Methods for long read RNA-seq transcriptomics

Thesis by
Rebekah Kiana Loving

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy, Biology

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2025
Defended May 30, 2025

Rebekah Kiana Loving
ORCID: 0000-0001-8725-0376

ACKNOWLEDGEMENTS

My journey as a PhD student has been one filled with great joy; exploring fascinating biology with the freedom to create new and better analysis approaches and working with wonderful, generous people has been the most amazing experience! Due to Covid-19 disruptions, the joy of marriage and subsequently birthing and mothering three children, and most recently the Eaton Fire, it has been a journey full of unexpected turns. There are far too many people to name, so I will only name a handful. Thank you to Lior and Barbara for going above and beyond to not only help me stay in the PhD program, but also to help me do some of the most rewarding work I can imagine doing in science.

To Lior Pachter, thank you for being the most supportive advisor through every up and down of research and life. Thank you for your confidence in my ability to do excellent science and for teaching and challenging me in countless areas. Thank you for your compassion as you mentor and your diligent search for truth and sound methods.

To Barbara Wold, thank you for every time you have challenged me to answer questions with more depth and encouraged me to take opportunities to improve our work whether in science or in our community and the world. Thank you for your confidence in me.

To Ali Mortazavi, Matt Thomson, and Pietro Perona, thank you for all the feedback and advice you have given me.

Thank you to Lynn Yi, Jase Gehring, Taleen Dilanyan, Tara Chari, Ángel Gálvez-Merchán, Delaney Sullivan, Kayla Jackson, Anne Kil, Maria Carilli, Cat Felce, Nikki Swarna, Laura Leubbert, Charlene Kim, and all Pachter lab members past and present who have influenced my work and my journey in many ways. I, especially, want to thank Taleen and Ángel for their encouragement during the uncertainty of Covid-19 when I took Dependent Care Leave and Delaney for encouraging me when I felt like I was banging my head against the wall of long read challenges.

I thank all my collaborators, especially Pall Melsted and Shinjae Yoo. Thank you to the Mortazavi Lab (especially Dana, Fairlie, Gabby, and Liz) for their collaboration and taking me under their wings at the ENCODE meeting in 2019.

Thank you to the DOECSGF administrators and cohort for simply being amazing human beings as well as for the funding of the DOECSGF!

For invaluable friendship, I thank Maria Carmaca and Hannah Manetsch. I am deeply grateful to the rest of GCF for their prayers and support throughout my PhD, especially Bekah the Elder and Rachel Gehlhar.

To Tali Khain, Daniel Mukasa, Jean Badroos, and the rest of WAVE 2017, without the experience with you all in 2017, I might not be here today. Special thanks to Tali Khain, dear friend, for being a constant, sweet encouragement.

To the ladies at Student Faculty Programs, especially Maria and Carol, thank you for loving on the kids and for being cheerleaders as I took this unconventional journey through graduate school.

I thank my professors and mentors from undergrad, especially Efren Ruiz, Phillipe Binder, Michael Peterson, and Keith Edwards as well as Donna Neuberg and Heather Mattie, who not only mentored me during my time in undergrad, but continued to give encouragement as I pursued my PhD.

To my mom friends, especially Ashley, Christina, Cecilia, and Rachel, but many others as well. Thank you for helping me stay sane and healthy!

I thank my siblings: Joshua (who inspired much of my early interest in computational biology), Marissa (who had confidence in me to finish my PhD as a mother), Alyssa (who encouraged me to maintain a healthy balance in life), Sarah (who helped me keep my perspective right), Josiah, Nehemiah, Isaiah, Jeremiah, Elijah, Malachi, and Noah for their love, curiousity, and crazy stories.

I thank my parents: to my mom, without your faithful teaching throughout all my homeschooling years in academics, faith, and life, I would not have been well equipped for the challenges I faced. To both my parents, without your faithful parenting, I would not have been able to handle the stresses of becoming a wife and mother alongside graduate school.

I, Rebekah Kiana Loving Ngo, dedicate this thesis to my children (Lydia, Hannah, Silas, baby girl, and any more to come) and to my husband, Peter Ngo (so thankful the journey of my PhD was at your side!), without your undying curiousity, love, joy, and support I could not have completed this work.

Most of all, for the grace that God bestowed on me each step of this journey, I give all glory to God for every fruit that it has produced.

ABSTRACT

While short read RNA-seq dominated the field for decades, long read RNA-seq is particularly useful for isoform-level expression analysis, genome annotation, detecting novelly splicing transcripts, identifying exact breakpoints in gene fusions, and discovering chimeric RNAs. Long read RNA-seq has rapidly scaled to the point of producing terabytes of data from a single set of experiments. Technological advances in RNA and DNA sequencing library preparation, chemistry used in the Oxford nanopores, and basecalling algorithms have reduced long read sequencing error rates to sub-1% error. Further, the cost of long read sequencing has dropped to about one hundred US dollars per human genome. These two factors have lead to the mass production of high-throughput, long read, and single-cell RNA-seq data. While recent tools for long read RNA-seq have been developed, they have not kept pace in scalability and accuracy with long read RNA-seq in the fashion that short read RNA-seq tools have met computational scalability and accuracy challenges. To address this, in this thesis, we leverage long *k*-mers and pseudoalignment for mapping and quantifying long reads in the novel algorithm implemented within `lr-kallisto`, which yields both efficiency and higher accuracy for long read mapping and quantification than previous tools. We demonstrate that long read RNA-seq has reached sufficient depth and accuracy to yield accurate quantification of isoform-level expression for differential expression analysis. Furthermore, we explore the feasibilty of also utilizing long *k*-mers and pseudoalignment in both transcript discovery in `dn`-kallisto and gene fusion and immune receptor sequence discovery with `fugi` with measured success. Thus, our tools will enable a more complete, accurate, and scalable analysis of single-cell and bulk RNA-seq than has hitherto been possible in both quantifications and differential expression analysis as well as investigation of gene fusions, chimeric RNAs, and immune receptor sequences without bias.

PUBLISHED CONTENT AND CONTRIBUTIONS

Loving, Rebekah K. et al. (2025). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv*. R.K.L conceived project, implemented and benchmarked project, and was principal writer of manuscript. This work is included in this thesis. DOI: 10.1101/2024.07.19.604364. eprint: https://www.biorxiv.org/content/early/2025/01/29/2024.07.19.604364.full.pdf. URL: https://www.biorxiv.org/content/early/2025/01/29/2024.07.19.604364.

Rebboah, Elisabeth et al. (2025). "Systematic cell-type resolved transcriptomes of 8 tissues in 8 lab and wild-derived mouse strains captures global and local expression variation". In: *bioRxiv*. R.K.L. provided feedback throughout this analysis. This work is not included in this thesis. DOI: 10.1101/2025.04.21.649844.

IGVF Consortium (July 2023). "The Impact of Genomic Variation on Function (IGVF) Consortium". English. In: *ArXiv*. R.K.L. has lead a contribution to this consortium in the work for lr-kallisto and fugi. This work is included in this thesis. arXiv: 2307.13708. URL: https://arxiv.org/abs/2307.13708.

Park, David Keetae et al. (Sept. 2023). "Overestimated prediction using polygenic prediction derived from summary statistics". In: *BMC Genomic Data (Online)* 24.1. R.K.L. participated in analysis and manuscript writing/editing. This work is not included in this thesis. DOI: 10.1186/s12863-023-01151-4.

Loving, Rebekah et al. (June 2021). "A Network Fusion Model Pipeline for Multi-Modal, Deep Learning for Tree Crown Detection". In: 1.1. R.L. conceived project, participated in implementation and analysis, and lead writing manuscript. This work is not included in this thesis., pp. 1–8.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

NOMENCLATURE

next generation sequencing (NGS). RNA-sequencing technologies that lead to the highly parallel, high throughput era of RNA sequencing allowing the capture of gene- and transcript-level expression and dynamics.

RNA-seq. sequencing of the RNA (ribonucleic acid) which is the transcribed DNA (deoxyribonucleic acid) from your genome at the time of sampling.

Sanger sequencing. The first form of DNA sequencing which made way for the first sequencing of the human genome..

third generation sequencing (TGS). DNA- and RNA-sequencing technologies that allow for long and ultra-long sequencing from hundreds to hundreds of thousands of basepairs in one read.

Chapter 1

OUTLINE, BACKGROUND, AND INTRODUCTION

1.1 Outline

This thesis is organized into six chapters. We outline the chapters below:

Chapter I We introduce the thesis, provide an outline, and give a conceptual background in both biology and computer science.

Chapter II We describe lr-kallisto, a scalable and accurate tool for long read RNA-sequencing quantification, as well as a benchmark of this method.

Chapter III We introduce unpublished, preliminary work demonstrating the usefulness of lr-kallisto for transcriptome discovery using long read RNA.

Chapter IV We provide the detailed workflow for single-cell lr-kallisto.

Chapter V We introduce the challenges in gene fusion and immune cell receptor sequence discovery and annotation and provide a solution for these aforementioned challenges with fugi, a fusion gene and immune cell receptor sequence discovery and annotator tool.

Chapter VI We provide a review of thesis topics, conclusions, and future avenues of interest with these newly developed tools.

1.2 Background

Genomics

On April 14, 2003, the Human Genome Project announced the sequencing of a first draft of the human genome with 92% coverage for $2.7 billion after 13 years (National Human Genome Research Institute, 2003)[1]. Advances in DNA and RNA sequencing over the last two decades have dramatically reduced the time and cost of sequencing genomes, and today genome sequencing is routine. Moreover, with Oxford Nanopore Technologies' long-read sequencing, "telomere-to-telomere" genomes can now be sequenced and in 2021 a complete genome only missing 0.3% was released (NCBI, 2021)[2], which was then improved to create a gapless genome in 2022 (NCBI, 2022)[3]. Today, a human genome can be sequenced at high fidelity

for a few hundred dollars. The advances in sequencing of DNA and RNA are leading us into a new age of discovery and medical care, but several technological and computational challenges remain. In this thesis we focus on the computational challenges and present solutions to some key problems. We begin by providing a brief background of the relevant biology and then introduce the topics of this thesis.

DNA sequencing history and methods

The first genome was sequenced almost entirely with Sanger sequencing technology, a tedious process often called "chain termination", where dideoxynucleotides (ddNTPs) are used to stopped DNA replication. When ddNTPs terminate the chains in enough successive rounds of heating and cooling, there is a near guarantee that a fragment of each subsequence length capped with a ddNTP is contained within the experiment sample, which can then be passed through a capillary gel electrophoresis. The capillary gel performs the size sorting of the fragments, as the shortest fragments get read off first and one basepair longer fragments are read off successively, resulting in a chromatogram readoff of the basepairs of the sequence. While this approach to sequencing is accurate, it is slow and much more expensive than next generation and third generation sequencing methods, which also provide more information for genome annotation (Sanger, Nicklen, and Coulson, 1977).

In the 1990s, microarrays became a popular technology for sequencing both DNA and RNA via an approach called "sequencing by hybridization". However, microarrays rely on predetermined DNA probes and can depend on bias-prone hybridization of these probes with DNA fragments. Moreover, although microarrays provided a method for gathering gene expression information that was not possible with the low throughput of Sanger sequencing, this method could not provide exon-level or most transcript-level expression information.

While the idea of next generation RNA-sequencing (RNA-seq) was suggested in 2002 (Consortium and RIKEN Genome Exploration Research Group Phase I & II Team, 2002), the first applications to bulk tissue sequencing was in 2008 (Mortazavi et al., 2008). RNA-seq is very similar in methodology to Sanger sequencing; however, instead of being limited to the sequencing of a single fragment per capillary gel, next generation sequencing (NGS) allows for massively parallel, high throughput gene and transcript-level analysis that makes novel discoveries of both transcripts, genes, and expression patterns possible. This is enabled by "sequencing by synthesis" where cycles of additions of nucleotides and reagents can be imaged

in realtime as the RNA strand is synthesized. Furthermore, because of its comparatively low cost and high throughput, NGS RNA-seq allows for the study of the highly dynamic transcriptome. Thus, RNA-seq has become highly useful for studying both the mechanisms of healthy organisms, disease, and drug impacts.

PacBio and Oxford Nanopore Technologies have now introduced and greatly advanced third generation sequencing (TGS) technologies. When these technologies came on the market, their sequencing error rates were in the neighbourhood of 15-20% error rates (Sahlin, Baudeau, et al., 2023). Thus, while they provided helpful insights for transcript discovery when paired with short read RNA-seq, there were significant difficulties in using them for building genome annotations and the throughput was too low for transcript-level quantification. In recent years, the sequencing error rate has dropped to sub 1% sequencing error (even to .01-.1% sequencing error) and the throughput has increased massively, this has paved the way for the first time to attempt a complete transcript annotation of genomes as well as high resolution transcript-level quantification of expression and studying transcript-level expression dynamics (Amarasinghe et al., 2020; Pardo-Palacios et al., 2023; Reese et al., 2023). However, the current long-read tools for transcript quantification and discovery have been found to both lack in accuracy and produce exons in transcript models from data that are unjustified in the data used to create them (Pardo-Palacios et al., 2023). Thus, in this thesis, we seek to address these challenges and make new discoveries with TGS.

RNA transcription

RNA (ribonucleic acid) transcription is the process by which RNA molecules are read out from the DNA (deoxyribonucleic acid) of the genome. There are three stages of transcription: initiation, elongation, and termination. Initiation alone has many components at play from transcription factors, promoters and inhibitors. These include not only protein transcription factors, but also RNA transcription factors, such as a whole collection of non-coding RNA which bind to RNA, DNA, and protein molecules to regulate transcription. When binding occurs in such a manner as to promote RNA polymerase binding to the specific DNA region called the promoter, the synthesis of the RNA can commence. Second, the stage of elongation begins where the RNA polymerase unzips the DNA double helix and synthesizes the new RNA molecule from base nucleotides using one of the strands of DNA as a template. Finally, as soon as the RNA polymerase reaches a termination sequence in the DNA, the transcription of the new molecule of RNA is complete

and released by the RNA polymerase.

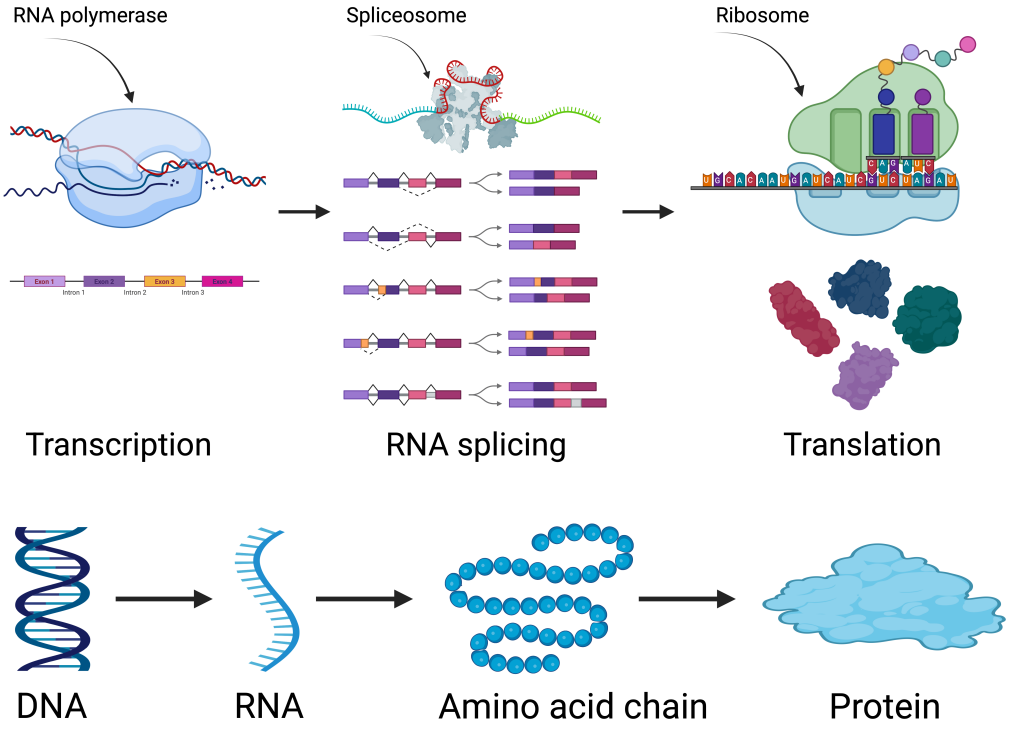


Figure 1.1: RNA Polymerase synthesizes the full RNA sequence from the start sequence to the termination sequence, including all exons and introns. Following synthesis, the RNA can be alternatively spliced by the spliceosome into different mature RNA. These mature RNA either are non-coding RNA or are translated by the ribosome into amino acid sequences which fold into proteins.

RNA splicing

The spliceosome is a large protein and RNA complex which is made up of 5 small nuclear ribonucleoproteins (snRNPs). These snRNPs bind to intronic regions and remove them by folding them and bringing the exons together in a single splice event or recursively removing long intronic regions. There are different sets of 5 snRNPs that have different properties, leading them to bind at different 5'-end and 3'end locations, thereby creating even more splicing diversity than is creating by different removal and retention of exons. Furthermore, trans-splicing is an event where two RNA are spliced and their exons joined from different genes, creating a chimeric RNA. This event has been found to occur both between endogenous and exogenous RNA as well as endogenous only RNA. Differential splicing and mis-splicing is implicated in many functions, and mis-splicing has been found to be important in the context of disease (Landrith et al., 2020).

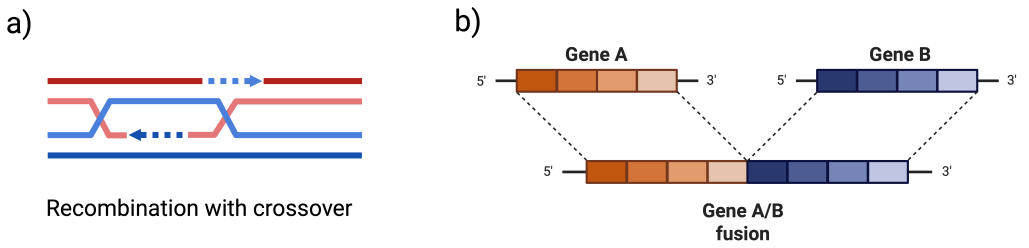


Figure 1.2: a) graphical depiction of DNA recombination with crossover. A cut occurs at the crossover between the pink and light blue at the two X's followed by DNA repair to join the light blue and light pink into the drawn positions. b) graphical depiction of fusion of Gene A and Gene B into GeneA/B.

RNA to protein translation and functional RNA

The ribosome is then responsible for converting the RNA sequence into its corresponding protein by recruiting the amino acids and forming them into a amino acid chain. This chain of amino acids then folds into a protein. Importantly, not every mature RNA is translated into a protein. Some mature RNA remain as RNA until they degrade and are highly functional in various mechanisms, including gene regulation and chromosome structure.

DNA recombination

The diversity of eukaryotic organisms is due to a large degree on the genetic recombination that occurs during meiosis of germ cells. During recombination homologous regions of DNA are broken and repaired joining regions of homologous DNA either from other chromosomes where there are homologous regions, referred to as interchromosomal recombination, or from paired chromosomes from the merged germ cell, intrachromosomal recombination (Figure 1.2a). Recombinases are key enzymes catalyzing recombination and DNA repair proteins, such as RAD51 and DMC1, are essential for meiotic recombination. While recombination typically occurs between homologous regions, it can also occur between DNA with no homology, leading to chromosomal translocation, which is sometimes cancer causing.

Immune cell receptor sequences

Adaptive immune response is a complex and amazing feat of biology where a specific population of T-cell receptors (TCRs) respond to an immune stimulus creating a TCR clonal expansion through genomic rearrangement. Prolific recombination along with the transmembrane structure of protein heterodimers are responsible for

immune cell receptors incredible level of diversity. In the case of T-cell receptor (TCR) diversity, there are $\alpha\beta$ and $\delta\gamma$ heterodimers. Though there are only four genes TCRα, TCRβ, TCRδ, and TCRγ, each of these genes are recombined in many, many ways to form different TCR proteins through what is called Variable (V) Diversity (D) Joining (J), V(D)J, recombination, where all four TCR genes have variable and joining regions and TCRβ and TCRδ also have diversity regions. Due to the many variable segments and the "random" recombination of these segments that occurs, one TCR gene can produce many amino acid sequences to create many different TCR protein complexes. These four TCR genes have been estimated to produce millions of distinct molecules. Thus, in response to immune stimuli, TCRs have the ability to randomly rearrange through clonal expansion into the correct configuration to fight the pathogenic invaders and, as these TCRs are more successful, they reproduce and fight the infection. Thus, examining TCR diversity, TCR lineage, and the specific V(D)J recombinations provides key insights into the immune system response. Historically, TCR sequences were not amenable to Sanger sequencing (Robins et al., 2009), but we now can use NGS to study the immune cell repertoire. Through study of the TCR reperoire and divergence between peripheral and within cancers or diseased tissue, immune response to cancers and disease can be monitored and may be useful biomarkers for monitoring and predicting outcomes to treatment (Sims et al., 2016) (Page et al., 2016) (Postow et al., 2015) (Robert et al., 2014).

DNA translocation

DNA translocation is simply the movement of a portion of one chromosome to a different chromosome. This often occurs as the result of DNA double-stranded breakage and a misrepair of this breakage. The repair is typically mediated by a mechanism called non-homologous end joining, as the name implies the fidelity of this does not rely on homologous regions and so error in the repair is much more likely than other DNA repair mechanisms. The outcomes of translocation may include gene fusions (Figure 1.2b), changes in gene regulation, and chromosome structural imbalances (Gingeras, 2009).

Gene fusions and chimeric RNAs

Gene fusions, which are created by DNA translocations, were previously used as a method for early cancer detection. However, we now know that gene fusions are not always cancerous, but may sometimes be benign or even beneficial. Chimeric RNAs,

which can sometimes be the result of a gene fusion and other times the result of a non-canonical splicing event such as trans-splicing or cis-splicing, similarly have been found to be functionally important in cellular mechanisms (Gingeras, 2009). To date, there has not been a thorough effort to catalogue the profundity of gene fusions and chimeric RNAs in the healthy context as well as in the diseased/atypical status (Mertens et al., 2015). However, it is known that there are significant implications of fusions creating pathogenic behaviors in cancer as well as fusions which give rise to immunotherapies for cancers (Stransky et al., 2014) (Y. Wang et al., 2021). Recently, high quality datasets have been created that are fit to fully delve into these questions; however, the workflows for analysis of this data are lagging with CTAT-LR Fusion being one of the only options, which relies on minimap2 for alignment (Qin et al., 2025). Thus, here we present fusion gene detection software (fugi) and an analysis of IGVF data with fugi, which instead uses kallisto as its base reducing memory and computational demands for processing.

Biology obfuscated

The standard methods of RNA sequencing which utilized fragmented RNA molecules and the analysis methods which focused completely on the gene-level analysis rather than the transcript-level analysis has obscured at best and misdirected or blatantly falsified at worse the true dynamics. For instance, a gene that is called as differentially expressed when looking at gene-level counts between two samples will often no longer be called as differentially expressed when considering the transcript-level counts between the two samples. Further, we now know that transcripts within the same gene may have significantly different functionality within the cell. Without high depth and without long read RNA sequencing, transcript differential expression, transcript switching, gene fusions, and immune cell receptor information are all tentative at best and simply wrong at worst.

transcript deBruijn Graphs (t-DBG)

transcript deBruijn Graphs (t-DBG) are colored de Bruijn graphs that are built from the transcriptomes and colored by the corresponding transcripts. Each node of the t-DBG is a k-mer, which is colored for each transcript that contains it. Contigs are then formed by linear stretches of the t-DBG sharing the same coloring, giving us that all k-mers within the contig share the same transcript compatibility set. Once the t-DBG is constructed with all of its composing contigs, a hash table is created mapping each k-mer to the contig and position within the contig which contains the

k-mer. This is how kallisto's index is constructed.

