# Scavenger: A pipeline for recovery of unaligned reads utilising similarity with aligned reads

Andrian Yang[1,2,†], Joshua Y. S. Tang[1,2,†], Michael Troup[2], and Joshua W. K. Ho[1,2]

[1]Victor Chang Cardiac Research Institute, Sydney, NSW, Australia
[2]University of New South Wales, Sydney, NSW, Australia
[†]These authors contributed equally to the paper as first authors

June 13, 2018

**Abstract**

**Motivation:** Read alignment is an important step in RNA-seq analysis as the result of alignment forms the basis for further downstream analysis. However, recent studies have shown that published alignment tools have variable mapping sensitivity and do not necessarily align reads which should have been aligned, a problem we termed as the false-negative non-alignment problem.

**Results:** We have developed Scavenger, a pipeline for recovering unaligned reads using a novel mechanism which utilises information from aligned reads. Scavenger performs recovery of unaligned reads by re-aligning unaligned reads against a putative location derived from aligned reads with sequence similarity against unaligned reads. We show that Scavenger can successfully recover unaligned reads in both simulated and real RNA-seq datasets, including single-cell RNA-seq data. The reads recovered contain more genetic variants compared to previously aligned reads, indicating that divergence between personal and reference genome plays a role in the false-negative non-alignment problem. We also explored the impact of read recovery on downstream analysis, in particular gene expression analysis, and showed that Scavenger is able to both recover genes which were previously non-expressed and also increase gene expression, with lowly expressed genes having the most impact from the addition of recovered reads. We also found that the majority of genes with >1 fold change in expression after recovery are of pseudogenes category, indicating that pseudogenes expression can be substantially affected by the false-negative non-alignment problem.

**Availability:** Scavenger is available via an open source license in https://github.com/VCCRI/Scavenger/
**Contact:** j.ho@victorchang.edu.au

# 1 Introduction

Read alignment is the process of mapping high-throughput sequencing reads against a reference genome or transcriptome to identify the locations from which the reads originate. This step is typically one of the first steps in the analysis of RNA sequencing (RNA-seq) data prior to downstream analysis such as variant calling and gene expression analysis. There have been a number of published tools which have been developed to perform RNA-seq alignment, such as HISAT2 [11], STAR [5] Subread [15], CRAC [18], MapSplice2 [21] and GSNAP [22]. More recently, new alignment-free tools have been developed specifically for gene expression analysis which skips alignment of reads to the reference and instead performs probabilistic alignment. However, these alignment-free tools are only applicable to specific types of analysis and have limitations compared to traditional alignment methods [8]. The correctness of alignment programs are crucial to the accuracy of the downstream analysis. Unfortunately, previous studies have shown that while these tools have low false positive rates, they do not necessarily have low false negative rate [3, 2]. This means that while many of the reads were likely to be correctly aligned, there are still many incorrectly unaligned reads which should have been aligned. The problem of these incorrectly unaligned reads, or false negative non-alignment, affects the accuracy of the alignment produced and can affect the result of downstream analysis, such as variant calling, indel (insertion-deletion) detection and gene fusion detection [2].

There are a number of factors which contribute to the false negative non-alignment problem. One such factor is the type of algorithm utilised in many alignment tools. In order to efficiently perform alignment against a typically large reference genome in an acceptable amount of time, and to account for splicing events inherent in RNA-sequencing data, many alignment tools use heuristic-based matching of seed sequences generated from read sequences. Due to the typically short length of a seed sequence and the existence of repetitive regions within the genome, there may be multiple locations assigned to a given read which results in the alignment tool excluding the alignment due to ambiguity – a problem known as multi-mapping reads. Another factor which affects false negative non-alignment problems is the divergence between the reference genome and the personal genome of the

organism being sequenced. The reference genome is typically constructed from a small number of samples and thus will only represent a limited degree of the organism's diversity. Alignment of reads to the reference genome will thus be imperfect due to natural variation present in an individual organism. While alignment tools do take into account the variability between the reference genome and an individual's genome by allowing for mismatches, insertion and deletion during alignment, they are unable to handle a substantial degree of genetic variation, such as hyper-edited sites, gene fusion and trans-splicing.

Correcting for a false negative non-alignment problem is much more difficult compared to correcting false positive reads. For false positive reads, there are a number of strategies which can be employed to help filter these type of reads, such as by removing lower quality alignments, removing reads with multiple alignment locations and re-aligning reads with a more specific alignment tool. Recovering false negative reads, on the other hand, is not as straightforward as it is not possible to identify their putative alignment region in the genome. One possible strategy for solving the false negative non-alignment problem is to tune the parameters used for alignment in order to maximise the amount of reads aligned, such as by increasing the threshold for multi-mapping reads and/or increasing the number of mismatches allowed. However, this approach is limited as there is no ground truth in real data to help with optimisation, and increasing the number of reads aligned will also result in an increase in the number of false positive reads. Another strategy for solving the false-negative non-alignment problem is by incorporating variation information during alignment, in the form of utilising alternate locus sequences within the reference genome [13] or integration of single nucleotide polymorphism database to the reference [11], to help minimise the effect of divergence of personal genome compared to reference genome. This approach is also limited as they require existing variation information, which may not be available in non-model organisms.

We have recently applied the idea of Metamorphic Testing – a software testing technique designed for situation where there is an absence of an oracle (a method to verify the correctness of any input) – for performing software testing on the STAR sequence aligner [20]. Metamorphic testing involves multiple executions of the program to be tested with differing inputs, constructed based on a set of relationships (Metamorphic relations - MR), and checking that the outputs produced satisfy the relationships [4, **?**]. In our previous study [20], we developed an MR to test the re-alignability of previously aligned reads in the presence of irrelevant 'control' chromosomes constructed from previously unaligned reads. We discovered that a

non-trivial amount of reads that were previously aligned to the reference genome were now aligned to the control chromosomes consisting of reads which were unable to be aligned to the reference. Further investigation indicated that some of the unaligned reads have high similarity to the aligned reads, indicating the possibility of these reads being false negatives.

In this study, we aim to tackle the problem of false-negative non-alignments by taking inspiration from our previous work on metamorphic testing. We have developed Scavenger, a pipeline designed to recover incorrectly unaligned reads by exploiting information from reads which are successfully aligned. We applied the Scavenger pipeline on a number of simulated and actual RNA-seq datasets, including both bulk (normal) and single-cell RNA-seq dataset, and demonstrated the ability of Scavenger in recovering unaligned reads from these datasets. We then analysed the impact of adding these recovered reads on downstream analysis, in particular gene expression analysis, and discovered that lowly expressed genes, in particular genes of the pseudogenes category, are more affected by the false-negative non-alignment problem. We also verified that the divergence between personal genome and reference genome is a contributing factor to the false-negative non-alignment problem and showed that Scavenger is able to recover reads which are unaligned due to higher degree of variability within the reads sequence.

## 2 Methods

Scavenger is a python-based pipeline designed to recover unaligned reads by utilising information from aligned reads. The pipeline takes in sequencing reads in FASTQ format as the input, along with a reference genome sequence in FASTA format and a corresponding index for the alignment tool built using the reference genome. There are 4 main steps in the Scavenger pipeline - source execution of alignment tool, follow-up execution using aligned reads as input and unaligned reads as index, consensus filtering of follow-up execution result to obtain putative alignment location, and re-alignment of unaligned reads to reference genome (Figure 1). The unaligned reads which are able to be successfully re-aligned back to the genome are then re-written back to the alignment result from the source execution.

