatches/gaps that each 'step' of the scanning routine will need to scan the reference genome with. Reverse-alignment scans the read in increasing 'steps' of alignment-cost. In this step, we pick non-overlapping 75 bp segments from the 5' end of a read as seeds. For each hit of the seed, a maximum of 5 mismatches and 1 gap are allowed in a single seeded region.

5.4.3.3 Deep-scan

The best-scoring alignment need not be the correct alignment, even if it turns out to be the only hit with such a mismatches/gap combination. It is best if we can get the set of next-best alignments too. With these additional hits and using the quality information of the sequencing base-calls, we can better differentiate the correct hit from a pool of putative candidate hits. Furthermore, these extra hits will help BatAlign to assess the quality of the final alignment better as the mapping quality of the final alignment is computed from the two-best hits. If the first hits found during *Reverse-alignment* are multiple hits, then we return all of these hits. Otherwise, if it is a unique hit pertaining to such a mismatch/gap combination, then *Deep-scan* will be activated to scan for the next-best alignments.

5.4.3.4 Handling long reads

For reads longer than or equal to 150 bp, we will split the read into non-overlapping 75 bp reads. Each of the 75 bp segment will be aligned as described above. For instance, for 250 bp reads, BatAlign will obtain 3 consecutive segments of a read starting from the first base of the read and map each of them individually. If the first or best hit from each segment are non-repetitive and fall within the locality of each other, we will try to align the original read onto this region of the reference. By doing this, we avoid realigning the original read to more than one location of the reference. However, if the first or best hits from each segment are repetitive or not mapped to the locality to one another, BatAlign will examine and align the whole read onto each of the putative locations reported by each of the segment. Among these alignments, the best-scoring hit is reported.

5.4.3.5 Mapping with mate-pair information

BatAlign will first align all paired-reads in unpaired-fashion. If the top hits from each of the paired reads are confidently mapped and are within expected distance to each other, BatAlign will report this pair of alignments. However, if the reads cannot be paired up within the expected distance or one of the pair-reads is unmapped, SW-algorithm will be applied to the neighboring region of the anchored alignments to rescue the mate of the anchored reads. From here, we can calculate the mapQ for all the hits of the paired-reads. Instead of using just a cut-off for the alignment score of the rescued read, we also try to discriminate the goodness of the rescued seeds using mapQ, alignment score and mate-pair information simultaneously. Thus, for unbiased detection of structural variations caused by discordant paired-reads, the calculated alignment scores will precede mate-pair information.

5.4.3.6 Supplementary alignment of SV breakpoint-spanning read

The part of the read, which spans across a genomic rearrangement breakpoint will have many mismatches with respect to the reference genome, possibly incurring a negative alignment score, and will be soft-clipped away. For example, a CIGAR alignment string of "65M35S" is possible for a length-100bp read. In this example, we might be clipping away useful information, which can be crucial to identifying the partnering breakpoint of a SV.

Hence, BatAlign will realign the clipped portion from the primary alignment of a read whenever the clipped length of the alignment exceeds 20 bp. A fast 0-mismatch scan is applied to the last 20 bp of the clipped bases to find the candidate locations near potential SVs. Next, the same read will be realigned locally to the candidate locations to recover the auxiliary alignments. The chosen auxiliary alignment should complement the primary alignment of a read and together with the primary alignment, be able to represent the full length of the original read. In other words, the primary and auxiliary alignment can be used interchangeably for the same read.

5.4.3.7 Hardware-accelerated SW alignment

After the seed alignments are found for a read, BatAlign can perform either SW alignment or semi-global alignment to extend the alignment of the read. We have devised a semi-global alignment method that is faster than SW-alignment by ~30%, and the default mode of BatAlign is to extend the seeds using this semi-global alignment method. When the alignment score of the semi-global alignment drops below 90% of the maximum alignment score (i.e. the score for an exact match), a SW-alignment is done. If the user wants to perform SW-extensions only, an option is provided to do so.

The SW alignment is SIMD accelerated via SSE2 instructions. Our implementation was based on an extension of SSW library [111] that modifies Farrar's method [110]. This algorithm determines the best alignment in two steps: First it will calculate the best SW-score and then it will perform a banded SW alignment to get the optimal trace-back of the alignment from the DP-table.

5.4.3.8 Accelerating alignment

The speed of the algorithm is improved by limiting the number of SW-alignments performed for each read. Another way is to stop performing SW-extensions when the best alignment score has failed to increase after a determined number of attempts. To trace back the optimal alignment path in the DP table, we need to perform a non-SIMD banded version of SW-algorithm. This step is time consuming. However, we can skip this step if the SW-score of the alignment falls below the current best SW-score.

5.4.3.9 Alignment score and mapping quality

Sequencing data can contain a per-base quality score that indicates the reliability of a base call. If the probability of a base call at position i being correct is P[i], the quality score Q[i] assigned to location i is given by the equation $P[i] = 1 - 10^{-Q[i]/10}$. Assuming that there is no bias to a particular set of nucleotides, the probability of a base being miscalled at location i can be calculated by the formula $1 - P[i]/3$. For a given alignment, we compute an alignment score based on an affine-gap scoring scheme, where the score for a match or a mismatch at R[i] is the Phred-scaled value of P[i].

5.5 Conclusion

We presented BatAlign for the gapped alignment of short reads onto a reference genome with improved accuracy and sensitivity. The mapping strategies discussed in the *Method* section, such as *Reverse-alignment*, *Deep-scan* and *Mapping with mate-pair information*, produced mappings with increased accuracy when compared with other methods in simulated data (ART-simulated, Indel-aberrant, paired-end, variant-spanning). In addition, BatAlign also aligned over sites of PCR-validated SVs and SNVs on real data more robustly over various down-sampling rates of the input data. A new *faster semi-global alignment algorithm* and other heuristics have also been used to replace the traditional SW routine to speed up BatAlign. In general, BatAlign is an improved aligner for accurate gapped alignment of DNA sequencing reads.

Recently, a number of aligners such as YAHA [113] and CUSHAW2 [100] were developed to handle long reads (500 bp or more). A possible future work is to develop an accurate tool for the alignment of long reads.

Chapter 6

Spliced Alignment Problem

6.1 Introduction

RNA, together with DNA and proteins, is one of the three major macromolecules that are needed for life. Pre-mRNA is synthesized from the DNA through transcription and is matured by having its introns removed [151]. In mammalian genomes, alternate splicing of the same gene region adds onto the genomic complexity by generating multiples variants of a single gene known as mRNA isoforms [152]. The disruption in the synthesis of mRNA isoforms can cause genetic disease [153, 154]. Hence, it is critically important to accurately identify and quantify the splicing sites in both normal and diseased cell states.

RNA-seq can interrogate gene expression levels at genome-wide scale. *De novo* detection of splice junctions and quantification of novel gene expression was also not possible with microarray technologies before. Each sequencing run, from next-generation-sequencing (NGS) technologies, of an RNA-seq experiment can yield up to hundreds of millions of bases, allowing the accurate relative quantification of expressed transcripts. In all, RNA-seq has provided a quantum leap to the analysis of novel features in the transcriptome [33] from hybridization-based microarray techniques.

6.2 Challenges in Spliced Alignment

The first step to analyzing RNA-seq data is to align the sequenced reads back onto a known reference genome or annotated transcriptome. The alignment of RNA-seq also brought along an additional set of challenges as compared to aligning DNA-seq data. The first challenge is to align in the presence of large gaps due to the presence absence of introns from the sequenced reads with respect to the reference genomic text, which we are aligning the reads onto. From empirical studies, ~38% of 100 bp RNA-seq reads can span across two or more exons that can be thousands of bases apart [166]. Due to splicing junctions between adjacent exons in a read, different subparts of a read can map to different adjacent exonic regions of the reference genome but with a large intronic gap in between them. Other than the presence of large intronic gaps, alignment is further complicated by the presence of polymorphisms, indels and sequencing errors. In addition, it was also observed that ~25.8% of 100 bp long reads, has an exon-exon boundary within 10 bp on either ends of a read. This short residual exon, which we call short 'overhang', can be represented spuriously by the reference genome and is both computationally and algorithmically hard for aligner to accurately locate its correct alignment efficiently. Short exons can also appear in the middle of a read, sandwiched between two exon-exon boundaries within a single read. Without loss of generality, RNA-seq reads poses a new set of challenges for aligners to work with as compared to its siblings of DNA-seq aligners.

Other than intronic gaps, pseudogenes also make splice alignment harder than DNA gapped alignment. Pseudogenes are dysfunctional relatives of genes which are highly similar to RNA sequences [226]. An ideal RNA-seq aligner should be able to avoid aligning reads to processed pseudogenes at all times as pseudogenic regions do not transcribe to mRNA sequences. The authors of TopHat2 [41] has also found that ~26.9% of reads in the RNA-seq data from [227] can be aligned to the full length of pseudogenic regions with at least 80% identity. This poses a challenge to us as reads can sometimes be aligned to pseudogenic locations with higher percentages of identity than to their original location of transcription. For instance, a read can align in an ungapped fashion onto a pseudogenic region as an exact match but the correct alignment should be exact matches of two non-overlapping and adjacent substrings of the read marked by an exon-exon boundary (intronic gap on the reference genome) between them.

6.3 Related Work

Several alignment algorithms have been developed to align mRNA-seq reads [34-38, 40, 167, 170]. Here, we review some of the published RNA-seq aligners which we compared our methodology with in this section of the thesis. MapSplice uses consecutive contiguous 20-25 bp long seeds of a read to determine the candidate alignments of the mRNA read. Based on the seed locations, MapSplice will determine the most likely alignment of each mRNA read to a reference genome. Similar to SpliceMap, OLego also uses sub-sequences of a read to obtain anchor locations of an mRNA read. However, OLego uses relatively shorter seeds of 12-14 bp long and is aimed at sensitive recovery of micro-exon (~20 bp). STAR uses the idea of finding a Maximal Mappable Prefix (MMP) as its seed finding routine. This concept is similar to the maximal exact unique match used by genome alignment tools Mummer [125] and MAUVE [228]. Due to this, STAR is very efficient. However, the use of small seeds and MMP as the seed finding routine may produce spurious candidate locations, which are hard to disambiguate, and correct candidate locations may be missed respectively. This affects the accuracy of existing methods.

Our work described in this chapter, BatRNA, is based on a fast BWT data-structure for efficient detection of splice junctions and focuses on distinguishing spliced reads from exonic reads by using phased aligned strategies to handle each type of reads automatically with high sensitivity and accuracy. BatRNA is also the fastest method among the compared programs, which use similar amount of physical working memory.

6.4 Results

The simulated data is produced by BEERS using hg19 configuration files which can be downloaded from the RUM website. The performance of aligning real data was evaluated with reads from ERP00196 (Sample: 11T). We will report the performance of BatRNA based on evaluation of aligning simulated and real data below.

6.4.1 Setup of experiments and performance measures used

In order to evaluate the performance of our method, we have benchmarked against the following programs: OLego v1.1.1, MapSplice v2.1.2, STAR 2.3.0e and TopHat2 v2.0.8b. Some RNA-seq reads can map to multiple genomic locations and since a read can only come from at most one point of origin we only validate unambiguous mappings which were indicated by a non-zero mapping quality.

Whenever ground truth is available, we will use F-measure to compare the mapping performance of all the compared methods. F-measure is defined as $(2 * (R * P)/(R + P))$, where *R* is Recall and *P* is Precision. As for real datasets, we will use the cumulative number of spliced mappings over edit-distances of ≤ 3 to compare the performance of the methods.

To address the lack of ground truth for the alignments of real data, we have adopted a variant of validation used in TopHat2 paper [41] which we will further elaborate in the later section to evaluate the alignment performance on real data.

All our experiments were run on a server equipped with Intel Xeon X5680 @ 3.33GHz and 48 GB of RAM. We allowed the same CPU threads on all the compared programs in our experiments. Other parameters were kept at default.

6.4.2 Evaluation on the simulated RNA-seq Illumina-like reads

We have used BEERS in the RUM package to simulate two datasets of 75 bp and 100 bp read-lengths. We aligned the reads in these datasets and summarized the number of correct and wrong alignments in Figure 6.1.

In terms of recall, BatRNA is second to MapSplice with ~1% lower recall on the simulated datasets. In terms of precision, BatRNA and OLego tied as the methods with the best precision. However, BatRNA was able to obtain the highest F-measures on aligning these two simulated datasets. This can be seen from Figure 6.1, that MapSplice being the best recall method, was ranked 4th out of our 5-methods comparison on precision. Despite having good prevision, OLego was ranked last for its low recall rates.

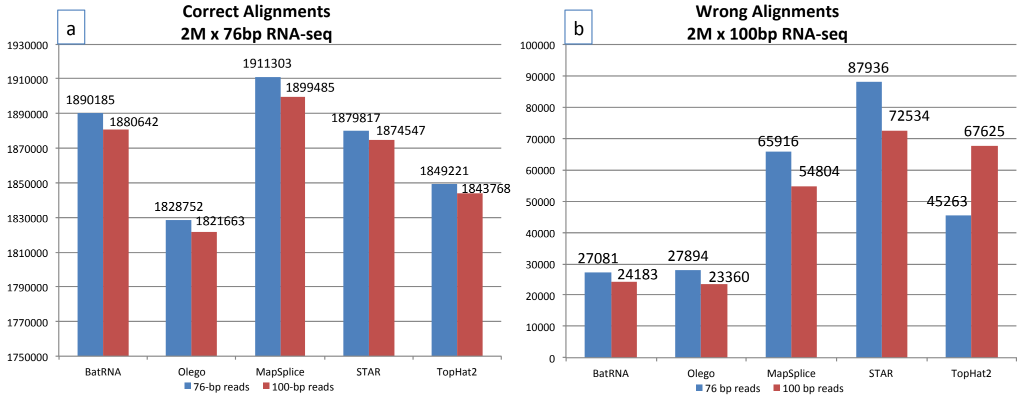


Figure 6.1. The counts of (a) correct alignments and (b) wrong alignments from the compared methods on 76 bp and 100 bp BEERS-simulated datasets.

The precision of alignments generally increases with increasing read-lengths as reads of longer read-lengths can be represented more uniquely on the reference that it is simulated from than its shorter counterparts [104]. From Figure 6.1, we can observe such a general trend of increased specificity across all the compared methods except for TopHat2. Table 6.1 reports the F-measure of the various methods using oracle information available from the two simulated datasets.

Table 6.1. The F1-scores of the compared methods on BEERS-simulated 2M datasets.