Alignment methods

Conventional alignment methods often employ seed, extend, and chaining with a weighting metric to penalize and reward different properties of the chain, i.e. longer chains with smaller gaps is a "better" alignment than a shorter chain with larger gaps. The seed from an RNA-seq read, a subsequence of the read, can be extended along the reference for as far as it matches in both directions. The regions that map and extend within the reference then can be chained together into longer regions of mapping with some gaps. These "chains" must then be ranked by a scoring or weighting function to determine the best mapping.

Quantification approaches

Many quantification methods for short reads have adopted length normalization approaches. However, for long read RNA-seq quantification, many quantification methods simply perform a counting of best alignments. The usefulness of length normalization in long reads is explored in Chapter 2 of this thesis.

1.3 Introduction

Overview of current tools and methods for long read sequence analysis, excerpt with modifications from lr-kallisto (Chapter 2)

Advances in long-read RNA sequencing are changing the paradigm of transcript discovery, annotation improvements, and detection of isoform switching, thanks to reductions in cost and decreasing error rates as the fundamental technologies of long read sequencing mature (Amarasinghe et al., 2020; Pardo-Palacios et al., 2023; Reese et al., 2023). Specifically, long-read RNA-seq can now readily detect gene fusion transcripts as well as their exact breakpoints, chimeric RNAs, and other expressed rearrangements in cancer (Sakamoto, Sereewattanawoot, and Suzuki, 2020), and isoform switching of biological consequence across development and disease (Chaoyang Wang et al., 2024; Penter et al., 2024). In translational genomics, precision medicine workflows are increasingly including gene and transcript ontology, as we now know gene ontology is not sufficient due to the sometimes differential functioning of transcripts form the same gene. These capabilities depend, in part, on accurate annotation of the genomes and transcriptomes of human and model organisms, though they remain incomplete (Zhang et al., 2020; Frankish et al., 2021). Improvements in long-read sequencing now allow for much needed refine-

ment of annotations for human and model organisms, coupled with rapid generation of genomes and annotations for non-model organisms (Warburton and Sebra, 2023). Importantly, while annotation is mainly facilitated by transcript discovery, quantification of isoforms is critical for filtering and thresholding steps that are prerequisites for resolving locus structure and quantifying their expression products (Cook et al., 2019).

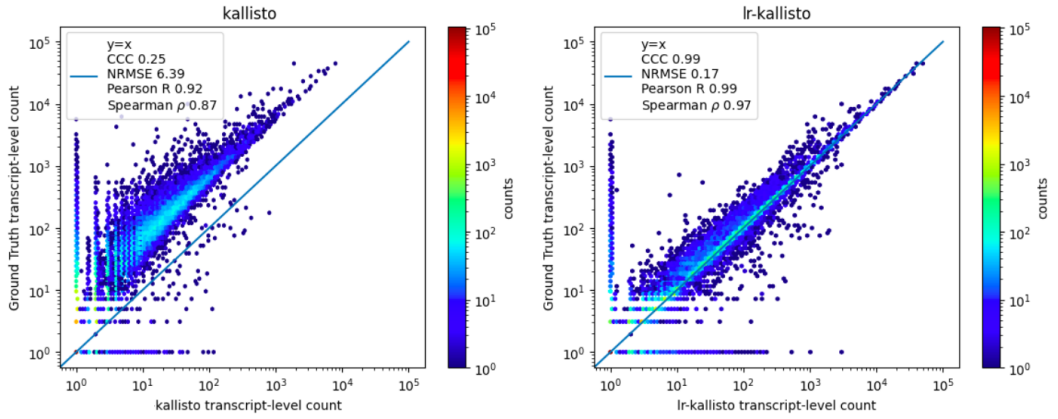


Figure 1.3: Motivation: Comparison of kallisto vs lr-kallisto on PacBio 1.4% error simulation.

While recent increases in affordability and sequence quality are bringing full-isoform quantification within reach, the long-read platforms are still rapidly changing and less mature than short-read technologies (Pardo-Palacios et al., 2023). For example, Oxford Nanopore Technology (ONT) sequencing has evolved over many versions of chemistry in the library preparation kits, pores, and signal processing algorithms. This has resulted in a range of ONT data with various error profiles and error distributions within the sequences. Of the quantification tools that have been developed so far (Tang et al., 2020; Tian et al., 2021; Wyman et al., 2019; Lienhard et al., 2023; Chen et al., 2023; Jousheghani and Patro, 2024; Prjibelski et al., 2023; Yang et al., 2017; Kabza et al., 2023), many are optimized for performance with a given generation of long-read data and are now antiquated, in both accuracy and efficiency, for processing the low error rate ONT data currently being produced. Moreover, many methods are based on the assumption of near uniform distribution of sequencing error along reads; we found, as have others (H. Li, 2018), that this does not hold in practice. Furthermore, some ONT sequencing biases have now been described, including non-uniformly distributed sequencing error and sequence influenced error, such as higher GC content and repeat regions increasing sequencing/base calling error (Delahaye and Nicolas, 2021).

Many approaches have been applied to RNA-seq quantification from classical alignment approaches to pseudoalignment paired with likelihoods and expectation-maximization (EM). Due to its speed, efficiency, and accuracy, pseudoalignment with likelihoods and EM has been widely adopted for the mapping of short read RNA-seq. However, for long-read RNA-seq, minimap2 has become the standard for aligning long-reads. Minimap2 follows the standard genome alignment methodology of seed-chain-align (H. Li, 2018). It creates a reference index in the form of hashing minimizers into keys for a reference hash table storing the list of genomic/transcriptomic locations of the minimizer. For each read, minimap2 uses read minimizers as seeds matching these to the reference hash table and identifies the optimal collinear chain(s) of matches. While this method is accurate and has been developed to be highly efficient for the alignment strategy used, it is still time and resource expensive with high memory storage demands.

lr-kallisto, building on the existing framework of kallisto and adapting the pseudoalignment and expectation-maximization algorithm for long-reads, gives an accurate, fast, and low resource solution for mapping long-reads. The main technical challenge of long-reads lies in the higher sequencing error rates, though others include the differing rates of substitutions, deletions, and insertions between long-read sequencing technologies, sequencing length, repetitive regions, and concatemers. To address the challenge of higher sequencing error, different methods, including minimap2 (H. Li, 2018), uLTRA (Sahlin and Mäkinen, 2021), and STAR (Dobin et al., 2013) have utilized various approaches to long-read alignment. Minimap2 uses a small k-mer size of 14 and 15 for long-reads, while uLTRA employs a two-pass chaining algorithm to improve alignment accuracy. Strobemers have been suggested using fuzzy k-mers that allow error tolerance (Sahlin, 2021). In lr-kallisto, we, instead, propose a long k-mer length and "chaining" pseudoalignment for addressing the challenges of long-read alignment.

Pseudoalignment methods

Instead of directly performing global alignments, pseudoalignment uses indexing of subsequences of length k, termed k-mers, of the reference to their matching sets of compatible transcripts. This is further excelerated by using a hashing map where the key is the k-mer and the map contains an integer corresponding to the transcript compatibility set that the k-mer belongs to. To find the best alignment for each read, we can then take the intersection of the transcript compatibility sets that the k-mers within a read map to. Furthermore, there are several optimizations that are

implemented within the pseudoalignment. For example, when a k-mer from a read maps the match is extended along the read to identify the full length of the matching region with the read to that same transcript compatibility set. Other optimizations are also performed for high accuracy reads, which have been modified for good performance on higher sequencing error and longer reads. To better understand how this pseudoalignment works we give a brief description of the data structure that is used as the index for the transcriptome, the transcript de Bruijn Graph.

Challenges for long read sequence analysis

Long read RNA sequencing has evolved over the past two decades from a very high sequencing error rate of up to around 20% sequencing error to a sequencing error rate of sub-1%. While this is an monumental improvement that allows for new applications of long read sequencing, the sequencing error rate as well as the length of the sequences themselves still present challenges to RNA and DNA mappers and expression quantifiers both in the domain of accuracy and efficieny. Long read RNA sequences create high memory usage demands in the current standard software for alignment and quantification as well as long run-times. Furthermore, the historically high error rates, which have recently significantly decreased while still being about 100-fold higher than short read sequencing, have created challenges. Here, we present two tools which successfully shorten run time and require similar memory usage to the most efficient short read alignment and quantification tools, while increasing mapping and quantification accuracy with lr-kallisto in Chapters 2, 3 (application to discovery), and 4 (application to single-nuclei data) and providing the first unbiased, selection free detection and annotation of immune cell receptor sequences (Chapters 5 and 6, which also includes fusion gene detection and applications).

Discoveries with long read sequences

Long read RNA sequencing provides the ability to confidently address biology obfuscation that short reads created, including improving genome annotations, accurately acquiring transcript-level expression and dynamics, accurately detecting gene fusions, and characterizing immune cell receptor sequences. Each of these individually bares significant potential contributions to the study of developmental and mechanisms of biology, cancers and other diseases, and drug development paradigm. In particular, with

- lr-kallisto:
  - `kb count` with its pseudoalignment and quantification algorithms provide a tractable and accurate new approach for mapping and transcript-level expression from TGS reads
  - the discovery framework provides a simple, unbiased framework for exploring transcript models
- fugi:
  - fusion detection is directly useful for studying both nominally healthy gene fusions and detecting possible cancer-related gene fusions
  - the immune cell receptor sequence identification and annotation provide an unbiased method for building expression databases of resident T-cells in different states of healthy and diseased tissues and beginning to seive through their possible healthy and unhealthy responses.

BIBLIOGRAPHY

Amarasinghe, Shanika L et al. (2020). "Opportunities and challenges in long-read sequencing data analysis". In: *Genome Biology* 21.1, p. 30.

Chen, Ying et al. (2023). "Context-aware transcript quantification from long-read RNA-seq data with Bambu". In: *Nature Methods* 20.8, pp. 1187–1195.

Consortium, The FANTOM and the RIKEN Genome Exploration Research Group Phase I & II Team (2002). "Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs". In: *Nature* 420, pp. 563–573. DOI: 10.1038/nature01266. URL: https://doi.org/10.1038/nature01266.

Cook, David E et al. (2019). "Long-read annotation: Automated eukaryotic genome annotation based on long-read cDNA sequencing". In: *Plant Physiology* 179.1, pp. 38–54.

Delahaye, Clara and Jacques Nicolas (2021). "Sequencing DNA with nanopores: Troubles and biases". In: *PloS One* 16.10, e0257521.

Dobin, Alexander et al. (2013). "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1, pp. 15–21.

Frankish, Adam et al. (2021). "GENCODE 2021". In: *Nucleic Acids Research* 49.D1, pp. D916–D923.

Gingeras, Thomas R. (2009). "Implications of chimeric non-co-linear transcripts". In: *Nature* 461, pp. 206–211. DOI: https://doi.org/10.1038/nature08452.

Jousheghani, Zahra Zare and Rob Patro (Mar. 2024). "Oarfish: Enhanced probabilistic modeling leads to improved accuracy in long read transcriptome quantification". In: *bioRxiv*. DOI: 10.1101/2024.02.28.582591.

Kabza, Michal et al. (2023). "Accurate long-read transcript discovery and quantification at single-cell resolution with Isosceles". In: *bioRxiv*, pp. 2023–11.

Landrith, Tyler et al. (2020). "Splicing profile by capture RNA-seq identifies pathogenic germline variants in tumor suppressor genes". In: *NPJ precision oncology* 4.1, p. 4.

Li, Heng (2018). "Minimap2: Pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094–3100.

Lienhard, Matthias et al. (2023). "IsoTools: a flexible workflow for long-read transcriptome sequencing analysis". In: *Bioinformatics* 39.6, btad364.

Mertens, F. et al. (2015). "The emerging complexity of gene fusions in cancer". In: *Nature Reviews Cancer* 15.6, pp. 371–381. DOI: 10.1038/nrc3947. URL: https://doi.org/10.1038/nrc3947.

Mortazavi, Ali et al. (2008). "Mapping and quantifying mammalian transcriptomes by RNA-Seq". In: *Nature Methods* 5.7, pp. 621–628. DOI: 10.1038/nmeth.1226.

National Human Genome Research Institute (2003). *Human Genome Project Completion: Frequently Asked Questions*. Accessed April 2025. URL: https://www.genome.gov/human-genome-project/Completion-FAQ.

NCBI (2021). *CHM13 T2T v1.1 – Genome – Assembly*. Accessed April 2025. URL: https://www.ncbi.nlm.nih.gov/assembly/GCF_009914755.1/.

– (2022). *T2T-CHM13v2.0 – Genome – Assembly*. Accessed April 2025. URL: https://www.ncbi.nlm.nih.gov/assembly/GCA_009914755.4/.

Page, David B. et al. (2016). "Deep Sequencing of T-cell Receptor DNA as a Biomarker of Clonally Expanded TILs in Breast Cancer after Immunotherapy". In: *Cancer Immunology Research* 4.10, pp. 835–844. DOI: 10.1158/2326-6066.CIR-16-0013. URL: https://doi.org/10.1158/2326-6066.CIR-16-0013.

Pardo-Palacios, Francisco J et al. (July 2023). "Systematic assessment of long-read RNA-seq methods for transcript identification and quantification". In: *bioRxiv*. DOI: 10.1101/2023.07.25.550582.

Penter, Livius et al. (2024). "Integrative genotyping of cancer and immune phenotypes by long-read sequencing". In: *Nature Communications* 15.1, p. 32.

Postow, Michael A. et al. (2015). "Peripheral T cell receptor diversity is associated with clinical outcomes following ipilimumab treatment in metastatic melanoma". In: *Journal for ImmunoTherapy of Cancer* 3.1. DOI: 10.1186/s40425-015-0070-4. eprint: https://jitc.bmj.com/content/3/1/23.full.pdf. URL: https://jitc.bmj.com/content/3/1/23.

Prjibelski, Andrey D et al. (2023). "Accurate isoform discovery with IsoQuant using long reads". In: *Nature Biotechnology* 41.7, pp. 915–918.

Qin, Qian et al. (2025). "CTAT-LR-fusion: accurate fusion transcript identification from long and short read isoform sequencing at bulk or single cell resolution". In: *Genome Research* 35.4. Originally published as a bioRxiv preprint: doi:10.1101/2024.02.24.581862, pp. 967–986. DOI: 10.1101/gr.279200.124. URL: https://doi.org/10.1101/gr.279200.124.

Reese, Fairlie et al. (May 2023). "The ENCODE4 long-read RNA-seq collection reveals distinct classes of transcript structure diversity". In: *bioRxiv*. DOI: 10.1101/2023.05.15.540865.

Robert, Caroline et al. (2014). "Association of T-cell repertoire diversity with clinical outcomes following ipilimumab treatment in metastatic melanoma". In: *Clinical Cancer Research* 20.20, pp. 5346–5354. DOI: 10.1158/1078-0432.CCR-13-3320. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4469400/.

Robins, Harlan S. et al. (2009). "Comprehensive assessment of T-cell receptor -chain diversity in T cells". In: *Blood* 114.19, pp. 4099–4107. DOI: 10.1182/blood-2009-04-217604. URL: https://doi.org/10.1182/blood-2009-04-217604.

Sahlin, Kristoffer (2021). "Effective sequence similarity detection with strobemers". In: *Genome Research* 31.11, pp. 2080–2094.

Sahlin, Kristoffer, Théophile Baudeau, et al. (2023). "A survey of mapping algorithms in the long-reads era". In: *Genome Biology* 24.1, p. 133. DOI: 10.1186/s13059-023-02972-3. URL: https://doi.org/10.1186/s13059-023-02972-3.

Sahlin, Kristoffer and Veli Mäkinen (2021). "Accurate spliced alignment of long RNA sequencing reads". In: *Bioinformatics* 37.24, pp. 4643–4651.

Sakamoto, Yoshitaka, Sarun Sereewattanawoot, and Ayako Suzuki (2020). "A new era of long-read sequencing for cancer genomics". In: *Journal of Human Genetics* 65.1, pp. 3–10.

Sanger, Frederick, Steven Nicklen, and Alan R. Coulson (1977). "DNA sequencing with chain-terminating inhibitors". In: *Proceedings of the National Academy of Sciences* 74.12, pp. 5463–5467. DOI: 10.1073/pnas.74.12.5463.

Sims, J. S. et al. (2016). "Diversity and divergence of the glioma-infiltrating T-cell receptor repertoire". In: *Proceedings of the National Academy of Sciences* 113.25, E3529–E3537. DOI: 10.1073/pnas.1601012113. URL: https://doi.org/10.1073/pnas.1601012113.

Stransky, N. et al. (2014). "The landscape of kinase fusions in cancer". In: *Nature Communications* 5, p. 4846. DOI: 10.1038/ncomms5846. URL: https://doi.org/10.1038/ncomms5846.

Tang, Alison D et al. (2020). "Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns". In: *Nature communications* 11.1, p. 1438.

Tian, Luyi et al. (2021). "Comprehensive characterization of single-cell full-length isoforms in human and mouse with long-read sequencing". In: *Genome biology* 22, pp. 1–24.

Wang, Chaoyang et al. (2024). "Single-cell analysis of isoform switching and transposable element expression during preimplantation embryonic development". In: *PLoS Biology* 22.2, e3002505.

Wang, Y. et al. (2021). "Gene fusion neoantigens: Emerging targets for cancer immunotherapy". In: *Cancer Letters* 506, pp. 45–54. DOI: 10.1016/j.canlet.2021.02.023. URL: https://doi.org/10.1016/j.canlet.2021.02.023.

Warburton, Peter E and Robert P Sebra (2023). "Long-read DNA sequencing: recent advances and remaining challenges". In: *Annual Review of Genomics and Human Genetics* 24, pp. 109–132.

Wyman, Dana et al. (2019). “A technology-agnostic long-read analysis pipeline for transcriptome discovery and quantification”. In: *Biorxiv*, p. 672931.

Yang, Chen et al. (2017). “NanoSim: nanopore sequence read simulator based on statistical characterization”. In: *GigaScience* 6.4, gix010.

Zhang, David et al. (2020). “Incomplete annotation has a disproportionate impact on our understanding of Mendelian and complex neurogenetic disorders”. In: *Science Advances* 6.24. DOI: 10.1126/sciadv.aay8299.

Chapter 2

LONG READ SEQUENCING TRANSCRIPTOME QUANTIFICATION WITH lr-kallisto

Loving, Rebekah K et al. (2024). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv v1*, pp. 2024–07.

2.1 Abstract

RNA abundance quantification has become routine and affordable thanks to high-throughput "short-read" technologies that provide accurate molecule counts at the gene level. Similarly accurate and affordable quantification of definitive full-length, transcript isoforms has remained a stubborn challenge, despite its obvious biological significance across a wide range of problems. "Long-read" sequencing platforms now produce data-types that can, in principle, drive routine definitive isoform quantification. However some particulars of contemporary long-read datatypes, together with isoform complexity and genetic variation, present bioinformatic challenges. We show here, using ONT data, that fast and accurate quantification of long-read data is possible and that it is improved by exome capture. To perform quantifications we developed lr-kallisto, which adapts the kallisto bulk and single-cell RNA-seq quantification methods for long-read technologies.

2.2 Introduction

Advances in long-read RNA sequencing are facilitating transcript discovery, annotation improvements, and detection of isoform switching, thanks to reductions in cost and decreasing error rates as the technologies mature (Amarasinghe et al., 2020; Pardo-Palacios et al., 2023; Reese et al., 2023). Specifically, long-read RNA-seq can readily detect gene fusion transcripts and other expressed rearrangements in cancer (Sakamoto, Sereewattanawoot, and Suzuki, 2020), and isoform switching of biological consequence across development (Chaoyang Wang et al., 2024; Penter et al., 2024). In translational genomics, precision medicine workflows are increasingly including gene and transcript ontology. These capabilities depend, in part, on accurate annotation of the genomes and transcriptomes of human and model organisms, though they remain incomplete (Zhang et al., 2020; Frankish et al., 2021). Improvements in long-read sequencing now allow for much needed refine-

ment of annotations for human and model organisms, coupled with rapid generation of genomes and annotations for non-model organisms (Warburton and Sebra, 2023). Importantly, while annotation is mainly facilitated by transcript discovery, quantification of isoforms is critical for filtering and thresholding steps that are prerequisites for resolving locus structure and quantifying their expression products (Cook et al., 2019).

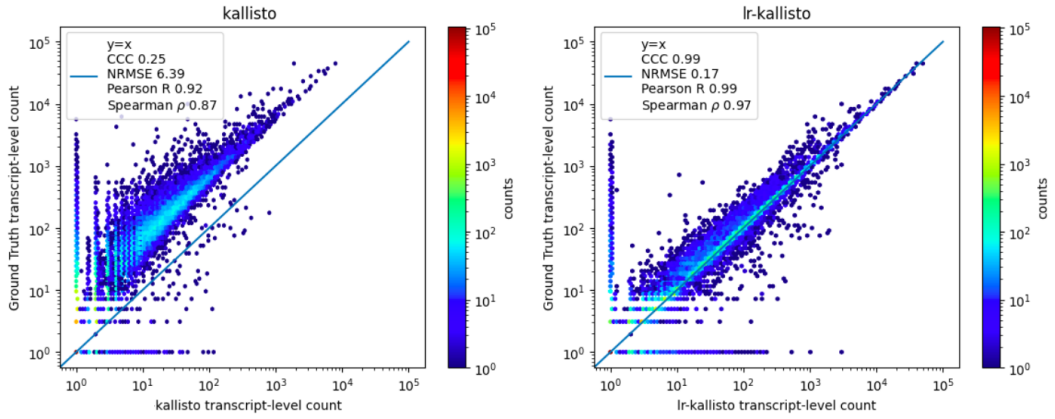


Figure 2.1: Motivation: Comparison of kallisto vs lr-kallisto on PacBio 1.4% error simulation.

While recent increases in affordability and sequence quality are bringing full-isoform quantification within reach, the long-read platforms are still rapidly changing and less mature than short-read technologies (Pardo-Palacios et al., 2023). For example, Oxford Nanopore Technology (ONT) sequencing has evolved over many versions of chemistry in the library preparation kits, pores, and signal processing algorithms. This has resulted in a range of ONT data with various error profiles and error distributions within the sequences. Of the quantification tools that have been developed so far (Tang et al., 2020; Tian et al., 2021; Wyman et al., 2019; Lienhard et al., 2023; Chen, Sim, et al., 2023; Jousheghani and Patro, 2024; Prjibelski et al., 2023; Yang et al., 2017; Kabza et al., 2023), many are optimized for performance with a given generation of long-read data and are now antiquated, in both accuracy and efficiency, for processing the low error rate ONT data currently being produced. Moreover, many methods are based on the assumption of near uniform distribution of sequencing error along reads; we found, as have others (H. Li, 2018), that this does not hold in practice. Furthermore, some ONT sequencing biases have now been described, including non-uniformly distributed sequencing error and sequence influenced error, such as higher GC content and repeat regions increasing sequencing/base calling error (Delahaye and Nicolas, 2021).

