# An Evaluation of Alignment Free Approaches for Transcript Abundance Quantification

Meghana Ginjpalli[1], Xinxin Huang[2]
University of California, Los Angeles
[1]mginjpalli@ucla.edu
[2]xinxinh@ucla.edu

*Abstract* - **With RNA-Seq technology, people have the ability to quantify, discover, and profile RNA. Previous methods such as TopHat or STAR use an alignment approach which can be time consuming and computationally expensive. Recently, several alignment free approaches have been proposed to speed up the process of transcript abundance quantification via RNA-Seq. These methods include Sailfish, RNA-Skim, and Kallisto. However, a thorough evaluation of these methods have not yet been performed. Our goal is to compare the running time, memory, and accuracy of these three methods when they are evaluated on Homo Sapiens GRCh37 (hg19) transcriptome. Our results show that in terms of time, memory, and accuracy, Kallisto has the fastest quantification time, uses the least amount of memory, and has similar accuracies to Sailfish.**

*Keywords* – **RNA-Seq, Transcript Abundance Quantification, Sailfish, RNA-Skim, Kallisto**

## I. INTRODUCTION

The onset of RNA sequencing (RNA-Seq) technology brought the ability for people to be able to quantify, discover, and profile RNA. In addition, one can sequence hundreds of millions of short sequences, ranging from 35 bp to 150 bp, in a single run in a short period of time. With RNA-seq being able to produce many short reads, there are also many data analysis software that can be used to analyze these reads. For example, the raw sequence data can be used to generate millions of RNA seq reads. Then these reads can be aligned to the reference genome using Bowtie or Tophat. Other tools can generate various transcripts or isoforms or perform differential expression. However, the alignment approach implemented by some of these methods can be computationally expensive and can take hours or even days.

Recently, several alignment free approaches have been proposed to speed up the process of transcript abundance quantification via RNA-Seq. These methods include Sailfish [1], RNA-Skim [2], and Kallisto [3]. These methods are stated to be faster due to avoiding the alignment process by using k-mers instead, which refers to all possible subsequences of length k from a read obtained through RNA-Seq. However, a thorough evaluation of these methods have not yet been performed. Our goal is to compare the running time, memory, and accuracy of these three methods when they are evaluated on the human genome.

This paper is organized as follows. Section 2 covers the related work and background on RNA-Seq. Section 3 goes over our approach and methods. Section 4 covers how we evaluated Sailfish, RNA-Skim, and Kallisto while Section 5 goes over our results. Lastly, Section 6 discusses our results and goes over potential future work.

## II. RELATED WORK

Before the discovery of alignment free approaches, previous methods would align short reads to the reference genome or transcriptome. This process would be time consuming and would often take hours or days to just to quantify transcript abundance. Some of these previous methods which use alignment methods include TopHat and STAR.

TopHat is a tool developed in 2009 and is used for fast alignment of cDNA sequencing reads by using Bowtie first and then mapping to the reference genome to find splice junctions between exons. It is currently being maintained by Cole Trapnell at the University of Washington. Bowtie is a method commonly used for sequence alignment and was developed in 2009 at the University of Maryland. Given the short reads and the reference genome, Bowtie uses the Burrows-Wheeler transform for the alignment process which also reduces the memory footprint. It can align short reads at the rate of over 25 million 35bp reads per hour. Compared to the suffix array which needs 12 GB for the human genome, the Burrows-Wheeler transform only needs less than 2 GB. Although TopHat is widely used and accepted, newer methods such as STAR resulted in faster speeds.

Spliced Transcripts Alignment to a Reference (STAR) is a tool published in 2012 which uses sequential maximum mappable seed search.

STAR shows better sensitivity and precision than TopHat and performs better than other aligners in terms of mapping speed. STAR can also detect non-canonical splices and chimeric junctions. Although STAR performs 50 times faster than TopHat, it requires a lot more memory such as almost 30GB for the human genome.

While TopHat and STAR are just a few of these alignment based methods, there are many others that each have their own advantages and disadvantages. However, one thing that is common among these alignment based methods is that the time it takes for the alignment step is very slow and computationally intensive.

## III. Approach

For our dataset, we chose to use Homo Sapiens GRCh37 (hg19) release 84 from Ensembl as it is a large and commonly used dataset and will allow for a fair and accurate evaluation of Sailfish, RNA-Skim, and Kallisto. Sailfish is a tool used for rapid alignment free quantification of isoform abundance. RNA-Skim is also a rapid method for RNA-Seq quantification at transcript level and is stated to be faster than Sailfish in terms of quantification time. Kallisto is another tool that is used to quantify abundances of transcripts from RNA-Seq data using pseudoalignment. The hg19 dataset from Ensembl provided us with two options: the complete transcriptome or the cDNA, which is only the protein coding transcripts. Initially, we chose to go with the cDNA because the human genome is quite large and for the extent of this comparison, we thought that the cDNA would suffice. The cDNA contains a total of 176,241 transcripts.