Method	F-measure	
	76 bp	100 bp
BatRNA	**96.51%**	**96.32%**
OLego	94.84%	94.75%
MapSplice	96.11%	96.07%
STAR	94.75%	94.98%
TopHat2	94.97%	94.28%

RNA-seq datasets generally contain unequal proportions of exonic and spliced sequenced reads. When the read-length increases, the chances of a read spanning across an exon-exon boundary increase. At current popular RNA-seq read-lengths of ~100 bp, the proportion of exonic reads is expected to be ~82.2% from empirical studies of simulated reads using BEERS. As exonic reads are the dominant portion of alignments sequenced from a typical un-diseased human sample using reads of length ~100 bp, the accurate quantification of transcript abundance can be achieved solely with an unspliced aligner albeit the inability to identify exon-exon junctions in the sampled data. In order to gauge the true alignment performance of the compared splice alignment methods, we will procure the previously discussed simulated datasets, segregate the exonic and spliced reads from each other, align them and present their alignment results in Table 6.2 separately. Without loss of generality, this enabled us to gauge the alignment performance of the compared methods with better granularity on RNA-seq reads.

Table 6.2. Breakdown of alignment performance by exonic and spliced reads using simulation.

	Methods	Spliced read count	Exonic read count	Sensitivity (Spliced)	Precision (Spliced)	Sensitivity (Exonic)	Precision (Exonic)
2M x 76 bp	BatRNA	355811	1644189	**85.84%**	**92.93%**	96.39%	**99.76%**
	OLego			71.05%	90.93%	95.85%	**99.83%**
	MapSplice			**86.31%**	90.93%	**97.57%**	97.85%
	STAR			77.91%	82.46%	**97.47%**	98.22%
	TopHat2			75.51%	**93.28%**	96.13%	98.39%
2M x 100 bp	BatRNA	447170	1552830	**83.75%**	**94.80%**	96.99%	**99.76%**
	OLego			72.64%	93.95%	96.39%	**99.84%**
	MapSplice			**84.58%**	93.14%	**97.97%**	98.26%
	STAR			78.70%	87.52%	**98.05%**	98.55%
	TopHat2			76.97%	**94.54%**	96.57%	96.91%

The best-2 recalls and precisions scores of each experiment for this table are in bold.

Apart from reporting on non-ambiguous (mapQ > 0) hits, we also report on the sensitivity of the top-10 hits reported by each method to determine if multi-mappings from the other programs can correctly quantify transcript abundance. In addition to showing the rank of the reported correct hit among the top-10 multi-hits that a method has reported for a read, the tabulated statistics in Table 6.3a also indirectly showed the cumulative number of wrong hits that an aligner has reported by allowing the report of multi-hits. For instance, if aligner A was to report k number of rank-2 hits, it would also mean that aligner A has also reported k number of top-rank (rank-1) for k reads. It should also be noted that the correct hit for a read should preferentially be reported as a rank-1 hit. From Table 6.3a, BatRNA reported the least number of non rank-1 hits indicating its ability to discriminating against spurious hits effectively. Correspondingly, Table 6.3b tabulates the number of wrong multi-hits that were reported alongside with a rank-k correct hit.

Table 6.3a. Tabulation of correct hits ranked by the order in which they were reported for a read.

		75bp dataset									
		#Correct hits									
Methods	Rank of hits	1	2	3	4	5	6	7	8	9	10
BatRNA	Spliced	335922	1272	312	96	56	49	23	14	5	2
OLego		280648	2381	651	197	111	66	26	23	18	6
MapSplice		336955	2232	462	122	48	29	5	2	1	0
STAR		335195	4085	826	241	170	104	38	21	7	5
TopHat2		283938	2278	627	133	77	27	6	21	20	11
BatRNA	Exonic	1585428	480	11	1	8	1	2	0	2	0
OLego		1595295	15038	2456	378	7	0	0	0	1	0
MapSplice		1604210	18779	5071	1986	1058	746	377	237	184	161
STAR		1603880	19726	5123	2059	1001	569	255	134	52	30
TopHat2		1582341	18820	5102	1971	1044	784	397	295	201	181

Table 6.3b. Tabulation of wrong hits being reported alongside a rank-k correct hit.

		75bp dataset									
		Cumulative #Wrong hits for reported correct rank-k hit									
Methods	Rank of hits (k)	1	2	3	4	5	6	7	8	9	10
BatRNA	Spliced	0	1272	624	288	224	245	138	98	40	18
OLego		0	2381	1302	591	444	330	156	161	144	54
MapSplice		0	2232	924	366	192	145	30	14	8	0
STAR		0	4085	1652	723	680	520	228	147	56	45
TopHat2		0	2278	1254	399	308	135	36	147	160	99
BatRNA	Exonic	0	2163	940	306	208	115	54	63	88	0
OLego		0	480	22	3	32	5	12	0	16	0
MapSplice		0	15038	4912	1134	28	0	0	0	8	0
STAR		0	18779	10142	5958	4232	3730	2262	1659	1472	1449
TopHat2		0	19726	10246	6177	4004	2845	1530	938	416	270

k is an integer from 1 to 10 inclusive. k is used to denote the order of a hit in which it is reported. A correct hit of rank-k will also mean that it has generated (k-1) * #correct_rank-k_hits.

6.4.3 Evaluation on real RNA-seq Illumina-like reads

Although the lack of ground truth makes our validations much more difficult on real data, we would still like to use real data to provide a measure of corresponding performance in practice as if we were dealing with simulated data.

6.4.3.1 Edit-distance as a measure of correctness in real-data set

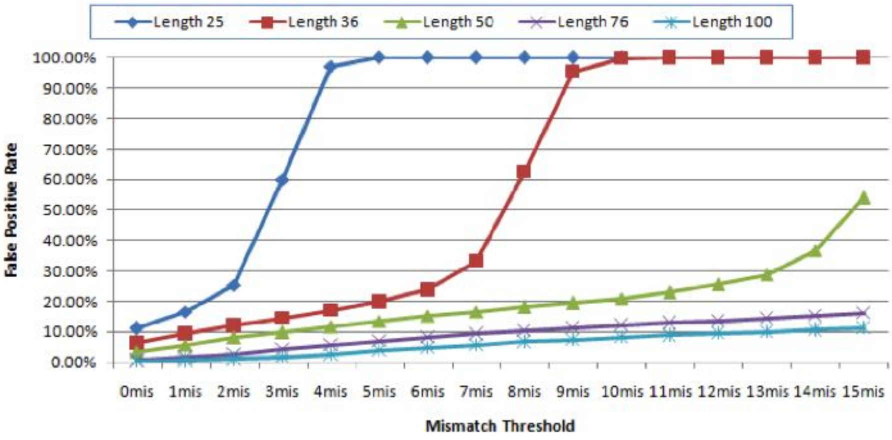


Figure 6.2. Chromosome-1 reads were mapped to a chromosome-1-deficit hg19. False positive rate was calculated by the number of simulated reads that were mapped to the modified hg19, divided by the total number of reads.

Due to the lack of ground truth, we adopted one experiment from TopHat2 paper whereby the authors estimated the performance of real-read alignment with cumulative number of alignments with edit distances of ≤ 3; assuming these candidate hits with low edit distance are correct. But first, we would have to study the behavior of this validation using simulated data. Using simulated reads from chromosome 1 of hg19, we mapped these reads back to a chromosome 1-deficit reference genome to investigate how increasing edit-distances actually correlate with the noise rates in our alignments. Figure 6.2 shows the false mapping rates of these simulated chromosome-1 reads, of various read-lengths, increased when the allowed maximum edit distance to their respective alignments also increased. As the false positive mapping rates for the reads become significant when edit distance of more than 3 was allowed for the alignments, we will only assume alignments with edit distance lower than or equals to 3 as correct in our experiments on real data.

First, we will apply this validation to the simulated dataset to observe the relationships of sensitivity and specificity with increasing edit-distances of alignments. Figure 6.3 is generated from the results of the immediate preceding section on simulated data. It gave us a feeling of how the trend of results for a method would behave with more/less and correct/wrong alignments. As we can see from Figure 6.3, the gain in recall rates is marginal as the edit distance of the alignments approaches 3. To add on, the relatively large drop in specificity with respective to the number of correct alignments with higher edit-distances discouraged us from using alignments with high edit-distance for downstream analysis too.

Using edit distances to test the goodness of spliced alignments is far from satisfactory as the results from Figure 6.3 and 6.4 do not agree with each other. From Figure 6.3, a method with many wrong and low edit distance alignments will have its wrong alignment eluded for scrutiny if they are represented as what has already been shown in Figure 6.4. Upon deeper investigation, we found out that the pseudogenic regions caused the disparity between the results. The reads were mapped to pseudogenic regions either with a lower edit distance or it is mapped preferentially without splicing junctions in them. For instance, a 0-edit distance spliced read can be mapped to a pseudogenic region in an unspliced fashion, both locations will yield 0-edit distance alignments, and still be considered as a good alignment under this form of validation. From this observation, the validation of alignment on real RNA-seq reads using edit distances will be more appropriate if it was applied solely to spliced alignments. This will directly prevent the wrong classification of spurious unspliced alignments as correct hits.

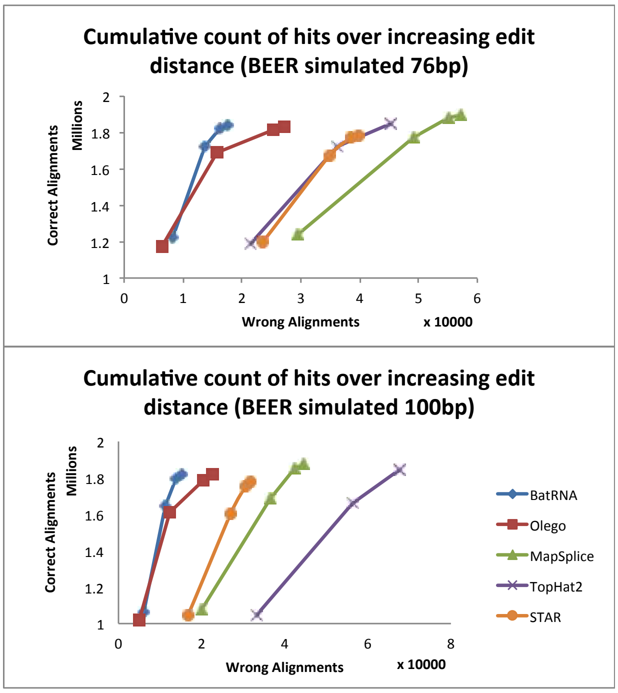


Figure 6.3. The counts of correct and wrong alignments for simulated RNA-seq 76bp and 100bp of 2 million reads each stratified by edit-distances of 0 to 3.

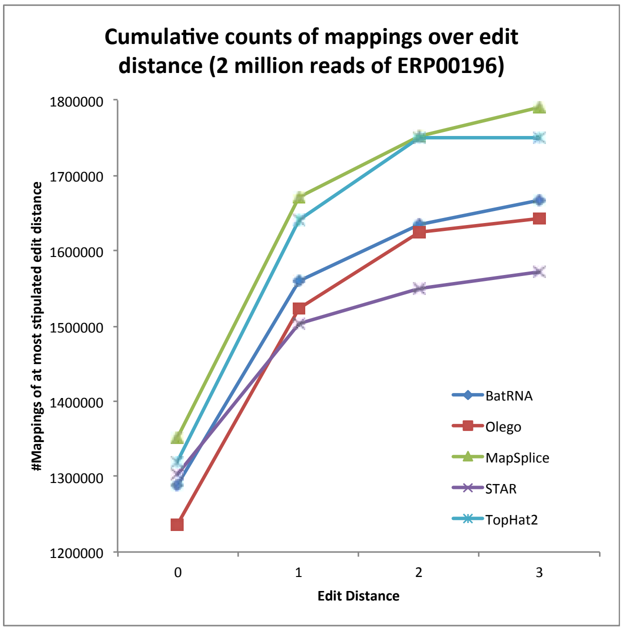


Figure 6.4. The cumulative counts, over edit distances of 0-3, of all non-ambiguous mappings from the various spliced mappers on 2 million real reads taken from Sample 11T of ERP00196.

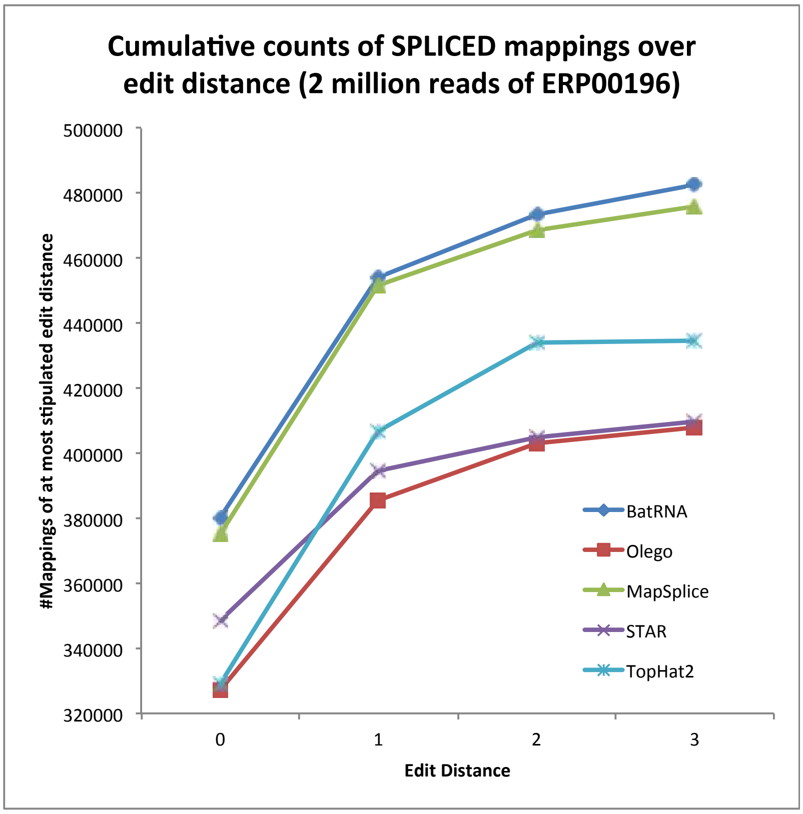


Figure 6.5. The cumulative counts, over edit distances of 0-3, of all non-ambiguous spliced mappings from the various spliced mappers on 2 million real reads taken from Sample 11T of ERP00196.

The results from the section on simulated data now coincide with the results shown in Figure 6.5. Without loss of generality, BatRNA was reported as the top performing method for both simulated and spliced alignments.

6.5 Evaluation on running time

The same sample of 2 millions reads, from the simulated datasets and patient 11T of ERP00196, were used to determine the runtime efficiency of the compared methods. The index-loading time was not recorded, as it does not reflect mapping efficiency and will be amortized to negligible timing over an actual life-sized dataset. The start time, wall-clock times, were recorded when the threads reached ~100% efficiency (indicating the complete loading of the primary index of the reference genome) and the end times were marked by the termination of their execution. Table 6.4 reports the recorded wall-clock times for dataset of different origins and read-lengths. STAR is the fastest method due to the search for MMP on a 29.8 GB human reference genome index. The runner-up method, in terms of running time, would be BatRNA. We also executed Tophat2 with the parameters "--no-sort-bam" and "--no-convert-bam" to avoid it from incurring additional execution times due to non-mapping related operations.

