Master Degree Project

A Comparison of Sensitive Splice Aware Aligners in RNA Sequence Data Analysis in Leaping towards Benchmarking

Master Degree Project in Bioinformatics
30 ECTS, B1758A
Spring 2020

Chizoba Oguchi
b18chiog@student.his.se

Supervisor:Angelica Lindlöf
angelica.lindlof@his.se
Examiner: Björn Olsson
bjorn.olsson@his.se

Abstract

Bioinformatics, as a field, rapidly develops and such development requires the design of algorithms and software. RNA-seq provides robust information on RNAs, both already known and new, hence the increased study of the RNA. Alignment is an important step in downstream analyses and the ability to map reads across splice junctions is a requirement of an aligner to be suitable for mapping RNA-seq reads. Therefore, the necessity for a standard splice-aware aligner. STAR, Rsubread and HISAT2 have not been singly studied for the purpose of benchmarking one of them as a standard aligner for spliced RNA-seq reads. This study compared these aligners, found to be sensitive to splice sites, with regards to their sensitivity to splice sites, performance with default parameter settings and the resource usage during the alignment process. The aligners were matched with *featureCounts*. The results show that STAR and Rsubread outperform HISAT2 in the aspects of sensitivity and default parameter settings. Rsubread was more sensitive to splice junctions than STAR but underperformed with *featureCounts*. STAR had a consistent performance, with more demand on the memory and time resource, but showed it could be more sensitive with real data.

Table of Contents

Abbreviations

BAM	Binary Alignment Map
BLAT	Blast Like Alignment Tool
cDNA	Complementary Deoxyribonucleic Acid
CPU	Central Processing Unit
DNA	Deoxyribonucleic Acid
FM	Ferragina and Manzini (index)
GB	Gigabyte
GOseq	Gene Ontology analysis on RNA-seq data
GRCh	Genome Reference Consortium Human Build
GTF	Gene Transfer Format
HBRR	Human Brain Reference RNA
HISAT2	Hierarchical Indexing for Spliced Alignment of Transcripts
HTSeq-count	High Throughput Sequence count
KEGG	Kyoto Encyclopaedia of Genes and Genomes
mRNA	Messenger Ribonucleic Acid
NCBI	National Centre for Biotechnology Information
NGS	Next-Generation Sequencing
PE	Pair End
RAM	Random Access Memory
Refseq	Reference Sequence
RNA	Ribonucleic Acid
RNA-seq	Ribonucleic Acid sequencing
SAM	Sequence Alignment Map
SEQC	Sequencing Quality Control
SNPs	Single Nucleotide Polymorphisms
STAR	Spliced Transcripts Alignment to a Reference
TB	Terabyte
UHRR	Universal Human Reference RNA

1. Introduction

The study of the central dogma of molecular biology from the middle of the 20th century delved deeper into ribonucleic acid (RNA) leveraging on high throughput technologies. RNA sequencing (RNA-seq) is an accessible method that applies next-generation sequencing (NGS) in observing the structure and activity of the gene on genomic level[1]. The use of this method to further assess the abundance and content of RNA greatly improved gene expression analyses. Unlike the previously used microarray technology, RNA-seq enriched the knowledge base of the existing RNA and led to new knowledge of RNAs due to its ability to detect novel transcripts[2]. This technique made contributions to biological researches by providing a high level of coverage and resolution of the dynamic nature of the transcriptome[3]. It further led to genome-wide analysis of transcription, detection of allele-specific expression, identification of alternatively spliced genes, etc. in addition to gene expression quantification[3]. A paramount characteristic of this cutting edge sequencing method is its ability to rapidly generate data in large volumes[4,5]. The accurate generation and manipulation of the generated data is pivotal to the results and conclusions of researchers after their experiments thereby placing a huge responsibility of ensuring high quality RNA-seq data analysis on bioinformaticians[4,6,7].

1.1 Background

Alignment is a critical and foundational step as well as the most computationally complex and expensive part of nearly all RNA-seq data analysis pipelines[1,5,6,8–12]. This essential step determines the accuracy of downstream analyses (Figure 1). Mapping of the RNA-seq reads can either be to a reference genome or a transcriptome, if a guided-analysis is required, in order to deduce the location from which they originated[1,5,10,13]. This approach is advantageous in that it has higher sensitivity and computational efficiency. There is an alternative alignment, which is done in the absence of a reference genome, where sequenced reads can be assembled and mapped to the assembled transcriptome, which has become a novel transcriptome. However, it has a low performance on alignment-guided analysis[6,13].

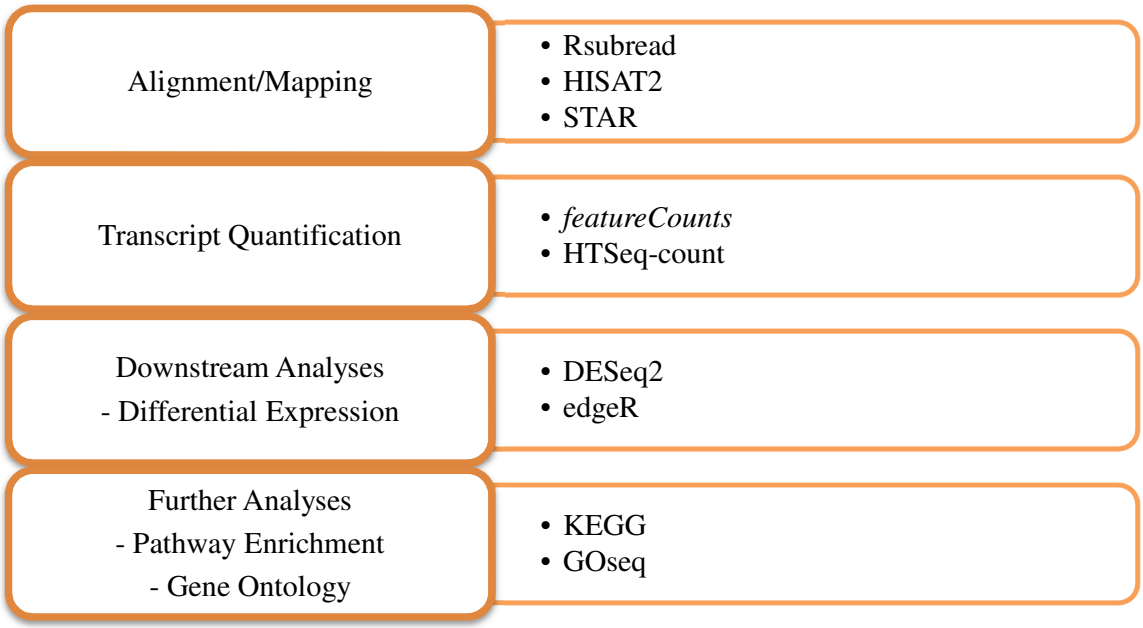

Figure 1: A typical RNA-seq data analysis pipeline and corresponding analysis tools. The analysis input is quality controlled sequenced reads. These reads are mapped to the reference genome using aligners, such as Rsubread, HISAT2 etc. After the alignment, the aligned reads are quantified or counted. In downstream analyses, differentially expressed genes are determined and analysed further to identify enriched annotation terms, such as biological processes and molecular functions.

This beneficial approach of mapping reads to a reference genome comes with its own challenges[6,13]. The presence of splice junctions poses a problem to mapping reads to a reference especially sequencing errors, differences with the reference and identification of fusion transcripts (if they are the aim of the research). Since the eukaryotic genomes contain introns, the RNA-seq aligner should be able to handle the gaps as sequenced reads from mature messenger RNA (mRNA) transcripts do not contain introns[14]. Therefore, selection of aligners for RNA-seq depends on its ability to align reads across splice junctions, handle paired end-reads (as modern sequencers return reads in this form), take on strand-specific data and carry out its processes with computational efficiency[6]. Foremost, RNA-seq alignment challenges like splicing effects creates an increasing demand for software or algorithms that can tackle them effectively[10]. The consequent design of splice aware alignment software in response to the advancement of high throughput technologies requires regular comparison of these tools. Ultimately, this should lead towards a standard tool that will have to undergo continuous revision to tackle future alignment issues. Due to the absence of splice alignment benchmarks, researchers select tools based on the referrals of advanced users and the developers' claims thereby introducing user-sensitivity bias in this basic stage of downstream analysis. This reveals a need not just to ensure that tools are designed and updated but to also set a standard splice aware alignment tool to ensure correctness of results[15].

Earlier studies have compared versions of splice aligners such as ContextMap, CRAC, GSNAP, HISAT, HISAT2, Olego, RUM, SOAPsplice, STAR, Subread, TopHAT, Rsubread but failed to solve the issue of standardizing a tool[6-7,10, 12,14,16-23]. These splice-aware aligners differ in their algorithms (Table 1). Wang et al[24] in his comparison of four popular RNA-seq tools emphasized the non-consistent result of the use of different algorithms. Baruzzo et al[6]

discovered that HISAT2 has good performance on short anchors without annotation and analysis of splice signals in junction calls in human data. The same report discovered that HISAT2 and Subread used the least time in completing the alignment program unlike other tools they were compared with. Hong et al[25] pointed out that Subread should be used solely for quantifying gene expression levels and not in RNA-editing analysis as its performance was quite low. Kim et al[14] revealed that STAR can detect more multi-mapped reads than HISAT2, but the latter showed an overall better performance in spliced alignment for small anchored reads. Kim et al[14] found STAR2 to have better alignment results for 109 million real reads of 101 bp long each of human sample with soft-clipped alignments and realigned soft clip alignments. Liao et al[12] showed that Rsubread was sensitive to splice junctions because it detected more exon-exon junctions than STAR and TopHat2. Williams et al[2] stated that STAR seemed to have a higher sensitivity to splice junctions because it mapped a greater quantity of splice junctions than the tools it was compared with. Kim et al[14] showed that HISAT2 performed very well in sensitivity unlike STAR and TopHat2 in aligning short anchored reads and intermediate length reads.

Table 1: A summary of previously developed aligners and description on their algorithm approaches. First column states the name of the splice-aware aligners, middle column provides a description on their algorithm and last column is a reference to published article.

