Accurate and Rapid Detection of Circular RNA through Splice-Aware Pseudo-Alignment Scheme

by

Hossein Asghari

B.Sc., University of Tehran, 2016

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2019

Approval

Name:	**Hossein Asghari**
Degree:	**Master of Science (Computing Science)**
Title:	**Accurate and Rapid Detection of Circular RNA through Splice-Aware Pseudo-Alignment Scheme**
Examining Committee:	**Chair:** Ramesh Krishnamurti Professor
	Binay Bhattacharya Senior Supervisor Professor
	Faraz Hach Supervisor Assistant Professor Department of Urologic Sciences University of British Columbia
	S. Cenk Sahinalp Supervisor Adjunct Professor
	Cedric Chauve Internal Examiner Professor Department of Mathematics
Date Defended:	**August 22, 2019**

Abstract

The ubiquitous abundance of circular RNAs (circRNAs) has been revealed by performing high throughput sequencing in a variety of eukaryotes. circRNAs are related to some diseases such as cancer in which they act as oncogenes or tumor-suppressors, and therefore have the potential to be used as biomarkers or therapeutic targets. Accurate and rapid detection of circRNAs from short reads remains computationally challenging. This is due to the fact that identifying chimeric reads, which is essential for finding back-splice junctions, is a complex process. The sensitivity of discovery methods, to a high degree, relies on the underlying mapper that is used for finding chimeric reads. Furthermore, all the available circRNA discovery pipelines are resource intensive.

We introduce CircMiner, a novel stand-alone circRNA detection method that rapidly identifies and filters out linear RNA-Seq reads and detects back-splice junctions. CircMiner employs a rapid pseudo-alignment technique to identify linear reads that originate from transcripts, genes, or the genome. CircMiner further processes the remaining reads to identify the back-splice junctions and detect circRNAs with single-nucleotide resolution. When requested, with extra time overhead, CircMiner also reports the mapping locations of the linear reads. We evaluated the efficacy of CircMiner using simulated datasets generated from known back-splice junctions and showed that CircMiner has superior accuracy and speed compared to the existing circRNA detection tools. Additionally, on two RNase R treated cell line datasets, CircMiner was able to detect more consistent, high confidence circRNAs compared to untreated samples of the same cell line.

Keywords: Computational Biology; Bioinformatics; RNA-seq; Pseudo-Alignment; Quasi-Mapping, Circular RNA

Dedication

To my dearest family

Acknowledgements

The writing of this thesis would not have been possible without the help of many people. I would like to extend my sincere gratitude towards all of them. First and foremost, I am highly indebted to my supervisor, Dr. Faraz Hach for his extensive support, stimulating guidance, and unrestricted patience. My heartfelt thanks to Prof. S. Cenk Sahinalp and Prof. Binay Bhattacharya for their support. I am especially thankful to Prof. Cedric Chauve, my thesis examiner, for his careful reading and comments.

I would like to thank Dr. Yen-Yi Lin, who gave me invaluable advice and feedbacks during my research. A special thank goes to Hossein Sharifi-Noghabi for giving me hope and motivation. It was a great pleasure for me to be a member of the Computational Cancer Genomics Lab at Vancouver Prostate Centre. I like to thank my lab members for their help, especially Ehsan Haghshenas, Tunc Morova, Fatih Karaoglanoglu, Baraa Orabi, and Steven Xu.

Last but not least, I would like to express my deepest gratitude to my parents and my sisters for their unconditional love and support. I especially owe a great debt to my mother and father for always putting us and our future first and giving me moral support throughout all years of my studies. This could not have been accomplished without their endless love and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

High throughput sequencing (HTS) technologies are capable of providing sequence of hundreds of millions DNA or RNA molecules in parallel with high precision [1]. This enabled more aspects of Genetics and Transcriptomics to be revealed. The existence of circular RNAs (circRNAs)–RNA molecules with back-splice junctions from a downstream $3'$ splice site and an upstream $5'$ splice site–in eukaryotic cells was first reported in 1979 [2]. However, they did not garner much attention until 2012, when they were found to be abundantly expressed across the human and mouse genomes with the aid of HTS [3, 4]. It was subsequently determined that circRNA expression extended to essentially all eukaryotes [5]. Extensive efforts have since been made to understand the characteristics, function, and possible applications of circRNAs.

CircRNAs are formed when a downstream donor $3'$ splice site is covalently joined to an upstream $5'$ splice site by a $3'$-$5'$ phosphodiester bond. This event, termed back-splicing, appears to involve the same splicing signals and machinery as canonical mRNA splicing [6]. Circularization has been associated with long flanking intronic sequences and increased RNA polymerase II transcription speed [7, 8]. There is evidence suggesting that back-splicing occurs both co-transcriptionally and post-transcriptionally, with the length of a circRNA's flanking intron repeats determining the favored mechanism [9, 8, 10]. Back-splicing is far less efficient than canonical splicing, providing a probable explanation to the low steady-state levels at which most circRNAs are expressed [8, 11]. Nonetheless, circRNAs are more stable than their linear counterparts and accumulate within cells to allow widespread detection [8].

The pervasive expression of circRNAs across the eukaryotes domain of life indicates that they were either conserved over a billion years of evolution or evolved independently in multiple kingdoms, both of which suggest a functional role [5]. As circRNAs are frequently transcribed from the same genes as mRNA, their formation competes with the synthesis of mature mRNA [12]. CircRNAs may also be significant in cancer as some expressed circRNAs have oncogenic or tumor-suppressor functions [13]. Fusion circRNAs, transcribed from exons originating from distinct genes, have been found to promote oncogenesis and confer

resistance to therapy [14]. In patients with prostate cancer, the level of circRNA expression may predict tumor progression and prognosis [15].

Despite their abundant expression, circRNAs were largely overlooked prior to 2012 due to the selection for poly(A) tails in the preparation of most RNA sequencing (RNA-Seq) libraries, leading to circRNA depletion [16]. Today, there is a much greater emphasis on biochemical protocols such as ribosomal RNA (rRNA) depletion and poly(A) depletion in non-coding RNA studies to preserve circRNAs. The discovery of widespread circRNAs also revealed shortcomings in previously published splice detection algorithms, given some circRNAs would inevitably survive poly(A) selection [16]. This spurred the development of novel computational detection tools aimed specifically at back-splice junction detection including CIRI [17] / CIRI2 [18], CIRCexplorer [19] / CIRCexplorer2 [20], KNIFE [21], circRNA_finder [22], MapSplice [23], segemehl [24], and CircMarker [25]. However, comparisons show their output diverge dramatically with no tool having a clear advantage [26, 27]. These tools can be classified into two categories based on the way they determine chimeric reads: (i) *de-novo* discovery methods (e.g., segemehl and CircMarker) where they detect the circRNAs from the sequencing reads; and (ii) mapping based tools (e.g., CIRCexplorer) where they are built on top of general-purpose RNA-Seq mappers such as STAR. In both cases they are time consuming and usually require significant computational resources.

Moreover, detection of chimeric reads (e.g., back-splice junction reads) from RNA-seq is challenging and has a dramatic effect on the sensitivity of the circRNA detection methods. A general-purpose mapper such as STAR is mainly designed to map linear mRNA reads back to transcriptome because most of the analysis involving RNA-Seq relies on expression estimation. In recent years, a new generation of mappers such as Kallisto [28] and RapMap [29] have been developed that apply pseudo-alignment and quasi-mapping techniques on the transcriptome, respectively. They have a higher speed than STAR as they only focus on transcriptome compatible reads. However, these tools to this date are not splice-aware, i.e., they only locate reads on the transcriptome and are unable to identify reads on the genome.

1.1 Contribution

Here, we introduce CircMiner, a novel method employing splice-aware pseudo-alignment using hash tables to quickly filter linear reads and report chimeric reads potentially related to aberrations such as back-splicing or gene fusion. CircMiner's detection module examines the reported chimeric reads and identifies the reads that can generate back-splice junctions. Finally, it identifies the high confidence back-splice junction and reports them. CircMiner was evaluated on both simulated datasets and real datasets. On simulated datasets, CircMiner achieves the best time, recall and F1 score while having a similar precision to the other discovery tools. We further evaluated the performance of these tools on two RNase R treated cell-line samples. RNase R is used to digest linear mRNAs; thus, the treated

sample will have higher abundance of circRNAs. In this comparison, CircMiner was able to detect more common circRNAs between treated and untreated samples that have exact same breakpoints. This suggests that CircMiner has a lower false positive rate than any others. It is worth mentioning that CircMiner can report full alignments at an additional processing time per user request. Besides, CircMiner has the potential to be used in the detection of other aberrations, such as gene fusion, by adding another detection module designed to process the potential fusion reads that CircMiner reports.

CircMiner is implemented in C++ and is freely available online at our GitHub repository at `https://github.com/vpc-ccg/circminer`.

1.2 Organization of the Thesis

The rest of the thesis is organized as follows. In Chapter 2 we describe splice-aware sequence mapping and circular RNA detection problems as well as pseudo-alignment technique. In chapter 3 we introduce CircMiner, our developed method for splice-aware pseudo-alignment and circRNA detection. The results and performance evaluation of CircMiner is described in chapter 4. And finally, in chapter 5, we conclude by providing a summary of our contributions, as well as a discussion on possible directions for future work.

Chapter 2

Background and Related Work

This chapter provides preliminaries for next generation sequencing (NGS) and its typical analysis. It starts with an overview of sequencing techniques and DNA and RNA sequencing technologies. It follows by describing typical transcriptome analysis using RNA sequencing data. For this, we explain some of the splice mapping algorithms that are used as the first step of RNA-Seq analysis. Afterwards, we discuss about aberrant detection from RNA-Seq data, more focusing on non-canonical splicing and in particular circular RNA detection. We conclude this chapter by describing pseudo-alignment and quasi-mapping concepts and two computational tools implementing them in order to achieve fast alignment of RNA-Seq reads to transcriptome.

2.1 Next Generation Sequencing (NGS)

Next Generation Sequencing technologies emerged a few decades after Sanger DNA Sequencing [30] which was used to sequence the first Human Genome and several other organisms. Some of the drawbacks of Sanger sequencing are high cost and low throughput, making it less suitable for personal genome sequencing. As a result, Next Generation Sequencing was developed to provide extremely high-throughput sequencing (NGS) from multiple samples in parallel at significantly reduced cost [31]. Next Generation Sequencing technologies have had an enormous impact in the fields of Genomics, Transcriptomics, and Proteomics. Rapid progress in NGS technologies along with the simultaneous development of bioinformatics tools that are required to analyze the generated data have given us the ability to study biological aspects of many different species and diseases. Nowadays, due to the reduced cost, Next Generation Sequencing has many applications such as detection of genomic variations - Single Nucleotide Variation (SNV), Copy Number Variation (CNV), and Structural Variation (SV). For instance, by sequencing cancer genome, the mutational landscapes of different cancer types have been revealed [32]. In addition, by using NGS data, tumor heterogeneity, clonal evolution and the mechanisms underlying drug resistance can be studied [32].

Through the past couple of decades, many commercial NGS platforms evolved some of which are widely used by genomic community to illuminate topics that were previously difficult to study. Here, I will briefly explain some of the most popular NGS technologies that have been adapted during past decade for short read sequencing.

Illumina. Illumina is one of the most prominent technologies in HTS market particularly with HiSeq, MiSeq, and NovaSeq platforms. Illumina's sequencing process involves PCR amplification of adaptor-ligated DNA fragments that were placed on a solid glass surface [33]. During PCR amplification, some clusters containing several identical copies of each original sequence are created. In each cluster, there are approximately one million copies of the original sequence [34]. In order to specify nucleotides, each type of nucleotide is labeled with a specific fluorescent color. Then, the clusters are excited by laser emitting a specific light signal coming from each nucleotide. These signals are captured by a camera and translated into a nucleotide sequence. In each round, one layer of nucleotides is washed so that the nucleotides beneath them become visible to the camera. This cycle continues until all the nucleotides in the clusters are sequenced and then reported in a text file in standard FASTQ format. The overall error rate of Illumina sequencing technology is between 0.2% to 1% mainly dominated by substitution errors [35]. There are several reasons behind these sequencing errors such as difficulties in image processing for base calling and imperfect washing of nucleotides in some rounds. All the sequencing data being used in this work are generated by Illumina sequencing platform.

Ion Torrent. The Ion Torrent semiconductor sequencing technology is another sequencing technology that was commercialized in 2010 [36]. It works by releasing hydrogen ion that change the pH of the solution. An electronic sensor will capture this change which will be converted to a voltage signal and later will be translated to nucleotide sequence. This technology is capable of producing longer reads (200-600 bp) with almost the same accuracy (99%) in less time.

One of the drawbacks of both of these sequencing technology is their low accuracy in reading homopolymer regions and repeats. NGS technologies are being mainly used for DNA sequencing and Whole Transcriptome Shotgun Sequencing (WTSS) – also called RNA-Seq. A general overview of these two types of sequencing is given in the following sections.

2.1.1 DNA Sequencing

Each cell inside an organism keeps all the organism's genetic information as *Deoxyribonucleic acid (DNA)* that is made up of molecules called *nucleotides*. There are four types of nucleotides: Adenine (A), Cytosine (C), Guanine (G), Thymine (T) each of which is represented by its corresponding character by the sequencing machines. DNA sequencing is a process of precisely identifying the content of the genome, or a selected regions on the genome. Based on this definition, they can be categorize as follows: (i) Whole Genome Sequencing (WGS) where the whole genome is covered by the reads generated from the

sequencing machine. The advantage of WGS is that it gives a uniform read coverage over the entire genome and is able to reveal genome-wide variants and rearrangements. Sometimes high coverage of genes is required such as in cancer research where the proportion of tumor DNA in the sample is much less than normal DNA. In this case, it will be expensive and impractical to have high read coverage over the whole genome especially for human genome which is more than 3 billion nucleotides long. Thus another form of DNA sequencing has been introduced. (ii) Targeted Sequencing (TS) is used when a specific region on the genome should be analyzed. For example, when studying a disease with known pathways and causing genes. In this case a special panel is designed containing only the regions on the genome that are going to be captured. Since the covered area is smaller than the whole genome, it gives us the ability to generate high coverage data to study those specific regions of many samples. Targeted sequencing is extremely useful in studying subclonal events in cancer where a small portion of the sample is representing the event. (iii) Whole Exome Sequencing (WES) in which the only sequenced regions are Exome. Exome is defined as the entire set of exons in a genome. WES is a special case of TS which provides coverage for expressed parts of the genome (exons) in order to investigate protein-conding regions. Sometimes it is required to have the whole protein-coding regions being sequenced for a study and in this case WES is the answer. Since a small proportion of genome is covered by exonic regions, WES is the cost-effective alternative to WGS and due to this fact, sequencing with higher coverage than WGS is possible. Some applications of WES are copy number variation (CNV) and mutation calling for many samples [37].

Based on the advantages and limitations of each type of DNA sequencing and the needs of the project, an appropriate type should be chosen.

2.1.2 RNA Sequencing

Ribonucleic acid (RNA) is another type of biological macromolecules. Genetic information that is stored in DNA will be transcribed into messenger RNA (mRNA), which will be translated into proteins. RNA only contains expressed regions of the DNA (exons). The region between two consecutive exons is called intron (intragenic region) which is removed as part of transcription process. In this process: (i) precursor mRNA (pre-mRNA) is synthesized from DNA by RNA polymerase II; and (ii) the exons are extracted from pre-mRNA and concatenated together in a process called splicing. In addition to splicing, a 5' cap is attached to the 5' end of mRNA and a poly(A) tail is attached to the 3' end of mRNA. Mature mRNA is read by ribosomes, which synthesize proteins according to the sequence encoded by mRNA in a process known as translation.

RNA sequencing (RNA-seq), also called whole transcriptome shotgun sequencing (WTSS), is an approach to study transcriptome of a given biological sample using capabilities of next generation sequencing. The RNA-Seq libraries generated prior to sequencing are either based on selecting poly(A)+ mRNAs, or depleting total RNA of highly abundant ribosomal

RNAs (rRNA) [38]. The rRNA depletion protocols have an advantage over poly(A)+mRNA -selected libraries which is allowing simultaneous characterization of coding and non-coding RNAs. On the other hand, a significant amount of RNA-Seq data originated from rRNA-depleted procedures might be mapped to non-coding regions on the genome which is not desired in some studies. There are different types of RNA-Seq libraries in terms of strand-specificity.

2.1.2.1 Strand Specific RNA-Seq

Some RNA-Seq preparation kits do not keep information about which strand was originally transcribed. During the process of library preparation for RNA sequencing, RNA will be converted to double-stranded complementary DNA (cDNA) which is more stable than RNA and more importantly it can be easily PCR amplified. Due to synthesis of randomly primed double-stranded cDNA that is followed by addition of adapters used in next generation sequencing in non-strand specific RNA-Seq libraries, the strand information will be lost. However, knowing the originating strand is useful in many applications such as sense and antisense transcript structure prediction, and more accurate estimation of expression levels of sense and antisense genes [39]. As a result many preparation methods have been developed for strand-specific RNA-Seq each of which add some modifications to the standard RAN-Seq protocol and fall into two categories: (i) First category is based on attaching two distinct adapters to the 5' and 3' ends, marking the ends differently. (ii) Second category uses chemical modification to mark one strand. It can be done on RNA itself or later on synthesized cDNA.

2.1.2.2 RNase R

Ribonuclease R (RNase R) from *E. Coli* plays an important role in many aspects of RNA metabolism. RNase R is primarily a degradative enzyme that acts on different types of RNA such as mRNA and rRNA under certain conditions [40]. It is typically used to digest all linear RNAs but does not digest circular RNA structures and intron *lariats* – byproducts of the pre-mRNA splicing process where the 5' end of intron pairs with the downstream sequence of the same intron and forms a circular structure (see Figure 2.1). The remaining RNA after digestion can be sequenced using the RNA-Seq libraries. This is an useful process when we intend to study circular RNAs and lariats. Although the sequence content will be derived from circular region or intron, it will not represent the linear form of intron or the circular region. Thus, some computational tools have been developed for detection of circular RNAs and the breakpoints.

