AUTOMATIC, FAST AND ACCURATE SEQUENCE DECONTAMINATION

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Biotechnology

by
Caner BAĞCI

July 2016
İZMİR

We approve the thesis of **Caner BAĞCI**

Examining Committee Members:

Assoc. Prof. Dr. Jens ALLMER
Department of Molecular Biology and Genetics, İzmir Institute of Technology

Asst. Prof. Dr. Selma TEKİR
Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Bünyamin AKGÜL
Department of Molecular Biology and Genetics, İzmir Institute of Technology

Asst. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering, İzmir Institute of Technology

Assoc. Prof. Dr. Hüseyin AKCAN
Department of Software Engineering, İzmir University of Economics

29 July 2016

Assoc. Prof. Dr. Jens ALLMER
Supervisor, Department of Molecular Biology and Genetics
İzmir Institute of Technology

Asst. Prof. Dr. Selma TEKİR
Cosupervisor, Department of Computer Engineering
İzmir Institute of Technology

Prof. Dr. Volga BULMUŞ
Head of the Department of Biotechnology and Bioengineering

Prof. Dr. Bilge KARAÇALI
Dean of the Graduate School of Engineering and Sciences

ACKNOWLEDGMENTS

I would like to thank my supervisor Dr. Jens Allmer, who have introduced me to bioinformatics and computational biology during my undergraduate studies. I would also like to thank him for being patient and supportive at times.

I grateful to my co-supervisor Dr. Selma Tekir, from whom I learned to look at things from a different perspective.

I thank my committee members, Dr. Mustafa Özuysal, Dr. Hüseyin Akcan and Dr. Bünyamin Akgül for accepting to be in my thesis defence committee and for their valuable criticism towards completion of this thesis.

I also thank TÜBİTAK for their financial support in my master studies with the grant number 113E326.

Finally, I must express my gratitude to my family and my friends, especially my colleagues from JLab, for their constant support and motivation.

ABSTRACT

AUTOMATIC, FAST AND ACCURATE SEQUENCE DECONTAMINATION

The introduction of massively parallel sequencing technologies was a revolutionary step in genomics. Their decreasing cost and powerful features have put them more and more on demand in the last decade. It is now possible to sequence even complete genomes of organisms, using massively parallel sequencing technologies even for small laboratories around the world.

However, the power of this powerful technology comes with its challenges. The challenges are both in technological and computational side of the work. In this work, one of these computational challenges is addressed and a novel algorithm is offered to solve the problem.

Sequencing by synthesis is one of the methods used in many different massively parallel sequencing instruments. This method utilizes the biological process of DNA replication and with the help of different means of detection, it allows sequencing a DNA molecule while it is replicated.

Since DNA polymerase requires a primer to start the replication reaction, short oligonucleotide adapters are used in sequencing by synthesis methods to initiate the reaction. However, certain circumstances allow these adapters to contaminate final sequence reads. Several tools have been offered to trim adapters from reads; but all depend on the prior knowledge of the adapter sequence by the bioinformatician.

In this work, an algorithm is offered to detect and trim adapters only using the sequences of reads, without relying on prior knowledge of adapter sequences. The algorithm was shown to perform better or on the same grounds with existing methods in terms of speed and efficiency.

ÖZET

OTOMATİK, HIZLI VE DOĞRU DİZİ DEKONTAMİNASYONU

Kitlesel parallel dizileme yöntemlerinin ortaya çıkışı genomik alanında devrim niteliğinde bir adım oldu. Giderek düşen fiyatları ve güçlü özellikleri bu yöntemleri her geçen gün daha ilgi çekici hale getirdi. Günümüzde bu yöntemlerin kullanımı, dünya çapında küçük laboratuvarların bile genom düzeyinde dizileme yapabilmesine olanak sağlamaktadır.

Ancak bu yöntemin de güçlü özellikleri yanında bazı problemleriyle geliyor. Bu problemler hem teknolojik, hem de bilişimsel alanlardadır. Bu çalışmada, bu bilişimsel problemlerden biri ele alınmış ve çözümü için yeni bir algoritma önerilmiştir.

Sentez ile sekanslama, bir çok kitlesel parallel sekanslama aletinde kullanılan yaygın bir yöntemdir. Bu yöntem biyolojik DNA kopyalanması reaksiyonunu kullanarak, değişik algılama yöntemleriyle DNA dizilimesi yapmayı sağlar.

DNA polimeraz enzimi kopyalama reaksiyonunu başlatabilmek için bir primer'e ihtiyaç duyduğu için, sentez ile sekanslama yöntemlerinde kısa adaptör sekansları kullanılır. Ancak bazı durumlar bu adaptörlerin sonuçta çıkan dizi okumalarını kontamine etmesine sebep olur. Bu dizileri temizlemek için çeşitli yöntemler önerilmiş olsa da, bunların hepsi adaptör dizilerinin önceden biliniyor olması varsayımı üzerine çalışır.

Bu çalışmada, adaptör sekanslarını önceden herhangi bir bilgi olmadan sadece okumaların kendilerini kullanarak bulan ve temizleyen bir algoritma önerilmektedir. Algoritmanın hız ve etkinlik açısından, var olan yöntemlerden daha iyi veya eşit düzeylerde olduğu gösterilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

bp . . . . . base pairs
NGS . . . . . Next-generation sequencing
MPS . . . . . Massively parallel sequencing
SBS . . . . . Sequencing by synthesis
miRNA . . . . . microRNA
nt . . . . . nucleotides
SRA . . . . . Sequence Read Archive

CHAPTER 1

INTRODUCTION

Genome is the inheritable material of an organism and genomics is the field that is concerned with the study of genomes in order to establish links between the genotypes and phenotypes (Mardis, 2008a). Two separate studies published in 2001 on the first draft of the human genome (Lander et al., 2001) (Venter et al., 2001) were the milestones for the so-called genomic era (Guarnaccia et al., 2014). The availability of the genomic data has changed the way genomes had been studied and it accelerated the advances in high-throughput methods to study genomes.

1.1. History of DNA Sequencing

The sequencing of nucleic acids (Deoxyribonucleic acid or Ribonucleic acid), that is determining the order of bases (nucleotides) in a nucleic acid was first achieved by Fiers et. al. in 1972 (Min Jou et al., 1972) and 1976 (Fiers et al., 1976), when they sequenced the complete RNA molecule of Bacteriophage MS2.

Maxam-Gilbert sequencing was the first of its kind that offered sequencing of DNA molecules by chemically modifying and later cleaving them at specific bases. The positions of cleavage (determined from the length of the cleaved molecule) were then used to construct the sequence of DNA (Maxam and Gilbert, 1977).

An independent work, also in 1977, by Frederick Sanger demonstrated the use of chain-termination method (Sanger sequencing) to sequence DNA molecules (Sanger et al., 1977). The method relies on the biological process of DNA replication and utilizes fluorescently labelled dideoxynucleotides which terminate the elongation of the DNA when incorporated. The DNA fragment of interest is first amplified (i.e. multiple copies of it are made). The amplified fragments, normal deoxynucleotides, fluorescently labelled dideoxynucleotides, the enzyme DNA polymerase and primers are then mixed to carry out the reaction. Each copy of the amplified DNA fragment results in a different length depending on the time a dideoxynucleotide is incorporated. Running the DNA fragments of different lengths in a capillary gel electrophoresis separates them by their length, thus

revealing the sequence of the DNA of interest (Figure 1.1) (Kircher and Kelso, 2010) (Shendure and Ji, 2008).

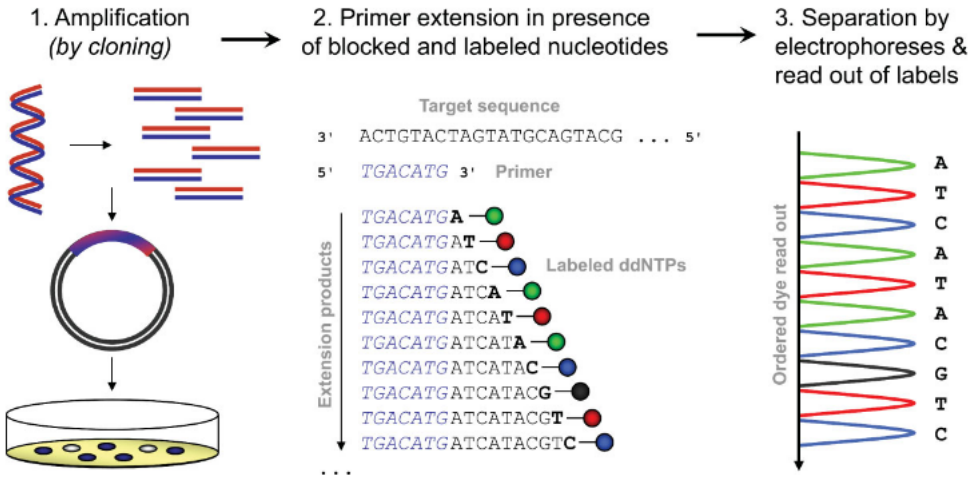

Figure 1.1. A schematic representation of Sanger sequencing. The fragment of amplified to multiple copies. The amplified fragments are extended in a polymerization reaction with normal dNTPs and fluorescently labeled ddNTPs. They are then separated based on their length on a capillary gel electrophoresis to determine the sequence of the fragment based on light emitted at different lengths of the extended copies. (Source: (Kircher and Kelso, 2010))

Technological developments and automated procedures on the method described by Sanger et. al., has shortly enabled sequencing of complete genes and eventually eukaryotic genomes (Goffeau et al., 1996) (Blattner et al., 1997) (Adams et al., 2000). It has reached the capability to sequence with an accuracy of 99.999% and read lengths longer than 1000 base pairs (bp) (Shendure and Ji, 2008). The first drafts of two human genome projects (Lander et al., 2001) (Venter et al., 2001) were also sequenced using automated Sanger sequencing technologies (Schuster, 2008). Although both projects producing a successful and satisfactory outcomes for the time; they took 13 years to complete. The Sanger sequencing method were not throughput and it was not applicable to individual cases; as it required too much time, labour and had very high costs (Schuster, 2008).

1.2. Massively Parallel Sequencing

The DNA sequencing technology has continued to rapidly develop after the completion of the human genome projects. A funding program initiated by the National Human Genome Research Institute in 2004, aiming to reduce the cost to sequence a human genome to $1000, has greatly encouraged the developments in this area (van Dijk et al., 2014). The first ”next-generation sequencing” (NGS) platforms became commercially available in 2004 (Mardis, 2008b) and they quickly replaced traditional Sanger sequencing for large scale genomic approaches (Metzker, 2010).

These massively parallel sequencing (MPS) (High throughput sequencing, Next-generation sequencing) platforms, although different at the way they sequence the nucleic acids, all share one thing in common that make them superior to the traditional Sanger sequencing technologies - sequencing millions of DNA fragments in parallel, thus yielding a high throughput (Schuster, 2008) (Shendure and Ji, 2008). Another feature mostly common between these methods is that the library preparation methods for preparing the sample to be sequenced varies only slightly between different instruments (Mardis, 2008b).

The low cost, which still continues to decrease, of massively parallel sequencing has made the genome wide sequencing available to even small laboratories around the world (van Dijk et al., 2014). It has enabled detection of variations between individuals, *de novo* assembly of genomes of organisms, replaced microarrays to quantify gene expression and even made it possible to detect novel genes or isoforms of genes (van Dijk et al., 2014).

The cost of sequencing today is decreasing even faster than the cost of storing data computationally (Figure 1.2) (Hayden, 2014). This has created a necessity for algorithms to analyze data become much more efficient as they form the bootleneck in many studies, where sequencing finishes in a day but a complete *de novo* genome assembly can take even months of CPU time.

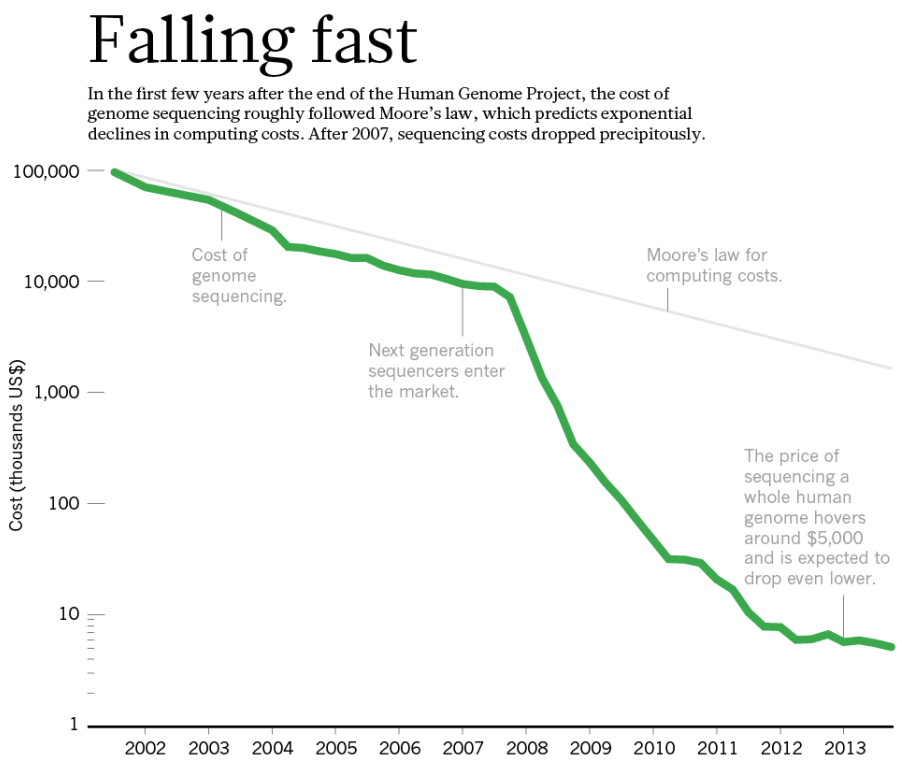

Figure 1.2. The rapid decline in the cost of sequencing. The cost of sequencing decreases at a rate much faster than the cost to store the sequencing data. The scales are logarithmic. (Source: (Hayden, 2014))

1.2.1. Sequencing by Synthesis

Sequencing by synthesis (SBS) is a principle of sequencing used in many massively parallel sequencing instruments (e.g. Illumina sequencing, pyrosequencing, ion torrent semiconductor sequencing). The main idea behind SBS is that it utilizes a polymerase enzyme which replicates the DNA, similar to the Sanger sequencing method. During the replication of the DNA, nucleotides or short oligonucleotides are either added one at a time or modified version of them (e.g. fluorescently labeled) are added to the system to determine the nucleotide incorporated to the growing chain of DNA (Fuller et al., 2009).