Table 6.4. Wall-clock time of compared methods on different sets of 2 million reads.

	Runtime on 2 million reads (seconds)		
Method	BEER 76 bp	BEER 100 bp	Real 90 bp
BatRNA	72	82	92
OLego	239	237	272
MapSplice	235	277	418
STAR	13	14	11
TopHat2	630	709	694

We have also observed that although only ~20% of the reads will have their primary candidate locations passed onto the second phase, this small portion of reads will take up more than 80% of the total runtime needed to run datasets with read-length of ~100 bp. Overall, BatRNA offers considerable improvements in alignment efficiency over the other compared methods with similar physical working memory footprint of << 30 GB.

6.6 Methods

BatRNA (Basic Alignment Tool for RNA-seq) was developed to address the issues of efficiency and accuracy on performing spliced-alignment of RNA-seq reads. BatAlign was used to align and produce candidate alignments of the reads. By emulating paired-end information in a single-read, an efficient pairing data structure was used to exhaustively search for the presence of splice junctions in a read. The putative mappings from both BatAlign and the splice-detection algorithm were then ranked according to their alignment score and reported to the users.

6.6.1 Simulation of data and validation of simulated data

The BEERS package in RUM is used to simulate the RNA-seq reads used in our benchmarks. 2 millions reads are simulated for current popular read lengths of 76bp and 100 bp. We have used BEERS as the simulator as it is built on an extensive platform of oracle information from 11 sets of annotations, namely, AceView, Ensembl, Geneid, Genscan, NSCAN, RefSeq, SGP, Transcriptome, UCSC, Vega, Other RefSeq databases. BEERS was also trained from these annotations and is able to simulate ~1.7M exons and ~1.1 introns, based on ~672K distinct gene models, with ground truth for validation.

For the aligned location of the simulated reads to be considered correct, the reported locations must be within 50 bp of the locations as generated by the simulator. For simulated spliced reads, in additional to the condition required for simulated exonic reads to be considered correct, we required that at least one of its simulated intronic gap(s) correctly identified before its reported alignment is considered correct, else our verifier will consider the reported alignment as an erroneous alignment.

We have also further broken down the mixture of exonic and spliced reads in the BEERS-simulated dataset for clearer illustrations of how the compared spliced aligner perform on each type of these reads.

6.6.2 Overview of Method

RNA-seq aligners can be classified into two main approaches: Exon-first and Seed-extend approaches. To the best of our knowledge, BatRNA is the first method that uses a pre-mapping tools meant for DNA-seq reads but can still be considered as a seed-extend approach. The reason that we developed BatRNA as a seed-extend approach is that it provides an unbiased mapping over both exonic and spliced reads. BatRNA is a three-phased method whereby the first phase is to find the list of candidate locations for the contiguous exonic region within a read, the second phase is to map the unmapped reads or low quality hits from the initial phase using a k-mer splicing seed-extend strategy and the last phase is to refine the alignments, from the previous phases, to identify splice junctions accurately. Figure 6.6 shows the schematic workflow of aligning RNA-seq reads using the methodology implemented in BatRNA.

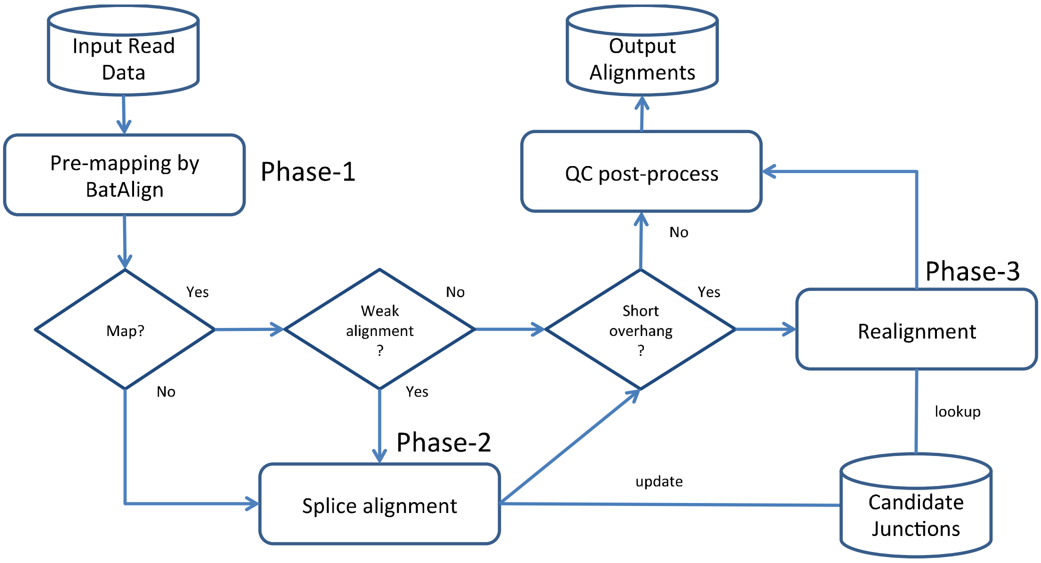


Figure 6.6. A schematic flowchart showing how input RNA-seq reads is aligned using the 3-phased methodology of BatRNA.

6.6.3 Motivation for using BatAlign as a seeding tool

As we have observed from BEERS-simulated reads of 100 bp long, ~62% of them can be mapped as if they are DNA-seq reads without the presence of large intronic gaps within them. Aside from this, less than 5% of exons have lengths shorter than 42 bases. This means that BatAlign can be used to seed the mapping location of the longest exonic region within a junction read successfully with its long mismatch-gapped seed on a 100 bp RNA-seq fragment.

The efficacy of using BatAlign is further highlighted by its ability to accurately align genomic reads. On the dataset of 100 bp RNA-seq reads, it was able to map 97.3% and 74.4% of the simulated exonic and spliced junction reads with an accuracy of 99.8% on the exonic reads; leaving only 9.4% of the dataset unmapped. The percentage of reads, from a general 100 bp RNA-seq dataset, that was delegated onto the later phases of BatRNA to align is ~18%.

6.6.4 Phase 1 – Resolve exonic region within a single read

The first phase of BatRNA is to use BatAlign as a seeding tool to align the input RNA-seq reads. For the putative alignments from BatAlign, we will assign a mapping quality score and a text-edit CIGAR string to each of them. If the mapping quality score is low, Phase-2 and Phase-3 of BatRNA will remap these reads, and this could mean three things. Firstly, the putative exonic alignment is repetitive due to its location in repeat or pseudogenic regions. Secondly, the alignment is weak due to high number of text-edit operations required to align the read back onto the reference genome. Thirdly, the putative alignment is heavily clipped with only a small percentage of the read being aligned to the reference genome by a local alignment routine. The last, unmentioned and trivial case of a read from BatAlign is that it is left unmapped by BatAlign.

Figure 6.7 shows the possible alignments of an RNA-seq read which spans across single or multiple exon-exon boundaries. This figure shows the possibility of using a DNA-seq gapped aligner as a pre-mapping tool for RNA-seq reads but still retains the unbiased alignment property towards both exon and junction reads of the seed-extend methodology towards RNA-seq alignment. Unlike TopHat1/2 that only realign primary candidate alignments with a certain threshold of edit distance on them from pre-mapping tools, BatRNA takes into account for the existence of splice junctions even at the DNA-seq gapped alignment step.

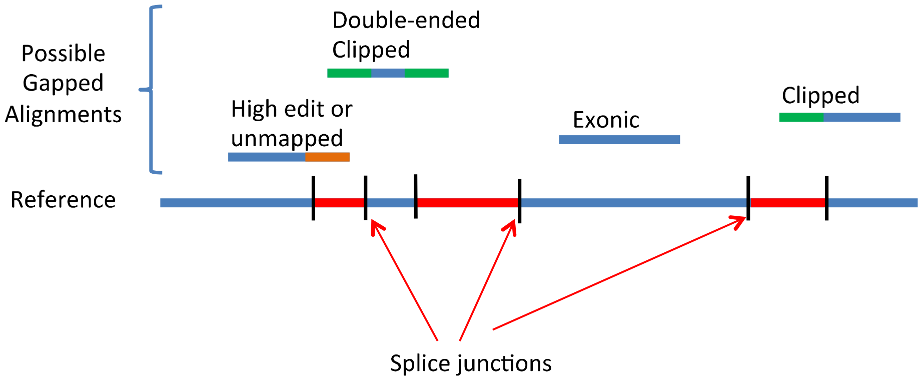


Figure 6.7. Possible alignments on RNA-seq read from BatAlign.

An example CIGAR string given to a simulated junction read by BatAlign would be “66M34S”. The largest matching segment will be the first 66 bases of the read and this matching sequence will be treated as an exonic transcript that lies on the left of an exon-exon boundary of the sample transcriptome. As described later, Phase-2 of BatRNA will align the rest of the remaining clipped 34 bases of the read downstream, at most 20 kbp away, of the anchored longest exonic region of this read.

6.6.5 Phase 2 – Search for junctions from an anchored region

There are two types of reads that will be passed into this phase of BatRNA: low-quality alignments and unmapped alignments from Phase-1. Figure 6.8 shows a flowchart on how these two types of reads are processed by the splice alignment algorithm in BatRNA. We will start to explain Phase-2 with the similar but simpler case of unmapped alignments. BatRNA’s splice mapping algorithm is based on a perfect-matching seed-extend-pairing strategy. It will first align the first two adjacent non-overlapping 18-mer of a read segment. If the first 18-mer cannot be anchored, then 5 bases are trimmed from the 5’ end of the read for each time it fails to anchor itself. If the first 18-mer is anchored then the second 18-mer of the read will be aligned and be paired using an efficient pairing (shown in Figure 6.9a) data structure to form a 36 bp exonic segment of the read. However, if the immediate 18-mer cannot be anchored (shown in Figure 6.9b), Phase-2 will try to pair the anchored portion of the read with the next adjacent and non-overlapping 18-mer of the read, this will continue until the end of the read. In the event that two 18-mer can be paired up successfully within the neighbor of each other (within 20 kbp), we will extend the paired candidate alignments of the two 18-mers, called gap-filling (shown in Figure 6.9c), towards each other, while respecting the donor-recipient canonical/non-canonical splicing signals. In the event, whereby there are more than one possible candidate location which can be paired with the anchored region of the read, the splice junctions detected by the gap-filling procedure are stored in a candidate junction files for use in Phase-3.

The second type of input to Phase-2 is partial-alignments from Phase-1. If the longest contiguous matching sequence of the partial-alignment is at least 25 bp then the partial-alignment is discarded and the read is treated as unmappable by Phase-1. We have decided on this threshold because the smallest seed used in BatAlign is 25 bp. From here, the longest contiguous matched sequence is treated as the anchored alignment and due to the presence of clippings in the partial-alignments, a junction is assumed to exist within or before the next 18-mer that needs to be aligned. Hence, the second 18-mer away from the clipped location will be aligned and paired with the already anchored partial alignment. If the 18-mer can be paired up then it will be extended towards the previously found partial-alignment (as shown in Figure 6.9d); if not, the algorithm will recursively proceed to the next non-overlapping 18-mer as described in the preceding paragraph.

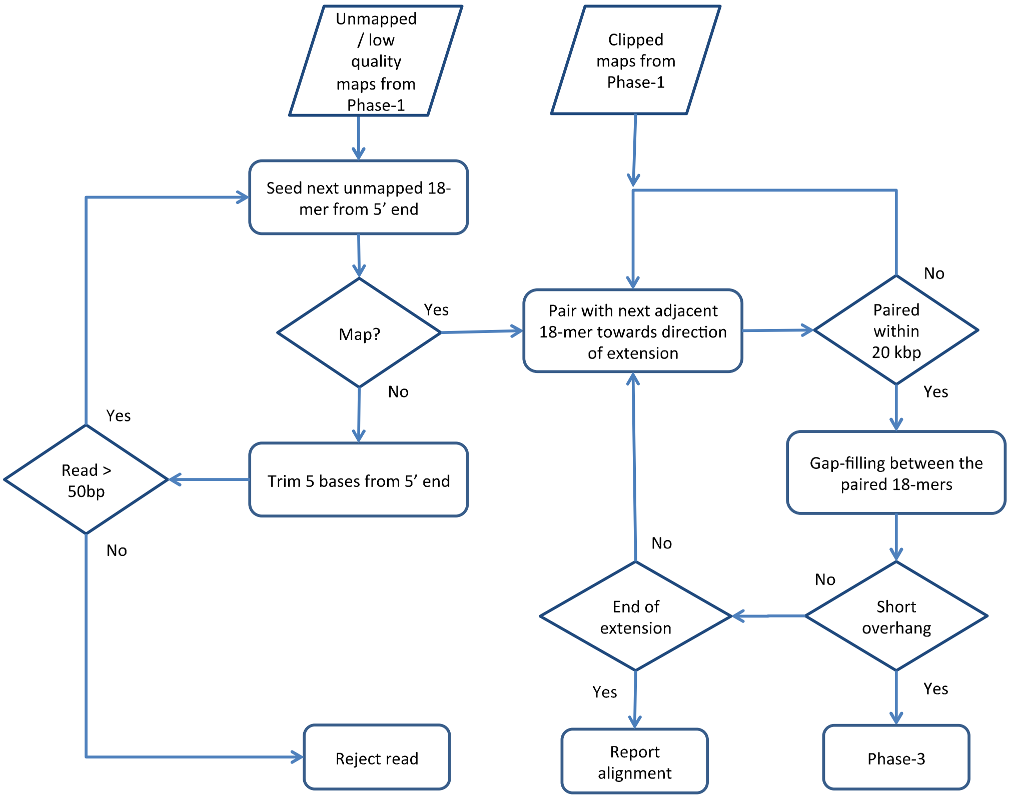

Unmapped / low quality maps from Phase-1
Clipped maps from Phase-1
Seed next unmapped 18-mer from 5' end
Map?
Yes
No
Pair with next adjacent 18-mer towards direction of extension
Paired within 20 kbp
No
Yes
Gap-filling between the paired 18-mers
Short overhang
No
Yes
End of extension
No
Yes
Report alignment
Phase-3
Trim 5 bases from 5' end
Read > 50bp
Yes
No
Reject read

Figure 6.8: A flowchart showing how the splice alignment algorithm in BatRNA performs splice alignment.

During the extension of the partial-alignments, short-overhanging exons can exist at the ends of the reads that are due to the presence of splice junctions being sequenced into the near-ends of the reads, these short-overhangs will be soft-clipped by Phase-2. Phase-3 will refine these exon-exon junctions that appears as soft-clippings in the reads from both Phase-1 and Phase-2.