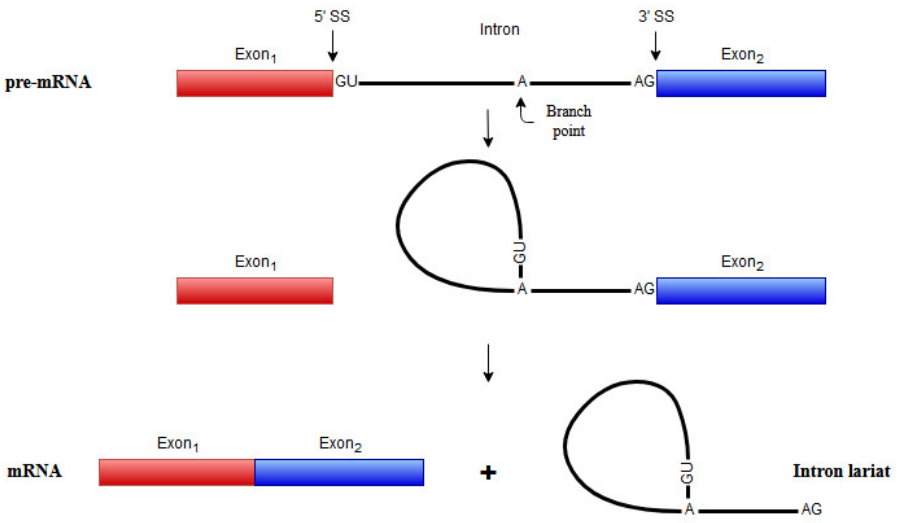


Figure 2.1: A schematic view of the biochemical process involved in RNA splicing, resulting in a spliced mRNA and intron lariat.

2.2 Analysis

2.2.1 Spliced Mappings for RNA-Seq Data

Transcriptome analysis usually starts with mapping the reads back to the reference genome.
Definition *Splice Site*: The specific site on the genome at which splicing takes place during the transformation of precursor messenger RNA (pre-mRNA) into mature messenger RNA (mRNA). In Figure 2.2 the splice sites are denoted as SS.
Definition *Spliced Mapping*: Given a data set of RNA-Seq reads R, a reference genome G, a maximum intron length L, an error threshold e, a scoring function δ, and optionally a database of splice sites (a set of locations on G) or a set of gene annotations M, the goal is to find original location(s) of each read $r \in R$ in reference G, such that the following conditions are met: (i) Total error on a mapping is less than or equal e under δ. (ii) Any splice junction on r, follows splice codes (GT/AG, etc.) and has a distance not greater than L. (iii) If a gene model M is provided, mapping results are compatible with it (see Figure 2.2). In the gene model, for each gene, exon boundaries and splicing site locations are specified per transcript. A mapping for read r is called compatible with a given gene model M if either (i) r falls completely inside an exon; or (ii) all the splice junctions on the mapping of r happen exactly on exon boundaries of a single transcript as shown in Figure 2.2.

Different RNA-seq mappers have different approaches for determining splicing sites. Splice mappers usually start the mapping process by first finding anchors that are either locations of some *k-mers* (sub-strings of size k) on the read or longest common sub-strings between the read and reference genome. Hash based mappers use fixed k-mer location look-up while suffix array or BWT based mappers try to find longest common sub-strings.

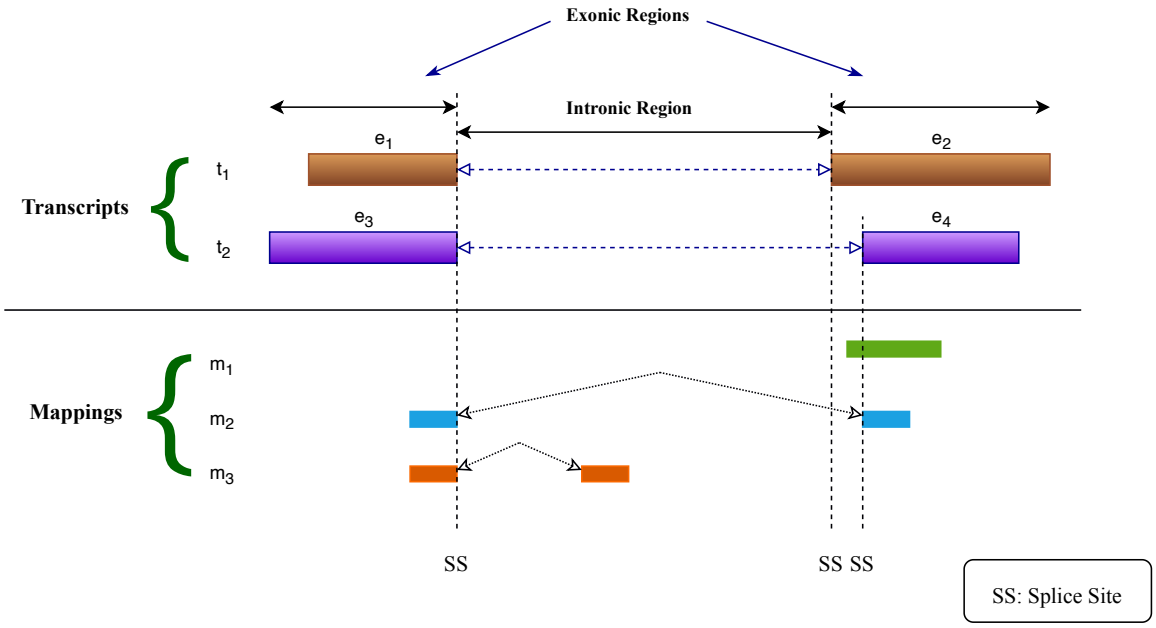


Figure 2.2: Read compatibility shown on a gene with 2 isoforms each of which containing 2 exons. There are two valid junctions according to this model: $e_1 \rightarrow e_2$ and $e_3 \rightarrow e_4$. A read mapping is compatible with a transcript if it is fully covered by an exon or it is split exactly on the junction breakpoints. Read mapping m_1 is compatible with t_1 but not with t_2 since it crosses beginning of e_4. m_2 is only compatible with t_2 based on its junction end points. m_3 is not compatible with this model because it has a junction to intron.

Another way of mapping RNA-Seq short reads is to align them to a reference transcriptome which is artificially built from a gene model by augmenting exon–exon segments [41]. But this is not preferred because it will only map reads to the junction sites that are already known and it will not be able to map reads to unexpected junctions that occur due to some events such as gene fusion, cryptic splicing, or exon skipping.

In the following subsection, some of the popular mappers are described.

2.2.1.1 HISAT

Hierarchical Indexing for Spliced Alignment of Transcripts (HISAT) [42] uses a Burrows-Wheeler transform (BWT) [43] and the Ferragina-Manzini (FM) [44] indexing scheme. It uses two types of indexes: (i) a whole genome FM index and (ii) many local FM indexes for rapid extension of the alignments that cover the whole genome and overlap with their adjacent regions by 1024 bp. For instance, for the human genome, 48,000 local FM indexes is built, each of which representing a genomic region of 64,000 bp. These indexes require around 4 GB of memory which is much less than the memory required for keeping a suffix array. The implementation of HISAT is done on top of Bowtie2 and it is intended to be used as the core alignment engine for the next version of TopHat, TopHat3.

There are three variants of HISAT being implemented: (i) HISATx1 which uses a single pass approach and aligns the reads independently to each other. (ii) HISATx2 is a two-pass version that generates a list of splice sites supported by the reads with at least 16 bp. These splice site information will be used in the second run to align the reads with short anchors (less than 16 bp). (iii) HISAT (the default version) combines the idea of the previous two versions and gains almost the same accuracy of HISATx2 in a single run. HISAT make use of splice sites found by earlier reads that were processed in the same run. HISAT does not depend on gene annotation information for mapping but has an option to use known splice sites from gene annotation.

In the first step of the algorithm, it tries to find candidate locations across the genome by searching through the global FM index which in most cases provides a small number of candidates. Then one of the local indexes will be selected for each candidate based on its location in order to align the remainder of the read. When the read involves long introns, two or more local indexes will be used up to a maximum intron size of 500,000 bp. For paired-end reads, each mate is aligned separately and then the alignments will be combined together. But if the alignment of one mate is not successful, the other mate's alignment will be used as anchors to map the unaligned mate. An issue happening in FM index when it has been built on large strings is too many cache misses due to lack of locality in the structure of BWT. Small size of local FM indexes gives it an advantage to fit in cache completely, thus avoiding cache misses. This makes HISAT run faster than other BWT-based mapping tools.

HISAT has different mapping strategies for different reads based on their position with respect to splice sites. Since global search is relatively time consuming, FM index search will be paused when the following conditions meet: (i) the partial alignment contains a least 28 bp; and (ii) there is only one exact match found for it. Then instead of continuing the search on the FM index, the partial alignment will be extended by directly comparing the remaining of read sequence and its corresponding genomic sequence. If a mismatch happens during global FM index search, the corresponding local index will be retrieved and the search continues on the local index until: (i) at least 8 bp has been matched; and (ii) 5 or fewer candidate locations are remained in the candidates list. In this case, the remaining of the read will be matched by local extension same as before and the compatibility of the split parts will be examined afterwards. When a 1 to 7 bp anchor is remaining, the program will use splice site information that was gathered so far (HISAT) or in the first run (HISATx2) to construct a hypothetical gene model and based on that, ignore the intronic regions in the extension step. In general, the combination of global search, local search and local extension makes HISAT run fast.

2.2.1.2 STAR

Spliced Transcripts Alignment to a Reference (STAR) [45] is a splice mapper based on sequential maximum mappable seed search in uncompressed suffix arrays which is specifically designed for RNA-seq data mapping. It is capable of mapping short and long reads (several kilobases) directly to a reference genome without any *a priori* knowledge of splice junctions' loci. STAR algorithm consists of two main steps: (i) seed searching; (ii) seed clustering, stitching, and extension.

In the first step it tries to locate the longest common sub-string between the read and reference genome sequence. This is done by locating seeds for a given read by sequentially finding its longest prefix exact match on genome which is noted as Maximal Mappable Prefix (*MMP*) in the original paper. The genomic location for such a prefix will be stored as a"seed" and the process continues for the remaining of the read. *MMP* in STAR is implemented through uncompressed suffix arrays because *MMP* can be easily extracted from suffix array by performing standard binary search and no additional computational effort is required. Because of the binary nature of suffix arrays, the search time will be logarithmic with respect to the reference genome length, allowing relatively fast search even against large genomes. Also, with a little computational overhead, all distinct genomic matches to a *MMP* can be reported which is helpful for "multimapping". It also enables finding mismatches and indel. If the *MMP* search cannot reach to the end of a read, the seeds will be extented later using alignment in order to report possible mismatches and indels. The *MMP* search is performed in both forward direction of the read sequence and its reverse complement.

In the next step, the closer seeds are first clustered together and then will be concatenated together assuming a linear transcription model. The locations of two seed are stitched together if: (i) the distance between the locations is not violating a user-defined maximum intron length L (10^6 in default settings); and (ii) the concatenation of the seeds is compatible to the original order of the seeds on the read. A dynamic programming algorithm is used to stitch pairs in which any number of mismatch and only one insertion or deletion is allowed. If an alignment cannot cover the entire read, STAR will try to find two or more windows that cover the whole read, which results is a chimeric mapping where different parts of the read map to distal genomic loci or different chromosomes or strands. In paired-end mode, STAR also marks a read as chimeric when a chimeric junction is located in the unsequenced part of a fragment between two mates. At the end, local alignment with scoring scheme is done if need be and the stitched regions with highest scores will be reported. A drawback of STAR is the memory usage. For human genome STAR requires at least $28GB$ available memory on the machine which makes it not runnable on mediocre systems such as personal computers.

2.2.1.3 GSNAP

GSNAP [46] uses hash-based indexing since it only requires locations of k-mers. It only reports the mapping location(s) of a read that have the highest score based on a scoring function that is designed in terms of the number of mismatches as well as gap or splice open penalty. The index contains 12-mers with a shift of 3 base pairs on the genome along with a sorted position list of the 12-mers. Because of the 3bp shift in index, the memory usage will be one third compared to the situation that every genomic location is indexed. The algorithms contains steps of generating, filtering, and verifying candidate genomic regions. Their program can detect splicing, multiple mismatches, long indels, and combinations of these events, up to a user-specified threshold, with the limitation of at most one splice or indel per read, provided the read has a consecutive stretch of 14 nt that exactly matches to the reference genome. GSNAP is able to use a user-provided gene model showing exon-intron boundaries in order to reduce false negative rate, but it is not dependent on this information and can evaluate the surrounding genomic region using probabilistic models of donor and acceptor splice sites. Splicing events may detect long-distance intrachromosomal deletions or inversions, and interchromosomal translocations as well as short distance splicing events.

Single nucleotide polymorphism (SNP) represents substitution of a single nucleotide at a specific position in the genome. The nucleotide variations for this position are termed as alleles where the minor allele is the second most common allele. One advantage of GSNAP is its SNP-tolerant alignment which treats minor alleles as match instead of mismatch. The minor alleles can be given to the program from databases such as dbSNP [47]. Depending on the error threshold e that is allowed by the user and the read length L, a multiway merging process can be formulated in two different ways to generate and filter genomic regions. When e is small, it uses a merging procedure based on *"Spanning Set"* of k-mers that filters regions based on the count of supporting k-mers. Otherwise, a *"Complete Set"* of overlapping k-mers will be considered which will filter genomic regions based on the pattern of supporting k-mers.

Spanning Set. A spanning set is defined as the minimal set of 12-mers that cover the read. Pigeonhole principle provides a lower bound on the number of mismatches; however, because of the indexing scheme of the method that samples every 3 nt, GSNAP construct six spanning sets, one for each shift of 0, 1, and 2 for both forward and reverse complement directions. It is possible that spanning set contains overlapping 12-mers, but an overlapping pair of 12-mers is considered as a single element. So The resulting set will be non-overlapping and pigeonhole principle holds where k non-supporting elements implies a lower bound of k mismatches. So the region can be filtered out if $k > e$.

Complete Set. GSNAP also has a strategy of using complete set of overlapping 12-mers that works with any error constraint, as long as the read an candidate region have at least 14nt consecutive matches. A multiway merging of position lists is done in two

passes (forward and reverse complement) to generate candidates. The pattern of 12-mers supporting each candidate region provides a lower bound on mismatches in a read. If the the distance between two supporting 12-mers is Δp, the minimum number of mismatches between them will be $\lfloor(\Delta p + 6)/12\rfloor$. This will be added up over the entire read to evaluate the read. At the end the candidates are verified by an alignment to the compressed genome.

2.2.2 Alternative Splicing

A single gene might have multiple RNA isoforms, which are the set of expressed transcripts coming from the same genomic region that can potentially code different proteins. This mechanism is called alternative splicing because it shows that a protein coding gene might have multiple splicing forms containing different exons. Alternative Splicing is believed to be a cause for protein diversity in eukaryotes [48]. Some studies estimate that alternative splicing occurs in approximately 95% of multi-exon genes [49]. There are different types of alternative splicing events that happen on the genome. The most common one is *exon skipping* or *cassette exon*. Other alternative splicing events might occur such as *intron retention* where the boundary of an exon is not preserved and beginning or end of the immediate intron is present in mRNA after splicing. In this event, a particular exon might be ommitted from the mRNA in some transcripts while it is present in some other transcripts.

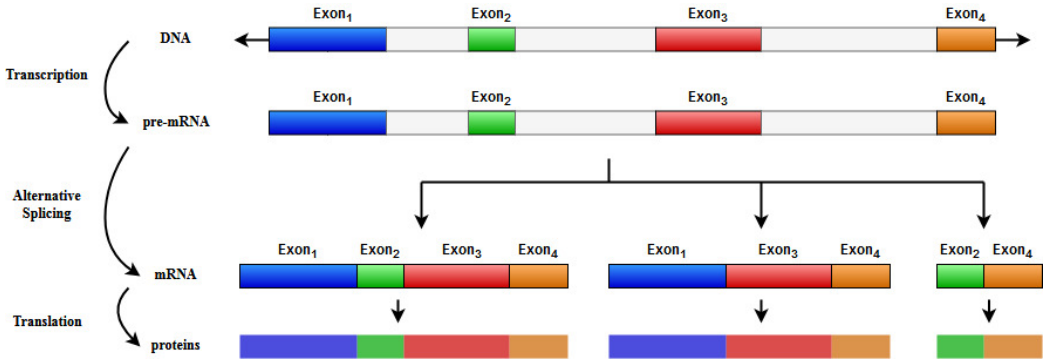


Figure 2.3: Alternative splicing in a gene containing 4 exons and 3 RNA isoforms. The introns will be removed from pre-mRNA during the splicing process. Then proteins are created from translation of mRNA.

Figure 2.3 shows the transition from DNA to protein. DNA is converted to pre-mRNA in a process called *transcription* and then during *splicing*, intronic regions will be removed from pre-mRNA to achieve mRNA. In this step a variety of mRNAs may be created due to alternative splicing, having a different selection of exonic regions in each mRNA. Finally, proteins are going to be constructed from mRNAs during *translation* step.

Abnormal splicing variants contribute to development of some genetic disorders and cancer [50].

2.2.3 Isoform Identification and Quantification

Definition *Isoform Identification:* Given a set of RNA-Seq reads R and their corresponding mapping locations on the reference genome M, the goal is to determine the set of expressed transcripts I in the sample.
Identification tools usually build a directed acyclic graph G that represents the mapping records on each gene and the existing paths in G correspond to a transcript or a partial transcript. By finding a set of paths covering all the nodes and edges in G, the expressed transcripts for each gene will be determined. However, the challenge is to find the set of transcripts that cover all observed exons and junctions in R efficiently. Two types of graphs are usually used for isoform identification: *Splicing Graph* and *Overlapping Graph* [51]. For instance, Cufflinks [52] is a transcriptome identification tool based on overlapping graphs which tries to find the minimal set of isoforms covering the mapping records.

Definition *Isoform Quantification:* Given a set of RNA-Seq reads R and the set of expressed transcripts I, the goal is to quantify the number of full length copies which is called expression level, for all the isoforms in I. The number of reads coming from a given transcript t_i is proportional to both transcript length (l_i) and the expression level (e_i) of t_i. A read r is considered as a mapped read to a transcript t_i if and only if r is fully compatible with t_i as defined in 2.2.1. Multi-mapping reads that can be mapped to more than one location on the transcriptome cause the main challenge for isoform quantification. Different isoform quantification tools have different strategies for handling mapping ambiguity. (i) Expectation-Maximization (EM): in this strategy, multi-mapping reads are distributed into mapped transcripts proportional to their relative original abundance. They are usually well-suited for comparing expression level of different transcripts and genes within a single sample. (ii) Read counting: simply counting the number of reads mapped to a transcript or gene. This approach cannot deal with multi-mapping reads since they either consider all the multi-mapping reads or discard them all. This type of methods are usually used for comapring expression levels in different samples and not within a sample. Because the values are not normalized by transcript length.