Unlike in Sanger sequencing, the addition of labelled deoxynucleotides to the growing chain of DNA molecule do not terminate the replication reaction. This in turn enables them to be magnitudes of order faster than the traditional Sanger sequencing

methods. In Illumina sequencing (described in more detail in Subsection 1.2.1.1), for instance, the addition of a fluorescently labelled deoxynucleotide pauses the replication at each step. This; however, is reversible and the cleavage of the fluorescence tag after it is detected makes it possible for the reaction to continue.

The mean of detecting the nucleotide added to the DNA at each step in SBS methods differ from one technology to another. Illumina sequencing and Pyrosequencing methods detect the light emitted by either fluorescently labelled deoxynucleotides or by a chemical reaction between the pyrophosphate molecule released the deoxynucleotide and ATP, respectively. Ion semiconductor sequencing, on the other hand, detects the changes in the pH from the hydrogen ions released when a deoxynucleotide is successfully incorporated by adding them one at a time in each cycle (Liu et al., 2012).

Since the SBS principle is based on the biological process of DNA replication and it utilizes a DNA polymerase, raw DNA fragments right after isolation cannot be sequenced using these methods. They need to undergo a collection of steps called library preparation, which differ from instrument to instrument but basically include DNA fragmentation and adapter ligation at its core. Ligation of the adapters is essential, as the enzyme DNA polymerase requires a primer in order to initiate the polymerization (replication of DNA) reaction. The primers are then designed to be complementary to the adapter, which bind to them and allow the initiation of the polymerization by DNA polymerase (Figure 1.4) (Fuller et al., 2009).

As the details of each sequencing methodology is beyond the scope of this study, Illumina sequencing will be explained in further detail as an example to SBS methods.

1.2.1.1. Illumina Sequencing

The library preparation step for Illumina sequencing begins with fragmentation of the purified DNA into smaller pieces (these will be referred as fragments). These fragments are further modified with end repair and dA tailing. They are then ligated with adapters whose aim is to initiate polymerization reaction and optionally with marker (index) sequences in order to differentiate between fragments coming from different sources but are mixed to be sequenced in one run. A mean of selection can be applied on adapter ligated fragments (e.g. size selection, poly-A selection), and later they are PCR-enriched (Figure 1.3 (Mardis, 2013) (Zhong et al., 2011).

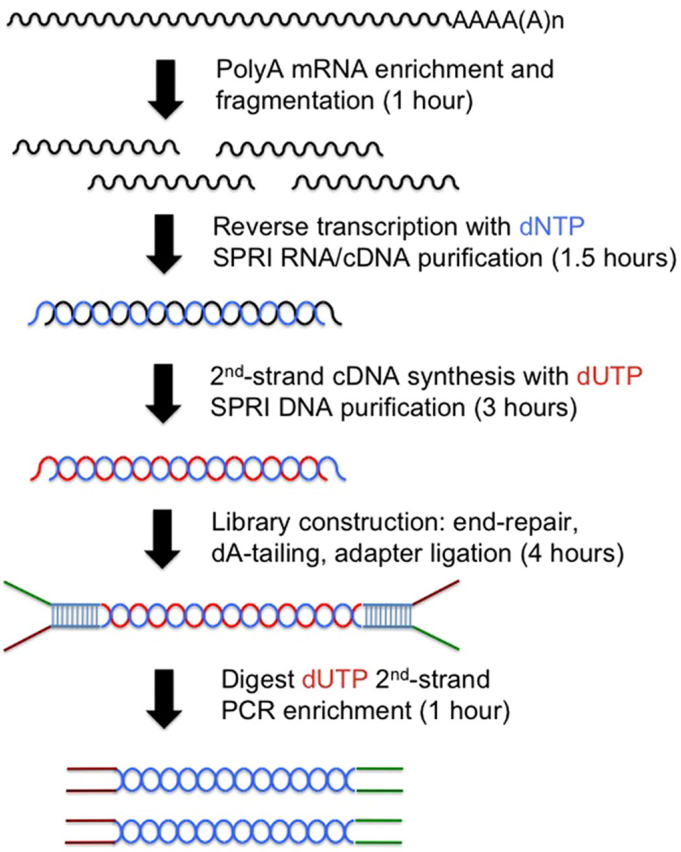

Figure 1.3. An example to Illumina library preparation protocol. Red and green segments in the end product are a pair of adapters that are ligated to the fragmented DNA molecule. (Source: (Zhong et al., 2011))

These PCR-enriched fragments then attach to specialized chips and undergo cluster generation by bridge amplification. Sequencing by synthesis begins with attachment of DNA-polymerase onto the fragments. At each cycle of the sequencing run, all 4 nucleotides, each labelled with a different fluorescent label, are added to the system. Incorporation of one of these prevents another nucleotide binding the growing fragment with the help of a blocking group attached to 3'OH of the ribose sugar. After the incorporation, remaining nucleotides are washed away and the fluorescence image of the added nucleotides are captured. Then the fluorescence tag and the 3'OH blocking group is removed from the nucleotide to allow the next nucleotide incorporation. This repeats until a predefined number of cycles, which determine the length of the reads (Figure 1.4) (Mardis, 2013).

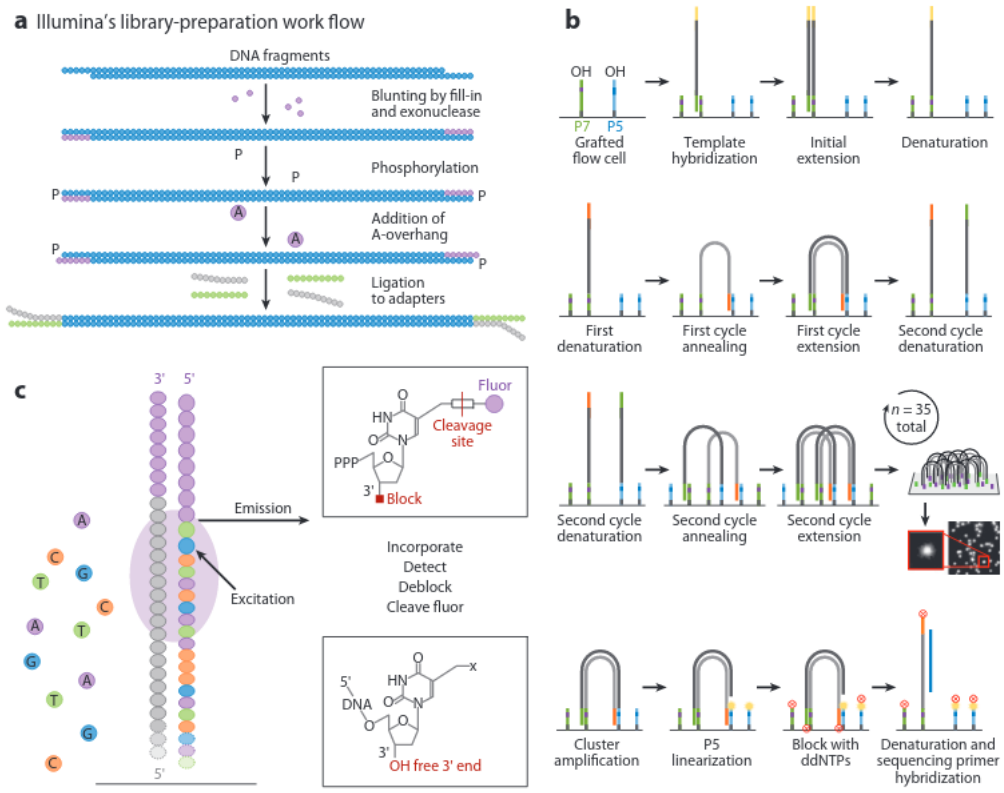

Figure 1.4. Illumina reversible dye termination sequencing methodology. (Source: (Mardis, 2013))

1.3. Computational Analysis of Sequence Data

The end products of an MPS run is a collection sequence reads (sequencing library). A sequence read (usually referred as a "read") is an ordered stretch of nucleotides that are detected from one of the sequenced fragments and a sequencing library is made up of the collection of these reads which come from the fragments generated in the first step of the library preparation. Sequence reads are computationally stored as strings. Signals detected from sequencing instrument (e.g. the light emitted) are converted to computational data with the help of base-calling algorithms (Ledergerber and Dessimoz, 2011). Base-calling algorithms also assigns a quality score to each nucleotide in a read. A quality score is a measure of how confident the algorithm is for the base-calling to be correct at that position. There are several reasons that can affect the quality of base-calling due to the noise in detected signals.

Reads and their quality values are stored in 2 standard data formats. FASTQ file format stores each read in 4 lines in a sequencing library. The first line starts with ”@” is followed by a unique identifier for that read. The second line stores the sequence of the read. The third line starts with ”+” and can contain the same unique identifier for the read; and lastly the fourth line has the quality scores in letters and special characters (there exists different quality formats) for each nucleotide in the read in the same order (Figure: 1.5) (Cock et al., 2010). The format is then repeated for all reads coming from a run.

```
1  @SRR445991.3 ILLUMINA-08A740:1:FC636GEAAXX:1:1:2268:1162 length=40
2  NGTAAACATCCTCGACTGGAAGC
3  +SRR445991.3 ILLUMINA-08A740:1:FC636GEAAXX:1:1:2268:1162 length=40
4  #,,,*'+)),@@@@:@@@@:@@@
5  @SRR445991.8 ILLUMINA-08A740:1:FC636GEAAXX:1:1:4699:1160 length=40
6  NAGCTCGTCGGGCCCGGGGGGAGG
7  +SRR445991.8 ILLUMINA-08A740:1:FC636GEAAXX:1:1:4699:1160 length=40
8  ########################
9  @SRR445991.14 ILLUMINA-08A740:1:FC636GEAAXX:1:1:6362:1163 length=40
10 NAGCTTTAATGCTAATTGTGATAGGGGTT
11 +SRR445991.14 ILLUMINA-08A740:1:FC636GEAAXX:1:1:6362:1163 length=40
12 #-0--54455@C@@CCC@@@:<6<<@@@@
```

Figure 1.5. An example to FASTQ file format. 3 reads from an Illumina sequencing run is shown.

The reads can also be stored in a pair of files: a FASTA file and a qual file. A FASTA file only stores the sequences; each taking at least 2 line. The identifier in FASTA format is preceded with a greater than sign instead of ”@” and the sequences can take multiple lines instead of only one like in FASTQ. qual files store the quality information associated with each nucleotide in the FASTA file. They have the same identifiers and the same format for the identifier as in FASTA format. They store the quality information in a numeric format, separated by space for each nucleotide (Cock et al., 2010).

The left side of a sequence read is called its ”5 prime” (5’), while the right side of it is called ”3 prime” (3’) for biological reasons. This is because, DNA is replicated from its 5’ end to 3’ end; and computational storage of sequence data has also adopted this and sequences are stored computational in 5’-to-3’ orientation.

Although the computational methods employed in the analysis of sequence data produced from MPS methods differ from study to study depending on its aim; some steps are expected in every pipeline. In this section, the focus will be more on preprocessing

of sequence reads, which is a collection of steps that should be applied to raw sequence data before any downstream analysis; and what the downstream analysis can include afterwards will briefly be introduced.

1.3.1. Preprocessing of Sequence Reads

The preprocessing step in sequencing data analysis mainly includes trimming sequence reads from adapters and low quality regions. Adapters, as explained in ”Sequencing by synthesis” section, are short nucleic acids that are used to initiate polymerization reaction, thus the sequencing. The adapters do not belong to the sequence of interest, but are inserted only to be used as a tool. Quality in sequence reads means the probability of a base calling at a position being correct. Regions with adapters and of low quality needs to be trimmed of sequences, because they do not represent the sequence of interest and have the potential to disrupt downstream analysis steps (Bolger et al., 2014), as also shown with examples in the Results chapter.