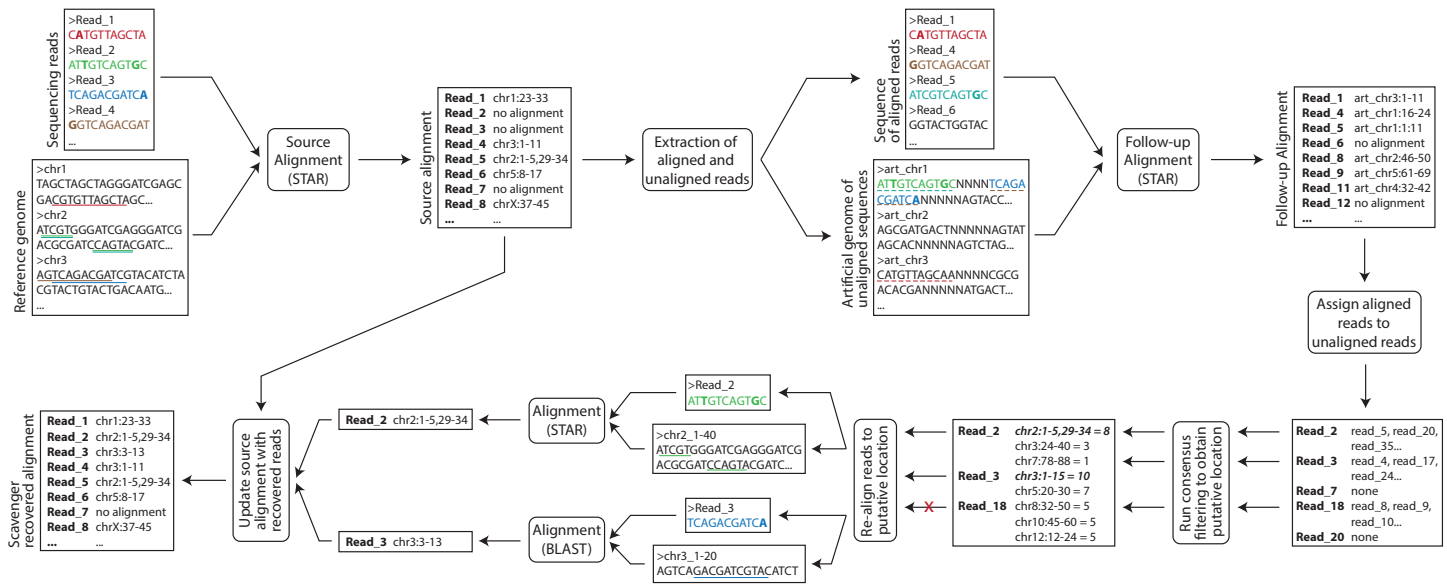


Figure 1: Overview of the Scavenger pipeline. Scavenger first align sequencing reads against the reference genome using STAR alignment tool in source execution step. Scavenger then extracts both the aligned and unaligned reads from the source alignment result and creates a sequencing reads file based on the aligned reads and an artificial genome file containing chromosomes built using sequences of unaligned reads. The sequence of aligned reads are then aligned to the artificial genomes using the same alignment tool from source execution (STAR) in the follow-up execution steps to find aligned reads which have similar sequence to unaligned reads. Next, consensus filtering is performed to select putative sites for re-aligment based on where the majority of aligned reads originates from in the reference genome. Finally, re-alignment is performed for unaligned reads which pass consensus filtering and the source alignment result is updated based on result of re-alignment.

## 2.1 Source execution

The first step of the Scavenger pipeline is the source execution where sequencing reads are aligned to the reference genome using a sequence alignment program. The alignment program used must satisfy the three properties which are required to validate the metamorphic relation underlying the read recovery pipeline - deterministic alignment, realignability of mapped reads, and non-realignability of unmapped reads. Currently, STAR is utilised for aligning RNA sequencing reads in the Scavenger pipeline as it has been previously evaluated as being a reliable general-purpose RNA-seq aligner, with good default performance, [3] as well as satisfying the three properties above [20]. The source execution step can be skipped if the user has previously performed alignment of sequencing reads by passing in the alignment file produced in either SAM or BAM format as input to the Scavenger pipeline.

## 2.2 Follow-up execution

In the follow-up execution step, both aligned and unaligned reads are first extracted from the alignment file produced during source execution. For reads which have been successfully aligned, a sequencing reads file (in FASTQ format) is created using the reads' sequence and qualities retrieved from the alignment records. In the case of reads which did not align to the reference genome, the reads are first grouped based on their sequence in order to minimise computational complexity and to reduce the potential location for alignment. The unique unaligned sequences are then extended with spacer sequences (sequence of N nucleotides) in order to form sequence bins of equal length and to ensure that aligned reads do not align between two unaligned sequences. These sequence bins are concatenated into artificial chromosomes and stored into a new temporary genome file. Depending on the alignment program utilised, a new index will then need to be created based on the temporary genome containing the artificial chromosomes prior to alignment. Finally, sequencing reads of previously aligned reads are aligned to the temporary genome containing unaligned reads sequence using the alignment tool used in source execution. In the current Scavenger pipeline, STAR is again utilised in the follow-up execution with a number of extra parameters in order to disable spliced alignment to ensure that input reads only align to one unaligned read sequence and to remove the restriction of the number of locations (i.e. unaligned read sequence) that the input reads can align to in the temporary genome.

## 2.3 Consensus Filtering

The next step of the Scavenger pipeline is consensus filtering. Reads which align during the follow-up execution step are extracted from the alignment file produced from the previous step to obtain information regarding similarity between reads aligned during source execution and reads which did not align during source execution. Each unaligned sequence may have alignments to multiple aligned reads from source execution. As these aligned reads may be aligned to different regions in the reference genome, consensus filtering is performed to select putative sites for re-alignment. For each unaligned sequence, intervals are created based on the reference genome location of previously aligned reads that align to the unaligned sequence. Overlapping intervals are then merged to form longer intervals to both reduce the number of putative sites and to increase the support for the interval to be selected as putative site. An interval is considered as being a putative site if the level of support for the interval (i.e. the number of aligned reads that falls within the interval) is greater than the consensus threshold, which is set to 50% of the number of previously aligned reads that align to the unaligned sequence by default.

## 2.4 Re-alignment

The final step is the re-alignment step where unaligned sequences which pass the filtering steps are re-aligned to the reference genome using the putative location obtained from reads aligned during source execution as a guide. For each unaligned sequence, the reference genome sequence around the putative location (extended 100 base pair at both start and end of the putative location) is extracted and stored as the new genome for aligning the unaligned sequence. Alignment of the unaligned sequence is then performed against the new genome using either BLAST [1] or STAR depending on if the putative location of the unaligned sequence originate from unspliced alignment or spliced alignment during source execution, respectively. BLAST is utilised for unspliced alignment due to its high sensitivity, though a strict parameter of 64% overlap and 85% query identity is also utlised to reduce false positive recovery of sequence. Unaligned sequences which are successfully aligned back to the reference genome are then added back to the alignment file of the source execution by modifying the alignment records of previously unaligned reads whose sequence matches the recovered unaligned sequence.