Since Sailfish, RNA-Skim, and Kallisto require both the transcriptome and the short reads as input, we decided to use Polyester, which is an R package used to generate 100 bp length RNA-Seq reads. Given the transcriptome data, it can simulate the steps of the RNA-Seq experiment and produce raw RNA-Seq reads. Polyester also requires a fold change matrix to simulate differential expression. Since we were not conducting experiments involving differential expression, we made the fold change matrix to contain only 1's. However, since large transcriptome datasets can require lots of time and computing resources to analyze, Polyester can only simulate smaller datasets which proved to be a challenge when generating reads for the human transcriptome.

Since the human cDNA contains 176,241 transcripts we found that Polyester could only handle 1000 transcripts at a time. Therefore, we let Polyester run in a loop 177 times in order to simulate RNA-Seq reads for the whole transcriptome and we later merged the resulting files. We ran this experiment on a desktop computer in the Scalable Analytics Institute (ScAI) lab at UCLA. We also used an external hard disk (2 TB) as the memory in the lab computers were not enough to contain the RNA-Seq reads of the human cDNA. Polyester has several parameters that can be changed such as whether we want paired or single reads or different error rates. The error rate is the probability that the sequencer records the wrong nucleotide at any given base. For this experiment, we chose to create four different datasets using Polyester. The datasets are as follows:

1. Single end reads with 0% error rate and 20X coverage
2. Single end reads with 1% error rate and 20X coverage
3. Paired end reads with 0% error rate and 20X coverage
4. Paired end reads with 1% error rate and 20X coverage

Generating the reads for each dataset took about 9 hours while merging the 177 files took around 5 hours. After the reads were generated for each dataset, we evaluated the three alignment free approaches for transcript abundance quantification using Sailfish (0.10.0), RNA-Skim, and Kallisto (0.42.5). In terms of choosing the k-mer size, we decided to go with a k-mer length of 31 since both Sailfish and Kallisto have that value has their default size. For these reasons and in order to make a fair comparison, we decided to test RNA-Skim using a k-mer length of 31 as well.

However, running RNA-Skim with the k-mer value of 31 resulted in a much lower accuracy rate compared to Sailfish and Kallisto. We believe the reason for this is that RNA-Skim was not able to estimate all the counts for all the transcripts in the cDNA since some of the transcript lengths were very short. In RNA-Skim, they build a simple method to select sig-mers after clustering and indexing. They evenly chose the sig-mers based on the sig-mer locations such that any two sig-mers are at least 50 base pairs away from each other in the given transcript. In cDNA data we used, some

of the transcripts are very small, which may be only hundreds of base pair or even dozens of base pair long. Thus, for these kind of transcripts, the selection approach may not work efficiently. That's why we cannot use the default k-mer length for RNA-Skim on cDNA data. Even though its quantification time was faster than Sailfish's quantification time, we realize this is due to the fact that RNA-Skim was only estimating counts for around 30,000 transcripts instead of all 176,000 transcripts. We also found that running RNA-Skim with a k-mer length of 31 has been shown to result in lower accuracies. In fact, RNA-Skim was stated to perform better when using its default k-mer value which is 60. Also we noticed that some of the transcripts in the cDNA contained a lot of N's which means that the nucleotide is not known and could be either A,T,G, or C. While Sailfish and Kallisto were able to appropriately handle these N's without influencing their accuracy, RNA-Skim was not able to handle the N's. These initial results are shown in Section 5. Furthermore, we realized that simulating RNA-Seq reads on the whole cDNA is not accurate in a real world situation as not all transcripts will ever be expressed at the same time. Therefore, we decided to change our approach and try a different method as described next.

In our next approach, we decided to use only the protein coding transcripts which can be generated by RNA-Skim because cDNA tends to also contain short transcripts which do not code for proteins. The total number of protein coding transcripts are 143,838. Then, we randomly sampled 5% of the protein coding transcripts to simulate expression of these specific transcripts. We also took 5 different replicates containing this 5% sample to get variations among the protein coding transcripts. Each replicate contained 7,191 transcripts. For each replicate, we used Polyester at 20X coverage to generate 100 bp length reads for each of the 4 datasets mentioned above. Generating the reads for each replicate took about 2 minutes and each replicate had around 2.5 million reads. We also built a filter to make sure all of the reads are 100 bp or longer. After generating the reads for each of the 4 datasets, we were finally able to run Sailfish, RNA-Skim, and Kallisto using the default k-mer values for each method. This means both Sailfish and Kallisto had a k-mer value of 31 while RNA-Skim had a k-mer value of 60.

## IV. EVALUATION

In order to evaluate Sailfish, RNA-Skim, and Kallisto, we measured the time, memory, and accuracy to run these three methods on each of the four datasets we generated. We used the command, "/usr/bin/time -v [command to run program]", to determine the time of execution and memory usage. The total CPU time was determined by summing the user time and sys time. For memory usage, we used the Max RSS. Resident Set Size (RSS) is the amount of space of physical memory (RAM) held by a process so the Max RSS is the peak amount of memory a process has had up to that point. For accuracy, we used relative difference to compare the ground truth against the method's estimated count.