1.3.1.1. Adapter Trimming

Adapters are short oligonucleotides that are used in SBS methods to initiate sequencing. They are expected to serve as a tool; however, undesired circumstances can lead to sequence reads which contain either partial or full length adapter sequences with or without the biological sequence of interest. The main reason adapters being present in sequence data is that due to the nature of fragmentation methods used to fragment purified DNA, a percentage of produced fragments ends up with lengths shorter than the desired read length (Jiang et al., 2014a). When this is the case, a read belonging to a fragment shorter than the read length turns out to contain a portion of the adapter used in sequencing, sometimes even the full length adapter sequence (Figure 1.6). The percentage of adapter contaminated reads depends on the targeted fragment length in library preparation and desired read length.

Another possible source is the multiplexed 5’ adapters in the fragment being sequenced. Sometimes more than one adapter may be ligated to a fragment (adapters ligating to each other) and the primer to initiate sequencing may end up attaching to one of

the multiplexed adapters other than the right-most in 5' of the fragment. This makes it inevitable to sequence the other multiplexed adapters downstream the adapter a primer is attached to (Figure 1.6).

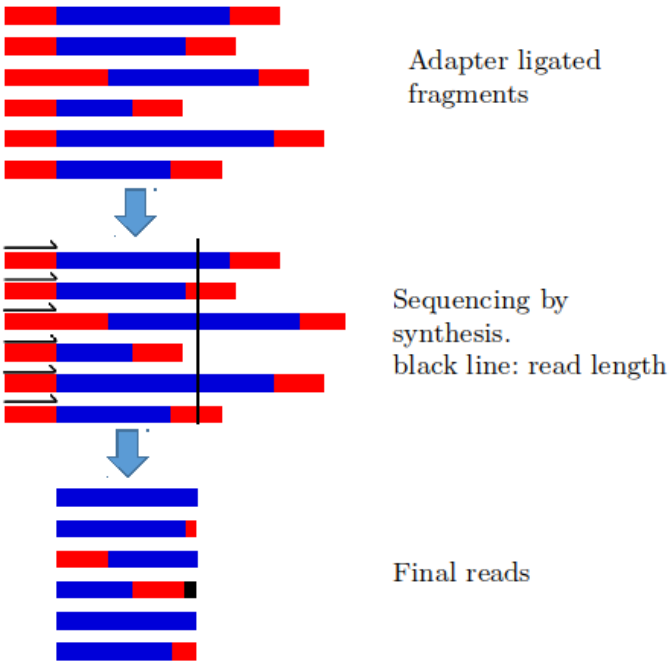

Figure 1.6. Sources of adapter contamination. Blue lines are fragments, while red lines are adapters ligated to them. The arrows show the direction of the sequencing. Black line shows the desired read length (number of cycles in Illumina sequencing).

It is fundamental for downstream analysis to remove adapters and restore the target DNA. Adapters left in the sequences may cause misalignment or reads not mapping back to their reference genome/transcriptome; as read mapping is usually performed in end-to-end alignment where the complete read must align to the reference. They also affect *de novo* assemblies of genomes or transcriptomes badly. Assembly is the procedure of organizing sequence reads based on the information of their ends matching with each other in order to reconstruct the DNA or RNA molecule before it was fragmented to be sequenced. Having adapter contaminations left; however, prevents the ends to align and leaves gaps in the final assembly (Figure 1.7).

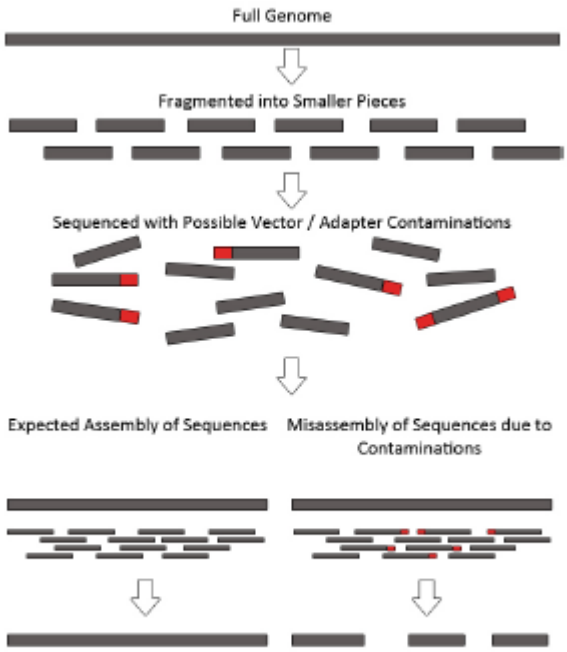

Figure 1.7. Sources of adapter contamination. Blue lines are fragments, while red lines are adapters ligated to them. The arrows show the direction of the sequencing. Black line shows the desired read length (number of cycles in Illumina sequencing).

Adapter trimming is especially important in microRNA (miRNA) sequencing studies. The length of a mature miRNA ranges from 18 to 30 (nt), whereas the typical minimum read length for sequencing platforms is 36 nt. This results in all reads (assuming the sample was perfectly purified for miRNAs) having a portion of the adapter and the trimming becomes obligatory to obtain real mature microRNA sequences. This is similarly true for ancient DNA studies as well, where DNA from ancient samples are subject to degradation, and the molecules that can be recovered are usually shorter than 100 bp (Sawyer et al., 2012).

1.3.2. Quality Trimming

Next-generation sequencing datasets tend to have sequencing errors due to the nature of sequencing methods (Fuller et al., 2009). There are several possibilities which can lead to wrong base callings or biases in reads, such as random hexamer priming (Hansen et al., 2010), phasing errors (one read falling out of timing compared to others) (Fuller et al., 2009), and accumulation of errors towards the end of the fragments (Dohm et al., 2008). Like adapter contaminations, it is reported that these error have an impact on the portion of reads that can be mapped back to a reference genome or transcriptome (Fabbro et al., 2013). It has also been reported that they change the expression estimates in RNA-Seq studies (Williams et al., 2016).

Each nucleotide in a read is given a quality score by the base calling algorithm, which represents its error rate. Quality trimming tools for NGS data make use of these quality scores and trim reads of low quality regions of base calling or discard them completely if the overall base calling generally of low quality for the whole read.

1.3.3. Downstream Data Analysis

There exists two many paths which can be taken depending on the situation after the raw reads have been preprocessed - reference based approaches and *de novo* approaches. The path to take usually depends on the availability and quality of a reference genome and/or transcriptome assembly.

1.3.3.1. Reference Based Approaches

Reference based next-generation sequence data analysis pipelines are used in resequencing studies where a reference genome and/or transcriptome assembly is available. Reads are first mapped to reference assembly and the next steps may differ from experiment to experiment depending on the aim. There exists many tools to map reads back to reference assemblies, each with different aims and advantages; such as mapping efficiently reads of different lengths or mapping them with gaps taking splicing events into consideration for RNA-Seq data. The mapping results can then be used in reference

based assemblies of genes or genomes, further annotation of the reference, quantification of gene expression and comparison of them among samples, variation calling, detection of novel genes or isoforms of known genes, microRNA identification and detection organisms from a sample in metagenomics samples.

1.3.3.2. *De novo* Approaches

De novo next-generation sequencing data analysis approaches are used when a reference genome and/or transcriptome is not yet available. The reads are assembled by *de novo* assembly methods from overlapping fragments between reads into larger contigs. These larger contigs may then be used in scaffolding to produce final transcripts (transcriptome) or chromosomes (genome). The resulting *de novo* assembly (genome or transcriptome) can be used in annotation to assign it a functional meaning, discovery of unknown genes or isoforms of known genes, development of genomic markers on the genome.

1.4. Problem Definition

Having described possible paths an NGS dataset can go through, the analysis should always start with preprocessing of reads which include trimming them from technical sequences (i.e. adapters) and from low quality regions. In this study, the main interest is the process of trimming reads from adapter contaminations. Several methods have been offered before to trim reads from adapters; all of which have based their operation on finding the sequence of an adapter or a set of user defined adapters in the sequence of reads.

1.4.1. Current Methodologies

Several methodologies have been offered before to trim sequencing adapters from sequence reads (Table 1.1). They all offer different advantages and focus on different aspects of the problem. However; all methods that have been offered so far rely on the

prior knowledge of the adapters used during the library preparation for an MPS experiment. Some methods focus on trimming adapters from paired-end sequencing experiments, some do not take advantage of mate-pair information, some focus on one specific method of SBS (e.g. Nextera); and they also differ at the algorithms used to find the adapter inside the reads. The algorithms to trim adapters from reads usually employ semi-global alignments, where the given adapter sequence is aligned to all reads iteratively.

Table 1.1. Algorithms Offered for Adapter Trimming.

Name	Trimming From	Quality Control	Citation
FastX	3' SE	Ns	(Gordon and Hannon, 2010)
SeqTrim	3' SE	Ns & LQ	(Falgueras et al., 2010)
TagCleaner	5' & 3' SE	No	(Schmieder et al., 2010)
Cutadapt	3' & 5' SE & PE	LQ	(Martin, 2011)
Btrim	5' & 3' SE & PE	LQ	(Kong, 2011)
Flexbar	5' & 3' SE & PE	Ns & LQ	(Dodt et al., 2012)
Trimmomatic	3' SE & PE	LQ	(Bolger et al., 2014)
AdapterRemoval	5' & 3' SE & PE	Ns & LQ	(Lindgreen, 2012)
AlienTrimmer	5' & 3' SE & PE	LQ	(Criscuolo and Brisse, 2013)
NextClip	LMP	No	(Leggett et al., 2013)
Skewer	5' & 3' SE & PE & LMP	Ns & LQ	(Jiang et al., 2014b)

5' and 3' shows whether the algorithm can detect and trim adapters in 5' or 3' ends of the reads. SE and PE shows whether the algorithm can take single-end and/or paired-end data as input. Ns mean the algorithm can also trim reads from Ns (unknown nucleotides). LQ means the algorithm can also trim reads from low quality regions.

1.4.2. Aim

The aim of this study is to automatically detect adapters and any other contamination in sequence reads of a massively parallel sequencing dataset and to trim them effeciently. To achieve this, an algorithm (RAT - RAdix Tree based read trimmer) employing radix trees to detect adapters in a single sequencing run and trim them automatically is offered.

CHAPTER 2

METHODOLOGY

The source of technical contaminations (e.g. adapters) in a single sequencing dataset ought to be similar due to the nature of sequencing by synthesis methods (the same pair of adapters is used for all fragments). In this study, adapters are identified and trimmed from reads without prior need for the information on adapter sequences used during library preparation.

Biological sequencing reads are computationally stored as strings, each character of the string denoting one of the nucleotides (A, C, G, T). The computational task to identify adapters is to find a substring of length (l) at either end in a majority of a collection of strings (reads). However, the length of the adapter (l is not constant and each read can contain an unknown portion of it ranging from 0 to l.

2.1. Trees

The problem is offered to be solved by representing a library of reads (collection of strings) in a tree data structure and finding paths in the tree, from root a node, where many subsequences will branch (finding common subsequences at the ends of sequences).

In computer science, a tree is a non-linear data structure that arranges data (a collection of items) hierarchically starting from its root to nodes that are linked to each other. Their usage spans many areas of computer science, including analysing of electrical circuits, representing electrical circuits, organizing data in database systems and indexing biological data in bioinformatics (Aho et al., 1983).

A tree is basically a collection of nodes that are connected to each other by edges (vertices) in a hierarchical manner, starting from the ”root” node. If n1, n2, ..., nk is a sequence of nodes in a tree, meaning that n(i+1) can be reached from the connections of ni, this is called a ”path” from n1 to nk. Hierarchically, ni is the ”parent” of n(i+1) in this case, and n(i+1) is called a ”child” of ni. A node that does not have any child is called a ”leaf” or ”terminal node” (Aho et al., 1983).

The use of tree structures for string operations has been offered long before computational biology has made an impact (Weiner, 1973). Suffix trees, for example, provide linear time solutions to exact string matching problem. Although, this is the worst-case boundary as in Knuth-Morries-Pratt or Boyer-Moore algorithms as well; the power of suffix trees comes in substring finding problem. Suffix trees can offer *O(m)* time to process a string *T* of length *m*, and then *O(n)* time to solve the question of whether a string *S* of length *n* is contained within the string *T* (Gusfield, 1997).

2.1.1. Tries and Radix Trees

A trie is a special type of tree used to store a collection of strings. All edges of a trie must have a label and each node can have 0 to *n* children. The labels can be only one character long for each edge. Only the root does not have a parent. All strings are stored in unique paths from the root to leaves, thus each leaf represents the end point for a string in the given collection of strings. The concatenation of edge labels in a root to leaf path returns a complete string item (Figure 2.1).

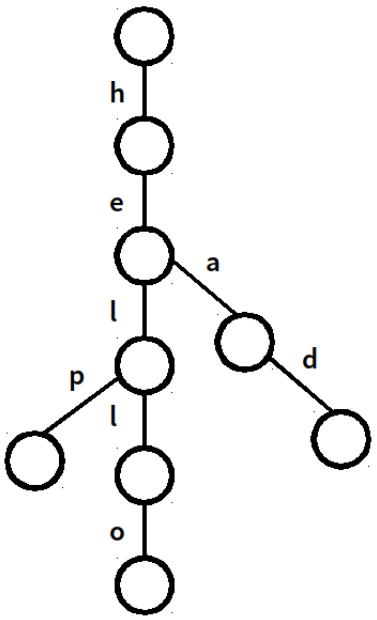

Figure 2.1. An example to a trie, containing words ”hello”, ”help” and ”head”.