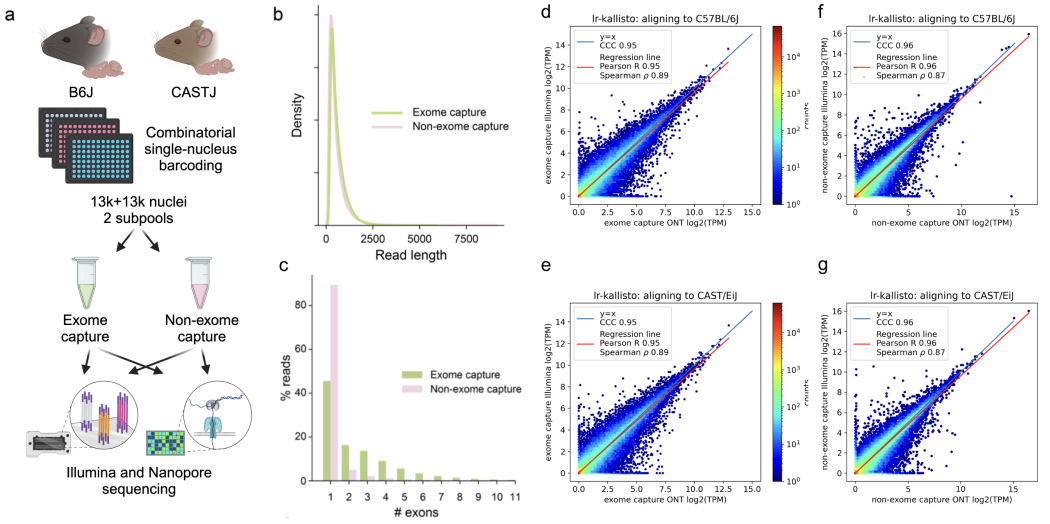


Figure 2.2: lr-kallisto demonstrates high concordance between Illumina and ONT. (a) Experimental overview for comparison of exome capture vs. non-exome capture LR-Split-seq libraries. (b) Kernel density estimations for read length distributions by capture strategy. (c) Percentage of demultiplexed reads by number of exons in each read between exome and non-exome capture. (d-g) Each point is a hexbin representing the number of transcript in the bin with expression in log2(TPM) with x-coordinate quantified from long reads and y-coordinate quantified from short reads. Total number of points is the total number of annotated transcripts in the reference transcriptome. CCC is a measure of how close the data is to $x = y$, while Pearson R and Spearman ρ are measures of correlation between x and y. (d) lr-kallisto pseudobulk quantifications of exome capture for the C57BL/6J sample. (e) lr-kallisto pseudobulk quantifications of exome capture for the CAST/Eij sample. (f) lr-kallisto pseudobulk quantifications of non-exome capture for the C57BL/6J sample. (g) lr-kallisto pseudobulk quantifications of non-exome capture for the CAST/Eij sample. Concordance Correlation Coefficient (CCC), Pearson, and Spearman correlations are shown for each comparison. Created with `https://BioRender.com`

By contrast, several accurate and efficient tools have been developed for short read RNA-seq preprocessing (Kaminow, Yunusov, and Dobin, 2021; Sullivan, Min, et al., 2023; Melsted et al., 2021; Bray et al., 2016; Sarkar et al., 2020; Patro et al., 2017). However, even with the recent significant reduction in the long-read RNA-seq error rates to ~0.5%, sequencing errors remain informatically problematic and are comparatively much higher than the ~0.01% in short-read RNA-seq. This makes the application of the fastest pseudoalignment methods (Bray et al., 2016; Patro et al., 2017) to long-reads nontrivial (Figure 2.1). Our approach, which builds on kallisto (Bray et al., 2016; Hjörleifsson et al., 2022; Sullivan, Min, et al., 2023; Melsted et al., 2021) and which we term lr-kallisto, demonstrates the feasibility of pseudoalignment

for long-reads; we show via a series of results on both biological and simulated data that lr-kallisto retains the efficiency of kallisto thanks to pseudoalignment, and is accurate on long-read data. Furthermore, we show that lr-kallisto is robust to error rates, making it suitable also for the analysis of previously published older long-read sequencing data.

2.3 Results

To assess the accuracy of lr-kallisto with respect to data from the current Oxford Nanopore Technologies platform (see Materials and Methods) we generated a deep coverage, high fidelity dataset using long-read sequence and an Illumina short-read sequence SPLiT-Seq nuclei of the left cortices of two mouse strains as part of the IGVF consortium (Consortium, 2023). Specifically, four biological replicates (two males and two females) were assayed from both C57BL6/J and CAST/EiJ mice, all at ten weeks of age, with libraries generated with and without targeted exome capture of all mouse protein coding exons using the Twist Biosciences Mouse exome panel of 215,000 probes (Figure 2.2a; see Methods). Thus our exome capture transcriptome will be enriched for reads overlapping one or more coding exons in the same cell. This platform and experimental design was chosen to produce starting data with a highly relevant sequencing error profile for two very well characterized genomes whose natural genetic variation between strains is similar to that found within individual human genomes. This also sets the stage for using lr-kallisto to study natural genetic variation.

We found no effective difference in read lengths with reads generated from exome capture as opposed to non-exome capture libraries (Figure 2.2b), though the exome capture library showed a smaller fraction of mono-exonic reads (Figure 2.2c). This indicated that exome capture is an effective approach to increasing the transcriptome complexity of libraries. The Illumina and ONT sequenced libraries displayed high transcript abundance concordance after quantification with lr-kallisto (Figure 2.2d-g), showing that lr-kallisto accurately quantifies transcripts from long-reads, as well as demonstrating that deeply sequenced ONT libraries are suitable for high accuracy quantification. The concordance correlation coefficients (CCC), which measure how close the ONT and Illumina quantifications are to being identical, were high for both the exome capture and non-exome capture libraries (0.95 and 0.96, respectively). Importantly, when comparing all long-reads that were subject to exome capture versus those that were not, we observed a three-fold increase in the percentage of spliced reads aligning (Figure 2.3). Thus, we find that exome capture

reads help overcome the limitations of RNA sampling in the nucleus, including expected reads from unspliced precursor transcripts. The effect, as others have noted (Landrith et al., 2020) is to provide more informative reads to study full-length, spliced transcript isoform usage at lower cost. Furthermore, lr-kallisto outperforms Bambu (Chen, Sim, et al., 2023), IsoQuant (Prjibelski et al., 2023), and Oarfish (Jousheghani and Patro, 2024) with respect to CCC, Pearson correlation, and Spearman correlation (Figure 2.4). In particular, the lr-kallisto CCC is 0.95 versus 0.82 for the recently published Oarfish long-read quantification tool. We found that lr-kallisto also outperforms Bambu (CCC = 0.86) and IsoQuant (CCC = 0.78) (Figure 2.4), which have previously been shown to outperform other long-read quantification methods (Pardo-Palacios et al., 2023; Dong et al., 2023). In addition to being more accurate than other methods, lr-kallisto is also more computationally efficient (Figure 2.5). Note that the dramatic difference in time scales between PacBio and ONT is due to the number of reads in the ONT datasets being much higher, in general. We further benchmarked walltime and maximum resident set size of bambu, IsoQuant, lr-kallisto, and oarfish on the exome capture data, showing that lr-kallisto is 3-10x more memory efficient than oarfish, 18-50x more memory efficent than bambu, and 15-46x more memory efficient than IsoQuant (Table 2.1).

Importantly, lr-kallisto can be used for both high-throughput bulk RNA-seq as well as single-cell or single-nuclei RNA-seq datasets (Figure 2.6-2.8), and is not only faster than other tools, but also benefits from the low-memory requirements of kallisto (Sullivan, Min, et al., 2023; Hjörleifsson et al., 2022) (Table 2.1). For single-nuclei RNA-seq processing, we used splitcode (Sullivan and Pachter, 2024) to extract nuclei barcodes, umis, and reads from the raw ONT reads and then pseudoaligned and quantified the reads with lr-kallisto (see Methods). 100% of barcodes from the ONT reads that passed filtering were also found in Illumina sequenced reads (Figure 2.6). Increased UMI depth per nucleus yields higher Spearman correlations, indicating that with deeper sequencing depth, short and long read correlations will only improve (Figure 2.6I). To assess the observed correlations between Nanopore and Illumina, we evaluated random oligo vs 3' priming in Illumina sequenced reads and ONT sequenced reads separately, in the same fashion, finding lower correlations (majority of nuclei having a Spearman ρ between 0.10 and 0.30) than between ONT sequenced reads and Illumina sequenced reads (Figure 2.6II-III). We expect that with increased depth of ONT sequencing (i.e. a comparable number of error corrected barcodes) we would achieve a near 100% overlap of barcodes.

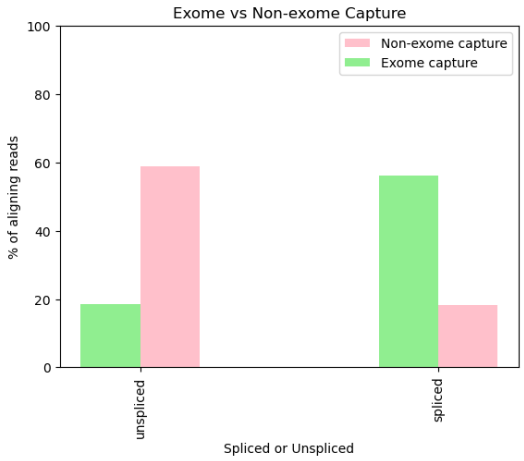


Figure 2.3: Comparison of the percent of reads mapping as spliced vs. unspliced reads with and without exome capture.

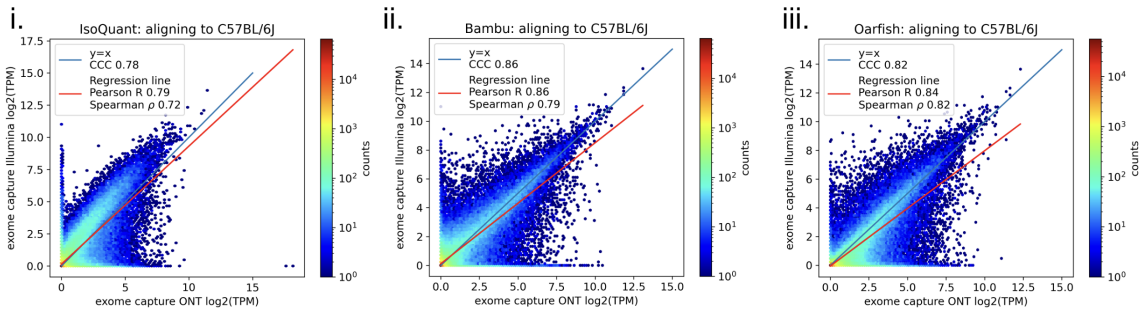


Figure 2.4: Quantifications of the C57BL/6J exome capture samples with Bambu, IsoQuant, and Oarfish.

We also examined the concordance between the exome capture and non-exome capture in both long and short reads, and found it to be only CCC = 0.88, highlighting the distortion resulting from the coupling of exome capture with a mix of 3'-end and randomly primed read sequencing that is characteristic of Parse (Figure 2.7). We further explored the difference in 3'-end and randomly primed read sequencing by looking at the mean transcript-level expression of 3'-end (polyT) reads in a cell in ONT vs Illumina, in randomly primed (randO) reads in a cell in ONT vs Illumina, and when reads from the two priming methods are merged for a cell. We found that randomly primed reads in a cell exhibit greater correlation to the corresponding randomly primed reads in Illumina in the highest mean expressed 100 transcripts in ONT than with 3'-end reads in a cell exhibit (Figure 2.8). We found that the expression of long non-coding RNAs (lnc-RNAs) is significantly different between the exome capture and non-exome capture with 8,392 lnc-RNAs

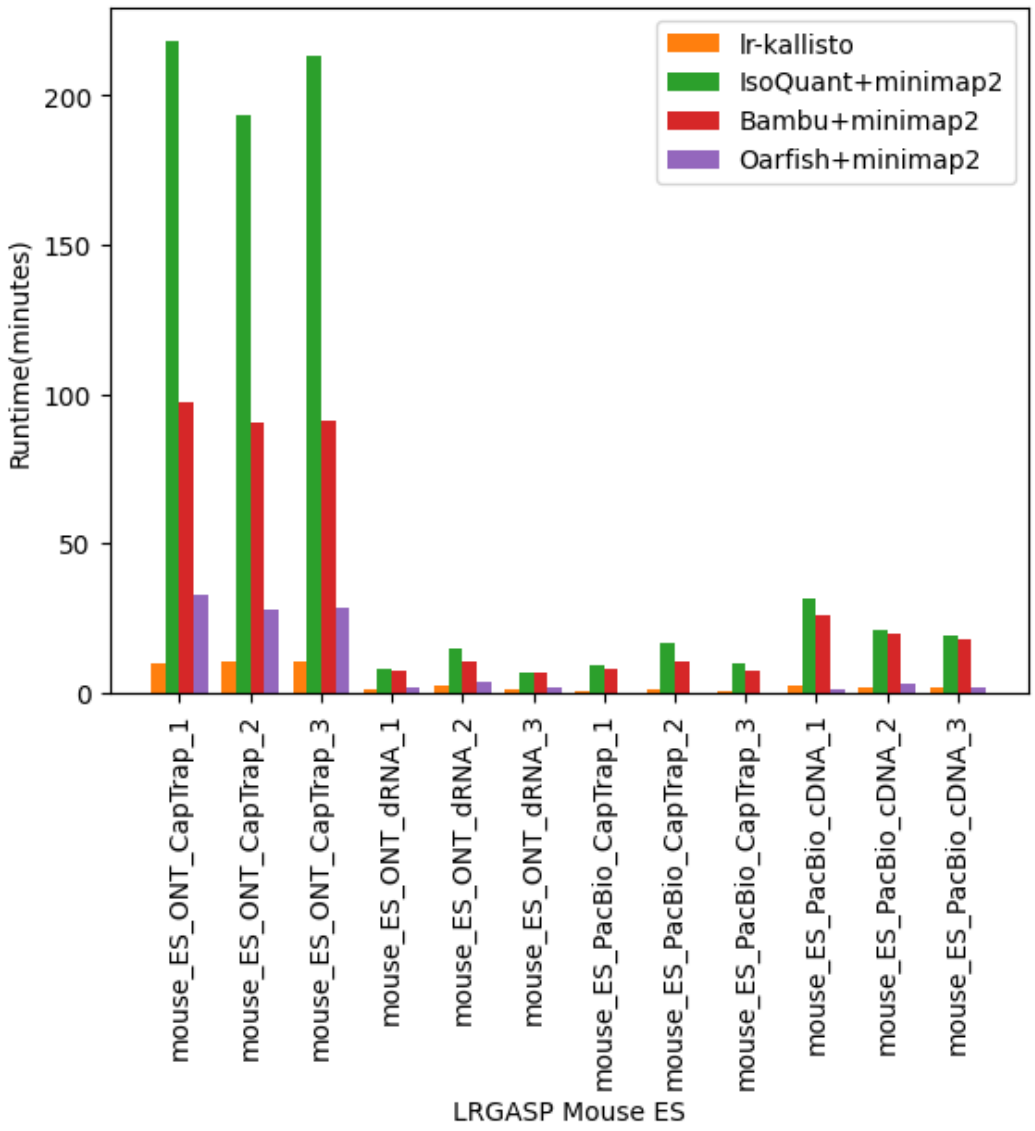


Figure 2.5: Runtime performance comparisons for lr-kallisto, IsoQuant, Bambu, and Oarfish.

with non-exome capture expression greater than 150% of exome capture expression, which combined with the difference in 3'-end and randomly primed reads highlights the bias of 3'-end reads, neglecting the non-trivial existence of functional internally priming transcripts.

The lr-kallisto quantification results are corroborated when comparing its performance to other tools on previously published data that is less deeply sequenced. In a comparison of Illumina and ONT on the HCT116 cancer cell line dataset generated by SG-NEx ((Chen, Davidson, et al., 2021)), we found that lr-kallisto could accu-

Tool	Max RSS	hh:mm:ss	% Aligned	% Unique	Total # of Reads
lr-kallisto (d-list, pseudobam)	1.8 Gb	1:30:19	56.0%	19.6%	105,591,654
lr-kallisto (no d-list, pseudobam)	5.3 Gb	2:50:46	75.7%	28.4%	105,591,654
lr-kallisto quant-only (d-list)	1.8 Gb	0:02:03	—	—	105,591,654
lr-kallisto quant-only (no d-list)	5.3 Gb	0:01:47	—	—	105,591,654
IsoQuant (post minimap2)	≥84.3 Gb	8:59:12	—	—	105,591,654
bambu (post minimap2)	95.1 Gb	3:37:11	—	—	105,591,654
minimap2+oarfish	17.9 Gb	5:09:36	—	—	105,591,654
minimap2 (transcriptome)	5.3 Gb	4:29:15	90.3%	17.1%	105,591,654
oarfish (post minimap2)	17.9 Gb	0:40:21	—	—	105,591,654
kallisto (Illumina)	—	—	64.2%	15.7%	158,034,313
kallisto (Illumina, nac)	—	—	92.4%	29.1%	158,034,313

Table 2.1: Comparison of tools on memory usage, runtime, alignment rate, and uniquely aligned reads.

rately quantify isoform level expression, in performance comparisons constituting two replicates of direct cDNA and direct RNA (Figure 2.9). The CCC performance of lr-kallisto exceeded that of Oarfish, evaluated on this data in ((Jousheghani and Patro, 2024)). Spearman correlations were lower overall in this dataset, indicating poor data quality, perhaps due to the lower coverage and higher sequencing error rate. We also compared lr-kallisto's performance on direct RNA to direct cDNA (Figure 2.10). We also found better performance with direct cDNA versus direct dRNA in replicate 4, and hypothesize that this is likely due to ~4 times the depth of coverage for replicate 4 in direct cDNA (7,656,893 reads) vs the direct RNA replicate 4 (1,896,643 reads), whereas replicate 3 direct cDNA (873,077 reads) vs direct RNA (1,185,183 reads) does not have the increased depth of coverage. We also compared lr-kallisto to Bambu, IsoQuant, and Oarfish on a previously sequenced

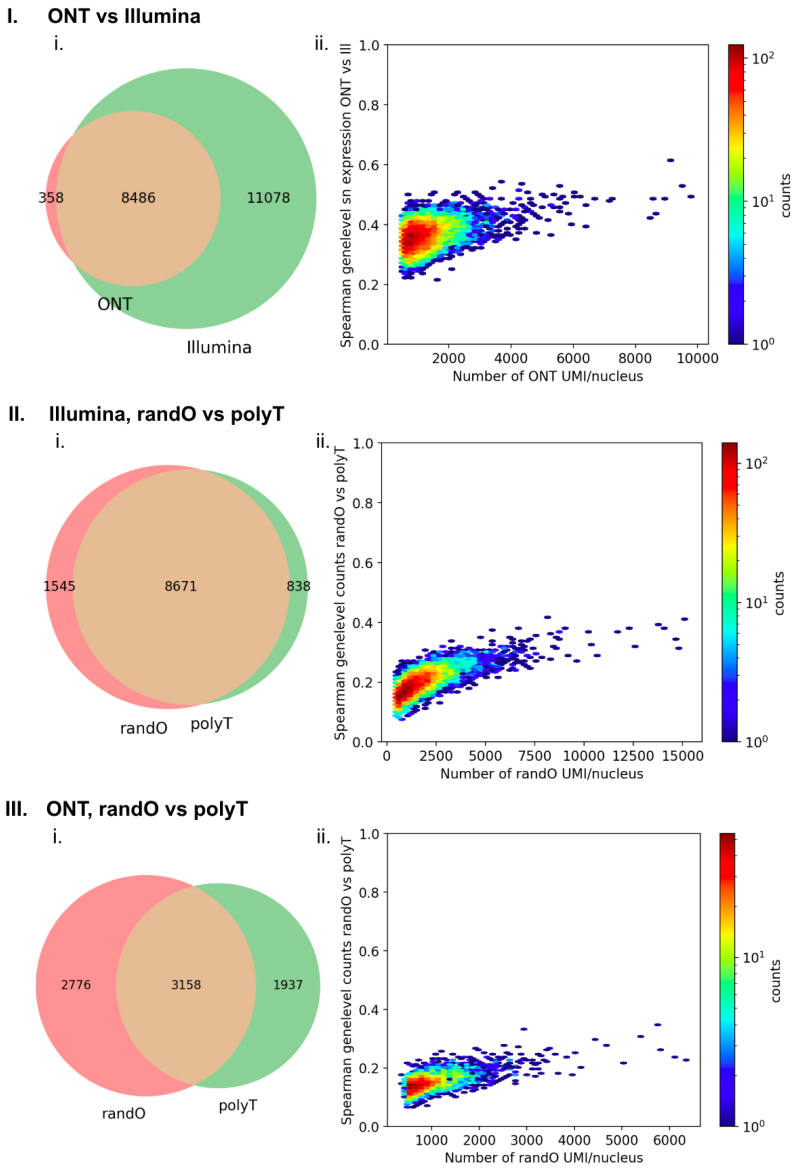


Figure 2.6: Barcode calling analysis and single-nuclei quantifications between Illumina vs ONT. I. i. Venn diagram of barcodes in ONT and Illumina. ii. Number of ONT UMI/nucleus vs Spearman correlation between ONT and Illumina single-nucleus gene level counts. II. i. Venn diagram of barcodes in Illumina random oligo (randO) and Illumina poly dT. ii. Number of randO UMI/nucleus vs Spearman correlation between Illumina randO and Illumina polydT single-nucleus gene level counts. III. i. Venn diagram of barcodes in ONT random oligo (randO) and ONT poly dT. ii. Number of randO UMI/nucleus vs Spearman correlation between ONT randO and ONT polydT single-nucleus gene level counts.

mouse cortex PacBio dataset (Figure 2.11). On this dataset ((Leung et al., 2021; Castanho et al., 2020)), which has an error rate of 12.4% (see Methods) and a differ-

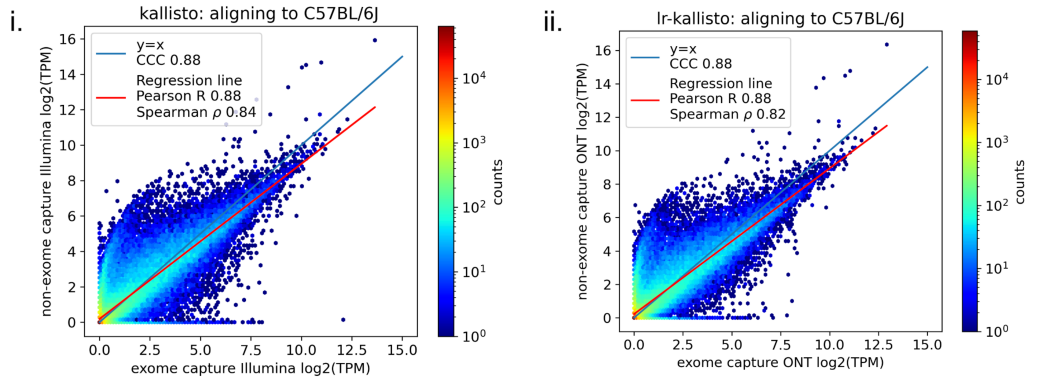


Figure 2.7: Contrast of non-exome vs exome capture in Illumina and ONT.