Relative difference =

$$\frac{|\alpha - \hat{\alpha}|}{\frac{1}{2}(\alpha + \hat{\alpha})}$$

For relative difference, the closer the value is to 0, the higher the accuracy. The ground truth was determined by the reads generated by Polyester. For each transcript, we created a script to first determine the length of each transcript and then figure out the number of reads coming from each transcript. Polyester determines the reads per transcript as a function of the length of each transcript. The equation is as follows:

Reads per transcript = transcript length / read length * 20

The read length is 100bp in this case and the 20 refers to 20X coverage. The estimated counts for each transcript are provided in an output file from each method. We created a script to extract the ground truth and estimated counts to easily compute the relative difference and determine the accuracy.

## V. RESULTS

The following are the results we have after applying Sailfish, RNA-Skim, and Kallisto on our data sets. Each section has two tables. The first table shows the indexing and quantification time, in terms of seconds. Both the indexing and quantification steps are run on 1 thread only. The second table displays the time, memory, and accuracy of running these three methods on each

of these datasets. For each column, we also calculated the average and standard deviation.

Below are our initial results after running Sailfish, RNA-Skim, and Kallisto on the whole cDNA transcriptome and using 31 as k-mer value.

**Table 1 Indexing and quantification time for Sailfish, RNA-Skim, and Kallisto on whole cDNA transcriptome after generating single end reads with 0% error rate and 20X coverage**

| | Sailfish Time | | RNA-Skim Time | | Kallisto Time | |
|---|---|---|---|---|---|---|
| Samples | Indexing | Quantification | Indexing | Quantification | Indexing | Quantification |
| 1 | 127.95 | 481.45 | 13,830 | 303.01 | 268.5 | 168.92 |
| 2 | 127.95 | 460.6 | 13,830 | 279.93 | 268.5 | 171.25 |
| 3 | 127.95 | 461.27 | 13,830 | 327.05 | 268.5 | 168.87 |
| 4 | 127.95 | 477.33 | 13,830 | 276.43 | 268.5 | 170.87 |
| 5 | 127.95 | 466.34 | 13,830 | 314.73 | 268.5 | 173.18 |
| 6 | 127.95 | 471.44 | 13,830 | 343.04 | 268.5 | 169.78 |
| 7 | 127.95 | 465.11 | 13,830 | 327.39 | 268.5 | 169.84 |
| 8 | 127.95 | 470.36 | 13,830 | 291.42 | 268.5 | 167.17 |
| 9 | 127.95 | 464.55 | 13,830 | 337.38 | 268.5 | 166.55 |
| 10 | 127.95 | 491.29 | 13,830 | 279.69 | 268.5 | 168.49 |
| 11 | 127.95 | 464.95 | 13,830 | 305.51 | 268.5 | 168.8 |
| 12 | 127.95 | 471.61 | 13,830 | 315.39 | 268.5 | 171.94 |
| 13 | 127.95 | 482.03 | 13,830 | 306.31 | 268.5 | 169.4 |
| 14 | 127.95 | 470.79 | 13,830 | 320.19 | 268.5 | 168.91 |
| 15 | 127.95 | 465.29 | 13,830 | 323.29 | 268.5 | 169.98 |
| 16 | 127.95 | 468.08 | 13,830 | 281.79 | 268.5 | 171.71 |
| 17 | 127.95 | 474.13 | 13,830 | 292.34 | 268.5 | 171.84 |
| 18 | 127.95 | 483.05 | 13,830 | 300.37 | 268.5 | 168.12 |
| 19 | 127.95 | 492.39 | 13,830 | 312.64 | 268.5 | 170.58 |
| 20 | 127.95 | 496.33 | 13,830 | 301.91 | 268.5 | 173.78 |
| Average | 127.95 | 473.9195 | 13,830 | 306.9905 | 268.5 | 169.999 |
| Standard Deviation | 0 | 10.6 | 0 | 19.5 | 0 | 1.9 |