A radix tree (also called Patricia tree, PAT tree, compressed trie) is a data structure that represents a collection of strings in the form of a tree. The difference it holds compared to tries is that the edge labels are compressed to make it more space optimized. Thus a node having only one child, can be combined with its child to become a node with an edge labeled not only with a character but the concatenation of characters that will have to at least 2 children (shared by at least 2 children). Thus, every node in a radix trie, except the leaf nodes, must have at least 2 children different from the definition of tries.

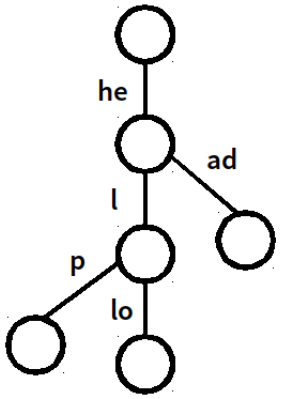

Figure 2.2. An example to a radix tree, containing words ”hello”, ”help” and ”head”.

2.2. Radix Trees for Sequence Decontamination

If all reads belonging to a single sequencing run is stored on a radix tree in an inverted orientation (i.e. starting from the end of the read - inverted radix tree), the frequencies of substrings in the 3’ of all reads could be retrieved easily. Since the 3’ adapter contamination, although for different lengths, would be present in an unexpected rate of reads compared to the biological sequences, one would expect shared subsequences at the right end of all reads containing a portion of the adapter.

Based on this assumption, a radix tree from all reads of a sequencing run is built. If given FASTA input, sequence headers are treated as values in the leaves of the tree; whereas for FASTQ input, a quality object is created with the sequence header and the quality attached to it and it is used as the value in leaves. While building the tree, each

node of the tree is assigned a score, which equals to the number of leaves reachable from that node. This score represents the number of reads that end with the label (subsequence) bound to that node (the path from the root to the node). After building the tree, it is traversed starting from the root and visiting each node. While traversing, the scores of each node and the label it is associated with (the path from root to that node) are collected and sorted in a decreasing order; and subsequences are inverted back to their original form. When the traversal and sorting is complete, starting from the label with the highest score (assuming it is a contamination), the label is extended until further extension is not possible. The extension is done by going down the list of labels and finding the next longer label that contains the previous one. The labels are again extended in the original orientation (5' to 3'). The latest label when the extension is finished, is then assumed to be the adapter. The radix tree is then searched for subsequences of this adapter and reads are trimmed from it when found.

When the extension is not possible for the highest scoring label, or all extended labels contain mostly the same nucleotide (e.g. poly-A), the search is restarted from the next-highest scoring label and all previous labels used in extensions that ended up not being useful are removed from the list.

2.2.1. Implementation

All parts of the algorithm is implemented in JAVA programming language. The developed tool can work with both FASTA and FASTQ formatted sequences as input or output. It also supports paired-end sequencing files, and the algorithm has additional step of re-sorting the reads based on their definition lines for paired-end sequencing data in order to preserve mate-pair information. The information on trimmed reads and sequences trimmed from them are logged into separate files. Users can also tests the tool with the option of simulating a sequencing run, with user defined sequencing depth, average fragment and read length and their standard deviations.

Users can also decide to minimum number of nucleotides to trim from reads (default: 1), and a length threshold to completely discard reads if they fall below it (default: none).

The algorithm requires around 15GB of heap space for a typical sequencing run of 10000000 reads of length 100 nucleotides.

CHAPTER 3

RESULTS

In this chapter, the results for test cases with simulated and real datasets used to evaluate the effectiveness of the algorithm will be presented.

3.1. Simulation Tests

An in-house script was used to simulate NGS datasets with 5 different options: read length, fragment length, fragment length deviation, depth and adapter sequence. Read length is the number of bases that will be sequenced from a fragment (e.g. number of cycles in Illumina sequencing). Fragment length is the desired the length for a simulated fragment; however, it can deviate from that as much as given fragment length deviation following a normal distribution (e.g. if 100 is defined as the fragment length with a deviation of 50, fragments of length from 50 to 150 can be produced following a normal distribution). Depth is the number of fragments that will be produced. Adapter sequence is a user defined string that will be used in place of an adapter during simulation when necessary. The script does not use any reference genome or transcriptome, the reads produced consist of totally random bases. It records the length of the fragment a read is produced from; thus making it possible to tell whether an adapter has been trimmed correctly or not. The adapters are added to the end of the fragments until the read length is satisfied, when the length of the fragment is shorter than the read length.

4 different sets of simulated data were produced, each with 5 replicates. Simulation Dataset 1, 2 and 3 (sim1, sim2, sim3) had a depth of 1000000 fragments. Simulation Dataset 4 (sim4) had a depth of 10000000 fragments. sim1 and sim4 had a read length of 100 nt, average fragment length of 100 nt with a deviation of 50 nt. sim2 had a shorter read length at 50 nt. The fragments for the sim2 dataset had an average length of 50 nt with a deviation of 20 nt. sim3 had 250 nt long reads, which is longer than all other datasets. The average fragment length for sim3 dataset were 250 with a deviation of 50.

The simulated data were stored as FASTA files and these were used as inputs for both the algorithm described here (RAT) and 3 other adapter trimming tools that were used

as a reference to compare the performance of RAT: cutadapt (Martin, 2011), AdapterRemoval (Lindgreen, 2012), skewer (Jiang et al., 2014b). All tools were executed with the default settings and the complete sequence of the adapter that was inserted to the reads during the simulation. The analysis were performed on a Linux machine with a 8-core processor at 3.60GHz and 32GB of RAM. During the execution, the user runtimes and peak memory usages were recorded. The outputs were used to determine the number of reads in each simulation that were correctly trimmed (either trimmed to the length of the original fragment or did not require trimming - true positives and true negatives), overtrimmed (the simulated read was originally larger than the trimmed read- false positives) and undertrimmed (a portion of adapter is left. Final trimmed read still contained adapter and had a length longer than its fragment - false negatives).

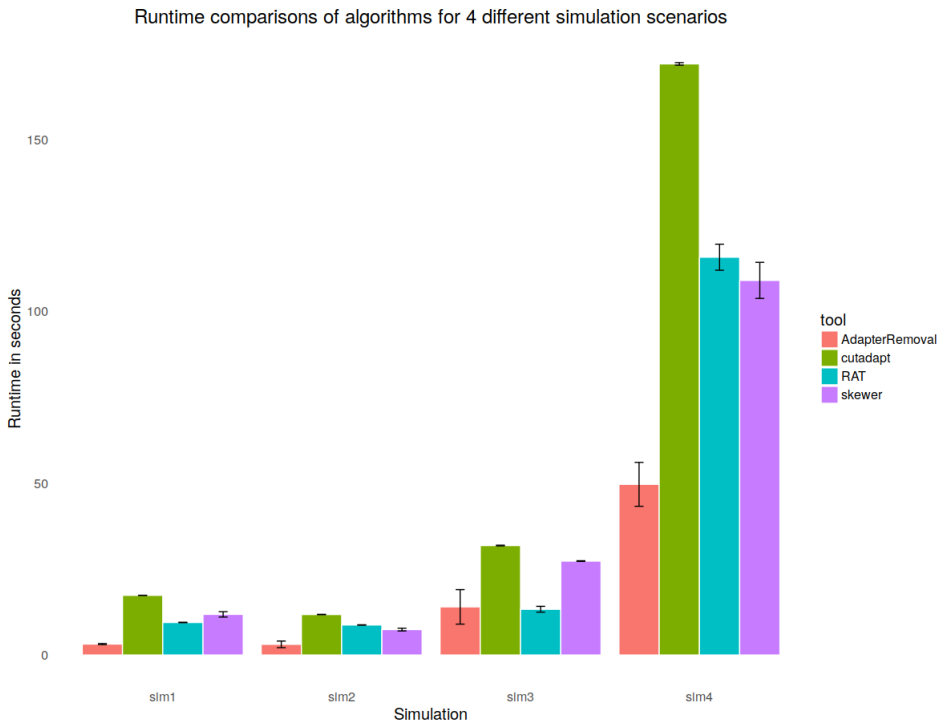

Figure 3.1. Runtime comparison of RAT and 3 other algorithms compared for 4 different simulation scenarios. sim1, sim2 and sim3 had a depth of 1000000. sim1 had a read length of 100nt, average fragment length of 100nt with a deviation of 50nt; sim2 had a read length of 50nt, average fragment length of 50nt with a deviation of 20nt; sim3 had a read length of 250nt, average fragment length of 250nt with a deviation of 50nt; sim4 had a depth of 10000000, read length of 100nt, average fragment length of 100nt with a deviation of 50nt.

Figure 3.1 shows the average runtimes for the 4 different simulation scenarios mentioned above with their error rates originating from 5 replicates for each scenario. In all 4 scenarios RAT has outperformed cutadapt in terms of speed. The runtimes were similar for RAT and skewer; but it was always slower than AdapterRemoval.

Table 3.1 lists the p-values for Welch Two Sample t-Test for testing the null hypothesis that the means of runtime are equal for RAT and other 3 tools used as a reference. It shows that RAT was in all test cases significantly faster than cutadapt, significantly slower than AdapterRemoval in sim1, sim2 and sim4; and significantly faster than skewer in sim1 and sim3 but slower in sim2.

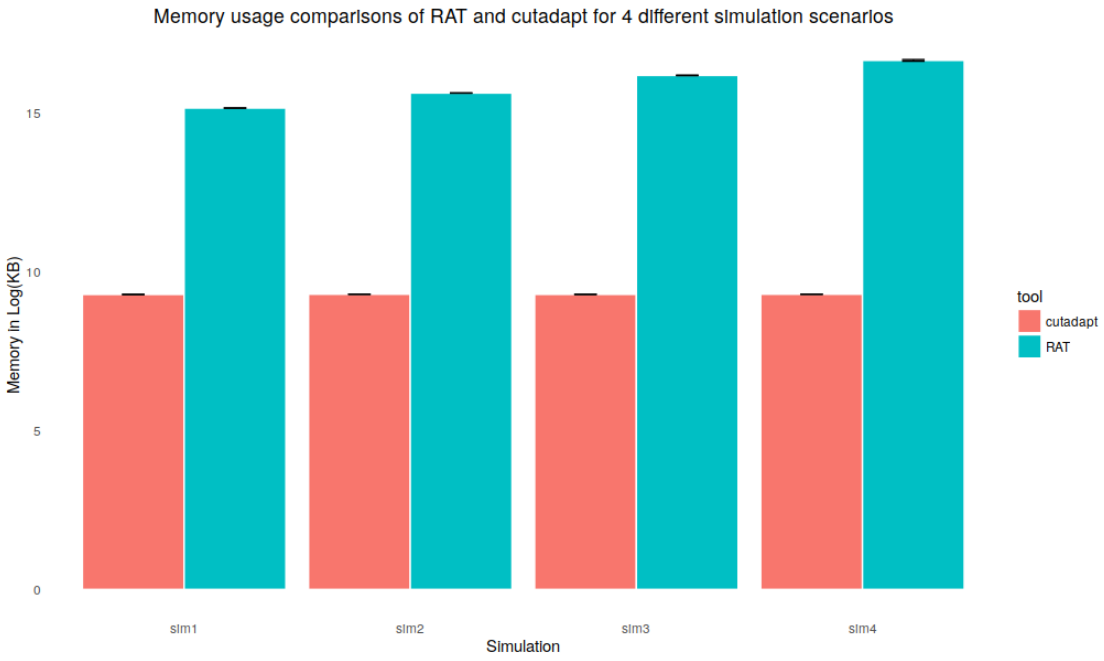

Figure 3.2. Peak memory usage comparison of RAT and cutadapt for 4 different simulation scenarios. The y-axis in the figure is in logarithmic scale. sim1, sim2 and sim3 had a depth of 1000000. sim1 had a read length of 100nt, average fragment length of 100nt with a deviation of 50nt; sim2 had a read length of 50nt, average fragment length of 50nt with a deviation of 20nt; sim3 had a read length of 250nt, average fragment length of 250nt with a deviation of 50nt; sim4 had a depth of 10000000, read length of 100nt, average fragment length of 100nt with a deviation of 50nt.

As shown in Figure 3.2, peak memory usages had a great difference between the two algorithms (note that the y-axis is in the logarithmic scale). RAT needed significantly more memory compared to cutadapt. This comes from the design of algorithms, as RAT

places all reads onto a radix tree; while cutadapt aligns them one-by-one two a reference adapter sequence. RAT; on the other hand; does not need the reference adapter sequence, and can trim sequences without prior knowledge of it.