2.2.4 Aberrant Detection

Not all transcript sequences follow normal splicing. In normal splicing mechanism, exonic regions are always combined together in consistence with their relative order on the genome. However, this is not always the case. There are some transcripts that are different from normal splicing in a sense that they do not follow the relative order of exnos on the genome. The reads that break the order of exons on transcripts/genes are called *chimeric reads*. Detecting these aberrations is difficult by using identification and quantification strategies described above. Besides, it has been shown that these aberrations can be potential drivers

for some human disease and cancer types such as prostate cancer [53]. Hence, other effective techniques are required to detect these events.

The detection of transcriptome aberrations can be done either by mapping-based approach or assembly-based approach. In mapping-based, the reads are first mapped to a reference and the reads that cannot be fully mapped will be examined more deeply to detect aberrations. Aberrant detection tools will then use the mapping information from these reads to report types and locations of aberrations. In assembly-based approach, first *de novo* assembly on RNA-Seq reads will be done to generate possible contigs. Aberrant detection tools will map these contigs back to reference genome to predict and report the events. Although this approach is still not completely independent from reference genome, it reduces the bias of gene annotation and it is more likely for them to report more novel events.

Here, I describe two types of transcript aberrations: *Gene Fusion* and *Non-Canonical Splicing*.

2.2.4.1 Gene Fusion

Fusion gene is a gene that is made by joining some parts of two previously separate genes. Gene fusions are generally produced through interchromosomal and intrachromosomal rearrangements. As a result the sequence content of the read cannot be found consecutively on the reference genome which makes it difficult for the typical isoform identification and quantification tools to detect them. Gene fusions are known to play an important role as drivers for cancer and tumor progression. Recently, more and more novel gene fusions have been identified because of rapid advancements in high throughput sequencing which means they are more common than previously imagined [54]. Although many of fusion proteins have low pathogenetic importance, a detailed study on them can provide vital clues for finding new therapies in oncology. Some computational tools such as TopHat-Fusion [55] and deFuse [56] have been developed to either detect potential fusion reads or report detailed information about predicted fusion genes found in a sample.

2.2.4.2 Non-Canonical Splicing

In canonical pairing, the 3' end of one nucleotide is linked to the 5' phosphate of the next nucleotide, conferring an overall 3' to 5' directionally consistent with the parent DNA strand. Since the discovery of mRNA, it has become increasingly clear that there exists a vast array of functionally important non-coding RNAs (ncRNAs) not translated into proteins. These include snRNAs which form part of the spliceosome, rRNAs which constitute a part of ribosomes, and miRNAs which regulate gene expression by silencing mRNA. Another class, circular RNAs (circRNAs), recently garnered attention after they were discovered to be a pervasive product of eukaryotic gene expression. As their name suggests, circRNAs are covalently closed circular ncRNA moecules in which each pair of nucleotides is joined by

canonical 5'-to-3' linkages. The existence of circRNAs in eukaryotic cells was first reported in 1979 [2], but they did not garner widespread attention until 2012, when they were found to be abundantly expressed across the human and mouse genomes [3]. It was subsequently determined that circRNA expression extended to essentially all eukaryotes [5]. Extensive efforts have since been made to understand the characteristics, function, and possible applications of circRNAs.

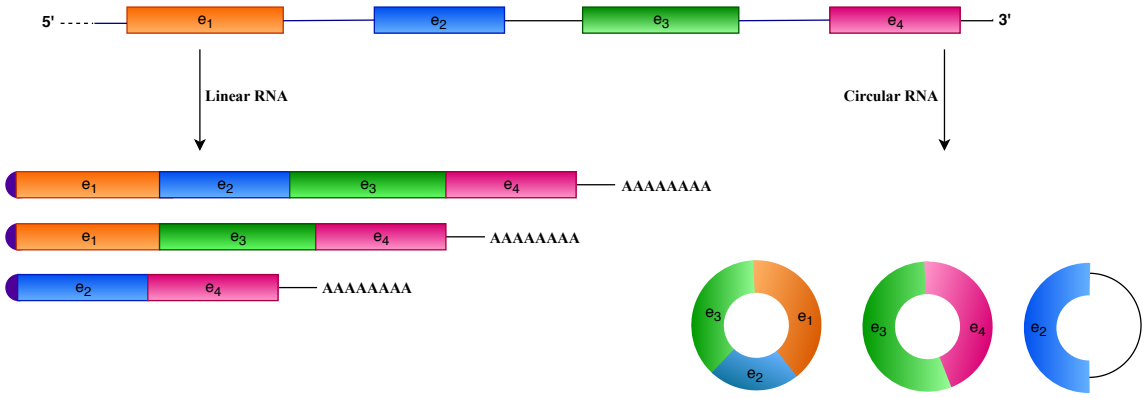


Figure 2.4: Alternative splicing resulting in linear RNAs (left) and circular RNAs (right) from the same gene sequence. Linear RNAs have poly(A) tail while circular RNAs do not possess this tail because of their circular shape.

In canonical splicing, the upstream 5' splice site of an intron is joined to the downstream 3' splice site. In contrast, a circRNA is created when the donor 5' splice site of an intron is joined with the 3' splice site of a separate upstream intron as illustrated in Figure 2.4. This event, termed backsplicing, appears to involve the same splicing signals and machinery as canonical mRNA splicing [6]. Circularization has been associated with long flanking intronic sequences and increased RNA polymerase II transcription speed [7, 8]. In addition, RNA binding proteins have been implicated in the formation of circRNAs. There is evidence suggesting that backsplicing occurs both co-transcriptionally and post-transcriptionally, with the length of a circRNA's flanking intron repeats serving as an important determining factor [8]. Backsplicing is far less efficient than canonical splicing, providing a probable explanation to the low steady-state levels at which most circRNAs are expressed. Nonetheless, circRNAs are more stable than their linear counterparts which lead them to accumulate within cells to allow widespread detection.

The abundant expression of circRNAs across the eukaryote domain of life indicates that they were either conserved over a billion years of evolution or evolved independently in multiple kingdoms, both of which suggest a functional role [5]. However, while specific functions of select circRNAs have been ascertained, a broader role remains elusive. As circRNAs are frequently transcribed from the same genes as mRNA, their formation competes

with the synthesis of mature mRNA [12]. Exon-intron circRNAs, which retain introns, have been observed to modulate the expression of their parent genes [57]. CircRNAs may also be significant in cancer and as a regulatory mechanism. Fusion circRNAs, transcribed from exons originating from distinct genes, have been found to promote oncogenesis and confer resistance to therapy [14].

Despite their abundant expression, circRNAs were largely overlooked prior to 2012 due to selection for poly(A) tails in the preparation of most RNA sequencing (RNA-Seq) libraries. However, some circRNAs would inevitably have survived poly(A) selection, futher revealing a limitation of existing RNA-Seq tools [16]. This revelation spurred the development of over 10 tools detecting backsplice junctions, though comparisons show their output diverge dramatically with no tool having a clear advantage [26, 27]. Some tools exclude intronic or intergenic circRNAs and there is currently no tool capable of detecting fusion circRNAs [27]. Dependencies similarly varied between the tools, with several capable of detecting circRNAs de novo while others require gene annotations.

2.2.4.2.1 CircRNA Detection Although widely expressed, circRNAs were largely overlooked prior to 2012 due to the selection for poly(A) tails in the preparation of most RNA sequencing (RNA-Seq) libraries [16]. Several algorithms were patched to detect circRNAs but results remain divergent and plagued by false positives [26], underlining the challenges faced in accurately detecting circRNAs.

In single-end RNA-Seq data, only reads crossing a backsplice junction can distinguish circRNAs from linear isoforms. This approach greatly reduces the sample size available for circRNAs and may further be susceptible to read density biases that are not completely understood. Moreover, homology in the form of degenerate sequence motifs at exon boundaries may lead to incorrect calls arising from sequencing errors.

Common strategies used to limit the number of false positives, such as the use of gene annotations or the requirement for canonical splice signals, compromise sensitivity. Other algorithms require high absolute read counts, but many important genes are expressed only at low levels. Still others use a statistical approach that relaxes other requirements. This has shown promising results, but improvements are required to minimize the reliance on stringent post-filtering.

Several methods including CIRCexplorer [19] and circRNA_finder [22] use spliced alignment algorithms such as STAR and build an extra index based on gene annotation file to search through the reported chimeric mappings. CIRI [17] uses a genomic mapper, BWA-MEM, and scans the reported alignments twice to find both balanced and unbalanced backsplice junction split reads. It checks the vicinity of soft-clipped segments of the split reads to match sequence content and splice signals. During the second scan unbalanced junction reads are detected using a dynamic programming alignment algorithm. There exist other methods including MapSplice [23] and segemehl [24] that are capable of performing de novo

splice mapping to identify various types of splice junction events. Their downside is that they are very slow and resource intensive. KNIFE [21] is another circRNA detection tool that first maps RNA-seq reads to genome, rRNA sequences, transcriptome and then tries to map the remaining reads to some customized linear and back-spliced junction databases. To generate the customized sequence databases, all possible break points that can form a regular or back-splice junction site are considered. Finally it maps the un-aligned reads to the customized databases and reports circRNAs. This tool is also resource intensive and slow.

The techniques used in the preparation of RNA-Seq libraries are varied, with profound consequences on the presence of circRNAs. Poly(A) selection depletes cirRNAs as they do not typically possess sequences rich in adenine. In contrast, RNase R, which depletes essentially all RNA with free 3' tails, enriches circRNAs. Enriching the transcriptome by depleting rRNA preserves circRNAs. A lower limit of 200nt is frequently imposed on RNA-Seq libraries, excluding the smallest circRNAs. Random priming preserves circRNas, but Oligo(dT) priming depletes them because they typically lack poly(A) sequences. Finally, circRNAs are depleted if RNA is not fragmented before adaptor ligation or priming as they do not have free ends.

Comparison of circRNA detection tools. A recent comparison of circRNA detections algorithms circRNA_finder, find_circ, CIRCexplorer, CIRI, and MapSplice applied to Hs fibroblast RNA-seq libraries found dramatic differences in their outputs [26].

Using the criteria of at least three reads spanning a proposed backsplice junction, they detected anywhere between 1532 (circ_finder) and 4067 (CIRI) circRNAs, with only 16.8% of observed circRNAs detected by all five algorithms. Among the detected circRNAs, between 12% (MapSplice and CIRCExplorer) and 28% (CIRI) were depleted by RNase R. Those that were detected by only one algorithm were generally more susceptible to RNase R, with up to 59-79% (circRNA_finder, CIRI, and find_circ) being depleted. This is particularly evident in the top 100 most expressed circRNA candidates detected by one algorithm, where 75-88% (circRNA_finder, CIRI, and find_circ) were depleted. These algorithms may be confounding common linear species with circRNAs because of less stringent requirements.

To assess sensitivity, the average number of reads of the backsplice junction in circRNAs detected by all algorithms was considered. The results were again highly divergent, varying between 9 (circRNA_finder) and 22 (CIRI). Sensitivity was correlated with the duration used to analyse the datasets which ranged from hours (find_circ and circRNA_finder) to days (CIRCexplorer, CIRI, and MapSplice) on the benchmarking system.

The minimal distance between splicing sites also differed between the algorithms. CIRI and circRNA_finder predicted circRNAs with splice sites separated by fewer than 100 nucleotides, while MapSplice appeared to require a distance of at least 300 base pairs despite setting the minimum distance at 200. Notably, allowing for closer proximity of splice sites

increases false positive rate as shorter circRNA candidates were much more readily depleted by RNase R.

Finally, whereas cicRNAs detected by only a single algorithm were much more susceptible to RNase R depletion, circRNAs detected by multiple algorithms are less likely to be false positives. Indeed, only 6% of the circRNAs predicted by all algorithms were depleted by RNase R.

Overall, CIRCexplorer and MapSplice were most reliable, although they require gene annotation lists and take longer to complete. In contrast, pairing circRNA_finder with find_circ produced reliable output while obviating the need for annotations. Notably, gene annotations are not required for find_circ, circRNA_finder, and CIRI.

At this time, there are no algorithms that detect fusion circRNAs from RNA-Seq data. Guarnerio et al. [14] used divergent primers that amplify RNA crossing the backsplice junction of candidate fusion circRNAs.

2.2.4.2.2 Benchmarking CircRNA-detection Algorithms. Benchmarking the performance of circRNA-detection algorithms is similarly challenging. Several methods have emerged, each with their own strengths and shortcomings, but a gold standard remains elusive.

The most widely employed approach to circRNA validation is RNase R treatment, which depletes essentially all linear RNAs while retaining circRNAs, lariat byproducts of intron splicing, and double-stranded RNA with 3' overhangs shorter than 7 nucleotides. CircRNA candidates that are depleted are likely to be false positives while enrichment provides validation. This approach is, however, subject to variability of RNase R treatment with fewer than half of the circRNAs resistant in one treatment being resistant in a second replicate. CircRNAs which are resistant in any replicate are frequently considered to be genuine. A further shortcoming is evident in the fact that several validated circRNAs have been known to be depleted by RNase R. Care must also be taken when considering fold-enrichment in treated samples, as confidence intervals will be larger for circRNAs which are detected at lower levels and different sequencing depths will change the expected result.

Other benchmarks have been used with varying degrees of success. Depletion in poly(A)-selected libraries can be used to validate circRNAs, although technical and statistical limitations similar to RNase R treatment apply. The quality of prediction of an algorithm can be assessed from the proportion of mate reads which map outside the proposed circRNA, although different interpretations for these reads exist. Template-switching artifacts are sometimes specific to RT, so a lack of RT specificity may provide an additional line of evidence although results in this area are conflicting. Simulated data is useful in identifying systemic limitations but may not reflect performance on real RNA-seq data.

2.3 RNA-seq Pseudo-Alignment

Many tools have been created to solve read-alignment problem in the past using different data structures. For example Bowtie [58] and BWA [59] use variants of FM-index [44] while tools like mrsFAST [60] use k-mer hashing strategy for mapping the reads to the reference. These tools try to provide accurate results in a timely manner. Although more recent tools such as STAR [45] have shown that rapid read alignment is possible and HISAT [42] provides a fast mapper with moderate memory usage, some of the provided information reported by mappers are not required in downstream analysis. For instance, in many transcript analysis tasks knowing the transcript and position in the transcript that the read maps to are enough. Thus, some tools have been recently created introducing novel concepts for ultra-fast read mapping. In this section two such tools are described.

2.3.1 Kallisto

Kallisto [28] is an RNA-Seq quantification tool that instead of mapping the reads to a reference genome, it pseudo-aligns them constructing a list of transcripts which are compatible with the reads and then quantifies the abundance of each transcript. The idea of pseudo-alignment came from the fact that accurate quantification does not require the mapping location of a read but rather which transcript it belongs to. Instead of aligning individual bases to a reference genome, i.e matching each base pair from the read to a reference, it uses a fast hashing of fixed size k-mers of reference *transcriptome de Bruijn graph (T-DBG)*. This gives it an advantage to run two orders of magnitude faster than other approaches giving similar accuracy.

At first it generates the T-DBG where each node v is a k-mer of the transcriptome. Each path on T-DBG corresponds to a transcript and a path covering of the the graph is defined as a set of paths whose the union includes all edges of T-DBG. In the original paper, k-compatibility class is defined as a multi-set on a vertex v which represents the transcripts associate to v. Furthermore, k-compatibility class of a path in the graph is defined as the intersection of the k-compatibility classes of the k-mers inside it. An error-free read can be represented as a path in T-DBG by matching it k-mers to the graph which gives its equivalence class information. An equivalence class for a read is defined as a multi-set of transcripts associated with the read which is also the k-compatibility class of the read assuming it is error-free. The equivalence class of a read ideally represents the transcripts it can be originated from which will be used in quantification step.

It uses the following formula for quantification and abundance estimation which involves maximum likelihood estimation using Expectation-Maximization (EM) algorithm.

$$L(\alpha) \propto \prod_{e \in E} \left(\sum_{t \in e} \frac{\alpha_t}{l_t} \right)^{c_e}$$

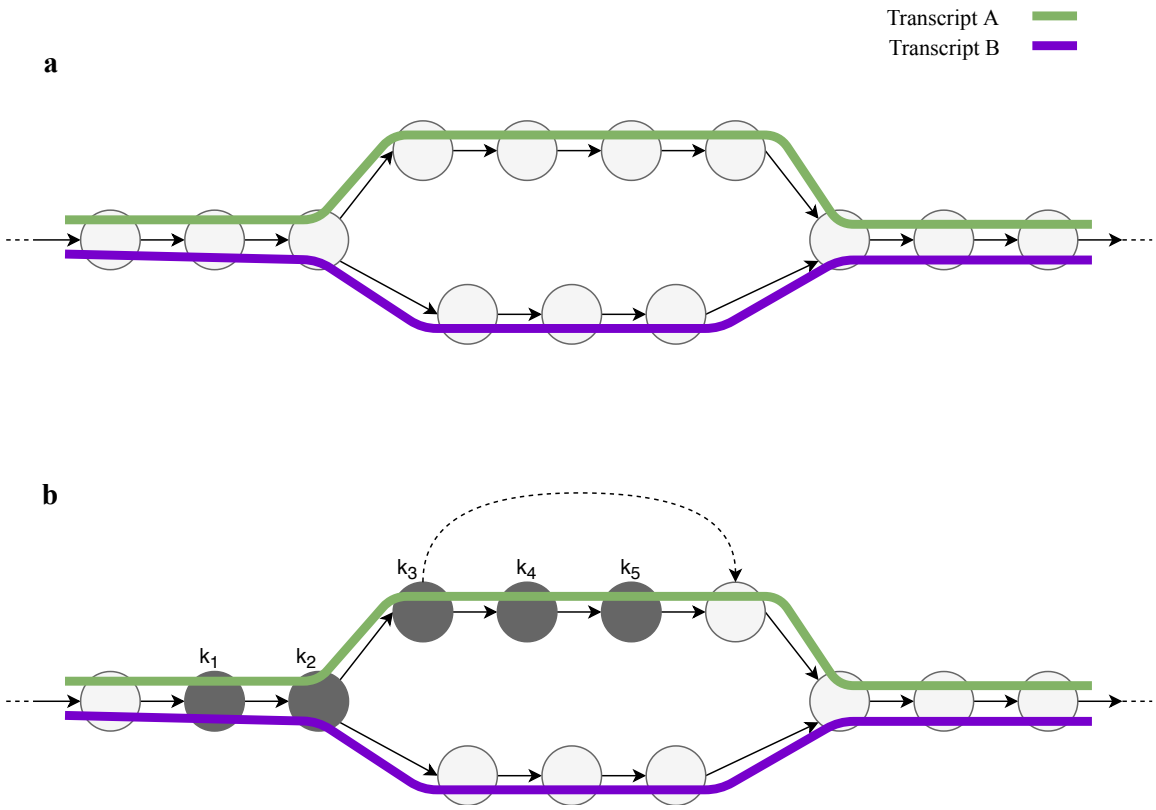


Figure 2.5: Overview of kallisto. (a) The constructed transcriptome de Bruijn Graph (T-DBG) where nodes are k-mers and each transcript corresponds to a colored path as shown and the path cover of the transcriptome induces a k-compatibility class for each k-mer. (b) The k-mers of the read (grey nodes) are hashed conceptually, to find its k-compatibility class. Non-informative k-mers with same k-compatibility class are skipped for speed-up as shown with dashed line.

where E is the set of equivalence classes, t is a transcript, l_t is the (effective) length of transcript t and c_e is the count of observing equivalence class e. Here, the parameters that must be optimized are α_t which are the probabilities of selecting fragments from a transcript.