| | | Sailfish | | | RNA-Skim | | | Kallisto | |
|---|---|---|---|---|---|---|---|---|---|
| Samples | Time | Memory | Relative Difference | Time | Memory | Relative Difference | Time | Memory | Relative Difference |
| 1 | 609.4 | 34 GB | 0.3098 | 14,133 | 52 GB | 0.5291 | 437.42 | 20 GB | 0.254 |
| 2 | 588.55 | 34 GB | 0.3082 | 14,110 | 52 GB | 0.5304 | 439.75 | 20 GB | 0.2546 |
| 3 | 589.22 | 34 GB | 0.3065 | 14,157 | 52 GB | 0.5283 | 437.37 | 20 GB | 0.2538 |
| 4 | 605.28 | 34 GB | 0.3088 | 14,106 | 52 GB | 0.5301 | 439.37 | 20 GB | 0.2549 |
| 5 | 594.29 | 34 GB | 0.3087 | 14,145 | 52 GB | 0.5299 | 441.68 | 20 GB | 0.2556 |
| 6 | 599.39 | 34 GB | 0.3087 | 14,173 | 52 GB | 0.5293 | 438.28 | 20 GB | 0.2556 |
| 7 | 593.06 | 34 GB | 0.3081 | 14,157 | 52 GB | 0.5287 | 438.34 | 20 GB | 0.2554 |
| 8 | 598.31 | 34 GB | 0.3089 | 14,121 | 52 GB | 0.5315 | 435.67 | 20 GB | 0.2545 |
| 9 | 592.5 | 34 GB | 0.3075 | 14,167 | 52 GB | 0.5279 | 435.05 | 20 GB | 0.2546 |
| 10 | 619.24 | 34 GB | 0.3092 | 14,110 | 52 GB | 0.5306 | 436.99 | 20 GB | 0.2553 |
| 11 | 592.9 | 34 GB | 0.3068 | 14,136 | 52 GB | 0.5298 | 437.3 | 20 GB | 0.2541 |
| 12 | 599.56 | 34 GB | 0.3083 | 14,145 | 52 GB | 0.5288 | 440.44 | 20 GB | 0.2548 |
| 13 | 609.98 | 34 GB | 0.308 | 14,136 | 52 GB | 0.5251 | 437.9 | 20 GB | 0.2552 |
| 14 | 598.74 | 34 GB | 0.3086 | 14,150 | 52 GB | 0.5285 | 437.41 | 20 GB | 0.2549 |
| 15 | 593.24 | 34 GB | 0.3077 | 14,153 | 52 GB | 0.5303 | 438.48 | 20 GB | 0.2547 |
| 16 | 596.03 | 34 GB | 0.3084 | 14,112 | 52 GB | 0.5277 | 440.21 | 20 GB | 0.2553 |
| 17 | 602.08 | 34 GB | 0.3086 | 14,122 | 52 GB | 0.5296 | 440.34 | 20 GB | 0.2558 |
| 18 | 611 | 34 GB | 0.308 | 14,130 | 52 GB | 0.5263 | 436.62 | 20 GB | 0.254 |
| 19 | 620.34 | 34 GB | 0.3083 | 14,143 | 52 GB | 0.5272 | 439.08 | 20 GB | 0.255 |
| 20 | 624.28 | 34 GB | 0.3093 | 14,132 | 52 GB | 0.5295 | 442.28 | 20 GB | 0.2552 |
| Average | 601.87 | 34 GB | 0.30832 | 14,137 | 52 GB | 0.52893 | 438.499 | 20 GB | 0.254865 |
| Standard Deviation | 10.6 | 0 | 0.0008 | 19.5 | 0 | 0.0015 | 1.9 | 0 | 0.0006 |

**Table 2 Time, memory, and accuracy of running Sailfish, RNA-Skim, and Kallisto on whole cDNA transcriptome generating single end reads with 0% error rate and 20X coverage**

From Table 1, we can see that in terms of indexing, Sailfish has the fastest time at 127.95 seconds, followed by Kallisto at 268.5 seconds, and then RNA-Skim at 13,830 seconds. In terms of quantification, Kallisto is the fastest at 169 seconds while Sailfish takes the longest at 473 seconds.

From Table 2, we can observe that in terms of time, Kallisto is the fastest while RNA-Skim is the slowest due to its slow indexing time. In terms of memory, Kallisto uses the least amount of memory at 20 GB while RNA-Skim uses the most amount of memory at 52 GB. Finally, in terms of relative difference, Kallisto performed the best at 0.25 while RNA-Skim performed the least at 0.52. As we mentioned in Section 3, we believe RNA-Skim had a lower accuracy due to setting the k-mer value to 31 and using the cDNA instead of the protein coding DNA. The rest of the results shown below are run on a 5% random sample of the protein coding DNA and using the default k-mer values provided by each method.

1. Single end reads with 0% error rate and 20X coverage

From Table 3, we can observe in terms of indexing time, Sailfish is the fastest while RNA-Skim is the slowest. In terms of quantification time, Kallisto is the fastest while RNA-Skim performs the slowest.

Figure 4 shows that in terms of overall time, Sailfish is the fastest while RNA-Skim is the slowest. In terms of memory, Kallisto uses the least amount of memory at 20 GB while RNA-Skim uses the most at 47 GB. In terms of accuracy, Sailfish performs a little better than Kallisto.

2. Single end reads with 1% error rate and 20X coverage

From Table 5, we can see that indexing times will not change due to the fact that we only had to run the indexing once for all the datasets. In terms of quantification time, Kallisto is the fastest while RNA-Skim is the slowest.

Table 6 shows that in terms of overall time, Sailfish still performs the fastest among single end reads with different error rates. The amount of memory used will be the same for all datasets since indexing uses the most amount of memory and is only done once. In terms of accuracy, Sailfish and Kallisto are comparable.