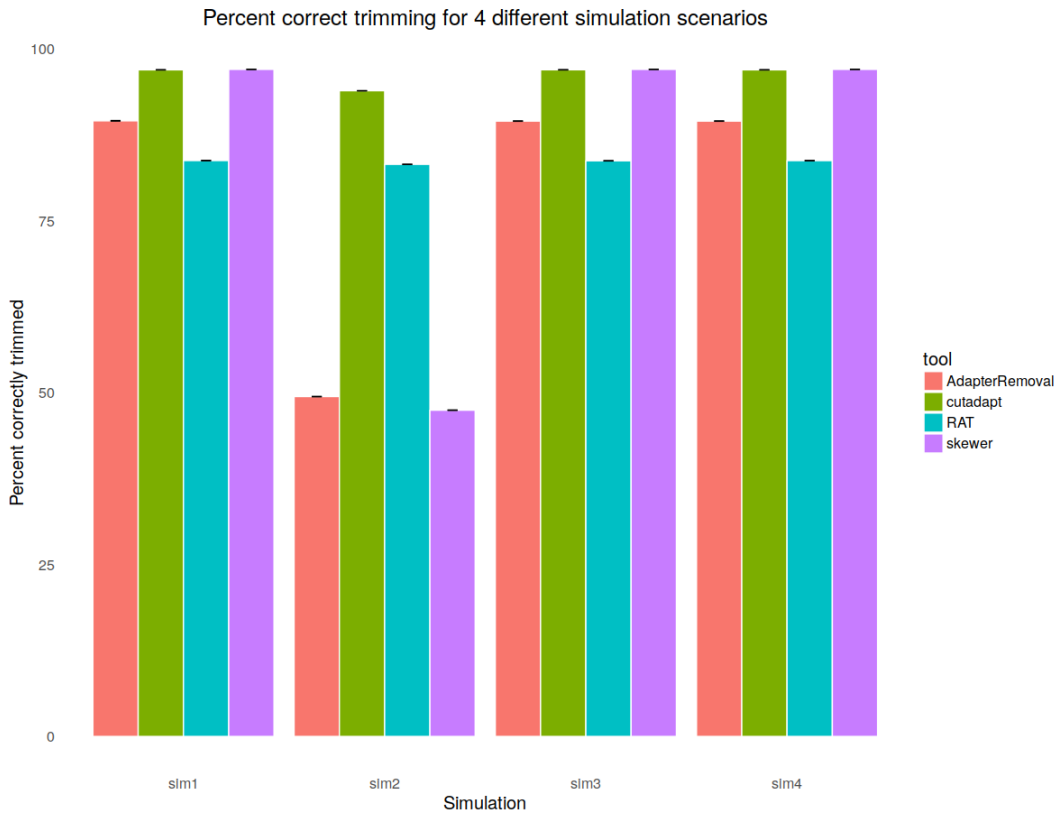

Figure 3.3. Percentage of correctly trimmed reads in 4 test cases by RAT, cutadapt, AdapterRemoval and skewer

In terms of efficiency (Figure 3.3, Figure 3.4 and Figure 3.5) both algorithms performed on similar rates; although Figure 3.3 shows that percentage of correctly trimmed reads are higher in cutadapt and AdapterRemoval in all test cases and generally higher in skewer than RAT. This is due to percentage of overtrimmed reads being higher in RAT (Figure 3.4. When added up, the efficiency in trimming the adapter is similar in both all 4 algorithms tested. RAT had significantly higher percentage of overtrimmed reads in all cases (Figure 3.4, because by default it tries to trim the matching sequences at the end of the reads regardless of their length even if it is only a single nucleotide. This causes many short overtrimmed nucleotides at the ends of the reads; when they actually belong the biological fragment but match to the beginning of the identified adapter sequence.

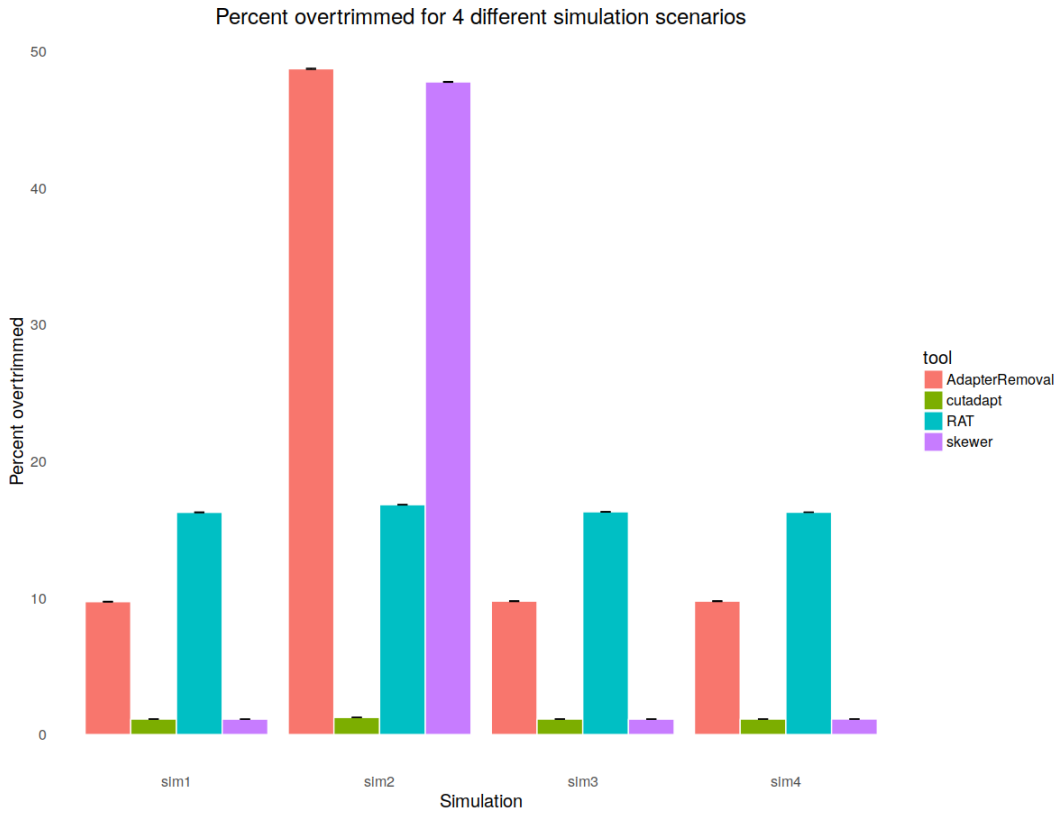

Figure 3.4. Percentage of overtrimmed reads in 4 test cases by RAT, cutadapt, AdapterRemoval and skewer

cutadapt for instance; however, do not trim the reads if the length of the match is shorter than 3. This is configurable by the user in RAT; however, in most cases, it would be a better approach to overtrim reads than leave adapters untrimmed even if the suspected contamination is very short. The average length of overtrimming by RAT was 1.35 nt, whereas it was 3.27 nt for cutadapt.

In all 4 test scenarios and for all 5 replicates of them, RAT had exactly 0 undertrimmed reads, because it tried to trim every suspected contamination it found (thus a high rate of overtrimming, as explained above). The other 3 algorithms; on the other hand, left up to 2% of reads undertrimmed (Figure 3.5).

The details of all simulation results are available in Appendix A.

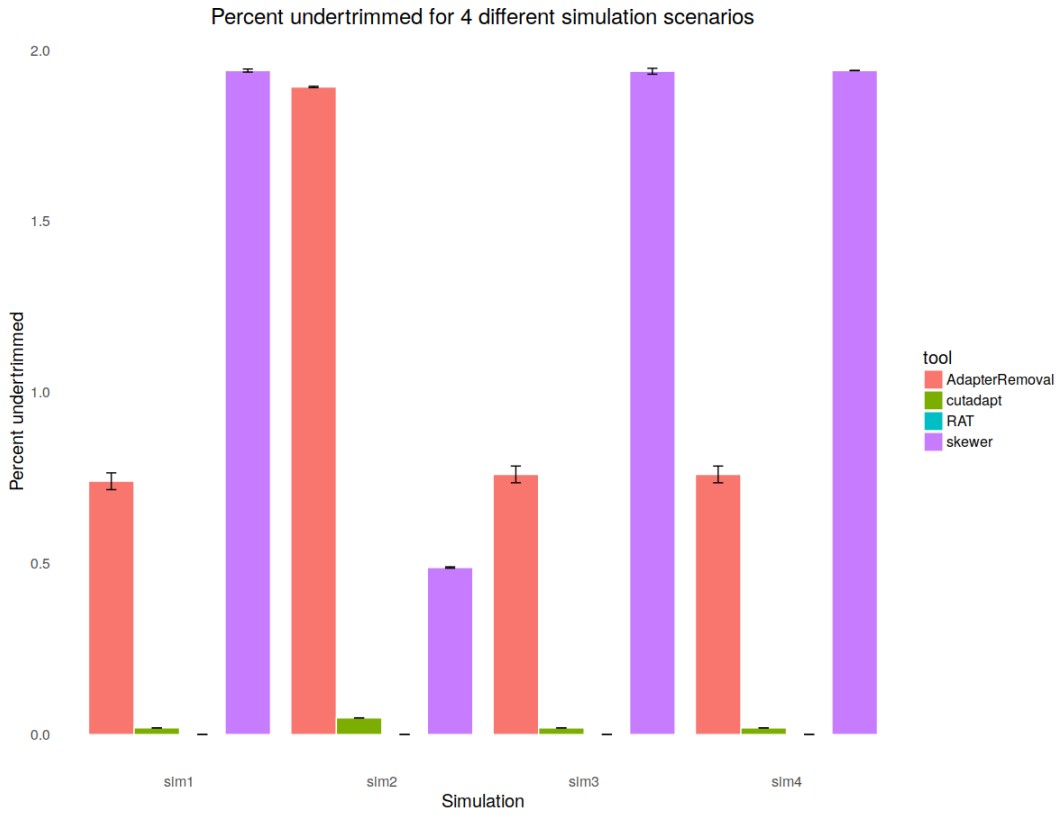

Figure 3.5. Percentage of undertrimmed reads in 4 test cases by RAT, cutadapt, AdapterRemoval and skewer

3.2. Effects of Adapter Trimming on Downstream Analysis

In order to test the effects of adapter trimming on downstream NGS data analysis steps, RAT was used on real NGS datasets and its efficiency was compared to cutadapt, AdapterRemoval and skewer, again. cutadapt, AdapterRemoval and skewer were again run with default parameters. The adapter sequence supplied to them was the one identified by RAT.

16 Illumina sequencing datasets were downloaded from Sequence Read Archive (SRA). The datasets belong to a mouse (*Mus musculus*) transciptome sequencing project (NCBI Bioproject: PRJNA66167). The accessions for retrieved datasets range from SRR3192188 to SRR3192203. Each dataset had around 5 million reads of length 100 nt.

The datasets were cleaned from adapters using RAT and cutadapt, AdapterRemoval, skewer in default settings, supplying the reference algorithms with full length adapters identified by RAT. The identified adapters from the datasets is ”CTGTCTCTTATACACATCTCCGAGCCCACGAGACTAAGGCGAATCTCGTAT”. It is a patented Illumina seqeunce (BAAS et al., 2012), used in library preparation for Illumina instruments.

Both raw data (without any trimming) and datasets after trimming by 4 algorithms mentioned were mapped back to *Mus musculus* reference genome (GRCm38) by Tophat (Kim et al., 2013), and number of reads that mapped successfully were recorded in each step.

The mapping rates in all datasets have shown an increase after adapter trimming. The rates at which RAT improved mapping rates were again similar to that of cutadapt, AdapterRemoval and skewer (Figure 3.6; although less because of RAT's current incapability to trim reads with mismatches in adapter sequences.

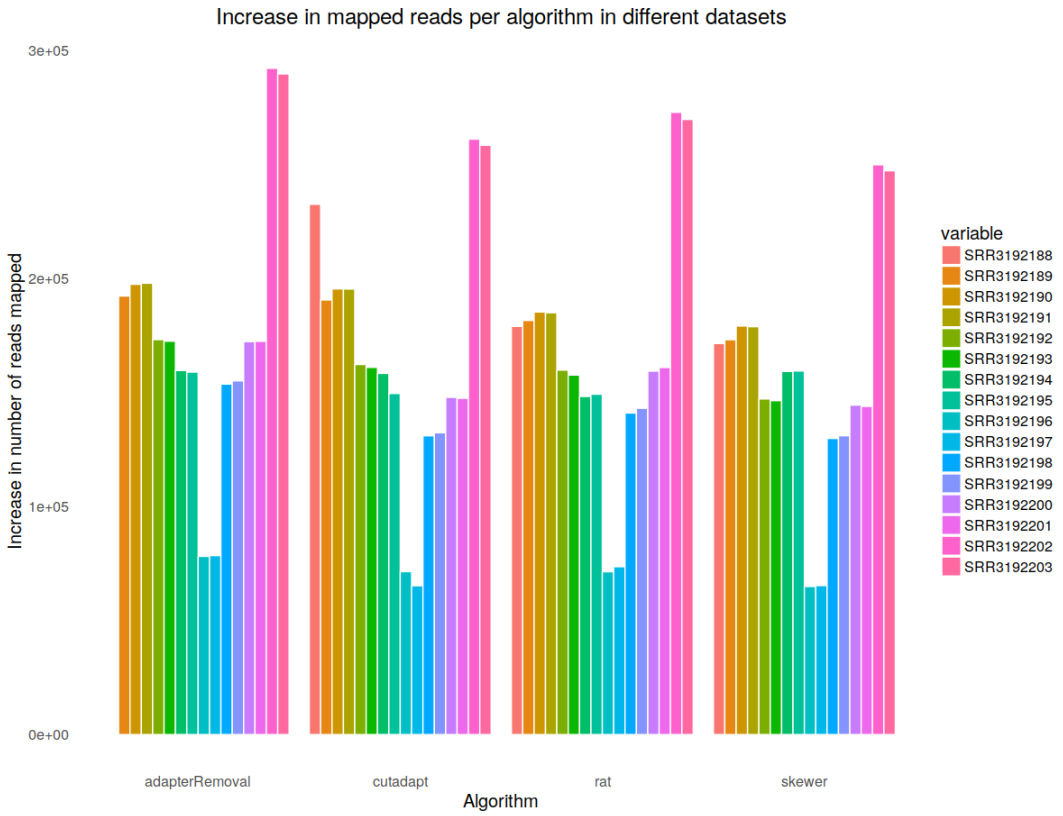

Figure 3.6. Number of reads recovered from the raw data by adapter trimming. 4 different algorithms were used to test the effect of adapter trimming.