6.6.6 Phase 3 – Refine alignments due to splice junctions near ends of reads

Alignment has always been an independent event between reads until TopHat devised the idea of exon-islands to localize putative splice junctions without annotations. By assembling the consensus of regions covered by the alignments of exonic reads from a

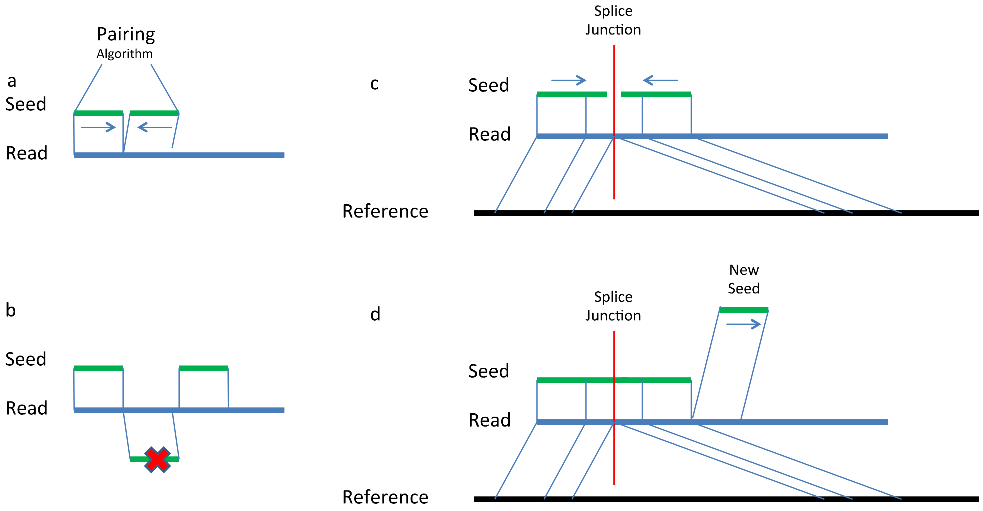


Figure 6.9. Schematic sketches of some possible scenarios that can happen in BatRNA splice algorithm. a) Adjacent non-overlapping seeds do not span across exon-exon junctions. b) Anchored seed is near to an exon-exon junction and next immediate 18-mer is used to seed the alignment. c) After successfully pairing of seeds within spanning distance of 20 kbp, alignments are extended towards each other to recover the splice junction on the reference genome. d) New seed is selected for the continual extension of a current partially anchored alignment.

Bowtie, exon islands can be obtained. Splice-junctions are then localized near the vicinity of these exon-islands.

Different from TopHat, the gap-filling component in Phase-2 of BatRNA has already identified the putative splice junctions. The unsupervised learning of splice junctions is done whenever two adjacent non-overlapping 18-mers from a read are aligned more than 20 bp apart and a gap-filling procedure is done to identify the splice junctions. These splice junctions are stored in a putative bed-coverage file for Phase-3 to refine the alignments of short-overhangs. For instance, the cigar string "12M2439N88M" was previously "11S89M" for read "CGAGAGCTAAAGGAGGTCTTTGGTGATGAC TCTGAGATCTCTAAAGAATCATCAGGAGTAAAGAAGCGACGAATACCCCGTT TTGAGGAGGTGGAACAAG". The 11 clipped bases are then locally aligned to each of the candidate splice junctions, within 20 kbp of the anchored 89 contiguous exonic bases, recorded by Phase-2. Figure 6.10 shows the possible inputs into Phase-3 that are realigned around a putative splice junction from the splice alignment algorithm in Phase-2. After which, the mappings are scored similarly with a scoring function similar to BatAlign. In the event that there are more than two candidate alignments to a read, the donor-recipient splicing signals will precede over the total intronic gap sizes in an alignment as a tiebreaker.

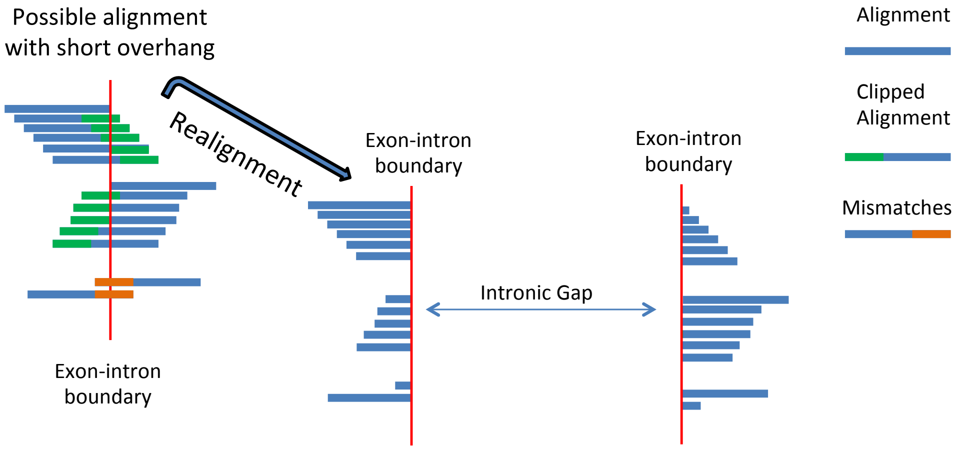


Figure 6.10. Possible short overhangs being recovered with local alignment by using preceding prediction as a guide in an unsupervised manner.

The coverage or the transcript-abundance of the dataset simulated or sequenced will matter for the performance of this realignment step. If the depth of coverage is low, Phase-2 may not be able to detect the splicing junctions needed for the realignment of short-overhang in a read. As such, the same read with alignment CIGAR “11S89M” will be left unchanged after Phase-3 is complete.

6.6.7 Data structure for efficient pairing of genomic coordinates

As the entire array of genomic locations is too large to fit into the main memory of common personal computers, genomic locations are often sampled at a fixed *k* interval to keep the index compact. However, after the alignment of a read is done on the suffix array, the intermediate suffix-interval can only be converted to a genomic location by referencing the sampled array in $O(\log|Reference|)$ time. If s is the number of steps needed to invert the suffix-interval back onto a sampled location, then the actual location of the alignment can be calculated from $[(Sampled\ location)\ - s]$. As each occurrence of the aligned read needs to be inverted back to a genomic location separately and independently, the total time needed to find all the genomic locations represented by the suffix-intervals is $O(|Occurrences|\ .\log|Reference|)$ time.

However, if we pre-compute and hash the genomic locations of the k-length string, the genomic locations can be retrieved in $O(|Occurrences|\ + \log|Reference|)$ time instead. Furthermore, if we pre-process the hashed genomic locations by sorting them, we can pair the genomic locations for two k-length strings within a distance D in $O(|Occurrences|)$ time. The data-structure used is a hash-map with the suffix-array interval and genomic locations as a key and value pair respectively. The building of this data structure is only a 1-time off effort. In order to avoid large memory overhead, only strings that have occurrences of more than 1 and less than 200 are hashed by our index-building routine. This data structure will incur 2.5 GB of the total memory footprint of BatRNA.

6.6.8 Details of implementation

The length of the seed is chosen as 18 bp long to represent more than 99% of UCSC RefSeq exon-lengths on a genomic reference without spanning over an exon-exon boundary [165, 166]. As 18 bp fragment can be over-represented spuriously on the reference genome for BatRNA to align efficiently, we do not allow any mismatches or/and gap in the 18-mer seed in our splice alignment algorithm. For the efficiency of the method, the maximum distance between adjacent exons within a read has to be within 20 kbp to each other. This threshold on the intronic gap size constitutes for less than 6.1% of the human genome [229].

6.6.9 Discussion

In this chapter, we represented BatRNA; a method that emulates pair-end information within a single RNA-seq read for efficient alignment of high throughput datasets from next generation sequencing technologies. Since the introduction of high throughput sequencing technologies, the dominant improvements brought about by it are increase in throughput and read-length. However, with longer read-lengths, RNA-seq alignment algorithms has to be developed to account for the long intronic gap that can exist in a RNA-seq read which are much larger than an indel gap in DNA-seq read. In order to handle these large intronic gaps, pioneering alignment tools for RNA-seq reads align reads gaplessly onto a pre-constructed reference of known transcripts, which is also known as transcriptome. This strategy is very efficient as gapless alignment can be performed efficiently using gapless BWT-based aligner in *O(Read-Length)* time for each read. However, using the transcriptome as the reference text to align RNA-seq reads with a gapless aligner will void us of doing de novo detection of novel splice junctions. Albeit a non *de novo* methodology, this strategy was extended and gave birth to the development of popular exon-first strategy such as TopHat (using Bowtie as premapping tool). By weeding out the exonic seeds out before the computationally dominant splice alignment algorithms align the unmapped spliced reads, exon-first approaches generally align faster than seed-extend approaches.

The main shortcoming of exon-first is that it favors towards the alignment of exonic reads over spliced reads. In other words, exonic reads from RNA-seq experiments may align more often to pseudogenic regions erroneously than seed-extend methods. TopHat2 tried to minimize this error rate by realigning reads of a certain threshold of edit-distance (capped at 3) from its DNA-seq gapped aligner through its seed-extend splice alignment routine in hope of realigning the same read with an exonic alignment to a splice alignment instead. Seed-extend was introduced by pioneering methods such as BLAT and exonerate to unbiasedly map RNA-seq reads regardless of the existence of splice junctions in the reads to the genomic reference. By picking the correct seed length and the intervals between each subsequent seeds on a read are critical to the success of a seed-extend approach. A long read-length will cause the seed to incur a high edit distance and miss the correct alignment. The lengths of the seeds are generally short in order to achieve good sensitivity. Specificity is dependent on how well the seeds are sampled such that the short seeds can capture the correct alignment of the read. For instance, BLAT indexes all the non-overlapping 11 bp tiles to achieve good alignment of long EST/cDNA sequences [32]. Additional heuristics such as perfect seed matches are also used to limit the number of preliminary partial alignments for post-processing for reasonable alignment efficiency.

BatRNA was developed as a hybrid between exon-first and seed-extend approaches. This was done to complement the shortcomings of both approaches under one unified method. In order to reduce the number of computationally expensive splice alignments needed to recover splice junctions, Phase-1 uses a gapped aligner that can align ~91% of reads in a general 100 bp read-length RNA-seq dataset. From here, the unmapped and low quality alignments from Phase-1 are realigned with our splice algorithm. Next, we reckon that 18 bp will produce a lot of spurious partially alignment to post-process and, as such, an efficient pairing data-structure was developed to align pairs of 18 bp in our splice alignment efficiently. Using our pairing data structure, we are effectively aligning a much longer seed of 36 bp (2 x 18 bp) than the current seed-extend methods seed-lengths of 10-25 bp such as SubRead [150].

In summary, our experiments have shown that BatRNA have achieved better sensitivity and specificity in handling RNA-seq reads of current common read-lengths on a reference genome than other compared methods. BatRNA is also the most efficient program among the compared methods of similar memory footprints.

Chapter 7

Conclusion

In this chapter, we will review on the main contributions of this thesis and discuss future developments that can be adopted to improve on the proposed methodologies.

The purpose of this thesis is to report on methodologies that provide accurate alignment of sequence reads from various genomic origins back onto a reference genome. In the earlier chapters of this thesis, we have reported on the methodologies that aimed at accurate alignment of bisulfite-treated, gapped and spliced reads.

7.1 BatMeth

The alignment of a read, against a reference index, comprises of two main sequential steps. The first step involves the retrieval of the suffix-array intervals that represents the occurrences of the read in the reference-index. The second step, where the bulk of computation takes place, involves the conversion of the suffix-array intervals to human-readable genomic locations.

In BatMeth, *List Filtering* performs the alignment of a read solely by counting the number of occurrences of the reads on each possible orientation of the reference-text without the need to know the exact genomic locations of the reads. *List Filtering* improved the sensitivity, specificity and speed of our proposed algorithm. BatMeth was also developed to account for mismatches attributed from deamination or/and sodium bisulfite-induced base conversion in both base-space (Illumina reads) and color-space reads (SOLiD reads). Experiments have also shown that BatMeth aligned reads with less bias on different genomic contexts (CG, CHH, CHG; where H ≠ G) and different levels of methylation. Bisulfighter [230] has also reviewed on BatMeth saying that it is the best method in deciding whether a base is methylated or not.

7.2 BatAlign

The pioneering aligners for next generation sequencing reads were originally designed to handle only mismatches in the query reads with respect to the reference genome. As genomic polymorphisms can also comprise of indels and genomic rearrangements, gapped aligners were developed to better study the complex nature of polymorphisms in both normal and diseased genomes.

Since a query read can be transformed back to the reference genome through a sequential order of text-edit operations, we can score such a transformation by assigning scores to each of the available text-edit operations. The text-edit operations mainly comprised of the match, mismatch, gap-open and gap-extend steps. By enumerating the number and order of text-edit operations to a read, we can score such a transformation that is used to align the read back onto a reference text. BatAlign uses *Reverse-alignment* to incrementally align a read in increasing order of alignment cost as defined by the combination of match/mismatch/gap scores needed for the alignment of a read. In addition, *Deep-scan* was also developed to better differentiate a real-SNP mismatch from a false base-call mismatch during on-the-fly alignment of a read. Experiments have shown that BatAlign was able to map highly polymorphic reads (75 to 250 bp long) with high sensitivity and accuracy over a large range of mapQ scores.

Paired-reads were also first aligned as single-read and were later paired up unbiasedly to yield accurate alignments. Chimeric/supplementary alignments were also reported for a single read, under the pair-end mapping mode of BatAlign, to better support the identification of breakpoints caused by genomic rearrangements. In general, BatAlign is an improved method for gapped reads.

7.3 BatRNA

The advent of RNA-seq data allowed scientists to quantify gene expression on a genome-wide scale. As RNA-seq reads can singly span across different exons, they can be challenging to be aligned back onto a reference genome.

BatRNA was developed as a hybrid of both exon-first and seed-extend methodologies. BatAlign was used as a non-spliced pre-mapping aligner. The main splice-alignment routine, which emulated paired-end information within a single read, was used to realign the clipped and unmapped reads from BatAlign. Our experiments have shown that BatRNA has achieved accurate alignments on both exonic and spliced reads from the simulation and real data. It is also reasonably efficient when compared to other methods of similar memory usage.

7.4 Future Developments

As sequencing technologies continue to advance, the error profiles of reads that are inherent to these technologies will also change with them. For instance, when the read length gets longer in the third generation sequencing technologies, the total edit-distance in a single read will require redevelopment of existing algorithms to better handle such challenges. Homopolymers can be a prevalent type of sequencing errors over erroneous single-base-calls too and aligners will have to be redesigned for the efficient and accurate resolution of such errors.

Alongside with sequencing technologies and alignment algorithms, genomic assemblers are also producing better assemblies of the reference genomes. For instance, the recent release of the GRCh38 human genome has included 261 new alternate loci, which are highly similar to the main loci of the GRCh37 genome. The alignment of reads to the newer, GRCh38, genome will directly yield more non-unique alignments as these newer 261 alternate loci are highly similar to the main chromosomal sequences. In the future, aligners should be aware if an alignment is from either the main or alternate loci. Multi-maps from within the main chromosomes and, between the main and alternate chromosomes should be treated differently.