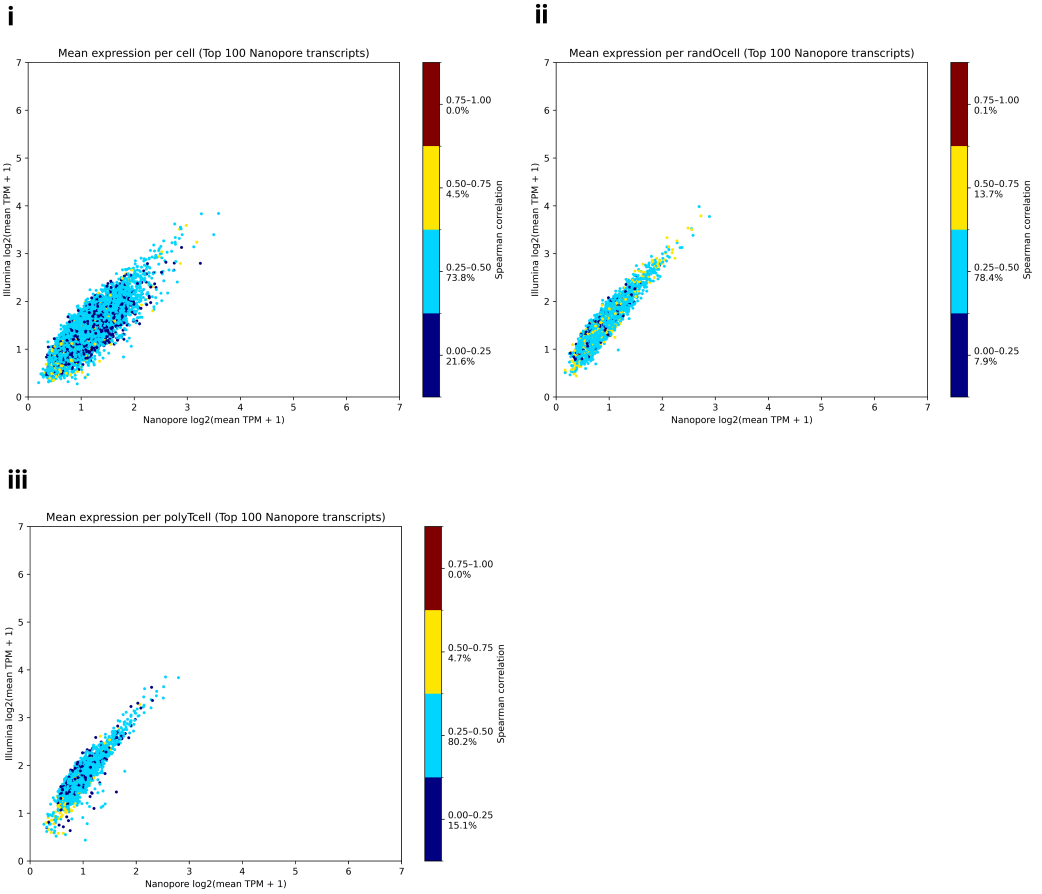


Figure 2.8: Contrast of priming methods in exome capture in Illumina vs. ONT.

ent error profile with errors more uniformly distributed along transcripts, we found similar performance between programs with lr-kallisto slightly outperforming other

tools in CCC. Further, since we use kallisto for the quantification of the short read Illumina data across these benchmarks, we highlight that kallisto, salmon, and other short read quantification tools are highly concordant. We demonstrate this with this dataset ((Leung et al., 2021; Castanho et al., 2020)) as we are able to perform analogous mapping and quantification with kallisto and salmon on the paired-end Illumina data. Here we observe a slightly higher CCC (.822 vs .81, .797 vs .78, .801 vs .80) in lr-kallisto, IsoQuant, and bambu, respectively and a slightly lower CCC (.806 vs .81) in Oarfish when comparing PacBio transcript expressions to Illumina transcript expressions with kallisto vs with salmon (Figure 2.11). However, with Parse single-nuclei short read data, there is not a truly analogous processing in salmon for the pseudoalignment and quantification. Since standard single-end bulk data significantly differs from Parse single-nuclei short read data, as it uses a combination of polyA priming and random hexamer priming. Therefore, with kallisto, we pseudobulked the counts without length normalization, which would not be appropriate for this priming chemistry, yielding comparable quantifications to the single-nuclei quantifications as this strategy avoids adding methodological bias in alignment, counting, and disambiguating multi-mapping reads.

We benchmarked lr-kallisto's stability and robustness compared to other long-read quantification tools across species, platforms, and protocols, by evaluating lr-kallisto's performance, along with Bambu, IsoQuant, and Oarfish using the LR-GASP's challenge 2 benchmark (Pardo-Palacios et al., 2023) of long-read quantification tools (Figure 2.12a-b). For our benchmarking, we chose to focus on the Mouse ES data, as it had lower sequencing error rates across 3 out of the 4 protocol/platform combinations, thereby serving as the closer proxy for current long-read data. Further, we also used LRGASP Human WTC11 to enable an analysis of lr-kallisto's relative performance on SIRV Set 4, which provides a benchmarking of ability to quantify complex and difficult transcript sets. The SIRV-Set 4 synthetic transcripts provide a useful control. This set includes the SIRV isoforms, the long SIRV isoforms, and the External RNA Controls Consortium (ERCC) transcripts. The SIRV-Set 4 isoforms include seven genes (SIRV1 through SIRV7) comprised of 69 fabricated human isoforms. SIRV1 through SIRV7 transcript isoforms were constructed to include mono-exon transcripts, single- and multi-exon skipping events, alternative start/stop sites, as well as antisense transcripts. ERCC transcripts range from 250 to 2000 bp in length, mimicking natural eukaryotic mRNAs. The long SIRVs (SIRV10 through SIRV12) includes 15 RNA transcripts of 4000 to 12,000 bp. All together this synthetic control provides a helpful benchmark across tools

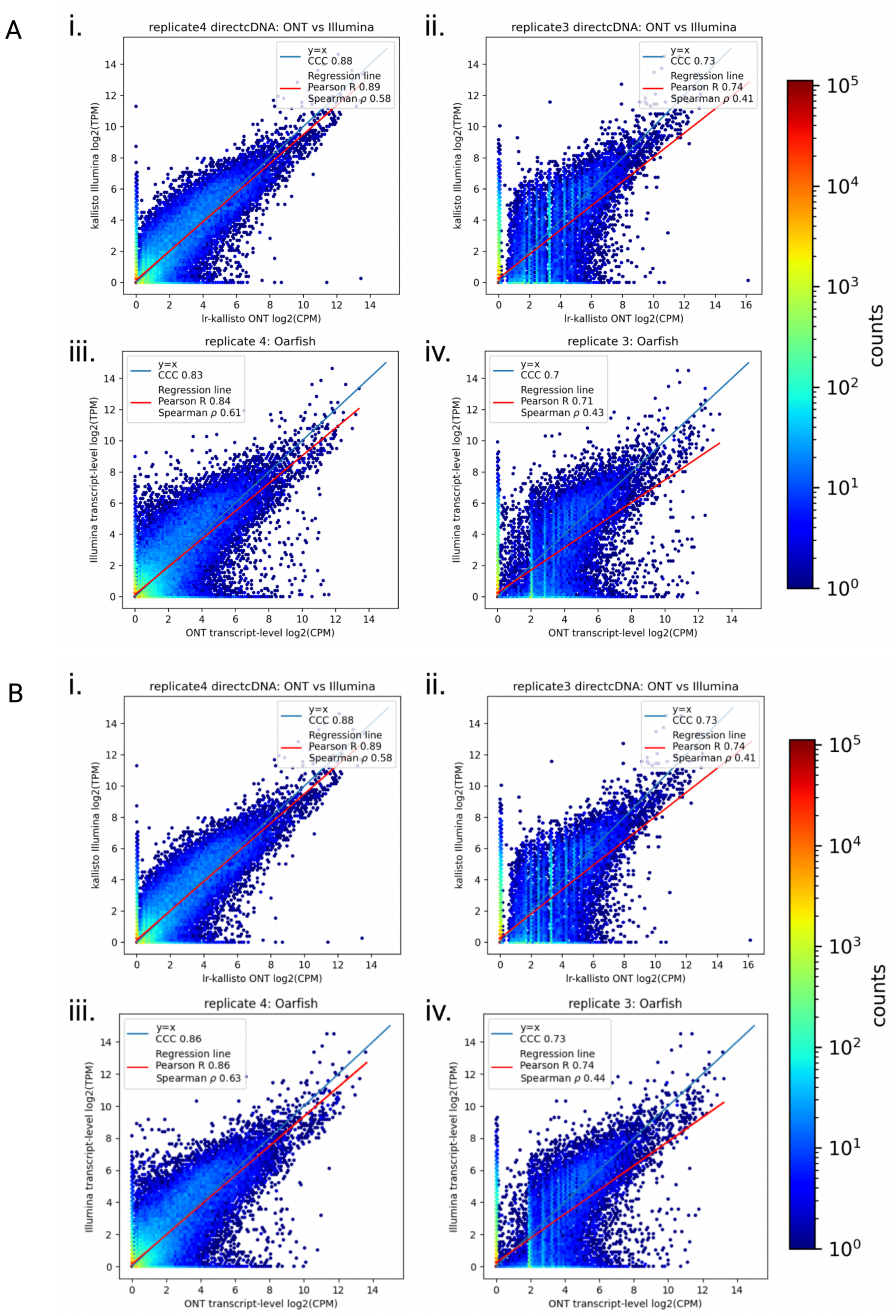


Figure 2.9: Performance of lr-kallisto on ONT sequenced direct cDNA libraries from the HCT116 cell line, where panel A is with Oarfish v0.3.1 and panel B is with Oarfish v0.5.1.

accuracy and sensitivity across transcript complexity within PacBio and ONT sequencing protocols and platforms. We found that Bambu, IsoQuant, lr-kallisto, and Oarfish all achieved reasonably high CCCs between replicates, both with respect to abundance estimates (Figure 2.12a), and variability between isoforms (Figure

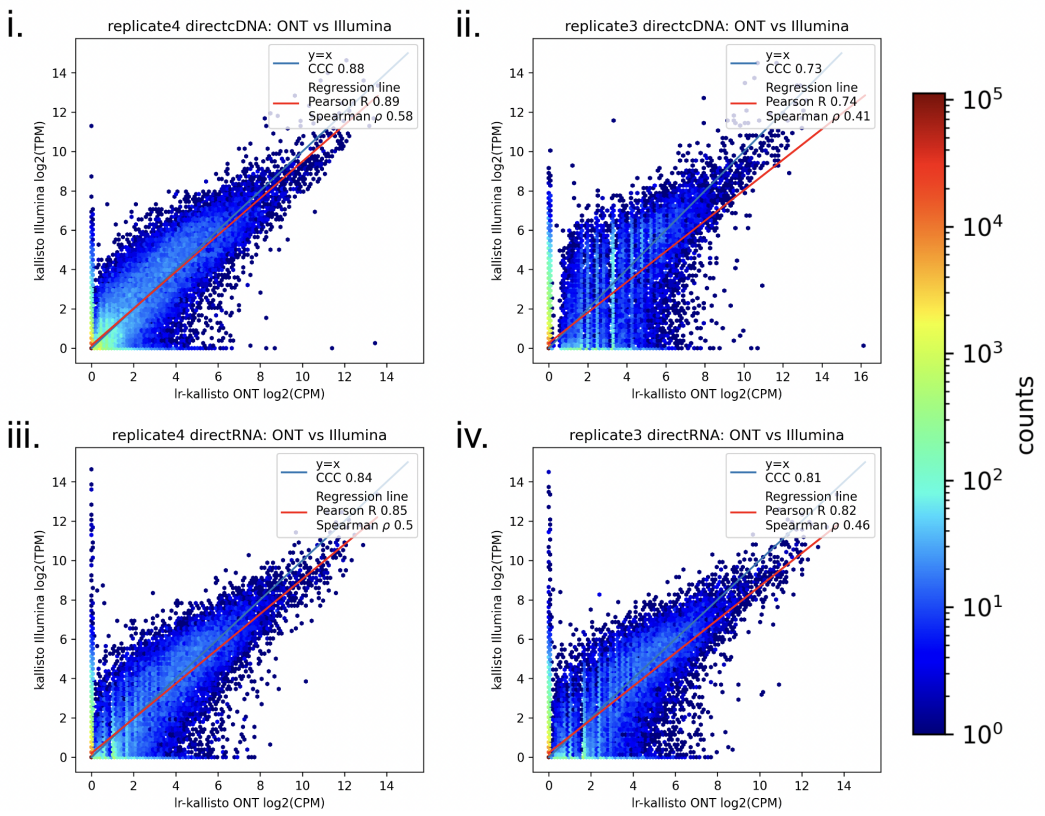


Figure 2.10: Comparison of lr-kallisto on ONT sequenced HCT116 cell line libraries generated with directRNA and direct cDNA between two replicates in each.

2.12b). With SIRV Set 4 analysis, we show both lr-kallisto and Oarfish lag in median relative difference (MRD) and outstrip across all platforms and protocols in Spearman Correlation Coefficient (SCC) compared to IsoQuant and Bambu, while for Normalized Root Mean Squared Error (NRMSE) lr-kallisto and Oarfish outperform IsoQuant and Bambu in PacBio, while underperforming IsoQuant in ONT (Figure 2.12c). Moreover, lr-kallisto and Oarfish are both high performing in the metrics of percent expressed transcripts (PET) across all three categories of transcript sets (SIRV, SIRV long transcripts, and ERCC), indicating higher detection accuracy even at low expression rates and for complex and long transcripts (Figure 2.12d). Furthermore, in a sequencing error free simulation with uniform expression of SIRV Set 4, we found that lr-kallisto detected all isoforms (with the exception of SIRV311, one of the mono-exonic isoforms, while maintaining detection of the other three mono-exonic isoforms: SIRV206, SIRV512, SIRV618) and was perfectly accurate in quantifying 88% of SIRV Set 4 isoforms. For completeness, we also compared the performance of lr-kallisto to Bambu, IsoQuant, Oarfish using the

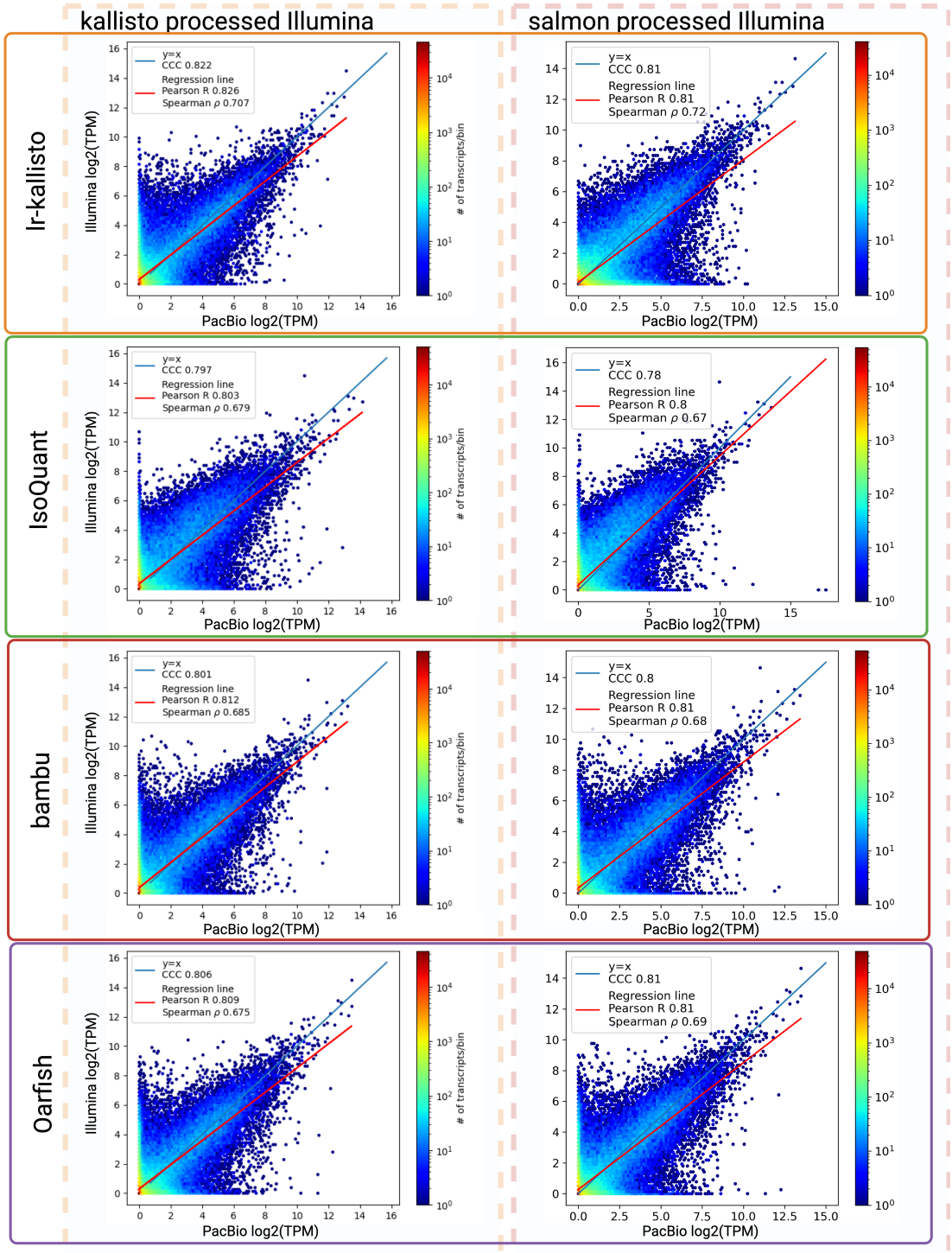


Figure 2.11: Evaluation of Bambu, IsoQuant, lr-kallisto, and Oarfish on mouse cortex high-depth PacBio data.

metrics of the LRGASP paper (Figure 2.13). Resolution Entropy (RE) is a measure of how well a tool uniformly quantifies at different expression levels. Irreproducibil-

ity Measure (IM) is a measure of how reproducibly the tool quantifies expression across replicates, i.e., whether the coefficient of variation between replicates is low. Consistency Measure (CM) is a measure of how consistent the tool is at detecting expressed transcripts, assuming that transcripts should be expressed simultaneously across replicates, and ACVC is the Area under the Coefficient of Variation Curve, which again assumes that for a given mean expression level across replicates the coefficient of variation should be low. We found that lr-kallisto performs as well as other programs on these stability and robustness measures. The variability that we found in quantifications of replicates can be explained by variable depth of sequencing between the replicates and between the protocols and platforms (Pardo-Palacios et al., 2023). The notable difference in ONT cDNA is due in part to a sequencing error rate of ∼12%, which is characteristic of data obtained in earlier ONT platform versions (Goodwin et al., 2015).

We assessed the performance of lr-kallisto using simulated data across a range of sequencing error profiles, and compared with results on the same simulated data for five other widely used or recently published programs. We used simulations generated by (Prjibelski et al., 2023) who used the IsoSeqSim simulator (see Data and Code Availability) to generate PacBio reads (6 million *Mus musculus* reads with ∼1.6% sequencing error rate), and NanoSim (Yang et al., 2017) to generate ONT.R10.4 reads (30 million *Mus musculus* reads with ∼2.8% sequencing error rate). The PacBio IsoSeqSim Simulation (Figure 2.14a) demonstrates lr-kallisto's high accuracy compared to the currently leading benchmarked long-read quantification tools Bambu, IsoQuant, and Oarfish, with lr-kallisto achieving a CCC of 0.99, vs 0.90, 0.91, and 0.99, respectively (Pardo-Palacios et al., 2023; Dong et al., 2023). Furthermore, in the ONT NanoSim R10.4 Simulation (Figure 2.14), lr-kallisto ties for the highest CCC of 0.97, vs 0.88 and 0.91, respectively.

We performed additional comparative evaluations of Bambu, IsoQuant, lr-kallisto, and Oarfish on a more extensive set of simulations to understand the strengths and weaknesses these tools when confronted with different sequencing error challenges (Figure 2.15). We found that lr-kallisto and IsoQuant were both robust to indel and substitution profiles simulated to match PacBio sequencing data and uniformly distributed. IsoQuant was also robust to uniformly distributed sequencing errors with indel and substitution profiles matched to ONT, whereas lr-kallisto performance degraded at higher ONT error rates in this simulation (Figure 2.15a). In particular, this highlights lr-kallisto's sensitivity to the unrealistic combination of uniform

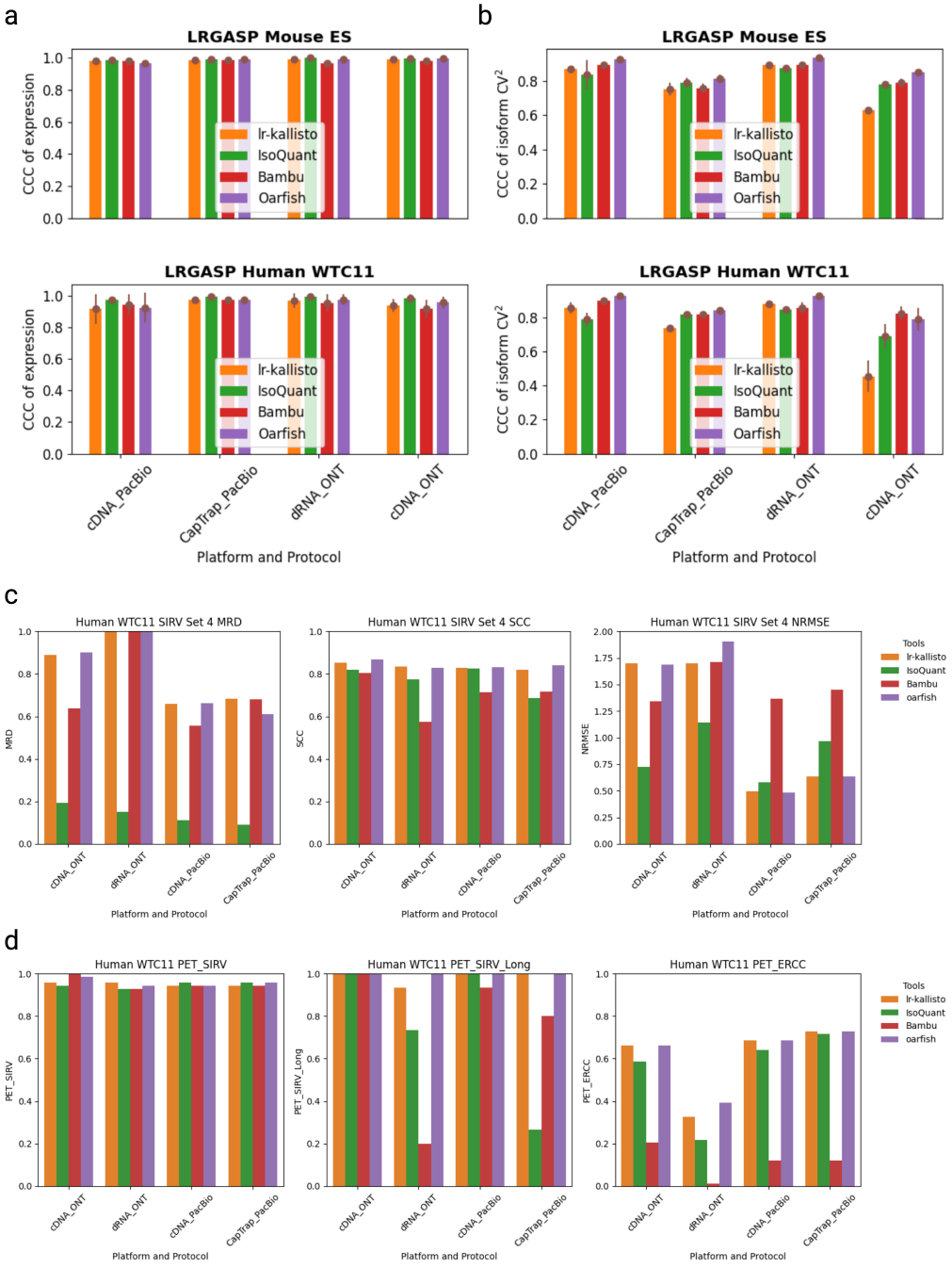


Figure 2.12: Comparison of Bambu, IsoQuant, lr-kallisto, and Oarfish on LRGASP data. (a) abundance estimates as measured by CCC of expression and (b) variability between isoforms as measured by CCC of isoform CV^2, with 90% CI to measure consistency and reproducibility among replicates between the tools.

sequencing error distribution and higher rate of insertion errors in ONT versus PacBio.