Splice-Aware Aligners	Information	Literature References
ContextMap	Employs context-based method in the identification of the most suitable alignment per read as well as parallel mapping to many reference genomes	Bonfert T et al[16]
CRAC	Uses integration method to detect sequencing errors and events such as splices when mapped to a reference genome	Philippe N et al[17]
GSNAP	Engages a successively constrained search procedure to filtering the lists of possible alignment positions in the genome index and this is fast	Wu TD et al[18]
HISAT	Incorporates Burrows-Wheeler Transform and Ferragina-Manzini (FM) index in an index strategy for reduced memory resources during alignment	Kim D et al[8]
HISAT2	Two-pass model of HISAT to detect more reads aligned to the reference genome	Kim D et al[19]
OLego	Employs Burrows-Wheeler transform, seed-and-extend strategy to perform *de novo* alignment of sequenced mRNA-seq reads	Wu J et al[21]
RUM	Incorporates reads alignment to both reference and transcriptome genome and applies BLAT to the unmapped reads before unifying the whole alignment	Grant GR et al[10]
SOAPsplice	Uses a two-stage method which comprises the discovery of many appropriate splice junctions and a subsequent filtration of detected false positives in splice junction detection in the absence of a previous knowledge	Huang S et al[22]
STAR	Engages seed search and clustering strategy in mapping spliced RNA-seq reads to the reference genome	Dobin A et al[7]
Subread	Algorithm for mapping RNA-seq reads to the genome of reference is based on a seed-and-vote strategy	Liao et al[23]
TopHat	Uses Bowtie to map reads to the genome without depending on splice sites that are known	Trapnell C et al[20]
Rsubread	Incorporates the Subread algorithm and programming in R for RNA-seq reads alignment	Liao et al[12]

Out of these several splice-aware aligners used for RNA-seq studies, some are used sparsely and others continuously. Aligners like STAR, HISAT and Rsubread are among the aligners widely used, updated, studied and referred to as sensitive to splice sites (Table 2). HISAT is the first aligner to apply the hierarchical indexing strategy in the alignment process[19]. Baruzzo et al[6] compared an earlier version of subread but new functionalities have recently been added to the Rsubread. Some improvements in the Rsubread alignment algorithm are the ability of *align* and *subjunc* to detect multiple short indels within a read and the reduction in alignment execution time due to multithreading[12]. Liao et al[12] recently showed that Rsubread outperformed the popular alignment tools but HISAT2 was not among the splice aligners that were compared. Consequently, HISAT2, Rsubread and STAR will be juxtaposed in order to achieve a possible benchmark.

Table 2: Representation of some known studies that compared the aligners selected for this research. The first two columns show references to published articles and the respective years they were published. The third and fourth columns show the different Subread versions compared in the corresponding researches. Fifth to seventh column informs the presence or absence of STAR, HISAT, HISAT2 among the aligners compared in the study. Last column shows that other aligners not mentioned were included in the aligners in comparison.

Research Reference	Year	Subread (other)	Rsubread (1.32.0)	STAR	HISAT	HISAT2	Other aligner
Kim D et al[19].	2015	-	-	X	X	X	X
Liao Y et al[23].	2013	X	-	-	-	-	X
Liao Y et al[12].	2019	-	X	X	-	-	X
Dobin; A et al[7].	2013	-	-	X	-	-	X
Raplee ID et al[26].	2019	-	-	X	-	X	X
Baruzzo G et al[6].	2018	X	-	X	X	X	X
Krizanovic et al[27].	2018	-	-	X	-	X	X
Engström et al[1].	2013	-	-	X	-	-	X

Some recent benchmark studies stressed the importance for benchmarking a splice aware algorithm for RNA-seq studies to facilitate quality analysis[1, 3]. The trend of benchmark studies and development of algorithms with the ability to correctly map junction-spanning reads has been the case. In general, these comparative studies have either been between all the versions of a specific tool and older or single version of other tools or just a comparison to propose tools that can be used for specific types of researches. There have been recent revisions of some splice-aware aligners like Rsubread and a study showing its high sensitivity to splice junctions[12]. To the best of my knowledge, there has not been any sole comparison of Rsubread, HISAT2 and STAR and there is still no benchmark splice alignment tool available. As a contribution to the biology field, this study aimed to provide a comparison that can lead to the selection of a standard tool for splice aware alignment. A further comprehensive

benchmark research could be carried out with exhaustive verification to propose and validate an aligner as the benchmark splice aware aligner for bioinformatics analysis.

1.2 Aims and Objectives

The main aim of this project was to compare the updated versions of three sensitive splice aware aligners: Rsubread, STAR and HISAT2, showing their levels of sensitivity to splice junction during alignment and the performance updates in line with the developers' conclusions. The aim was supported by matching the aligners to a read quantification program, *featureCounts*, to determine which aligner performs best[28]. The adoption of mapping RNA-seq reads to a reference genome for this research is because novel transcripts, richer transcript studies and better alignment yields are considered before the choice of splice aware aligner is made. This method produces these desired outputs.

The objectives employed to achieve this aim are:

Objective 1: Investigating the resource usage of the selected software during the execution of the alignment by observing and recording the runtime and memory used during the reads mapping process.

Objective 2: Checking the performance of the aligners at default parameter settings as described in the tool update manuals[12].

Objective 3: Investigating the sensitivity of the aligners under comparison to splice sites mapping of sequenced reads using the three splice-aware aligners and checking for the aligner with the most accurate detection of known splice sites.

2. Materials and Methods

2.1 Overview of Algorithms

The design of the algorithm for RNA-seq aligners is based on varying principles. While some have been designed as extensions of contiguous short read aligners and used in mapping short reads to splice junction databases or portions of split reads to a reference genome in a contiguous manner, others have been designed to directly map non-contiguous sequences to a reference genome[7].

2.1.1 STAR

The Spliced Transcripts Alignment to a Reference (STAR) algorithm comprises two main phases: the seed searching and clustering or stitching phase[7]. Central to the seed searching step is the sequential search for a Maximal Mappable Prefix (MMP). MMPs are the longest matching sequences to one or more locations on the reference genome when STAR is used to align reads[3,7]. A seed is the part of the read that is separately mapped[29]. STAR begins its seed searching process by mapping the first MMP to the genome and makes a next search, from the unmapped part of the read, to locate the next longest sequence that has an exact match in the reference genome[7,29]. In the case a mismatch or indel prevents the spliced aligner from locating an exact matching sequence, it extends the former MMP and if this extension fails to produce a good alignment then the poor quality or contaminating sequence is soft clipped[29].

For the next step of the two-phase process of STAR, the splice aligner, simultaneously, clusters the seeds together to a set of 'anchor' seeds based on proximity and stitches them together with the guide of a local alignment scoring scheme[26,29]. This scoring scheme allows for user defined scores for insertions, deletions, mismatches and splice junction gaps and the joined seeds with the highest score is selected as the best alignment of a read. STAR applies an uncompressed suffix array in its MMP search, thereby, explaining its ability to quickly search large reference genomes[3,7,29].

The parameters in STAR are categorized by their functions. For the tool's output, the regulating parameters begin with --out*. An example is the --outFilter*, which controls the filtering of the alignment output, and can be tuned to suit the research requirements. Generation of the genome is regulated by the --genome* parameters. In splice alignments, the –sjdb* controls the splice junction database (annotations) at the genome generation step[7,29].

2.1.2 HISAT

Hierarchical indexing for spliced alignment of transcripts (HISAT) uses an indexing scheme based on the **Burrows-Wheeler transform** and the **Ferragina-Manzini (FM)** index[19]. Burrows-Wheeler transform is an algorithm that transforms and restructures data in a more compressible form[19]. FM index, named a self-indexing tool, is a data structure that integrates compression and indexing into a single compressed file of the original file and some indexing information[30].

HISAT applies the Bowtie2 algorithm in its implementation for the low-level operations needed to construct and search the FM indices[19]. The **Bowtie2** algorithm is an extended index-based approach of Bowtie that allows gapped alignment by the broad division of the algorithm into two stages[31]. The first stage is the initial ungapped seed-finding stage, which benefits from the full-text minute index speed and efficiency. The next stage is the gapped extension stage that optimizes hardware-accelerated dynamic programming algorithms.

For alignment, this splice aware aligner employs two types of indexes (Figure 2)[19]. The first is a whole genome index that anchors each alignment and the next index consists of numerous overlapping local FM indexes for high-speed alignment extensions. Each of the local indexes represents 64,000bp. For the human genome, there are approximately 48,000 local FM indexes with each local index overlapping the next by 1,024 bp, thereby covering the 3 billion bases of the human genome[19,30]. These boundaries that overlap one another eases the alignment of reads that compass the regions where two indexes cover. This indexing strategy accounts for its speed and reduced memory usage.

HISAT2, as a two-pass version of HISAT, communicates a list output of the splice sites with long-anchored reads support in the first run. At the second run, the aligner utilizes the splice site report from the first run to map reads supported by short anchors.

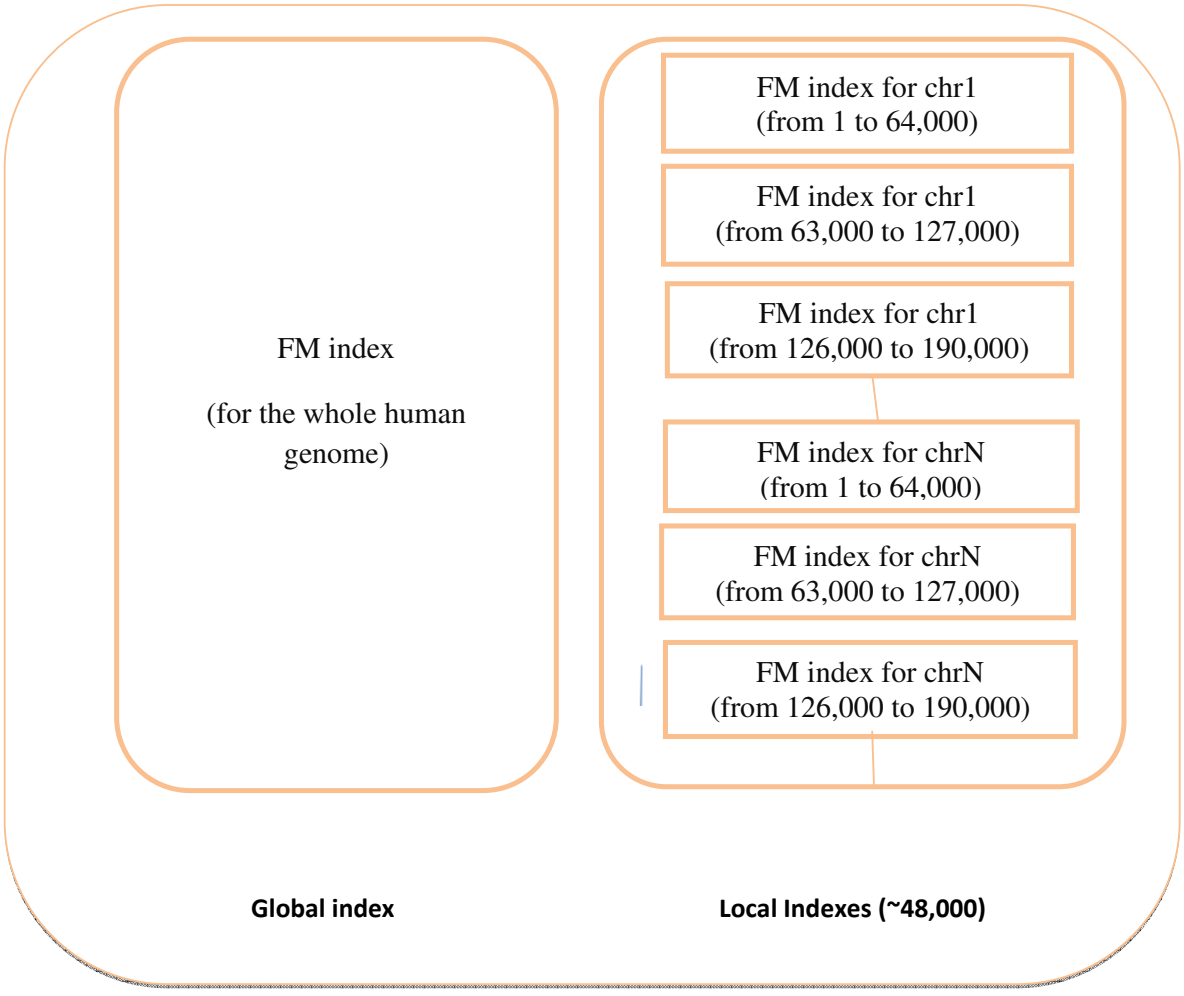

Figure 2: Pictorial representation of the hierarchical indexing applied by the HISAT aligner. The picture represents the two types of index utilized by the HISAT aligner. The left part of the diagram shows the global FM index, which represents the complete human genome while the local FM indexes are shown on the right part. There are approximately 48,000 local FM indexes with 64,000 bp in each. Every local FM index overlaps the next by 1,024bp.