**Table 4 Indexing and quantification time for Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA after generating single end reads with 0% error rate and 20X coverage**

| | Sailfish Time | | RNA-Skim Time | | Kallisto Time | |
|---|---|---|---|---|---|---|
| Samples | Indexing | Quantification | Indexing | Quantification | Indexing | Quantification |
| 1 | 110.72 | 33.47 | 7,094 | 148.27 | 230.52 | 23.52 |
| 2 | 110.72 | 33.56 | 7,094 | 163.44 | 230.52 | 23.6 |
| 3 | 110.72 | 34.64 | 7,094 | 139.21 | 230.52 | 24.33 |
| 4 | 110.72 | 31.85 | 7,094 | 133.27 | 230.52 | 24.07 |
| 5 | 110.72 | 32.79 | 7,094 | 191.27 | 230.52 | 23.3 |
| Average | 110.72 | 33.262 | 7,094 | 155.092 | 230.52 | 23.764 |
| Standard Deviation | 0 | 1.03 | 0 | 23.2 | 0 | 0.42 |

**Table 3 Time, memory, and accuracy of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating single end reads with 0% error rate and 20X coverage**

| | | Sailfish | | | RNA-Skim | | | Kallisto | |
|---|---|---|---|---|---|---|---|---|---|
| Samples | Time | Memory | Relative Difference | Time | Memory | Relative Difference | Time | Memory | Relative Difference |
| 1 | 144.19 | 33 GB | 0.0559 | 7,242 | 47 GB | 0.3774 | 254.04 | 20 GB | 0.062 |
| 2 | 144.28 | 33 GB | 0.0561 | 7,258 | 47 GB | 0.3779 | 254.12 | 20 GB | 0.0643 |
| 3 | 145.36 | 33 GB | 0.0544 | 7,233 | 47 GB | 0.3778 | 254.85 | 20 GB | 0.0601 |
| 4 | 142.57 | 33 GB | 0.0564 | 7,227 | 47 GB | 0.3721 | 254.59 | 20 GB | 0.0617 |
| 5 | 143.51 | 33 GB | 0.0519 | 7,285 | 47 GB | 0.3774 | 253.82 | 20 GB | 0.058 |
| Average | 143.982 | 33 GB | 0.05494 | 7,249 | 47 GB | 0.37652 | 254.284 | 20 GB | 0.06122 |
| Standard Deviation | 1.03 | 0 | 0.002 | 23.2 | 0 | 0.002 | 0.42 | 0 | 0.002 |

**Table 5 Indexing and quantification time for Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA after generating single end reads with 1% error rate and 20X coverage**

| | Sailfish Time | | RNA-Skim Time | | Kallisto Time | |
|---|---|---|---|---|---|---|
| Samples | Indexing | Quantification | Indexing | Quantification | Indexing | Quantification |
| 1 | 110.72 | 33.23 | 7,094 | 132.21 | 230.52 | 25.23 |
| 2 | 110.72 | 34.25 | 7,094 | 137.74 | 230.52 | 25.14 |
| 3 | 110.72 | 34.99 | 7,094 | 158.66 | 230.52 | 24.52 |
| 4 | 110.72 | 34.68 | 7,094 | 136.82 | 230.52 | 24.87 |
| 5 | 110.72 | 32.65 | 7,094 | 131.71 | 230.52 | 24.59 |
| Average | 110.72 | 33.96 | 7,094 | 139.428 | 230.52 | 24.87 |
| Standard Deviation | 0 | 0.98 | 0 | 11.08 | 0 | 0.31 |

**Table 6 Time, memory, and accuracy of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating single end reads with 1% error rate and 20X coverage**

| | | Sailfish | | | RNA-Skim | | | Kallisto | |
|---|---|---|---|---|---|---|---|---|---|
| Samples | Time | Memory | Relative Difference | Time | Memory | Relative Difference | Time | Memory | Relative Difference |
| 1 | 143.95 | 33 GB | 0.0561 | 7,226 | 47 GB | 0.3757 | 255.75 | 20 GB | 0.0628 |
| 2 | 144.97 | 33 GB | 0.0573 | 7,232 | 47 GB | 0.379 | 255.66 | 20 GB | 0.0644 |
| 3 | 145.71 | 33 GB | 0.0549 | 7,253 | 47 GB | 0.376 | 255.04 | 20 GB | 0.059 |
| 4 | 145.4 | 33 GB | 0.0564 | 7,231 | 47 GB | 0.3735 | 255.39 | 20 GB | 0.0521 |
| 5 | 143.37 | 33 GB | 0.0542 | 7,226 | 47 GB | 0.3771 | 255.11 | 20 GB | 0.0564 |
| Average | 144.68 | 33 GB | 0.05578 | 7,233 | 47 GB | 0.37626 | 255.39 | 20 GB | 0.05894 |
| Standard Deviation | 0.98 | 0 | 0.001 | 11.08 | 0 | 0.002 | 0.31 | 0 | 0.005 |

3. Paired end reads with 0% error rate and 20X coverage

From Table 7, in terms of quantification time, even though RNA-Skim improved from single end to paired end, Kallisto is still the fastest.

From Table 8, we can observe that Sailfish is still the fastest in terms of overall time. In terms of accuracy, both Sailfish and Kallisto improved significantly from the accuracy calculated from single end reads but both accuracies are still comparable.