## 2.5 Parallelising Scavenger

Both the consensus and re-alignment steps of the Scavenger pipeline are computationally expensive due to the potentially large number of unaligned reads to be processed. However, the processing of the inputs are independent to each other thus allowing for parallelisation of processing unaligned reads in order to reduce the overall runtime of the pipeline. Scavenger takes advantage of Python's built-in multiprocessing library in order to parallelise the consensus and re-alignment steps across the available CPU cores of the machine.

To enhance the scalability of Scavenger, a framework has been provided to enable parallel computation of a read recovery session on cloud computing resources. Cloud computing enables convenient, on-demand network access to a shared pool of configurable computing resources [17]. Central to the model of cloud computing is the virtualisation of computing resources to enable sharing of pooled resources. These resources can be commissioned and decommissioned as the user requires. Scavenger has a framework that employs the resources offered by the could provider Amazon Web Services (AWS). The cloud provider enables the user, using their own account credentials, to create a number of computing "instances", which are the virtual machines upon which the user can perform their computation workload. In the case of AWS, such resources are termed "EC2 instances". An instance typically can be provisioned within minutes of the user request, and the user is charged by the hour. Some cloud providers, such as AWS, offer reduced price "spot" instances at a greatly reduced price, such that the user places a "bid" for a spot instance on the proviso that the instance will be terminated should the current market price for the instance exceed the initial bid price. To minimise the cost for users, Scavenger utilises AWS spot instances.

The cloud computing feature of Scavenger, after initial configuration on the user's controlling computing resource, uses the AWS EC2 cloud instances to perform the various steps of read recovery, and also uses AWS cloud storage (S3) to store test data and results. The Scavenger cloud processing feature co-ordinates all interactions with the cloud resources, with logging information stored both locally and on the cloud. The user can elect to have a large job to be spread among a number of cloud instances, with Scavenger creating the instances and distributing the work load evenly amongst the instances. The cloud computing feature of Scavenger is optional, and the user can elect to use their own computing resources if desired.

Table 1: List of datasets used for Scavenger testing and evaluation. The datasets are divided into three sections: 1. Datasets from selected non-reference mouse strain, 2. Normal (bulk) RNA-seq dataset from either human or mouse, and 3. Single-cell RNA-seq dataset from mouse.

| Accession ID | Samples ID | Organism | Tissue/Source |
|---|---|---|---|
| SRP039411 | SRR1182782 - SRR1182783 | Mus Musculus | Liver |
| ERP000614 | ERR032989 - ERR032991; ERR032997 - ERR032998; ERR033006 - ERR033009; ERR033017 - ERR033019 | Mus Musculus | Brain |
| SRP020636 | SRR826292 - SRR826299; SRR826308 - SRR826315; SRR826340 - SRR826347; SRR826356 - SRR826363 | Mus Musculus | Liver |
| SRP068123 | SRR3087147 - SRR3087158; SRR3087171 - SRR3087176 | Mus Musculus | Hippocampus |
| SRP013610 | SRR504764 - SRR504766 | Mus Musculus | Eye |
| SRP076218 | SRR3641982 - SRR3641983; SRR3641990; SRR3642003 - SRR3642005; SRR3642012 - SRR3642014 | Mus Musculus | Heart |
| SRP045630 | SRR1554415 - SRR1554417 | Mus Musculus | Retina |
| SRP016501 | SRR594393 - SRR594401 | Mus Musculus | Brain; Colon; Heart; Kidney; Liver; Lung; Skeletal Muscle; Spleen; Testes |
| SRP075605 | SRR3578721 - SRR3578724 | Homo sapiens | Fibroblasts |
| SRP122535 | SRR6337339 - SRR6337343 | Homo sapiens | ESC |
| SRP013027 | SRR4422503 - SRR4422506; SRR4422535 - SRR4422538; SRR4422626 - SRR4422629 | Mus Musculus | Hindbrain; Limb; Heart |
| SRP045452 | 80 randomly selected samples | Mus Musculus | Hippocampus |

## 2.6 Datasets

Three different types of RNA-seq datasets – simulated, normal (bulk) and single-cell – were utilised to evaluate the Scavenger pipeline. The simulated datasets were obtained from previous study [3] which generated 3 sets of simulated RNA-seq datasets from the hg19 reference genome using BEERS simulator [6] with varying parameters to emulate different level of dataset complexity. As the simulated datasets were formatted in FASTA format, high quality scores were added to each of the simulated reads to produce corresponding FASTQ files. These files were then inputted to Scavenger for both source alignment and read recovery with either STAR v2.5.3a or Subread v1.6.0 as the alignment tool. Indexes were created for each alignment tool using the GRCh37.p13 reference genome and corresponding annotation file (version 19) obtained from GENCODE [7]. For STAR specifically, the annotation file was utilised in index creation to help increase the accuracy for alignment across splice junction. In the evaluation of alignment result for simulated datasets, a more stringent criteria is used where reads are only considered as being correctly aligned if 90% of the base of the reads are mapped to the correct position.

The normal and single-cell RNA-seq datasets were obtained from publicly available human and mouse datasets which were deposited to the NCBI Sequence Read Archive [14] (Table 1). Pre-processing of the datasets was performed using Trimmomatic v0.36 to remove low quality sequence and short reads. The pre-processed datasets were then analysed by Scavenger using STAR v2.5.3a as the alignment tool in source execution and for re-alignment of spliced read, together with BLAST v2.6.0 for re-alignment of unspliced reads. Indexes used for aligning of both human and mouse datasets were generated from GRCh38 and GRCm38 reference genome respectively, which were obtained from GENCODE together with the corresponding annotation files (version 27 for human and version 15 for mouse). As before, annotation was used to augment the index to increase accuracy for alignment. For mouse strain analysis, strain-specific VCF files for non-reference mouse strains containing SNPs derived against the reference C57BL/6J mouse genome were downloaded from the Mouse Genome Project (MGP) [10].

# 3 Results

## 3.1 Recovery of reads unrecoverable by simple alignment

To evaluate the ability of the Scavenger pipeline to recover false-negative non-aligned reads, we first tested Scavenger using previously published human simulated dataset. The varying level of complexity of the simulated datasets represents the degree of divergence between the sequencing reads generated compared to the reference genome, ranging from low polymorphism and error rate (T1), moderate polymorphism and error rate (T2) and high polymorphism and error rate (T3). The results of the source execution of STAR with default parameters are consistent with the previously published result, with >99% of reads being correctly aligned in both T1 and T2 and >90% of reads being correctly aligned in T3 (Table 2). After running the Scavenger pipeline, we were able to correctly recover 22.7% of the previously unaligned reads in T1 and ∼44.4% of the previously unaligned reads in T2 and T3, resulting in >99.5% of the reads being correctly aligned in both T1 and T2 and ∼95% of reads being correctly aligned in T3.

The difference in the number of aligned reads between the three datasets can be explained by the degree of divergence between the sequencing reads and the reference genome; and the limitation of the alignment tool in aligning reads with high degree of polymorphism. The simulated sequencing reads in both T1 and T2 have high homology to the reference genome due to the lower degree of polymorphism and error rate introduced meaning that the majority of these reads will be accurately mapped to the reference genome with very small number of mismatches during alignment. In contrast, the sequencing reads in T3 – with the higher polymorphism and error rate – have a much higher degree of divergence compared to reference genome thus resulting in more mismatches during alignment and therefore causing it to fail to be aligned. The Scavenger pipeline is able to recover more reads in T2 and T3 compared to T1 due to the greater number of aligned reads that contain mutations within the sequence. During follow up execution, Scavenger exploits the fact that these aligned reads will have closer similarity to the unaligned reads, which will also contains mutations, therefore resulting in the alignment of the aligned reads to the unaligned reads to obtain the putative location for the unaligned reads for recovery.