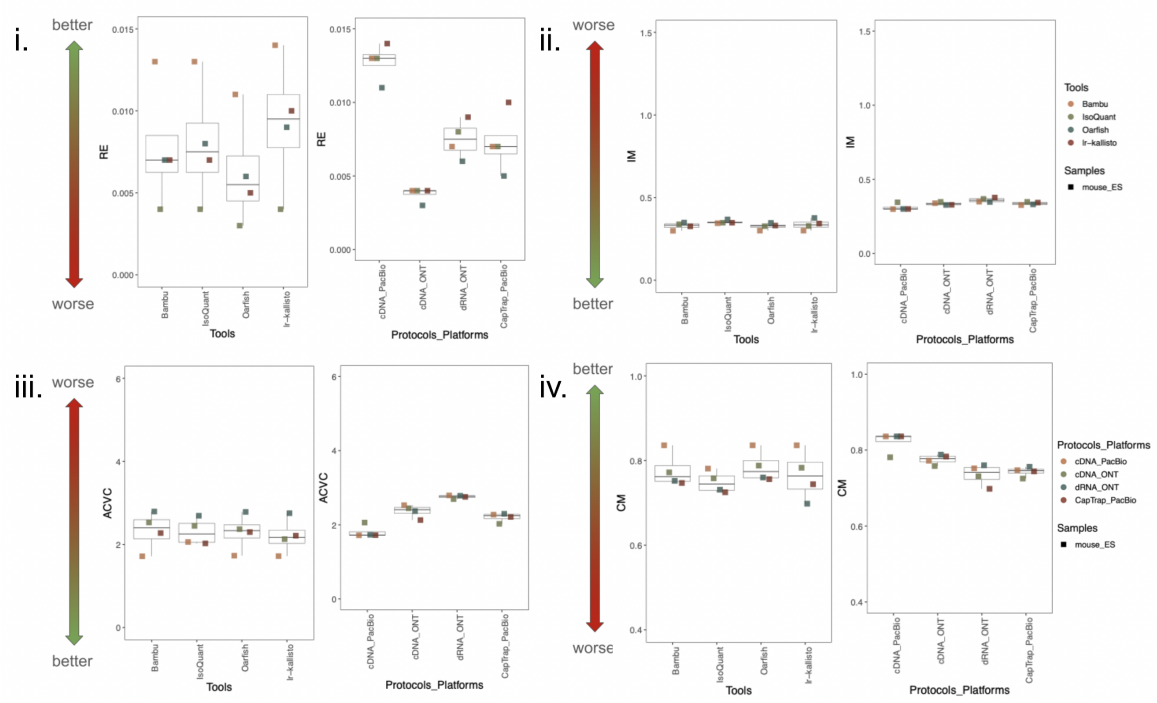


Figure 2.13: Evaluation of lr-kallisto, Bambu, IsoQuant and Oarfish according to LRGASP challenge 2 metrics in Mouse ES cells.

In another ONT simulation generated with NanoSim to produce reads with an 11.2% error rate (see Data and Code Availability), lr-kallisto achieved a CCC of 0.31 on all transcripts, outperforming IsoQuant (CCC = 0.28), and underperforming Bambu (CCC = 0.51), and Oarfish (CCC = 0.55) (Figure 2.15b). This was also the case at a higher error rate (15.2%), with lr-kallisto continuing to outperform IsoQuant and underperform Bambu and Oarfish (Bambu CCC = 0.53, IsoQuant CCC = 0.32, lr-kallisto CCC = 0.34, Oarfish CCC = 0.58) (Figure 2.15c).

The performance of lr-kallisto benefits from quantification with respect to a de Bruijn graph (Hjörleifsson et al., 2022). We tested whether and to what extent changing the *k*-mer length default in lr-kallisto to 63 bp long vs 31 bp long in the reference transcriptome de Bruijn graph creates a less connected and less complex structure (Figure 2.16). In this example, of the Pax2 gene, we find that a change of *k*-mer length simplifies the T-DBG with the reduction of a single node and 2 edges. However, when we scale this out to just the first 1000 transcripts listed in the LRGASP basic gencode human annotation, we found a reduction from 3,698 nodes using the 31 *k*-mer T-DBG to 2,708 nodes using the 63 *k*-mer T-DBG and 4,687 edges to 3,238 edges, respectively. Furthermore, the largest connected T-

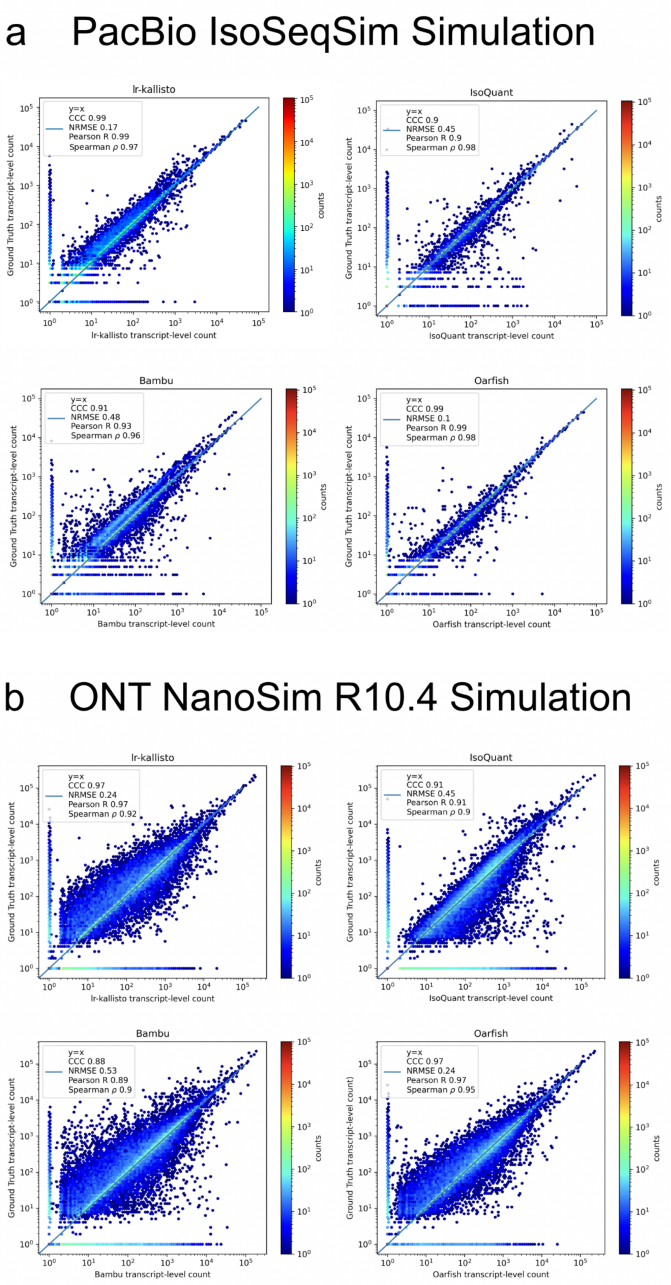


Figure 2.14: lr-kallisto is highly accurate in simulations with error up to ~3%. A comparison of performance of Bambu, IsoQuant, lr-kallisto, and Oarfish on PacBio (top) and ONT (bottom) simulations with Concordance Correlation Coefficient (CCC), Normalized Root Mean Squared Error, and Pearson's and Spearman's correlation coefficients reported.

DBG graph component in the 63 *k*-mer T-DBG is composed of 12.59% of the bp vs 65.90% in the 31 *k*-mer T-DBG. We believe that the selection of higher quality,

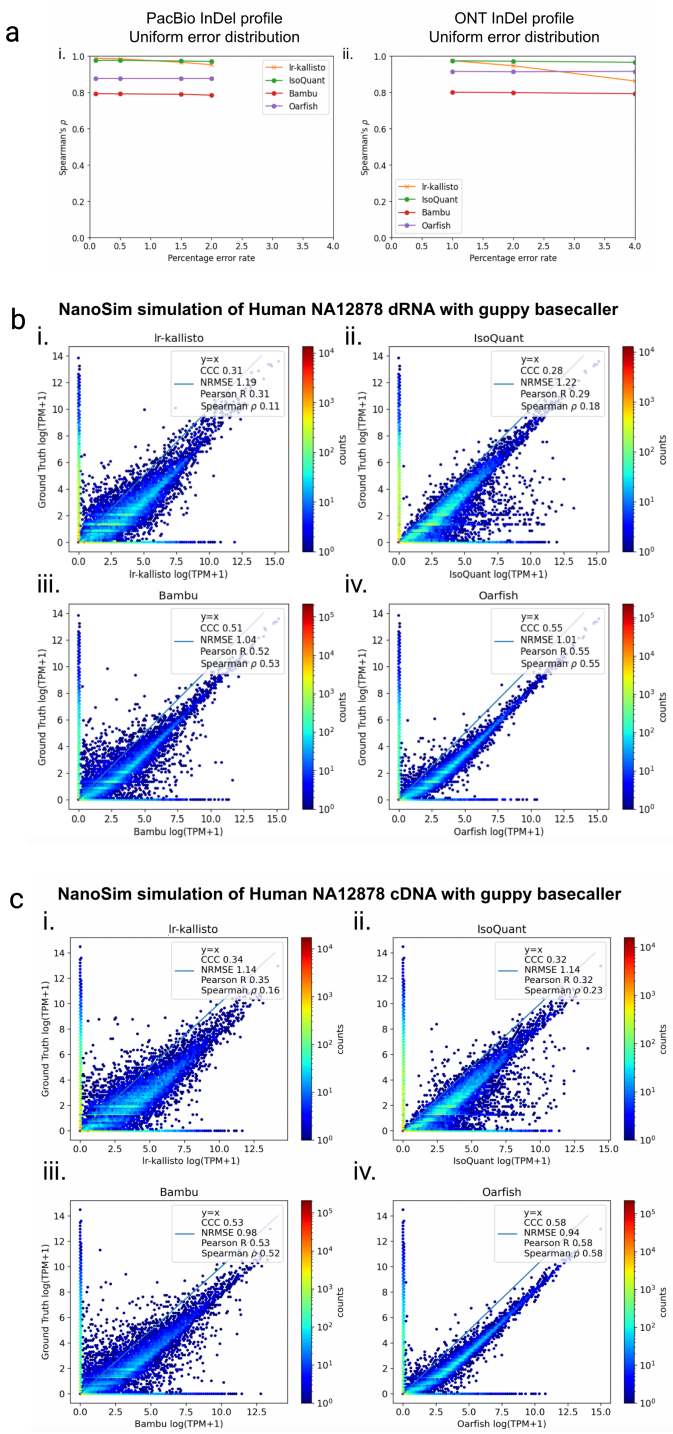


Figure 2.15: Extended benchmarks on simulations. a) Benchmarks of Bambu, IsoQuant, lr-kallisto, and Oarfish on simulations with a range of error parameters. b) Performance on all annotated transcripts at ONT 11.2% sequencing error rate. c) Performance on all annotated transcripts at ONT 15.2% sequencing error rate.

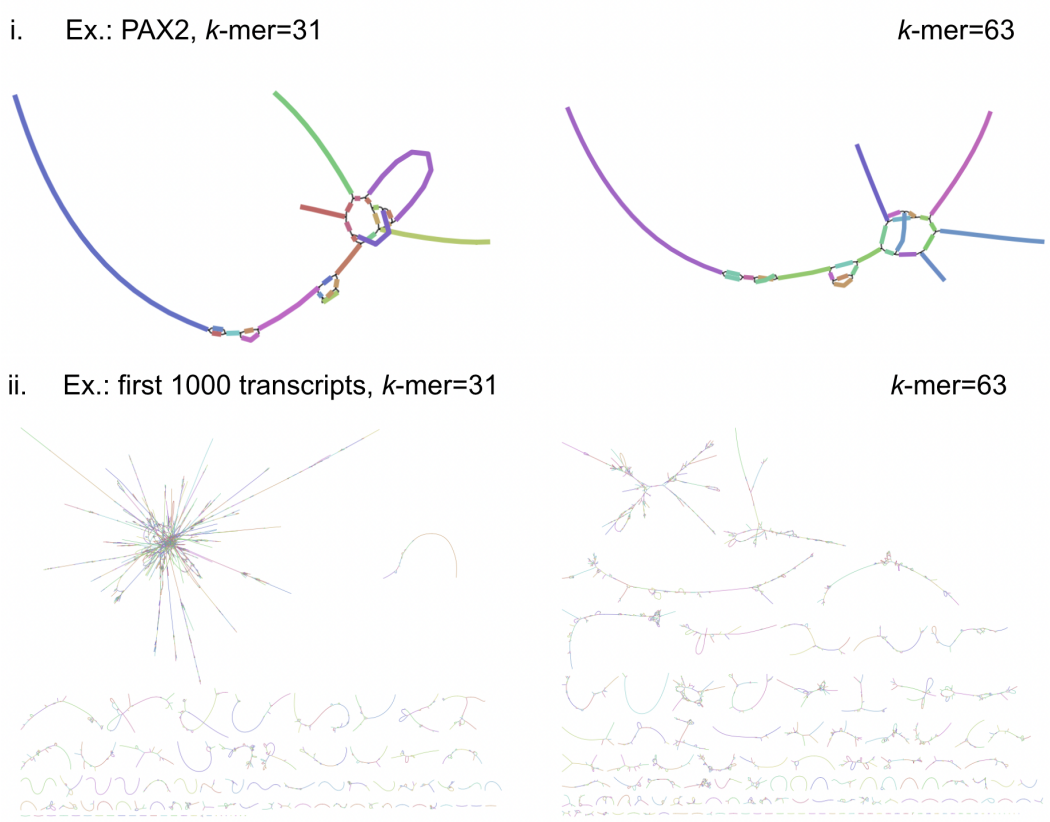


Figure 2.16: lr-kallisto transcript de Bruijn Graph bandage plots.

low sequencing error regions from the reads by the 63 *k*-mer T-DBG, combined increasing the probability of uniquely mapping, or at the very least mapping to a transcript compatibility class with less transcripts, is producing more accurate and more efficient pseudoalignment.

2.4 Discussion

With Oxford Nanopore sequencing becoming more accessible due to low entry costs and reduced sequencing error rate (Bloomfield et al., 2024), long-read sequencing is advancing our ability to decipher the complexity of transcriptomes. Increasing throughput now makes it possible to not only perform discovery with long-read sequencing, but also to accurately quantify transcript abundances, and we have shown that results comparable to short-read sequencing can be achieved at reasonable cost with exome capture, and with high accuracy quantification using lr-kallisto. Exome capture will be especially helpful for filtering out intronic reads that would be otherwise sequenced in (single-)nucleus data, as nuclei are replete with intron lariats and partially processed transcripts. lr-kallisto is highly accurate in producing quantifi-

cation results on data with less than 10% sequencing error rate comparable to those with short-read sequencing. This makes lr-kallisto immediately useful for current long-read sequencing transcriptome projects, although performance will not be as good on legacy higher error long-read sequencing datasets. While even standard kallisto is now useful and competitive with current long read tools for quantification on the high accuracy (>99.5%) long reads, demonstrating the suitability of pseudoalignment to long reads, lr-kallisto eclipses kallisto in accuracy performance and alignment rate.

Furthermore, as described in Methods, lr-kallisto is useful for long-read sequencing of single-cell and single-nucleus RNA-seq libraries when coupled with tools designed for barcode discovery (Sullivan and Pachter, 2024; Cheng et al., 2024). Furthermore, lr-kallisto is compatible with translated pseudoalignment, which can be useful for detection of viruses (Luebbert et al., 2023).

Finally, in this work we have focused on quantification. However, lr-kallisto can also be used, in principle, for transcript discovery. In particular, reads that do not pseudoalign with lr-kallisto can be assembled to construct contigs from unannotated, or incompletely annotated, transcripts. While we have not completed our investigation and benchmarking of this approach, the pseudoalignment algorithm and distinguishing flanking *k*-mers combine to allow filtering of unmapped reads that do not fit within the annotated model set of transcripts.

2.5 Methods

lr-kallisto

Many approaches have been applied to RNA-seq quantification from classical alignment approaches to pseudoalignment paired with likelihoods and expectation-maximization (EM). Due to its speed, efficiency, and accuracy, pseudoalignment with likelihoods and EM has been widely adopted for the mapping of short read RNA-seq. However, for long-read RNA-seq, minimap2 has become the standard for aligning long-reads. Minimap2 follows the standard genome alignment methodology of seed-chain-align (H. Li, 2018). It creates a reference index in the form of hashing minimizers into keys for a reference hash table storing the list of genomic/transcriptomic locations of the minimizer. For each read, minimap2 uses read minimizers as seeds matching these to the reference hash table and identifies the optimal collinear chain(s) of matches. While this method is accurate and has been developed to be highly efficient for the alignment strategy used, it is still time

and resource expensive with high memory storage demands.

lr-kallisto, building on the existing framework of kallisto and adapting the pseudoalignment and expectation-maximization algorithm for long-reads, gives an accurate, fast, and low resource solution for mapping long-reads (Figure 2.17). The main technical challenge of long-reads lies in the higher sequencing error rates, though others include the differing rates of substitutions, deletions, and insertions between long-read sequencing technologies, sequencing length, repetitive regions, and concatemers. To address the challenge of higher sequencing error, different methods, including minimap2 (H. Li, 2018), uLTRA (Sahlin and Mäkinen, 2021), and STAR (Dobin et al., 2013) have utilized various approaches to long-read alignment. Minimap2 uses a small k-mer size of 14 and 15 for long-reads, while uLTRA employs a two-pass chaining algorithm to improve alignment accuracy. Strobemers have been suggested using fuzzy k-mers that allow error tolerance (Sahlin, 2021). In lr-kallisto, we, instead, propose a long k-mer length and "chaining" pseudoalignment for addressing the challenges of long-read alignment.

We must address two points: first, that sequencing length and long-read sequencing error rates require a different algorithmic approach to pseudoalignment and, second, the handling of length bias which differs from that of short reads. To address the first, we propose the following algorithm for pseudoalignment and the change of index k-mer length to 63, which we discuss after describing the algorithm. Both of these changes take into consideration the sequencing error rate and repetitive regions across genes. While this idea is not a direct implementation of the chaining described in (H. Li, 2018), it can be understood in a similar way. Within kallisto's pseudoalignment, a read's transcript compatibility class is determined. For short reads, this is accomplished with a strategy that increases efficiency by checking the transcript compatibility class for the first, middle, and end of k-mers in the read if the distance to the end of the contig is longer than the read or the first, middle, and end k-mers of the read within the region that is consistent with the contig in the transcriptome de Bruijn graph (T-DBG) (to ensure that the read is consistent with the T-DBG junctions) and then proceeds to the next contig in the read. If they are all the same, these are the only k-mers checked, while if they differ a more iterative approach is taken. We then take the intersection of these transcript compatibility classes. Whereas, in lr-kallisto, if the intersection of transcript compatibility classes (TCCs) a read maps to is empty, we instead take the most often occurring TCC. Moreover, if at least one k-mer maps uniquely to a transcript, then we take the most

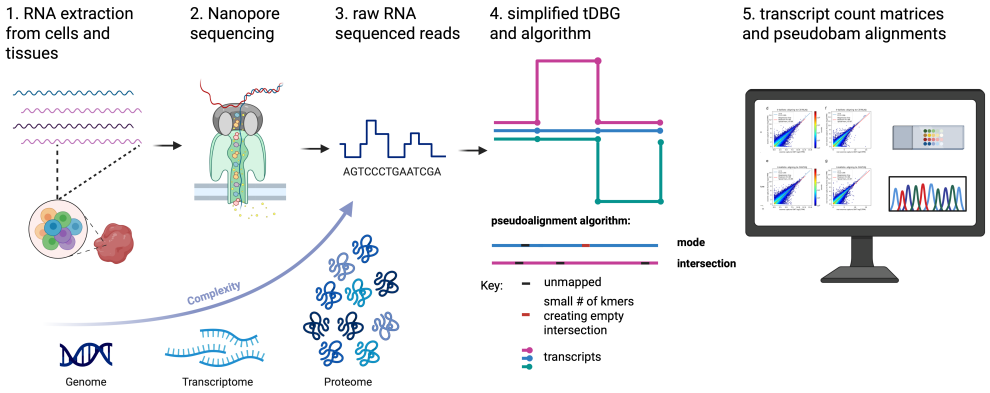


Figure 2.17: Overview of biosample to lr-kallisto workflow for long read RNA sequencing. To study the complexity of life, we can study the genome, transcriptome, and proteome. Through long read sequencing, we can achieve greater insight into both the workings of the genome and the proteome at the individual level and even the functionality of RNA as a molecule. Therefore, improving our ability to analyze long read RNA sequences increases our understanding of biology itself. 1. RNA is extracted from cells and tissues in either single-cell, single-nuclei, or bulk preparation of RNA creating an RNA sequencing library. 2. The RNA sequencing library is then sequenced with either PacBio or Oxford Nanopore Sequencing (Nanopore illustration shown). 3. The raw electrical signal from the nanopore or the raw fluorescent signal from PacBio is then basecalled to create the raw RNA sequenced reads. 4. The raw RNA sequenced reads are input to lr-kallisto outputting both transcriptome quantification of the tissue or single- cells or nuclei as well as the pseudobam alignments for the reads. 5. The analysis and visualization of lr-kallisto's outputs: single-cell or bulk transcript and gene count matrices and pseudobam (pseudoalignments are output in bam format). Created with `https://BioRender.com`

often occurring TCC among mapping k-mers that are uniquely mapping to a single transcript. If there are two uniquely mapping regions of the same length within a read to two distinct transcripts, then the read is mapped to the TCC of the first occurring transript in the transcriptome. In the case of the intersection, the intersection can directly be interpreted as the transcript or set of transcripts that the read has the longest combined stretches of compatibility with, since the intersection takes the subset of transcripts that coexist between all k-mers with compatible transcripts. However, the intersection may be empty in the case of a variant or error creating an isolated stretch of compatibility with a disjoint transcript compatibility class. Furthermore, in the case that the intersection is empty and the algorithm switches to using the mode of transcript compatibility classes with threshold one, the calculated mode across all transcript compatibility classes that k-mers in the read mapped to is

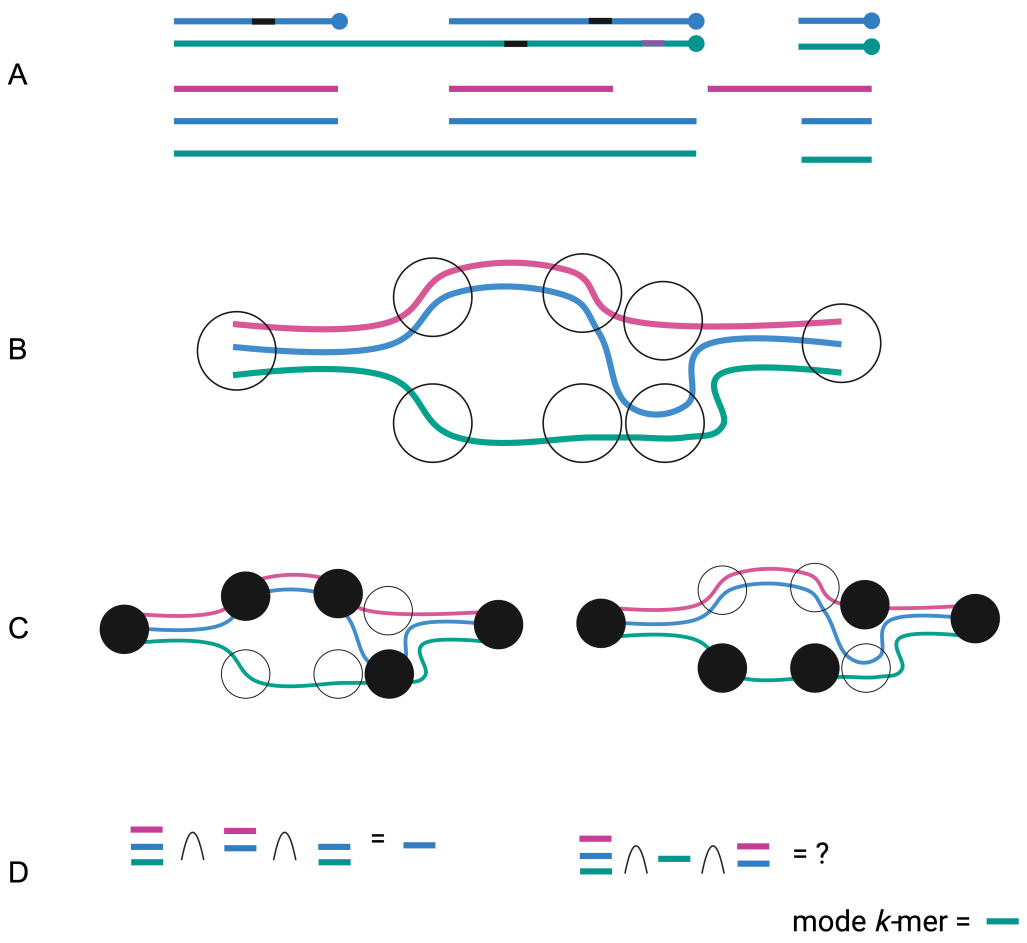


Figure 2.18: Overview of lr-kallisto pseudoalignment algorithm. The input consists of a reference transcriptome and reads from a long read RNA sequencing experiment. (A) An example of two reads (blue and green with unmapping regions (black) and erroneously mapped regions (purple)) and three (pink, blue, and green) overlapping transcripts. (B) An index is constructed by creating the transcriptome de Bruijn Graph (T-DBG) where nodes are k-mers, each transcript corresponds to a colored path as shown and the path cover of the transcriptome induces transcript compatibility class (TCC) for each k-mer. (C) Conceptually, the k-mers of a read are hashed (black nodes) to find the TCC of a read. (D) The TCC of the read is determined by taking the intersection of the transcript compatibility classes of its constituent k-mers, if it exists; otherwise, the mode of the TCCs of the k-mers of the read is taken. Created with `https://BioRender.com`

the transcript or set of transcripts that again is the "longest chain" of compatibility.

