An Analysis of Tophat: A Fast Splice Junction Mapper for RNA-Sequencing

Bashir Khan
Dept. of CS Engineering, UET
Peshawar, Pakistan
+92 322 9060477
malikbashirkhan1@gmail.com

Laiq Hasan
Dept. of CS Engineering, UET
Peshawar, Pakistan
+92 346 9311359
laiqhasan@uetpeshawar.edu.pk

Zahid Wadud
Dept. of CS Engineering, UET
Peshawar, Pakistan
+92 333 9211630
zahidmufti@nwfpuet.edu.pk

M. Yahya Bangash
Dept. of CS Engineering, UET Peshawar, Pakistan
+92 334 0915225
yahya_bungash@yahoo.com

Ikram Ullah
Dept. of Computer Science, University of Twente, Netherlands
+31 687451464
i.ullah@utwente.nl

ABSTRACT

In order to boost the theatrical performance of a TUX-E-DO Pipeline, we concentrated on the programs (tools) running within a pipeline to optimize their processing time. Initially we figured out the programs executing in the central part of the tuxedo pipeline which consume time more critically. We processed multiple raw RNA-Seq datasets on a tuxedo pipeline and recorded the time consumed by each tool to achieve this task. Therefore, we identified tophat as the maximum time consuming program (tool). Anyhow, tophat is a fast and efficient spliced aligner, as aligning RNA-Seq reads to a reference genome comparatively it consumes more time than the other programs. To find the logic behind the lengthy processing of tophat we executed multiple independent raw RNA-Seq data-sets by tophat used different number of threads and the execution-time of a data-set is recorded. As we know that, increasing the number of threads reduces the processing time. Contrarily, the results show that the processing time increases with increasing the number of threads. After the analysis and comprehensive simulations of the data processing-time of all data-sets, we found that between the threads there is a lack of communication and synchronization. To increase number of threads requires increase resolution of communication and synchronization. There is an enormous increase in alignment time resulting in processing time elongation.

CCS Concepts

•General and reference → Performance; **•Computing methodologies → Massively parallel and high-performance simulations.**

Keywords

NGS; RNA-Seq; Bioinformatics; High performance computing (HPC).

ICBBT '18, May 16–18, 2018, Amsterdam, Netherlands

ACM ISBN 978-1-4503-6366-2/18/05…$15.00
DOI: https://doi.org/10.1145/3232059.3232066

1. INTRODUCTION

Next Generation Sequencing (*N-G-S*) also known as High Through-put Sequencing (*HTS*). It's a modernized RNA-Sequencing (RNA-Seq) technology. Through NGS we can sequence DNA & RNA much more quickly and cheaply than the previously used sanger-sequencing Technology. Before next generation sequencing, sanger-sequencing were used which revolutionized the study of molecular biology and genomics. Using next generation sequencing with-in a single day we can sequence the human genome. To sequence human-genome with sanger-sequencing it takes huge amount of time. In parallel next generation sequencing sequences thousands of DNA small fragments [1]. Because of the accuracy, cost and speed the next generation sequencing succeeded sanger-sequencing.

RNA-Seq is a modernized technique uses next generation technology to discover, profile and quantify RNA's. It's a transcriptome-profiling approach. In a cell the study of all transcripts (m RNA molecules) is called transcriptomics. To set transcriptional structure of genes is the main goal of transcriptomics [2]. Gene Expression Micro-array was used before RNA-Seq. RNA-Seq substituted gene expression micro-array because its accurate, efficient, cost effective and fast. It allows researchers to find in a single assay both known and novel features, also allows us to discover single nucleotide variants *(SNPs)* [3].

Tuxedo pipeline provides a complete RNA-Sequencing analysis workflow, so it is appropriate for RNA-Sequencing experiments [7]. It is used to find unique splice-variants and genes, and will also correlate transcripts-expressions and genes [1]. It starts from raw RNA-Sequencing reads and ends with the publication ready visualisation of analysis results. It explains how to perform RNA-Seq analysis to use cufflinks and tophat. These tools perform multiple tasks such as gene and transcript quantifications, genome-annotation, and read alignment.