Lastly, the alignment of genomic reads can also be tackled from the reference index's point of view. Compressed data structures can be developed to compress multiple reference indices into a single succinct index of the former, for more efficient memory usage when aligning to two or more organisms' reference genomes at once.

Bibliography

1. Darwin C: **On the origins of species by means of natural selection.** *London: Murray* 1859.
2. Mendel G: **Versuche über Pflanzenhybriden.** *Verhandlungen des naturforschenden Vereines in Brunn 4: 3* 1866, **44**.
3. Avery OT, MacLeod CM, McCarty M: **Studies on the chemical nature of the substance inducing transformation of pneumococcal types induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III.** *The Journal of experimental medicine* 1944, **79:**137-158.
4. Watson JD, Crick FH: **Molecular structure of nucleic acids.** *Nature* 1953, **171:**737-738.
5. Franklin RE, Gosling RG: **Evidence for 2-chain helix in crystalline structure of sodium deoxyribonucleate.** *Nature* 1953, **172:**156-157.
6. Wilkins MH, Seeds WE, Stokes AR, Wilson HR: **Helical structure of crystalline deoxypentose nucleic acid.** *Nature* 1953, **172:**759-762.
7. Sanger F: **DNA Sequencing with Chain-Terminating Inhibitors.** *Proceedings of the National Academy of Sciences* 1977, **74:**5463-5467.
8. Maxam AM: **A New Method for Sequencing DNA.** *Proceedings of the National Academy of Sciences* 1977, **74:**560-564.
9. Noble I: **Human genome finally complete.** In *BBC News*; 2003.
10. Maher B: **ENCODE: The human encyclopaedia.** *Nature* 2012, **489:**46-48.
11. Bernstein BE, Birney E, Dunham I, Green ED, Gunter C, Snyder M: **An integrated encyclopedia of DNA elements in the human genome.** *Nature* 2012, **489:**57-74.
12. Sanger F, Air GM, Barrell BG, Brown NL, Coulson AR, Fiddes CA, Hutchison CA, Slocombe PM, Smith M: **Nucleotide sequence of bacteriophage phi X174 DNA.** *Nature* 1977, **265:**687-695.
13. Sanger F, Coulson AR: **A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase.** *J Mol Biol* 1975, **94:**441-448.
14. Liu L, Li Y, Li S, Hu N, He Y, Pong R, Lin D, Lu L, Law M: **Comparison of next-generation sequencing systems.** *J Biomed Biotechnol* 2012, **2012:**251364.
15. Shendure J, Porreca GJ, Reppas NB, Lin X, McCutcheon JP, Rosenbaum AM, Wang MD, Zhang K, Mitra RD, Church GM: **Accurate multiplex polony sequencing of an evolved bacterial genome.** *Science* 2005, **309:**1728-1732.
16. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Berka J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes XV, Godwin BC, He W, Helgesen S, Ho CH, Irzyk GP, Jando SC, Alenquer ML,

Jarvie TP, Jirage KB, Kim JB, Knight JR, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, et al: **Genome sequencing in microfabricated high-density picolitre reactors.** *Nature* 2005, **437:**376-380.

17. Bentley DR, Balasubramanian S, Swerdlow HP, Smith GP, Milton J, Brown CG, Hall KP, Evers DJ, Barnes CL, Bignell HR, Boutell JM, Bryant J, Carter RJ, Cheetham RK, Cox AJ, Ellis DJ, Flatbush MR, Gormley NA, Humphray SJ, Irving LJ, Karbelashvili MS, Kirk SM, Li H, Liu XH, Maisinger KS, Murray LJ, Obradovic B, Ost T, Parkinson ML, Pratt MR, et al: **Accurate whole human genome sequencing using reversible terminator chemistry.** *Nature* 2008, **456:**53-59.

18. McKernan KJ, Peckham HE, Costa GL, McLaughlin SF, Fu Y, Tsung EF, Clouser CR, Duncan C, Ichikawa JK, Lee CC, Zhang Z, Ranade SS, Dimalanta ET, Hyland FC, Sokolsky TD, Zhang L, Sheridan A, Fu H, Hendrickson CL, Li B, Kotler L, Stuart JR, Malek JA, Manning JM, Antipova AA, Perez DS, Moore MP, Hayashibara KC, Lyons MR, Beaudoin RE, et al: **Sequence and structural variation in a human genome uncovered by short-read, massively parallel ligation sequencing using two-base encoding.** *Genome Res* 2009, **19:**1527-1541.

19. Ronaghi M: **DNA SEQUENCING:A Sequencing Method Based on Real-Time Pyrophosphate.** *Science* 1998, **281:**363-365.

20. Eid J, Fehr A, Gray J, Luong K, Lyle J, Otto G, Peluso P, Rank D, Baybayan P, Bettman B, Bibillo A, Bjornson K, Chaudhuri B, Christians F, Cicero R, Clark S, Dalal R, Dewinter A, Dixon J, Foquet M, Gaertner A, Hardenbol P, Heiner C, Hester K, Holden D, Kearns G, Kong X, Kuse R, Lacroix Y, Lin S, et al: **Real-time DNA sequencing from single polymerase molecules.** *Science* 2009, **323:**133-138.

21. Drmanac R, Sparks AB, Callow MJ, Halpern AL, Burns NL, Kermani BG, Carnevali P, Nazarenko I, Nilsen GB, Yeung G, Dahl F, Fernandez A, Staker B, Pant KP, Baccash J, Borcherding AP, Brownley A, Cedeno R, Chen LS, Chernikoff D, Cheung A, Chirita R, Curson B, Ebert JC, Hacker CR, Hartlage R, Hauser B, Huang S, Jiang Y, Karpinchyk V, et al: **Human Genome Sequencing Using Unchained Base Reads on Self-Assembling DNA Nanoarrays.** *Science* 2010, **327:**78-81.

22. Thompson JF, Steinmann KE: **Single molecule sequencing with a HeliScope genetic analysis system.** *Curr Protoc Mol Biol* 2010, **Chapter 7:**Unit7 10.

23. Ng SB, Turner EH, Robertson PD, Flygare SD, Bigham AW, Lee C, Shaffer T, Wong M, Bhattacharjee A, Eichler EE, Bamshad M, Nickerson DA, Shendure J: **Targeted capture and massively parallel sequencing of 12 human exomes.** *Nature* 2009, **461:**272-276.

24. Ng SB, Buckingham KJ, Lee C, Bigham AW, Tabor HK, Dent KM, Huff CD, Shannon PT, Jabs EW, Nickerson DA, Shendure J, Bamshad MJ: **Exome sequencing identifies the cause of a mendelian disorder.** *Nat Genet* 2010, **42:**30-35.

25. Acinas SG, Sarma-Rupavtarm R, Klepac-Ceraj V, Polz MF: **PCR-induced sequence artifacts and bias: insights from comparison of two 16S rRNA clone libraries constructed from the same sample.** *Appl Environ Microbiol* 2005, **71:**8966-8969.

26. Ingram VM: **A Specific Chemical Difference Between the Globins of Normal Human and Sickle-Cell Anæmia Hæmoglobin.** *Nature* 1956, **178:**792-794.

27. Niidome T, Huang L: **Gene therapy progress and prospects: nonviral vectors.** *Gene Ther* 2002, **9:**1647-1652.

28. Riddihough G, Zahn LM: **Epigenetics. What is epigenetics? Introduction.** *Science* 2010, **330:**611.
29. Bernstein BE, Meissner A, Lander ES: **The mammalian epigenome.** *Cell* 2007, **128:**669-681.
30. Treangen TJ, Salzberg SL: **Repetitive DNA and next-generation sequencing: computational challenges and solutions.** *Nat Rev Genet* 2012, **13:**36-46.
31. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool.** *J Mol Biol* 1990, **215:**403-410.
32. Kent WJ: **BLAT - The BLAST-like alignment tool.** *Genome Res* 2002, **12:**656-664.
33. Wang Z, Gerstein M, Snyder M: **RNA-Seq: a revolutionary tool for transcriptomics.** *Nat Rev Genet* 2009, **10:**57-63.
34. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR: **STAR: ultrafast universal RNA-seq aligner.** *Bioinformatics* 2013, **29:**15-21.
35. Wu J, Anczukow O, Krainer AR, Zhang MQ, Zhang C: **OLego: fast and sensitive mapping of spliced mRNA-Seq reads using small seeds.** *Nucleic Acids Research* 2013.
36. Wang K, Singh D, Zeng Z, Coleman SJ, Huang Y, Savich GL, He X, Mieczkowski P, Grimm SA, Perou CM, MacLeod JN, Chiang DY, Prins JF, Liu J: **MapSplice: accurate mapping of RNA-seq reads for splice junction discovery.** *Nucleic Acids Research* 2010, **38:**e178.
37. Zhang Y, Lameijer EW, t Hoen PA, Ning Z, Slagboom PE, Ye K: **PASSion: a pattern growth algorithm-based pipeline for splice junction detection in paired-end RNA-Seq data.** *Bioinformatics* 2012, **28:**479-486.
38. De Bona F, Ossowski S, Schneeberger K, Ratsch G: **Optimal spliced alignments of short sequence reads.** *Bioinformatics* 2008, **24:**i174-180.
39. Au KF, Jiang H, Lin L, Xing Y, Wong WH: **Detection of splice junctions from paired-end RNA-seq data by SpliceMap.** *Nucleic Acids Research* 2010, **38:**4570-4578.
40. Trapnell C, Pachter L, Salzberg SL: **TopHat: discovering splice junctions with RNA-Seq.** *Bioinformatics* 2009, **25:**1105-1111.
41. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL: **TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions.** *Genome Biol* 2013, **14:**R36.
42. Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L: **Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks.** *Nature Protocols* 2012, **7:**562-578.
43. Li B, Dewey CN: **RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome.** *Bmc Bioinformatics* 2011, **12:**323.
44. Anders S: **HTSeq: Analysing high-throughput sequencing data with Python.** *URL http://www-huber embl de/users/anders/HTSeq/doc/overview html* 2010.
45. Bohnert R, Ratsch G: **rQuant.web: a tool for RNA-Seq-based transcript quantitation.** *Nucleic Acids Research* 2010, **38:**W348-351.
46. Nicolae M, Mangul S, Mandoiu, II, Zelikovsky A: **Estimation of alternative splicing isoform frequencies from RNA-Seq data.** *Algorithms Mol Biol* 2011, **6:**9.
47. Szulwach KE, Li XK, Li YJ, Song CX, Han JW, Kim S, Namburi S, Hermetz K, Kim JJ, Rudd MK, Yoon YS, Ren B, He C, Jin P: **Integrating 5-**

Hydroxymethylcytosine into the Epigenomic Landscape of Human Embryonic Stem Cells. *Plos Genetics* 2011, **7**.

48. Crick F: **Central dogma of molecular biology.** *Nature* 1970, **227:**561-563.

49. Williams R, Peisajovich SG, Miller OJ, Magdassi S, Tawfik DS, Griffiths AD: **Amplification of complex gene libraries by emulsion PCR.** *Nature Methods* 2006, **3:**545-550.

50. Schuster SC: **Next-generation sequencing transforms today's biology.** *Nature Methods* 2008, **5:**16-18.

51. Ronaghi M, Karamohamed S, Pettersson B, Uhlen M, Nyren P: **Real-time DNA sequencing using detection of pyrophosphate release.** *Anal Biochem* 1996, **242:**84-89.

52. Pennisi E: **Genomics. Semiconductors inspire new sequencing technologies.** *Science* 2010, **327:**1190.

53. Purushothaman S, Toumazou C, Ou CP: **Protons and single nucleotide polymorphism detection: A simple use for the ion sensitive field effect transistor.** *Sensors and Actuators B-Chemical* 2006, **114:**964-968.

54. Metzker ML: **Emerging technologies in DNA sequencing.** *Genome Res* 2005, **15:**1767-1776.

55. Sambrook J, Russell DW: **Fragmentation of DNA by sonication.** *CSH Protoc* 2006, **2006**.

56. Sambrook J, Russell DW: **Fragmentation of DNA by nebulization.** *CSH Protoc* 2006, **2006**.

57. Adessi C, Matton G, Ayala G, Turcatti G, Mermod JJ, Mayer P, Kawashima E: **Solid phase DNA amplification: characterisation of primer attachment and amplification mechanisms.** *Nucleic Acids Research* 2000, **28:**E87.

58. Mardis ER: **Next-generation DNA sequencing methods.** *Annu Rev Genomics Hum Genet* 2008, **9:**387-402.

59. Huang YF, Chen SC, Chiang YS, Chen TH, Chiu KP: **Palindromic sequence impedes sequencing-by-ligation mechanism.** *BMC Syst Biol* 2012, **6 Suppl 2:**S10.

60. Quail MA, Smith M, Coupland P, Otto TD, Harris SR, Connor TR, Bertoni A, Swerdlow HP, Gu Y: **A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers.** *Bmc Genomics* 2012, **13:**341.

61. Nagarajan N, Bertrand D, Hillmer AM, Zang ZJ, Yao F, Jacques PE, Teo AS, Cutcutache I, Zhang Z, Lee WH, Sia YY, Gao S, Ariyaratne PN, Ho A, Woo XY, Veeravali L, Ong CK, Deng N, Desai KV, Khor CC, Hibberd ML, Shahab A, Rao J, Wu M, Teh M, Zhu F, Chin SY, Pang B, So JB, Bourque G, et al: **Whole-genome reconstruction and mutational signatures in gastric cancer.** *Genome Biol* 2012, **13:**R115.

62. Adams M, Kelley J, Gocayne J, Dubnick M, Polymeropoulos M, Xiao H, Merril C, Wu A, Olde B, Moreno R, et a: **Complementary DNA sequencing: expressed sequence tags and human genome project.** *Science* 1991, **252:**1651-1656.

63. Bernstein BE, Kamal M, Lindblad-Toh K, Bekiranov S, Bailey DK, Huebert DJ, McMahon S, Karlsson EK, Kulbokas EJ, 3rd, Gingeras TR, Schreiber SL, Lander ES: **Genomic maps and comparative analysis of histone modifications in human and mouse.** *Cell* 2005, **120:**169-181.