**Table 7 Indexing and quantification time of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating single end reads with 0% error rate and 20X coverage**

| | Sailfish Time | | RNA-Skim Time | | Kallisto Time | |
|---|---|---|---|---|---|---|
| Samples | Indexing | Quantification | Indexing | Quantification | Indexing | Quantification |
| 1 | 110.72 | 46.88 | 7,094 | 146.42 | 230.52 | 25.83 |
| 2 | 110.72 | 49.47 | 7,094 | 135.73 | 230.52 | 26.56 |
| 3 | 110.72 | 48.44 | 7,094 | 132.67 | 230.52 | 26.44 |
| 4 | 110.72 | 46.31 | 7,094 | 129.89 | 230.52 | 26.16 |
| 5 | 110.72 | 44.77 | 7,094 | 136.88 | 230.52 | 24.69 |
| Average | 110.72 | 47.174 | 7,094 | 136.318 | 230.52 | 25.936 |
| Standard Deviation | 0 | 1.83 | 0 | 6.27 | 0 | 0.75 |

**Table 8 Table 4 Time, memory, and accuracy of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating paired end reads with 0% error rate and 20X coverage**

| | | Sailfish | | | RNA-Skim | | | Kallisto | |
|---|---|---|---|---|---|---|---|---|---|
| Samples | Time | Memory | Relative Difference | Time | Memory | Relative Difference | Time | Memory | Relative Difference |
| 1 | 157.6 | 33 GB | 0.0173 | 7,240 | 47 GB | 0.3529 | 256.35 | 20 GB | 0.0166 |
| 2 | 160.19 | 33 GB | 0.0161 | 7,230 | 47 GB | 0.3538 | 257.08 | 20 GB | 0.0158 |
| 3 | 159.16 | 33 GB | 0.0171 | 7,227 | 47 GB | 0.3546 | 256.96 | 20 GB | 0.0161 |
| 4 | 157.03 | 33 GB | 0.0192 | 7,224 | 47 GB | 0.3572 | 256.68 | 20 GB | 0.0179 |
| 5 | 155.49 | 33 GB | 0.0169 | 7,231 | 47 GB | 0.3526 | 255.21 | 20 GB | 0.0152 |
| Average | 157.894 | 33 GB | 0.01732 | 7,230 | 47 GB | 0.35422 | 256.456 | 20 GB | 0.01632 |
| Standard Deviation | 1.83 | 0 | 0.001 | 6.27 | 0 | 0.002 | 0.75 | 0 | 0.001 |

4. Paired end reads with 1% error rate and 20X coverage

From Table 9, we can still observe that Kallisto has the fastest quantification time even for paired ends.

From Table 10, we can observe that the accuracies have not changed significantly from different error rates on paired end reads. Both Sailfish and Kallisto have very similar accuracy rates.

**Table 10 Table 4 Indexing and quantification time of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating paired end reads with 1% error rate and 20X coverage**

| | Sailfish Time | | RNA-Skim Time | | Kallisto Time | |
|---|---|---|---|---|---|---|
| Samples | Indexing | Quantification | Indexing | Quantification | Indexing | Quantification |
| 1 | 110.72 | 45.75 | 7,094 | 150.22 | 230.52 | 24.81 |
| 2 | 110.72 | 47.1 | 7,094 | 163.06 | 230.52 | 26 |
| 3 | 110.72 | 47.56 | 7,094 | 140.06 | 230.52 | 26.32 |
| 4 | 110.72 | 47.31 | 7,094 | 133.93 | 230.52 | 26.1 |
| 5 | 110.72 | 46.17 | 7,094 | 149.51 | 230.52 | 24.98 |
| Average | 110.72 | 46.778 | 7,094 | 147.356 | 230.52 | 25.642 |
| Standard Deviation | 0 | 0.77 | 0 | 11.1 | 0 | 0.69 |

**Table 9 Time, memory, and accuracy of running Sailfish, RNA-Skim, and Kallisto on 5% sample of protein coding DNA generating paired end reads with 1% error rate and 20X coverage**

| | | Sailfish | | | RNA-Skim | | | Kallisto | |
|---|---|---|---|---|---|---|---|---|---|
| Samples | Time | Memory | Relative Difference | Time | Memory | Relative Difference | Time | Memory | Relative Difference |
| 1 | 156.47 | 33 GB | 0.0163 | 7,244 | 47 GB | 0.353 | 255.33 | 20 GB | 0.016 |
| 2 | 157.82 | *33 GB* | 0.0178 | 7,257 | 47 GB | 0.3534 | 256.52 | *20 GB* | 0.0174 |
| 3 | 158.28 | 33 GB | 0.0165 | 7,234 | 47 GB | 0.3539 | 256.84 | 20 GB | 0.0152 |
| 4 | 158.03 | 33 GB | 0.0185 | 7,228 | 47 GB | 0.3568 | 256.62 | 20 GB | 0.0175 |
| 5 | 156.89 | 33 GB | 0.0163 | 7,244 | 47 GB | 0.3528 | 255.5 | 20 GB | 0.015 |
| Average | 157.498 | 33 GB | 0.01708 | 7,241 | 47 GB | 0.35398 | 256.162 | 20 GB | 0.01622 |
| Standard Deviation | 0.77 | 0 | 0.001 | 11.1 | 0 | 0.002 | 0.69 | 0 | 0.001 |

## VI. DISCUSSION

The following section discusses an overview and approach of Sailfish, RNA-Skim, and Kallisto. Next, each of three methods are compared with each other to infer why a certain method performs a certain way in regards to time, memory, and accuracy. Finally, we summarize our findings and discuss the future work we would like to do.