Figure 3.7 shows the distribution of number of recovered reads in mapping by adapter trimming. The performance of RAT was again similar to all 3 algorithms it was compared to. Table 3.2 also shows that p-values of the difference in means (Student's t-Test) were not significant enough to say that one algorithm better than another in terms of number of reads recovered.

Another dataset that was tested to evaluate the effect of adapter trimming and compare RAT's performance to cutadapt, AdapterRemoval and skewer was a microRNA profiling dataset for human *Homo sapiens* embryonic stem cells (SRA: SRR026762). It contained around 5 million reads of length 36 nt. microRNAs are short RNA molecules of length 18 to 30 nt. This means that for true microRNAs in the sample, all reads must contain the adapter sequence as the read length was 36 nt. Bowtie (Langmead et al., 2009) mapping algorithm was used to map short microRNA reads back to human genome (GRCh38). The adapter contaminations were removed from the datasets following the

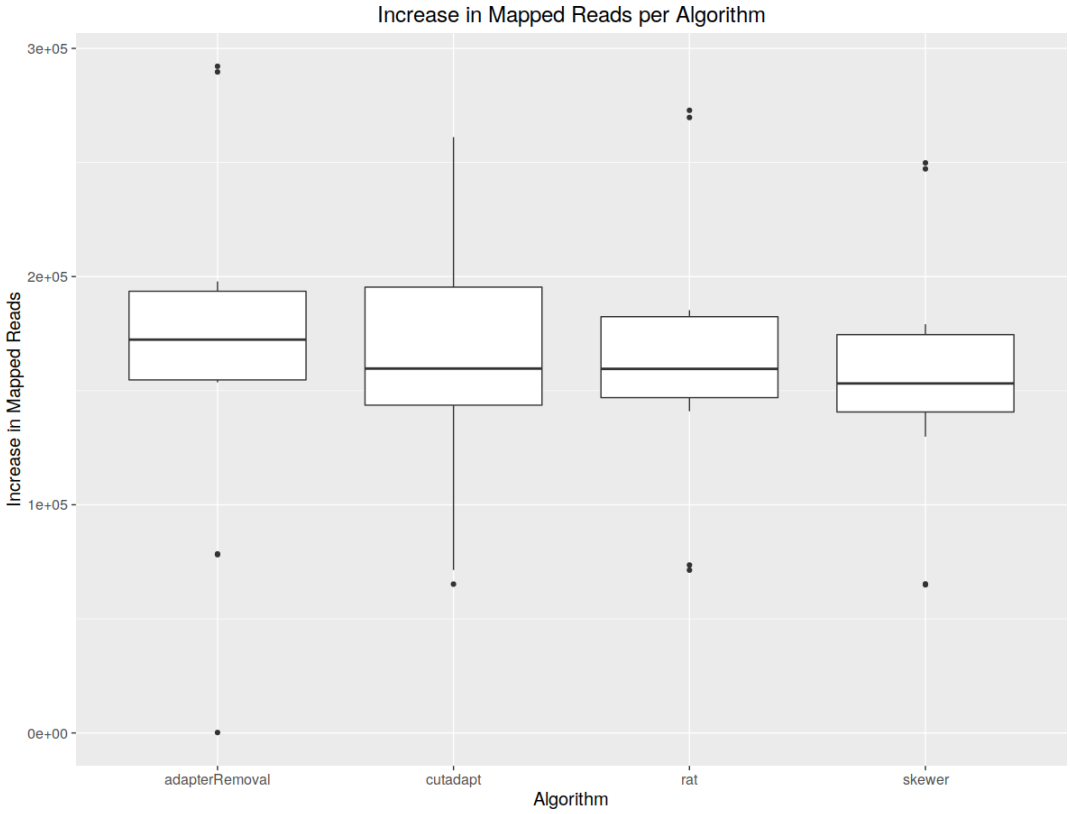

Figure 3.7. Distribution of number of reads recovered by adapter trimming for 4 algorithms tested.

same procedure as in mouse transcriptome dataset.

Raw reads, without any trimming, showed a very low mapping rate around 3%. As shown in Figure 3.8, removing the contaminations increased this from 46% up to 67%. RAT, again has performed on a similar level to other 3 algorithms.

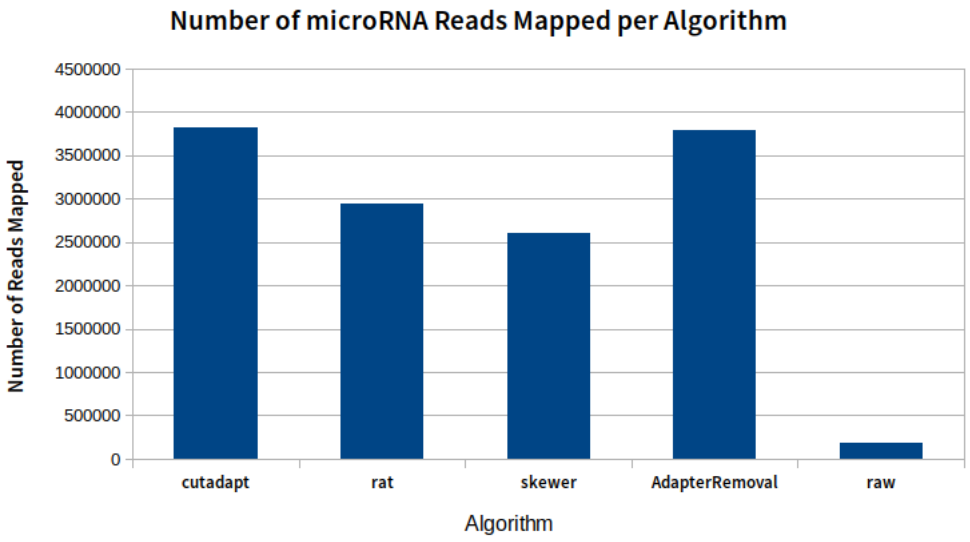

Figure 3.8. Number of reads recovered by adapter trimming for the microRNA sequencing dataset by 4 algorithms tested.

Table 3.1. p-Values for Welch Two Sample t-Test for the null hypothesis that the means of runtimes are equal for RAT and reference tools.

	cutadapt	AdapterRemoval	skewer
sim1	2e-10	2e-7	0.04
sim2	2.8e-10	0.004	0.03
sim3	2e-5	0.9	6.9e-5
sim4	1.8e-4	7e-5	0.33

Table 3.2. P-values for Student's t-Test on the increase in number of mapped reads for mouse transcriptome project datasets.

	cutadapt	AdapterRemoval	skewer
RAT	0.94	0.99	0.62

CHAPTER 4

CONCLUSION

The need for pre-processing of next-generation sequencing reads have been shown multiple times before (Del Fabbro et al., 2013) (Chen et al., 2014). It is a known fact that adapter contaminations and low quality regions in NGS reads disrupt resequencing studies based on mapping to reference genomes or transcriptomes or *de novo* assemblies of genomes and transcriptomes. Although, a crucial step in MPS data analysis, the sequence of adapters used in library preparation are most of the time not available to the bioinformatician, regardless of the dataset being obtained from an online resource, such as SRA or through a sequencing service.

In this work, a novel algorithm (RAT) was offered to tackle this problem and identify adapters used in an MPS run *de novo*; using the sequences of reads themselves. This was accomplished placing all reads from a single sequencing run onto a reversed radix tree and finding suffices which are common in a majority of the reads.

The use of speed-efficient radix trees has allowed RAT to perform on a similar or even better level than similarity based adapter trimming approaches (e.g. cutadapt, AdapterRemoval, skewer) in terms of speed and efficiency and it made RAT distinct from all other adapter trimming tools that have been offered so far (see. Section: Current Methodologies) in terms of not needing prior knowledge of the adapter sequence used in library preparation. The efficiency of RAT was also better in simulated datasets (when there was no sequencing errors) than similarity based approaches and it was comparable in real sequencing data as well.

CHAPTER 5

FURTHER WORK

Further work planned to improve the functionality of RAT is to enable it to trim adapters from reads when a sequencing error occurs. This would improve its efficiency to the level of, or even further than, current state-of-art methods. Another planned feature is to offer quality trimming along with adapter trimming directly on the radix tree. Since the qualities have to be stored on radix tree in order to produce the fastq file at the end, this would not affect the runtime and memory requirement considerably; and statistics, such as the average quality at a region, would be possible to calculate easily directly on the tree since all leaves reachable from a node is always known.

REFERENCES

Adams, M. D., S. E. Celniker, R. A. Holt, C. A. Evans, J. D. Gocayne, P. G. Amanatides, S. E. Scherer, P. W. Li, R. A. Hoskins, R. F. Galle, R. A. George, S. E. Lewis, S. Richards, M. Ashburner, S. N. Henderson, G. G. Sutton, J. R. Wortman, M. D. Yandell, Q. Zhang, L. X. Chen, R. C. Brandon, Y.-H. C. Rogers, R. G. Blazej, M. Champe, B. D. Pfeiffer, K. H. Wan, C. Doyle, E. G. Baxter, G. Helt, C. R. Nelson, G. L. Gabor, Miklos, J. F. Abril, A. Agbayani, H.-J. An, C. Andrews-Pfannkoch, D. Baldwin, R. M. Ballew, A. Basu, J. Baxendale, L. Bayraktaroglu, E. M. Beasley, K. Y. Beeson, P. V. Benos, B. P. Berman, D. Bhandari, S. Bolshakov, D. Borkova, M. R. Botchan, J. Bouck, P. Brokstein, P. Brottier, K. C. Burtis, D. A. Busam, H. Butler, E. Cadieu, A. Center, I. Chandra, J. M. Cherry, S. Cawley, C. Dahlke, L. B. Davenport, P. Davies, B. d. Pablos, A. Delcher, Z. Deng, A. D. Mays, I. Dew, S. M. Dietz, K. Dodson, L. E. Doup, M. Downes, S. Dugan-Rocha, B. C. Dunkov, P. Dunn, K. J. Durbin, C. C. Evangelista, C. Ferraz, S. Ferriera, W. Fleischmann, C. Fosler, A. E. Gabrielian, N. S. Garg, W. M. Gelbart, K. Glasser, A. Glodek, F. Gong, J. H. Gorrell, Z. Gu, P. Guan, M. Harris, N. L. Harris, D. Harvey, T. J. Heiman, J. R. Hernandez, J. Houck, D. Hostin, K. A. Houston, T. J. Howland, M.-H. Wei, C. Ibegwam, M. Jalali, F. Kalush, G. H. Karpen, Z. Ke, J. A. Kennison, K. A. Ketchum, B. E. Kimmel, C. D. Kodira, C. Kraft, S. Kravitz, D. Kulp, Z. Lai, P. Lasko, Y. Lei, A. A. Levitsky, J. Li, Z. Li, Y. Liang, X. Lin, X. Liu, B. Mattei, T. C. McIntosh, M. P. McLeod, D. McPherson, G. Merkulov, N. V. Milshina, C. Mobarry, J. Morris, A. Moshrefi, S. M. Mount, M. Moy, B. Murphy, L. Murphy, D. M. Muzny, D. L. Nelson, D. R. Nelson, K. A. Nelson, K. Nixon, D. R. Nusskern, J. M. Pacleb, M. Palazzolo, G. S. Pittman, S. Pan, J. Pollard, V. Puri, M. G. Reese, K. Reinert, K. Remington, R. D. C. Saunders, F. Scheeler, H. Shen, B. C. Shue, I. SidÃ©n-Kiamos, M. Simpson, M. P. Skupski, T. Smith, E. Spier, A. C. Spradling, M. Stapleton, R. Strong, E. Sun, R. Svirskas, C. Tector, R. Turner, E. Venter, A. H. Wang, X. Wang, Z.-Y. Wang, D. A. Wassarman, G. M. Weinstock, J. Weissenbach, S. M. Williams, T. Woodage, K. C. Worley, D. Wu, S. Yang, Q. A. Yao, J. Ye, R.-F. Yeh, J. S. Zaveri, M. Zhan, G. Zhang, Q. Zhao, L. Zheng, X. H. Zheng, F. N. Zhong, W. Zhong, X. Zhou, S. Zhu, X. Zhu, H. O. Smith, R. A. Gibbs, E. W. Myers, G. M. Rubin, and J. C. Venter (2000, March). The Genome Sequence of Drosophila melanogaster. *Science 287*(5461), 2185–2195.