The change of k-mer length to 63 was based on empirical evidence showing improved

performance over the standard k-mer length of 31 for short reads. We found that across long-read technologies and simulations there was an improvement in metrics of Normalized Root Mean Squared Error and Spearman's correlation between lr-kallisto quantifications and the simulation's ground truth (Figure 2.14-2.15). In real data (both PacBio and ONT), we observed an increased rate of alignment of reads with a longer k-mer length for PacBio sequencing error rate less than 2% and for ONT sequencing error rate less than 10%. Moreover, the longer k-mer length improves the quality of mapping k-mers making it more probable that the read originates from the transcript compatibility class it maps to. As k increases, the number of distinct k-mers also increases, but the number of contigs decreases. This implies that the number of transcripts in a transcript compatibility class decreases on average with increasing length of k. Overall, the complexity of the T-DBG decreases (Figure 2.16), increasing the probability of the read originating from the transcript compatibility class it is mapping to. Furthermore, this also increases the probability of the intersection of equivalence classes being nonempty, which increases the overall mapping rate.

To address the second point, we adapted the effective length, l_e, within kallisto to be transcript specific, i.e., defining the effective transcript length, l_{e_t} for a transcript t to be:

$$l_{e_t} = \frac{\sum l_{r_t}}{\sum 1_{r_t}} - k$$

where k is k-mer length, l_{r_t} is the length of a read aligning to transcript t, and 1_{r_t} is the boolean function that returns 1 if a read aligns to t and 0, otherwise.

We use the first 1 million aligning reads to compute these effective, transcript-specific lengths. The choice of 1 million reads is to be able to compute this expression for every transcript expressed in the data, which is achieved with unordered data and transriptomes of the size of humans and mice. We found that length normalization was effective at low sequencing error rates (< 2%) when sequencing error is uniform, providing a slight improvement in results, and was detrimental to performance at high sequencing error rates and in cases where sequencing error is non-uniform.

Finally, we implemented a change to the expectation-maximization (EM) algorithm for long-reads. In the default option, we initialize transcript abundances to a uniform distribution on the multi-mapping counts with the unique counts for each transcript added to the initialization of a transcript abundance. In the long-read option, we

first apportion multi-mapping reads using the EM algorithm starting with a uniform distribution of multi-mapping reads among those mapped to transcripts, and then post EM we add the uniquely mapping counts to each transcript. We found that the latter option works better for the PacBio InDel profile with uniform error in reads in simulations, but that it has reduced performance with real PacBio and ONT reads and simulations based on real data such as with NanoSim's simulations based on profiling of real data.

Mice and Tissue Collection

Mice were housed at the UC Irvine Transgenic Mouse Facility (TMF) in a temperature-controlled pathogen-free room under 12-hour light/dark cycles (lights on at 07:00 hr, off at 19:00 hr). The animal experiments were reviewed and approved by the Institutional Animal Care and Use Committee (IACUC), protocol AUP-21-106, "Mouse genomic variation at single cell resolution". Left cerebral cortex tissues of 10-week-old mice were harvested from 4 C57BL/6J and 4 CAST/EiJ (2 males and 2 females per genotype) between the hours of 09:00 to 13:00. Tissues were stored in 1 mL Bambanker media in cryotubes kept at -80°C until nuclei isolation.

Purification of Nuclei from Mouse Tissues

Tissues were thawed in Bambanker media on ice until the tissue could be extracted and lysed using Nuclei Extraction Buffer (Miltenyi Biotec cat. #130-128-024). Using forceps, tissues were transferred to a chilled gentle MACS C Tube (Miltenyi Biotec cat. #130-093-237) with 2 mL Nuclei Extraction Buffer supplemented with 0.2 U/µL RNase Inhibitor (New England Biolabs cat. M0314L). Nuclei were dissociated from whole tissue using a gentleMACS Octo Dissociator (Miltenyi Biotec cat. #130-095-937). The resulting suspension was filtered through a 70 µm MACS SmartStrainer then a 30 µm strainer (Miltenyi Biotec cat. #130-110-916 and #130-098-458, respectively). Nuclei were resuspended in 3 mL PBS + 7.5% BSA (Life Technologies cat. #15260037) and 0.2 U/µL RNase inhibitor for manual counting using a hemocytometer and DAPI stain (Thermo Fisher cat. #R37606).

Nuclei Fixation

After counting, 4 million nuclei per sample were fixed using Parse Biosciences' Nuclei Fixation Kit v2 (cat. #ECF2003), following the manufacturer's protocol. Briefly, nuclei were incubated in fixation solution for 10 minutes on ice, followed by permeabilization for 3 minutes on ice. The reaction was quenched, then nuclei

were centrifuged and resuspended in 300 µL Nuclei Buffer (Parse Biosciences cat. #ECF2003) for a final count. DMSO (Parse Biosciences cat. #ECF2003) was added before freezing fixed nuclei at -80°C in a Mr. Frosty (Sigma-Aldrich cat. #635639).

Split-Seq Experimental Protocol

Nuclei were barcoded using Parse Biosciences' WT Kit v2 (cat. #ECW02030), following the manufacturer's protocol. Fixed, frozen nuclei were thawed in a 37°C water bath and added to the Round 1 reverse transcription barcoding plate at 19,500 nuclei per well, with alternating columns in rows A and C containing C57BL/6J males and females and rows B and D containing CAST/EiJ males and females. In situ reverse transcription (RT) and annealing of barcode 1 + linker was performed using a thermocycler (Bio-Rad T100, cat. #1861096). After RT, nuclei were pooled and distributed in 96 wells of the Round 2 ligation barcoding plate for the in situ barcode 2 + linker ligation. After Round 2 ligation, nuclei were pooled and redistributed into 96 wells of the Round 3 ligation barcoding plate for the in situ barcode 3 + UMI + Illumina adapter ligation. Finally, nuclei were counted using a hemocytometer and distributed into 8 subpools of 13,000 nuclei. The nuclei in each subpool were lysed and cDNA was purified using AMPure XP beads (Beckman Coulter cat. #A63881), then the barcoded cDNA underwent template switching and amplification. Importantly, for two subpools ("13G" and "13H") we increased the number of PCR cycles to 13 cycles from 12, and increased the extension time from 3 minutes to 13 minutes in order to increase the yield of full-length barcoded cDNA. cDNA from one of the subpools ("13G") also received exome capture treatment using Parse Biosciences' Custom Gene Capture Kit (cat. #GCE1001) and a Mouse Exome Panel (Twist Bioscience, cat. #102036). 1 µg of cDNA was hybridized with a blocker solution to block repetitive sequences, then hybridized with the exome panel overnight. Captured molecules were purified using Streptavidin beads, then amplified again using the cDNA amplification reagents from the WT Kit v2 (Parse Biosciences cat. #ECW02030). The cDNA for all 8 subpools were cleaned using AMPure XP beads and quality checked using an Agilent Bioanalyzer before proceeding to Illumina and Nanopore library preparation. All 8 subpools were fragmented, size-selected using AMPure XP beads, and Illumina adapters were ligated. The cDNA fragments were cleaned again using beads and amplified, adding the fourth barcode and P5/P7 adapters, followed by size selection and quality checking with a Bioanalyzer. Libraries were sequenced with two runs of the Illumina NextSeq 2000 sequencer with P3 200 cycles kits (1.1 billion reads) and paired-end

run configuration 140/86/6/0. Libraries with 5% PhiX spike-in were loaded at 1000 pM for one run and 1100 pM for the second run and sequenced to an average depth of 301 million reads per library.

Long-Read-Split-Seq Experimental Protocol and Base Calling

Nuclei were barcoded and cDNA was purified as specified in the previous section. LR-Split-seq libraries were generated using an input of 200 fmol from the amplified, barcoded Split-seq cDNA before fragmentation (section 2 of the Split-seq protocol). Libraries were built using Oxford Nanopore Technologies Ligation Sequencing Kit (SQK-LSK114) and NEBNext Companion Module for Oxford Nanopore Technologies Ligation Sequencing (E7180L). The Short Fragment Buffer (SFB) from the Ligation Sequencing Kit (SQK-LSK114) during the second wash step. Libraries were loaded on R10.4.1 flowcells (FLO-PRO114M, FLO-MIN114) with an input of 20 fmol and 12 fmol, respectively. Sequencing was performed on the GridION and PromethION 2 Solo instruments using the MinKNOW software.

Bases were called from reads with Oxford Nanopore base-calling software Dorado v0.5.0 (`https://github.com/nanoporetech/dorado`) in super-accurate mode using config file dna_r10.4.1_e8.2_400bps_sup@v4.1.0 for both the exome capture and non-exome capture data, as well as the MinION and PromethION data.

Long-Read-Split-Seq Preprocessing and Quantification with splitcode and lr-kallisto

We first used splitcode to find barcodes and umis using linkers and reverse complements of linkers, allowing a total of 3 errors in linkers. We then used a custom python script to reverse the order of barcodes extracted from reverse strand to be in the same order as forward strand barcodes. Subsequently, we apply splitcode to combine and split randO and polyT barcodes from round 1 of Split-Seq barcoding, allowing 1 substitution or indel per barcode, 39,027,314 out of 105,591,654 raw reads passed this workflow. We then use lr-kallisto to pseudoalign and quantify the resulting reads; 22,197,716 of the reads pseudoalign. We performed QC with a 500 UMI threshold per nuclei and filtered to genes present in at least 100 cells.

Error rate estimation

Error rates for the PacBio dataset (Leung et al., 2021) were calculated by analyzing a subsample of 1/8th of the reads using the NanoSim read characterization module with the command ‘read_analysis.py transcriptome -i *fastq* -rg references/genome.fa -rt

references/transcriptome.fa -annot references/annotations.gtf -t 8 -o output_folder‘. Error rates for the LRGASP datasets were also calculated this way, without need for subsampling.

Benchmarking and comparisons

In benchmarks and comparisons of programs, we used Bambu v3.4.1, IsoQuant v3.3.0, and Oarfish v0.5.1. For the HCT116 data we also ran Oarfish 0.3.1 so as to be able to make a direct comparison with the results of (Jousheghani and Patro, 2024). We ran Oarfish according to the scripts at https://github.com/COMBINE-lab/lr_quant_benchmarks/blob/0b89465420250d3511044fdc3d988a320aba73c6/snakemake_rules/isoquant_sim_data/alignment/alignment_transcriptome/align.snk and https://github.com/COMBINE-lab/lr_quant_benchmarks/blob/0b89465420250d3511044fdc3d988a320aba73c6/snakemake_rules/isoquant_sim_data/quantification/oarfish_quant/quant.snk. In a previous version of this preprint (Loving et al., 2024), Oarfish v0.3.1 and v0.4.0 were used and the simulation data was run with SAMtools sort as in (Ji and Pertea, 2024). This appears to have resulted in overcounting that degraded Oarfish's performance.

Data Simulation

The simulation details for SIRV Set 4, where we generate error free reads uniformly expressed across isoforms in SIRV Set 4, are contained in the code for Figure 2 in the GitHub repo https://github.com/pachterlab/LSRRSRLFKOTWMWMP_2024. To see simulation details for Fig 3, see section Data Simulation in (Prjibelski et al., 2023), which describes in detail the simulation steps used starting with IsoSeqSim and NanoSim as the base simulators and using modifications to NanoSim to better preserve real ONT characteristics. For simulations presented in Supplement Figure 4a, we used a custom simulator based solely on error profiles, using ONT error profile of 38.5% of errors are deletions, 38.5% of errors are substitutions, and 23% of errors are insertions and PacBio error profile of 24.5% of errors are deletions, 52.4% of errors are substitutions, and 23.1% of errors are insertions with uniform error distribution within the read, which is full-length, available in the GitHub repo https://github.com/pachterlab/LSRRSRLFKOTWMWMP_2024 with details included with the upload at https://zenodo.org/records/11201284. For simulations presented in Supplement Figure 4b-c, we include NanoSim simulation details with the simulated data deposited at https://zenodo.org/records/11201284.

2.6 Data and code availability

The LRGASP data can be accessed from the accessions and ftp links listed in the data folder of `https://github.com/pachterlab/LSRRSRLFKOTWMWMP_2024`. IGVF Bridge exome capture and non-exome capture can be accessed from the IGVF portal with the accession IDs in the provided table.

Accession ID	Subpool Name	Read Type
IGVFDS4803WKTQ	B01_13G	Nanopore
IGVFDS9445YYVB	B01_13H	Nanopore
IGVFDS9522BMQK	B01_13G	Illumina
IGVFDS0356VCIO	B01_13H	Illumina

Table 2.2: IGVF Bridge exome capture and non-exome capture accession IDs.

Accession ID	File Name
IGVFDS4705QPIK	b01_nanopore_13G_single_cell_k63_both_mm39
IGVFDS7467TPQO	b01_nanopore_13G_single_cell_k63_polyT_mm39
IGVFDS3821ZEWS	b01_nanopore_13G_single_cell_k63_randO_mm39
IGVFDS1377KBXL	b01_next1_13G_single_cell_k31_both_mm39
IGVFDS2498XYWS	b01_next1_13G_single_cell_k31_polyT_mm39
IGVFDS9180SYAE	b01_next1_13G_single_cell_k31_randO_mm39
IGVFDS4019MYIG	b01_nanopore_13G_bulk_k63_casteij
IGVFDS6540HMFT	b01_nanopore_13G_bulk_k63_mm39
IGVFDS3833XYEY	b01_nanopore_13H_bulk_k63_casteij
IGVFDS5673HQEN	b01_nanopore_13H_bulk_k63_mm39
IGVFDS2760LQIX	b01_next1_13G_bulk_k31_casteij
IGVFDS9744VNMR	b01_next1_13G_bulk_k31_mm39
IGVFDS0231GDWH	b01_next1_13H_bulk_k31_casteij
IGVFDS1622ABWA	b01_next1_13H_bulk_k31_mm39

Table 2.3: IGVF Bridge exome capture and non-exome capture processed accession IDs.

The HCT116 cell line SG-NEx data was accessed on March 13, 2024 at `https://registry.opendata.aws/sg-nex-data`. The lr-kallisto method is available via release 0.51 of kallisto at `https://github.com/pachterlab/kallisto`.

We used bambu v3.4.1, IsoQuant v3.3.0, and oarfish v0.5.1 (with the exception of analysis of HCT116 data). In the initial version of the preprint, oarfish (v0.3.1 and v0.4.0) were used and the simulation data was run with samtools sort (genome coordinate sorting), causing overcounting in oarfish's performance due to oarfish's use of consecutive alignments of the same read filtering; this has been updated in this

version of the manuscript. Simulation data is available at https://zenodo.org/records/11201284. Processed abundance matrices for Figures 1-3 are available at https://zenodo.org/records/13755772. Code for reproducing the results and figures in the manuscript is available at https://github.com/pachterlab/LSRRSRLFKOTWMWMP_2024.

2.7 Author contributions

The lr-kallisto project was conceived by RKL and LP and the lr-kallisto method was developed and implemented by RKL. Benchmarking was conducted by RKL. The exome capture / non-capture experiment was conceived by AM, BWo and BWi. The experiment, data generation and curation was supervised by AM. Experiments were conducted by ER, HL, GF, SK and GM. Data curation was performed by FR, JS, DT and NR. Analysis of the data was conducted by RKL, FR and LP. RKL, ASB, and DKS streamlined the single-cell workflow using seqspec and splitcode. Supplementary data analysis was performed by RKL, LP and CO. Software testing and release was performed by RKL and DKS. The manuscript was drafted by RKL and LP. LP, RKL, AM, BW, FR, DKS and CO commented on and edited the manuscript. All authors approved the manuscript. LP supervised the lr-kallisto project with BW.

2.8 Acknowledgements

This work was partially supported by UM1 HG012077 to A.M., B.J.W., and L.P. as well as a United States Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0020347 to R.K.L. D.K.S. was supported by the UCLA-Caltech Medical Scientist Training Program (NIH NIGMS training grant T32 GM008042). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. We thank Zahra Zare Jousheghani, Noor Pratap Singh, and Rob Patro for comments on consistency and version control following the first version of this manuscript on bioRxiv.

References

Amarasinghe, Shanika L et al. (2020). "Opportunities and challenges in long-read sequencing data analysis". In: *Genome Biology* 21.1, p. 30.

Bloomfield, Max et al. (2024). "Oxford Nanopore next generation sequencing in a front-line clinical microbiology laboratory without on-site bioinformaticians". In: *Pathology* 56.3, pp. 444–447.

Bray, Nicolas L et al. (2016). "Near-optimal probabilistic RNA-seq quantification". In: *Nature Biotechnology* 34.5, pp. 525–527.

Castanho, Isabel et al. (2020). "Transcriptional signatures of tau and amyloid neuropathology". In: *Cell Reports* 30.6, 2040–2054.e5.

Chen, Ying, Nadia M Davidson, et al. (2021). "A systematic benchmark of nanopore long read RNA sequencing for transcript level analysis in human cell lines". In: *bioRxiv*. DOI: 10.1101/2021.04.21.440736.

Chen, Ying, Andre Sim, et al. (2023). "Context-aware transcript quantification from long-read RNA-seq data with Bambu". In: *Nature Methods* 20.8, pp. 1187–1195.

Cheng, Oliver et al. (2024). "Flexiplex: a versatile demultiplexer and search tool for omics data". In: *Bioinformatics* 40.3. DOI: 10.1093/bioinformatics/btae102.

Consortium, IGVF (2023). "The impact of genomic variation on function (IGVF) consortium". In: *ArXiv*. DOI: 10.1101/2023.03.28.533945.

Cook, David E et al. (2019). "Long-read annotation: Automated eukaryotic genome annotation based on long-read cDNA sequencing". In: *Plant Physiology* 179.1, pp. 38–54.

Delahaye, Clara and Jacques Nicolas (2021). "Sequencing DNA with nanopores: Troubles and biases". In: *PloS One* 16.10, e0257521.

Dobin, Alexander et al. (2013). "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1, pp. 15–21.

Dong, Xueyi et al. (2023). "Benchmarking long-read RNA-sequencing analysis tools using in silico mixtures". In: *Nature Methods* 20.11, pp. 1810–1821.

Frankish, Adam et al. (2021). "GENCODE 2021". In: *Nucleic Acids Research* 49.D1, pp. D916–D923.

Goodwin, Sara et al. (2015). "Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome". In: *Genome research* 25.11, pp. 1750–1756.

Hjörleifsson, Kristján Eldjárn et al. (2022). "Accurate quantification of single-nucleus and single-cell RNA-seq transcripts". In: *bioRxiv*. DOI: 10.1101/2022.12.02.518832.

Ji, Hyun Joo and Mihaela Pertea (2024). "Enhancing transcriptome expression quantification through accurate assignment of long RNA sequencing reads with TranSigner". In: *bioRxiv v2*, pp. 2024–08.

Jousheghani, Zahra Zare and Rob Patro (Mar. 2024). "Oarfish: Enhanced probabilistic modeling leads to improved accuracy in long read transcriptome quantification". In: *bioRxiv*. DOI: 10.1101/2024.02.28.582591.

Kabza, Michal et al. (2023). "Accurate long-read transcript discovery and quantification at single-cell resolution with Isosceles". In: *bioRxiv*, pp. 2023–11.

Kaminow, Benjamin, Dinar Yunusov, and Alexander Dobin (2021). "STARsolo: Accurate, fast and versatile mapping/quantification of single-cell and single-nucleus RNA-seq data". In: *bioRxiv*. DOI: 10.1101/2021.05.05.442755.

Landrith, Tyler et al. (2020). "Splicing profile by capture RNA-seq identifies pathogenic germline variants in tumor suppressor genes". In: *NPJ precision oncology* 4.1, p. 4.

Leung, Szi Kay et al. (2021). "Full-length transcript sequencing of human and mouse cerebral cortex identifies widespread isoform diversity and alternative splicing". In: *Cell Reports* 37.7, p. 110022.

Li, Heng (2018). "Minimap2: Pairwise alignment for nucleotide sequences". In: *Bioinformatics* 34.18, pp. 3094–3100.

Lienhard, Matthias et al. (2023). "IsoTools: a flexible workflow for long-read transcriptome sequencing analysis". In: *Bioinformatics* 39.6, btad364.

Loving, Rebekah K et al. (2024). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv v1*, pp. 2024–07.

Luebbert, Laura et al. (2023). "Efficient and accurate detection of viral sequences at single-cell resolution reveals novel viruses perturbing host gene expression". In: *bioRxiv*.

Melsted, Páll et al. (2021). "Modular, efficient and constant-memory single-cell RNA-seq preprocessing". In: *Nature Biotechnology* 39.7, pp. 813–818.

Pardo-Palacios, Francisco J et al. (July 2023). "Systematic assessment of long-read RNA-seq methods for transcript identification and quantification". In: *bioRxiv*. DOI: 10.1101/2023.07.25.550582.

Patro, Rob et al. (2017). "Salmon provides fast and bias-aware quantification of transcript expression". In: *Nature Methods* 14.4, pp. 417–419.

Penter, Livius et al. (2024). "Integrative genotyping of cancer and immune phenotypes by long-read sequencing". In: *Nature Communications* 15.1, p. 32.

Prjibelski, Andrey D et al. (2023). "Accurate isoform discovery with IsoQuant using long reads". In: *Nature Biotechnology* 41.7, pp. 915–918.

Reese, Fairlie et al. (May 2023). "The ENCODE4 long-read RNA-seq collection reveals distinct classes of transcript structure diversity". In: *bioRxiv*. DOI: 10.1101/2023.05.15.540865.

Sahlin, Kristoffer (2021). "Effective sequence similarity detection with strobemers". In: *Genome Research* 31.11, pp. 2080–2094.

Sahlin, Kristoffer and Veli Mäkinen (2021). "Accurate spliced alignment of long RNA sequencing reads". In: *Bioinformatics* 37.24, pp. 4643–4651.

Sakamoto, Yoshitaka, Sarun Sereewattanawoot, and Ayako Suzuki (2020). "A new era of long-read sequencing for cancer genomics". In: *Journal of Human Genetics* 65.1, pp. 3–10.