Sailfish is an alignment free tool that can infer gene expression from high-throughput RNA-seq data. It was first published in 2013, but for this comparison, we are using the latest version of Sailfish which released in April 2016. The newest version of Sailfish claimed to make substantial improvements and increase the speed and accuracy greatly. Sailfish basically creates a k-mer index of a transcriptome. A transcriptome is a set of all known transcripts or isoforms. The k-mer index is the mapping of all unique k-mers in the transcriptome to the transcripts it belongs to. Like most existing methods, Sailfish has a two-step process: the "alignment" step and the quantification step. In the "alignment" step or k-mer counting step, Sailfish performs lookup and keeps a record for every k-mer in all the reads. In the quantification step, Sailfish uses the Expectation Maximization (EM) algorithm to probabilistically resolve all the multi-mapping k-mers to get expression estimates from the k-mer x transcript "alignment" profile. The multi-mapping k-mers are k-mers that perfectly maps to multiple transcripts. Compared to previous methods which used an alignment step, Sailfish first preprocesses the transcripts by building a k-mer index, counts the k-mers in the reads, shuffles and allocates the k-mers, and then finally computes the abundance. Some benefits of using Sailfish is that adopting k-mers allows parallel k-mer lookup and counting. Because Sailfish uses a minimal perfect hash function, the k-mer lookup is very fast. Also using the two step EM procedure speeds up convergence substantially and collapsing k-mers into equivalence classes reduces memory usage and speeds inference.

RNA-Skim is also known as a rapid method for RNA-Seq quantification at transcript level. It was published in 2014 and its main approach is by using special k-mers or sig-mers which simplifies the EM algorithm. RNA-Skim states that there is no need to use all the k-mers and by doing so, would be redundant. Furthermore, it states that using sig-mers does not result in loss of accuracy. They state that it results in a faster quantification method than Sailfish. RNA-Skim first clusters the transcripts based on sequence similarity which is a k-mer based similarity measure. For each cluster, RNA-Skim finds all the k-mers and uses a subset that are unique to the cluster which are called sig-mers. In fact, using the sig-mers makes the EM step much more simpler since we are running EM separately on sig-mers from each cluster and therefore the number of k-mers that EM deals with is really small. RNA-Skim has certain benefits in that it only uses a small number of k-mers that are the most informative to similar transcripts and unlike Sailfish which uses one big EM algorithm, RNA-Skim uses multiple, smaller EM computations which results in a faster quantification time.

Kallisto is another tool that claims to be near optimal with RNA-Seq quantification. This tool was published in 2016 and claims to be faster than Sailfish. Kallisto states that only using k-mers throws away information that may be present in a read which leads to inaccurate abundance estimates. Therefore, it uses pseudoalignment for rapidly determining the compatibility of reads with targets, without the need for alignment. By pseudoalignment, it is saying from which transcript could a paired read have originated from. Kallisto states that it can quantify 6 million reads per minute. In the first step, reads are pseudoaligned to the reference transcriptome and in the second step, the EM algorithm deconvolutes pseudoalignments to obtain transcript abundances. Kallisto first creates each k-mer in the transcriptome, builds the transcriptome de Bruijn graph (T-DBG), and colors the transcripts. Next, it uses the k-mers in the reads to find which transcript it came from. Each k-mer appears in a set of transcripts and the intersection of the sets in our pseudoalignment. The benefits of using Kallisto is that the pseudoalignment step preserves the key information needed for quantification. Furthermore, Kallisto is known to be the fastest and most accurate among Sailfish and RNA-Skim.

A. Sailfish vs. RNA-Skim

In this paper, we compared the latest version of Sailfish (0.10.0) which came out in April 2016 to RNA-Skim which came out in 2014. In terms of overall time, RNA-Skim is on average 50X slower than Sailfish when using single ends and 46X slower when using paired ends. When RNA-Skim first came out, they claimed that they have a faster quantification time with regards to Sailfish. However, the comparison was to an older version of Sailfish and since then, Sailfish has made some improvements which claim to have improved their speed and accuracy substantially. The reason why RNA-Skim's overall time is large is due to its indexing time. RNA-Skim's indexing step which takes up around 7,100 seconds takes up quite a bit of time compared to the other methods' indexing

time. RNA-Skim first clusters the similar genes based on similarity which took the majority of the time, then it indexes to find all the sig-mers for each cluster, and then selects which sig-mers it wants to use. This three step process takes up quite some time while Sailfish just indexes the k-mers instead of clusters. When observing quantification time, RNA-Skim performs 4X slower than Sailfish when using single ends and 3X slower when using paired ends. The reason for this is because while Sailfish's quantification time increases from single ends to paired ends, RNA-Skim's quantification time stays the same throughout single and paired ends. While RNA-Skim applies the EM algorithm on each cluster, Sailfish replaced their old alignment step of using k-mers instead by doing an improved mapping that is also used by Salmon.