TopHat a splice-junction mapper for rna-sequencing read. It uses bowtie (short read aligner) to map rna-sequencing reads to the reference genome. It also analyzes mapping results to identify splice-junctions between exon [4]. For alignment-engine bowtie is used [8]. Its annotation independent it finds novel-exons and new splice-sites missing in gene-annotation.

The main step in all these analysis workflows such as chip sequencing, ribosome profiling and rna-sequencing is to align reads to reference genome.

Tophat is suitable for all experiments of RNA-Seq. Its alignment reports are precise comparative to other tools. Now-a-days all biological scientists are using rna-sequencing experiments. It gives us authentic results in rna-seq experiments. It also processes the data-sets having reads that's of variable-length [5].

1.1 Objectives of Tophat

Top-hat is designed on the cap of bow-tie. Over small indels it aligns reads with precisions. It's a factor which is necessary to study the effects of genetic-mutations on genes and transcript expressions. With rna-sequencing experiments it performs well. Its designed to align reads of long paired end instead of two or one splicing-sites. It creates a new challenge by spanning multiples splicing-sites; i.e. we assume the sequences (reads) of length 150 bp spans multiple exons of human. To keep accuracy and speed fixed, the tophat algorithmic improvements address the challenge.

Upto 98 % tophat precisely maps reads, GsNap aligns upto 94.14 %, Rum aligns upto 88.11 %, and Map-splice aligns upto 97.28 %. The star aligns upto 92.16%. Its accuracy scope is 88.0% to 97.0% [5].

1.2 TopHat Alignment Algorithm

It discovers a splice-junction without a reference-annotation. Initially, Potential Exons were identified by aligning reads (sequences) to reference genomes. A database of splicing junctions were developed using the information of the initial mapping (aligning). To verify these junctions we have to align (map) the reads across these splicing junctions.

Presently, shorter reads sequencing tools is producing sequences (reads) of 100 bp or longer [8], the exons are shorter than 100 bp so in the initial mapping it will be missed. This problem is resolved by tophat such as to it divides the input reads into a small segments so that individually it could be mapped. In last, the end to end alignment is produced by putting back together all alignment segments.

With the uses of two sources of info, it creates a possible splice-junction database. The initial and cohesive source of information for a splice-junction is that, when at a specific distance two segments from same reads maps on same genomic-sequence. Or the internal segment of mapping again fails to recommend that following reads spanning multiple-exons. By this method the following introns "AG-GC", "GC-AG", "AT-AC", and "GT-AT" is discovered as ab-initio. The pairing of Coverage-island is the nearest source of information, which is piled-up, reads specific regions in the first mapping.

1.3 Challenge for TopHat

Tophat gets into low-maintenance and low-support stage which were superseded by HISAT-2. HISAT-2 hands-over much more accurately and efficiently the same core functionalities such as spliced-alignment of rna-sequencing reads [6]. It's very important to find the reasons behind due to which it's succeeded by HISAT-2. After finding the issues, it will again succeed the HISAT-2.

2. DATA PROCESSING USING TOPHAT

We identified TopHat as the most time-consuming program. To investigate the reason behind the lengthy processing of TopHat, we executed multiple raw rna-sequencing data-sets through tophat using different number of threads and recorded the processing-time of every dataset, where the processing time is consists of Real, User and System time.

Real time refers to actual elapsed time, whereas elapsed time includes the time slices used by other processes and the waiting time for I/O's (input-output) to complete.

The CPU-time elapsed in user mode (outside of kernel) with-in the process is the user time. It's the real cpu-time used in executing a process.

The CPU-time elapsed in the kernel-mode during the execution of the processes is the system time.

User time and system time is the total cpu-time elapsed by the process. To demonstrate the phenomenon, processing time consumed by each thread for two datasets is elaborated in the following subsections

2.1 Data Processing-Time for Data-Set 1

We processed two raw rna-sequencing files (SRR-630-464_1.fast-q and SRR-630-464_2.fast-q) with tophat using 1, 2, 4, 8 and 16 numbers of threads and noted-down the processing-time. Table 1 shows the processing time for dataset-1.