64. Bird AP: **CpG-rich islands and the function of DNA methylation.** *Nature* 1986, **321:**209-213.

65. Meissner A, Mikkelsen TS, Gu H, Wernig M, Hanna J, Sivachenko A, Zhang X, Bernstein BE, Nusbaum C, Jaffe DB, Gnirke A, Jaenisch R, Lander ES: **Genome-scale DNA methylation maps of pluripotent and differentiated cells.** *Nature* 2008, **454:**766-770.
66. Frommer M, Mcdonald LE, Millar DS, Collis CM, Watt F, Grigg GW, Molloy PL, Paul CL: **A Genomic Sequencing Protocol That Yields a Positive Display of 5-Methylcytosine Residues in Individual DNA Strands.** *Proceedings of the National Academy of Sciences of the United States of America* 1992, **89:**1827-1831.
67. Ferragina P, Manzini G: **Opportunistic data structures with applications.** *41st Annual Symposium on Foundations of Computer Science, Proceedings* 2000**:**390-398.
68. Burrows M, Wheeler D: **A block-sorting lossless data compression algorithm.** 1994.
69. Smith TF, Waterman MS: **Identification of common molecular subsequences.** *J Mol Biol* 1981, **147:**195-197.
70. Needleman SB, Wunsch CD: **A general method applicable to the search for similarities in the amino acid sequence of two proteins.** *J Mol Biol* 1970, **48:**443-453.
71. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R: **The Sequence Alignment/Map format and SAMtools.** *Bioinformatics* 2009, **25:**2078-2079.
72. Vingron M, Waterman MS: **Sequence alignment and penalty choice. Review of concepts, case studies and implications.** *J Mol Biol* 1994, **235:**1-12.
73. Ewing B, Hillier L, Wendl MC, Green P: **Base-calling of automated sequencer traces using phred. I. Accuracy assessment.** *Genome Res* 1998, **8:**175-185.
74. Li H, Ruan J, Durbin R: **Mapping short DNA sequencing reads and calling variants using mapping quality scores.** *Genome Res* 2008, **18:**1851-1858.
75. Pelizzola M, Ecker JR: **The DNA methylome.** *FEBS Lett* 2011, **585:**1994-2000.
76. Lim JQ, Tennakoon C, Li G, Wong E, Ruan Y, Wei CL, Sung WK: **BatMeth: improved mapper for bisulfite sequencing reads on DNA methylation.** *Genome Biol* 2012, **13:**R82.
77. Ondov BD, Cochran C, Landers M, Meredith GD, Dudas M, Bergman NH: **An alignment algorithm for bisulfite sequencing using the Applied Biosystems SOLiD System.** *Bioinformatics* 2010, **26:**1901-1902.
78. Krueger F, Andrews SR: **Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications.** *Bioinformatics* 2011, **27:**1571-1572.
79. Harris EY, Ponts N, Levchuk A, Roch KL, Lonardi S: **BRAT: bisulfite-treated reads analysis tool.** *Bioinformatics* 2010, **26:**572-573.
80. Harris EY, Ponts N, Le Roch KG, Lonardi S: **BRAT-BW: efficient and accurate mapping of bisulfite-treated reads.** *Bioinformatics* 2012, **28:**1795-1796.
81. Chen PY, Cokus SJ, Pellegrini M: **BS Seeker: precise mapping for bisulfite sequencing.** *BMC Bioinformatics* 2010, **11:**203.
82. Lee TF, Zhai J, Meyers BC: **Conservation and divergence in eukaryotic DNA methylation.** *Proc Natl Acad Sci U S A* 2010, **107:**9027-9028.
83. Smith AD, Chung WY, Hodges E, Kendall J, Hannon G, Hicks J, Xuan Z, Zhang MQ: **Updates to the RMAP short-read mapping software.** *Bioinformatics* 2009, **25:**2841-2842.

84. Campagna D, Telatin A, Forcato C, Vitulo N, Valle G: **PASS-bis: a bisulfite aligner suitable for whole methylome analysis of Illumina and SOLiD reads.** *Bioinformatics* 2013, **29:**268-270.

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

86. Kreck B, Marnellos G, Richter J, Krueger F, Siebert R, Franke A: **B-SOLANA: An approach for the analysis of two-base encoding bisulfite sequencing data (In Press).** *Bioinformatics* 2011.

87. Kondrashov AS, Rogozin IB: **Context of deletions and insertions in human coding sequences.** *Hum Mutat* 2004, **23:**177-185.

88. Ma L, Zhang TT, Huang ZR, Jiang XQ, Tao SH: **Patterns of nucleotides that flank substitutions in human orthologous genes.** *Bmc Genomics* 2010, **11**.

89. Sung WK, Zheng H, Li S, Chen R, Liu X, Li Y, Lee NP, Lee WH, Ariyaratne PN, Tennakoon C, Mulawadi FH, Wong KF, Liu AM, Poon RT, Fan ST, Chan KL, Gong Z, Hu Y, Lin Z, Wang G, Zhang Q, Barber TD, Chou WC, Aggarwal A, Hao K, Zhou W, Zhang C, Hardwick J, Buser C, Xu J, et al: **Genome-wide survey of recurrent HBV integration in hepatocellular carcinoma.** *Nat Genet* 2012, **44:**765-769.

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

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

92. Langmead B, Trapnell C, Pop M, Salzberg SL: **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.** *Genome Biol* 2009, **10:**R25.

93. Langmead B, Salzberg SL: **Fast gapped-read alignment with Bowtie 2.** *Nature Methods* 2012, **9:**357-359.

94. Li R, Li Y, Kristiansen K, Wang J: **SOAP: short oligonucleotide alignment program.** *Bioinformatics* 2008, **24:**713-714.

95. Li R, Yu C, Li Y, Lam TW, Yiu SM, Kristiansen K, Wang J: **SOAP2: an improved ultrafast tool for short read alignment.** *Bioinformatics* 2009, **25:**1966-1967.

96. Novocraft: **Novoalign.** www.novocraft.com.

97. Lunter G, Goodson M: **Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads.** *Genome Res* 2011, **21:**936-939.

98. Campagna D, Albiero A, Bilardi A, Caniato E, Forcato C, Manavski S, Vitulo N, Valle G: **PASS: a program to align short sequences.** *Bioinformatics* 2009, **25:**967-968.

99. Liu Y, Schmidt B, Maskell DL: **CUSHAW: a CUDA compatible short read aligner to large genomes based on the Burrows-Wheeler transform.** *Bioinformatics* 2012, **28:**1830-1837.

100. Liu Y, Schmidt B: **Long read alignment based on maximal exact match seeds.** *Bioinformatics* 2012, **28:**i318-i324.

101. Gontarz PM, Berger J, Wong CF: **SRmapper: a fast and sensitive genome-hashing alignment tool.** *Bioinformatics* 2013, **29:**316-321.

102. Mu JC, Jiang H, Kiani A, Mohiyuddin M, Bani Asadi N, Wong WH: **Fast and accurate read alignment for resequencing.** *Bioinformatics* 2012, **28:**2366-2373.

103. Cox A: **ELAND: Efficient Local Alignment of Nucleotide Data.** 2006.

104. Smith AD, Xuan ZY, Zhang MQ: **Using quality scores and longer reads improves accuracy of Solexa read mapping.** *Bmc Bioinformatics* 2008, **9**.

105. Rumble SM, Lacroute P, Dalca AV, Fiume M, Sidow A, Brudno M: **SHRiMP: accurate mapping of short color-space reads.** *PLoS Comput Biol* 2009, **5:**e1000386.

106. David M, Dzamba M, Lister D, Ilie L, Brudno M: **SHRiMP2: sensitive yet practical SHort Read Mapping.** *Bioinformatics* 2011, **27:**1011-1012.

107. Lin H, Zhang ZF, Zhang MQ, Ma B, Li M: **ZOOM! Zillions of oligos mapped.** *Bioinformatics* 2008, **24:**2431-2437.

108. Malhis N, Butterfield YS, Ester M, Jones SJ: **Slider--maximum use of probability information for alignment of short sequence reads and SNP detection.** *Bioinformatics* 2009, **25:**6-13.

109. Malhis N, Jones SJ: **High quality SNP calling using Illumina data at shallow coverage.** *Bioinformatics* 2010, **26:**1029-1035.

110. Farrar M: **Striped Smith-Waterman speeds database searches six times over other SIMD implementations.** *Bioinformatics* 2007, **23:**156-161.

111. Zhao M, Lee WP, Marth GT: **SSW Library: An SIMD Smith-Waterman C/C++ Library for Use in Genomic Applications.** *arXiv preprint arXiv:12086350* 2012.

112. Zhang Z, Schwartz S, Wagner L, Miller W: **A greedy algorithm for aligning DNA sequences.** *J Comput Biol* 2000, **7:**203-214.

113. Faust GG, Hall IM: **YAHA: fast and flexible long-read alignment with optimal breakpoint detection.** *Bioinformatics* 2012, **28:**2417-2424.

114. Baeza-Yates RA, Perleberg CH: **Fast and practical approximate string matching.** In *Combinatorial Pattern Matching*. Springer; 1992: 185-192.

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

116. Li H, Homer N: **A survey of sequence alignment algorithms for next-generation sequencing.** *Brief Bioinform* 2010, **11:**473-483.

117. Nakamura K, Oshima T, Morimoto T, Ikeda S, Yoshikawa H, Shiwa Y, Ishikawa S, Linak MC, Hirai A, Takahashi H, Altaf-Ul-Amin M, Ogasawara N, Kanaya S: **Sequence-specific error profile of Illumina sequencers.** *Nucleic Acids Research* 2011, **39:**e90.

118. Burkhardt S, Karkkainen J: **Better filtering with gapped q-grams.** *Fundamenta Informaticae* 2003, **56:**51-70.

119. Jokinen P, Ukkonen E: **Two algorithms for approxmate string matching in static texts.** In *Mathematical Foundations of Computer Science 1991*. Springer; 1991: 240-248

120. Weese D, Emde AK, Rausch T, Doring A, Reinert K: **RazerS--fast read mapping with sensitivity control.** *Genome Res* 2009, **19:**1646-1654.

121. Weese D, Holtgrewe M, Reinert K: **RazerS 3: faster, fully sensitive read mapping.** *Bioinformatics* 2012, **28:**2592-2599.

122. Siragusa E, Weese D, Reinert K: **Fast and accurate read mapping with approximate seeds and multiple backtracking.** *Nucleic Acids Research* 2013, **41:**e78.

123. Manber U, Myers G: **Suffix Arrays: A New Method for On-Line String Searches.** *SIAM Journal on Computing* 1993, **22:**935-948.

124. Weiner P: **Linear pattern matching algorithms.** 1973:1-11.

125. Delcher AL, Kasif S, Fleischmann RD, Peterson J, White O, Salzberg SL: **Alignment of whole genomes.** *Nucleic Acids Research* 1999, **27:**2369-2376.

126. Meek C, Patel JM, Kasetty S: **OASIS: an online and accurate technique for local-alignment searches on biological sequences.** In *Proceedings of the 29th*

international conference on Very large data bases - Volume 29. pp. 910-921. Berlin, Germany: VLDB Endowment; 2003:910-921.

127. Farach M: **Optimal suffix tree construction with large alphabets.** *38th Annual Symposium on Foundations of Computer Science, Proceedings* 1997**:**137-143.

128. Abouelhoda MI, Kurtz S, Ohlebusch E: **Replacing suffix trees with enhanced suffix arrays.** *Journal of Discrete Algorithms* 2004, **2:**53-86.

129. Hoffmann S, Otto C, Kurtz S, Sharma CM, Khaitovich P, Vogel J, Stadler PF, Hackermuller J: **Fast Mapping of Short Sequences with Mismatches, Insertions and Deletions Using Index Structures.** *PLoS Comput Biol* 2009, **5**.

130. Marco-Sola S, Sammeth M, Guigo R, Ribeca P: **The GEM mapper: fast, accurate and versatile alignment by filtration.** *Nature Methods* 2012, **9:**1185-1188.

131. Lam TW, Sung WK, Tam SL, Wong CK, Yiu SM: **Compressed indexing and local alignment of DNA.** *Bioinformatics* 2008, **24:**791-797.

132. Luebke D, Harris M, Govindaraju N, Lefohn A, Houston M, Owens J, Segal M, Papakipos M, Buck I: **GPGPU: general-purpose computation on graphics hardware.** 2006**:**208.

133. Liu CM, Wong T, Wu E, Luo R, Yiu SM, Li Y, Wang B, Yu C, Chu X, Zhao K, Li R, Lam TW: **SOAP3: ultra-fast GPU-based parallel alignment tool for short reads.** *Bioinformatics* 2012, **28:**878-879.

134. Homer N, Merriman B, Nelson SF: **BFAST: an alignment tool for large scale genome resequencing.** *PLoS One* 2009, **4:**e7767.

135. Chaisson MJ, Tesler G: **Mapping single molecule sequencing reads using basic local alignment with successive refinement (BLASR): application and theory.** *Bmc Bioinformatics* 2012, **13:**238.

136. Schatz MC: **CloudBurst: highly sensitive read mapping with MapReduce.** *Bioinformatics* 2009, **25:**1363-1369.

137. Liu Y, Schmidt B: **CUSHAW2-GPU: empowering faster gapped short-read alignment using GPU computing.** *IEEE Design & Test* 2013**:**1-1.

138. Clement NL, Snell Q, Clement MJ, Hollenhorst PC, Purwar J, Graves BJ, Cairns BR, Johnson WE: **The GNUMAP algorithm: unbiased probabilistic mapping of oligonucleotides from next-generation sequencing.** *Bioinformatics* 2010, **26:**38-45.

139. Ahmadi A, Behm A, Honnalli N, Li C, Weng L, Xie X: **Hobbes: optimized gram-based methods for efficient read alignment.** *Nucleic Acids Research* 2012, **40:**e41.

140. Eaves HL, Gao Y: **MOM: maximum oligonucleotide mapping.** *Bioinformatics* 2009, **25:**969-970.

141. Lee W-P, Stromberg M, Ward A, Stewart C, Garrison E, Marth GT: **MOSAIK: A hash-based algorithm for accurate next-generation sequencing read mapping.** 2013.

142. Alkan C, Kidd JM, Marques-Bonet T, Aksay G, Antonacci F, Hormozdiari F, Kitzman JO, Baker C, Malig M, Mutlu O, Sahinalp SC, Gibbs RA, Eichler EE: **Personalized copy number and segmental duplication maps using next-generation sequencing.** *Nat Genet* 2009, **41:**1061-1067.

143. Hach F, Hormozdiari F, Alkan C, Birol I, Eichler EE, Sahinalp SC: **mrsFAST: a cache-oblivious algorithm for short-read mapping.** *Nature Methods* 2010, **7:**576-577.

144. Hormozdiari F, Hach F, Sahinalp SC, Eichler EE, Alkan C: **Sensitive and fast mapping of di-base encoded reads.** *Bioinformatics* 2011, **27:**1915-1921.

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

146. Kim YJ, Teletia N, Ruotti V, Maher CA, Chinnaiyan AM, Stewart R, Thomson JA, Patel JM: **ProbeMatch: rapid alignment of oligonucleotides to genome allowing both gaps and mismatches.** *Bioinformatics* 2009, **25:**1424-1425.

147. Frousios K, Iliopoulos CS, Mouchard L, Pissis SP, Tischler G: **REAL: an efficient REad ALigner for next generation sequencing reads.** 2010**:**154.

148. Jiang H, Wong WH: **SeqMap: mapping massive amount of oligonucleotides to the genome.** *Bioinformatics* 2008, **24:**2395-2396.