2.1.3 Rsubread

Rsubread is a Bioconductor software package that engages R functions in the implementation of high-performance alignment of RNA-seq reads[12]. It incorporates mapping of reads and quantification, therefore does not depend on any other software but R. The Rsubread algorithm employs the seed-and-vote paradigm whereby the subreads or seeds (short equally spaced seeds from each RNA-seq read) choose or vote the mapped genomic location for the read[12,23].

The R subread pipeline for alignment and quantification comprises five functions in R as briefly described below[23].

1. **buildIndex** builds a one-time index file for each reference genome from a FASTA file at either a single or 3-base resolution. Although it takes more time and memory space to build the full index at single-base resolution, it will speed up the alignment process afterwards.

2. **align** performs the basic alignment for gene-level analyses of RNA- or DNA-seq. It takes raw files in Fastq, SAM or BAM format as input and outputs the mapped reads in SAM or BAM format. This function is very flexible because it maps reads locally and gives a report on the largest region that each read could be mapped to. The unmapped reads are soft clipped afterwards. **align** performs an automatic detection of insertions and deletions.

3. **subjunc** functions similarly to align except that it focuses on comprehensive detection of exon-exon junctions and in turn gives a report of junction-spanning read full alignment.

   Together with the **align** function, they use a two-pass process to attain high accuracy. The seed-and-vote-phase is the first pass. In this step, many 16mer subreads retrieved each RNA-seq output are aligned to the genome with the use of the hash table. This step enables the indels and splice junction detection and decides the major location where the read is aligned. The second step is the phase where each read is locally realigned in detail with the help of indels and junctions.

4. **propmapped** carries out the mapping statistics. It computes the measure of reads that are aligned correctly.

5. The *featureCounts*, earlier mentioned in the aims and objectives section of the problem formulation, is the function of Rsubread that performs the quantification. It counts and reports the number of reads overlapping specific features of the genome.

2.2 Software Tools

The splice-aware aligner versions to be used in this research are Rsubread (1.32.4), STAR (2.7.3a) and HISAT2 (2.2.0). Rsubread is an R package that can be downloaded fromhttp://www.bioconductor.org/[12]. HISAT2 is open-source and is available from https://ccb.jhu.edu/software/hisat2/index.shtml[19,32]. STAR is a Unix-command line tool found

in https://github.com/alexdobin/STAR/releases[1]. *featureCounts* is a quantification function available in the Rsubread package.

All alignments were performed on a LinuxOS Ubuntu 19.04 server with Intel i5-6500 3.20 GHz CPU cores and 62.8GB of memory located at the University of Skövde, Sweden. Due to limited memory size of the available server, an external hard drive of 1TB was used.

2.3 Dataset

The choice of dataset to be used in any bioinformatic benchmark study is critical to the research[33]. Since alignment algorithms are designed to align reads without a combination of information across either reads or samples, reads can be generated from an appropriate reference genome and models of genes, incorporating sequence errors, intron signals, indels and substitutions to a certain extent[6]. This simulated data should be able to reflect the necessary properties of the real data. Introns and indels pose a high level of difficulty to alignment algorithms and as such the splice aware aligners should be able to handle the varying degrees of complexity caused by these polymorphisms[6].

The simulated data from the study by Liao et al[12] was used. This data was simulated after the GRCh38 genome and the SEQC real data gene annotation and comprises in total 15 million 100-base pair (bp) reads. Paired-end reads are preferred because of the rich and high-quality information obtained from them[34.] Simulation of complexity was done with the introduction of germline variants at the respective rates of 0.0009 SNPs and 0.0001 indel in the reference genome prior to the extraction of the sequenced reads. The exon-exon splice junction content of the data includes 233,021 sites, and 25% of the simulated reads originate from a junction. The reference genome used for this research was the human gene annotation GRCH38/hg38 (build 38.2). The choice of this data was because it had been used in a published article.

In real datasets, the true origin of the reads is not known and can only be estimated from the alignment process[27]. Even though the use of simulated data gives a better benchmarking assessment of the aligners, this study was applied on a real dataset, in order to see how the aligners, perform on such data. The real RNA-seq reads mapping to the reference genome was carried out with the Liao et al[12] data. The data was generated by the Sequencing Quality control Consortium (SEQC) project. SEQC project is a project of the United States Food and Drug Administration that generated data sets with over 100 billion reads for the robust evaluation of RNA-seq analyses for regulatory and clinical purposes[35]. The reason for the choice is that the expression profiles of the data are known, and the data will be suitable for benchmarking studies. The file of the selected Universal Human Reference RNA (UHRR) data comprises 15 million 100 bp read pairs.

2.4 Bioinformatic Pipeline

The workflow designed for this study is detailed below.

1. Data generation and mapping of reads to the reference genome.
   The data was first downloaded and the RNA-seq reads were aligned to the reference genome using each aligner. The output and resource usage of all the

aligners during the process were recorded. Baruzzo et al[6] and Liao et al[12] performed their analysis on STAR using the STAR two-pass alignment and this mode was also employed in this study. For HISAT, the aligner first attempted to discover candidate locations across the human genome by mapping each read globally using the global FM index. It was expected that few candidates would be identified after this step. The aligner would then choose a local index for each identified candidate and align the rest of the read with it. For paired end reads, it aligns each mate separately and combines both alignments.

Rsubread has a built-in gene annotation for the selected genome. The difference in the reference genomes is that in the built in RefSeq gene annotation, overlapping exons from the same gene are fused together to yield a set of exons that do not overlap. Otherwise, it is similar with the NCBI annotation. **buildIndex**, when called, would use a hashing algorithm to build an index for the genome. The alignment functions would be called, and output documented.

2. Individual analysis of the splice-aware aligners
   The resource usage of each aligner was summarized after the mapping. The output of the aligners at default settings was documented and the quantification summary of the aligned reads were also reported in a tabular form. The sensitivity for each aligner was calculated as the percentage true splice sites identified from the total present splice sites. The results were documented.

3. Comparative analysis step
   On completion of the individual analysis, the outcomes were collated and analyzed with the respective findings duly discussed and documented. Conclusions were drawn and represented graphically.

2.5 Alignment methods

Reads can either be mapped to a genome or transcriptome, if reference-based analysis is the objective, or assembled if there is no reference genome for the species the data originates from[1,6,10,13]. Mapping reads to a reference genome results in 70-90 percent of uniquely mapped reads and a notable fraction of multi-mapped reads (reads aligned to several similar regions)[13]. Aligners, used for alignment, determine the percentage of uniquely mapped reads. The primary cause of genomic multiread is the presence of repeated sequences or of shared domains of paralogs[13]. Mapping to a reference genome produces a higher percentage of aligned reads thereby improving the quality parameter of the alignment[13]. Novel transcripts are discovered with the application of this method. These rich output with this method propelled the selection of the method in this research.

In the case of mapping RNA-seq reads to a transcriptome, there is slight reduction in percentage of uniquely mapped reads due to the loss of reads from unannotated transcripts and an obvious increase in multi-mapped reads. The increase is as a result of read alignment to shared exons by the same gene transcript isoforms. This means that the read mapped to all existing gene isoforms in the transcriptome that share the same exon.

As an alternative to reference-based alignment, reference-free assembly can be applied[6,13]. Here, reads are first assembled into contigs or transcripts which becomes a reference transcriptome[13]. The reads are then mapped to the assembled transcriptome prior to downstream analysis. Applying this method increases the complexity of the alignment stage and demands more quality control and improvement measures.

2.6 Implementation

2.6.1 Data Generation and Reads Alignment

2.6.1.1 Generation of Data

Benchmarking studies on algorithms for RNA-seq alignment are usually carried out with simulated data[6]. The simulated and real data from Liao et al[12] were downloaded from the link stated in the article. They were extracted for the alignment purpose.

2.6.1.2 Alignment with STAR

The primary workflow of STAR splice aligner comprises of two stages[36]. The first stage generates genome index files where the researcher provides the sequences of the reference genome and the annotation to be used in the index files generation. The reference genome sequence is stored in a file of FASTA format and the annotation file format is GTF. It should be noted that the generation of index files is performed once in the process. With a generation of the index by the researcher, instead of using an available STAR author-designed genome, there is an advantage of updated assembly and annotation which could have an impact on the read mapping.

Alignment of reads to the reference genome is the second stage. At this stage, the already generated genome index files and the RNA-seq reads are provided for alignment processing. The sequence file format is either FASTA or FASTQ. After alignment, the aligner returns files in SAM or BAM format alongside a summary statistic file of the process. The STAR algorithm allowed reads mapping in the absence of annotations, but the developer strictly recommended otherwise, and the recommendation was adhered to.

STAR 2.7.3a, which was released in 2020, was downloaded from Github and run from source on the Ubuntu server with 68GB of RAM and the external memory of 1TB as the storage memory. With the media drive as the working drive, the working directory was first created before the genome generation process. A failure to create the working directory will result in an output file error and the aligner will not be able to process the genome generation step.

The first step of the splice aligner's workflow was performed with default settings. In this setting, the thread setting of one was used thereby saving some storage space and consuming more operation time. Memory bound tasks linearly increase the memory usage and multithreading improves speed but for STAR it is different. The speed, engaged by the aligner in mapping up to 300 million read pairs in an hour, balances with the RAM size[7]. Running the

genome generation command in Linux with one core and default parameter, the genome files creation was started, then the suffix array sorting followed. Next to the suffix array sorting, which took the most time of the genome index file generation, was the generation of the suffix array index. Sequel to that was the GTF annotations processing and the insertion of junctions into the generated indexes. The genome generation was concluded by the writing of the genome suffix array index to the disk.

STAR multi-sample 2-pass mapping was selected for the mapping of reads step of the workflow. Alignment in this mode allows the detection of more aligned splice reads to the novel junctions[7]. Using 2-pass mapping does not affect the number of novel junctions that are detected but rather it enables the splice aligner to discover a higher number of spliced reads that are mapped to novel junctions. The operation basis of this mapping mode is to run the first pass alignment with the default parameter setting and use the detected junctions in the first pass as the second pass alignment annotated junctions.