Table 1. Data processing time for dataset-1

No of Threads	User-Time	System-Time	Total-Time (System+user)
1	137.0 minutes	5.12 minutes	142.12 minutes
2	120.33 minutes	6.40 minutes	127.13 minutes
4	124.20 minutes	19.54 minutes	144.14 minutes
8	122.42 minutes	18.31 minutes	141.12 minutes
16	122.22 minutes	21.27 minutes	143.49 minutes

As we increase the number of threads, the total processing time increases rather than decreasing because of the system time. The system time increases which increase the overall alignment time as shown in Table I. Figure 1 shows the data processing time while Figures 2, 3 and 4 show user, system and total (user and system) time for Dataset-1 respectively. As we increase the number of threads, the system time increases which in turn increase the overall alignment or processing time. Figures 1-4 exhibits that the best performance is achieved by using two threads and the worst performance by using 4, 8 and 16 threads.

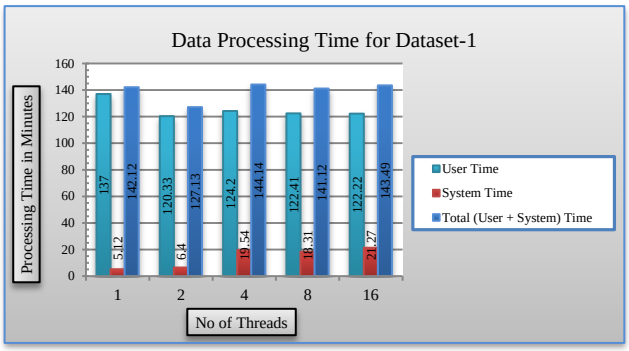

Figure 1. Data processing time for dataset-1.

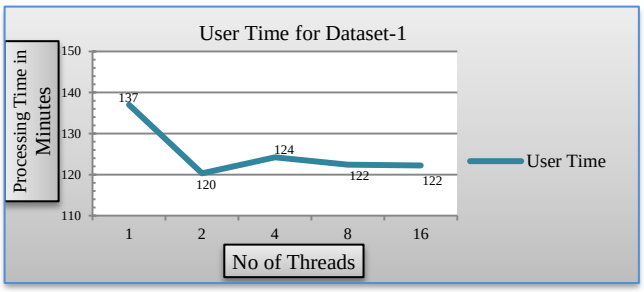

Figure 2. User time for dataset-1.

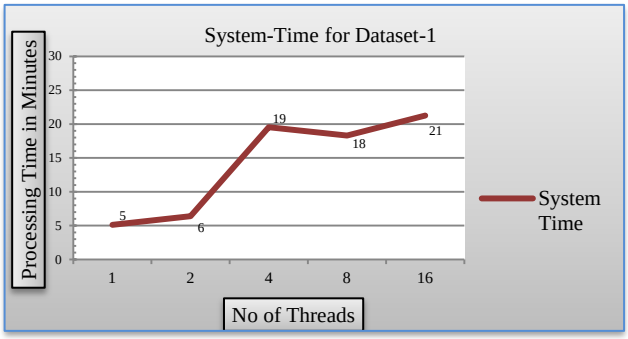

Figure 3. System time for dataset-1.

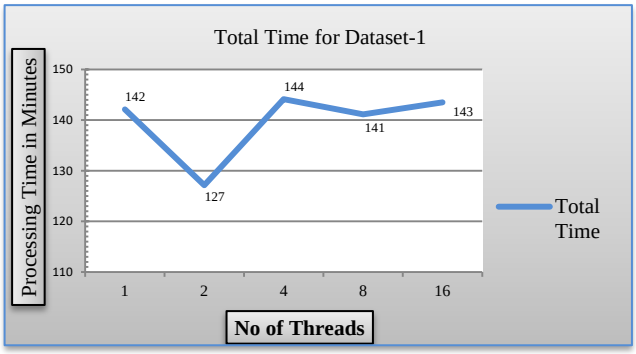

Figure 4. Total (user and system) time for dataset-1.

