


DOI: [https://dx.doi.org/10.21123/bsj.2021.18.4\(Suppl.\).1413](https://dx.doi.org/10.21123/bsj.2021.18.4(Suppl.).1413)

## Fast Processing RNA-Seq on Multicore Processor

Lee Jia Bin <sup>1</sup>

Nor Asilah Wati Abdul Hamid <sup>1,2\*</sup> 

Zurita Ismail <sup>2</sup> 

Mohamed Faris Laham <sup>2</sup> 

<sup>1</sup>Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia.

<sup>2</sup>Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM, Serdang, Selangor, Malaysia

\*Corresponding author: [asila@upm.edu.my](mailto:asila@upm.edu.my)

E-mails: [193905@student.upm.edu.my](mailto:193905@student.upm.edu.my), [zurita@upm.edu.my](mailto:zurita@upm.edu.my), [mohdfaris@upm.edu.my](mailto:mohdfaris@upm.edu.my)

Received 14/10/2021, Accepted 14/11/2021, Published 20/12/2021



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).

### Abstract:

RNA Sequencing (RNA-Seq) is the sequencing and analysis of transcriptomes. The main purpose of RNA-Seq analysis is to find out the presence and quantity of RNA in an experimental sample under a specific condition. Essentially, RNA raw sequence data was massive. It can be as big as hundreds of Gigabytes (GB). This massive data always makes the processing time become longer and take several days. A multicore processor can speed up a program by separating the tasks and running the tasks' errands concurrently. Hence, a multicore processor will be a suitable choice to overcome this problem. Therefore, this study aims to use an Intel multicore processor to improve the RNA-Seq speed and analyze RNA-Seq analysis's performance with a multiprocessor. This study only processed RNA-Seq from quality control analysis until sorted the BAM (Binary Alignment/Map) file content. Three different sizes of RNA paired end has been used to make the comparison. The final experiment results showed that the implementation of RNA-Seq on an Intel multicore processor could achieve a higher speedup. The total processing time of RNA-Seq with the largest size of RNA raw sequence data (66.3 Megabytes) decreased from 317.638 seconds to 211.916 seconds. The reduced processing time was 105 seconds and near to 2 minutes. Furthermore, for the smallest RNA raw sequence data size, the total processing time decreased from 212.380 seconds to 163.961 seconds which reduced 48 seconds.

**Keywords:** Bioinformatics, High-performance computing, Multicore processors, RNA Sequencing.

### Introduction:

RNA molecules are collectively known as the transcriptome because RNA will undergo a transcription process that will essentially take the information encoded in the gene in DNA and encode that same information in mRNA. The mRNA will then migrate out of the nucleus to a ribosome and turning into an amino acid. A sequence of amino acids will construct a protein that plays a crucial role in repairing the human body's tissue, taking place the metabolic reactions, and coordinating body functions. If there is no RNA to proceed with transcription, the human body cannot maintain a proper pH and fluid balance <sup>1</sup>.

RNA-Seq is the sequencing and analysis of transcriptomes. RNA-Seq analysis is widely used in basic research, clinical diagnosis, research and development of a drug, and other fields. For example, RNA-Seq helps identify hormone-related

genes associated with the prognosis of triple-negative breast cancer <sup>2</sup>. In addition, RNA-Seq offers a significant increase in the rate of diagnosis of mendelian muscle disorders <sup>3</sup>. However, the massive memory of RNA raw data has caused problems for biologists. An RNA raw data file can range from a few Mb to as many as hundreds of Gb.

Since the process of RNA sequence analysis from preprocessing, mapping, quantification until differential expression analysis will take several hours until a few days. Hence, it will become time-consuming. There must be some ways to speed up the RNA-Seq process as faster as possible since the RNA data amount cannot be reduced, such as using a multicore processor. A multicore processor is a single integrated chip that comprises more than one core processing unit. By running several tasks concurrently, the multicore processor can improve

the processing speed. Compared to using separate processors, the reduced distance between cores on an integrated circuit allows a shorter latency of resource access and a higher cache speed.

Nevertheless, the size of the performance increase mainly depends on the number of cores, the use of shared resources, and the level of real concurrency in the actual software, see <sup>4</sup> and <sup>5</sup>. <sup>6</sup> introduced the message-passing algorithm extends the multi-threaded workflow-based HPG Aligner BW to map short reads onto a reference genome on a cluster of multicore processors. Reads are distributed among the cluster servers in the multi-node implementation, which perform asynchronously for most of the execution, following HPG Aligner BW's original workflows.

Many studies have been conducted to improve the speed of RNA-Seq process, for instance <sup>7</sup>, <sup>8</sup> and <sup>9</sup>. Hence, this study focused on using a multicore processor to speed up the RNA-Seq process to help the researchers obtain the result file in a shorter time.

### Methodology:

Several procedures are carrying out in this study. Firstly, the preparation for RNA-Seq. The preparation work included learning knowledge relating to RNA-Seq, recognizing the tools required and its usage method, understanding RNA raw sequence data, and other necessary reference genome file and annotation files. Next, a set up for the project environment such as installation on Ubuntu system, bioinformatics tools, and data. Then, processed RNA-Seq from quality control analysis stage until sorting stage (see Figure 1) with 1, 2, and 4 threads. To ensure output accuracy, RNA-Seq was running three times for the different number of threads, respectively. The average execution time and speedup have been calculated, recorded, and used to plot a line graph. The above process (RNA-Seq from quality control analysis stage until sorting stage) was repeated for the other two datasets with bigger data sizes. After that, the total average execution time of RNA-Seq has been calculating for each stage's average execution time. The final speedup calculated then used to plot a graph.