Aho, A. V., J. E. Hopcroft, and J. Ullman (1983). *Data Structures and Algorithms* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

BAAS, B., I. GORYSHIN, M. Maffitt, and R. Vaidyanathan (2012, August 2). Oligonucleotide replacement for di-tagged and directional libraries. WO Patent App. PCT/US2012/023,139.

Blattner, F. R., G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, M. Riley, J. Collado-Vides, J. D. Glasner, C. K. Rode, G. F. Mayhew, J. Gregor, N. W. Davis, H. A. Kirkpatrick, M. A. Goeden, D. J. Rose, B. Mau, and Y. Shao (1997, September). The Complete Genome Sequence of Escherichia coli K-12. *Science 277*(5331), 1453–1462.

Bolger, A. M., M. Lohse, and B. Usadel (2014). Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics*.

Chen, C., S. S. Khaleel, H. Huang, and C. H. Wu (2014). Software for pre-processing illumina next-generation sequencing short read sequences. *Source Code for Biology and Medicine 9*(1), 1–11.

Cock, P. J., C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice (2010). The sanger fastq file format for sequences with quality scores, and the solexa/illumina fastq variants. *Nucleic acids research 38*(6), 1767–1771.

Criscuolo, A. and S. Brisse (2013). Alientrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. *Genomics 102*(5), 500–506.

Del Fabbro, C., S. Scalabrin, M. Morgante, and F. M. Giorgi (2013, 12). An extensive evaluation of read trimming effects on illumina ngs data analysis. *PLoS ONE 8*(12).

Dodt, M., J. T. Roehr, R. Ahmed, and C. Dieterich (2012). Flexbar - flexible barcode and adapter processing for next-generation sequencing platforms. *Biology 1*(3), 895–905.

Dohm, J. C., C. Lottaz, T. Borodina, and H. Himmelbauer (2008, September). Substantial

biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research 36*(16), e105.

Fabbro, C. D., S. Scalabrin, M. Morgante, and F. M. Giorgi (2013, December). An Extensive Evaluation of Read Trimming Effects on Illumina NGS Data Analysis. *PLOS ONE 8*(12), e85024.

Falgueras, J., A. J. Lara, N. Fernandez-Pozo, F. R. Canton, G. Perez-Trabado, and M. G. Claros (2010). Seqtrim: a high-throughput pipeline for pre-processing any type of sequence read. *BMC Bioinformatics 11*(1), 1–12.

Fiers, W., R. Contreras, F. Duerinck, G. Haegeman, D. Iserentant, J. Merregaert, W. Min Jou, F. Molemans, A. Raeymaekers, A. Van den Berghe, G. Volckaert, and M. Ysebaert (1976, apr). Complete nucleotide sequence of bacteriophage MS2 RNA: primary and secondary structure of the replicase gene. *Nature 260*(5551), 500–507.

Fuller, C. W., L. R. Middendorf, S. A. Benner, G. M. Church, T. Harris, X. Huang, S. B. Jovanovich, J. R. Nelson, J. A. Schloss, D. C. Schwartz, and D. V. Vezenov (2009, November). The challenges of sequencing by synthesis. *Nature Biotechnology 27*(11), 1013–1023.

Goffeau, A., B. G. Barrell, H. Bussey, R. W. Davis, B. Dujon, H. Feldmann, F. Galibert, J. D. Hoheisel, C. Jacq, M. Johnston, E. J. Louis, H. W. Mewes, Y. Murakami, P. Philippsen, H. Tettelin, and S. G. Oliver (1996, October). Life with 6000 genes. *Science (New York, N.Y.) 274*(5287), 546, 563–567.

Gordon, A. and G. Hannon (2010). Fastx-toolkit. fastq/a short-reads pre-processing tools. *Unpublished Available online at: http://hannonlab. cshl. edu/fastx_toolkit*.

Guarnaccia, M., G. Gentile, E. Alessi, C. Schneider, S. Petralia, and S. Cavallaro (2014). Is this the real time for genomics? *Genomics 103*(2-3), 177–182.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. New York, NY, USA: Cambridge University Press.

Hansen, K. D., S. E. Brenner, and S. Dudoit (2010, July). Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Research 38*(12), e131.

Hayden, E. C. (2014). Technology: the $1,000 genome. *Nature 507*(7492), 294–295.

Jiang, H., R. Lei, S.-W. Ding, and S. Zhu (2014a). Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics 15*, 182.

Jiang, H., R. Lei, S.-W. Ding, and S. Zhu (2014b). Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC bioinformatics 15*(1), 1.

Kim, D., G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg (2013). Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology 14*(4), 1.

Kircher, M. and J. Kelso (2010). High-throughput dna sequencing–concepts and limitations. *Bioessays 32*(6), 524–536.

Kong, Y. (2011). Btrim: A fast, lightweight adapter and quality trimming program for next-generation sequencing technologies. *Genomics 98*(2), 152 – 153.

Lander, E. S., L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, R. Funke, D. Gage, K. Harris, A. Heaford, J. Howland, L. Kann, J. Lehoczky, R. LeVine, P. McEwan, K. McKernan, J. Meldrim, J. P. Mesirov, C. Miranda, W. Morris, J. Naylor, C. Raymond, M. Rosetti, R. Santos, A. Sheridan, C. Sougnez, N. Stange-Thomann, N. Stojanovic, A. Subramanian, D. Wyman, J. Rogers, J. Sulston, R. Ainscough, S. Beck, D. Bentley, J. Burton, C. Clee, N. Carter, A. Coulson, R. Deadman, P. Deloukas, A. Dunham, I. Dunham, R. Durbin, L. French, D. Grafham, S. Gregory, T. Hubbard, S. Humphray, A. Hunt, M. Jones, C. Lloyd, A. McMurray, L. Matthews, S. Mercer, S. Milne, J. C. Mullikin, A. Mungall, R. Plumb, M. Ross, R. Shownkeen, S. Sims, R. H. Waterston, R. K. Wilson, L. W. Hillier, J. D. McPherson, M. A. Marra, E. R. Mardis, L. A. Ful-

ton, A. T. Chinwalla, K. H. Pepin, W. R. Gish, S. L. Chissoe, M. C. Wendl, K. D. Delehaunty, T. L. Miner, A. Delehaunty, J. B. Kramer, L. L. Cook, R. S. Fulton, D. L. Johnson, P. J. Minx, S. W. Clifton, T. Hawkins, E. Branscomb, P. Predki, P. Richardson, S. Wenning, T. Slezak, N. Doggett, J.-F. Cheng, A. Olsen, S. Lucas, C. Elkin, E. Uberbacher, M. Frazier, R. A. Gibbs, D. M. Muzny, S. E. Scherer, J. B. Bouck, E. J. Sodergren, K. C. Worley, C. M. Rives, J. H. Gorrell, M. L. Metzker, S. L. Naylor, R. S. Kucherlapati, D. L. Nelson, G. M. Weinstock, Y. Sakaki, A. Fujiyama, M. Hattori, T. Yada, A. Toyoda, T. Itoh, C. Kawagoe, H. Watanabe, Y. Totoki, T. Taylor, J. Weissenbach, R. Heilig, W. Saurin, F. Artiguenave, P. Brottier, T. Bruls, E. Pelletier, C. Robert, P. Wincker, A. Rosenthal, M. Platzer, G. Nyakatura, S. Taudien, A. Rump, D. R. Smith, L. Doucette-Stamm, M. Rubenfield, K. Weinstock, H. M. Lee, J. Dubois, H. Yang, J. Yu, J. Wang, G. Huang, J. Gu, L. Hood, L. Rowen, A. Madan, S. Qin, R. W. Davis, N. A. Federspiel, A. P. Abola, M. J. Proctor, B. A. Roe, F. Chen, H. Pan, J. Ramser, H. Lehrach, R. Reinhardt, W. R. McCombie, M. d. l. Bastide, N. Dedhia, H. BlÃ¶cker, K. Hornischer, G. Nordsiek, R. Agarwala, L. Aravind, J. A. Bailey, A. Bateman, S. Batzoglou, E. Birney, P. Bork, D. G. Brown, C. B. Burge, L. Cerutti, H.-C. Chen, D. Church, M. Clamp, R. R. Copley, T. Doerks, S. R. Eddy, E. E. Eichler, T. S. Furey, J. Galagan, J. G. R. Gilbert, C. Harmon, Y. Hayashizaki, D. Haussler, H. Hermjakob, K. Hokamp, W. Jang, L. S. Johnson, T. A. Jones, S. Kasif, A. Kaspryzk, S. Kennedy, W. J. Kent, P. Kitts, E. V. Koonin, I. Korf, D. Kulp, D. Lancet, T. M. Lowe, A. McLysaght, T. Mikkelsen, J. V. Moran, N. Mulder, V. J. Pollara, C. P. Ponting, G. Schuler, J. Schultz, G. Slater, A. F. A. Smit, E. Stupka, J. Szustakowki, D. Thierry-Mieg, J. Thierry-Mieg, L. Wagner, J. Wallis, R. Wheeler, A. Williams, Y. I. Wolf, K. H. Wolfe, S.-P. Yang, R.-F. Yeh, F. Collins, M. S. Guyer, J. Peterson, A. Felsenfeld, K. A. Wetterstrand, R. M. Myers, J. Schmutz, M. Dickson, J. Grimwood, D. R. Cox, M. V. Olson, R. Kaul, C. Raymond, N. Shimizu, K. Kawasaki, S. Minoshima, G. A. Evans, M. Athanasiou, R. Schultz, A. Patrinos, and M. J. Morgan (2001, February). Initial sequencing and analysis of the human genome. *Nature 409*(6822), 860–921.

Langmead, B., C. Trapnell, M. Pop, and S. L. Salzberg (2009). Ultrafast and memory-efficient alignment of short dna sequences to the human genome. *Genome biology 10*(3), 1.

Ledergerber, C. and C. Dessimoz (2011). Base-calling for next-generation sequencing platforms. *Briefings in bioinformatics*, bbq077.

Leggett, R. M., B. J. Clavijo, L. Clissold, M. D. Clark, and M. Caccamo (2013). Nextclip: an analysis and read preparation tool for nextera long mate pair libraries. *Bioinformatics*, btt702.

Lindgreen, S. (2012). Adapterremoval: easy cleaning of next-generation sequencing reads. *BMC research notes 5*(1), 1.

Liu, L., Y. Li, S. Li, N. Hu, Y. He, R. Pong, D. Lin, L. Lu, and M. Law (2012). Comparison of next-generation sequencing systems. *BioMed Research International 2012*.

Mardis, E. R. (2008a). The impact of next-generation sequencing technology on genetics. *Trends in Genetics 24*(3), 133 – 141.

Mardis, E. R. (2008b). Next-Generation DNA Sequencing Methods. *Annual Review of Genomics and Human Genetics 9*(1), 387–402.

Mardis, E. R. (2013). Next-generation sequencing platforms. *Annual review of analytical chemistry 6*, 287–303.

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal 17*(1).

Maxam, A. M. and W. Gilbert (1977, February). A new method for sequencing DNA. *Proceedings of the National Academy of Sciences of the United States of America 74*(2), 560–564.

Metzker, M. L. (2010, January). Sequencing technologies - the next generation. *Nature Reviews. Genetics 11*(1), 31–46.

Min Jou, W., G. Haegeman, M. Ysebaert, and W. Fiers (1972, may). Nucleotide sequence of the gene coding for the bacteriophage MS2 coat protein. *Nature 237*(5350), 82–8.

Sanger, F., S. Nicklen, and A. R. Coulson (1977, December). DNA sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America 74*(12), 5463–5467.

Sawyer, S., J. Krause, K. Guschanski, V. Savolainen, and S. Paabo (2012). Temporal patterns of nucleotide misincorporations and DNA fragmentation in ancient DNA. *PloS One 7*(3), e34131.

Schmieder, R., Y. W. Lim, F. Rohwer, and R. Edwards (2010). Tagcleaner: Identification and removal of tag sequences from genomic and metagenomic datasets. *BMC Bioinformatics 11*(1), 1–14.

Schuster, S. C. (2008, January). Next-generation sequencing transforms today's biology. *Nature Methods 5*(1), 16–18.

Shendure, J. and H. Ji (2008, October). Next-generation DNA sequencing. *Nature Biotechnology 26*(10), 1135–1145.

van Dijk, E. L., H. Auger, Y. Jaszczyszyn, and C. Thermes (2014, September). Ten years of next-generation sequencing technology. *Trends in Genetics 30*(9), 418–426.