The aligner was set to allow a maximum of one multiple alignment for each read. This implies that the read would be taken as an unmapped read if the read maps to a number of loci that are more than one. Since the aligner could function with zipped files, the read files input for alignment were zipped and the function to unzip the files was called.

2.6.1.3 Alignment with Rsubread

The workflow requires that the index is built once for the indexing operation of each genome[12]. Once the index is built, it can be used for other alignment studies where the same reference genome that the index was built from is a requirement. Although it was possible to download a built index, it was not done in this study in order to fulfil the aim and objectives.

The Rsubread workflow is such that it can carry out its processes together, once instructed, instead of executing one call per time. A call to the *align* or *subjunc* function and thereafter the *featureCounts* function outputs a matrix of counts in R, which is annotated having all the samples in the same order that they were originally.

The *align* and *subjunc* functions in Rsubread are both used for alignment. While the *align* function is better for analyses of expressions of genes, the *subjunc* function focuses on junction detection and produces alignment result of reads that span the junctions in the sequence. Although the *align* function can map reads from DNA and RNA sequencing, it does not fully align reads that span the exon junctions. However, the *subjunc* function reports a full alignment of each RNA-seq read and was recommended for analyses that require the use of RNA and splicing[23]. The *subjunc* function was used for the alignment discussed under Rsubread aligner. Alignment was performed with the default settings in Rsubread although the number of threads was set to five in accordance with the core number of the server. The inbuilt annotation was used for the alignment reported in this study.

For the alignment, the working directory was set, and the Bioconductor package was loaded. Then the index file was built using the *buildindex* function to generate a hash table index for the GRCh38 (build 38.2) reference genome. The files were stored in the working directory. Rsubread can either build a full or gapped index but its index function default is a full index. Full index increases the speed of alignment but requires more memory resource while gapped index will save memory space and use more time for mapping[12]. At this stage, the memory to be designated for the alignment process is regulated. This can be done with the choice of

building either full or gapped index with an option of allowing index split for more memory space reduction. By design, the index building step would need ~15 GB memory space. The full index in accordance with the aligner's default setting was created and used for the read mapping task.

The required arguments for the alignment were the index file, readfiles of the paired-end read data, data type specification, output format, output file and other arguments that could be set as needed. The data type specification was set as "rna" and the BAM format was required for the output. The software was assigned five threads for the mapping process.

2.6.1.4 Alignment with HISAT2

HISAT2 source package was downloaded from https://ccb.jhu.edu/software/hisat2 and the hisat2-2.2.0-beta-source was built into the Linux server[19,32]. The splice aware aligner has a capacity to build an index of any size of reference genomes and the wrapper script automatically does the index building. On the completion of the index building, an extremely large amount of local index was chunked out, but these were stored in small files that also included other algorithm design optimizations that reduce the allocation of memory resources to the process.

The hierarchical indexing strategy implemented in HISAT2 is based on the Burrows-Wheeler transform and the FM index. A combination of Burrows-Wheeler transform and the FM index in HISAT2 enhances speedy RNA-seq alignment with reduced memory[19]. For the alignment stage, HISAT2 was set to align the input simulated reads with a multithreading of five cores of CPU engaged for the process. Any alignment software that employs a model of multithreading in its algorithm runs the mapping of reads with increased speed. The thread parameter option is such that every thread runs on a unique processor searching alignments simultaneously and in parallel thereby resulting in expanded alignment throughput. All other parameters were set as default.

3. Results

3.1 Results from the Splice Aware Aligners (Simulated Data)

The report of the alignment of the reads were output showing the way the reads were mapped. While some reads were distinctly mapped within an exon, others could be mapped to more than one exon explaining the concept of multiple mapping or multi-mapping. Also, the splicing effect, which occurs either during or after the transcription process leads to the mapping of sequenced reads to the exon-exon junctions around the splice sites (Figure 3).

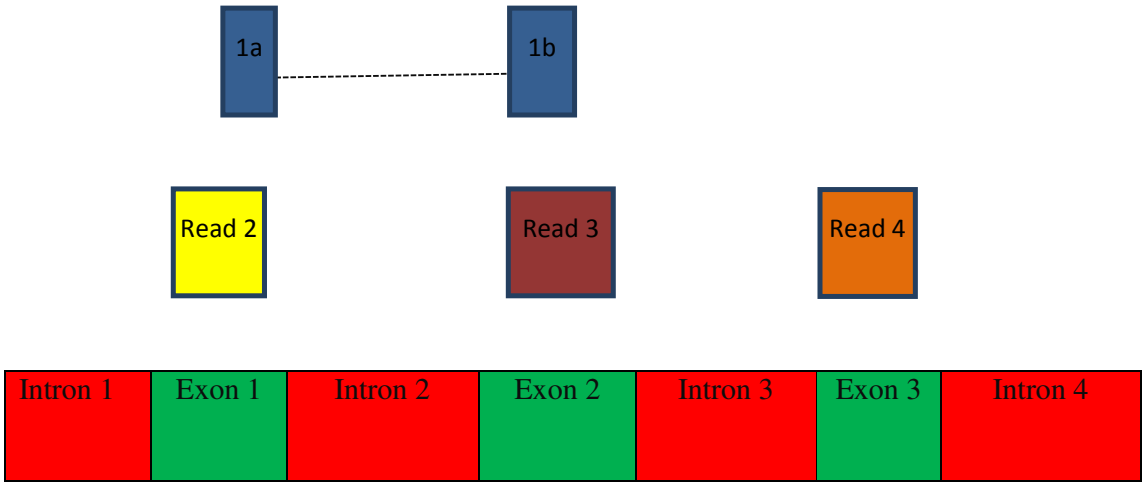

Figure 3: Reads and their alignment. Part of the first read was aligned to exon 1 and the other part to exon 2 showing how a read spans more than one exon. This occurrence was as a result of the splicing effect of the pre-mRNA to produce mature mRNA after transcription. Then the reverse transcription of the mature mRNA to cDNA and its consequent sequencing produces reads that span may exons. Reads 2, 3 and 4 align specifically within a distinct exon after the mapping to the reference genome. Read 1 was split so that it could map to the exon close to the splice junction.

3.1.1 STAR

2-pass STAR alignment was performed using the server stated in the alignment section. The aligner was designed to use a minimum of 100 GB disk space for the indexing of the human genome. The resource usage was represented (Table 3). During the genome index file generation, the aligner was not set to use any additional thread. There was no file in the genome directory prior to the step of generating the genome. More than 27 GB memory was engaged for the indexing process. While the index was built in 4 hrs and 25 minutes, the alignment was completely executed in approximately ten minutes for both passes. For the 2-pass STAR alignment with the simulated data and the generated genome, the memory and time resource usage in the first pass were 30.6 GB and 9.5 minutes, respectively, while in the second pass, the memory usage remained the same with the run time exactly at 10 minutes.

Table 3: Resource Usage by STAR aligner for generating the index and the 2-pass alignment. The first column contains the stages of the alignment workflow, the second column consists of the numerical representation of the memory size and the third column contains the number of minutes used during the corresponding process.

STAR Process	Memory Resource Allocated (GB)	Time expended during the process (minutes)
Index files generation	27.3	265
1st pass alignment	30.6	9.5
2nd pass alignment	30.6	10

From the 1st pass, approximately 14.88 million pair-end (PE)reads out of the input 15 million simulated PE reads were mapped and 121,453 PE reads were unmapped (Table 4). In the second pass mapping, 14,521,822 PE reads mapped uniquely and 356,909 were multi-mapped. Considering the two passes, the increase in uniquely mapped reads, reduction in multi-mapped reads and ultimately reduction in unmapped reads evidenced the better output using the 2-pass mode. An increase in the mapping speed in the second alignment was observed possibly due to the fact that the data was simulated, and the aligner would not find new junctions in the 2nd pass. The percentage alignment, calculated as the sum of the percentages of the uniquely aligned reads and multi-mapped reads, resulted in 99.2 for both passes. The aligner was able to detect 220,653 and 220,629 splice junctions in the 1st and 2nd pass respectively.

As anticipated, the percentage of alignment was high with the data that modelled unreal errors, complexity, alternative splicing and polymorphisms. The few reads that were unmapped were so because they were too short and for other reasons. These reasons could be the aligner's inability to discover the exact position of the read or none existence read location in the referenced genome sequence as a result of contamination[6].

Table 4: STAR Simulated Reads Mapping Output. The details of the alignment output make up the content of the first column while the second and third columns are the results of the 1st and 2nd pass mapping respectively. The percentage of aligned reads is taken as the percentage of uniquely aligned reads added to the percentage of the multi-mapped reads.

STAR Alignment Details	1st Pass Output Values	2nd Pass Output Values
Mapping Speed (million reads per hour)	158.36	159.29
Total Input Reads (100 base pair read)	15,000,000	15,000,000
Uniquely Mapped Reads	14,520,865	14,521,822
Multi-mapped Reads	357,682	356,909
Unmapped Reads	121,453	121,269
Percentage Aligned Reads (Percentage Uniquely Aligned + Percentage Multi-mapped Reads)	99.19	99.19
Percentage of splice Junctions detected	94.69	94.68

3.1.2 Rsubread

Mapping simulated reads with *subjunc* function in Rsubread executed on the server was done with the five cores. However, the index building was performed with the default thread number of one. Memory requirement by the aligner for building index according to the design is 15GB but the aligner used 14.8 GB to build the index used in this research in 48.2 minutes. The simulated read mapping utilized 18.3 GB memory and a run time of 9.8 minutes with a bam output file of 3.5GB size.

Almost all reads were mapped, although 270,409 of the reads were mapped to multiple loci (Table 5). In total, 0.47% of the simulated PE reads were not mapped. Of the exon-exon junction content in the data, Rsubread could detect 224,516 junctions accounting for 96.4% of the total splice junctions in the simulated data.

The flexibility and user friendliness characterized by the aligner was outmatched by the extensive information produced after the *subjunc* function was called. Foremost to the read mapping by the Rsubread aligner was the seed and vote algorithm where the aligner discovered insertion and deletions in the sequences, splice junctions and primary alignment sites in the read. Then an exhaustive remapping using the accumulated exon-exon junctions and indels followed afterwards. Indel use in the re-alignment is part of the modifications that were recently done to the splice aligner for increased accuracy of alignment[12].

Table 5: Simulated Reads Alignment Output using *subjunc* in Rsubread. The first column contains the output statistics of concern from *subjunc* function in Rsubread, while the second column shows the values output against the statistic. Rsubread had nearly 100% aligned reads.