Furthermore, an extra workflow involved only the quality control analysis step (see Figure 2) with 1, 2, and 4 threads. RNA-Seq was running three times for the different number of threads, respectively. Then, calculate the average execution time and speedup, recorded, and used to plot a line graph. This extra workflow repeated for 2 and 4 sets of RNA raw sequence data. In the end, comparison

and analysis based on the average execution time graph and speedup graph of all tools and RNA-Seq have been displayed.

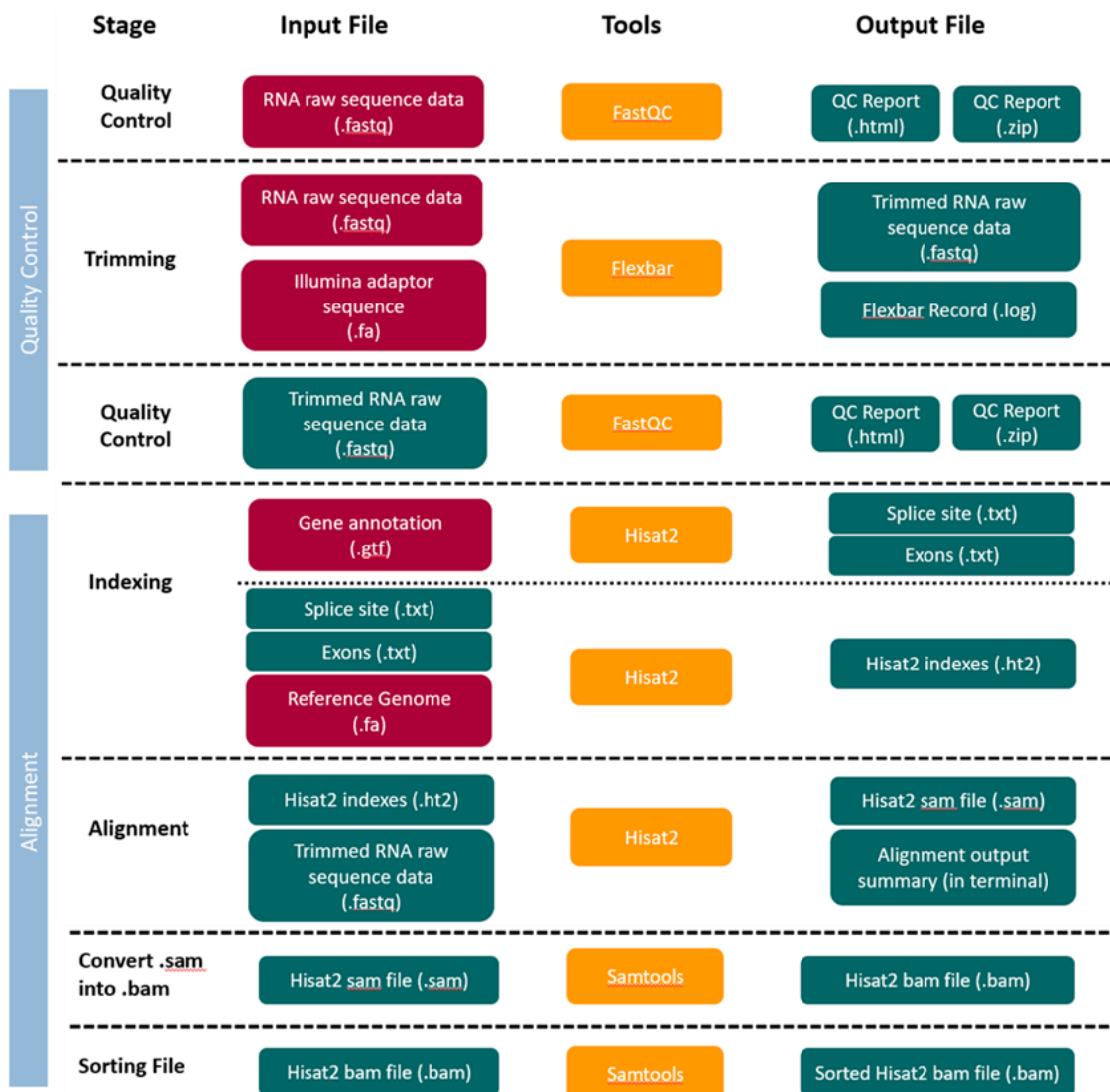


Figure 1. RNA-Seq workflow

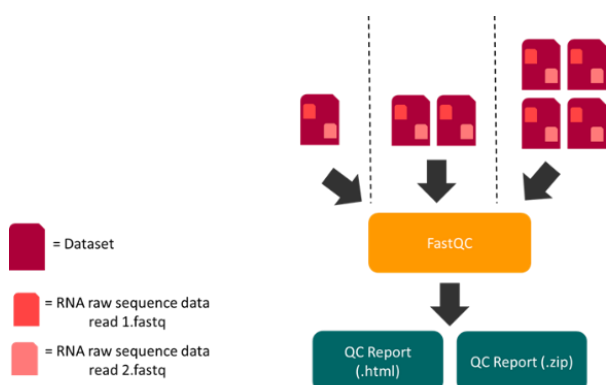


Figure 2. Extra Quality Control Analysis

These studies have been conducted in 64-bit Ubuntu 18.04 built-in Oracle VM VirtualBox 6.0.12 installed in 64-bit Windows 10 Home system. The Ubuntu system installed had 150 GB of storage, 5193 MB of memory (RAM), and used Intel (R)

Core (TM) i5-8265U CPU with 1.80 GHz of the base speed. The total available number of threads was four.