2.2 Data Processing-Time for Dataset-2

We processed two raw rna-sequencing files (SRR-630-467_1.fast-q and SRR-630-467_2.fast-q) with tophat using 1, 2, 4, 8 and 16 numbers of threads and noted down the processing-time consumed. Table 2 shows the processing time for dataset-2.

Table 2. Data processing time for dataset-2

No of Threads	User-Time	System-Time	Total-Time (System + User)
1	64.32 minutes	2.29 minutes	67.01 minutes
2	57.14 minutes	3.17 minutes	60.31 minutes
4	59.34 minutes	9.23 minutes	68.57 minutes
8	58.24 minutes	8.55 minutes	67.19 minutes
16	63.46 minutes	9.14 minutes	73.00 minutes

Due to the increase in system time, the overall alignment time also increased as shown in Table 2. Figure 5 shows the data processing time while Figures 5, 7, and 8 shows user, system and total (user and system) time for Dataset-2 respectively. As we increase the number of threads, the system time increases and therefore increase the overall alignment or processing time. It shows the best performance for two threads and worst performance for 4, 8 and 16 number of threads.

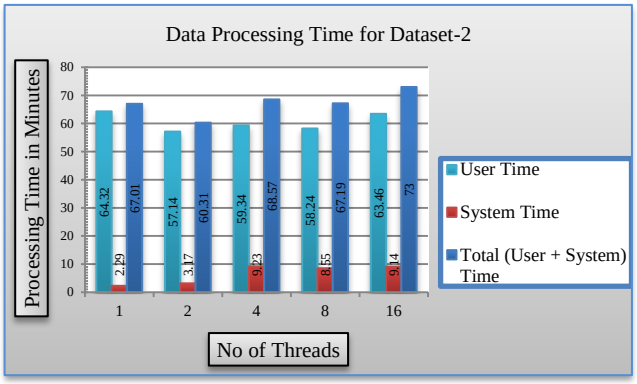

Figure 5. Data processing time for dataset-2.

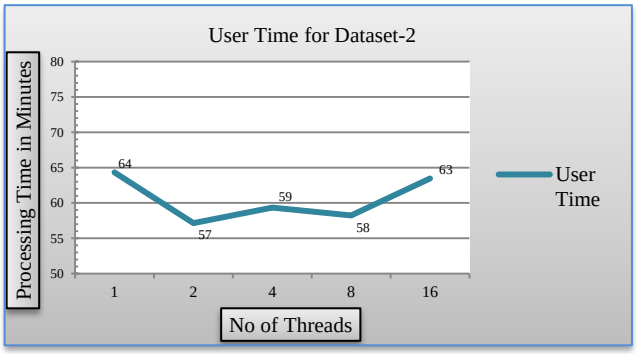

Figure 6. User time for dataset-2.

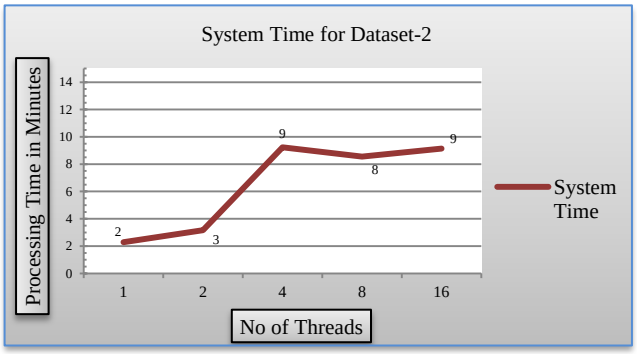

Figure 7. System time for dataset-2.

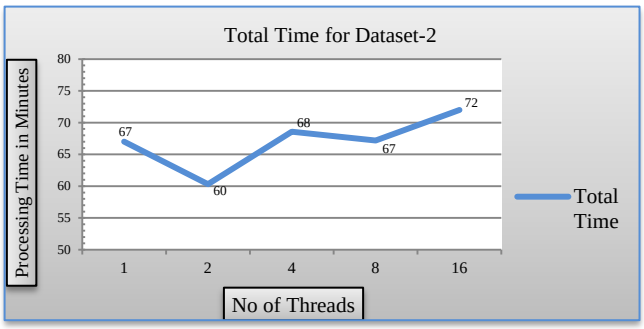

Figure 8. Total (user and system) time for dataset-2.