Sarkar, Hirak et al. (2020). "Accurate, efficient, and uncertainty-aware expression quantification of single-cell RNA-seq data". In: *bioRxiv*. DOI: 10.6084/m9.figshare.13198100.

Sullivan, Delaney K, Kyung Hoi Joseph Min, et al. (Nov. 2023). "Kallisto, bustools, and kb-python for quantifying bulk, single-cell, and single-nucleus RNA-seq". In: *bioRxiv*. DOI: 10.1101/2023.11.21.568164.

Sullivan, Delaney K and Lior Pachter (2024). "Flexible parsing, interpretation, and editing of technical sequences with splitcode". In: *Bioinformatics* 40.6.

Tang, Alison D et al. (2020). "Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns". In: *Nature communications* 11.1, p. 1438.

Tian, Luyi et al. (2021). "Comprehensive characterization of single-cell full-length isoforms in human and mouse with long-read sequencing". In: *Genome biology* 22, pp. 1–24.

Wang, Chaoyang et al. (2024). "Single-cell analysis of isoform switching and transposable element expression during preimplantation embryonic development". In: *PLoS Biology* 22.2, e3002505.

Warburton, Peter E and Robert P Sebra (2023). "Long-read DNA sequencing: recent advances and remaining challenges". In: *Annual Review of Genomics and Human Genetics* 24, pp. 109–132.

Wyman, Dana et al. (2019). "A technology-agnostic long-read analysis pipeline for transcriptome discovery and quantification". In: *Biorxiv*, p. 672931.

Yang, Chen et al. (2017). "NanoSim: nanopore sequence read simulator based on statistical characterization". In: *GigaScience* 6.4, gix010.

Zhang, David et al. (2020). "Incomplete annotation has a disproportionate impact on our understanding of Mendelian and complex neurogenetic disorders". In: *Science Advances* 6.24. DOI: 10.1126/sciadv.aay8299.

Chapter 3

LONG READ SEQUENCING TRANSCRIPTOME DISCOVERY WITH lr-kallisto

3.1 Abstract

Accurately generating novel transcript candidates has been an illusive challenge for long read RNA-seq transcriptomics. In order to accurately generate novel transcript candidates, accurately detecting already well-annotated transcripts is fundamental. As demonstrated by the LRGASP consortium, tools vary in detection of already annotated transcripts (full splice match, FSM), transcripts missing 3' or 5' end exons (incomplete splice match, ISM), transcripts which have novel junctions (novel in catalogue, NIC) and lastly containing novel junctions that are not yet annotated (novel not in catalog, NNC) (Pardo-Palacios et al., 2023). Of the tools analyzed in Pardo-Palacios et al., 2023, the sensitivity of tools to both known and unknown transcripts in real data with spike-ins is shown to be wanting (Pardo-Palacios et al., 2023, Fig.2f). In simulation, all are known to return a low percentage of NIC and NNC with both very low sensitivity and precision. As demonstrated in Chapter 1, lr-kallisto outperforms the standard tools for long read RNA-seq transcriptomics in detection and quantification; therefore, the extension of its methods to transcript discovery is natural although unconventional.

3.2 Introduction

Many methods have been developed for and focus on transcript discovery (Lienhard et al., 2023) (Prjibelski et al., 2023) (Kabza et al., 2024) (Chen et al., 2023). However, it remains a field where even the best tools create false positive novel annotations that are not backed by reads within the data used for the transcript discovery. Furthermore, due to the use of machine learning and user-defined parameters in methods for transcript discovery, it can be hard to determine the source of true discovery vs false positive annotations (Chen et al., 2023). Here, we present a completely data-driven and mapping-driven method for transcript discovery that relies entirely on where reads map within the transcriptome and genome and has a single parameter, the number of times a novel pattern defining a novel transcript candudate must be observed to be included in the analysis.

3.3 Results

Within the analyzed dataset of 793,577 long reads, 123,611 reads were found to have novel characteristics before any filtering on read support. Here we present a few reads that are called as novel and the pattern in which they map, which shows alternative splicing relative to the transcripts which share exons with the novel read. In Figure 3.1, 3.2 and 3.4, it is clear that alternative splicing has occurred simply from the graph. However, in Figure 3.2, we would suggest further analysis of whether or not this is a true positive or due to only error in the reads.

In Figure 3.1, we observe two reads with the same alternatively spliced arrangement that is a possible novel transcript in the human genome. While the displayed red exons are all annotated and shared with the transcript directly below the reads, these reads were not mapped to this transcript due to the d-list, where distinguishing flanking *k*-mers between the annotated intronic reion flanking the annotated exons likely lead to this read not mapping. In this example, there may be enough support for this candidate novel transcript given that our depth of sequencing is relatively low at less than 500,000 reads. In Figure 3.2, we observe that 187_unmapped is a alternatively spliced transcript with exons from multiple different transcripts that is not yet annotated in the human genome. However, in this example, there may not be enough support for this candidate novel transcript. In Figure 3.3, we observe that 146_unmapped appears to be possibly a false positive but still potentially a alternatively spliced transcript that is not yet annotated in the human genome; a more detailed analysis of the junctions betweeen 146_unmapped and ENST00000016171 is needed. However, in this example, there may not be enough support for this candidate novel transcript anyway. In Figure 3.4, we observe that, clearly, 49_unmapped is a alternatively spliced transcript that is not yet annotated in the human genome. However, in this example, there again may not be enough support for this candidate novel transcript.

3.4 Discussion

Methods for transcript discovery have primarily relied on splice-aware alignment followed by building intron or exon graphs which are used to detect novel splice sites. While kallisto is not a traditional splice-aware alignment tool, the distinguishing flanking *k*-mers provide a method for determining if a read has regions in it which are annotated as intronic or are *k*-mers which are contained in both introinic and exonic regions. In the latter case, we do not want to determine that these *k*-mers directly make the read novel, as these are unlikely true positive novel transcripts.

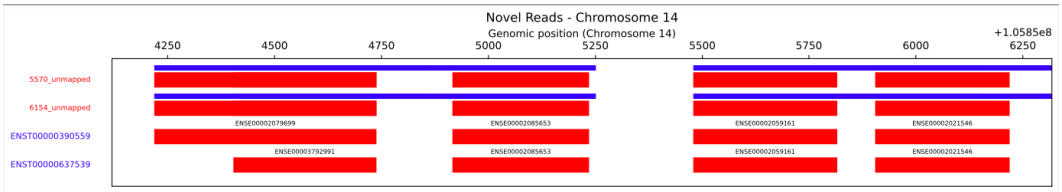


Figure 3.1: The top row of this plot displays the read (in blue) and its pseudoaligned exons (in red) that did not map in the initial pseudoalignment with the d-list, but now maps in the exon- and fusion- based approach to mapping novel reads. Below are listed all other transcripts that also contain overlapping exons with the read.

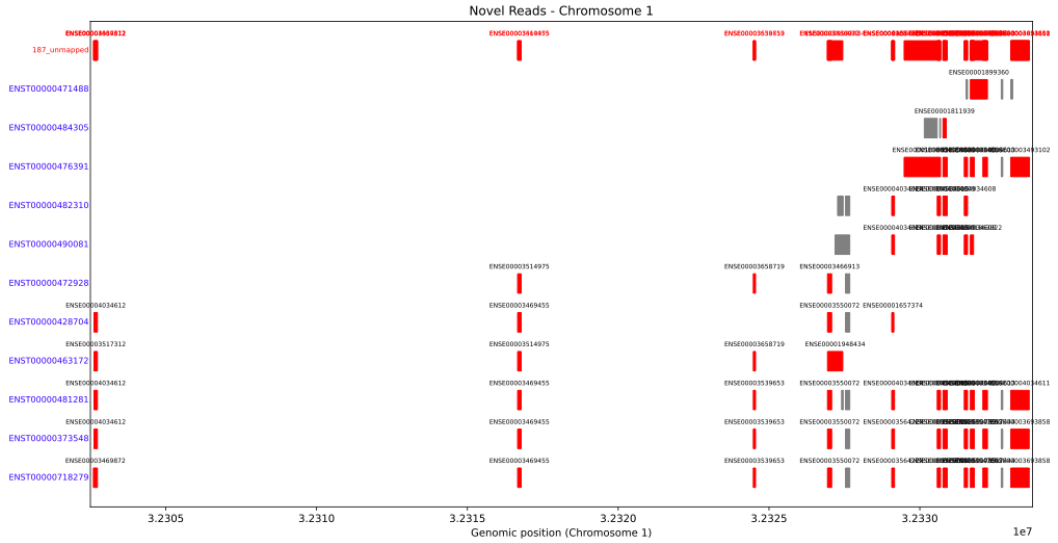


Figure 3.2: The top row of this plot displays the read that did not map in the initial pseudoalignment with the d-list, but now maps in the exon- and fusion- based approach to mapping novel reads. Below are listed all other transcripts that also contain overlapping exons with the read.

Figure 3.3: The top row of this plot displays the read that did not map in the initial pseudoalignment with the d-list, but now maps in the exon- and fusion- based approach to mapping novel reads. Below are listed all other transcripts that also contain overlapping exons with the read.

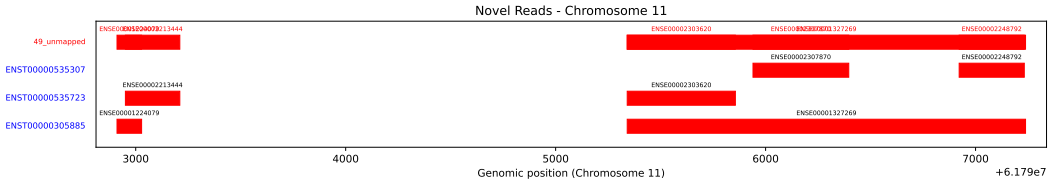


Figure 3.4: The top row of this plot displays the read that did not map in the initial pseudoalignment with the d-list, but now maps in the exon- and fusion- based approach to mapping novel reads. Below are listed all other transcripts that also contain overlapping exons with the read.

However, if a read extends into an intron, then either there is intron retention or it is an incomplete annotation of the UTR. In this case, a different UTR length can be consequential in the expression of the transcript, so determining if the annotation is, in fact, incorrect is consequential. Thus, if there is support for an extended UTR for a transcript, it is an important update to the annotation. Second, within kallisto, we can also determine when there are disjoint mapping k-mers, i.e. there are k-mers which must originate from different transcript models, but were found within the same RNA molecule. This is the case often considered when building intron or exon graphs. Within *dn*-kallisto, we can use an exon index without d-listing the genome to build a proxy for an exon graph. When we align the novel candidates output by lr-kallisto in this exon t-DBG, we are performing the mapping within the exon graph to build out any novel annotations. Finally, there is a third option, which is further described in methods, for detecting when a molecule does not match anywhere in the transcriptome or d-list for a user-defined threshold of the molecule length. This can be useful for detecting regions of high variability in the genome where annotations are not useful for mapping.

3.5 Methods

Within lr-kallisto, distinguishing flanking k-mers have been implemented to increase the precision of quantification between spliced and unspliced RNAs, as well as virus contamination and infection within samples and host species. The concept of distinguishing flanking k-mers is also very useful in the context of mapping long reads and detecting when a long read does not map within the annotated transcriptome. There are many different cases that can occur in the case of unannotated transcripts, such as unannotated exons, undocumented alternative splicing events, and incorrectly or differing transcription start sites and transcription end sites. Here we do not attempt to handle every case of unannotated transcript that exists but describe a workflow

that determines candidate novel transcripts. This workflow can provide insight on $> 99\%$ of the reads that are not mapped with lr-kallisto and, thus, provide useful data for both biological and technical discoveries.

There are three steps to performing this analysis:

First, lr-kallisto is used with the `-unmapped` flag to pseudoalign the reads with a d-list index, i.e. an index that has been built with a transcriptome as the input and a genome as the d-list. During this pseudoalignment reads that do not map are filtered into `novel.fastq` along with reads that have a disjoint intersection of mapping k-mers and reads that have too many unmapping k-mers, which is paramater we set within this work this parameter is set to .8, which we have empirically found to be useful and will further explore to.

Second, lr-kallisto is used to pseudoalign with option `-union` to take the union set of k-mers across the exon compatibilities using two exon-based indices that are generated with a custom script to extract exonic cdna from the genome where in one index we again are d-listing the genome and in the other case we do not d-list the genome and instead use the nascent index. Furthermore, we also use the `-fusion` option, which is described in Chapter 5. This results in each read being partitioned into the non-intersecting regions of transcript compatibility, where the transcript compatibility set is now exon compatibility set.

Third, we collate the pseudobam results from 2. to create candidates for novel transcripts. These collated results are then sorted to those of the same structure being grouped together and filtered by the desired read count cut-off per candidate, which is a user provided parameter.

3.6 Data and Code Availability

```
ENCSR706ANY gm12878_1_1 gm12878 GM12878 gm12878 blood   cell_line
   #0798c8   ENCFF596ODX ENCFF475ORL ENCFF234YIJ "https://www.
   encodeproject.org/documents/fc272a30-b9a5-4652-b255-424
   b61d4587b/ , https://www.encodeproject.org/documents/7ec9d66a
   -3b7e-4183-8677-e1df14770b44/" "'ENCODE PacBio Iso-seq
   Analysis Protocol (v.1.0)', 'ENCODE Long Read RNA-Seq Analysis
    Protocol for Human Samples (v.1.0)'" Pacific Biosciences
   Sequel    FALSE 715140  FALSE     RNeasy mini kit SuperScript
   II  ExoIII and ExoVII to remove failed ligation products  "
   SMRTbell  Template Prep Kit 1.0,"  Protocol 1: non-size-
```

```
selected cDNA libraries v0 pb_cdna
```

Listing 3.1: "data and metadata"

The code for `dn-kallisto` is included in the repository for Chapter 5 `https://github.com/bound-to-love/fugi.git`.

3.7 Future Benchmarking and Directions

This is a proof of concept of the application of pseudoalignment to transcript discovery. Further benchmarking is needed to show the real proficiency of this method in this task. However, we expect what is observed within this preliminary analysis to stand the test of further data and exploration, due to the prior partial success of exon structure analysis and splicing analysis in transcript discovery. Thus, the use of pseudoalignment to perform exon compatibility analysis and then splicing awareness demonstrating efficacy is natural. Furthermore, TCCs have already been used in transcript discovery and quantification as shown in Isosceles development (Kabza et al., 2024). dn-kallisto is, therefore, a new approach combining pseudoalignment, exon compatiblity (instead of transcript compatibility) and fusion analysis for transcript discovery, which we will continue to extend and benchmark to provide a data-forward, unambiguous method of transcript discovery.

References

Chen, Ying et al. (2023). "Context-aware transcript quantification from long-read RNA-seq data with Bambu". In: *Nature Methods* 20.8, pp. 1187–1195.

Kabza, Michal et al. (Aug. 2024). "Accurate long-read transcript discovery and quantification at single-cell, pseudo-bulk and bulk resolution with Isosceles". en. In: *Nat. Commun.* 15.1, p. 7316.

Lienhard, Matthias et al. (2023). "IsoTools: a flexible workflow for long-read transcriptome sequencing analysis". In: *Bioinformatics* 39.6, btad364.

Pardo-Palacios, Francisco J et al. (July 2023). "Systematic assessment of long-read RNA-seq methods for transcript identification and quantification". In: *bioRxiv*. DOI: 10.1101/2023.07.25.550582.

Prjibelski, Andrey D et al. (2023). "Accurate isoform discovery with IsoQuant using long reads". In: *Nature Biotechnology* 41.7, pp. 915–918.

Chapter 4

SINGLE-CELL ANALYSIS WITH lr-kallisto

4.1 Introduction

In this chapter, we will provide a more thorough description of the workflow for single-cell analysis with lr-kallisto using splitcode and seqspec as well as further analyze the IGVF Bridge data and the IGVF 8cube data (Rebekah K Loving et al., 2024; Sullivan and Pachter, 2023; Booeshaghi, Chen, and Pachter, 2024). The analysis of single-cell long read data, in particular Oxford Nanopore long read data, is complicated by the fact that the motor protein which guides the sequence through the nanopore to produce the reads may be in any of the four orientations on the double stranded DNA producing reads in forward, complement, reverse complement, or reverse orientation and, moreover, that the existence of concatemers is frequent (12% in the data that we present here).

Therefore, we must extract from the reads the barcodes and UMIs in order to perform single-cell analysis. In order to extract the barcodes and UMIs from the reads, the structure of the read must be known. A `seqspec` yaml provides the structural information for RNA-seq reads. This machine readable `seqspec`, which specifies all needed information about the single-cell long reads, can then be used to create a configuration file specfying the extraction patterns which can then be passed to `splitcode` to preprocess the sequences prior to mapping and quantification of single-cell long reads by lr-kallisto (Booeshaghi, Chen, and Pachter, 2024; Sullivan and Pachter, 2024; Rebekah K Loving et al., 2024). Thus, this provides a complete and easily adaptable method for processing single-cell long read RNA-seq to obtain transcript- and gene-level single-cell RNA-seq expression quantifications.

4.2 Documentation

Pseudoalignment of single-cell long read RNA seq data

Note

Reference: Loving, R, Sullivan, DK, Reese, F, Rebboah, E, Sakr, J, Rezaie, N, Liang, HY, Filimban, G, Kawauchi, S, Oakes, C, Trout, D, Williams, BA, MacGregor, G, Wold, BJ, Mortazavi, A, Pachter, L Long-read sequencing transcriptome quantification with lr-kallisto. bioRxiv 2024.07.19.604364 https://doi.org/10.1101/2024.07.19.604364

kallisto can perform long-read pseudoalignment of nucleotide sequences against a large *k*-mer reference while retaining single-cell (for single-cell RNA sequencing data) or sample (for bulk RNA seq data) resolution. To perform long-read pseudoalignment, first add `-k 63` to `kb ref` and, second, add the `--long` flag to the `kb count` commands.

Long-read pseudoalignment is performed by the longer k-mer length improving the quality of mapping k-mers in the higher sequencing error rates (relative to short read sequencing), making it more probable that the read originates from the transcript compatibility class it maps to. As k increases, the number of distinct k-mers also increases, but the number of contigs decreases. This implies that the number of transcripts in a transcript compatibility class decreases on average with increasing length of k. Overall, the complexity of the T-DBG decreases (Supplementary Fig. 5), increasing the probability of the read originating from the transcript compatibility class it is mapping to. Furthermore, this also increases the probability of the intersection of equivalence classes being nonempty, which increases the overall mapping rate.

The workflow can be executed in three lines of code, and computational requirements do not exceed those of a standard laptop. Building on kallisto's versatility, the workflow is compatible with all state-of-the-art single-cell and bulk RNA sequencing methods, including but not limited to SMART-Seq [add citation]_ and SPLiT-Seq [add citation]_ (including Parse Biosciences) and performance is state-of-the-art on both PacBio and Oxford Nanopore Technologies long-read data.

Note

For long-read single-cell data to be processed with lr-kallisto some preprocessing steps are required. Here we present the use of seqspec and splitcode to facilitate an automated processing of LR-SPLiT-Seq [add citation]_. seqspec is used to create a configuration file for splitcode to extract barcodes, umis, and the biological sequences from the reads. seqspec requires as input a machine readable specification file for the sample protocol that is in the seqspec format. splitcode can then be called on the reads with the configuration file created by seqspec to extract from the reads the barcodes, umis, and biological sequences. The output of splitcode can be piped directly into lr-kallisto or output to files that are processed with lr-kallisto.

The long-read pseudoalignment workflows can be used to align RNA sequencing data to any transcriptome reference:

1. Install *kb-python* (optional: install gget to fetch the host genome and transcriptome) as well as seqspec and splitcode:

```
pip install kb-python gget git+https://github.com/pachterlab/seqspec
git clone https://github.com/pachterlab/splitcode
cd splitcode
mkdir build
cd build
cmake ..
make
make install
```

2. Create splitcode config file using seqspec:

```
seqspec index -m rna -s file -t splitcode spec.yaml > seqspec-config.txt
```

3. Use splitcode to extract barcodes, umis, and biological sequences:

```
splitcode -c seqspec-config.txt $sample_data.fastq.gz -o $sample_data_modified.fastq.gz -t 32
```

4. Importantly, the extracted sequences may need reorienting for the sample to be processed appropriately; we give an example of this in the case of ONT single-nuclei samples in the tutorials.
5. Create reference index (using the D-list of human genome):

```
kb ref \
    -k 63 \
    --d-list $(gget ref --ftp -w dna homo_sapiens) \
    --workflow standard \
    -i index.idx \
    -g t2g.txt \
    -f1 fasta.fa \
    $(gget ref --ftp -w dna,gtf homo_sapiens)
```

6. Align and quantify sequencing reads:

```
kb count \
    --long \
    -i index.idx \
    -g t2g.txt \
    --parity single \
    --tcc \
    --matrix-to-directories \
    -x '0,0,0:1,0,0:2,0,0' \
    $sample_barcode.fastq.gz $sample_umi.fastq.gz $sample_bioseq.fastq.gz
```

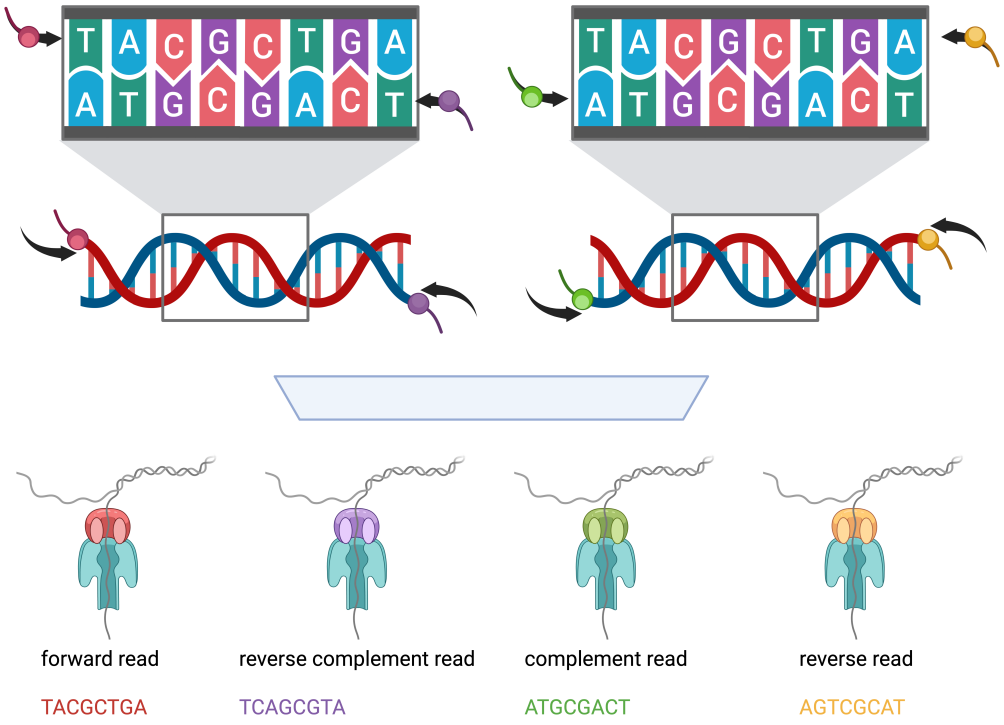


Figure 4.1: Motor protein ligation and the motor protein which engages with the nanopore determine read orientation. Here we illustrate the effect of where motor proteins ligate and which motor protein reaches the nanopore determining the orientation of the Oxford Nanopore read output. The circles with tails indicate the motor proteins and their positions on the double stranded DNA. The red circle (and motor protein) indicate that the resulting read will be a forward read. The purple circle (and motor protein) indicate that the resulting read will be a reverse complement read. The green circle (and motor protein) indicate that the resulting read will be a complement read. Finally, the yellow circle (and motor protein) indicate that the resulting read will be a reverse read.