2.3.2 RapMap

RapMap [29] is a tool that introduces a similar concept called *quasi-mapping* and an algorithm implementing this approach for mapping RNA-Seq reads to a transcriptome. Full base-to-base alignment is again prevented in this approach which makes it substantially faster than existing alignment tools. Not all the information a mapping tool provides is required for some transcriptome downstream analysis. For each query (read) r, RapMap provides the transcript it originated from, strand, and the position. This information is enough for scenarios like transcript quantification, clustering of *de novo* assembled tran-

scripts, and filtering of potential target transcripts. But it might not be appropriate for variant detection.

In their algorithm a combination of different data structures has been used– suffix array $SA[T]$ of the transcriptome T, a hash table h mapping each k-mer in T to its SA interval (default k = 31), and an efficient rank data structure. T is a concatenation of all transcript sequences where a single separator \$ is added between two adjacent transcript sequences. The original T should be kept as well as the transcripts' name and length. The algorithm for quasi-mapping is done in a repetition of three steps below: (i) finding the next hash table index for the k-mer k_i that starts from the current query position i; (ii) computing the next maximum mappable prefix (MMP) on the SA starting from current k-mer – the longest substring of the read that is prefixed by k_i appearing in the reference sequence is denoted as MMP_i; and (iii) determining the next informative position (NIP) by performing a longest common prefix (LCP) between two chosen suffixes on SA.

The k-mers of r are first hashed one by one from left to right until some k-mer k_i is found in h and points to a valid SA interval. Then MMP_i can be found using a SA binary search. Since the interval that $k-i$ can be extended in is known on the suffix array, the binary search will be bound only to this interval which is usually small. Although, the base appearing immediately after the MMP_i will not agree with the corresponding base on the SA entries inside the SA interval, it is possible that suffixes at the lower and upper bound of the SA interval be the same. In other words assuming the interval on SA corresponding to MMP_i is $[b, e)$, we will have: $|LCP(T[SA(b)], T[SA(e-1)])| > |MMP_i|$. This might happen because of a sequencing error or a variant. Thus, instead of starting from the next unmatched base pair, RapMap skips uninformative bases i.e the bases that most likely return the same set of transcripts and positions. Next informative position (NIP) for a MMP_i with $[b, e)$ interval is defined as follows:

$$NIP(MMP_i) = |LCP(T[SA(b)], T[SA(e-1)])| + 1$$

The search procedure will then be repeated from $i + NIP(MMP_i) - k$ on the read. This will continue until the $NIP(MMP_i) < l_r - k$; where l_r is the read length.

The result of this procedure will be a set of triplets containing read position, MMP orientation, and SA interval. In order to determine the final mapping result, the intersection of transcripts appearing in the set mentioned earlier that have consistent orientation will be calculated. Meanwhile, small matches that are not likely to to result in a correct mapping are filtered out during the process of finding the set of *best* matching transcripts and positions. Quasi-mapping is different from Pseudo-alignment that was introduced in Kallisto [28] in some ways. For example, pseudo-alignment only finds compatible targets for the reads and further processing is required to find the mapping location, while quasi-mapping tries to find the *best* mappings.

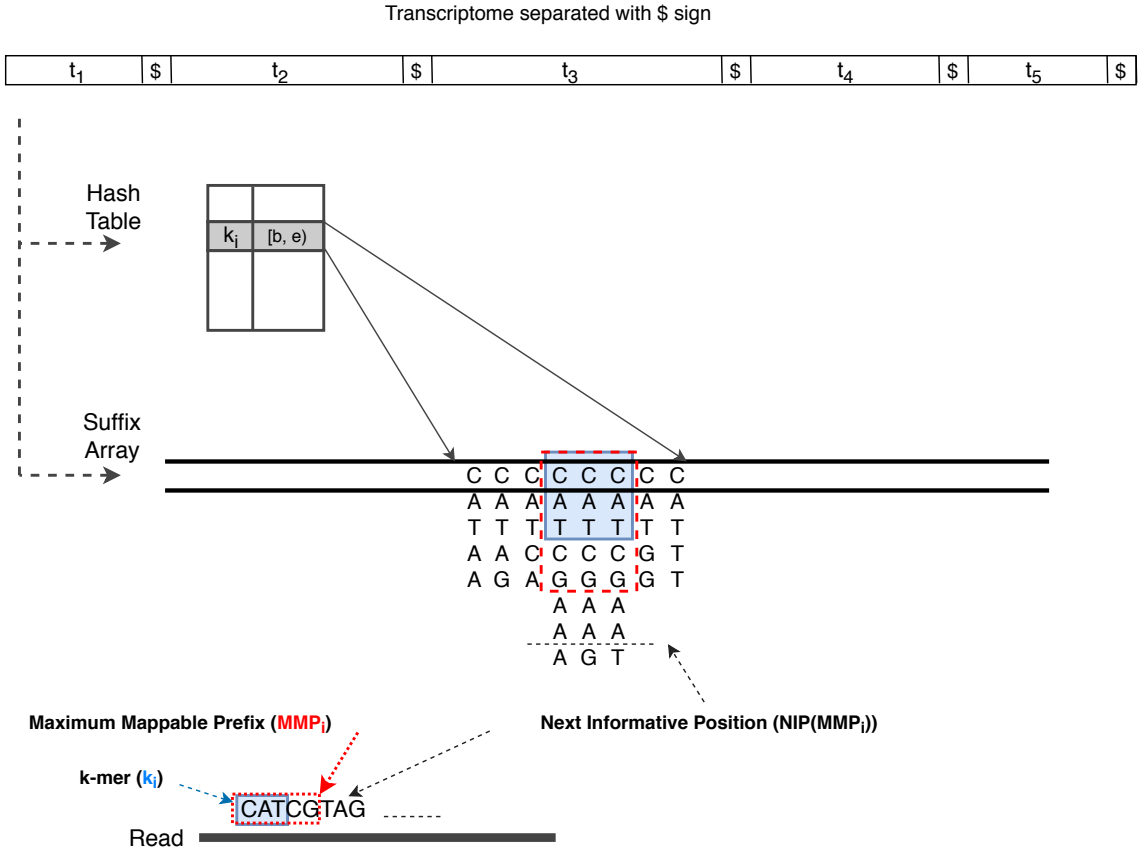


Figure 2.6: Overview of RapMap. The transcriptome (in this example: t_1 to t_5) is concatenated with \$ sign, on which a hash table and a suffix array (SA) is constructed. The mapping operation begins with a k-mer (here, $k = 3$) mapping to an interval [b,e) in the SA. Given this interval and the read, MMP_i and $NIP(MMP_i)$ are calculated as described. The search for the next hashable k-mer begins k bases before $NIP(MMP_i)$.

2.4 Conclusion

In this chapter, the Next Generation Sequencing (NGS) technologies for DNA and RNA sequencing are described. RNA sequencing has many applications, including aberrant detection such as gene fusion and circRNA detection. Several circRNA detection methods exist which use general-purpose RNA-Seq mappers. As a result, their sensitivity towards circRNA detection highly relies on the underlying RNA-Seq mapper. The RNA-Seq mappers are mainly designed to map linear RNA reads back to a reference by doing splice alignment. These mappers also report some of the chimeric reads including back-splice junction reads. Reporting chimeric reads is not the main aim of such tools and modifying them towards this goal may increase their running time. On the other hand, pseudo-alignment tools are ultra-fast methods to locate the reads on the transcriptome; and they only focus on isoform

quantification. In this thesis, we focus on developing a highly sensitive circRNA detection method by applying the idea of pseudo-alignment technique to detect chimeric reads.

Chapter 3

Method

In this chapter, we describe our developed method, CircMiner, that is designed for solving circRNA detection problem and we explain its modules and algorithms. CircMiner has two main modules: (i) pseudo-alignment and linear read filtration module; and (ii) back-splice junction detection module. CircMiner first rapidly filters those reads which fully map to linear transcripts or the genome using the pseudo-alignment module. The remaining partially mapped reads would be further analyzed using the back-splice junction detection module to extract the potential back-splice junction reads. For each such read, CircMiner maps its unmapped part to the surrounding loci within the same transcript and determines the breakpoints of the back-splice junctions. The back-splice junction reads are then grouped together based on their breakpoints within the same transcripts. Finally, circRNAs are reported along with the breakpoints, transcript information, and supporting reads spanning the back-splice junction. These steps are illustrated in Figure 3.1. Furthermore, Figure 3.2 depicts the overview of the read processing for linear and circular reads.

3.1 Rapid Splice-aware Read Filtration

CircMiner starts by rapidly discarding reads that do not provide any signal for aberrations, i.e., reads that originate from known transcripts or the reference genome and therefore could be mapped using splice-aware mappers. For a given read, CircMiner partitions it into non-overlapping segments, also known as seeds. However, this restriction can easily be loosened to consider overlapping seeds. For each seed, CircMiner queries exact matching positions on the genome from its index. Next, CircMiner tries to find the compatible chains of these genomic locations on the transcriptome and genome. Note that properly mapped reads can be identified if the relative order and distances of the seeds on the read are consistent with the order and distances on the transcriptome or genome. Finally, CircMiner goes through each compatible chain and tries to extend the chain to full alignment. Reads whose chains are successfully extended to full alignments are eliminated from further processing. In the following subsections, we will provide details of read filtration module.

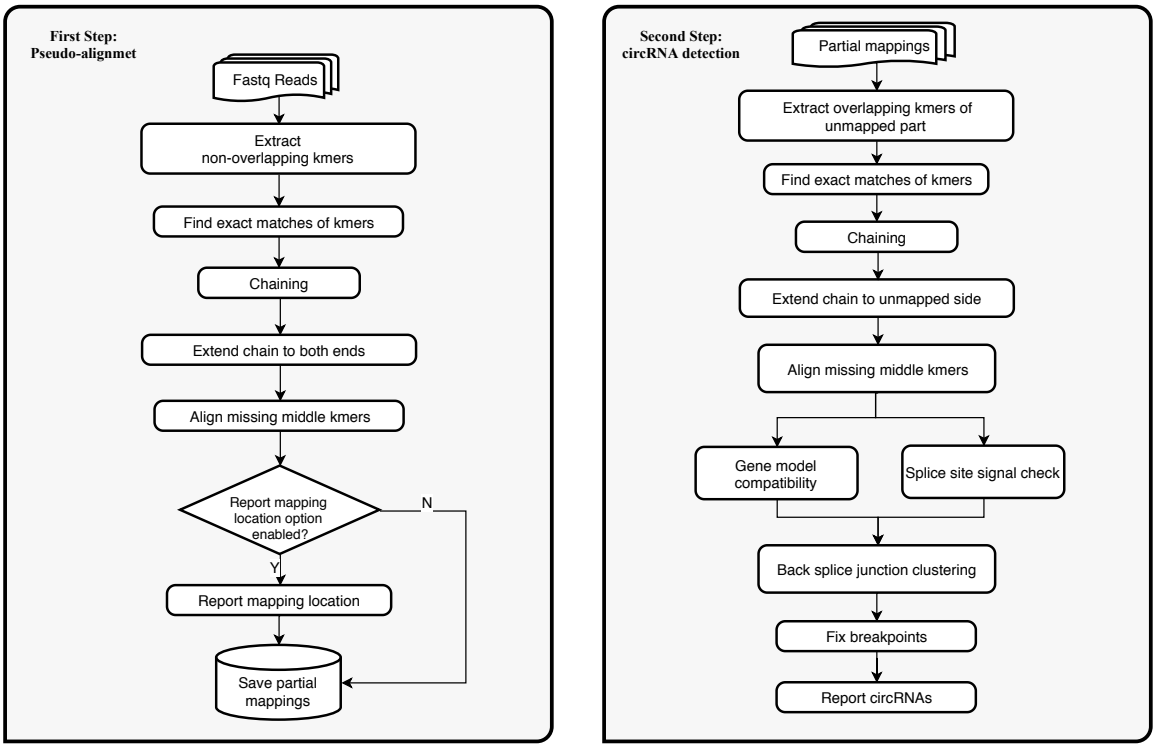


Figure 3.1: A flowchart of CircMiner's two main modules (pseudo-alignment and circRNA detection). Pseudo-alignment module is responsible for rapid linear read filtration and candidate back-splice junction read selection. CircRNA detection module only processes the saved candidate back-splice jucntion reads from previous stage and tries to locate the unmapped segment of the read.

3.1.1 Indexing

CircMiner builds a hybrid index that consists of two parts: (i) genome index, and (ii) transcriptome index. For the genome index, CircMiner uses a modified version of mrsFAST-Ultra's [61] index, which is a hash table of k-mer prefixes. Each entry of the table, p, keep a sorted list of genomic k-mers (and their coordinates) that have p as their prefix. Querying a given k-mer from this index requires $O(\log |P|)$ where P is the set of locations on the genome whose their prefix of size p that exactly matches with the same size prefix on the seed. This query is done in two steps: (i) an $O(1)$ lookup of prefix p-mer is performed on hash table which returns P that is sorted by sequence content lexicographically and then by genomic location; and (ii) a binary search requiring $O(\log |P|)$ time on the sequences will return all the matches. In practice, the parameter p is smaller than k but large enough that memory usage for the index is reasonable for contemporary computers. For example, if k is 19bp, then a prefix of size 14 is reasonable for generating fast lookup table with reasonable memory usage.

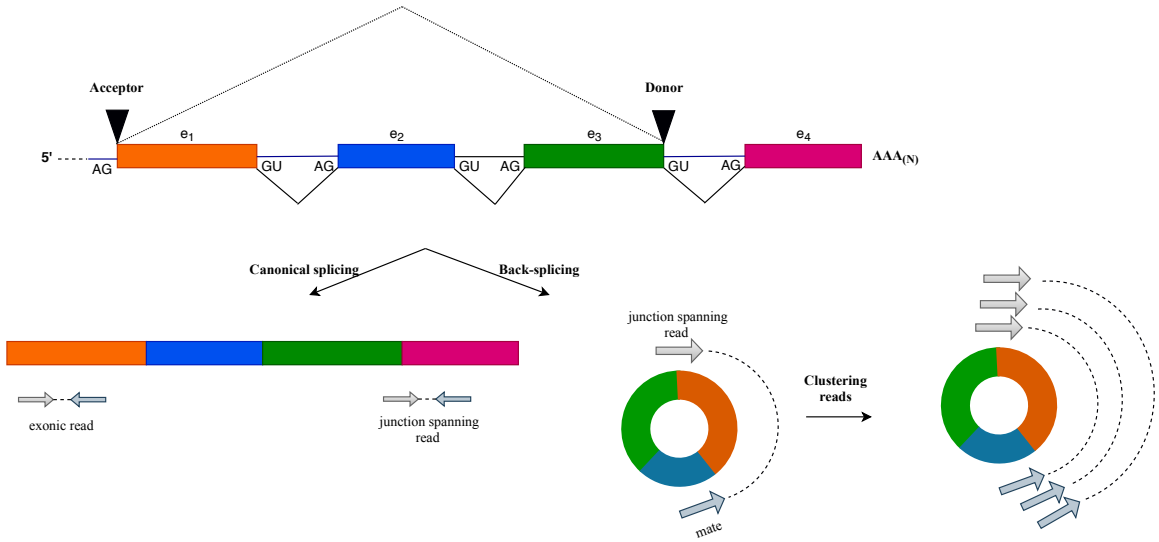


Figure 3.2: Canonical splicing vs. back-splicing formed from the same transcript. All the linear reads including exonic and junction spanning reads are eliminated in the filtration stage and confirmed back-splice junction spanning reads are then clustered together based on same breakpoint to find circRNAs.

In the implementation, multiple chromosomes from the given reference file are concatenated together while keeping the merged size smaller than a user defined threshold. For human genome, we set the threshold to 1.2 billion, so we end up having 3 contigs from concatenation of the chromosomes where chromosomes are separated by putting a sequence of N between them. This will avoid indexing k-mer on the concatenation point between chromosomes. Then, the index is created separately for each new contig as described above.

Two versions of index are available to use: (i) compact version, which only keeps the number of hits per p-mer, and (ii) comprehensive version, where extra information is also kept per p-mer along with the number of hits. The compact index is built faster and requires less storage space on disk (around 2 GB for human genome), but has the drawback of being slower in loading since the genomic locations in the hash table should be recalculated on the fly in the loading stage. When comprehensive version is used, the k-mer size ($k \geq p$) should be specified in the indexing stage and the same k-mer size should be used in the searching stage unless a new index is built. In this version, in addition to the number of occurrences, a list of pairs is also stored per p-mer. Each pair consists of a genomic location for a k-mer and its suffix which is a $(k-p)$-mer. Such pair is stored in p-mer's list, stored first by suffix content and then by location to make the exact match search faster.

To carry out pseudo-alignment at transcriptome level, CircMiner uses a secondary index to keep the gene model and transcriptome information from an annotation file (e.g. GTF file). CircMiner builds an interval tree from the exons of the transcripts that are obtained from the gene model as demonstrated in Figure 3.3. The interval tree is queried to see if

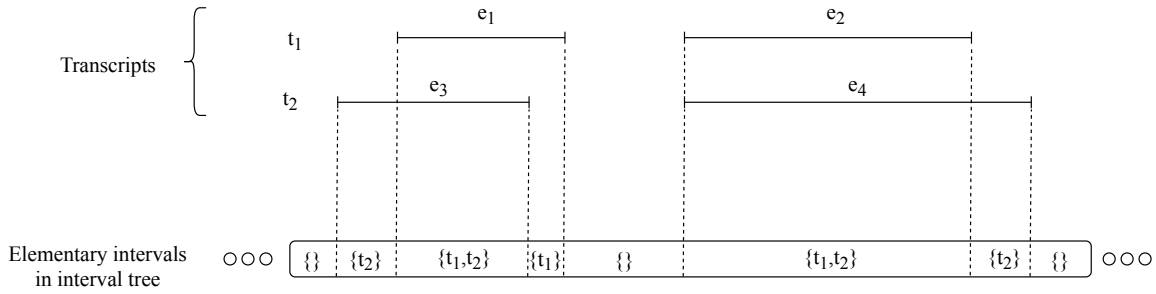


Figure 3.3: An example of an interval tree constructed on two transcripts t_1 and t_2. Each interval (exon) divides the linear number space (genome) into partitions called elementary intervals. As a result, each elementary interval is either completely inside an exon or completely outside of it. The data structure used for elementary interval keeps the information about the original transcripts that overlap with it.

a given genomic interval (e.g. k-mer) is on a gene and if so, to obtain all the transcripts that overlap with this genomic location. Each search query on the interval tree requires $O(\log |T|)$ time with $|T|$ being the number of segments in the interval tree T. Note that many of k-mers fall in intergenic regions and thus there is no need to query them. To reduce the number of interval tree lookups, CircMiner augments the interval tree with a simple, compact yet powerful bitwise array that corresponds to all the intergenic locations on the genome. If a location on the genome is intergenic, the value of that location in the array will be 0. In practice, the array reduces the total number of lookups on the interval tree almost threefold.