Another method to solve the false-negative non-alignment problem is to adjust the parameters of the alignment tool utilised in order to allow alignment of reads with higher degree of polymorphism. As has been shown previously, alignment of the simulated datasets using STAR with optimised

Table 2: Alignment statistics for simulated datasets before and after recovery of reads with Scavenger using default parameters for STAR.

| Dataset | Source execution | | Scavenger pipeline | | Unaligned reads recovered (%) |
|---|---|---|---|---|---|
| | Aligned reads | Unaligned reads | Aligned reads | Unaligned reads | |
| T1 | 9,955,582 | 44,418 | 9,965,682 | 34,319 | 22.7 |
| T2 | 9,930,827 | 69,173 | 9,959,920 | 40,080 | 42.1 |
| T3 | 9,031,739 | 968,261 | 9,483,478 | 516,522 | 46.7 |

The result shown is an average from 3 samples.

Table 3: Alignment statistics for simulated datasets before and after recovery of reads with Scavenger using optimised parameters for STAR.

| Dataset | Source execution | | Scavenger pipeline | | Unaligned reads recovered (%) |
|---|---|---|---|---|---|
| | Aligned reads | Unaligned reads | Aligned reads | Unaligned reads | |
| T1 | 9,985,062 | 14,938 | 9,986,663 | 13,337 | 10.7 |
| T2 | 9,988,553 | 11,447 | 9,990,503 | 9,497 | 17.0 |
| T3 | 9,907,921 | 92,079 | 9,968,228 | 31,772 | 65.5 |

The result shown is an average from 3 samples.

parameters results in >99% of the reads being aligned, with T1 and T2 reaching nearly 99.9% of reads being aligned (Table 3). The Scavenger pipeline is unable to obtain the high degree of alignment achieved with parameter optimisation due to limitations in Scavenger's approach to recover reads. Since Scavenger utilises information from aligned reads to find putative location of unaligned reads for recovery, it is not possible to recover any unaligned reads from regions which have no read alignment. As such, the reads that the Scavenger pipeline is able to recover are reads from regions which already have alignment. This is unlike parameter optimisation, which allows for alignment with higher threshold of mismatches in any region irrespective of whether there was alignment in the region. This observation can be seen in the high degree of overlap (∼90%) of the reads recovered by the Scavenger pipeline compared to the reads recovered by optimised parameters. However, the Scavenger pipeline is still able to recover the small percentage of reads which are still unaligned with optimised parameters, ranging from 10% of previously unaligned reads in T1 up to 65% of unaligned reads in T3, thus resulting in >99.5% of reads being aligned in all three simulated datasets. Furthermore, there is an overlap between reads recovered by the Scavenger pipeline with original execution and reads recov-

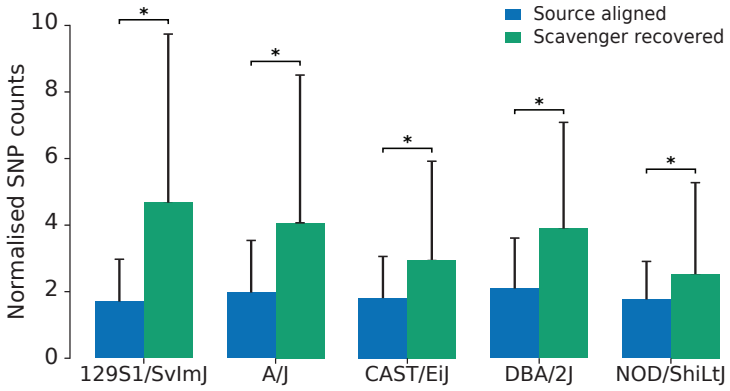


Figure 2: The number of strain-specific SNPs found within reads aligned in source execution and reads recovered by Scavenger. Due to variations in read lengths of the RNA-seq datasets used, normalisation was performed on the number of SNP found based on the length of the aligned reads.

ered by the Scavenger pipeline with optimised execution which suggest that these reads are likely to be unrecoverable with 'simple' alignment alone due to potentially higher degree of divergence of the reads. These results indicate that while parameter optimisation can help with solving the false-negative non-alignment problem, it alone is not enough to fully solve the problem. Moreover, as performing parameter optimisation is not trivial due to lack of ground truth in real datasets, Scavenger can be utilised as an alternative to help recover false-negative non-aligned reads.

### 3.2 Divergence of personal genome results in false-negative non-aligned reads

One factor which may affect the false-negative non-alignment problem is the divergence of sequences between the reference genome and personal genome which results in alignment tools being unable to properly align the reads due to the higher number of mismatches. To evaluate the ability of Scavenger in recovering these false-negative non-aligned reads which arise due to divergence of the personal genome, an experiment was devised where reads from non-reference inbred laboratory mouse strains were aligned to the reference C57BL/6J mouse genome to imitate alignment of reads from the personal genome against the reference genome. Multiple non-reference mouse strains – 129S1/SvImJ, A/J, CAST/EiJ, DBA/2J and NOD/ShiLtJ – were utilised as the genomes of these strains have previously been characterised by the

Mouse Genome Project (MGP), with variations from each strain identified relative to the reference mouse genome. We collected 80 publicly available RNA-seq samples from the selected mouse strains, with each strain having greater than 13 samples from at least 3 different projects with varying characteristics, and performed alignment of these samples against the reference genome using STAR with default parameters. The result of the source alignments shows that there is generally a high degree of mappability of the reads, ranging from 82.2% up to 98.1%. After recovery with Scavenger, we were able to re-align ∼17% of unaligned reads in source execution, corresponding to an increase in the number of aligned reads ranging from 60,000 to 1,000,000 reads (Table 4).

Further analysis was performed to evaluate the hypothesis that reads recovered by Scavenger have a higher degree of polymorphism due to the divergence between the 'personal' non-reference mouse strain genome against the reference genome. We randomly selected 1,000 unspliced reads which are aligned in the source execution and 1,000 unspliced reads recovered by Scavenger from each sample, and then calculated the number of single nucleotide polymorphism (SNP) found within the location of the aligned reads from the list of strain-specific SNPs published by MGP against the reference mouse genome. The majority of the reads which are either successfully aligned or recovered did not contain any of known SNPs. However, the number of reads which contain SNPs is higher in the reads recovered by Scavenger compared to the reads aligned in the source execution for 4 of the 5 strains analysed. Furthermore, the number of SNPs found within recovered reads are also significantly higher (p-value $< 10^{-20}$) in all of the strains analysed indicating that the reads recovered by Scavenger are more polymorphic compared to the reads aligned during source execution (Figure 2). The same analysis was then repeated a further 9 times with the same consistent result being observed in all 10 iterations of the analysis. These results validate the hypothesis that reads recovered by Scavenger have higher degree of polymorphism as a result of divergence between the personal genome and the reference genome and further demonstrate the ability of Scavenger in dealing with the false-negative non-alignment problem.