2.3 Data Processing Time for Dataset-3

We processed a raw rna-sequencing file (SRR-129-1414.fast-q) with tophat using 1, 2, 4, 8 and 16 numbers of threads and noted down the processing-time consumed. Table 3 shows the processing time for dataset-3.

Table 3. Data processing time for dataset-3

Threads Used	User-Time	System-Time	Total Time (system+user)
1	32.13 minutes	1.20 minutes	33.33 minutes
2	30.12 minutes	1.43 minutes	31.55 minutes
4	30.56 minutes	4.40 minutes	35.36 minutes
8	30.32 minutes	4.29 minutes	35.01 minutes
16	31.13 minutes	4.38 minutes	35.51 minutes

Due to the increase in system time, the overall alignment time also increased as shown in Table 3. Figure 9 shows the data processing time while Figures 10, 11, and 12 shows user, system and total (user and system) time for Dataset-3 respectively. As we increase the number of threads, the system time increases and therefore increase the overall alignment or processing time. It also shows the best performance for two threads and worst performance for 4, 8 and 16 number of threads.

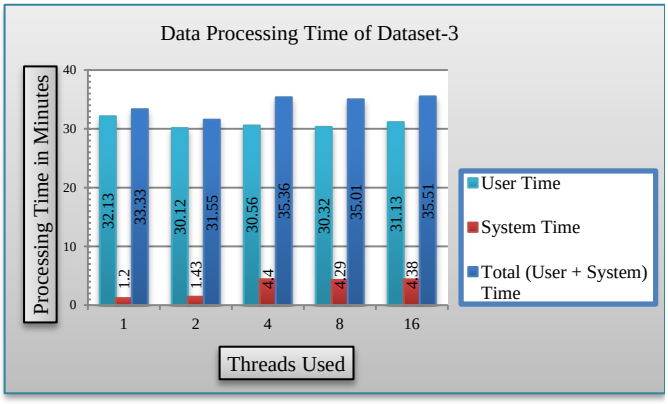

Figure 9. Data processing time for dataset-3.

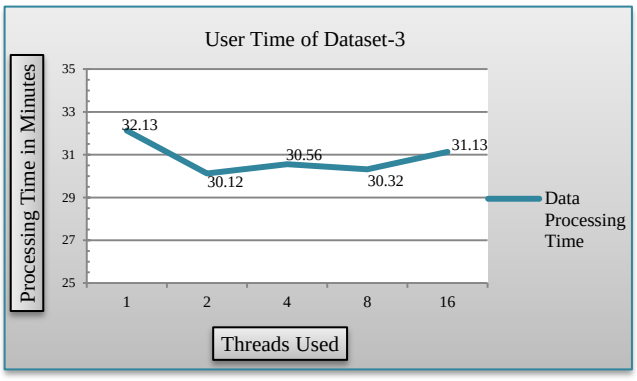

Figure 10. User time for dataset – 3.

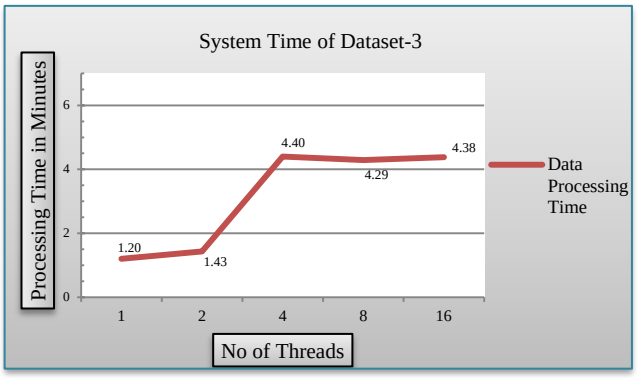

Figure 11. System time for dataset-3.