3.1.2 Optimal k-mer Chaining

CircMiner partitions a given read r into non-overlapping k-mers with starting positions $\mathcal{S} = (s_i)_{i=1}^{n}$ where $n = \lfloor \frac{|r|}{k} \rfloor$, s_i is the starting position of the i-th k-mer on the read, and $s_i < s_{i+1}$. For the seed starting at s_i, the set of its exact matches on the genome, $L_i = \{l_i^1, l_i^2, \cdots, l_i^z\}$, is extracted from the index where l_i^j is the starting position of its j^{th} match. A valid *chaining* $C = \{\mathcal{S}', \mathcal{L}'\}$ of r can be therefore defined by: (i) $\mathcal{S}' = (s_i)_{i=1}^{m}$, a sublist of $\mathcal{S}$ containing m seeds ($m \leq n$); and (ii) $\mathcal{L}' = \{l_{a_1}, \cdots, l_{a_m} \mid l_{a_j} \in L_j\}$, exact matches for seeds (in $\mathcal{S}'$) that preserves their relative order on the same chromosome. CircMiner scores a chain C as follows:

$$Score(C) = \alpha.m.k - \beta \sum_{1 \leq i \leq m-1} \mid Dist(l_{a_{i+1}}, l_{a_i}) - (s_{a_{i+1}} - s_{a_i}) \mid$$

where α and β are reward and penalty constant coefficients in the chaining algorithm and $Dist(l_{a_{i+1}}, l_{a_i})$ is the distance between starting positions of two consecutive k-mers of $\mathcal{S}'$ on the transcriptome or genome.

In the *Score* function, $m * k$ is the total length of matched bases from the read to the genome which is used as a reward for chaining to encourage building longer chains. On the other hand, the penalty is the aggregated difference between seeds' distances on the read and their corresponding matching locations distance on the genome or transcriptome. For calculating *Dist* between two consecutive seeds from $\mathcal{S}'$, when both seeds are fully located within the exons of the same transcript, CircMiner would discard introns that fall into $[l_{a_i}, l_{a_{i+1}}]$ to get the accurate distance on the transcriptome. If multiple transcripts contain the seeds, the transcript resulting in the smallest distance will be considered. For intronic and intergenic seed locations, genomic distance is used in *Dist* function. In both cases, only a few base pairs difference is allowed. Otherwise, the distance is set to ∞.

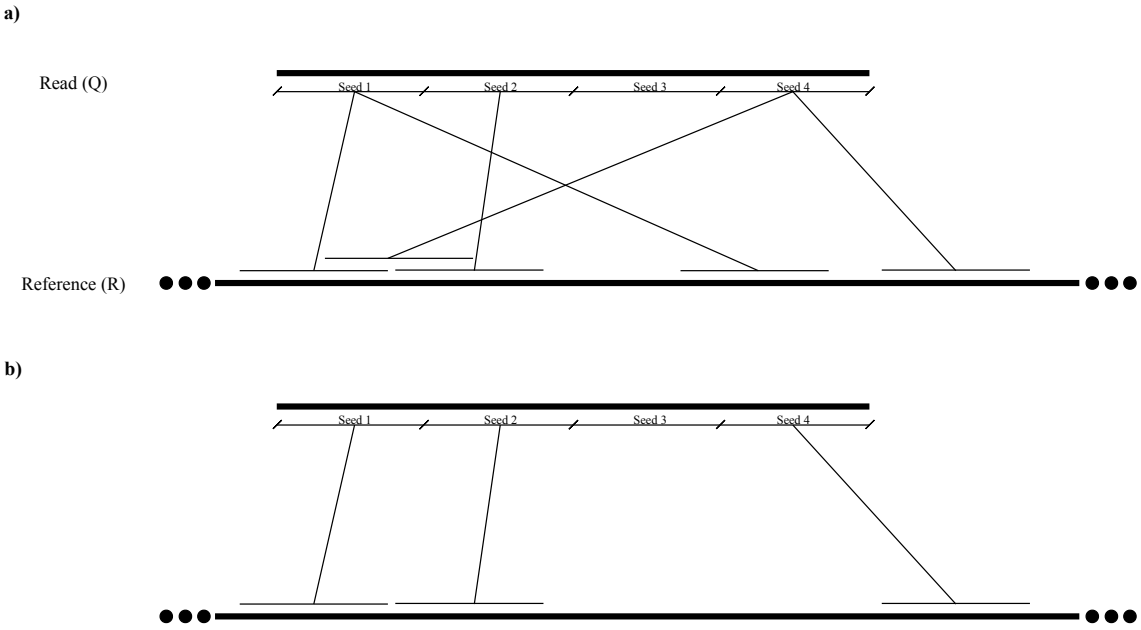


Figure 3.4: a. Read (Q) and its 4 non-overlapping same-size seeds. Each seed is connected with a line to its corresponding seed location on the reference genome (R). A seed can have 0, 1, or more seed locations.
b. A chaining example containing seeds 1, 2, and 4. In this example, seed 3 is missing.

In the chaining process, some seeds are ignored due to having too many seed locations. Seed limit (SL) is a user-defined parameter in CircMiner indicating the maximum number of locations a seed is allowed to have. Any seed whose number of occurrences on the reference genome is greater than SL will be discarded from the chaining stage. Changing the value of SL can affect the pseudo-alignment module's power in filtering linear reads as well as CircMiner's running time. A practical value for this parameter is empirically chosen to be 500 (see Results section for more details). CircMiner calculates scores for all chains of r using a dynamic programming approach. Top t chains ($t = 30$ by default) with the highest

chaining score are stored in Q^f for further processing in the next steps. CircMiner repeats the same procedure for the reverse complement of the read and stores in Q^r.

Existing chaining algorithms. In order to find an optimal chain of k-mers, a weighted directed acyclic graph $G = (V, E)$ can be constructed where V is the set of all the seed locations found on a read. Two seed locations are considered co-linear if and only if the order of the respective k-mers is the same in both sequences (as depicted by non-crossing lines in Figure 3.4). The weight on each edge is set to the difference of score when traversing the edge. It is calculated based on the scoring function. So to solve the chaining problem, we define and solve an equivalent of chaining problem on graph G which is a Directed Acyclic Graph (DAG). The problem to solve on DAG G is finding maximum weighted path on this graph. The typical dynamic programming (DP) approach to solve this problem on a DAG visits all the edges so its time complexity is $O(V + E)$ and memory complexity is $O(V)$. In the worst case that the graph is dense, the explained dynamic programming approach will be quadratic in terms of the total number of seed location on the read. There are data structures that can be used to reduce the time complexity of chaining problem. The key is to compute the maximum value in each DP recursion more efficiently without generating all the edges of the graph. The improvement can be achieved by using efficient data structures that are designed for the Range Maximum Query (RMQ) problem which is a fundamental operation in the field of computational geometry [62]. The data structures that can be used here are Adelson-Velsky and Landis (AVL) trees, red black trees, and K-Dimensional (KD) trees. By using these data structures, the time complexity of chaining problem will be reduced to $O(V \log V)$.

In CircMiner, the basic dynamic programming approach is implemented due to several reasons. First, implementing the basic DP approach is easier and less prone to bugs. Second, the size of the graph in this case is relatively small because of the usage of Seed Limit parameter. Thus, the number of nodes in the graph (seed locations) are usually no more than several thousand. As a result, using the mentioned data structure cannot make a big difference in the actual running time of the chaining section. On the other hand, although using these data structure will reduce the time complexity, they have bigger constant values that are going to make them not as fast in practice, especially on small graphs.

3.1.3 Extension of High-Scoring Chains

In general, popular short read aligners (genomic or transcriptomic) obtain a potential alignment of a read by finding a substring on the genome or transcriptome which has the smallest edit distance that is less than a user-defined threshold e. In doing so, they sometimes utilize heuristics to clip the ends of the reads (low-quality bases or unmappable in close vicinity) and generate soft clipped reads.

Since k-mer chaining narrows down such potential mapping locations, CircMiner can efficiently check if a chain can be extended into full or clipped alignment by (i) extending

the missing terminal k-mers using a prefix or suffix alignment; and (ii) by using banded local alignment for the missing middle k-mers in $\mathcal{S} \setminus \mathcal{S}'$. Note that CircMiner will extract corresponding transcript segments for the extension step if the chain is located on the transcript.

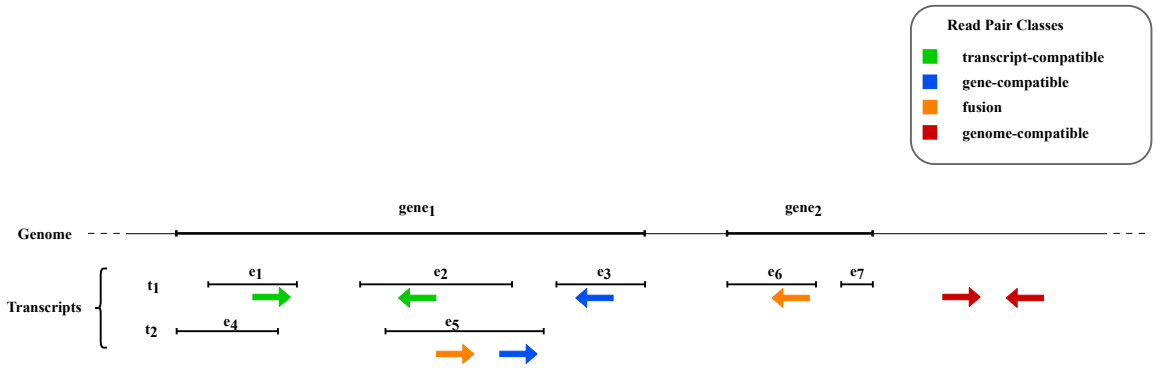


Figure 3.5: An example of four paired-end reads each from a different read-pairing class. The color of a read shows its corresponding class.

For any dataset that contains only single-end reads, any read that is not fully extended within user-defined error threshold will be considered for further processing. If the dataset contains paired-end reads, the extension is modified in order to take advantage of pair information. Consider $Q_1 = \{Q_1^f \cup Q_1^r\}, Q_2 = \{Q_2^f \cup Q_2^r\}$ to be the top candidate chains for mate one and mate two, respectively. CircMiner favors transcriptome mappings to genomic mappings. In doing so, it considers all the pairs from Q_1 to Q_2, where one of the chains is on forward strand (F) and the second one on reverse strand (R). CircMiner then classifies the pairs with respect to the following priorities: (i) transcript-compatible: all seeds from both mates are located on the same transcript. (ii) gene-compatible: all seeds are located on different transcripts of the same gene. (iii) fusion: the seeds of the two mates are on transcripts of two different genes. (iv) genome-compatible: where all seeds are located in intergenic regions and leftmost seed to the rightmost seed is less than a user-defined distance D (default value: 20,000bp). If a pair cannot be placed into any of the above categories, it will be discarded. For the remainder of the pairs, the extension will be performed within the selected region (e.g., transcript compatible pairs will be only extended in transcript boundaries).

3.1.4 Extraction of Back-splice signals

A back-splice junction read does not have a (co-)linear chain that covers the full length of the read. Thus it may have one or two chains in the top scoring chains that represent suffix and/or prefix of read. The goal of this stage is to identify and extract these reads where, (a) one mate is fully extended and the other mate has a full prefix or suffix extension on

the transcript; or (b) both mates have a full prefix or suffix extension. Specifically, any read that is fully extended on a transcript and has a proper orientation (FR mapping) will be marked as concordant and removed from further processing. A fully extended read on a transcript with incorrect orientation (RF mapping) is a potential non-spanning (does not cover the breakpoint) supporting read for back-splice junctions and is retained for circular RNA refinement. If a read has a partial extension on the transcript that is a prefix or suffix of the read, then that read has a potential back-splice signal and will be marked for the next stage. Finally, if there is no full or partial extension available for the read on any transcript, but it has a full extension on the categories (ii), (iii) and (iv), then it will be discarded from further processing. Note that, the information for the discarded reads, including their mapping information is recorded and can be extended to full alignment information upon user request and at an additional cost.

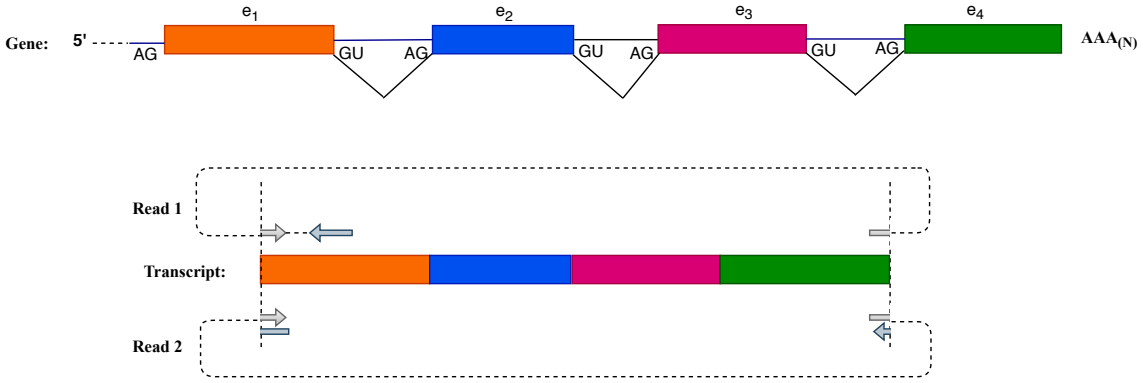


Figure 3.6: Two types of reads that are stored as partial mappings at the end of pseudo-alignment stage. In read 1 the mates are non-overlapping. The information of fully mapped mate and a partial mapping of the other mate is found and kept for further process. In read 2 the mates are overlapping and the breakpoint is happened on the overlap. So there are four pieces of partial mappings. If at least two of them are detected in this stage the information will be saved.

3.2 Circular RNA Detection

Due to the low abundance of circRNAs, usually less than 5% of the reads in an untreated RNA-Seq dataset remain as potential back-splice junction candidates. In order to detect circRNA breakpoints, for each candidate back-splice junction read, CircMiner examines if the unmapped suffix (or prefix) of the read can be aligned to upstream (or downstream) of the same transcript. CircMiner groups all the candidate back-splice junction reads from the same gene and processes them together as follows. For any gene with at least one back-splice junction read, it creates a hash table with smaller k'-mer size (8bp) for higher sensitivity. For simplicity, consider the unmapped substring of the read as U and mapped substring

as M. CircMiner extracts overlapping seeds (shifted by 3bp) of size k' from U, and chains the seeds. It then performs extension on the top scoring chains (5 top chains) as described in the previous section to obtain all the mappings of U. CircMiner keeps the mappings of U with minimum edit distance. If the total combined edit distance of the mapping of U and M is below the error threshold (e), CircMiner evaluates the mappings of U and M. If mappings of U and M forms a (co-)linear mapping of the read, it would be discarded. If mappings U and M forms a back-splice junction read, the breakpoint will be recorded for the next stage of clustering. Note that, if there are multiple mappings of U with same edit distance, CircMiner selects the formation that is consistent with splice sites of the transcript. Finally, the back-splice junction mappings will be clustered based on their breakpoints. For each circRNA, the genomic location of the detected breakpoints along with the supporting back-splice junction reads and the transcript and gene information is reported.

Chapter 4

Results

To evaluate the efficacy of the two modules in CircMiner, we designed two sets of experiments. First, we simulated datasets which contained reads that are generated from linear mRNAs. We used these datasets to calculate sensitivity and precision of pseudo-alignment module against the state-of-the-art splice aware mapper, STAR v2.6.0a [63]. In the second experiment, we used simulated as well as real datasets to compare circRNA detection module of CircMiner with several popular circRNA detection tools. All experiments were performed on a server running CentOS 7 equipped with 64 core CPU processors (Intel® Xeon® Gold 6130 CPU @ 2.10GHz) with 2 threads per core and 720 GB RAM. Furthermore, in all the experiments, Ensembl version GRCh38.90 is used as human genome reference and gene annotation.

4.1 Robustness of Linear Read Filtration

We first generated simulated RNA-Seq datasets containing linear mRNA reads to assess recall and precision of the pseudo-alignment and linear read filtration module. To simulate realistic datasets, we downloaded 3 RNA-Seq datasets from NCBI Sequence Reads Archive (accession numbers: SRR3146803, SRR3146859, and SRR3146914), and obtained their expression profile using kallisto v0.44.0 [28] based on Ensembl reference transcriptome version GRCh38.90. We excluded low-abundant transcripts from these expression profiles. Finally, we generated 5 paired-end datasets (2×100 bp reads with insert size of 350 ± 75 bp) per profile using ART Illumina simulator v2.5.8 [64] and Illumina HiSeq 2000 error model. Note that, the number of the reads simulated per isoform is taken from the expression profile. We included multiple datasets for each gene expression profile in order to increase diversity of junction reads such that we can test robustness of our splice-aware pseudo-alignment module.