Table 4: Alignment statistics for all RNA-seq datasets in source alignment with STAR and after recovery of reads with Scavenger.

| Accession ID | Read length | Total reads | Source aligned reads | Source un-aligned reads | Source mappa-bility | Rescue aligned reads | Rescue un-aligned reads | Rescue mappa-bility | Rescued reads | Unaligned reads rescued (%) |
|---|---|---|---|---|---|---|---|---|---|---|
| SRP039411 | 97 bp | 47,077,051 | 44,052,994 | 3,024,056 | 93.6% | 44,587,321 | 2,489,730 | 94.7% | 534,326 | 17.7% |
| ERP000614 | 73 bp | 30,406,321 | 29,529,186 | 877,136 | 97.1% | 29,679,085 | 727,236 | 97.6% | 149,899 | 17.9% |
| SRP020636 | 93 bp | 10,695,056 | 10,023,946 | 671,110 | 93.8% | 10,168,978 | 526,078 | 95.1% | 145,032 | 21.2% |
| SRP068123 | 89 bp | 36,237,495 | 29,132,806 | 7,104,689 | 82.2% | 29,539,843 | 6,697,652 | 83.2% | 407,038 | 6.3% |
| SRP013610 | 54 bp | 21,039,752 | 20,514,308 | 525,444 | 97.5% | 20,620,954 | 418,798 | 98% | 106,646 | 20.1% |
| SRP076218 | 86 bp | 20,183,248 | 19,802,286 | 380,962 | 98.1% | 19,860,969 | 322,279 | 98.4% | 58,683 | 15.9% |
| SRP045630 | 99 bp | 15,931,928 | 15,550,706 | 381,221 | 97.6% | 15,626,281 | 305,647 | 98.1% | 75,574 | 19.8% |
| SRP016501 | 48 bp | 86,537,128 | 83,020,118 | 3,517,011 | 96.2% | 84,056,398 | 2,480,731 | 97.3% | 1,036,280 | 29.6% |
| SRP075605 | 51 bp | 30,308,638 | 29,043,100 | 1,265,537 | 95.8% | 29,261,696 | 1,046,941 | 96.5% | 218,596 | 17.3% |
| SRP122535 | 50 bp | 15,813,104 | 15,368,396 | 444,708 | 97.3% | 15,433,813 | 379,291 | 97.7% | 65,417 | 14.9% |
| SRP013027 | 100 bp | 28,534,297 | 26,134,162 | 2,400,135 | 91.6% | 26,286,896 | 2,247,401 | 92.1% | 152,734 | 6.43% |
| SRP045452 | 51 bp | 2,423,662 | 1,426,200 | 997,463 | 58.9% | 1,446,192 | 977,471 | 59.7% | 19,992 | 2.31% |

The result shown is an average of all samples per accession ID.

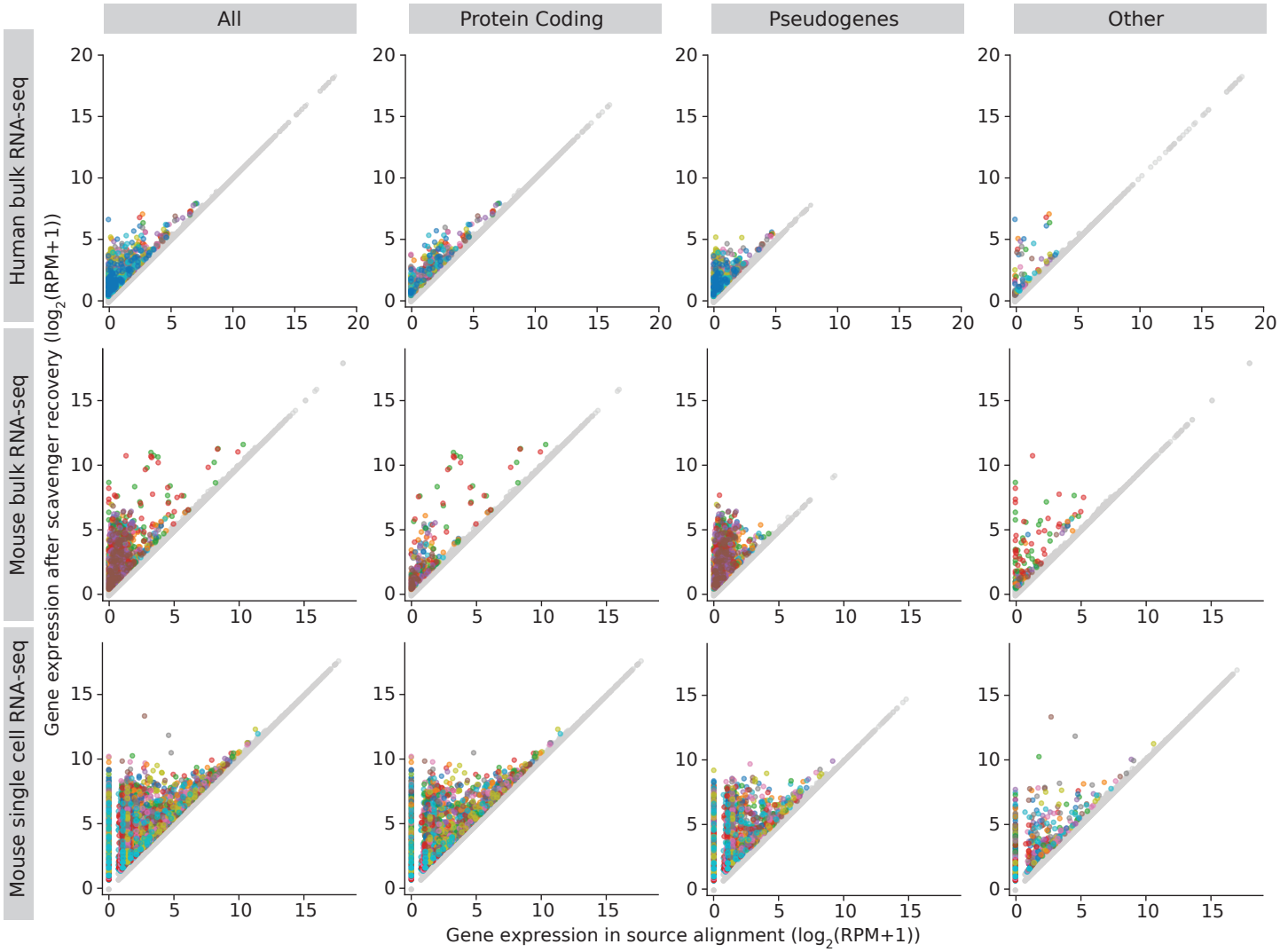


Figure 3: Gene expression in source alignment and after Scavenger recovery for genes whose reads are recovered. Coloured points indicates genes with expression difference of greater than 1 fold change.

### 3.3 Effect of Scavenger recovery pipeline on downstream analysis

While alignment of reads is an important step in RNA-seq analysis, further downstream analysis is required in order to interpret the data into meaningful results. As one of the most common applications of RNA-seq analysis is gene expression analysis, we focused on identifying the effect of adding reads recovered by Scavenger on the expression of genes. The dataset utilised for testing consisted of 22 publicly available RNA-seq samples selected from 3 separate projects of varying characteristics, with 10 samples originating from two human projects and 12 samples originating from a single mouse project. The result of source execution using STAR with default parameters shows a high degree of mappability in all datasets, ranging from ∼96.5% in human datasets and ∼91.6% in the mouse dataset (Table 4). After recovery of reads with Scavenger, we were able to recover ∼12% of unaligned reads on average across the three datasets, corresponding to an increase ranging from 60,000 reads up to 220,000 reads. While the number of reads recovered are quite low relative to the number of previously aligned reads, the addition of tens and hundreds of thousands of reads is still likely to affect the expression of the genes.