Venter, J. C., M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, J. D. Gocayne, P. Amanatides, R. M. Ballew, D. H. Huson, J. R. Wortman, Q. Zhang, C. D. Kodira, X. H. Zheng, L. Chen, M. Skupski, G. Subramanian, P. D. Thomas, J. Zhang, G. L. G. Miklos, C. Nelson, S. Broder, A. G. Clark, J. Nadeau, V. A. McKusick, N. Zinder, A. J. Levine, R. J. Roberts, M. Simon, C. Slayman, M. Hunkapiller, R. Bolanos, A. Delcher, I. Dew, D. Fasulo, M. Flanigan, L. Florea, A. Halpern, S. Hannenhalli, S. Kravitz, S. Levy, C. Mobarry, K. Reinert, K. Remington, J. Abu-Threideh, E. Beasley, K. Biddick, V. Bonazzi, R. Brandon, M. Cargill, I. Chandramouliswaran, R. Charlab, K. Chaturvedi, Z. Deng, V. D. Francesco, P. Dunn, K. Eilbeck, C. Evangelista, A. E. Gabrielian, W. Gan, W. Ge, F. Gong, Z. Gu, P. Guan, T. J. Heiman, M. E. Higgins, R.-R. Ji, Z. Ke, K. A. Ketchum, Z. Lai, Y. Lei, Z. Li, J. Li, Y. Liang, X. Lin, F. Lu, G. V. Merkulov, N. Milshina, H. M. Moore, A. K. Naik, V. A. Narayan, B. Neelam, D. Nusskern, D. B. Rusch, S. Salzberg, W. Shao, B. Shue, J. Sun, Z. Y. Wang,

A. Wang, X. Wang, J. Wang, M.-H. Wei, R. Wides, C. Xiao, C. Yan, A. Yao, J. Ye, M. Zhan, W. Zhang, H. Zhang, Q. Zhao, L. Zheng, F. Zhong, W. Zhong, S. C. Zhu, S. Zhao, D. Gilbert, S. Baumhueter, G. Spier, C. Carter, A. Cravchik, T. Woodage, F. Ali, H. An, A. Awe, D. Baldwin, H. Baden, M. Barnstead, I. Barrow, K. Beeson, D. Busam, A. Carver, A. Center, M. L. Cheng, L. Curry, S. Danaher, L. Davenport, R. Desilets, S. Dietz, K. Dodson, L. Doup, S. Ferriera, N. Garg, A. Gluecksmann, B. Hart, J. Haynes, C. Haynes, C. Heiner, S. Hladun, D. Hostin, J. Houck, T. Howland, C. Ibegwam, J. Johnson, F. Kalush, L. Kline, S. Koduru, A. Love, F. Mann, D. May, S. McCawley, T. McIntosh, I. McMullen, M. Moy, L. Moy, B. Murphy, K. Nelson, C. Pfannkoch, E. Pratts, V. Puri, H. Qureshi, M. Reardon, R. Rodriguez, Y.-H. Rogers, D. Romblad, B. Ruhfel, R. Scott, C. Sitter, M. Smallwood, E. Stewart, R. Strong, E. Suh, R. Thomas, N. N. Tint, S. Tse, C. Vech, G. Wang, J. Wetter, S. Williams, M. Williams, S. Windsor, E. Winn-Deen, K. Wolfe, J. Zaveri, K. Zaveri, J. F. Abril, R. Guigo, M. J. Campbell, K. V. Sjolander, B. Karlak, A. Kejariwal, H. Mi, B. Lazareva, T. Hatton, A. Narechania, K. Diemer, A. Muruganujan, N. Guo, S. Sato, V. Bafna, S. Istrail, R. Lippert, R. Schwartz, B. Walenz, S. Yooseph, D. Allen, A. Basu, J. Baxendale, L. Blick, M. Caminha, J. Carnes-Stine, P. Caulk, Y.-H. Chiang, M. Coyne, C. Dahlke, A. D. Mays, M. Dombroski, M. Donnelly, D. Ely, S. Esparham, C. Fosler, H. Gire, S. Glanowski, K. Glasser, A. Glodek, M. Gorokhov, K. Graham, B. Gropman, M. Harris, J. Heil, S. Henderson, J. Hoover, D. Jennings, C. Jordan, J. Jordan, J. Kasha, L. Kagan, C. Kraft, A. Levitsky, M. Lewis, X. Liu, J. Lopez, D. Ma, W. Majoros, J. McDaniel, S. Murphy, M. Newman, T. Nguyen, N. Nguyen, M. Nodell, S. Pan, J. Peck, M. Peterson, W. Rowe, R. Sanders, J. Scott, M. Simpson, T. Smith, A. Sprague, T. Stockwell, R. Turner, E. Venter, M. Wang, M. Wen, D. Wu, M. Wu, A. Xia, A. Zandieh, and X. Zhu (2001, February). The Sequence of the Human Genome. *Science 291*(5507), 1304–1351.

Weiner, P. (1973). Linear pattern matching algorithms. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory (Swat 1973)*, SWAT '73, Washington, DC, USA, pp. 1–11. IEEE Computer Society.

Williams, C. R., A. Baccarella, J. Z. Parrish, and C. C. Kim (2016, February). Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC Bioinformatics 17*.

Zhong, S., J.-G. Joung, Y. Zheng, Y. Chen, B. Liu, Y. Shao, J. Z. Xiang, Z. Fei, and J. J. Giovannoni (2011). High-throughput illumina strand-specific rna sequencing library preparation. *Cold Spring Harb Protoc 2011*(8), 940–9.

APPENDIX A

SIMULATION TESTS

Table A.1. Features of Simulation Datasets

Dataset	Depth	Read Length	Avg Frag Len	Frag Len Dev
sim1.r1	1000000	100	100	50
sim1.r2	1000000	100	100	50
sim1.r3	1000000	100	100	50
sim1.r4	1000000	100	100	50
sim1.r5	1000000	100	100	50
sim2.r1	1000000	50	50	20
sim2.r2	1000000	50	50	20
sim2.r3	1000000	50	50	20
sim2.r4	1000000	50	50	20
sim2.r5	1000000	50	50	20
sim3.r1	1000000	250	250	50
sim3.r2	1000000	250	250	50
sim3.r3	1000000	250	250	50
sim3.r4	1000000	250	250	50
sim3.r5	1000000	250	250	50
sim4.r1	10000000	100	100	50
sim4.r2	10000000	100	100	50
sim4.r3	10000000	100	100	50
sim4.r4	10000000	100	100	50
sim4.r5	10000000	100	100	50

Table A.2. Runtimes in seconds of tested algorithms on simulated datasets.

Dataset	RAT	cutadapt	AdapterRemoval	skewer
sim1.r1	9.54	17.30	3.13	12.15
sim1.r2	9.50	17.36	3.25	9.21
sim1.r3	9.27	17.34	2.91	11.73
sim1.r4	9.63	17.46	3.18	12.17
sim1.r5	9.41	17.36	3.81	14.13
sim2.r1	8.89	11.98	3.23	6.91
sim2.r2	8.57	11.73	6.81	7.28
sim2.r3	8.84	11.85	1.89	9.68
sim2.r4	8.71	11.71	1.91	7.34
sim2.r5	8.67	11.67	1.82	7.31
sim3.r1	15.84	32.21	33.3	27.70
sim3.r2	11.24	31.89	14.13	27.25
sim3.r3	12.13	31.57	7.14	27.31
sim3.r4	12.77	31.65	7.15	27.35
sim3.r5	14.70	31.92	7.14	27.15
sim4.r1	118.45	172.75	39.6	111.14
sim4.r2	124.10	172.89	30.00	93.95
sim4.r3	102.90	171.94	64.30	119.75
sim4.r4	121.27	170.92	56.80	119.85
sim4.r5	111.93	171.51	57.68	99.57

Table A.3. Efficiency of RAT on simulation tests

sim	tool	Correct %	Overtrimmed %	Undertrimmed %
sim1	RAT	83.7087	16.2913	0
sim1	RAT	83.8144	16.1856	0
sim1	RAT	83.7108	16.2892	0
sim1	RAT	83.6923	16.3077	0
sim1	RAT	83.7843	16.2157	0
sim2	RAT	83.1549	16.8451	0
sim2	RAT	83.1517	16.8483	0
sim2	RAT	83.2044	16.7956	0
sim2	RAT	83.1955	16.8045	0
sim2	RAT	83.2127	16.7873	0
sim3	RAT	83.7207	16.2793	0
sim3	RAT	83.7167	16.2833	0
sim3	RAT	83.6531	16.3469	0
sim3	RAT	83.7338	16.2662	0
sim3	RAT	83.6863	16.3137	0
sim4	RAT	83.72011	16.27989	0
sim4	RAT	83.75837	16.24163	0
sim4	RAT	83.75173	16.24827	0
sim4	RAT	83.72045	16.27955	0
sim4	RAT	83.713	16.287	0

Table A.4. Efficiency of cutadapt on simulation tests

sim	tool	Correct %	Overtrimmed %	Undertrimmed %
sim1	cutadapt	96.93	1.14	0.019
sim1	cutadapt	96.93	1.13	0.019
sim1	cutadapt	96.93	1.13	0.019
sim1	cutadapt	96.92	1.13	0.020
sim1	cutadapt	96.94	1.13	0.019
sim2	cutadapt	93.91	1.25	0.048
sim2	cutadapt	93.89	1.26	0.049
sim2	cutadapt	93.90	1.24	0.049
sim2	cutadapt	93.87	1.25	0.049
sim2	cutadapt	93.88	1.26	0.049
sim3	cutadapt	96.94	1.13	0.019
sim3	cutadapt	96.94	1.14	0.019
sim3	cutadapt	96.89	1.15	0.019
sim3	cutadapt	96.94	1.12	0.019
sim3	cutadapt	96.96	1.12	0.019
sim4	cutadapt	96.93	1.13	0.019
sim4	cutadapt	96.93	1.13	0.019
sim4	cutadapt	96.93	1.12	0.019
sim4	cutadapt	96.93	1.13	0.019
sim4	cutadapt	96.93	1.13	0.019

Table A.5. Efficiency of AdapterRemoval on simulated tests

sim	tool	Correct	Overtrimmed	Undertrimmed
sim1	AdapterRemoval	89.4	9.8	0.7
sim1	AdapterRemoval	89.6	9.7	0.7
sim1	AdapterRemoval	89.6	9.7	0.7
sim1	AdapterRemoval	89.5	9.7	0.8
sim1	AdapterRemoval	89.5	9.7	0.8
sim2	AdapterRemoval	49.4	48.8	1.89
sim2	AdapterRemoval	49.4	48.8	1.89
sim2	AdapterRemoval	49.5	48.6	1.9
sim2	AdapterRemoval	49.4	48.7	1.89
sim2	AdapterRemoval	49.4	48.7	1.89
sim3	AdapterRemoval	89.5	9.7	0.8
sim3	AdapterRemoval	89.5	9.8	0.7
sim3	AdapterRemoval	89.4	9.8	0.8
sim3	AdapterRemoval	89.5	9.7	0.8
sim3	AdapterRemoval	89.5	9.8	0.7
sim4	AdapterRemoval	89.5	9.7	0.8
sim4	AdapterRemoval	89.5	9.8	0.7
sim4	AdapterRemoval	89.4	9.8	0.8
sim4	AdapterRemoval	89.5	9.7	0.8
sim4	AdapterRemoval	89.5	9.8	0.7

Table A.6. Efficiency of skewer on simulated datasets

sim	tool	Correct %	Overtrimmed %	Undertrimmed %
sim1	skewer	97	1.13	1.93
sim1	skewer	97	1.12	1.94
sim1	skewer	97	1.12	1.95
sim1	skewer	97	1.12	1.95
sim1	skewer	97	1.13	1.93
sim2	skewer	47.4	47.8	0.48
sim2	skewer	47.4	47.8	0.49
sim2	skewer	47.5	47.7	0.49
sim2	skewer	47.4	47.8	0.49
sim2	skewer	47.5	47.7	0.49
sim3	skewer	97	1.13	1.93
sim3	skewer	97	1.13	1.92
sim3	skewer	97	1.14	1.97
sim3	skewer	97	1.12	1.94
sim3	skewer	97	1.11	1.93
sim4	skewer	97	1.13	1.94
sim4	skewer	97	1.13	1.94
sim4	skewer	97	1.13	1.94
sim4	skewer	97	1.14	1.94
sim4	skewer	97	1.14	1.94

APPENDIX B

MOUSE TRANSCRIPTOME DATA

Table B.1. Runtimes and memory usages for mouse transcriptome datasets. Time is in seconds, memory usages is in kilobytes.

	RATTime	**RATMem**	**cutadaptTime**	**cutadaptMem**
SRR3192188	70	12073728	77	10832
SRR3192189	69	12208392	90	10860
SRR3192190	69	11074728	66	10864
SRR3192191	61	10877264	72	10804
SRR3192192	89	12331204	95	10868
SRR3192193	84	12751048	97	10764
SRR3192194	86	12789604	91	10948
SRR3192195	59	11998812	87	10704
SRR3192196	50	8269300	61	10812
SRR3192197	84	9237892	61	10780
SRR3192198	83	12125200	88	10944
SRR3192199	97	10875504	86	10892
SRR3192200	86	12417884	92	10836
SRR3192201	94	14061692	92	10908
SRR3192202	96	13102412	99	10728
SRR3192203	69	13240248	104	10720
	adapterRemovalTime	**adapterRemovalMem**	**skewerTime**	**skewerMem**
SRR3192188	40	7952	55	2384
SRR3192189	34	7884	59	2444
SRR3192190	37	8160	113	2520
SRR3192191	45	7980	61	2584
SRR3192192	57	7940	77	2532
SRR3192193	57	7872	86	2432
SRR3192194	53	7940	85	2576
SRR3192195	54	8088	69	2456
SRR3192196	38	7960	58	2408
SRR3192197	38	7900	53	2424
SRR3192198	51	8028	89	2308
SRR3192199	49	7832	77	2540
SRR3192200	54	7844	76	2404
SRR3192201	53	7948	70	2380
SRR3192202	60	7848	77	2404
SRR3192203	59	8116	70	2412