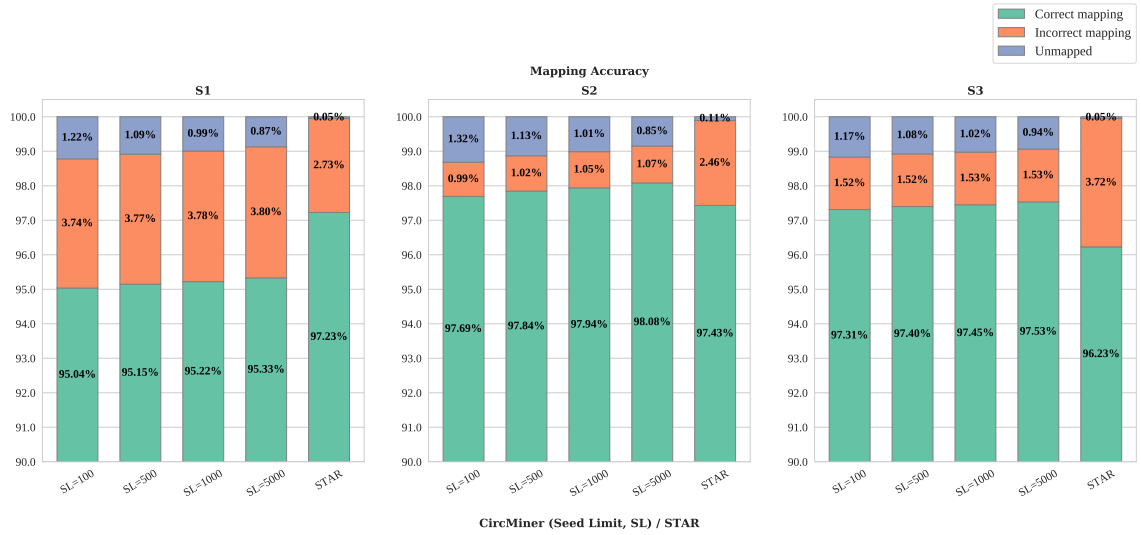


Figure 4.1: Performance comparison of CircMiner and STAR based on mapping accuracy on three simulated linear mRNA dataset.

4.1.1 Seed Limit Parameter Optimization

In these datasets, the origin of all the reads are known, thus we can evaluate the recall and precision of the pseudo-alignment module. As mentioned in Methods section, seed limit parameter (SL) has an effect on sensitivity and precision of CircMiner. In the first experiment, we ran CircMiner with 4 different values of SL including 100, 500, 1000, and 5000 and calculated (i) time, and (ii) precision and recall with respect to the ground truth. Figure 4.1 shows mapping accuracy of the 3 simulated samples (S1, S2, S3) using these seed limit values. The criteria for considering a read mapping or pseudo-alignment correct is that all junctions on the read be identical to the ground truth provided by ART. Though, a few base pair difference (10 bp) in the start and end positions are allowed in correct mappings which are potentially due to soft-clipping. For all datasets, the average value over 5 replicates is shown in the table and figure. The standard deviations of precision and recall across 5 replicates are less than 0.2% for all seed limit values, suggesting the robustness of our pseudo-alignment module in detecting coordinates of aligned reads.

Table 4.1: Read count and elapsed time comparison using four different seed limit (SL) values in CircMiner for three linear mRNA simulation datasets. Time format: (mm:ss).

Sample	Read Count	$SL = 100$	$SL = 500$	$SL = 1000$	$SL = 5000$
S1	23,460,612	20:20	21:06	22:08	26:46
S2	25,944,090	23:21	24:30	25:44	32:25
S3	33,138,471	24:11	24:57	25:30	29:18

Table 4.2: Mapping performance on three simulated datasets in terms of recall, precision, and F1 score.

Sample	SL	Recall (%)	Precision (%)	F1 Score
S1	100	95.04	96.21	0.96
	500	95.15	96.19	0.96
	1000	95.22	96.18	0.96
	5000	95.33	96.17	0.96
S2	100	97.69	98.99	0.98
	500	97.84	98.96	0.98
	1000	97.94	98.94	0.98
	5000	98.08	98.92	0.99
S3	100	97.31	98.46	0.98
	500	97.40	98.46	0.98
	1000	97.45	98.46	0.98
	5000	97.53	98.46	0.98

Table 4.1 and 4.2 show the required time, recall, precision, and F1 score of CircMiner with different values of SL on three simulated datasets. As SL increases so does the total running time of CircMiner, and although it improves recall and and F1 score, the effect is not considerable. Besides, precision drops slightly in some cases when greater SL values are selected. Thus, we choose to use $SL = 500$ as default value in CircMiner since it has the highest jump in recall and maintains a good balance between speed and mapping accuracy. This value is used in the rest of experiments explained in the results section in this thesis.

4.1.2 Performance Comparison Against STAR

We also compared CircMiner's results with the state-of-the-art splice-aware RNA-Seq mapper, STAR. Table 4.3 depicts the precision, recall, and F1 score of STAR versus CircMiner on all three simulated datasets. Furthermore, time and memory for CircMiner and STAR are reported in table 4.3 shows that CircMiner has $\sim\frac{1}{2}$ time and $\sim\frac{2}{5}$ memory consumption compared to STAR while the accuracy did not change significantly. These results suggest that even though CircMiner is not designed to be a full RNA-Seq mapper, it has comparable results to STAR. Based on Figure 4.1, our pseudo-alignment module achieves comparable ratio of mapped reads (99%) in all three datasets. As demonstrated in Table 4.3, in two out of three datasets, CircMiner outperforms STAR while STAR performs better in one of the datasets. Lower accuracy in S1 can be explained by multi-mapping. Since CircMiner does not have all the features of a typical mapper and only reports one mapping location per read, it is possible that for multi-mapper reads CircMiner report one of the secondary mappings that have identical error and soft-clip values. To test the multi-mapping hypothesis, we compared our pseudo-alignment results with every multi-mapping location reported by STAR. The new recall, precision and F1 score are shown in Table 4.4. It showed that the pseudo-alignment module's recall and precision in S1 go up more than 2.5% when we also consider alternate locations of multi-mapper reads as true positives. This indicates that for a mappable read, even though CircMiner might not perfectly locate its true origin, in many cases it already provided the best result other splice-aware mapper can achieve. The feature for reporting mapping locations is disabled in CircMiner by default because it increases running time and it is not needed when the user is only interested in circRNA detection. However, the user can enable this feature.

Table 4.3: Performance and time comparison between CircMiner ($SL = 500$) and STAR on linear simulation datasets.

Dataset		CircMiner	STAR
S1	Recall[a] (%)	95.15	**97.23**
	Precision[b] (%)	96.19	**97.27**
	F1 Score[c]	0.96	**0.97**
	Time[d] (mm:ss)	**21:06**	41:01
	Memory[d] (GB)	**10.31**	27.76
S2	Recall (%)	**97.84**	97.43
	Precision (%)	**98.96**	97.53
	F1 Score	**0.98**	0.97
	Time (mm:ss)	**24:30**	44:01
	Memory (GB)	**10.31**	27.76
S3	Recall (%)	**97.40**	96.23
	Precision (%)	**98.46**	96.28
	F1 Score	**0.98**	0.96
	Time (mm:ss)	**24:57**	54:52
	Memory (GB)	**10.31**	27.76

A read is considered to be correctly mapped if the mapping location of the endpoints are within a 10 bp vicinity of the correct alignment locus.
a Recall is defined as number of correctly mapped reads / read count.
b Precision is defined as number of correctly mapped reads / total number of mapped reads.
c F1 score is calculated as $2.\frac{precision.recall}{precision+recall}$.
d running time and memory usage are measured using time -v Unix command.

Table 4.4: Performance comparison between CircMiner ($SL = 500$) and STAR on linear simulation datasets with also accepting multi-mapped reads reported by CircMiner.

Dataset		CircMiner	STAR
S1	Recall (%)	**97.82**	97.23
	Precision (%)	**98.90**	97.27
	F1 Score	**0.98**	0.97
S2	Recall (%)	**98.39**	97.43
	Precision (%)	**99.52**	97.53
	F1 Score	**0.99**	0.97
S3	Recall (%)	**98.40**	96.23
	Precision (%)	**99.47**	96.28
	F1 Score	**0.99**	0.96

The most important task of this module is to find and keep the reads that can show signals for back-splice junction. Note that, these datasets do not contain any circular RNA reads, and ideally all simulated reads should be filtered out by the pseudo-alignment module. Circular RNA detection tools that rely on STAR use the chimeric reads that it reports. Thus, we extracted and calculated the portion of the reads reported by both CircMiner and STAR. As it can be seen in Table 4.5, although we do not expect any circRNAs in the

Table 4.5: Ratio of reported chimeric reads by CircMiner ($SL = 500$) and STAR. The values are shown as percentage(%).

Sample	CircMiner	STAR
S1	0.27	0.01
S2	0.25	0.01
S3	0.32	0.01

simulated datasets, both tools report $\leq 0.32\%$ chimeric reads for further processing which is a small portion. The proportion of chimeric reads reported by both methods are pretty small, but a bit higher in CircMiner. The reason behind this is that CircMiner is more sensitive on candidate back-splice junction read detection, thus it keeps any read with a partial mapping on a single transcript and do not allow large soft-clips. However, some of the detected chimeric reads by CircMiner may be result of other transcriptome events such as exons skipping. This is not an issue for STAR because it tries to address such events during the full mapping.

4.2 Simulation of Circular RNA Data

In order to simulate circularRNA reads, we used a modified version of CIRI-simulator [17] that accepts a set of back splice junctions, a gene annotation file, and a reference as input and generates a number of RNA-Seq reads that supports such a given circular RNAs and their juctions. To have realistic circular RNA back splice junctions, we utilized CIRCpedia v2 [65], a circRNA annotation database of various tissues and cell types from different species. In order to evaluate the detection module, we pass the following inputs to ciriSimul : (i) 110,128 validated human back-splice junctions that are consistent with Ensembl GRCh38.90 annotation from CIRCpedia; (ii) 1000 randomly selected back splice junctions form (i). ciriSimul generated two paired end datasets (C_1 and C_2) with 4,585,051 and 39,964 reads (2×101bp reads with insert size 350 ± 75) with 1% sequencing error rate. We confirmed that each back splice junction is supported by at least two reads in both of the datasets. These datasets are denoted *positive controls* because they only contain reads from circularRNA.

Furthermore, to better mimic real RNA-Seq datasets where both linear and circRNA reads exist in one dataset, we diluted each simulated dataset above with a *background set*

of simulated linear mRNA reads. The dilution ratios for sample DC_1 and DC_2 are 1:19 and 1:99, corresponding to 5% and 1% of circRNA reads in the diluted samples.

Background sets for both samples were generated using ART v2.5.8 based on gene expression profile of SRR3146803 as 2×101 bp reads of insert size 350 ± 75 under Illumina HiSeq 2000 error model. We combined background sets and positive controls to obtain final diluted datasets.

Based on positive controls and diluted datasets, we benchmarked circRNA detection performances of CircMiner, CIRCexplorer [19], CIRI2 [18], KNIFE [21], SEGEMEHL [24], and CircMarker [25]. CIRCexplorer, CIRI, and KNIFE are considered as best-performing circRNA detection tools that work on top of existing mappers [27]. SEGEMEHL is a multi-split mapper supporting back-splice junction detection, and recently published CircMarker is one of the first k-mer counting circRNA detection methods.

Figure 4.2 and 4.3 illustrate the precision and recall of CircMiner and other selected methods on simulated sample1 (SS1) and simulated sample2 (SS2) respectively. For each method, an arrow is connecting the precision and recall of positive control to diluted dataset. SEGEMEHL, KNIFE, CircMarker, and CIRI had the most significant drops in precision when diluted sample was used. This suggests that these methods are more likely to have high false positive rate in real datasets since both circRNA and mRNA are present in real samples. On the other hand, CircMiner and CIRCexplorer provided very robust results in simulation as well as low false positive rate which is shown in Figure 4.4a and 4.4b. Meanwhile, CircMiner was able to detect the most number of circRNAs and had highest recall. Among the robust methods, CircMiner had the higher recall which is more than 11% higher than CIRCexplorer's. This suggests that CircMiner is the best-performing method on the simulated data and is able to detect the most number of circRNAs with high precision.

In terms of running time, CircMiner is performing better compared to the state-of-the-art methods. Figure 4.5 demonstrates running time comparison on diluted datasets (DC1 and DC2). It shows that CircMiner is consuming a fraction of time other circRNA detection tools typically use. Besides, CircMiner requires less than 11 GB of memory which is the smallest memory consumption among all the tested methods. In the smaller dataset (DC2), CircMarker and CIRI have the lowest memory consumption, however; their memory usage rises in high read depth datasets. In high read depth dataset (DC1), CircMiner consumes the lowest amount of memory (10.3 GB) which makes it possible to run on a personal computer. The index construction stage of CircMiner is done in around 15 minutes for human genome on a single CPU. However, this step is done once and the generated index can be reused for many samples. The most time consuming stage of CircMiner is the pseudo-alignment stage and the final detection stage is done in a few minutes on the simulation sets as demonstrated in Table 4.6. The most time consuming sections of the pseudo-alignment stage are extension and seed location extraction with using almost 30% and 20% of the time respectively.

Table 4.6: Profile results of CircMiner on simulation dataset DC2.

Stage/Function	Time (mm:ss)
Index building	15:02
Index loading	1:35
Seed location extraction	1:14
Chaining	0:37
Alignment	2:21
Rest	0:36
circRNA Detection	1:12

Timing and profiling is done using GNU time and GNU profile commands.

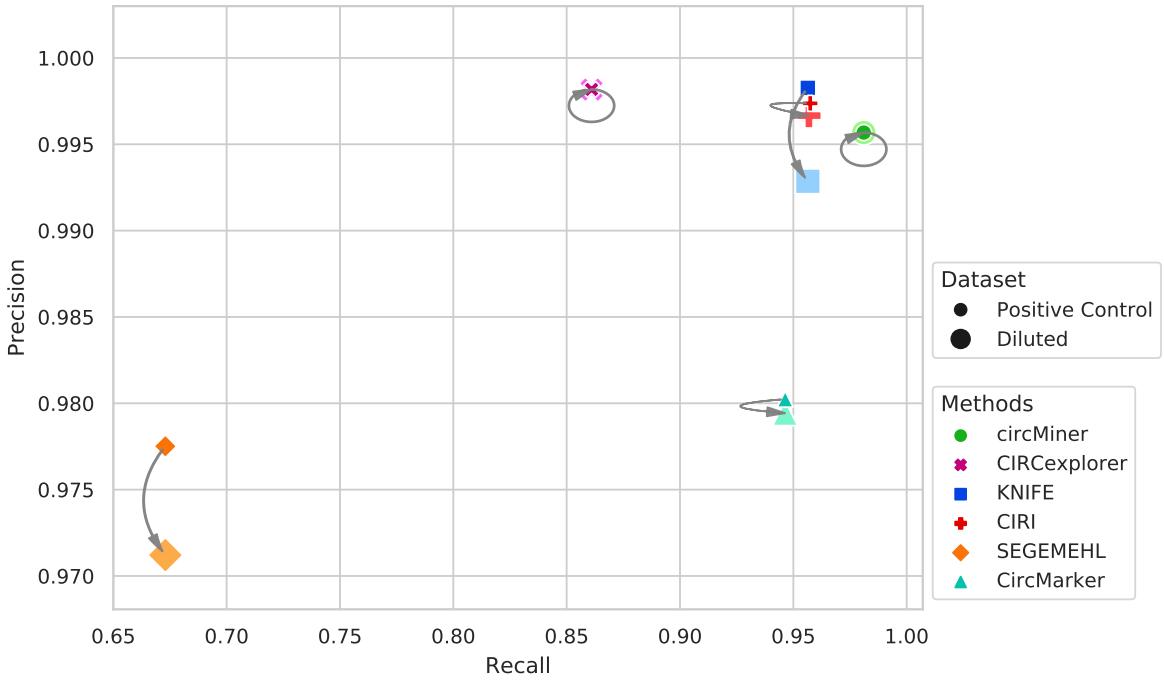


Figure 4.2: Precision and recall on simulated dataset 1 (SS1) comparing CircMiner and five other state-of-the-art circRNA detection methods.

Table 4.7: Circular RNA and read count in simulation datasets.

	Num. circRNAs		Num. reads	
	Positive control	Diluted	Positive control	Diluted
SS1	110,128	110,128	4,585,051	92,261,457
SS2	1,000	1,000	39,964	4,078,974

The number of embedded circRNAs and reads per positive control and diluted sample for each of the simulation sets.

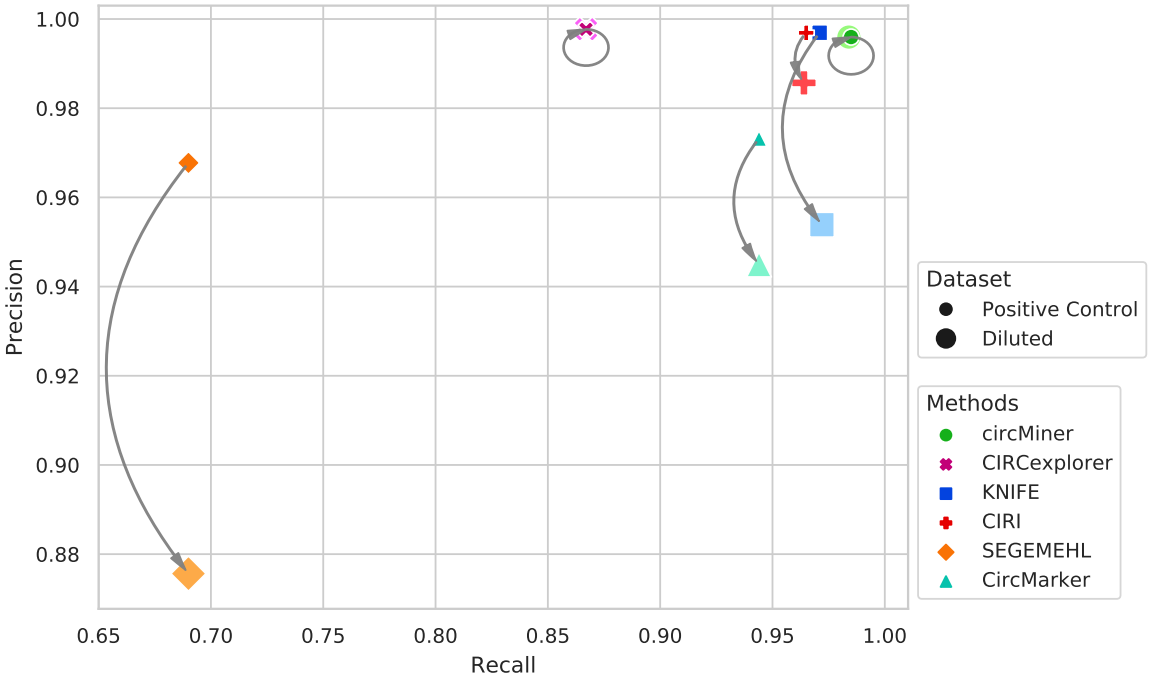
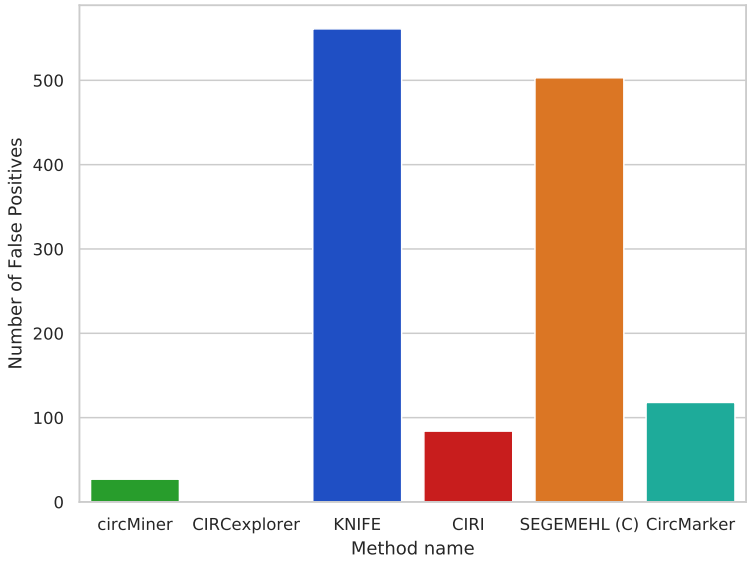
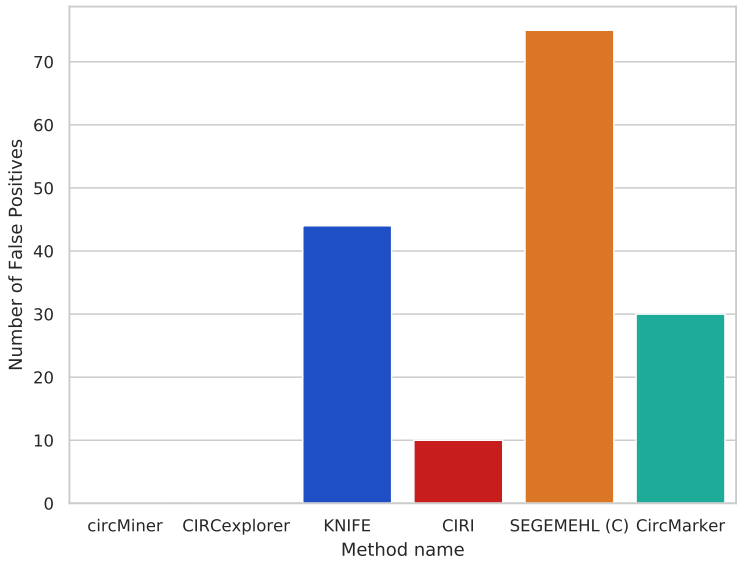


Figure 4.3: Precision and recall on simulated dataset 2 (SS2) comparing CircMiner and five other state-of-the-art circRNA detection methods.