149. Ning Z, Cox AJ, Mullikin JC: **SSAHA: a fast search method for large DNA databases.** *Genome Res* 2001, **11:**1725-1729.

150. Liao Y, Smyth GK, Shi W: **The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote.** *Nucleic Acids Research* 2013, **41:**e108.

151. Sharp PA: **The discovery of split genes and RNA splicing.** *Trends Biochem Sci* 2005, **30:**279-281.

152. Breitbart RE, Andreadis A, Nadal-Ginard B: **Alternative splicing: a ubiquitous mechanism for the generation of multiple protein isoforms from single genes.** *Annu Rev Biochem* 1987, **56:**467-495.

153. Goedert M, Spillantini MG, Jakes R, Rutherford D, Crowther RA: **Multiple isoforms of human microtubule-associated protein tau: sequences and localization in neurofibrillary tangles of Alzheimer's disease.** *Neuron* 1989, **3:**519-526.

154. Licatalosi DD, Darnell RB: **Splicing regulation in neurologic disease.** *Neuron* 2006, **52:**93-101.

155. Kapranov P, Cawley SE, Drenkow J, Bekiranov S, Strausberg RL, Fodor SP, Gingeras TR: **Large-scale transcriptional activity in chromosomes 21 and 22.** *Science* 2002, **296:**916-919.

156. Pan Q, Shai O, Misquitta C, Zhang W, Saltzman AL, Mohammad N, Babak T, Siu H, Hughes TR, Morris QD, Frey BJ, Blencowe BJ: **Revealing global regulatory features of mammalian alternative splicing using a quantitative microarray platform.** *Mol Cell* 2004, **16:**929-941.

157. Kwan T, Benovoy D, Dias C, Gurd S, Provencher C, Beaulieu P, Hudson TJ, Sladek R, Majewski J: **Genome-wide analysis of transcript isoform variation in humans.** *Nat Genet* 2008, **40:**225-231.

158. Slater GS, Birney E: **Automated generation of heuristics for biological sequence comparison.** *Bmc Bioinformatics* 2005, **6:**31.

159. Vanin EF: **Processed pseudogenes: characteristics and evolution.** *Annu Rev Genet* 1985, **19:**253-272.

160. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B: **Mapping and quantifying mammalian transcriptomes by RNA-Seq.** *Nature Methods* 2008, **5:**621-628.

161. Wang ET, Sandberg R, Luo S, Khrebtukova I, Zhang L, Mayr C, Kingsmore SF, Schroth GP, Burge CB: **Alternative isoform regulation in human tissue transcriptomes.** *Nature* 2008, **456:**470-476.

162. Nagalakshmi U, Waern K, Snyder M: **RNA-Seq: a method for comprehensive transcriptome analysis.** *Curr Protoc Mol Biol* 2010, **Chapter 4:**Unit 4 11 11-13.

163. Cloonan N, Xu Q, Faulkner GJ, Taylor DF, Tang DT, Kolle G, Grimmond SM: **RNA-MATE: a recursive mapping strategy for high-throughput RNA-sequencing data.** *Bioinformatics* 2009, **25:**2615-2616.

164. Wood DL, Xu Q, Pearson JV, Cloonan N, Grimmond SM: **X-MATE: a flexible system for mapping short read data.** *Bioinformatics* 2011, **27:**580-581.

165. Pruitt KD, Tatusova T, Maglott DR: **NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins.** *Nucleic Acids Research* 2005, **33:**D501-504.

166. Pruitt KD, Tatusova T, Brown GR, Maglott DR: **NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy.** *Nucleic Acids Research* 2012, **40:**D130-135.

167. Grant GR, Farkas MH, Pizarro AD, Lahens NF, Schug J, Brunk BP, Stoeckert CJ, Hogenesch JB, Pierce EA: **Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM).** *Bioinformatics* 2011, **27:**2518-2528.

168. Denoeud F, Aury JM, Da Silva C, Noel B, Rogier O, Delledonne M, Morgante M, Valle G, Wincker P, Scarpelli C, Jaillon O, Artiguenave F: **Annotating genomes with massive-scale RNA sequencing.** *Genome Biol* 2008, **9:**R175.

169. Huang S, Zhang J, Li R, Zhang W, He Z, Lam TW, Peng Z, Yiu SM: **SOAPsplice: Genome-Wide ab initio Detection of Splice Junctions from RNA-Seq Data.** *Front Genet* 2011, **2:**46.

170. Wu TD, Nacu S: **Fast and SNP-tolerant detection of complex variants and splicing in short reads.** *Bioinformatics* 2010, **26:**873-881.

171. Bryant DW, Jr., Shen R, Priest HD, Wong WK, Mockler TC: **Supersplat--spliced RNA-seq alignment.** *Bioinformatics* 2010, **26:**1500-1505.

172. Philippe N, Salson M, Commes T, Rivals E: **CRAC: an integrated approach to the analysis of RNA-seq reads.** *Genome Biol* 2013, **14:**R30.

173. Cortes C, Vapnik V: **Support-vector networks.** *Machine Learning* 1995, **20:**273-297.

174. Jean G, Kahles A, Sreedharan VT, De Bona F, Ratsch G: **RNA-Seq read alignments with PALMapper.** *Curr Protoc Bioinformatics* 2010, **Chapter 11:**Unit 11 16.

175. Schneeberger K, Hagmann J, Ossowski S, Warthmann N, Gesing S, Kohlbacher O, Weigel D: **Simultaneous alignment of short reads against multiple genomes.** *Genome Biol* 2009, **10:**R98.

176. Dimon MT, Sorber K, DeRisi JL: **HMMSplicer: a tool for efficient and sensitive discovery of known and novel splice junctions in RNA-Seq data.** *PLoS One* 2010, **5:**e13875.

177. Burset M, Seledtsov IA, Solovyev VV: **Analysis of canonical and non-canonical splice sites in mammalian genomes.** *Nucleic Acids Research* 2000, **28:**4364-4375.

178. Iwasaki R, Kiuchi H, Ihara M, Mori T, Kawakami M, Ueda H: **Trans-splicing as a novel method to rapidly produce antibody fusion proteins.** *Biochem Biophys Res Commun* 2009, **384:**316-321.

179. Lou SK, Ni B, Lo LY, Tsui SK, Chan TF, Leung KS: **ABMapper: a suffix array-based tool for multi-location searching and splice-junction mapping.** *Bioinformatics* 2011, **27:**421-422.

180. Yeo G, Burge CB: **Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals.** *J Comput Biol* 2004, **11:**377-394.

181. Bao H, Xiong Y, Guo H, Zhou R, Lu X, Yang Z, Zhong Y, Shi S: **MapNext: a software tool for spliced and unspliced alignments and SNP detection of short sequence reads.** *Bmc Genomics* 2009, **10 Suppl 3:**S13.

182. Hu J, Ge H, Newman M, Liu K: **OSA: a fast and accurate alignment tool for RNA-Seq.** *Bioinformatics* 2012, **28:**1933-1934.

183. Tang S, Riva A: **PASTA: splice junction identification from RNA-sequencing data.** *Bmc Bioinformatics* 2013, **14:**116.

184. Chen LY, Wei KC, Huang AC, Wang K, Huang CY, Yi D, Tang CY, Galas DJ, Hood LE: **RNASEQR--a streamlined and accurate RNA-seq sequence analysis program.** *Nucleic Acids Research* 2012, **40:**e42.

185. Wang L, Wang X, Liang Y, Zhang X: **Observations on novel splice junctions from RNA sequencing data.** *Biochem Biophys Res Commun* 2011, **409:**299-303.

186. Ameur A, Wetterbom A, Feuk L, Gyllensten U: **Global and unbiased detection of splice junctions from RNA-seq data.** *Genome Biol* 2010, **11:**R34.

187. Li Y, Li-Byarlay H, Burns P, Borodovsky M, Robinson GE, Ma J: **TrueSight: a new algorithm for splice junction detection using RNA-seq.** *Nucleic Acids Research* 2013, **41:**e51.

188. Law JA, Jacobsen SE: **Establishing, maintaining and modifying DNA methylation patterns in plants and animals.** *Nat Rev Genet* 2010, **11:**204-220.

189. Keshet I, Lieman-Hurwitz J, Cedar H: **DNA methylation affects the formation of active chromatin.** *Cell* 1986, **44:**535-543.

190. Reik W, Dean W, Walter J: **Epigenetic reprogramming in mammalian development.** *Science* 2001, **293:**1089-1093.

191. Li E, Beard C, Jaenisch R: **Role for DNA methylation in genomic imprinting.** *Nature* 1993, **366:**362-365.

192. Heard E, Clerc P, Avner P: **X-chromosome inactivation in mammals.** *Annu Rev Genet* 1997, **31:**571-610.

193. Walsh CP, Chaillet JR, Bestor TH: **Transcription of IAP endogenous retroviruses is constrained by cytosine methylation.** *Nat Genet* 1998, **20:**116-117.

194. Gopalakrishnan S, Van Emburgh BO, Robertson KD: **DNA methylation in development and human disease.** *Mutat Res* 2008, **647:**30-38.

195. Hultén MA, Papageorgiou EA, Ragione FD, D'Esposito M, Carter N, Patsalis PC: **Non-invasive prenatal diagnosis: An epigenetic approach to the detection of common fetal chromosome disorders by analysis of maternal blood samples** In *Circulating Nucleic Acids in Plasma and Serum*. Edited by Gahan PB; 2011: 133-142

196. Lister R, O'Malley RC, Tonti-Filippini J, Gregory BD, Berry CC, Millar AH, Ecker JR: **Highly integrated single-base resolution maps of the epigenome in Arabidopsis.** *Cell* 2008, **133:**523-536.

197. Cokus SJ, Feng S, Zhang X, Chen Z, Merriman B, Haudenschild CD, Pradhan S, Nelson SF, Pellegrini M, Jacobsen SE: **Shotgun bisulphite sequencing of the Arabidopsis genome reveals DNA methylation patterning.** *Nature* 2008, **452:**215-219.

198. Chung CAB, Boyd VL, McKernan KJ, Fu Y, Monighetti C, Peckham HE, Barker M: **Whole methylome analysis by ultra-deep sequencing using two-base encoding.** *PLoS ONE* 2010, **5:**e9320.

199. Hansen KD, Timp W, Bravo HC, Sabunciyan S, Langmead B, McDonald OG, Wen B, Wu H, Liu Y, Diep D, Briem E, Zhang K, Irizarry RA, Feinberg AP: **Increased methylation variation in epigenetic domains across cancer types.** *Nat Genet* 2011, **43:**768-775.

200. Frommer M, McDonald LE, Millar DS, Collis CM, Watt F, Grigg GW, Molloy PL, Paul CL: **A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands.** *Proc Natl Acad Sci U S A* 1992, **89:**1827-1831.

201. Pedersen B, Hsieh TF, Ibarra C, Fischer RL: **MethylCoder: software pipeline for bisulfite-treated sequences.** *Bioinformatics* 2011, **27:**2435-2436.
202. Shendure J, Ji H: **Next-generation DNA sequencing.** *Nat Biotechnol* 2008, **26:**1135-1145.
203. Homer N, Merriman B, Nelson SF: **Local alignment of two-base encoded DNA sequence.** *BMC Bioinformatics* 2009, **10:**175.
204. Krueger F, Kreck B, Franke A, Andrews SR: **DNA methylome analysis using short bisulfite sequencing data.** *Nature Methods* 2012, **9:**145-151.
205. Ondov BD, Varadarajan A, Passalacqua KD, Bergman NH: **Efficient mapping of Applied Biosystems SOLiD sequence data to a reference genome for functional genomic applications.** *Bioinformatics* 2008, **24:**2776-2777.
206. Karp RM, Rabin MO: **Efficient randomized pattern-matching algorithms.** *IBM Journal of Research and Development* 1987, **31:**249–260.
207. Smith AD, Xuan Z, Zhang MQ: **Using quality scores and longer reads improves accuracy of Solexa read mapping.** *BMC Bioinformatics* 2008, **9:**128.
208. **Sherman** [http://www.bioinformatics.bbsrc.ac.uk/projects/sherman/]
209. Laurent L, Wong E, Li G, Huynh T, Tsirigos A, Ong CT, Low HM, Kin Sung KW, Rigoutsos I, Loring J, Wei CL: **Dynamic changes in the human methylome during differentiation.** *Genome Res* 2010, **20:**320-331.
210. Tennakoon C, Purbojati RW, Sung WK: **BatMis: A fast algorithm for k-mismatch mapping.** *Bioinformatics* 2012.
211. Bird A, Taggart M, Frommer M, Miller OJ, Macleod D: **A fraction of the mouse genome that is derived from islands of nonmethylated, CpG-rich DNA.** *Cell* 1985, **40:**91-99.
212. Bhangale TR, Rieder MJ, Livingston RJ, Nickerson DA: **Comprehensive identification and characterization of diallelic insertion-deletion polymorphisms in 330 human candidate genes.** *Hum Mol Genet* 2005, **14:**59-69.
213. Yang H, Zhong Y, Peng C, Chen JQ, Tian D: **Important role of indels in somatic mutations of human cancer genes.** *BMC Med Genet* 2010, **11:**128.
214. Li H: **Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM.** *arXiv preprint arXiv:13033997* 2013.
215. JQ Lim, Chandana T, PY Guan, WK Sung: **BatAlign: an incremental method for accurate alignment of sequencing reads.** *Nucleic Acids Research* 2015.
216. Degner JF, Marioni JC, Pai AA, Pickrell JK, Nkadori E, Gilad Y, Pritchard JK: **Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data.** *Bioinformatics* 2009, **25:**3207-3212.
217. Huang W, Li L, Myers JR, Marth GT: **ART: a next-generation sequencing read simulator.** *Bioinformatics* 2012, **28:**593-594.
218. Wang J, Mullighan CG, Easton J, Roberts S, Heatley SL, Ma J, Rusch MC, Chen K, Harris CC, Ding L, Holmfeldt L, Payne-Turner D, Fan X, Wei L, Zhao D, Obenauer JC, Naeve C, Mardis ER, Wilson RK, Downing JR, Zhang J: **CREST maps somatic structural variation in cancer genomes with base-pair resolution.** *Nature Methods* 2011, **8:**652-654.
219. Zhang ZDD, Du J, Lam H, Abyzov A, Urban AE, Snyder M, Gerstein M: **Identification of genomic indels and structural variations using split reads.** *Bmc Genomics* 2011, **12**.
220. Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q, Locke DP, Shi X, Fulton RS, Ley TJ, Wilson RK, Ding L, Mardis ER: **BreakDancer: an algorithm for high-resolution mapping of genomic structural variation.** *Nature Methods* 2009, **6:**677-681.

221. Fernandez-Banet J, Lee NP, Chan KT, Gao H, Liu X, Sung WK, Tan W, Fan ST, Poon RT, Li S, Ching K, Rejto PA, Mao M, Kan Z: **Decoding complex patterns of genomic rearrangement in hepatocellular carcinoma.** *Genomics* 2014, **103:**189-203.