The bioinformatics tools used in this study were FastQC (version 0.11.9)<sup>10</sup>, Flexbar (version 3.5.0)<sup>11</sup>, Hisat2 (version 2.2.1)<sup>12</sup>, and Samtools (version 1.11)<sup>13</sup>. The tools have been downloaded from the developer's website to ensure that all tools used were in the latest version. This study used three sets of paired-end data for the main workflow, while the other four sets of paired-end data have used for the different workflow. All the RNA raw sequence data used the same gene annotation file, reference genome file, and adapter sequence file. All files and data in this study were obtained from the RNA-Seq online tutorial under Griffithlab on the Github website<sup>14</sup>.

**Table 1. Function and parallelism of bioinformatics tools**

RNA-Seq Tools	Function	Parallelism
FastQC	<ul style="list-style-type: none"> <li>Quality Control</li> </ul>	<ul style="list-style-type: none"> <li>Multithread (Only assign one thread to one input file)</li> </ul>
Flexbar	<ul style="list-style-type: none"> <li>Trimming</li> </ul>	<ul style="list-style-type: none"> <li>SIMD vectorization</li> <li>Multithread</li> </ul>
Hisat2	<ul style="list-style-type: none"> <li>Indexing</li> <li>Alignment</li> </ul>	<ul style="list-style-type: none"> <li>Multithread</li> <li>Tradeoff between runtime and memory</li> </ul>
Samtools	<ul style="list-style-type: none"> <li>Convert SAM file into BAM file</li> <li>Sorting BAM file</li> </ul>	<ul style="list-style-type: none"> <li>Multithread</li> </ul>

As shown in Table 1, this study used FastQC to do RNA raw sequence data quality control analysis followed by Flexbar, which used to trim the insufficient quality data. This can ensure the contamination data will not influence the other analysis result. After that, Hisat2 used indexing to generate indexes file and alignment between those indexes file and RNA raw sequence file. At the end of this study, Samtools converted the SAM file into BAM file and sorted the BAM file content.

All these tools use different parallelism methods to achieve a different purpose. FastQC

uses multithread but can only assign one thread to one file. Flexbar uses Single-Instruction-Multiple-Data (SIMD) vectorization and multithread method to speed up the trimming process. Hisat2 can trade-off between runtime and memory and uses multithread to boost up the processing speed. Meanwhile, Samtools also uses multithread technology to increase processing speed. Although all tools used the multithread method, only FastQC cannot assign more than one thread to one file.

**Table 2. RNA-Seq dataset for main workflow**

RNA-Seq dataset	Data	Total of read files	Data Size (MB)
1	Human Brain Reference (HBR) with ERCC ExFold RNA Spike-In Control Mixes 2 Replicate 1	2	13.6
2	Universal Human Reference (UHR) with ERCC ExFold RNA Spike-In Control Mixes 1 Replicate 1	2	28.0
3	HCC1395 breast cancer cell line replicate 1	2	66.3

**Table 3. RNA-Seq Dataset for Extra Workflow**

Number of RNA dataset	Data	Total of read files	Data Size (MB)
1	<ul style="list-style-type: none"> <li>HCC1395 breast cancer cell line replicate 1</li> </ul>	2	66.2
2	<ul style="list-style-type: none"> <li>HCC1395 breast cancer cell line replicate 1</li> <li>HCC1395 breast cancer cell line replicate 2</li> </ul>	4	131.3
4	<ul style="list-style-type: none"> <li>HCC1395 breast cancer cell line replicate 1</li> <li>HCC1395 breast cancer cell line replicate 2</li> <li>HCC1395BL matched lymphoblastoid line replicate 1</li> <li>HCC1395BL matched lymphoblastoid line replicate 2</li> </ul>	8	243.6

Table 2 shows the RNA raw sequence data used in the main workflow, while Table 3 listed the extra workflow.

**Table 4. Other data needed in RNA-Seq**

Other Data Type	Data	Data Size (MB)
Gene Annotation	Annotation of human GRCh38 chromosome 22 and the ERCC spike-in sequences	30.7 MB
Reference Genome	Genome of human GRCh38 chromosome 22 with the ERCC spike-in sequences	51.8 MB
Adapter Sequence	Illumina adapter sequence	161

Table 4 lists the other data used in this study including the Gene Annotation file, Reference Genome file, and Adapter Sequence file.

### Results and Discussion:

In this section, the average execution time graph and speedup graph of each stage and RNA-Seq will be illustrated, compared, and explained. A summary based on the performance of different tools on multicore processors will be made at the end.