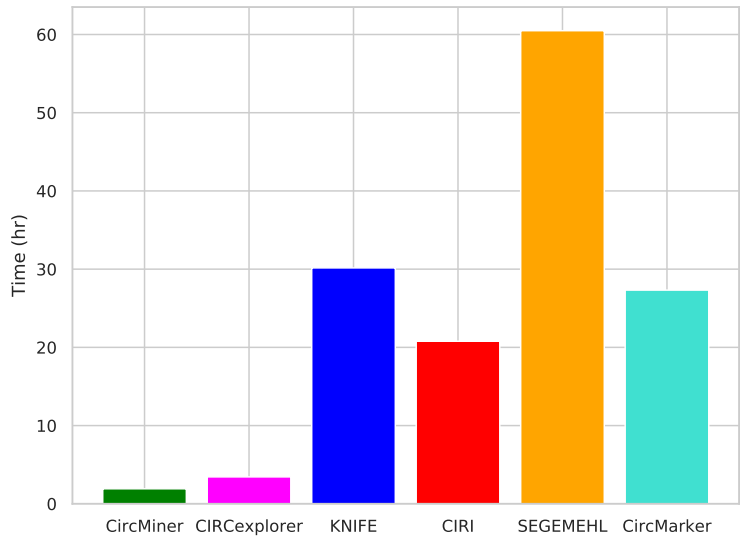


(a) Number of false positives detected by CircMiner and other selected methods on the background dataset of SS1 that only contains simulated linear mRNA reads.

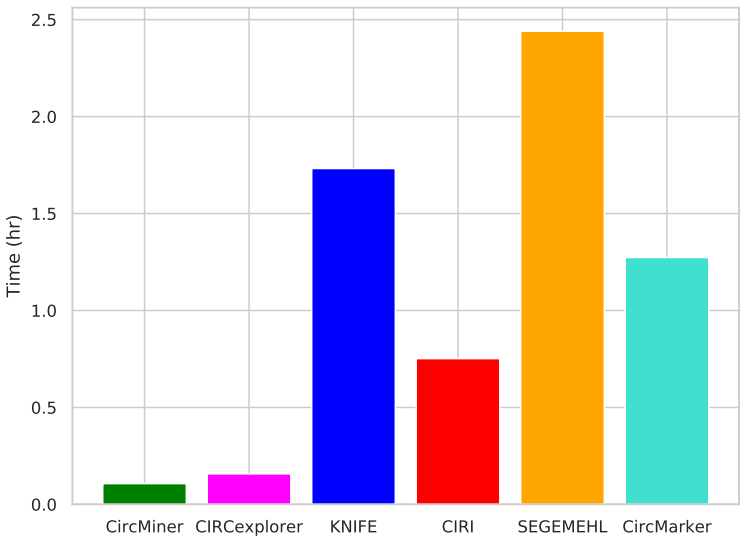


(b) Number of false positives detected by CircMiner and other selected methods on the background dataset of SS2 that only contains simulated linear mRNA reads.

Figure 4.4: False positives detected on background datasets (contain only linear simulated mRNA reads). CIRCexplorer and CircMiner have the lowest false positive rates among the tested tools.



(a) Time comparison of different tools on diluted sample DC1.



(b) Running time comparison of different tools on diluted sample DC2.

Figure 4.5: Running time comparison of CircMiner with 5 existing circRNA detection tools on diluted simulated samples.

Table 4.8: Information about 2 pairs of 2×101 bp RNA-Seq data from HeLa and Hs68 cell lines. SRA accession numbers and read count for RNaseR- (sample before treatment) and RNaseR+ (sample after treatment) is provided.

	RNaseR-		RNaseR+	
	SRA accession	# Reads	SRA accession	# Reads
HeLa	SRR1637089 SRR1637090	80,618,760	SRR1636985 SRR1636986	36,815,458
Hs68	SRR444975	206,362,733	SRR445016	199,922,486

4.3 Real Data Analysis

Our experiment on real data contains two pairs of 2x101bp Illumina sequencing RNA-seq data from cell lines HeLa and Hs68 from previous studies [4, 17]. For each cell line, one dataset is a typical RNA-seq data sequenced using ribosomal RNA (rRNA) depletion protocol, and the other is a treated sample where circRNAs are enriched after most linear RNA species are digested and removed by RNase R enzyme. We denote the former by *HeLa RNaseR-* and *Hs68 RNaseR-*, and the latter *HeLa RNaseR+* and *He68 RNaseR+*, respectively. Unlike simulation datasets, there is no ground truth of expressed circRNAs for these two cell lines. Therefore our evaluation for each tool is mainly based on the agreement rate of detected back-splice junctions before and after enrichment for each cell line, considering most circRNAs expressed in a cell line are expected to remain in its treated sample.

For each tool, a detected circRNA in the RNaseR- dataset is considered as a true positive or a *not-depleted* call if its *normalized support*– the number of back-splice junction reads divided by the total number of reads in the dataset– is not decreased in the treated RNaseR+ dataset. Moreover, we consider a circRNA as a high-confident or enriched call if its normalized support is at last 5-fold increased in the treated RNaseR+ dataset compared to the RNaseR- dataset.

We calculated the percentage of high-confident calls in top 10 and top 100 abundant calls in the untreated RNaseR- datasets for each tool. Even though true circRNAs are not always enriched and detectable in the treated sample, we expect a robust tool to report more high-confident calls in its top results. As in the previous simulation experiment, only the reported circRNAs with at least 2 supporting back-splice junction reads are considered in the comparison, and we eliminate those circRNAs which are not consistent to back-splice junction sites in the reference transcriptome. Table 4.9 and Figure 4.6 show the results and running time of each tool on Hs68 cell line sample. CircMiner not only provides the highest agreement rate between two datasets, but also detect most number of high-confident calls among its top 100 abundant predictions in RNaseR- dataset. The detail results on both cell line samples are depicted in Table 4.9. It shows that CircMiner is performing best on Hs68 and for HeLa sample, CIRCexplorer and CircMiner report close results.

Figure 4.6 demonstrates that CircMiner also provides significant advantage in terms of running time. Both mapping and circRNA detection time is considered for all the tools. This means that for the mapping-based circRNA detection tools, running time of mapping stage is also included in the reported time.

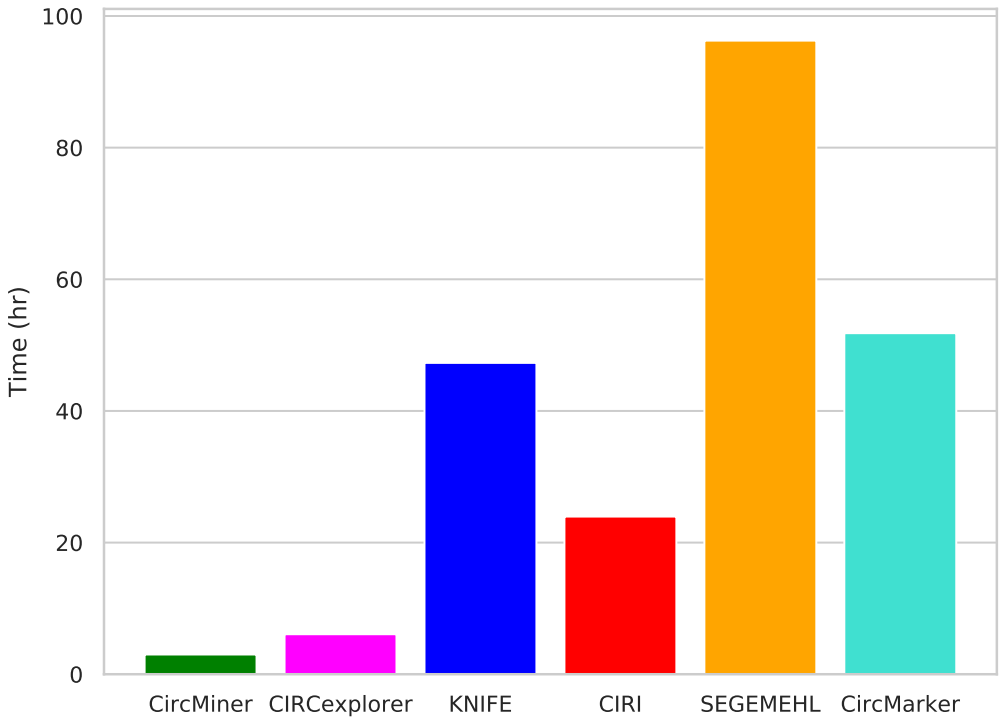


Figure 4.6: Time comparison for Hs68 sample among six different circRNA detection tools.

Table 4.9: Detailed comparison of circRNA detection tools on Hs68 and HeLa datasets in terms of not depleted and enriched circRNAs.

Hs68	RNaseR-	RNaseR+	Not depleted	Not depleted (%)	Top 10 not depleted (enriched)	Top 100 not depleted (enriched)
CircMiner	3,794	22,435	2,626	69.21	7(5)	79(62)
CIRCexplorer	2,919	21,879	1,990	68.17	8(5)	78(58)
KNIFE	4,120	23,014	2,798	67.91	7(5)	77(61)
CIRI	6,074	33,748	3,829	63.04	6(5)	76(61)
SEGEMEHL	67,650	150,788	11,000	16.26	3(2)	21(15)
CircMarker	5,789	24,435	3,159	54.57	5(3)	68(45)
HeLa	RNaseR-	RNaseR+	Not depleted	Not depleted (%)	Top 10 not depleted (enriched)	Top 100 not depleted (enriched)
CircMiner	4,560	6,025	2,202	48.29	9(1)	73(9)
CIRCexplorer	2,952	4,748	1,485	50.30	9(0)	75(12)
KNIFE	5,068	6,157	2,336	46.09	8(1)	76(8)
CIRI	7,303	8,949	3,377	46.24	7(0)	73(7)
SEGEMEHL	14,479	21,403	2,561	17.69	1(0)	30(1)
CircMarker	7,332	6,996	2,629	35.86	0(0)	47(2)

Chapter 5

Conclusion

This thesis presents rapid and efficient algorithms for pseudo-alignment and circular RNA (circRNA) detection which are implemented as a stand-alone package. We first presented an overview of available splice-aware RNA-seq mappers, circRNA detection tools, and pseudo-alignment methods. Then CircMiner was introduced, a highly accurate and sensitive tool for circRNA detection from RNA-seq data. We demonstrated that CircMiner can efficiently and rapidly detect circRNAs in a fraction of time compared to other stete-of-the-art tools. Unlike many other circRNA detection tools, CircMiner is not an extension on top of an existing splice-aware mapper, but was developed as a stand-alone method. This helped us to build a rapid and efficient algorithm that is able to pseudo-align 1 million 2x100 bp RNA-seq reads to the human genome in a minute on average. The mapping accuracy of the reads is comparable with the state-of-the-art splice-aware mappers such as STAR. The detection step is also quick since only a small portion of the reads will remain for further investigation of back-splice junctions. This step is usually done in a few minutes for large datasets. CircMiner achieved the best recall and F1-score on simulated samples as well as the best performance in terms of significant circRNA calling from RNase R treated samples in concordance to non-treated sample from the same cell-line.

5.1 Future Directions

Although CircMiner performs well in back-splice junction detection, it is not designed for full-length circRNA re-construction. Circular RNAs usually contain a few exons, thus they are usually short sequences that can be fully captured by long reads. Designing a method that uses long reads for full-length construction of circRNAs could be impactful for further analysis of circRNAs in patient data especially cancer patients with heterogeneity. In addition, it can be used for accurate circRNA quantification which is still a challenging problem.

Another possible future direction is to add a new fusion detection module on top of pseudo-alignment module for further processing fusion reads that are detected during

pseudo-alignment stage. Furthermore, it has been shown that fusion circRNAs exist while there is no computational method available to detect such events to date. Thus, developing a novel method to address this problem could form another future direction.

CircRNAs are found to be abundant and stable in human blood exosomes. They have longer half-life in cell-free samples (such as blood and urine) because of their stable circular structure, creating potential for their use as cancer biomarkers in patient samples from noninvasive sources. However, cell-free sequencing data have specific properties which are not necessarily similar to typical DNA or RNA sequencing data from tissue samples. These reasons suggest another possible future work to explore which is detecting circRNAs from cell-free RNA sequencing data.

Finally, we hope that CircMiner becomes an even more powerful, easy-to-use, accurate and faster pseudo-alignment and circRNA detection package, with important applications in the development of accurate and efficient RNA-seq analysis pipelines that include circRNA analysis in the future for different types of diseases such as cancer.

Bibliography

[1] Marcel Margulies, Michael Egholm, William E. Altman, Said Attiya, Joel S. Bader, Lisa A. Bemben, Jan Berka, Michael S. Braverman, Yi-Ju Chen, Zhoutao Chen, Scott B. Dewell, Lei Du, Joseph M. Fierro, Xavier V. Gomes, Brian C. Godwin, Wen He, Scott Helgesen, Chun He Ho, Gerard P. Irzyk, Szilveszter C. Jando, Maria L. I. Alenquer, Thomas P. Jarvie, Kshama B. Jirage, Jong-Bum Kim, James R. Knight, Janna R. Lanza, John H. Leamon, Steven M. Lefkowitz, Ming Lei, Jing Li, Kenton L. Lohman, Hong Lu, Vinod B. Makhijani, Keith E. McDade, Michael P. McKenna, Eugene W. Myers, Elizabeth Nickerson, John R. Nobile, Ramona Plant, Bernard P. Puc, Michael T. Ronan, George T. Roth, Gary J. Sarkis, Jan Fredrik Simons, John W. Simpson, Maithreyan Srinivasan, Karrie R. Tartaro, Alexander Tomasz, Kari A. Vogt, Greg A. Volkmer, Shally H. Wang, Yong Wang, Michael P. Weiner, Pengguang Yu, Richard F. Begley, and Jonathan M. Rothberg. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, July 2005.

[2] M. T. Hsu and M. Coca-Prados. Electron microscopic evidence for the circular form of RNA in the cytoplasm of eukaryotic cells. *Nature*, 280(5720):339–340, Jul 1979.

[3] J. Salzman, C. Gawad, P. L. Wang, N. Lacayo, and P. O. Brown. Circular RNAs are the predominant transcript isoform from hundreds of human genes in diverse cell types. *PLoS ONE*, 7(2):e30733, 2012.

[4] W. R. Jeck, J. A. Sorrentino, K. Wang, M. K. Slevin, C. E. Burd, J. Liu, W. F. Marzluff, and N. E. Sharpless. Circular RNAs are abundant, conserved, and associated with ALU repeats. *RNA*, 19(2):141–157, 2013.

[5] P. L. Wang, Y. Bao, M. C. Yee, S. P. Barrett, G. J. Hogan, M. N. Olsen, J. R. Dinneny, P. O. Brown, and J. Salzman. Circular RNA is expressed across the eukaryotic tree of life. *PLoS ONE*, 9(6):e90859, 2014.

[6] S. Starke, I. Jost, O. Rossbach, T. Schneider, S. Schreiner, L. H. Hung, and A. Bindereif. Exon circularization requires canonical splice signals. *Cell Rep*, 10(1):103–111, Jan 2015.

[7] X. O. Zhang, H. B. Wang, Y. Zhang, X. Lu, L. L. Chen, and L. Yang. Complementary sequence-mediated exon circularization. *Cell*, 159(1):134–147, Sep 2014.

[8] Y. Zhang, W. Xue, X. Li, J. Zhang, S. Chen, J. L. Zhang, L. Yang, and L. L. Chen. The Biogenesis of Nascent Circular RNAs. *Cell Rep*, 15(3):611–624, Apr 2016.

[9] D. Liang and J. E. Wilusz. Short intronic repeat sequences facilitate circular RNA production. *Genes Dev.*, 28(20):2233–2247, Oct 2014.

[10] M. C. Kramer, D. Liang, D. C. Tatomer, B. Gold, Z. M. March, S. Cherry, and J. E. Wilusz. Combinatorial control of Drosophila circular RNA expression by intronic repeats, hnRNPs, and SR proteins. *Genes Dev.*, 29(20):2168–2182, Oct 2015.

[11] J. U. Guo, V. Agarwal, H. Guo, and D. P. Bartel. Expanded identification and characterization of mammalian circular RNAs. *Genome Biol.*, 15(7):409, Jul 2014.