***subjunc* Required Output Statistics**	**Output Values**
Total Input Reads (100 base pair read)	15,000,000
Uniquely Mapped Reads	14,658,353
Multi-mapped Reads	270,409
Unmapped Reads	71,238
Percentage Aligned Reads (Percentage Uniquely Aligned + Percentage Multi-mapped Reads)	99.53
Percentage of splice junctions detected	96.35

3.1.3 HISAT2

HISAT2 was used to align the 15 million 100 bp simulated reads. The index of hg38 was built within 1hr 18 minutes and stored with 4.5GB memory space. Executing the alignment used up 15GB memory and was completed in 6 minutes of runtime. HISAT2 reported alignment in a different way. Uniquely mapped reads were taken as all the reads that were aligned concordantly exactly one time. The number of multi-mapped reads was the sum of all reads

aligned concordantly more than 1 time whereas the unmapped reads were the pairs aligned concordantly or discordantly 0 times. Discordantly aligned reads 1 time were the reads that were incorrectly mapped. This was according to the software designer's result interpretation.

As a surprising occurrence, the alignment was performed in 6 minutes with a SAM output file of 10.5 GB and 15 GB memory was assigned to the process. The SAM file consumed much memory asset, but when converted to BAM afterwards, it used only 2.3 GB. This showed that if the aligner could be redesigned to output SAM file directly instead of at the discretion of the user, it would be a considerable improvement.

Table 6: HISAT2 Simulated Reads Alignment Output. The statistics from HISAT2 are listed in the first column, while their result values are shown in the second column. Incorrectly mapped reads were clearly output by the aligner.

HISAT2 Output Statistics	Output Values
Total Input Reads (100 base pair read)	15,000,000
Uniquely Mapped Reads	14,049,750
Multi-mapped Reads	405,562
Incorrectly Mapped Reads	8,974
Unmapped Reads	535,714
Percentage Aligned Reads (Percentage Uniquely Aligned + Percentage Multi-mapped Reads + Incorrectly Mapped Reads)	96.43
Percentage of splice junctions detected	90.20

Out of the total aligned reads, 405,562 of them were mapped to more than one site and the rest were distinctly mapped (Table 6). In total, 3.6% of the simulated PE reads were unmapped. A sum of all the aligned reads including the discordantly aligned ones made up the reads considered in the calculation of the percentage aligned reads. The aligner detected 210,194 junctions out of the 233,021 exon-exon junctions in the data.

HISAT2, which is the two-pass design of HISAT, first returns an inventory of long-anchored splice locations and subsequently maps short-anchored reads in the second pass. This aligner model is anticipated to be more sensitive and slower than the earlier model that uses only one run for read mapping[19]. It performs both passes and outputs a summary afterwards different from the STAR aligner's two pass mode where each pass is scripted and run separately producing separate results for both passes.

3.2 Results from the Splice Aware Aligners (Real Data)

STAR had an approximated amount aligned reads of 95% for the two alignment processes. There was an increase of 6,762 in the number of the paired reads uniquely mapped in the second pass and a decrease of 5,434 PE reads in the multi-mapped reads resonating with the aligner's ability to map more reads in the second pass. STAR recorded a higher mapping speed in the first pass alignment than in the second pass with the Liao et al[12] UHRR data. Having built the index, the read mapping process in the 1st and 2nd pass by STAR consumed

30.4GB of memory for the alignment process in each pass. The 1st pass was completed in 9 mins whereas the 2nd pass took 10 minutes to conclude the mapping of reads. 212,338 junctions were discovered by the aligner in the 2nd pass whereas the 1st pass recorded 213,038 junctions. Both passes had equivalent output. There was an observed increase in the number of spliced reads in the second pass and a reduction in the number of novel or unannotated junctions in the second pass. This resonates with the advantage of 2-pass mapping to detect more reads mapping at those novel junctions.

Rsubread generated the largest number of mapped reads with a little over 100,000 PE aligned reads, as different from the STAR mapped reads output (Table 7). The number of HISAT2 mapped reads was the lowest and the number of unmapped reads was quite high based on the alignment result. The large difference observed in the output from HISAT2 could be as a result of the aligner's algorithm. STAR detected more splice junctions than Rsubread and HISAT2. In total, 96.08% of the PE reads were aligned by Rsubread using reads from the human brain reference and 94.24% of the PE reads from the universal human reference reads were mapped to the reference genome. 18.4GB of the memory was assigned to the alignment process and the *subjunc* function carried out the mapping of reads in about 12 minutes for the data from the Liao et al[12] UHRR.

Table 7: Alignment output using the Liao et al[12]UHRR data. The alignment output details of interest are listed in the first column, while the results, for the listed details and produced by the different splice aligners, respectively, are outlined in the second, third and fourth columns. HISAT2 total number of mapped reads included 101,008 incorrectly mapped reads.

Required Output Statistics	STAR	Rsubread	HISAT2
Total Input Reads (100 base pair read)	15,000,000	15,000,000	15,000,000
Total Number Mapped Reads	14,301,075	14,412,277	13,992,616
Uniquely Mapped Reads	13,657,670	13,724,099	13,265,737
Multi-mapped Reads	643,405	688,178	625,871
Unmapped Reads	698,925	587,723	1,007,384
Percentage Aligned Reads	95.34	96.08	93.28
Number of Splice Junctions Detected	212,338	211,170	207,449

The Liao et al[19] HBRR data mapped to the genome with each aligner produced a different result from the Liao et al[19] UHRR alignment (Table 8). The data alignment recorded a lower number of unmapped PE reads, for all the aligners than accounted for with the Liao et al[19] UHRR reads. This is as a result of the nature of the cells from where the samples used in the sequencing originated[12]. The allocated memory to STAR in aligning the reads from the Liao et al[19] HBRR data for the first pass was 30.4 GB and 30.3 GB, respectively, while the time of alignment completion were 8 and 10 minutes, respectively. In the case of Rsubread, 18.9GB of the memory was allocated, while the *subjunc* function carried out the mapping of reads in less than 12 minutes. Rsubread identified more splice junctions with this data than STAR and HISAT2.

Table 8: Read Mapping Result using the Liao et al[19] HBRR data. While the first column comprises interesting output details, the remaining columns contain the parallel values recorded by STAR, Rsubread and HISAT2. HISAT2 total number of mapped reads included 104,451 incorrectly mapped reads. All the alignment was executed on 5 threads and index was formed for each aligner for the reason of achieving fairness in their juxtaposition.

Required Output Statistics	STAR	Rsubread	HISAT2
Total Input Reads (100 base pair read)	15,000,000	15,000,000	15,000,000
Total Number Mapped Reads	14,002,296	14,137,316	13,387,216
Uniquely Mapped Reads	13,568,707	13,718,895	12,984,360
Multi-mapped Reads	433,589	418,421	402,856
Unmapped Reads	997,704	862,684	1,508,333
Percentage Aligned Reads	93.35	94.24	89.94
Number of Splice Junctions Detected	209,441	209,752	204,856

3.3 Read counting with *featureCounts*

The Rsubread has a function for summarizing (i.e. quantifying) reads that have been aligned to a feature in the reference genome. A feature can be an exon, a gene, transcripts or binding region[28]. *featureCounts* output is necessary for further analysis after the quantification. Researches that require expression of genes need the aligned reads to be quantified before the analyses can be performed.

The workflow of *featureCounts* is such that it accepts an output file from the read mapping stage as well as an annotation file, which matches the genome used for the study, as input. For this study, an annotation file, other than the one inbuilt in Rsubread, was used for the quantification. Then, *featureCounts* matches the alignment position of the reads to the range covered by the feature or meta-feature in the genome. *featureCounts* considers gaps, such as exon-exon junction, and records an instance, if any intersection between the feature and the read exists. The feature level parameter for this study was selected as read quantification on *exon* level. The algorithm has some filtering process, which is applied when summarizing aligned reads[28]. The filtering process order goes from unmapped to read type, reads that are single, mapping quality, chimeric fragments, fragment length. It further continues with duplicate, multi-mapping, secondary alignments, reads that are split or not, reads with overlap features, overlap length and finally to ambiguous alignment.

The function was designed to accept files in SAM or BAM format with an automatic detection of the file format. It is also not necessary to have the aligned reads from pair-end data to be sorted before calling the quantification function. For this research, the aligned reads were not sorted before the quantification. Multi-overlapped reads were not quantified since studies that concern RNA-seq require that the read fragments come from only one target gene. Therefore, the design of the count function has the multi-overlap parameter as false by default. Setting the requirement for both ends of the reads to be counted is important, if the

data used for alignment comes from pair-end reads and this was reflected in the parameter setting.

The *featureCounts* output for the alignment algorithm, in the respective order of HISAT2, STAR and Rsubread, was 77.6%, 89.7% and 86.4%. STAR had the highest amount of successfully assigned reads whereas HISAT2 had the lowest (Figure 4).

Aligned reads from HISAT2 that were filtered as unassigned because they were unmapped was due to some reasons. The reads accounted as unmapped alignment from the SAM file were not assigned to any feature when quantified. Some of the reads were mapped to multiple sites, while some others were not found to have any form of overlap with a feature in the annotation file. Ambiguity was another cause for the reads not being assigned. In the case of Rsubread, the reads not assigned successfully was because they were unmapped, had no read overlapped with any annotation feature and mostly due to ambiguity. STAR also recorded ambiguous reads and the inability to have aligned reads overlapping any feature.

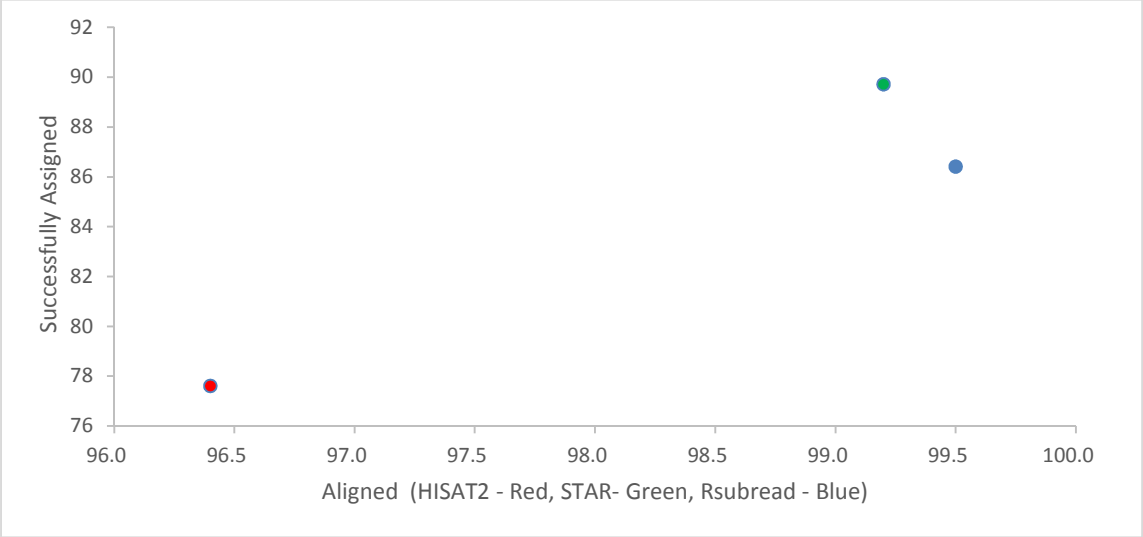

Figure 4: *featureCounts* successfully assigned alignment. Results from *featureCounts* output containing percentage assigned reads on the exon level (y-axis) plotted against the percentage aligned reads (x-axis). Red dot refers to HISAT2, green dot to STAR and blue dot to Rsubread.