### Quality Control

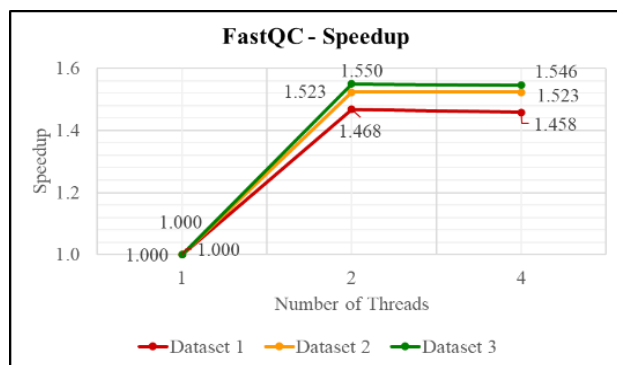


Figure 3. Speedup of FastQC During Quality Control

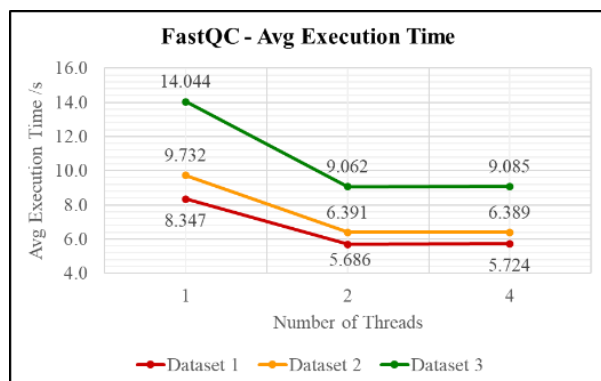


Figure 4. Average Execution Time of FastQC During Quality Control

Based on Figure 3, the average execution time of all three datasets decreased when using two threads then remained almost unchanged when using four threads. This was because FastQC will only assign one thread to one input file, although more than one thread is available. All three datasets were paired-end data. Therefore, each had one pair of read files. When using one thread, only this single thread handled two read files. When using two threads, two read files can be processed concurrently since one thread handled one read file, respectively. When using four threads, only two threads will obtain one task respectively, and the other two threads will stay idle. Hence, the average

execution time of using two and four threads had no significant differences. Figure 4 shows the average execution time results of three datasets caused their speedup to be increased when using two threads then remained almost unchanged when using four threads.

### Trimming

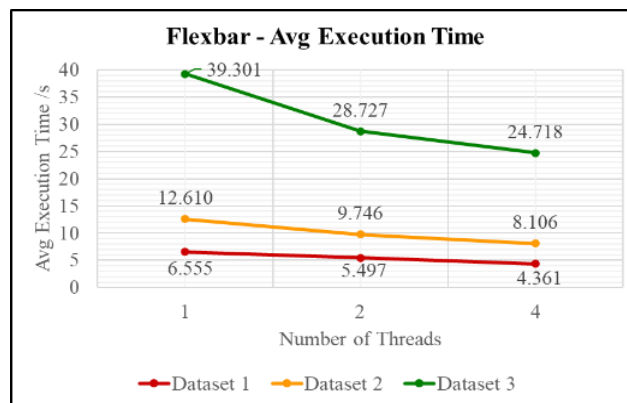


Figure 5. Average Execution Time of Flexbar During Trimming

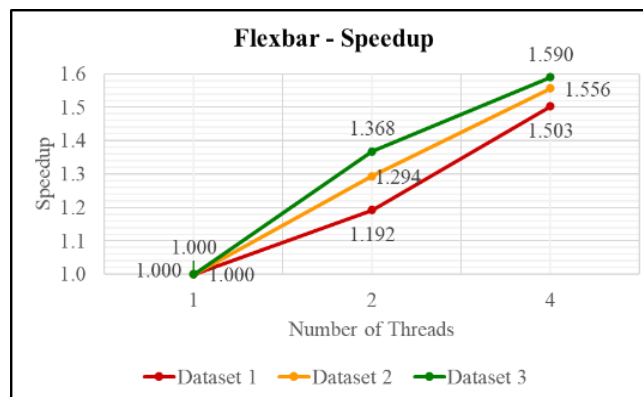
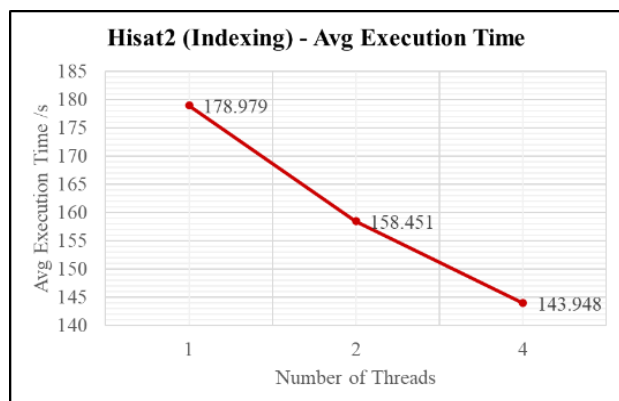