Gene quantification of aligned reads is performed using featureCounts [16] to produce read counts per gene, which is then normalised to reads per million (RPM). In source alignment, the number of genes expressed, defined as having non-zero read counts, in the human datasets average to 26,000 genes, while the number of genes expressed in the mouse dataset is 23,750 genes. In Scavenger recovered alignment, we see an increase of ∼154 and ∼94 in the number of expressed genes for human and mouse respectively, indicating the ability of Scavenger to recover genes which are falsely considered as non-expressed in source alignment (Figure 4A). The recovery of reads in previously non-expressed genes is likely due to the extension of putative alignment location, which may introduce regions which have no alignment in source execution. Further investigation into the reads recovered by Scavenger shows that the reads are not distributed evenly across all the expressed genes – only ∼5,200 and ∼9,700 genes receive an increase in read counts in human and mouse datasets, respectively. The majority of genes with increased read counts do not see much change in gene expression, with only ∼16% of genes having more than 1 fold-change difference between source expression and recovered expression. Interestingly, genes which have substantial difference after recovery are generally genes with low expression in source execution ($\log2(\text{RPM}) < 5$) , potentially indicating that

lowly expressed genes may actually have higher true expression than what is reported due to the alignment tool being unable to pick up these reads (Figure 3). This also has implications in further downstream analysis as lowly expressed genes are typically excluded from analysis, when instead it should not have been excluded as their true expression is actually higher.

We then performed further investigations into the genes with more than 1 fold-change difference after recovery to study the types of genes affected by the false-negative non-alignment problem. The majority of genes with recovered expression in the human and mouse dataset are classified as pseudogenes (>60%), with the second most frequent type being protein coding genes (30% and 17% for human and mouse dataset, respectively) (Figure 4B). Moreover, most genes with very low expression in source alignment (log2(RPM+1) $< 5$) are in the pseudogenes category implying that many pseudogenes expression are likely to be under-reported due to reads originating from pseudogenes not being picked up by the alignment tool (Figure 3). Frequency analysis of the recovered genes also shows that 14% and 26% of genes are consistently recovered across at least half of the samples in human and mouse datasets respectively, potentially indicating that these sets of genes are harder to be picked up by the alignment tool due to its sequence being highly polymorphic. The finding that expression of pseudogenes are particularly affected by the false-negative non-alignment problem is significant as recent studies have shown that pseudogenes are incorrectly assumed to be non-functioning and actually have a role in regulating biological processes, particularly in diseases such as cancer [9, 19]. The reason that pseudogenes are more affected by Scavenger recovery is likely due to a number of factors, including the large number of mutations accumulated which results in divergence between pseudogenes sequence and personal genomes; and the typically low expression of pseudogenes and correspondingly, the number of reads from pseudogenes, which therefore are more affected by increase in reads as a result of recovery by Scavenger.

### 3.4 Applying Scavenger recovery on single-cell RNA-seq data

Single cell RNA-sequencing (scRNA-seq) is fast becoming a mainstream method for transcriptomics analysis due its ability to elucidate transcriptional heterogeneity of individual cells. However, there are a number of challenges when dealing with scRNA-seq datasets due to systematically low read counts, as a result of the small amount of transcripts which are captured during library preparation, and high degree of technical noise [12]. Given Scavenger's ability in recovering false-negative non-recovered reads in

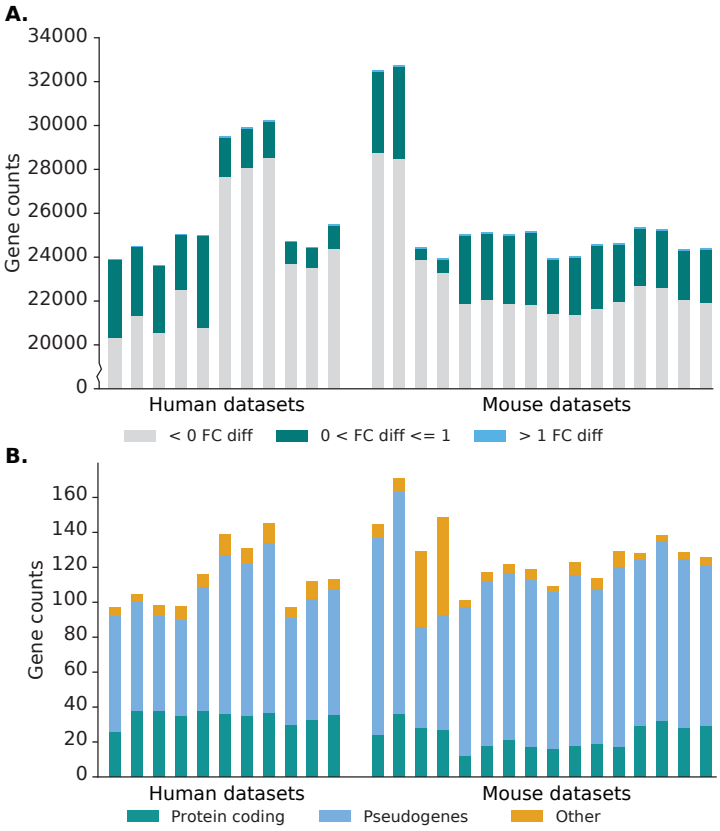


Figure 4: Affect of Scavenger read recovery on gene expression for normal (bulk) RNA-seq dataset. A. The number of genes whose reads are recovered by Scavenger, categorised based on the fold change in normalised expression (RPM) between source alignment and after Scavenger recovery. B. The number of genes with more than 1 fold change in normalised expression categorised based on their gene types.

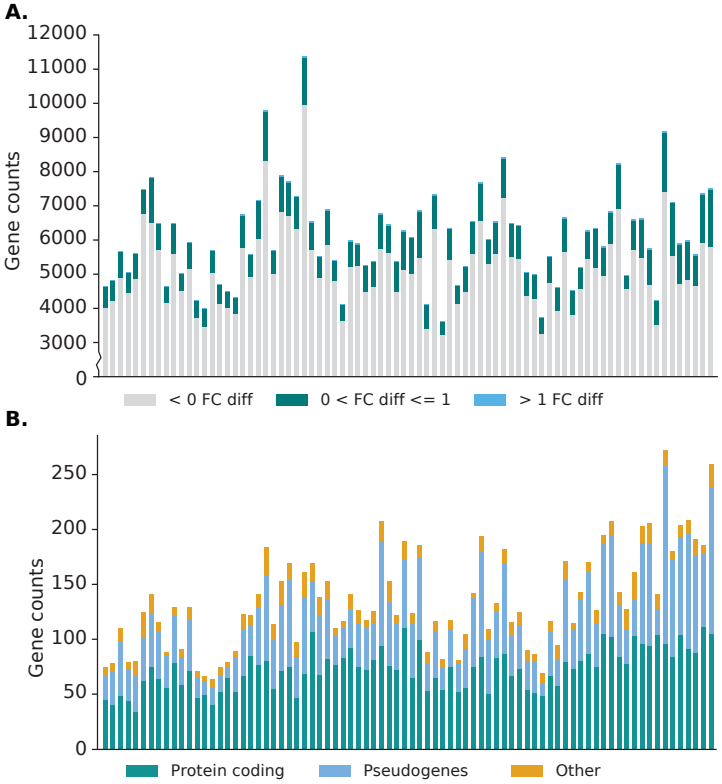


Figure 5: Affect of Scavenger read recovery on gene expression for single-cell RNA-seq dataset. A. The number of genes whose reads are recovered by Scavenger, categorised based on the fold change in normalised expression (RPM) between source alignment and after Scavenger recovery. B. The number of genes with more than 1 fold change in normalised expression categorised based on their gene types.