3.4 Comparative Analysis of the Splice-Aware Aligners

3.4.1 Resource Usage

The alignment step is considered as the first and most computational demanding step in analyses involving RNA-seq[1,2]. This step involves enormous complex stages of first building the index of a reference genome and subsequently mapping the sequenced reads to the genome using the built index. With splice junctions characterizing RNA-seq data, reads mapping is liable to consume as much computation asset at its disposal[26]. Therefore, it is expected that this bioinformatics pre-downstream analysis step will require a lot of memory. However, HISAT2 showed to defy this norm with an interesting consumption of less memory than STAR and Rsubread while building the index (Figure 5). STAR almost used twice as much memory as was allocated to Rsubread. STAR took up nearly 500% more GB than that

used by HISAT2, since the aligner was designed to make use of a suffix array in uncompressed form when indexing[9,36]. The algorithm of HISAT2 hinges on Burrows-Wheeler transform, a technique that enhances compression of text[19].

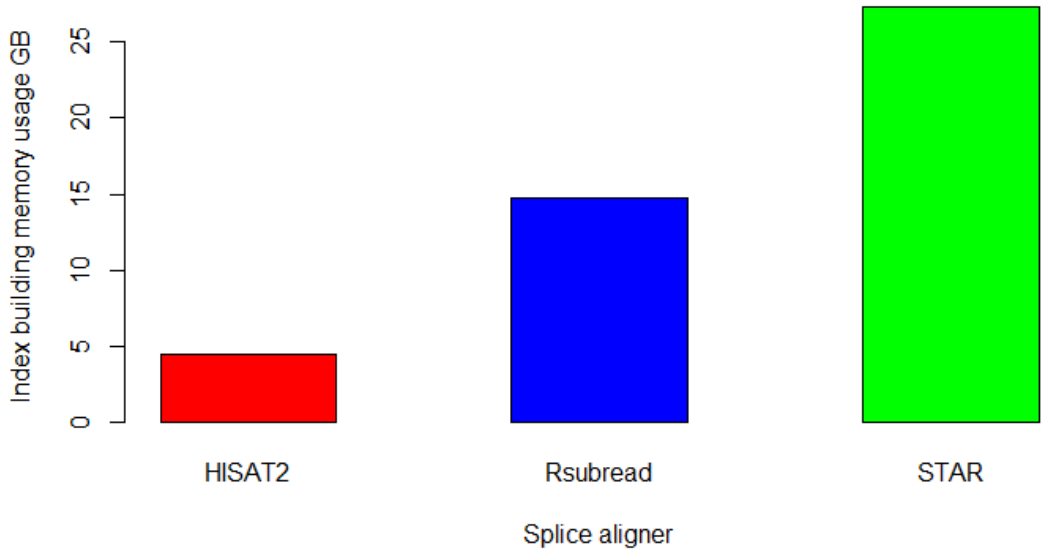

Figure 5: Memory resource usage for index building process. Every splice aligner was set to utilize one thread by default in building the indexes for the genome. HISAT2 used the least amount of memory space.

The indexing memory consumption is transferred on to the mapping of reads space asset since the index is required for alignment operation (Figure 6). STAR and Rsubread did not have much addition to the memory used while indexing, but HISAT2 had a heightened increase. The default output of alignment for Rsubread and STAR is a BAM file. Such files contain sequence data that have been mapped to the genome of reference in compressed arrangement. HISAT2 outputs a SAM file, which is text based and not a compact representation in binary format as is the case with the BAM file. Though the SAM file could be converted with tools such as SAMtools, it does not impact the extra memory allocated to the alignment activity, since the SAM file must be generated before the change to BAM file can be effected. The conversion could ease subsequent analyses of the aligned sequences but has no effect on memory resource reduction.

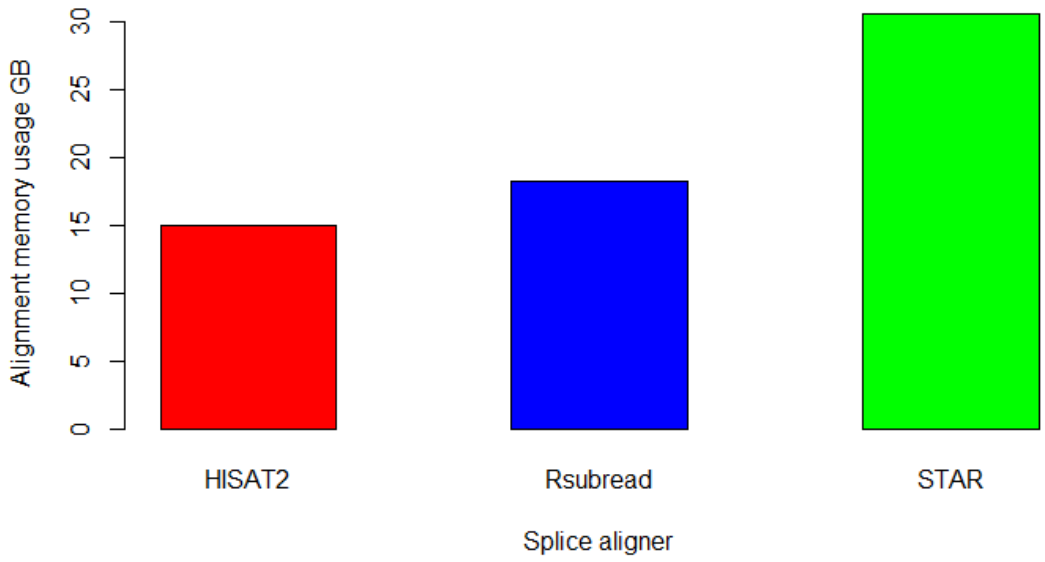

Figure 6: Alignment memory resource usage. The splice aligners were scripted to run the read mapping process of 15 million pair-end reads to the reference genome. STAR used up the most space for the alignment operation.

The run time race was not as close as that of memory. Each aligner had a specific technique for index building embedded in its algorithm. The suffix array of STAR tended to be most complex with an amplified crave for runtime. Rsubread having improved its hashing algorithm was rewarded in the time expended for the execution of that process. With the use of two kinds of index in HISAT2, the global FM index and multiple local FM index, described in methods section, the aligner engaged an appreciable amount of time to complete genome indexing. An improvement to the algorithm of Rsubread indexing employs an increased combative means to gain reduction in the time used in running the index building activity[12]. The *subjunc* function is more time consuming than its counterpart *align* function in Rsubread, but both functions maximize the use of full index to reduce the alignment speed[23].

Comprehensively, alignment runtime for Rsubread was almost five times less than its index building execution time (Figure 7). The number of minutes HISAT2 used to build index was 13 times greater than the time used for performing the actual read mapping or alignment. But the case of STAR was visibly different with a slower genome generation step execution. It took 25 times more runtime than was taken by the alignment process. However, Rsubread and STAR mapped reads to the genome in about the same time. HISAT2 speedily mapped reads unlike the other aligners. Additionally, STAR had an average mapping speed of 148 million 100bp reads per hour with 5 threads.

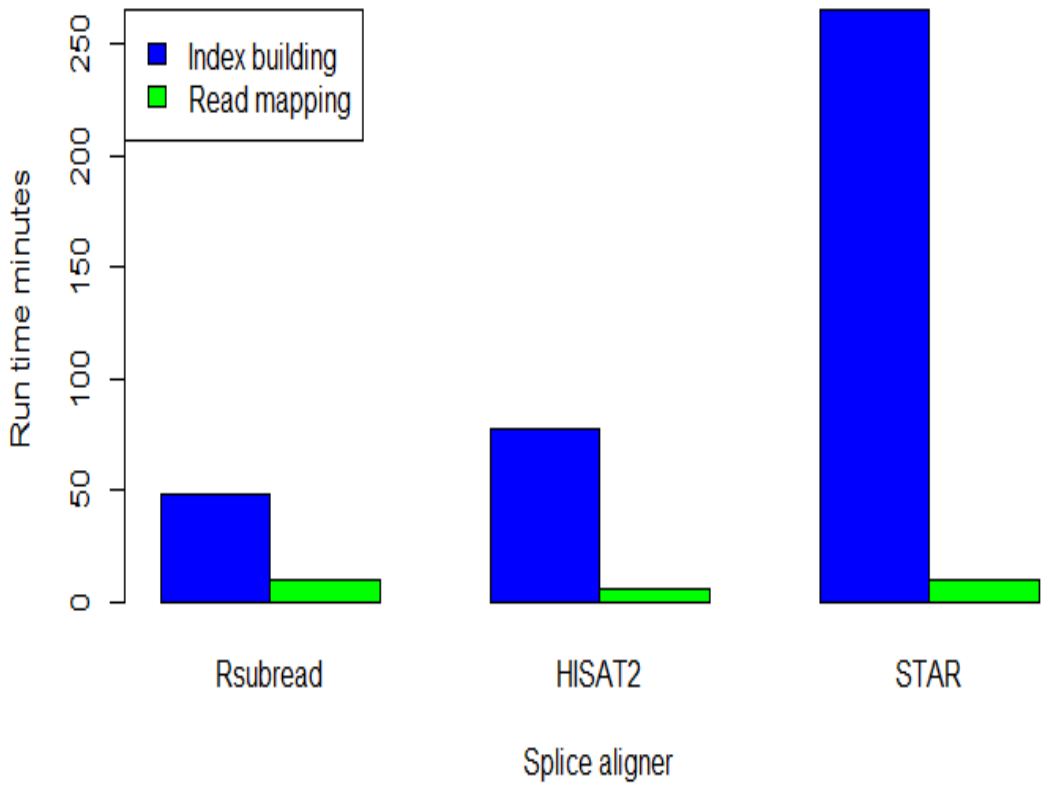

Figure 7: Investigation of runtime. The histogram shows the runtime used by the three different splice aware aligners for index building (blue staples) and read mapping (green staples), respectively. The index building runtime accounts for most time in the alignment step.

3.4.2 Parameter Setting

Alignment algorithms are designed with parameter settings that a user can tweak or tune to adapt to the usage. However, this study purposed to check the alignment performance at default setting. The aligners were only observed on default settings with a tweak in the number of threads for the read mapping process. The default settings that were used during the index building was to keep the number of threads for the process at the value of one. The alignments were performed with five threads. Rsubread was run with the inbuilt annotation. The results obtained with this setting were plotted to show the performance of the aligners.

With the default settings of each aligner, Rsubread had an overall alignment of 99.53%, STAR followed closely with 99.19% and HISAT2 was next with 96.43% (Figure 8). The simulated data used for the reads mapping contained 233,021 exon-exon junctions and the splice aligners detected some of those junctions. The percentage detection by each aligner was

calculated and Rsubread had 96.35%, HISAT2 had 90.20% and STAR had 94.68%. The result showed that the algorithms of Rsubread and STAR could detect splice junctions as well as align reads to a great percentage while that of HISAT2 could do the same but with visible lower percentage difference.