Figure 6. Speedup of Flexbar During Trimming

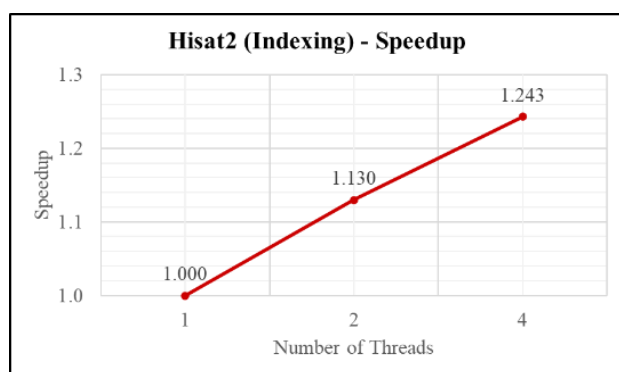
Figure 5 shows the average execution time of all three datasets showed a similar trend that declined with the increase of threads. This was because Flexbar can keep all thread's load balance and distribute more than one thread to one task. The same reason explained that the speedup of all datasets increased when using more threads, as shown in Figure 6.

Here had to mention that the two performance graphs showed in above were one possible result of Flexbar. Sometimes, dataset 1 achieved the highest speedup when using two and four threads, while sometimes dataset 2 achieved the highest speedup. This might be because the background program influenced the performance of Flexbar. The insufficient memory caused Flexbar to give an inconsistent speedup.

**Indexing**



**Figure 7. Average Execution Time of Hisat2 During Indexing**

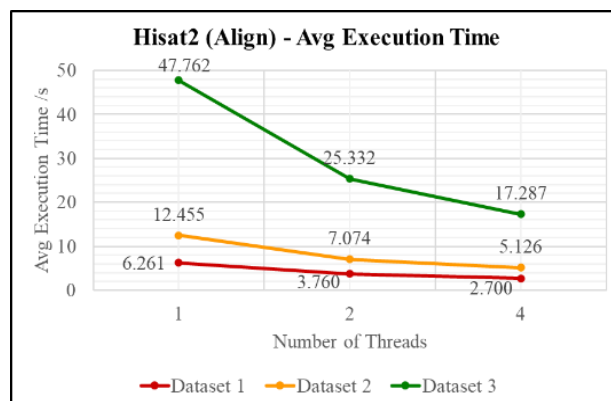


**Figure 8. Speedup of Hisat2 During Indexing**

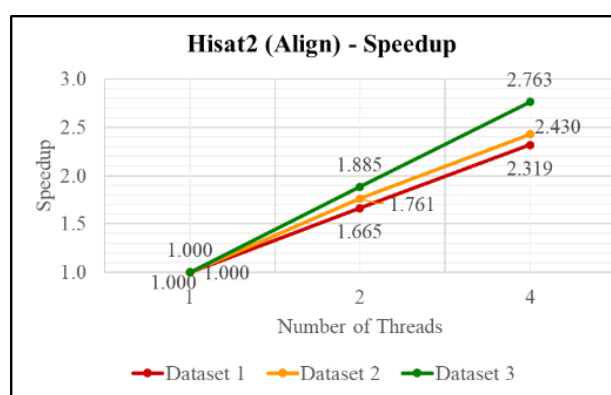
Since three datasets used the same reference genome file and annotation file, only one set of index files was built from these two files. Therefore, the two graphs above only had one line. Based on Figure 7, the average execution time decreased with the increased number of threads used. This decrease helped to speed up the indexing process, as shown in Figure 8.

Figures 9 and 10 show a big difference in the performance of Hisat2 in the alignment step and indexing step. It might be because the input data in the alignment step was parallelizable at a higher rate than the indexing step. Similar to the Flexbar and Hisat2, in the average execution time graph of Samtools, the average execution time decreased along with the increased threads used, as shown in Figure 11. Moreover, Figure 12 shows the speedup of Samtools was higher than Hisat2 speedup and very near to the superlinear speedup.