In terms of memory, we took the maximum memory from either the indexing or quantification stage. RNA-Skim needs 47 GB while Sailfish requires 33 GB. This value stays the same from single or paired ends because the max amount of memory is used during the indexing stage for Sailfish while the max amount of memory is coming from the quantification stage for RNA-Skim. RNA-Skim requires much more memory because they store all the sig-mers in a hash table in memory to count and estimate the number of reads coming from each transcript while Sailfish requires more memory for indexing the k-mers.

In terms of accuracy, we noticed that different error rates did not affect the accuracy. For single end reads, RNA-Skim has a relative difference of 0.3 while Sailfish has 0.05. When using paired ends, Sailfish's accuracy actually improves to 0.01 while RNA-Skim's accuracy stays almost the same. Generally, using paired ends improves the ability to identify positions of various reads. This also improves the assembly of repetitive regions and resolves structural rearrangements. This is why the accuracy improves from single to paired ends for Sailfish. However, for RNA-Skim, the accuracy improvement is very minimal so that's why we consider that the accuracy stayed the same.

B. Sailfish vs. Kallisto

In terms of overall time, Kallisto performs about 1.6X slower than Sailfish for both single and paired ends. If we consider the first dataset we use (the larger one), we can find that the overall time of Kallisto is less than it of Sailfish. When we break this time down into indexing and quantification time, Sailfish has a better indexing time while Kallisto has a faster quantification time. While Sailfish does a mapping for the indexing step, Kallisto builds a T-DBG for the transcriptome while takes a long time. However, Sailfish spends more time doing the EM steps for quantification while Kallisto just has to use the T-DBG to compare the k-mers and skip any k-mers that are redundant. In terms of memory, Sailfish uses 33 GB of memory because it performs lookup and keeps a record for every k-mer for the indexing stage while Kallisto uses 20 GB of memory to store the T-DBG graph. In terms of accuracy, both Sailfish and Kallisto have similar accuracies at 0.05 for single end reads. When using paired end reads, both also have similar accuracies at 0.01. We can observe that the accuracies improve from single to paired ends and in general, both accuracies are comparable as the original paper states.

C. RNA-Skim vs. Kallisto

In terms of overall time, RNA-Skim is 28X slower than Kallisto. Even when comparing indexing and quantification time, Kallisto has the faster indexing and quantification time. The reason for this is because Kallisto builds a T-DBG which is faster to do so than RNA-Skim which clusters the transcripts into sig-mers and computes EM on each cluster. Kallisto's times do not differ between single ends or paired ends. In terms of memory, Kallisto uses 20 GB while RNA-Skim uses 47 GB. The reasons for this is already mentioned in the previous comparisons. In terms of accuracy, RNA-Skim has a relative difference of 0.3 while Kallisto has 0.05 for single ends reads. When using paired ends, Kallisto's accuracy actually improves to 0.01 while RNA-Skim's accuracy is very minuscule.

In general, we observed that between single ends and paired ends, changes in error rate did not affect time, memory, or accuracy. As we used two different error rates: 0% and 1%, we assumed that the difference between these two error rates were not significant enough to show differences in accuracy. In terms of memory, Kallisto used the least amount of memory at 20 GB while RNA-Skim used the most amount of memory at 47 GB. These values did not change between single and paired ends since the maximum amount of memory was being used

during the indexing stage. In terms of accuracy, RNA-Skim had the least amount of accuracy at 0.3 and this value did not get affected by single or paired ends. However, both Sailfish and Kallisto had very similar accuracies and actually performed better when using paired ends. Overall, in terms of time, Kallisto had a faster quantification time, in terms of memory, Kallisto used the least amount of memory, and in terms of accuracy, both Kallisto and Sailfish had similar accuracies which improved when using paired end reads.

For future work, we would like to consider different coverage values. For example, for this project, we used 20X coverage which is actually a good number, however we would like to try 10X and 30X as well to see if that affects the time, memory, or accuracy. We would also like to try different k-mer values for each method to find the most optimal one. When we initially tried using RNA-Skim with a k-mer value of 31, the accuracy was quite low. After further online research, we actually found out that RNA-Skim was proven not to perform well with a k-mer value of 31. Due to this reason, we are curious as to whether there are even better k-mer values that would result in more optimal performance.

## ACKNOWLEDGMENT

We would like to thank Professor Wei Wang and the Scalable Analytics Institute (ScAI) at the University of California, Los Angeles for their time and resources.

## CONTRIBUTIONS

Both of us worked on this project and programming equally. In particular, Xinxin focused more on the experiment and Meghana focused more on the report.

## REFERENCES

[1] Rob Patro, Stephen M Mount, Carl Kingsford. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. Nature Biotechnology 32, 462–464 (2014). doi:10.1038/nbt.2862

[2] Zhaojun Zhang, Wei Wang. RNA-Skim: a rapid method for RNA-Seq quantification at transcript level. Bioinformatics (2014) 30 (12): i283-i292. doi: 10.1093/bioinformatics/btu288

[3] Nicolas L. Bray, Harold Pimentel, Páll Melsted and Lior Pachter, Near-optimal probabilistic RNA-Seq quantification, Nature Biotechnology (2016), doi:10.1038/nbt.3519

[4] Alyssa Frazer et al. Polyester: simulating RNA-seq datasets with differential transcript expression (2014).