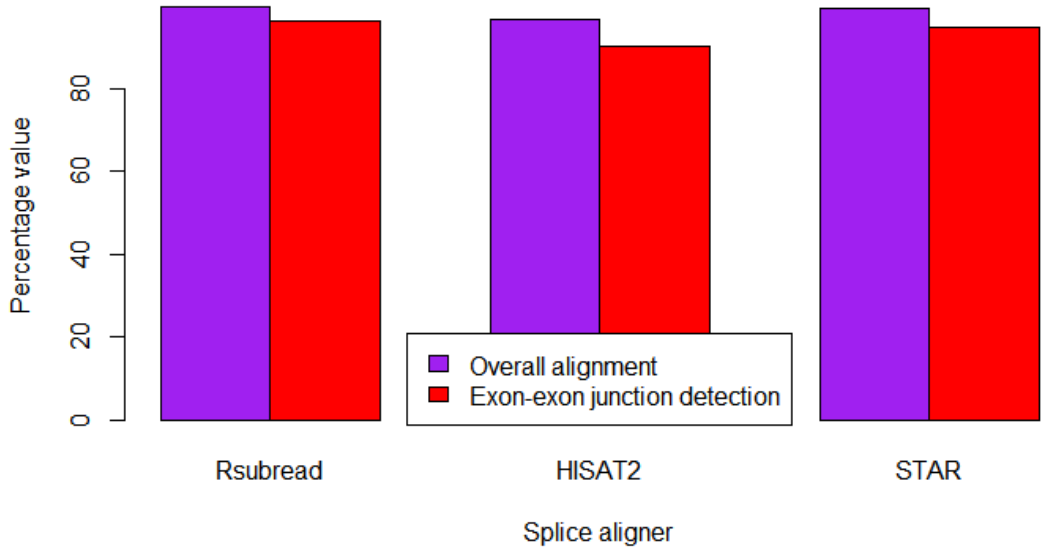

Figure 8: Reads mapping performance at default setting. The y-axis refers to the percentage values of the mapping of reads and the discovery of exon-exon junctions using the simulated data. The x-axis refers to the aligners. The purple staple is the percentage of reads mapping while the red staple is that of the junction detection.

There could be a possibility of having a different output, if the settings were tweaked, but it was not done due to the time limitation during the research. Rsubread could have had a different performance with external annotation since the inbuilt annotation modified the gene annotation from NCBI RefSeq by merging exons of a gene that overlap in order to reduce ambiguity of annotation[12,23]. The developers of these aligners made some recommendations that were followed for the purpose of getting optimum results while using the aligners. The Rsubread developers suggested the use of full index on a server instead of a personal computer while aligning reads to the reference genome. They also advocated the use of *subjunc* function in Rsubread for studies where detection of exon-exon junctions is the priority.

HISAT2 had few recommendations which were specific to the aligner build from source. The recommendation on the use of multiple threads for HISAT2 performance was considered. However, that was used by all the aligners. Also, STAR developers recommended the use of 2-pass mapping in order to have the most sensitive discovery of novel junctions. For genome generation, it was recommended that researchers should generate genome indexes required for

their research instead of downloading indexes as this would give the recent updates of the annotation. A high priority recommendation on the use of annotations in read alignment with STAR, although the aligner could map reads in the absence of annotations, was adhered to. The aligner was designed to use the annotation to enhance reads mapping. As a further aspect for this study, parameters like quality score, mismatch penalties could be tuned to investigate any effect on alignment.

3.4.3 Splice Junction Detection Statistical Analysis

From the alignment output using the simulated data, it was observed that the splice aligners detected several splice junctions. In order to determine the sensitivity of this exon-exon junction detection, the formula below, adopted from the study carried out by Kim et al[19], was applied to the output from the aligners.

Sensitivity = (No of true splice sites reported / Total splice sites in the simulation) * 100

Positive Predictive Value = (No of true splice sites reported / Total splice sites reported by aligner) * 100

The total number of true splice sites was deduced using an intercept script, a program to calculate the interception of output junctions by the aligners and the actual splice junctions in the data. Liao et al (2019) simulated data contained 233,021 exon-exon junctions and Rsubread detected 224,516 splice junctions, STAR discovered 220,629 splice junctions and HISAT2 detected 210,194 splice junctions (Table 9).

The sensitivity of Rsubread was highest with 96.25%, followed closely by STAR with 94.67% but HISAT2 was least with 89.64%. all the aligners had above 99% positive predictive value. This means that for the total number of splice junctions detected with the simulated data, the aligners can have at least 99% correct splice junctions from them.

Table 9: Statistical analysis of splice junctions using the simulated data. The aligners all had a high positive value showing their ability to predict correct detection of splice junctions, if they are sensitive to those junctions.

Statistical Measurement	STAR	Rsubread	HISAT2
Number of Splice Junctions Detected	220,629	224,516	210,194
Total Number of True Splice Sites	220,591	224,276	208,887
Sensitivity	94.67	96.25	89.64
***Positive Predictive Value**	99.98	99.89	99.38

4. Discussion

The progression of NGS technologies, characterized by large data production, necessitate memory and time demanding processes in processing and analyzing the data[4,5]. RNA-seq pipelines usually begin with the mapping of sequenced reads to a selected genome or transcriptome of study[6]. Read mapping is a complex process[37]. The existence of splice sites in these sequenced reads poses a challenge to this computationally demanding level in the RNA-seq analysis[6,9]. Aligners used for mapping RNA-seq reads should be able to process paired end reads, map reads that cross the exon-exon junctions and perform optimally with default settings[6].

Splice-aware aligners that can align a high percentage of reads and are sensitive to splice junctions with a considerable use of time and memory resources will always be relevant[13]. The resource utilization of the aligners was investigated as an objective towards achieving the aim of this study. Although HISAT2 was the fastest in reads mapping, the percentage of aligned reads recorded by the aligner was the least. Rsubread built indexes for the human genome using least time, but used a considerable amount of memory for its operations, the percentage of successful alignments evidenced by *featureCounts* was not more than that obtained for STAR. On the other hand, STAR used the most memory asset but had an alignment percentage very close to that of Rsubread. HISAT2 required the least memory during the alignment stage (Figure 6). On the other hand, the data handling challenge as a result of the aligner's SAM format reads mapping output and the not-quite-clear result interpretation makes it difficult to use.

Rsubread was improved with a two-fold index building time reduction in 2018 (from version 1.30.9) and the version used for this study (1.32.4) was released in April 2019[12]. Therefore, the improvement had effect on the aligner's usage of the least time to build the full genome index (Figure 7). The read quantification function in Rsubread, *featureCounts*, was modified recently to count aligned reads without having the BAM/SAM files ordered, as against the older versions that required the files to be sorted by name. The mapped reads files input for the quantification process in this study were unsorted and the quantification algorithm was able to execute the reads quantification on those files.

The result from the index building run time resource use and increased percentage of alignment agreed with the Liao et al's earlier study on Rsubread[12]. This high point is commendable but there is need for further study of the quality of the alignment in order to be emphatic on the aligner's ability to produce quality mapped reads. The results from the read quantification step produced a percentage that does not match the approximate 100% of aligned reads by STAR and Rsubread though STAR had a better result when compared with Rsubread from the quantification output (Figure 4).

Baruzzo et al[6] opined that the algorithm of STAR aligner is such that it requires much RAM, but has a fascinating speed increase. This was also the case in this study, as shown in its alignment time being almost the same as that of Rsubread (Figure 7). The exception is that in the use of the two-pass method of STAR alignment for detection of more spliced reads across splice junctions the run time will be doubled. Additionally, the aligner used most memory resources (Figure 5).

Congruent to the objective of this study, the aligners showed good results for detection of known exon-exon junctions from the simulated data reads mapping (Tables 4 and 5). Nevertheless, STAR and Rsubread performed better than HISAT2 using default parameter settings (Figure 8). The results obtained from the splice-aware aligners show that all aligners

have over 90% overall alignment mapping. This exceeds Conesa et al's expectation of an overall alignment percentage range from 70% to 90% when RNA-seq reads are mapped to the human genome[10]. The use of *featureCounts* showed that the aligned reads with STAR were assigned to exonic features more frequently than the other aligners.

The result of the statistical analysis obtained revealed that Rsubread was more sensitive to splice junction detection than HISAT2 and STAR (Table 9). HISAT2 had the lowest sensitivity, but measured up with Rsubread and STAR in the positive predictive value. This means that the algorithms used by the aligners have a high probability of predicting correct splice junctions (true positives) sites.

Previous study by Baruzzo et al[6] compared 14 splice aware aligners, but Rsubread had since been updated. Grant et al[10] developed a mapping algorithm and compared it with other algorithms, but none of the aligners in this study was included in their study. Engström et al[1] did a comprehensive comparison on splice aligners' algorithm, but Rsubread and HISAT2 were not included. Liao et al[12] compared other aligners, including Rsubread and STAR, but HISAT2 was not included in their study. Krizanovic et al[27] evaluated splice aligners, but Rsubread was not among the compared algorithms. The majority of results from this study agree with previous studies[6].

STAR generates a detailed and highly understandable statistics, which is unlike the rest. On the other hand, it will really pose a difficult challenge for a beginner bioinformatician to understand and use the results output from HISAT2 if there are no modifications for ease of understanding. Also, modifying the algorithm to output BAM file directly instead of engaging it in a subsequent process of converting SAM file to BAM would be beneficial. This would reduce the memory resource allocation and the alignment process would be more simplified than it currently is. BAM file format is mostly used in downstream analyses unlike SAM format and the conversion to BAM increases runtime, as it becomes an additional process time[9]. Rsubread has undergone modifications that require quality control investigations in order to have increased benchmark recommendation. The algorithm of the aligners can be further modified for standardization purposes.

5. Conclusion

Alignment of sequenced reads to a reference genome enhances alignment-guided downstream analyses[6]. However, this does not come without obstacles as the results produced, from this complex step of alignment in RNA-seq data analysis, are greatly affected by the presence of splice, polymorphism and other variants that characterize real data[2,6-7,10,13,26]. Any aligner that can be used to minimize the impact of splicing on the output of RNA-seq reads alignment should have the capacity to detect these splice junctions, output a good percentage of overall alignment and perform optimally with default settings. The percentage of correctly detected splice junctions output by the aligner is a pointer to the aligner's sensitivity to the splice junctions in the data. Undoubtedly, the use of simulated data is preferred in order to ascertain these results, since the observable trends in alignment with simulated data is equivalent to the performance of alignment using real RNA-seq data[1].