**Alignment**

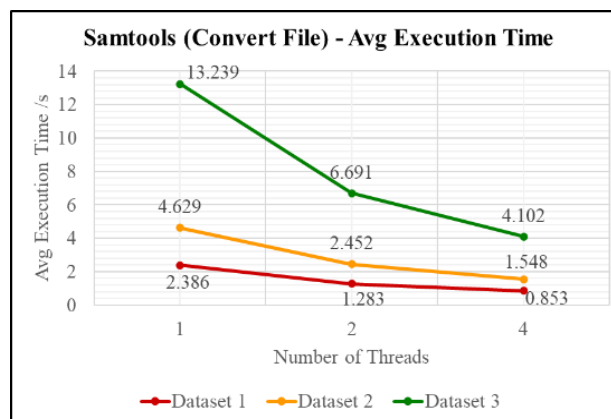


**Figure 9. Average Execution Time of Hisat2 During Alignment**



**Figure 10. Speedup of Hisat2 During Alignment**

**File format conversion**



**Figure 11. Average Execution Time of Samtools During Converting File Format**

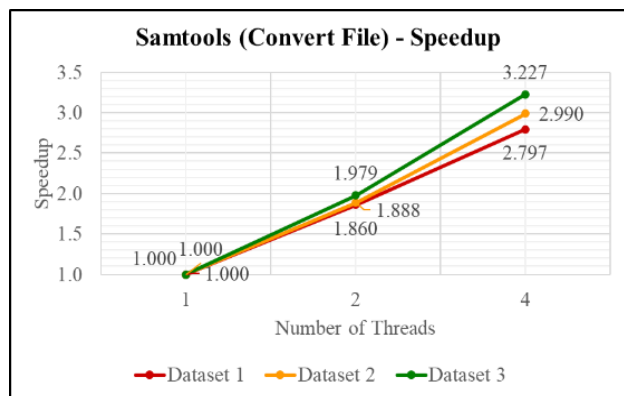


Figure 12. Speedup of Samtools During Converting File Format

Sorting

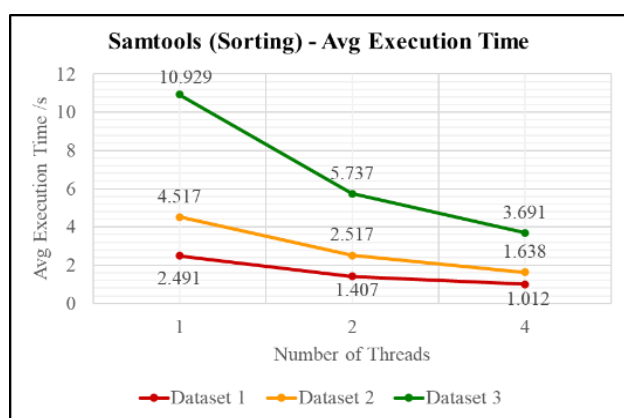


Figure 13. Average Execution Time of Samtools During Sorting File

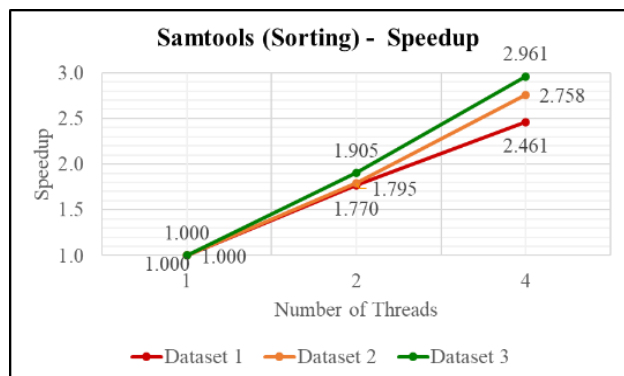


Figure 14. Speedup of Samtools During Sorting File

Both graphs in Figures 13 and 14 show the same trend of the performance in converting the file format of Samtools as the sorting step. This defined that the average execution time of all datasets decreased when the number of threads was used increasingly. On the other hand, the speedup of Samtools of all datasets is still very near to the superlinear speedup.

RNA-Seq (Until Sorting Stage)

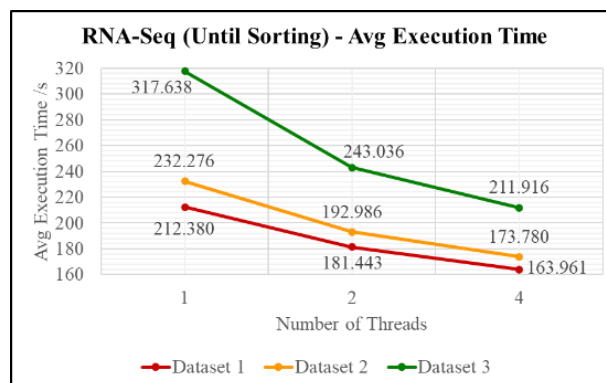


Figure 15. Average Execution Time of Processed RNA-Seq on Multicore Processor

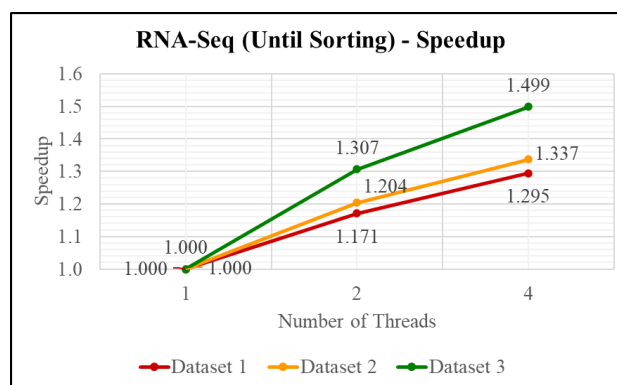
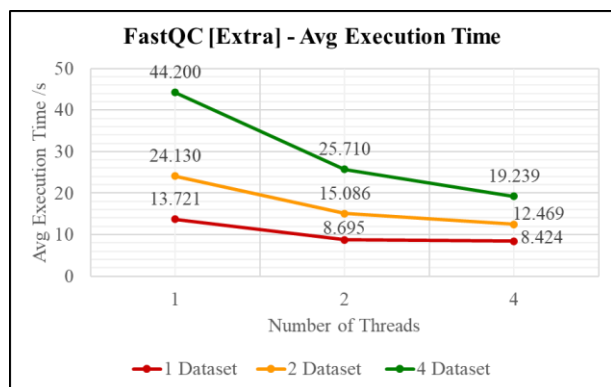