[12] R. Ashwal-Fluss, M. Meyer, N. R. Pamudurti, A. Ivanov, O. Bartok, M. Hanan, N. Evantal, S. Memczak, N. Rajewsky, and S. Kadener. circRNA biogenesis competes with pre-mRNA splicing. *Mol. Cell*, 56(1):55–66, Oct 2014.

[13] L S Kristensen, T B Hansen, M T Venø, and J Kjems. Circular RNAs in cancer: opportunities and challenges in the field. *Oncogene*, 37(5):555–565, oct 2017.

[14] Jlenia Guarnerio, Marco Bezzi, Jong Cheol Jeong, Stella V. Paffenholz, Kelsey Berry, Matteo M. Naldini, Francesco Lo-Coco, Yvonne Tay, Andrew H. Beck, and Pier Paolo Pandolfi. Oncogenic Role of Fusion-circRNAs Derived from Cancer-Associated Chromosomal Translocations. *Cell*, 165(2):289–302, Apr 2016.

[15] Sujun Chen, Vincent Huang, Xin Xu, Julie Livingstone, Fraser Soares, Jouhyun Jeon, Yong Zeng, Junjie Tony Hua, Jessica Petricca, Haiyang Guo, Miranda Wang, Fouad Yousif, Yuzhe Zhang, Nilgun Donmez, Musaddeque Ahmed, Stas Volik, Anna Lapuk, Melvin L.K. Chua, Lawrence E. Heisler, Adrien Foucal, Natalie S. Fox, Michael Fraser, Vinayak Bhandari, Yu-Jia Shiah, Jiansheng Guan, Jixi Li, Michèle Orain, Valérie Picard, Hélène Hovington, Alain Bergeron, Louis Lacombe, Yves Fradet, Bernard Têtu, Stanley Liu, Felix Feng, Xue Wu, Yang W. Shao, Malgorzata A. Komor, Cenk Sahinalp, Colin Collins, Youri Hoogstrate, Mark de Jong, Remond J.A. Fijneman, Teng Fei, Guido Jenster, Theodorus van der Kwast, Robert G. Bristow, Paul C. Boutros, and Housheng Hansen He. Widespread and functional RNA circularization in localized prostate cancer. *Cell*, 176(4):831–843.e22, feb 2019.

[16] Linda Szabo and Julia Salzman. Detecting circular RNAs: bioinformatic and experimental challenges. *Nat. Rev. Genet.*, 17(11):679–692, 10 2016.

[17] Yuan Gao, Jinfeng Wang, and Fangqing Zhao. CIRI: an efficient and unbiased algorithm for de novo circular RNA identification. *Genome Biology*, 16(1):4, 2015.

[18] Yuan Gao, Jinyang Zhang, and Fangqing Zhao. Circular RNA identification based on multiple seed matching. *Briefings in Bioinformatics*, 19(5):803–810, feb 2017.

[19] Xiao-Ou Zhang, Hai-Bin Wang, Yang Zhang, Xuhua Lu, Ling-Ling Chen, and Li Yang. Complementary sequence-mediated exon circularization. *Cell*, 159(1):134–147, sep 2014.

[20] Xiao-Ou Zhang, Rui Dong, Yang Zhang, Jia-Lin Zhang, Zheng Luo, Jun Zhang, Ling-Ling Chen, and Li Yang. Diverse alternative back-splicing and alternative splicing landscape of circular RNAs. *Genome Research*, 26(9):1277–1287, jun 2016.

[21] Linda Szabo, Robert Morey, Nathan J. Palpant, Peter L. Wang, Nastaran Afari, Chuan Jiang, Mana M. Parast, Charles E. Murry, Louise C. Laurent, and Julia Salzman. Statistically based splicing detection reveals neural enrichment and tissue-specific induc-

tion of circular RNA during human fetal development. *Genome Biology*, 16(1), jun 2015.

[22] Jakub O. Westholm, Pedro Miura, Sara Olson, Sol Shenker, Brian Joseph, Piero Sanfilippo, Susan E. Celniker, Brenton R. Graveley, and Eric C. Lai. Genome-wide analysis of drosophila circular RNAs reveals their structural and sequence properties and age-dependent neural accumulation. *Cell Reports*, 9(5):1966–1980, dec 2014.

[23] Kai Wang, Darshan Singh, Zheng Zeng, Stephen J. Coleman, Yan Huang, Gleb L. Savich, Xiaping He, Piotr Mieczkowski, Sara A. Grimm, Charles M. Perou, James N. MacLeod, Derek Y. Chiang, Jan F. Prins, and Jinze Liu. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18):e178–e178, aug 2010.

[24] Steve Hoffmann, Christian Otto, Gero Doose, Andrea Tanzer, David Langenberger, Sabina Christ, Manfred Kunz, Lesca M Holdt, Daniel Teupser, Jörg Hackermüller, and Peter F Stadler. A multi-split mapping algorithm for circular RNA, splicing, trans-splicing and fusion detection. *Genome Biology*, 15(2):R34, 2014.

[25] Xin Li, Chong Chu, Jingwen Pei, Ion Măndoiu, and Yufeng Wu. CircMarker: a fast and accurate algorithm for circular RNA detection. *BMC Genomics*, 19(S6), aug 2018.

[26] T. B. Hansen, M. T. VenÃÿ, C. K. Damgaard, and J. Kjems. Comparison of circular rna prediction tools. *Nucleic Acids Res.*, 44(6):e58, Apr 2016.

[27] X. Zeng, W. Lin, M. Guo, and Q. Zou. A comprehensive overview and evaluation of circular RNA detection tools. *PLoS Comput. Biol.*, 13(6):e1005420, Jun 2017.

[28] Nicolas L Bray, Harold Pimentel, Páll Melsted, and Lior Pachter. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, 34(5):525–527, apr 2016.

[29] Avi Srivastava, Hirak Sarkar, Nitish Gupta, and Rob Patro. RapMap: a rapid, sensitive and accurate tool for mapping RNA-seq reads to transcriptomes. *Bioinformatics*, 32(12):i192–i200, jun 2016.

[30] F. Sanger, S. Nicklen, A. R. Coulson, F. Sanger, S. Nicklen, and A. R. Coulsen. DNA sequencing with chain-terminating inhibitors. 1977. *Biotechnology*, 24:104–108, 1992.

[31] Elaine R. Mardis. A decade's perspective on DNA sequencing technology. *Nature*, 470(7333):198–203, feb 2011.

[32] Jason A. Reuter, Damek V. Spacek, and Michael P. Snyder. High-throughput sequencing technologies. *Molecular Cell*, 58(4):586–597, may 2015.

[33] M. Fedurco. BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. *Nucleic Acids Research*, 34(3):e22–e22, feb 2006.

[34] Mehdi Kchouk, Jean Francois Gibrat, and Mourad Elloumi. Generations of sequencing technologies: From first to next generation. *Biology and Medicine*, 09(03), 2017.

[35] Melanie Schirmer, Rosalinda D'Amore, Umer Z. Ijaz, Neil Hall, and Christopher Quince. Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. *BMC Bioinformatics*, 17(1), mar 2016.

[36] Jonathan M. Rothberg, Wolfgang Hinz, Todd M. Rearick, Jonathan Schultz, William Mileski, Mel Davey, John H. Leamon, Kim Johnson, Mark J. Milgrew, Matthew Edwards, Jeremy Hoon, Jan F. Simons, David Marran, Jason W. Myers, John F. Davidson, Annika Branting, John R. Nobile, Bernard P. Puc, David Light, Travis A. Clark, Martin Huber, Jeffrey T. Branciforte, Isaac B. Stoner, Simon E. Cawley, Michael Lyons, Yutao Fu, Nils Homer, Marina Sedova, Xin Miao, Brian Reed, Jeffrey Sabina, Erika Feierstein, Michelle Schorn, Mohammad Alanjary, Eileen Dimalanta, Devin Dressman, Rachel Kasinskas, Tanya Sokolsky, Jacqueline A. Fidanza, Eugeni Namsaraev, Kevin J. McKernan, Alan Williams, G. Thomas Roth, and James Bustillo. An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348–352, jul 2011.

[37] Murim Choi, Ute I. Scholl, Weizhen Ji, Tiewen Liu, Irina R. Tikhonova, Paul Zumbo, Ahmet Nayir, AyÈŹin Bakkaloğlu, Seza Özen, Sami Sanjad, Carol Nelson-Williams, Anita Farhi, Shrikant Mane, and Richard P. Lifton. Genetic diagnosis by whole exome capture and massively parallel DNA sequencing. *Proceedings of the National Academy of Sciences*, 106(45):19096–19101, oct 2009.

[38] Marc Sultan, Vyacheslav Amstislavskiy, Thomas Risch, Moritz Schuette, Simon Dökel, Meryem Ralser, Daniela Balzereit, Hans Lehrach, and Marie-Laure Yaspo. Influence of RNA extraction methods and library selection schemes on RNA-seq data. *BMC Genomics*, 15(1):675, 2014.

[39] James Dominic Mills, Yoshihiro Kawahara, and Michael Janitz. Strand-specific RNA-seq provides greater resolution of transcriptome profiling. *Current Genomics*, 14(3):173–181, apr 2013.

[40] Sk Tofajjen Hossain, Arun Malhotra, and Murray P. Deutscher. How RNase r degrades structured RNA. *Journal of Biological Chemistry*, 291(15):7877–7887, feb 2016.

[41] Eric T. Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F. Kingsmore, Gary P. Schroth, and Christopher B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, nov 2008.

[42] Daehwan Kim, Ben Langmead, and Steven L Salzberg. HISAT: a fast spliced aligner with low memory requirements. *Nature Methods*, 12(4):357–360, mar 2015.

[43] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. Technical report, 1994.

[44] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE Comput. Soc, 2000.

[45] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, oct 2012.

[46] T. D. Wu and S. Nacu. Fast and SNP-tolerant detection of complex variants and splicing in short reads. *Bioinformatics*, 26(7):873–881, feb 2010.

[47] S. T. Sherry. dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311, January 2001.

[48] Barmak Modrek and Christopher Lee. A genomic view of alternative splicing. *Nature Genetics*, 30(1):13–19, jan 2002.

[49] Qun Pan, Ofer Shai, Leo J Lee, Brendan J Frey, and Benjamin J Blencowe. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40(12):1413–1415, nov 2008.

[50] A Sveen, S Kilpinen, A Ruusulehto, R A Lothe, and R I Skotheim. Aberrant RNA splicing in cancer; expression changes and driver mutations of splicing factor genes. *Oncogene*, 35(19):2413–2427, aug 2015.

[51] Michael Sammeth. Complete alternative splicing events are bubbles in splicing graphs. *Journal of Computational Biology*, 16(8):1117–1140, August 2009.

[52] Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, 28(5):511–515, may 2010.

[53] Scott A. Tomlins, Bharathi Laxman, Saravana M. Dhanasekaran, Beth E. Helgeson, Xuhong Cao, David S. Morris, Anjana Menon, Xiaojun Jing, Qi Cao, Bo Han, Jindan Yu, Lei Wang, James E. Montie, Mark A. Rubin, Kenneth J. Pienta, Diane Roulston, Rajal B. Shah, Sooryanarayana Varambally, Rohit Mehra, and Arul M. Chinnaiyan. Distinct classes of chromosomal rearrangements create oncogenic ETS gene fusions in prostate cancer. *Nature*, 448(7153):595–599, aug 2007.

[54] Fredrik Mertens, Bertil Johansson, Thoas Fioretos, and Felix Mitelman. The emerging complexity of gene fusions in cancer. *Nature Reviews Cancer*, 15(6):371–381, jun 2015.

[55] Daehwan Kim and Steven L Salzberg. TopHat-fusion: an algorithm for discovery of novel fusion transcripts. *Genome Biology*, 12(8):R72, 2011.

[56] Andrew McPherson, Fereydoun Hormozdiari, Abdalnasser Zayed, Ryan Giuliany, Gavin Ha, Mark G. F. Sun, Malachi Griffith, Alireza Heravi Moussavi, Janine Senz, Nataliya Melnyk, Marina Pacheco, Marco A. Marra, Martin Hirst, Torsten O. Nielsen, S. Cenk Sahinalp, David Huntsman, and Sohrab P. Shah. deFuse: An algorithm for gene fusion discovery in tumor RNA-seq data. *PLoS Computational Biology*, 7(5):e1001138, may 2011.

[57] Z. Li, C. Huang, C. Bao, L. Chen, M. Lin, X. Wang, G. Zhong, B. Yu, W. Hu, L. Dai, P. Zhu, Z. Chang, Q. Wu, Y. Zhao, Y. Jia, P. Xu, H. Liu, and G. Shan. Exon-intron circular RNAs regulate transcription in the nucleus. *Nat. Struct. Mol. Biol.*, 22(3):256–264, Mar 2015.

[58] Ben Langmead, Cole Trapnell, Mihai Pop, and Steven L Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25, 2009.

[59] H. Li and R. Durbin. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760, may 2009.

[60] Faraz Hach, Fereydoun Hormozdiari, Can Alkan, Farhad Hormozdiari, Inanc Birol, Evan E Eichler, and S Cenk Sahinalp. mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nature Methods*, 7(8):576–577, aug 2010.

[61] Faraz Hach, Iman Sarrafi, Farhad Hormozdiari, Can Alkan, Evan E. Eichler, and S. Cenk Sahinalp. mrsFAST-ultra: a compact, SNP-aware mapper for high performance sequencing applications. *Nucleic Acids Research*, 42(W1):W494–W500, may 2014.

[62] Edward M McCreight. Priority search trees. *SIAM Journal on Computing*, 14(2):257–276, 1985.

[63] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, oct 2012.

[64] Weichun Huang, Leping Li, Jason R. Myers, and Gabor T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, dec 2011.

[65] Rui Dong, Xu-Kai Ma, Guo-Wei Li, and Li Yang. CIRCpedia v2: An updated database for comprehensive circular RNA annotation and expression comparison. *Genomics, Proteomics & Bioinformatics*, 16(4):226–233, aug 2018.

Appendix A

Commands

A.1 Simulation Generation

The following commands were used for generating simulation datasets.

For linear mRNA read simulation: **ART (v2.5.8)**

```
# art_illumina -1 HiSeq2kL100R1.txt -2 HiSeq2kL100R2.txt -ss HS20 -sam -i ${TRANSCRIPT FASTA}
-p -l 100 -c ${TRANSCRIPT EXPR} -m 350 -s 75 -o ${OUTPUT}
```

For circRNA read simulation: **modified version of CIRIsimulator**

```
# perl ./CIRIsimulator.pl -1 ${FQ1} -2 ${FQ2} -O ${OUTPUT} -G ${GTF} -DB ${CIRCRNA DATABASE}
-C 10 -LC 0 -R 1 -LR 1 -L 101 -E 1 -I 350 -M 75 -CHR1 0 -D ${REF DIR}$
```

A.2 Mapping Accuracy Experiment

The maximum intron length was set to 2Mb in CircMiner based on the maximum intron size observed in Ensemble gene annotation file version GRCh38.90. The maximum number of mismatches was set to 4 per mate for CircMiner and 8 per paired-end read for STAR. Also, the maximum soft clip length in CircMiner is set to 8 which means that split back-splice junction reads that have less than or equal to 8 split part cannot be detected.

We run CircMiner for mapping accuracy experiment as follows:

```
# circminer  -r ${REF} -g ${GTF} -f ${FQ1} -o ${OUTPREF} -k 20 -s 2 -t 1
-S ${SEEDLIM} -stage 0 -pe
```

For STAR (v2.6.0a), we use the following commands for mapping:

```
# STAR -genomeDir ${INDEXDIR} -runThreadN 1 -readFilesIn ${FQ1} ${FQ2}
-outSAMtype BAM Unsorted -outSAMunmapped Within -outFilterMismatchNmax 8
```

A.2.0.1 circRNA detection

We run CircMiner for all circRNA detection experiments as follows:

```
# circminer  -r ${REF} -g ${GTF} -f ${FQ1} -o ${OUTPREF} -k 20 -s 0 -t ${THREAD} -S 500 -pe
```

Other circRNA detection tools' commands used:

KNIFE (v1.4):

```
#  sh completeRun.sh ${FQ} appended ${OUTPUTDIR} ${PREFIX} 13 sam circReads 66 1
```

CIRI (v2.0):

```
#  bwa mem -T 19 -t ${THREAD} ${REF} ${FQ1} ${FQ2} > ${SAM}

#  perl ${CIRI2} -T ${THREAD} -I ${SAM} -O ${CIRI2_OUT} -A ${GTF} -G ${CIRI2_LOG}
-F ${REF} -0 -M ${MT}

#  perl ${CIRI_AS} -S ${SAM} -C ${CIRI2_OUT} -A ${GTF} -O ${AS_OUT} -G ${LOG} -F ${REF} -D yes

#  java -jar ./CIRI-full.jar RO1 -1 ${FQ1} -2 ${FQ2} -t ${THREAD} -o ${FULL_OUTPUT}/out

#  bwa mem -T 19 -t ${THREAD} ${REF} out_ro1.fq > out_ro1.sam

#  java -jar ./CIRI-full.jar RO2 -r ${REF} -t ${THREAD} -s ${FULL_OUTPUT}/out_ro1.sam -l 101
-o ${FULL_OUTPUT}/out

#  java -jar ./CIRI-full-1000.jar Merge -a ${GTF} -t ${THREAD} -c ${DEST}/${CIRI2_OUT}
-as ${DEST}/${AS_OUT_FILE} -ro ${RO} -o ${FULL_OUTPUT}/final -r ${REF}
```

CIRCexplorer (v1.1.10):

```
#  STAR -genomeDir ${INDEXDIR} -runThreadN ${THREAD} -readFilesIn ${FQ1} ${FQ2}
-outSAMtype BAM Unsorted -outSAMunmapped Within -chimSegmentMin 10

#  star_parse.py ${STAR}/Chimeric.out.junction fusion.junction &&
CIRCexplorer.py -j fusion.junction -g ${REF} -r ${GENEREF} -o ${OUTPUT}
```

CircMarker (downloaded on 12/04/2019):

For each chromosome in the genome a config file is generated specifying chromosome reference file and gtf file.

```
#  CircRnaDetectDraft ${CHR}/config.ini
```

SEGEMEHL (v0.3.4):

```
#  segemehl.x -q ${FQ1} -p ${FQ2} -t ${THREAD} -d ${REF} -i ${INDEX} -S ${BASENAME} > ${SAM}
```

To get circRNA count from SEGEMEHL's bed file output, run the following command:

```
#  cat ${BED} | tail -n+2 | grep ";C;" | cut -f 1-3 | sort | uniq -c > ${OUTPUT}
```