normal bulk RNA-seq datasets and the effect it has on downstream analysis, we hypothesise that recovery of unaligned reads in scRNA-seq datasets with Scavenger will likely have a greater impact on downstream analysis due to limited amount of reads available, while also helping with reducing technical noise. To test this hypothesis, 80 randomly selected samples were collected from a mouse brain scRNA-seq dataset and which are then aligned with STAR, followed by recovery of reads with Scavenger. The scRNA-seq samples have an average read depth of ∼3.5 million reads, with ∼58.9% of the reads able to be aligned in source execution (Table 4). Scavenger was only able to recover 2% of the unaligned reads, corresponding to an increase of ∼20,000 reads. The low number of reads which are able to be successfully recovered by the Scavenger pipeline is likely due to the low number of aligned in reads in source alignment, which provides less information that Scavenger can utilise during the follow-up execution.

As per the norm for scRNA-seq datasets, the number of genes with non-zero read counts is much lower compared to the number of non-expressed genes in bulk RNA-seq dataset, averaging to 6,000 in source alignment and 6,070 after recovery by Scavenger, corresponding to a 1% increase in the number of non-expressed genes recovered which are likely due to the extension of putative region as previously mentioned. Of these expressed genes, only 25% of the genes (∼1,537) have an increase in read counts, with the majority of these genes having little difference in expression and 9% of the genes (∼134) having a fold change difference greater than 1 (Figure 5A). Unlike in bulk RNA-seq datasets, genes with substantial difference after recovery range from lowly expressed genes up to highly expressed genes, though genes with the greatest difference in expression are still those with low expression in source alignment (Figure 3). Furthermore, a different pattern was also observed in the types of genes which have substantial difference in scRNA-seq datasets, with both protein coding and pseudogenes categories being both equally represented (Figure 5B). One interesting pattern observed in the gene expression of scRNA-seq datasets is the gap in expression between non-expressed genes and genes with low expression, which is still maintained even on genes which become expressed after recovery indicating that the gap in expression is not caused by an alignment issue (Figure 3). Rather, it is likely that this gap is a result of the limitation in current scRNA-seq technology which is unable to efficiently capture very lowly expressed genes due to the small amount of transcripts available.

# 4 Discussion

The false-negative non-alignment problem is a prevalent problem in many of the published RNA-seq alignment tools, resulting in loss of information from incorrectly unaligned reads. To help solve the false-negative non-alignment problem, we have developed Scavenger – a pipeline for recovery of unaligned reads using a novel mechanism based on sequence similarity between unaligned and aligned reads. Scavenger utilises the follow-up execution concept adapted from our previous work on metamorphic testing to find aligned reads from source execution which have similar sequence to the unaligned reads by aligning the aligned reads against unaligned reads. The location of the aligned reads are then used as a guide to re-align the unaligned reads back to the reference genome using either BLAST or the original alignment tool depending on if the putative location originate from unspliced or spliced alignment, respectively, to ensure that splicing information is retained in recovered reads.

We have applied Scavenger on simulated datasets with varying degree of complexity and showed that Scavenger is able to recover unaligned reads across all complexity levels. In particular, Scavenger is able to recover the most amount of reads in datasets with high degree of complexity where read sequence is more divergent compared to the reference genome. We further show that Scavenger is able to recover reads even from alignment with optimised parameters, which produce a higher number of aligned reads compared to after recovery with Scavenger. The lower number of reads recovered by after Scavenger is a result of Scavenger using information from aligned reads to find putative location for unaligned reads, meaning that Scavenger is unable to recover reads from region with no alignment unlike parameter optimisation which does not have the same limitation. Given that the reads recovered by Scavenger have high degree of overlap to reads recovered with parameter optimisation and the non-trivial difficulty of performing parameter optimisation on real dataset, we recommend the use of Scavenger as an alternative to help with recovering incorrectly unaligned reads.

There are a number of possible factors which may contribute to the false-negative non-alignment problem. One such factor is the divergence between the reference genome and the personal genome, leading to higher mismatches during alignment of sequenced reads against the reference genome. In order to validate that divergence of genomic sequence does result in incorrectly unaligned reads, we devised an experiment whereby RNA-seq datasets from non-reference mouse strains were aligned against the reference mouse strain. We then analysed the reads which are aligned in source execution against

those recovered by Scavenger and showed that the reads recovered by Scavenger have significantly higher number of reported strain-specific SNPs. This result both confirms that divergence of sequence between reference genome and personal genome does affect the false-negative non-alignment problems and that Scavenger is able to recover reads which are incorrectly unaligned due to higher degree of sequence divergence.

As alignment of reads is only the first step in RNA-seq data analysis, we also investigated the effect of the false-negative non-alignment problem on downstream analysis, in particular on gene expression analysis. After recovery of reads with Scavenger, we show a 4% increase in the number of genes with non-zero expression, corresponding to the recovery of expression for >95 genes which was considered falsely as being non-expressed in source alignment. We also show that >100 genes have more than 1 fold change in expression compared to source alignment and that these genes are typically genes with low expression. Interestingly, the majority of genes with >1 expression difference belong pseudogenes category, indicating that the expression of pseudogenes are likely to be under-reported due to reads from pseudogenes being incorrectly unaligned by alignment tool. Based on this result, we recommend the use of Scavenger in studies which focuses on pseudogenes expression as pseudogenes expression can be substantially affected by the false-negative non-alignment problem.

Given the ability of Scavenger in recovering gene expression in normal (bulk) RNA-seq dataset, we then investigated the ability of Scavenger in recovering reads from scRNA-seq dataset as scRNA-seq datasets have the characteristics of having low reads counts and high degree of technical noise. Scavenger were able to recover ∼70 genes which were previously non-expressed, corresponding to a 1% increase in expressed genes. Furthermore, the addition of the recovered reads affected the expression of 25% of the expressed genes, with >134 genes having more than 1 fold change in expression. Unlike the bulk RNA-seq dataset, the genes with >1 change in expression range from lowly expressed genes up to highly expressed genes, with the genes belonging to protein coding and pseudogenes category in equal measure.

The current version of Scavenger supports STAR as the alignment tool for source execution and re-alignment of spliced reads. However, user can choose to modify the alignment tool utilised by Scavenger with the alignment tool of their choice. Ideally the tool should satisfy the three properties underlying the read recovery pipeline – deterministic alignment, realignability of mapped reads, and non-realignability of unmapped reads – to ensure that the recovered reads are deterministic. To show the extensibility of

Table 5: Alignment statistics for simulated datasets before and after recovery of reads with Scavenger using default parameters for Subread.

| Dataset | Source execution | | Scavenger pipeline | | Unaligned reads recovered (%) |
|---|---|---|---|---|---|
| | Aligned reads | Unaligned reads | Aligned reads | Unaligned reads | |
| T1 | 9,070,081 | 929,918 | 9,176,184 | 823,816 | 11.4 |
| T2 | 8,777,255 | 1,222,745 | 9,023,136 | 976,864 | 20.1 |
| T3 | 4,771,886 | 5,228,113 | 5,452,695 | 4,547,305 | 13.0 |

The result shown is an average from 3 samples.