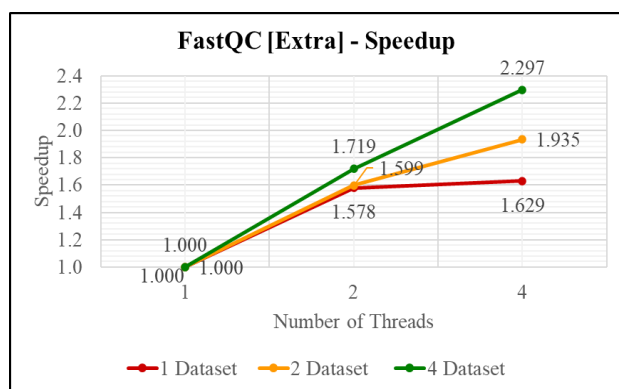
Figure 16. Speedup of Processed RNA-Seq on Multicore Processor

According to Figure 15, the bigger the data, the more the decreased average execution time. When increased the number of threads from one to four, the average execution time of dataset 3 decreased by almost two minutes (105 seconds). Besides, all datasets achieved an increase of speedup when using more threads, as shown in Figure 16.

**Extra Quality Control**



**Figure 17. Average Execution Time of FastQC during Extra Quality Control**



**Figure 18. Speedup of FastQC during Extra Quality Control**

Based on Figure 17, the average execution time of FastQC to process one dataset decreased when using two threads then remained almost unchanged when using four threads. This was because one dataset had two read files while FastQC only assigns one thread to one input file. Although four threads were generated, only two threads will handle the task concurrently, and the other two threads get no job. However, when FastQC processed with two datasets with four read files, FastQC average execution time decreased along with the increased threads. When the number of datasets increased to four, which had eight read files, the decreased the average execution time of FastQC became more apparent. The speedup result of processing one dataset and four datasets in Figure 18 shows that a large number of data could achieve a higher speedup.

**Table 5. Comparison RNA-Seq speed with and without using multicore processor.**

RNA-Seq dataset	Data Size (MB)	RNA-Seq speed (seconds)	
		Without multicore processor	With multicore processor
Human Brain Reference (HBR) with ERCC ExFold RNA Spike-In Control Mixes 2 Replicate 1	13.6	212.380	163.961
Universal Human Reference (UHR) with ERCC ExFold RNA Spike-In Control Mixes 1 Replicate 1	28.0	232.276	173.780
HCC1395 breast cancer cell line replicate 1	66.3	317.638	211.916

Table 5 reported the speed improvement of RNA-Seq with and without using the multicore processor. It clearly shows that the smallest RNA data (13.6 MB) with a total processing time reduced from 212.380 seconds to 163.961 seconds. While the second biggest data (28.0 MB) with whole processing time reduced 58 seconds from 232.276 seconds to 173.780 seconds, and the biggest RNA data (66.3 MB) with total processing time reduced by 105 seconds from 317.638 seconds to 211.916 seconds.

**Conclusion:**

To summarize, a bigger data size and a larger number of data can decrease more execution time

and achieve higher speedup. This condition happened when using different tools in different RNA-Seq stages. By comparing FastQC, Flexbar, Hisat2, and Samtools in this study, we have shown that Samtools achieved the highest speedup followed by Hisat2. Meanwhile, Hisat2 gave more speedup in the alignment step compared to in indexing step. The FastQC may have higher performance when processed with more data and bigger data size. Even though Flexbar had an unstable performance but still achieved a decrease in average execution time when using more threads. Therefore, RNA-Seq processing time was able to decrease when using more threads. This concludes



that a multicore processor can help to speed up the RNA-Seq process.

This study shows that the speed of RNA-Seq analysis can be improved using a multicore processor. Besides that, this study used paired-end RNA raw sequence data only, and the biggest data size used was only 66.3 MB. Moreover, this study also used up to four cores to process RNA-Seq. 5193 MB of RAM, which limits the performance of the multicore processor.

### Future Works:

Future work suggested using single-end RNA raw sequence data and a bigger data size of around 1GB or more, and the number of the core number may increase to more than four cores with a larger RAM size such as 8GB. The multicore processor can reduce more runtime for bigger data. In addition, a different or new algorithm can be proposed in future work to improve the existing algorithm in the RNA-Seq toolset.

### Acknowledgment:

This research was supported by the Geran Putra (GP/2020/9693400), funded by Universiti Putra Malaysia (UPM).

### Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- The author has signed an animal welfare statement.
- Ethical Clearance: The project was approved by the local ethical committee in University of Putra Malaysia.

### Authors' contributions:

Lee Jia Bin and Nor Asilah Wati Abdul Hamid conceived the study and design the simulation scenario. Zurita Ismail and Mohamed Faris Laham contributed to the analysis method. All authors provided feedback throughout the work and contributed to the writing of the manuscript.