This study aimed at comparing the modified versions of Rsubread, STAR and HISAT2 to show their sensitivity level to splice junction detection and if their current alignment performance agrees with the claims of the developers. In achieving this aim, the objectives applied were calculating the sensitivity of the aligners to splice sites, finding out the aligners'

output with default parameter settings and investigating the aligners' resource usage. Results from this study have shown that all three aligners are sensitive, but STAR and Rsubread had higher splice junction detection output than HISAT2, despite HISAT2's low resource usage.

The contribution of this research would lead to the modification or acceptance of STAR or Rsubread as a standard tool for spliced reads alignment. The sensitivity, greater number of assigned reads to their exon feature, and consistency of the aligner in most comparative studies are among the factors that give STAR an edge. Possible hybrid modifications to the algorithm of the aligners could improve their sensitivity and performance. The suffix array indexing of STAR is dependable, but could be improved upon to reduce the resource usage since it is notorious for its huge demand on alignment resources. Therefore, there is need for further development on the aligners' algorithm, but with the intent of creating a benchmark algorithm for the alignment step, which is basic to the downstream analysis of RNA-seq data.

6. Novelty of Methods or Results

Although there have been some benchmark studies previously, the pipeline of specifically mapping reads with HISAT2, STAR, Rsubread and quantifying the mapped reads with *featureCounts* to adopt a benchmark tool has not been done before. To the best of my knowledge, earlier benchmarking studies employed different pipelines other than the one used in this study. Baruzzo et al[6] carried out a comprehensive study on 14 splice aligners. However, there have been much improvement in Rsubread after that study and *featureCounts* was not matched with the aligners. Krizanovic et al[27] studied tools for long RNA-seq reads splice alignment, but Rsubread was not among the tools in their study and *featureCounts* was not used for quantifying the aligned reads. Engström et al's[1] detailed research on spliced alignment algorithms did not include HISAT2, Rsubread and the read quantification tool used in this study in his work. Liao et al[12], after the improvements in Rsubread, carried out a study to compare the improved Rsubread package with other aligners but did not incorporate HISAT2 in his study.

7. Ethical Aspects and Impact on Society

In fulfilling the principles of research ethics, there were guidelines followed during this research. The understanding of research ethics as a fundamental aspect of the study, to be complied with from the starting to the finishing point, served as a guide to this research. The study was carried out with objectivity from the literature review, pipeline design, implementation, results reporting and analysis. Reviewed literatures were cited as required by the authors. The data was used with the author's article permission. These sensitive aligners were compared by a researcher, without any benefits from the algorithm developers, to reveal the outputs obtained from the study as they are. From the results and analysis of this study, bioinformaticians, researchers and algorithm designers can be inspired to more comprehensively study these aligners on different scale and data. Next, there is a possibility of creating a new or hybrid technology to close the gap for the race of a standard splice-aware aligner that has been narrowed by this research.

RNA-seq has clinical applications and as such improving alignments to be more sensitive to splice events is beneficial to clinical routines[38]. The detection and interpretation of the splicing events in the data increases the rate of diagnosis by about 10% to 35%. Alignment is critical to downstream analyses and splicing event is one of the challenges faced in mapping reads to a reference genome. Therefore, sensitivity to the presence of splices will enrich the quality of transcripts obtained after the step of alignment.

8. Future Directions

With consideration to the scope of this study, guidelines for a future study would include these four aspects. As a future step of this study, a more comprehensive study of these aligners to determine the accuracy of their alignments. Parameters relevant to improved alignment of spliced data for each tool would be tweaked to discover any impact on the sensitivity of the aligner to spliced junctions. Quantifying the aligned reads with benchmarked quantification software would be necessary to determine the assignment of the aligned reads to the genomic feature, i.e. exons in this case. Secondly, the performance of the aligners would be checked with much increased or decreased amounts of data in order to discover the effect of scaled data on the aligners, since some of them have better results with increased or reduced data. Thirdly, further statistical and downstream analyses would be beneficial to determine if the aligner's sensitivity has any improved or reduced impact on the results of these analyses. Lastly, other public data would be used to check the consistency of the improved aligner's sensitivity to splice sites.

9. Acknowledgements

In the academic part of this research, I acknowledge the contribution of Liao et al[12] in making the data available. I would like to express my heartfelt gratitude to my supervisor, Angelica Lindlöf for her dedication and commitment to giving timely and valuable feedbacks, encouragements and ensuring that the resources needed for this study were accessible. I deeply appreciate the objective and inspirational contributions of my examiner, Björn Olsson to this research. My appreciation also goes to the lecturer, Zelmina Lubovac for the detailed research lecture and for exposing me to research ethics.

My deepest gratitude goes to God Almighty for the inspiration and wisdom during the research. I would like to specially appreciate my husband for his sacrifices and encouragement while carrying out the study.

10. References

1. Engström PG, Steijger T, Sipos B, Grant GR, Kahles A, The RGASP Consortium, et al. Systematic evaluation of spliced alignment programs for RNA-seq data. Nat Methods. 2013;10(12):1185-1191.
2. Williams AG, Thomas S, Wyman SK, Holloway AK. RNA-seq data: challenges in and recommendations for experimental design and analysis. Current Protocols in Human Genetics. 2014;83:11.13.1-11.
3. Kukurba KR, Montgomery SB. RNA sequencing and analysis. Cold Spring Harb Protoc. 2015;11:951-969.
4. Vikman P, Fadista J, Oskolkov N. RNA sequencing: current and prospective uses in metabolic research. Journal of Molecular Endocrinology. 2014;53(2):93-101.
5. Jun H, Huanying G, Newman M, Kejun L. OSA: a fast and accurate alignment tool for RNA-seq. *Bioinformatics*.2012;28(14):1933-1934.
6. Baruzzo G, Hayer KE, Kim EJ, Di Camillo B, FitzGerald GA, Grant GR. Simulation-based comprehensive benchmarking of RNA-seq aligners. Nat Methods. 2017;14(2):135-139.
7. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*. 2013;29(1):15–21.
8. Kim D, Langmead B, Salzberg SL. HISAT: Hierarchical Indexing for Spliced Alignment of Transcripts.bioRxiv. 2014;012591.
9. Dobin A, Gingeras TR. Mapping RNA-seq reads with STAR. Current protocols in bioinformatics. 2015; 51:11.14.1–11.14.19.
10. Grant GR, Farkas MH, Pizarro AD, Lahens NF, Schug J, Brunk BP, Pierce EA. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). Bioinformatics (Oxford, England).2011; 27(18):2518–2528.
11. Garber M, Grabherr MG, Guttman M, Trapneil C. Computational methods for transcriptome annotation and quantification using RNA-seq. Nat Methods. 2011;8:469-477.
12. Liao Y, Smyth GK, Shi W. The R package *Rsubread* is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. *Nucleic Acids Research*. 2019;47(8):e47.
13. Conesa A, Madrigal P, Tarazona S, Gomez-Cabrero D, Cervera A, McPherson A, Sczesniak MW, Gaffney DJ, Elo LL, Zhang X, Mortazavi A. A survey of best practices for RNA-seq data analysis. Genome Biology. 2016;17:13.
14. Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzeberg S. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biology. 2013;14:R36.
15. Gatto A, Torroja-Fungairiño C, Mazzarotto F, et al. FineSplice, enhanced splice junction detection and quantification: a novel pipeline based on the assessment of diverse RNA-Seq alignment solutions. Nucleic Acids Res. 2014;42(8):e71.
16. Bonfert T, Kirner E, Csaba G, Zimmer R, Friedel CC. ContextMap 2: fast and accurate context-based RNA-seq mapping. BMC Bioinformatics. 2015;16:122.
17. Philippe N, Salson M, Commes T, Rivals E. CRAC: an integrated approach to the analysis of RNA-seq reads. Genome Biol. 2013;14:R30.
18. Wu TD, Nacu S. Fast and SNP-tolerant detection of complex variants and splicing in short reads. Bioinformatics. 2010;26:873–881.

19. Kim D, Langmead B, Salzberg SL. HISAT: a fast spliced aligner with low memory requirements. Nature Methods. 2015;12:357.
20. Trapnell C, Pachter L, Salzberg S. TopHat: discovering splice junctions with RNA-seq. Bioinformatics. 2009;25(9):1105-1111.
21. Wu J, Anczuków O, Krainer AR, Zhang MQ, Zhang C. OLego: fast and sensitive mapping of spliced mRNA-Seq reads using small seeds. Nucleic Acids Res. 2013;41:5149–5163.
22. Huang S, et al. SOAPsplice: Genome-wide *ab initio* detection of splice junctions from RNA-Seq data. Front Genet. 2011;2:46.
23. Liao Y, Smyth GK, Shi W. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. Nucleic Acids Res. 2013;41:e108.
24. Wang W, Wu C, Lu T, Tsai M, Lai L, Chuang EY. Comparisons and performance evaluations of RNA-seq alignment tools. *2014 International Conference on Electrical Engineering and Computer Science (ICEECS).*2014;215-218.
25. Hong JH, Ko YH, Kang K. RNA variant identification discrepancy among splice-aware alignment algorithms. PloS ONE. 2018;13(8): e0201822.
26. Raplee ID, Evsikov AV, Marin de Evsikova C. Aligning the Aligners: Comparison of RNA sequencing data alignment and gene expression quantification tools for clinical breast cancer research. J Pers Med. 2019;9(2):18.
27. Krizanovic K, Echchiki A, Roux J, Sikic M. Evaluation of tools for long read RNA-seq splice-aware alignment. Bioinformatics. 2018;34(5):748-754.
28. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics. 2014;30(7):923-930.
29. Harvard Chan Bioinformatics Co. Accessed 2020-03-05. Available at :https://hbctraining.github.io/Intro-to-rnaseq-hpc-O2/lessons/03_alignment.html.
30. FMindex. Accessed 2020-03-06. Available at: http://people.unipmn.it/manzini/fmindex/.
31. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012;9(4):357-359.
32. John Hopkins University. Accessed 2020-03-10. Available at:http://daehwankimlab.github.io/hisat2/.
33. Weber LM, Saelens W, Cannoodt R, Soneson C, Hapfelmeier A, Gardener PP et al. Essential guidelines for computational method benchmarking. Genome Biol. 2019;20:125.
34. Columbia Genome Centre. Genome Sequencing: Defining Your Experiment. Accessed 2020-03-16. Available at: https://systemsbiology.columbia.edu/genome-sequencing-defining-your-experiment
35. SEQC Consortium. A comprehensive assessment of RNA-seq accuracy, reproducibility and information content by the Sequencing Quality Control consortium. Nat Biotechnol. 2014;32(9):903-914.
36. Dobin A. STAR manual 2.7.0a. 2019. Accessed 2020-03-28. Available at:https://physiology.med.cornell.edu/faculty/skrabanek/lab/angsd/lecture_notes/STARmanual.pdf.
37. Thankaswamy-Kosalai S, Sen P, Nookaew I. Evaluation and assessment of read-mapping by multiple next-generation sequencing aligners based on genome-wide characteristics. Genomics. 2017;109(3-4):186-191.
38. Marco-Puche G, Lois S, Benitez J, Trivino JC. RNA-seq perspectives to improve clinical diagnosis. Frontiers in Genetics. 2019;10.