Table 6: Alignment statistics for simulated datasets before and after recovery of reads with Scavenger using optimised parameters for Subread.

| Dataset | Source execution | | Scavenger pipeline | | Unaligned reads recovered (%) |
|---|---|---|---|---|---|
| | Aligned reads | Unaligned reads | Aligned reads | Unaligned reads | |
| T1 | 9,237,798 | 772,202 | 9,320,555 | 679,445 | 12.0 |
| T2 | 9,031,594 | 968,406 | 9,170,754 | 829,246 | 14.4 |
| T3 | 8,230,631 | 1,769,369 | 8,292,048 | 1,707,952 | 3.47 |

The result shown is an average from 3 samples.

Scavenger, we have tested Subread, another RNA-seq alignment tool, as a replacement for STAR within the Scavenger pipeline and demonstrated that Scavenger is still able to recover incorrectly unaligned reads (Table 5 and 6). It should be noted that the recovery performance of Subread is different compared to STAR due to the different algorithm employed by Subread for alignment and, potentially, due to Subread violating the deterministic alignment properties.

## Funding

A.Y. is supported by an Australian Postgraduate Award. J.W.K.H is supported by a Career Development Fellowship by the National Health and Medical Research Council (1105271) and a Future Leader Fellowship by the National Heart Foundation of Australia (100848). This work was also supported in part by Amazon Web Services (AWS) Credits for Research

## References


[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, oct 1990.

[2] J. Audoux, M. Salson, C. F. Grosset, S. Beaumeunier, J. M. Holder, T. Commes, and N. Philippe. SimBA: A methodology and tools for evaluating the performance of RNA-Seq bioinformatic pipelines. *BMC Bioinformatics*, 18(1):1–14, 2017.

[3] G. Baruzzo, K. E. Hayer, E. J. Kim, B. Di Camillo, G. A. FitzGerald, and G. R. Grant. Simulation-based comprehensive benchmarking of RNA-seq aligners. *Nature Methods*, 14(2):135–139, dec 2016.

[4] T. Chen, J. W. Ho, H. Liu, and X. Xie. An innovative approach for testing bioinformatics programs using metamorphic testing. *BMC Bioinformatics*, 10(1):24, 2009.

[5] A. Dobin, C. a. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics (Oxford, England)*, 29(1):15–21, 2013.

[6] G. R. Grant, M. H. Farkas, A. D. Pizarro, N. F. Lahens, J. Schug, B. P. Brunk, C. J. Stoeckert, J. B. Hogenesch, and E. A. Pierce. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM). *Bioinformatics*, 27(18):2518–2528, 2011.

[7] J. Harrow, A. Frankish, J. M. Gonzalez, E. Tapanari, M. Diekhans, F. Kokocinski, B. L. Aken, D. Barrell, A. Zadissa, S. Searle, I. Barnes, A. Bignell, V. Boychenko, T. Hunt, M. Kay, G. Mukherjee, J. Rajan, G. Despacio-Reyes, G. Saunders, C. Steward, R. Harte, M. Lin, C. Howald, A. Tanzer, T. Derrien, J. Chrast, N. Walters, S. Balasubramanian, B. Pei, M. Tress, J. M. Rodriguez, I. Ezkurdia, J. van Baren, M. Brent, D. Haussler, M. Kellis, A. Valencia, A. Reymond, M. Gerstein, R. Guigo, and T. J. Hubbard. GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research*, 22(9):1760–1774, sep 2012.

[8] K. E. Hayer, A. Pizarro, N. F. Lahens, J. B. Hogenesch, and G. R. Grant. Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data. *Bioinformatics*, 31(24):3938–3945, 2015.

[9] S. Kalyana-Sundaram, C. Kumar-Sinha, S. Shankar, D. R. Robinson, Y.-M. Wu, X. Cao, I. A. Asangani, V. Kothari, J. R. Prensner, R. J. Lonigro, M. K. Iyer, T. Barrette, A. Shanmugam, S. M. Dhanasekaran, N. Palanisamy, and A. M. Chinnaiyan. Expressed Pseudogenes in the Transcriptional Landscape of Human Cancers. *Cell*, 149(7):1622–1634, jun 2012.

[10] T. M. Keane, L. Goodstadt, P. Danecek, M. A. White, K. Wong, B. Yalcin, A. Heger, A. Agam, G. Slater, M. Goodson, N. A. Furlotte, E. Eskin, C. Nellåker, H. Whitley, J. Cleak, D. Janowitz, P. Hernandez-Pliego, A. Edwards, T. G. Belgard, P. L. Oliver, R. E. McIntyre, A. Bhomra, J. Nicod, X. Gan, W. Yuan, L. van der Weyden, C. A. Steward, S. Bala, J. Stalker, R. Mott, R. Durbin, I. J. Jackson, A. Czechanski, J. A. Guerra-Assunção, L. R. Donahue, L. G. Reinholdt, B. A. Payseur, C. P. Ponting, E. Birney, J. Flint, and D. J. Adams. Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature*, 477(7364):289–294, sep 2011.

[11] D. Kim, B. Langmead, and S. L. Salzberg. HISAT: a fast spliced aligner with low memory requirements. *Nature methods*, 12(4):357–60, 2015.

[12] A. A. Kolodziejczyk, J. K. Kim, V. Svensson, J. C. Marioni, and S. A. Teichmann. The Technology and Biology of Single-Cell RNA Sequencing. *Molecular Cell*, 58(4):610–620, may 2015.

[13] W. Lee, K. Plant, P. Humburg, and J. C. Knight. AltHapAlignR: improved accuracy of RNA-seq analyses through the use of alternative haplotypes. *Bioinformatics*, mar 2018.

[14] R. Leinonen, H. Sugawara, and M. Shumway. The Sequence Read Archive. *Nucleic Acids Research*, 39(Database):D19–D21, jan 2011.

[15] Y. Liao, G. K. Smyth, and W. Shi. The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, 41(10), 2013.

[16] Y. Liao, G. K. Smyth, and W. Shi. FeatureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30(7):923–930, 2014.

[17] P. Mell, T. Grance, et al. The nist definition of cloud computing. 2011.

[18] N. Philippe, M. Salson, T. Commes, and E. Rivals. CRAC: an integrated approach to the analysis of RNA-seq reads. *Genome Biology*, 14(3):R30, 2013.

[19] X. Shi, F. Nie, Z. Wang, and M. Sun. Pseudogene-expressed RNAs: a new frontier in cancers. *Tumour biology : the journal of the International Society for Oncodevelopmental Biology and Medicine*, 37(2):1471–8, feb 2016.

[20] J. Y. Tang, A. Yang, T. Y. Chen, and J. W. Ho. Harnessing Multiple Source Test Cases in Metamorphic Testing: A Case Study in Bioinformatics. In *2017 IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET)*, pages 10–13. IEEE, may 2017.

[21] K. Wang, D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, G. L. Savich, X. He, P. Mieczkowski, S. A. Grimm, C. M. Perou, J. N. MacLeod, D. Y. Chiang, J. F. Prins, and J. Liu. MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research*, 38(18):e178–e178, oct 2010.

[22] T. D. Wu, J. Reeder, M. Lawrence, G. Becker, and M. J. Brauer. GMAP and GSNAP for Genomic Sequence Alignment: Enhancements to Speed, Accuracy, and Functionality. pages 283–334. 2016.