### References:

1. Van De Walle G. 9 Important Functions of Protein in Your Body. Healthline. Retrieved September 1, 2020, from <https://www.healthline.com/nutrition/functions-of-protein>.
2. Chen F, Li, Y, Qin, N, Wang, F, Du, J, Wang, C, Du, F, Jiang, T, Jiang, Y, Dai, J, Hu, Z, Lu, C, Shen, H. RNA-seq analysis identified hormone-related genes

- associated with prognosis of triple negative breast cancer. *J Biomed Res.* 2020. 34(2), 129–138.
3. Wrighton KH. The diagnostic power of RNA-seq. *Nature Reviews Genetics.* 2017. 18(7), 392-392.
4. Chatterjee K. and Wan Y. RNA. *Encyclopedia Britannica.* 2018, July 13. Retrieved from <https://www.britannica.com/science/RNA>.
5. Firesmith D. Multicore Processing. Software Engineering Institute. 2017, August 21. Retrieved from [https://insights.sei.cmu.edu/sei\\_blog/2017/08/multicore-processing.html](https://insights.sei.cmu.edu/sei_blog/2017/08/multicore-processing.html).
6. Martínez, H., Barrachina, S., Castillo, M., Tárraga, J., Medina, I., Dopazo, J., Quintana-Ortí, E. S. Scalable RNA sequencing on clusters of multicore processors. 2015 IEEE Trustcom/BigDataSE/ISPA. IEEE. 2015. 3, 190-195.
7. Al-Ars, Z., Wang, S., & Mushtaq, H. SparkRA: enabling big data scalability for the GATK RNA-seq pipeline with apache spark. *Genes.* 2020. 11(1), 53.
8. Cascitti, J., Niebler, S., Müller, A., Schmidt, B. RNACache: Fast Mapping of RNA-Seq Reads to Transcriptomes Using MinHashing. In *International Conference on Computational Science.* Springer, Cham. 2021. 367-381.
9. Tran, S. S., Zhou, Q., Xiao, X. Statistical inference of differential RNA-editing sites from RNA-sequencing data by hierarchical modeling. *Bioinformatics,* 2020. 36(9), 2796-2804.
10. Andrews, S. FastQC: a quality control tool for high throughput sequence data [WWW document]. URL <http://www.bioinformatics.babraham.ac.uk/projects/fastqc>. 2010.
11. Roehr JT, Dieterich C, Reinert K. Flexbar 3.0—SIMD and multicore parallelization. *Bioinformatics.* 2017. 33(18), 2941-2942.
12. Kim D, Langmead B, Salzberg SL. HISAT: a fast spliced aligner with low memory requirements. *Nature methods,* 2015. 12(4), 357-360.
13. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The sequence alignment/map format and SAMtools. *Bioinformatics,* 2009. 25(16), 2078-2079.
14. Griffith M, Walker JR, Spies NC, Ainscough BJ, Griffith OL. Informatics for RNA-seq: A web resource for analysis on the cloud. *PLoS Comp Biol.* 2010. 11(8).

## المعالج متعدد النواة على RNA-SEQ السريعة المعالجة

لى جيا بين<sup>1</sup> نور عسيلة واتي عبد حميد<sup>1,2</sup> زوريتا اسماعل<sup>2</sup> محمد فارييس لاهم<sup>2</sup>

<sup>1</sup>قسم البلاخ التكنولوجيا والشبكة، فاكولتي من علوم الكمبيوتر وعلوم التكنولوجيا، الجامعة بوتر ماليزيا، 43400 سرداع، سلاور، ماليزيا.  
<sup>2</sup>المعهد للبحوث الرياضية، الجامعة بوتر ماليزيا، 43400 سرداع، سلاورا، ماليزيا.

### الخلاصة:

هو تسلسل وتحليل الترنسكربتوم RNA SEQUENCING (RNA-Seq) الغرض الرئيسي من التحليلات RNA-Seq هو معرفة وجود كمية من RNA في عينة تجريبية تحت ظروف معينة. بشكل أساسي، كانت بيانات التسلسل RNA الخام ضخمة. يمكن أن يصل حجمه إلى مئات الجيجابايت (GB). تعمل هذه البيانات الضخمة دائما على جعل وقت المعالجة أطول ويستغرق عدة أيام. يمكن للمعالج متعدد النواة (multicore processor) تسريع البرنامج عن طريق فصل المهام وتشغيل مهام المهام بشكل متزامن. وبالتالي، سيكون المعالج متعدد النواة (multicore processor) خيارا مناسباً للتغلب على هذه المشكلة. لذلك، تهدف هذه الدراسة إلى استخدام Intel multicore processor لتحسين سرعة RNA-Seq وتحليل أداء تحليل RNA-Seq باستخدام معالجات متعددة (multiprocessor). عالجت هذه الدراسة RNA-Seq فقط من تحليل مراقبة الجودة حتى فرز محتوى ملف BAM (Binary Alignment/Map). تم استخدام ثلاثة أحجام مختلفة من نهاية اقتران RNA لإجراء المقارنة. أظهرت نتائج التجربة النهائية أن تنفيذ RNA-Seq على معالج Intel Multicore يمكن أن يحقق سرعة أعلى في العملية. انخفض إجمالي وقت المعالجة لبيانات تسلسل RNA الخام (66.3 ميغابايت) من 317.638 ثانية إلى 211.916 ثانية. كان وقت المعالجة المخفض 105 ثانية، أي ما يقرب من دقيقتين. وعلاوة على ذلك، بالنسبة لأصغر حجم لبيانات تسلسل RNA الخام، انخفض إجمالي وقت المعالجة من 212.380 ثانية إلى 163.961 ثانية والذي تم تقليله بمقدار 48 ثانية.

**الكلمات المفتاحية:** المعلوماتية الحيوية، الحوسبة عالية الأداء، المعالجات متعددة النواة، تسلسل الحمض النووي الريبي.