222. Bartenhagen C, Dugas M: **RSVSim: an R/Bioconductor package for the simulation of structural variations.** *Bioinformatics* 2013, **29:**1679-1681.

223. Durbin RM, Altshuler D, Abecasis GR, Bentley DR, Chakravarti A, Clark AG, Collins FS: **A map of human genome variation from population-scale sequencing.** *Nature* 2010, **467:**1061-1073.

224. Mills RE, Luttig CT, Larkins CE, Beauchamp A, Tsui C, Pittard WS, Devine SE: **An initial map of insertion and deletion (INDEL) variation in the human genome.** *Genome Res* 2006, **16:**1182-1190.

225. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JY, Zhang J: **Bioconductor: open software development for computational biology and bioinformatics.** *Genome Biol* 2004, **5:**R80.

226. Karro JE, Yan Y, Zheng D, Zhang Z, Carriero N, Cayting P, Harrrison P, Gerstein M: **Pseudogene.org: a comprehensive database and comparison platform for pseudogene annotation.** *Nucleic Acids Research* 2007, **35:**D55-60.

227. Chen R, Mias GI, Li-Pook-Than J, Jiang L, Lam HY, Miriami E, Karczewski KJ, Hariharan M, Dewey FE, Cheng Y, Clark MJ, Im H, Habegger L, Balasubramanian S, O'Huallachain M, Dudley JT, Hillenmeyer S, Haraksingh R, Sharon D, Euskirchen G, Lacroute P, Bettinger K, Boyle AP, Kasowski M, Grubert F, Seki S, Garcia M, Whirl-Carrillo M, Gallardo M, Blasco MA, et al: **Personal omics profiling reveals dynamic molecular and medical phenotypes.** *Cell* 2012, **148:**1293-1307.

228. Darling AC, Mau B, Blattner FR, Perna NT: **Mauve: multiple alignment of conserved genomic sequence with rearrangements.** *Genome Res* 2004, **14:**1394-1403.

229. Gudlaugsdottir S, Boswell DR, Wood GR, Ma J: **Exon size distribution and the origin of introns.** *Genetica* 2007, **131:**299-306.

230. Saito Y, Tsuji J, Mituyama T: **Bisulfighter: accurate detection of methylated cytosines and differentially methylated regions.** *Nucleic Acids Research* 2014, **42:**e45.

231. Chargaff E, Zamenhof S, Green C: **Composition of human desoxypentose nucleic acid.** *Nature* 1950, **165:**756-757.

232. Meselson M, Stahl FW: **The replication of DNA in Escherichia coli.** *Proceedings of the National Academy of Sciences* 1958, **44:**671-682.

233. Holley RW, Apgar J, Everett GA, Madison JT, Marquisee M, Merrill SH, Penswick JR, Zamir A: **Structure of a Ribonucleic Acid.** *Science* 1965, **147:**1462-1465.

234. Kowalczyk J, Domal-Kwiatkowska D, Mazurek U, Zembala M, Michalski B, Zembala M: **Post-transcriptional modifications of VEGF-A mRNA in non-ischemic dilated cardiomyopathy.** *Cellular & Molecular Biology Letters* 2007, **12:**331-347.

235. Darnell JE, Jr.: **Implications of RNA-RNA splicing in evolution of eukaryotic cells.** *Science* 1978, **202:**1257-1260.

236. Early P: **Two mRNAs can be produced from a single immunoglobulin μ gene by alternative RNA processing pathways.** *Cell* 1980, **20:**313-319.

237. Pan Q, Shai O, Lee LJ, Frey BJ, Blencowe BJ: **Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing.** *Nat Genet* 2008, **40:**1413-1415.

Appendix A

A.1.1 DNA-DNA Replication

DNA comprises of nucleotides and each of them contains a deoxyribose sugar, a phosphate and a nucleobase. It is usually double-stranded and both strands are bonded together to form a double-helix structure. The deoxyribose sugar and phosphate will form the backbone of the double-helix structure and the nucleobase (Adenine, Cytosine, Guanine and Thymine; ACGT) will be forming hydrogen bonds with another nucleobase on the reverse-complementary strand of the DNA. The base pair makeup of the DNA was also hinted by Chargaff's 1950 experiment and provides a general but not exclusive rule that adenine and cytosine pairs up with thymine and guanine respectively on opposing strands of the DNA [231].

DNA replication is the process whereby a new copy of the DNA molecule is replicated from one original template DNA molecule. This is possible as DNA is composed of two strands and each strand of the original DNA molecule serves as a template for the replication of the new reverse-complementary strand. This results in two copies of double-stranded DNA molecules with each of them consisting of an 'old' template strand and a 'new' replicated strand; this is why DNA is semi-conservatively replicated and is demonstrated to be so in 1958 by Meselson-Stahl experiment [232]. Figure A.1 shows three postulated methods of replication before Meselson-Stahl experiment.

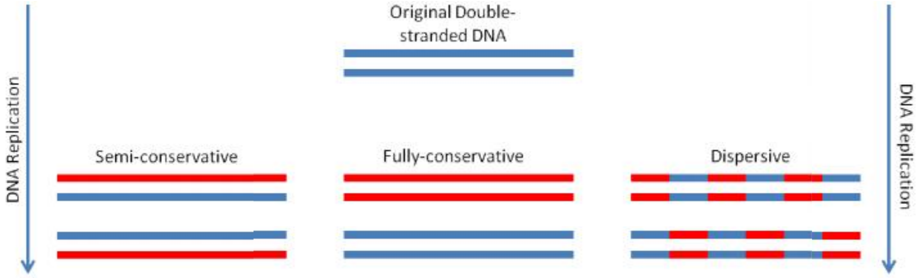


Figure A.1. Three postulated methods for DNA replication prior to Meselson-Stahl experiment.

As DNA replicates prior to mitosis, it must involve initiation of replication, elongation of DNA fragments and termination of synthesis. For a cell to divide, it must replicates its DNA first and this process can initialize at various sites known as replication origins. Initiator proteins will target A-T rich regions of the DNA and recruit other proteins, unzips the double-stranded DNA and prepares it for replication. As the new DNA is being synthesized and elongated on the old template DNA, the helicases keep breaking the hydrogen bonds between the two DNA strands to unwind more regions of the DNA for elongation.

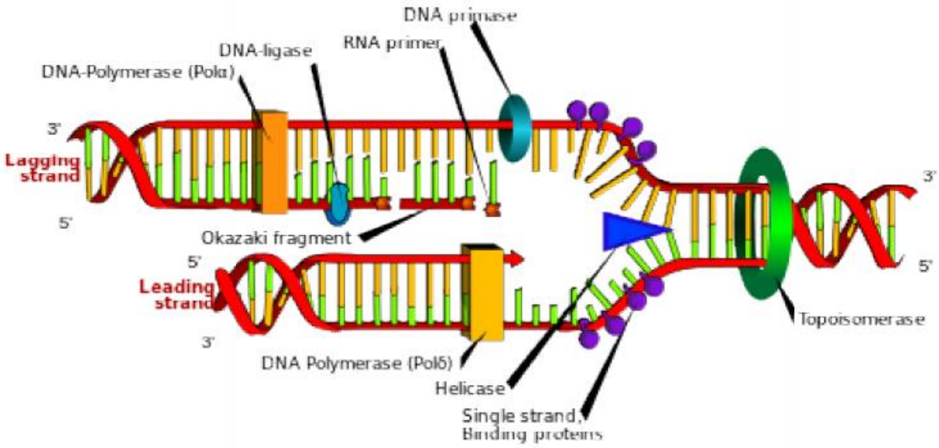


Figure A.2. Schematic diagram of DNA replication at a replication fork.

As DNA is always synthesized from the 5’ to 3’ direction, there will be one strand of the DNA that will be in the ‘wrong’ direction and this is called the lagging strand in DNA replication; the other strand will be the leading strand. The DNA polymerase will start to add complementary bases to the template strand after a small RNA fragment attaches itself to the site of replication origin to prime the elongation process. With respect to the leading strand, the DNA polymerase will move in the same direction of the helicase. However, for the lagging strand, the DNA polymerase can only add bases away from the direction of the helicase and results in replicating the DNA in disjoint but adjacent fragments called Okazaki fragments. Figure A.2 depicts the process of DNA replication at one instance of the DNA replication fork. Since there are multiple points of replication origins, termination of elongation happens when a replication forks meet and this can occur at many points in a single chromosome.

A.1.2 DNA-RNA Transcription

RNA comprises of nucleotides and each of them contains a ribose sugar, a phosphate and a nucleobase. It is usually single-stranded. However, RNA can form intra-strand double helix structure as in the case of the double-stranded DNA by complementary base-pairing with hydrogen bonds too; as in the case of tRNAs. The ribose sugar and phosphate will form the backbone of the structure for RNA and the nucleobase (Adenine, Cytosine, Guanine and Uracil; ACGU). Three main types of RNA are transcribed from a region of the DNA as a template and they are messenger-RNA (mRNA), transfer-RNA (tRNA) and ribosomal RNA (rRNA) [233]. mRNA is a near-duplicate of a region of the template DNA that will code for a protein sequence. tRNA is a short sequence of ~80 nucleotides that transfers amino acid to the site of protein synthesis. rRNA is responsible to link the amino acids from the tRNA to grow the polypeptide chain to form a protein.

The first step in achieving molecular function is to transcribe a gene region of the DNA into mRNA in a process called transcription. The mRNA will act as a blueprint for a protein to be translated from it. In eukaryotes, the process starts by having the RNA polymerase and other transcription factor(s) to bind to a core promoter sequence in the DNA, which is usually within a hundred, bases upstream from the transcription start site (TSS) of a gene. In prokaryotes, protein factors bind to the RNA polymerase, which affects the binding of the polymerase to the DNA. The RNA polymerase will next start to move along the promoter region and towards the TSS. Once the RNA polymerase enters the gene region, it will use base pairing complementarily with the DNA template (non-coding strand) to create an RNA copy. Different transcription levels of genes are usually resulted from multiple rounds of transcription or multiple RNA polymerases on a single DNA template. Elongation of the RNA terminates when the newly synthesized RNA segment contains a GC rich and subsequent Us rich sequence or the 'Rho' protein destabilize the interaction between the template DNA and the mRNA. These two mechanisms cause the template DNA and RNA polymerase to disengage from one another and the synthesis of any new RNA segments to cease.

A.1.2.1 Genes and Splicing

A gene is a biological unit of hereditary material. It can also refer to subsequences of DNA and it provides the blueprints for the RNA polymerase to synthesize proteins from it. In eukaryotic cells, the RNA that is transcribed from the DNA will undergo more post-transcription modifications [234]. At the 5' end of the pre-mRNA, a single G will have its 5' end attached to it, whereas at the 3' end, a poly-A tail will be added. This capping on both ends of the untranslated regions (UTRs) of the pre-mRNA fragment will result in 3' endings and protect the fragment from being cleaved at the 5' end by exonucleases.

Figure A.3 shows the differences in the markup of genomic features between pre-mRNA and mRNA.

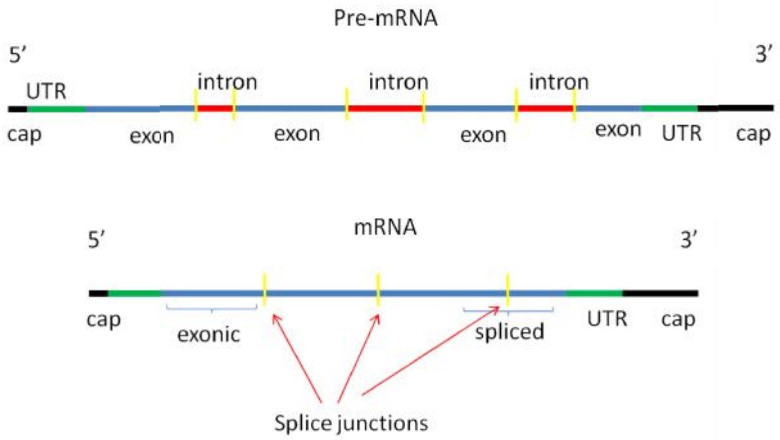


Figure A.3. Illustration of introns and exons in pre-mRNA and the maturation of mRNA by splicing.

A pre-mRNA fragment contains adjacent sequences of nucleotides that will either be translated to protein or not; namely, exons and introns respectively [235]. In eukaryotic cells, cleaving the introns away, leaving the exons behind, matures the pre-mRNA fragment. This event is known as splicing and the genomic locations where introns are being cleaved at are called splice sites. From the literature, we can observe that these splice sites tends to be conserved with canonical signals (GT-AG, donor-acceptor) at rate of >98% on splicing events in humans [177].

Splice sites can sometimes reside completely in exonic or intronic regions. In other words, splicing can sometimes happen or not happen at a splice site and this is known as alternate splicing [236]. This gives the possibility of a single gene to code for several proteins, which makes it more efficient as a single gene region may have more than one functional product. In fact, the human DNA is so efficient in this sense that ~95% of multi-exons gene regions can express more than one functional product [237].

Currently, SGS technologies produce RNA-seq data from sequencing matured mRNA fragments. As such, the intronic regions are left out from the spliced sequencing read. Before scientists can study the transcription levels of genes, they have to map the RNA-seq reads back to the human DNA reference genome by taking these intronic gaps into account too. The alignment of RNA-seq read proved to be a challenge as seen from the myriad of computational methods developed to solve it. In the following chapter, we will review on the techniques developed for the alignment of RNA-seq reads.

A.1.3 RNA-Protein Translation

Proteins are chains of polypeptide sequences that are made up of some combinations of amino acids. The polypeptide chain folds into a 3-D structure, which will define its cellular functions. Generally, proteins are studied at four levels of granularity. At the finest level, the structure of a protein can be studied by the sequence of amino acids, which represents it. Next, secondary local structures such as the α-helix and β-pleated sheets are formed when amino acids of the same polypeptide are joined together by hydrogen bonds. Thirdly, tertiary structures are folded into configurations due to the attractive/repulsive forces between secondary local structures. Lastly, quaternary structures are formed when two or more proteins come together to form a more complex 3-D structure.

Proteins are synthesized from an mRNA sequence by a ribosome complex through a process called translation. Translation starts with the ribosome binding to the 5' end of the mRNA. The ribosome will then decode the mRNA in consecutive non-overlapping frames of 3 bases called a codon. The start codon for translation is "ATG" and serves as an initiation site for translation. While the ribosome traverses across the mRNA, tRNAs carrying specific amino acids with complementary anti-codon sequences to that of the mRNA will have the amino acids chain together into a polypeptide. The chain will terminate when the ribosome faces a stop codon (UAA, UAG or UGA) and this recruit a release factor protein to disassemble the entire ribosome-mRNA complex. The synthesized chains of polypeptide will then give itself the molecular functions with the structure that it folds itself into or by integrating with other secondary or tertiary structures as mentioned before.