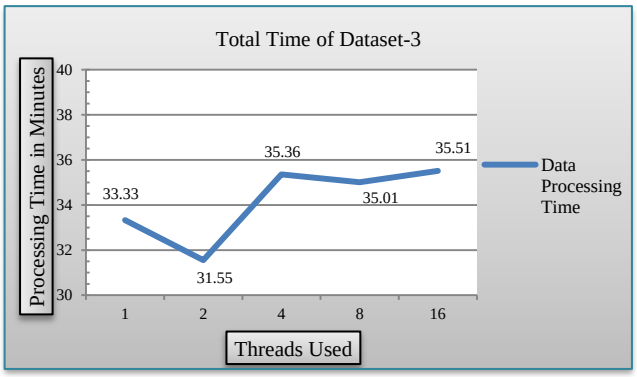

Figure 12. Total (user and system) time for dataset-3.

3. ANALYSIS OF RESULTS

As a matter of fact, the processing-time reduces as the number of threads increases. Contrarily, the results show that the processing time increases with increasing the number of threads.

As we increase the number of threads from one to two, the processing time decreases, while increasing the number of threads from 2 to 16 the processing time increases instead of reducing. The simulation results show the optimization of performance for two numbers of threads and indicate worst performance for more than two numbers of threads i.e. 4, 8 and 16.

After evaluating the processing-time of all data-sets, we concluded that there is a lackness of proper Communication and

Synchronization between the threads. It dissipates huge amount of time by assigning and gathering tasks from the threads.

4. CONCLUSION AND FUTURE WORK

We processed multiple raw ran-sequencing data-sets through tophat using different number of threads and recorded the processing-time of every dataset. Indeed, increasing the number of threads decreases the processing-time. Contrarily, results show that the processing time increases as we increase the number of threads. As we increase the number of threads from one to two, the processing time incredibly decreases. Further, increasing the number of threads from 2 to 16, the processing time is found to be increasing rather than decreasing. For multithreading, parallel processing its not well scaled (coded). It also can't Communicate and Synchronize precociously in between the threads.

The simulations results demonstrate that outstanding performance is achieved by using two numbers of threads. The performance of the system is degraded due to the lack of synchronization and communication between the threads for a higher number of threads.

Future work includes rewriting the Tophat code for efficient synchronization and communication between the threads. This will enable the Tophat to reduce the processing time for an increased number of threads.

5. ACKNOWLEGEMENT

We are thankful to our supervisor Chair Professor Dr. Laiq Hasan for his help, motivation, and enthusiasm. Special thanks to Ikram Ullah for his help and guidance.

6. REFERENCES

[1] B. Sam, and P. S. Tarpey. "What is next Generation Sequencing" Archives of Disease in Childhood, *Education and Practice Edition* 98.6 (2013): 236–238. PMC. Web. 22 June 2017.

[2] W. Zhong, M. Gerstein, and M. Snyder., "RNA-Seq: A Revolutionary Tool for Transcriptomics." *Nature reviews. Genetics* 10.1 (2009): 57–63. PMC. Web. 22 June 2017.

[3] Kukurba K. R., and Stephen B. M., "RNA Sequencing and Analysis." *Cold Spring Harbor protocols* 2015.11 (2015): 951–969. PMC. Web. 23 June 2017.

[4] T. Cole, L. Pachter, and S. L. Salzberg. "TopHat: Discovering Splice Junctions with RNA-Seq." *Bioinformatics* 25.9 (2009): 1105–1111. PMC. Web. 23 June 2017.

[5] K. Daehwan et al. "TopHat2: Accurate Alignment of Transcriptomes in the Presence of Insertions, Deletions and Gene Fusions." *Genome Biology* 14.4 (2013): R36. PMC. Web. 30 June 2017.

[6] K. Daehwan, B. Langmead, and Steven L. S., "HISAT: a fast spliced aligner with low memory requirements." *Nature methods* 12.4 (2015): 357-360.

[7] T. Cole et al. "Differential Gene and Transcript Expression Analysis of RNA-Seq Experiments with TopHat and Cufflinks." *Nature Protocols* 7.3 (2012): 562–578. PMC. Web. 4 July 2017.

[8] Veneman W. J., et al. "Analysis of RNAseq datasets from a comparative infectious disease zebrafish model using GeneTiles bioinformatics." *Immunogenetics* 67.3 (2015): 135-147