4.3 A deeper dive into each step

1. **seqspec** (Booeshaghi, Chen, and Pachter, 2024) - `seqspec` provides three important characteristics for long read RNA-seq processing. First, it provides a file format with the complete metadata for the RNA sequencing library structure information in a machine readable format. This encoding of RNA-seq libraries can then be validated with `seqspec`'s validate command to ensure the file is a valid specification. Lastly, `seqspec`'s index command can be used with the `seqspec.yaml` of a specific RNA-seq library to generate a configuration file, needed by `splitcode`, which defines extraction patterns for the specific RNA-seq library defined within the `seqspec`. The structure

of the single-cell ONT reads can then be processed by `splitcode` and piped directly to `lr-kallisto`.

```
#Given an input seqspec.yaml for the RNA-seq library being used
seqspec index -m rna -s file -t splitcode seqspec.yaml > splitcode.config
```

Listing 4.1: running seqspec index

Below we include Figure 1 from Booeshaghi, Chen, and Pachter, 2024 to elucidate the `seqspec` file format and then the `spec.yaml` for the LR-SPLiT-Seq used in the IGVF Bridge exome capture dataset included in Chapter 1 of this thesis:

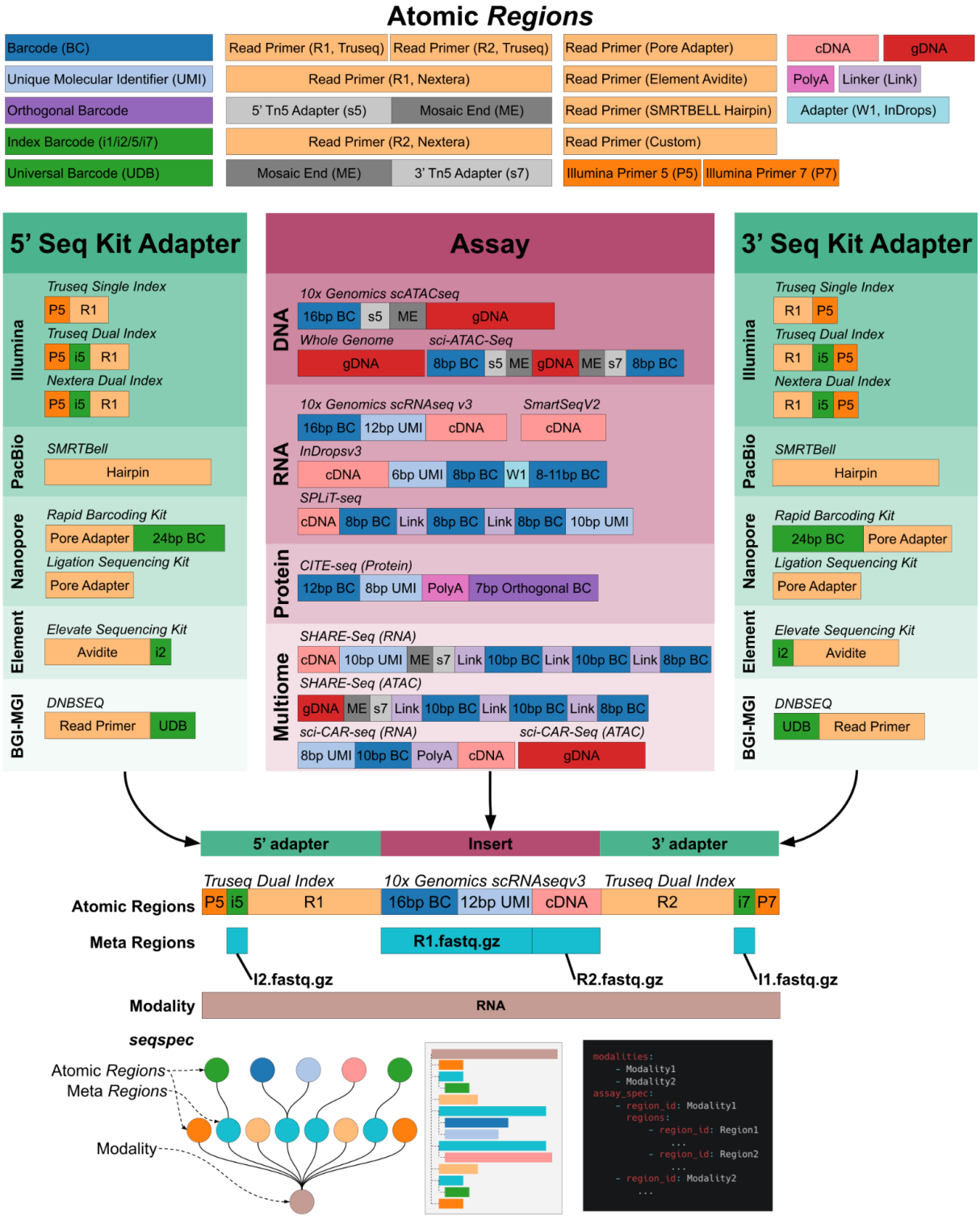


Figure 1: The structure of reads sequenced from genomics libraries. Sequencing libraries are constructed by combining Atomic *Regions* to form an adapter-insert-adapter construct. The *seqspec* for the assay annotates the construct with *Regions* and *Meta Regions.*

```
!Assay
seqspec_version: 0.3.0
assay_id: parse-wt-nanopore
name: WT Mega v2 nanopore
doi: https://docs.google.com/presentation/d/
17yKh6xE5b9Mo4DaXx5uPvFZOIHOW0kbK-QsU2ECwx8c/
edit#slide=id.g29abb1440dc_0_500
date: 13 November 2023
description: split-pool ligation-based transcriptome sequencing
modalities:
- rna
lib_struct: ''
library_protocol: Parse Bio Evercode WT Mega v2.2.1 with cDNA Exome
Capture v1.0.1
library_kit: Oxford Nanopore Ligation sequencing DNA V14 (SQK-LSK114)
sequence_kit: null
sequence_protocol: Oxford Nanopore PromethION 2 Solo
sequence_spec:
- !Read
  read_id: read.fastq.gz
  name: Nanopore read
  modality: rna
  primer_id: ont-1
  min_len: 177
  max_len: 2000194
  strand: pos
  files:
  - !File
    file_id: read.fastq.gz
    filename: read.fastq.gz
    filetype: ''
    filesize: 0
    url: ''
    urltype: ''
    md5: ''
library_spec:
- !Region
  region_id: rna
  region_type: rna
  name: rna
  sequence_type: joined
  sequence:
XXXXXXXXXXXXXXXXXXXXXXXXXXXAAGCAGTGGTATCAACGCAGAGTGAATGGGXXXXXXXNNNNNN
NNGTGGCCGATGTTTCGCATCGGCGTACGACTNNNNNNNNNATCCACGTGCTTGAGACTGTGGNNNNNNNN
XXXXXXXXXXAGATCGGAAGAGCACACGTCTGAACTCCAGTCACXXXXXXXXXXXXXXXXXXXXXXXXXX
X
  min_len: 211
  max_len: 2000228
  onlist: null
  regions:
```

```
  - !Region
    region_id: ont-1
    region_type: custom_primer
    sequence_type: random
    sequence: XXXXXXXXXXXXXXXXXXXXXXXXXX
    name: ont-2
    min_len: 27
    max_len: 36
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: truseq_read2
    region_type: truseq_read2
    name: Illumina Read 2
    sequence_type: fixed
    sequence: AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC
    min_len: 34
    max_len: 34
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: umi
    region_type: umi
    name: umi
    sequence_type: random
    sequence: XXXXXXXXXX
    min_len: 10
    max_len: 10
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: barcode-3
    region_type: barcode
    name: barcode-3
    sequence_type: onlist
    sequence: NNNNNNNN
    min_len: 8
    max_len: 8
    onlist: !Onlist
      file_id: onlist_bc3.txt
      filename: onlist_bc3.txt
      filetype: txt
      filesize: 864
      url: https://raw.githubusercontent.com/pachterlab/qcbc/main/
tests/parsebio_wtmega96/onlist_bc3.txt
      urltype: https
      md5: 1452e8ef104e6edf686fab8956172072
```

```
      location: local
    regions: null
    parent_id: rna
  - !Region
    region_id: linker-1
    region_type: linker
    name: linker-1
    sequence_type: fixed
    sequence: ATCCACGTGCTTGAGACTGTGG
    min_len: 22
    max_len: 22
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: barcode-2
    region_type: barcode
    name: barcode-2
    sequence_type: onlist
    sequence: NNNNNNNN
    min_len: 8
    max_len: 8
    onlist: !Onlist
      file_id: onlist_bc2.txt
      filename: onlist_bc2.txt
      filetype: txt
      filesize: 864
      url: https://raw.githubusercontent.com/pachterlab/qcbc/main/
tests/parsebio_wtmega96/onlist_bc2.txt
      urltype: https
      md5: 1452e8ef104e6edf686fab8956172072
      location: local
    regions: null
    parent_id: rna
  - !Region
    region_id: linker-2
    region_type: linker
    name: linker-2
    sequence_type: fixed
    sequence: GTGGCCGATGTTTCGCATCGGCGTACGACT
    min_len: 30
    max_len: 30
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: barcode-1
    region_type: barcode
    name: barcode-1
    sequence_type: onlist
```

```
    sequence: NNNNNNNN
    min_len: 8
    max_len: 8
    onlist: !Onlist
      file_id: onlist_bc1.txt
      filename: onlist_bc1.txt
      filetype: txt
      filesize: 1728
      url: https://raw.githubusercontent.com/pachterlab/qcbc/main/
tests/parsebio_wtmega96/onlist_bc1.txt
      urltype: https
      md5: 5c3b70034e9cef5de735dc9d4f3fdbde
      location: local
    regions: null
    parent_id: rna
  - !Region
    region_id: primer
    region_type: custom_primer
    name: primer
    sequence_type: random
    sequence: XXXXXX
    min_len: 6
    max_len: 6
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: cDNA
    region_type: cdna
    name: cDNA
    sequence_type: random
    sequence: X
    min_len: 1
    max_len: 2000000
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: tso
    region_type: custom_primer
    name: tso
    sequence_type: fixed
    sequence: AAGCAGTGGTATCAACGCAGAGTGAATGGG
    min_len: 30
    max_len: 30
    onlist: null
    regions: null
    parent_id: rna
  - !Region
    region_id: ont-2
```

```
  region_type: custom_primer
  name: ont-1
  sequence_type: random
  sequence: XXXXXXXXXXXXXXXXXXXXXXXXXX
  min_len: 27
  max_len: 36
  onlist: null
  regions: null
  parent_id: rna
parent_id: null
```

2. **splitcode** (Sullivan and Pachter, 2023) - splitcode is a very versatile tool for the manipulation, searching, and extraction of sequences. Within the single-cell workflow for lr-kallisto, splitcode is utilized to do the extraction of barcodes, UMIs, and cNDA relative to the location of linkers, which it searches the read sequence to find. Moreover, due to the sequencing error rates of long read RNA-sequencing, this search for the linkers between the barcodes must be done with error tolerance. Further, as described in Figure 4.1, the strand orientation of the read (and the linkers) can be in any of four orientations, forward, reverse complement, complement, and reverse, depending on where the motor proteins ligate onto the double stranded DNA and which of the motor proteins reaches the nanopore first. Thus, the different orientations of the linkers with error tolerance must also be taken into account. The barcodes, UMIs, and cDNA must then be reoriented according to the orientation of the linker that was found in the read so that all the resulting reads are in the same orientation for mapping. All of this preprocessing is cleanly executed within a single configuration file and a single run of `splitcode`. The resulting extracted sequences can then be piped directly to `splitcode` or they can be written out to files.

```
@extract <^GTGGCCGATGTTTCGCATCGGCGTACGACT^f_read[10]>8{linker-2f}
@extract <f_read[8]>{linker-2f},{linker-2f}<f_read[8]>{linker-1f},{linker-1f}<f_read[8]>
@extract {linker-1f}14<f_read>0:-1
@extract <^CACCGGCTACAAAGCGTAGCCGCATGCTGA^c_read[10]>8{linker-2c}
@extract <c_read[8]>{linker-2c},{linker-2c}<c_read[8]>{linker-1c},{linker-1c}<c_read[8]>
@extract {linker-1c}14<c_read>0:-1
@extract 0:0<^TCAGCATGCGGCTACGCTTTGTAGCCGGTG^r_read>14{linker-1r}
@extract <r_read[8]>{linker-1r},{linker-1r}<r_read[8]>{linker-2r},{linker-2r}<r_read[8]>
@extract {linker-2r}8<r_read[10]>
@extract 0:0<^AGTCGTACGCCGATGCGAAACATCGGCCAC^rc_read>14{linker-1rc}
@extract <rc_read[8]>{linker-1rc},{linker-1rc}<rc_read[8]>{linker-2rc},{linker-2rc}<rc_read[8]>
@extract {linker-2rc}8<rc_read[10]>
groups  ids tags  distances locations
group1  linker-2f GTGGCCGATGTTTCGCATCGGCGTACGACT  3:3:3 0:0:0
```

```
group1  linker-2c CACCGGCTACAAAGCGTAGCCGCATGCTGA  3:3:3 0:0:0
group1  linker-2r TCAGCATGCGGCTACGCTTTGTAGCCGGTG  3:3:3 0:0:0
group1  linker-2rc  AGTCGTACGCCGATGCGAAACATCGGCCAC  3:3:3
    0:0:0
group2  linker-1f ATCCACGTGCTTGAGACTGTGG  3:3:3 0:0:0
group2  linker-1c TAGGTGCACGAACTCTGACACC  3:3:3 0:0:0
group2  linker-1r GGTGTCAGAGTTCGTGCACCTA  3:3:3 0:0:0
group2  linker-1rc  CCACAGTCTCAAGCACGTGGAT  3:3:3 0:0:0

@nest
@extract {linker-2f}<splitcode_umi_barcode_cDNA>0:-1
@extract {linker-2c}<~c~splitcode_umi_barcode_cDNA>1:-1
@extract {linker-2r}<~r~splitcode_umi_barcode_cDNA>2:-1
@extract {linker-2rc}<~splitcode_umi_barcode_cDNA>3:-1
group ids tags  distances locations
group1  linker-2f GTGGCCGATGTTTCGCATCGGCGTACGACT  3:3:3 0:0:0
group1  linker-2c CACCGGCTACAAAGCGTAGCCGCATGCTGA  3:3:3 1:0:0
group1  linker-2r TCAGCATGCGGCTACGCTTTGTAGCCGGTG  3:3:3 2:0:0
group1  linker-2rc  AGTCGTACGCCGATGCGAAACATCGGCCAC  3:3:3
    3:0:0
```

Listing 4.2: example splitcode configuration file for LR-SPLiT-Seq

In this example `splitcode` configuration file, note that each orientation of linker is used with error tolerance of 3. Further, note that we insert linker2 in the orientation the read is in into the extraction in the first half of the configuration file prior to @nest and that the orientation of the reads are converted to all be forward oriented in the portion of the configuration file following @nest. Here we use the prefix linker2 to identify the orientation and then use one of the following to reorient the reads accordingly: with *c* complements the extraction, *r* reverses the extraction, and reverse complements the extraction.

```
splitcode -c splitcode.config -t 32 input.fastq.gz --x-only
    --gzip;
```

Listing 4.3: running splitcode for barcode

3. **lr-kallisto bus** (Rebekah K. Loving et al., 2025) - `lr-kallisto bus` performs the pseudoalignment step of mapping the reads to the transcriptome. This can be performed on either bulk data (`-x bulk`) or on single cell data, where a technology or technology string is needed. For a technology string, the following format must be followed:

-x barcode_file#,start_position,end_position:

UMI_file#,start_position,end_position:

cDNA_file#,start_position,end_position.

For example: -x '0,10,34:0,0,10:0,0,0' specifies a barcode of length 24 in the first file starting at position 10 in each read with a UMI in the first file as well starting at position 0 with length 10, and that the remainder of the read is the cDNA. `lr-kallisto bus` will output the following files: `output.bus`, which contains the bus records of the pseudoalignment, `novel.fastq`, which contains the reads that did not map and the reads that are considered novel due to properties of their *k*-mers that mapped, and `run_info.json`, which includes mapping statistics and metainformation regarding the run including software versions.

```
kallisto bus -x '1,0,24:2,0,10:0,0,0' --long --threshold 0.8
    --unmapped -i index_k-63.idx -o output_dir -t 4 input.
    fastq
```

Listing 4.4: running lr-kallisto bus

4. **lr-kallisto quant –pseudobam –long** (Rebekah K. Loving et al., 2025)- This command allows you to both acquire the bulk-level or pseudobulk-level quantifications as well as the pseudobam mappings of the alignments. The outputs are an `abundance.tsv` with transcript-level quantifications, a `pseudoalignments.bam` with BAM formatted alignments including CIGAR Strings, `novel.fastq` with reads considered novel transcript candidates, and `run_info.json` with metadata for the run.

```
kallisto quant -i index_k-63.idx -t 4 --pseudobam --long --
    output output_dir input.fastq
```

Listing 4.5: running lr-kallisto quant

5. **kb count** (Sullivan, Min, et al., 2023) (Rebekah K. Loving et al., 2025) - `kb` is a python wrapper for running kallisto and bustools for either bulk or single-cell RNA-seq data. Thus, with `kb count`, we can go from splitcode processed reads to quantifications in one command. We could even go from raw reads to quantifications in one command if we used splitcode's piping. Here we show the command using the output from splitcode, as having the saved processed fastq is useful for other purposes.

```
kb count -k 63 -t 4 --long --threshold 0.8 -i index_k-63.idx
    -g ref/index_k-63_t2g.txt -o kb_outdir -x '
    0,10,34:0,0,10:0,0,0' splitcode_umi_barcode_cDNA.fastq.gz
    --opt-off --overwrite --quant-umis --tcc -w onlist.txt --
    filter bustools --filter-threshold 150
```

Listing 4.6: running lr-kallisto with kb count

```
reads='igvf_b01_13G.fastq.gz'
path_to_lr_kallisto='kallisto_v0.51.0/kallisto/build/src/
    kallisto'
path_to_bustools='bustools/build/src/bustools'
output='b01_nanopore_13G_single_cell'
ref='ref/mm39'
tech='1,0,22:1,22,32:0,0,0'

kb ref-i ${ref}_k-63.idx -k 63 -f1 ${ref}.cdna.fa -g ${ref}.
    t2g ${ref}.fa.gz ${ref}.gtf.gz

${path_to_lr_kallisto} bus -x '1,0,24:2,0,10:0,0,0' --long --
    threshold (-r) 0.8 -i ${ref}_k-63.idx -o ${output} -t 4 ${
    reads}

${path_to_bustools} sort -t 4\
 ${output}/output.bus \
 -o ${output}/sorted.bus;

${path_to_bustools} whitelist -o ${output}/whitelist.txt \
 ${output}/sorted.bus;

${path_to_bustools} correct -w ${output}/whitelist.txt \
 -o ${output}/corrected.bus ${output}/sorted.bus;

${path_to_bustools} count ${output}/corrected.bus \
 -t ${output}/transcripts.txt \
 -e ${output}/matrix.ec \
 -o ${output}/count -m \
 -g ${ref}.t2g;

${path_to_lr_kallisto} quant-tcc -t 4 \
 ${output}/count.mtx \
 -i ${ref}_k-31.idx \
 -e ${output}/count.ec.txt \
 -o ${output};
```

```
33
34 ${path_to_bustools} count ${output}/corrected.bus \
35  -t ${output}/transcripts.txt \
36  -e ${output}/matrix.ec \
37  -o ${output}/gcount --em --genecounts \
38  -g ${ref}.t2g;
```

Listing 4.7: the commands wrapped in kb count –long

4.4 Future explorations

There are many exciting applications of both lr-kallisto and dn-lr-kallisto to be explored. Here we present some ideas for applications and analysis that can immediately be pursued with available data.

Exome capture vs non-exome capture in IGVF Bridge data

While we explored some of the differences between exome capture and non-exome capture in the IGVF Bridge data, this could be extended significantly to fully understand the impact of exome capture on the created datasets ability to perform exploratory analysis. This could be done using different versions of annotations to analyze the differences in expression between exome capture and non-exome capture, while we showed there is a signficant difference in expression of lnc-RNAs which we know many of these lnc-RNAs have functional properties, we did not thoroughly explore the difference in expression across coding proteins, which would be a useful analysis to explore the bias that exome capture may be creating even in the coding sequences. Furthermore, transcript discovery techniques could be applied to the analysis of exome capture and non-exome capture. An investigation of whether transcript discovery in protein coding regions improves or degrades with the use of exome capture would be an impactful result.

Differential transcript expression between cell-types and strains

An exciting next avenue of exploration of the IGVF 8cube data is a classical analysis of the differential expression between cell-types, tissues, and strains in the long reads and how it differs from paired short reads from the same cells will be done and will show the impact of long read RNA-sequencing on transcript-level expression (Rebboah et al., 2025). We will also show differential expression at the pseudobulked cell-type level and the pseudobulked tissue level between individuals and between strains.

Discovery of novel candidate transcripts in PBMC

Furthermore, we can also apply our dn-lr-kallisto workflow to identify possible novel transcripts using unmapped reads workflow with passing number of supporting reads.

LISTINGS

References

Booeshaghi, Ali Sina, Xi Chen, and Lior Pachter (Mar. 2024). "A machine-readable specification for genomics assays". en. In: *Bioinformatics* 40.4.

Loving, Rebekah K et al. (2024). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv v1*, pp. 2024–07.

Loving, Rebekah K. et al. (2025). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv*. R.K.L conceived project, implemented and benchmarked project, and was principal writer of manuscript. This work is included in this thesis. DOI: 10.1101/2024.07.19.604364. eprint: https://www.biorxiv.org/content/early/2025/01/29/2024.07.19.604364.full.pdf. URL: https://www.biorxiv.org/content/early/2025/01/29/2024.07.19.604364.

Rebboah, Elisabeth et al. (2025). "Systematic cell-type resolved transcriptomes of 8 tissues in 8 lab and wild-derived mouse strains captures global and local expression variation". In: *bioRxiv*. R.K.L. provided feedback throughout this analysis. This work is not included in this thesis. DOI: 10.1101/2025.04.21.649844.

Sullivan, Delaney K, Kyung Hoi Joseph Min, et al. (Nov. 2023). "Kallisto, bustools, and kb-python for quantifying bulk, single-cell, and single-nucleus RNA-seq". In: *bioRxiv*. DOI: 10.1101/2023.11.21.568164.

Sullivan, Delaney K and Lior Pachter (Dec. 2023). "Flexible parsing, interpretation, and editing of technical sequences with Splitcode". In: *bioRxiv*. DOI: 10.1101/2023.03.20.533521.

Sullivan, Delaney K and Lior Pachter (2024). “Flexible parsing, interpretation, and editing of technical sequences with splitcode”. In: *Bioinformatics* 40.6.

Chapter 5

FUSION GENE AND IMMUNE CELL RECEPTOR SEQUENCE DISCOVERY

5.1 Abstract

Gene fusions and chimeric RNAs are now known to have important biological functions as well as critical in the tumorigenesis of cancers. With the reduction of cost in long read RNA-seq and the advances in liquid biopsies, early and sensitive detection of cancers is becoming possible. However, with these clinical applications, the demand for higher accuracy, not only low false negative rate, but also low false positive rate, gene fusion detection and analysis methods grows. Properly identifying gene fusions for targeted therapies becomes essential. Immune therapies play a significant role in current cancer therapeutics. Thus, the simultaneous study and identification of novel gene fusions and immune receptor sequences in the same RNA-seq sample and with a standard approach to both analyses that is also efficient for both bulk and single-cell analysis would be useful. Here, we present `fugi` a tool to meet this challenge with preliminary results that we believe will continue with full benchmarking.

5.2 Introduction

Gene fusions and chimeric RNAs have been found to play an important role in both healthy and diseased tissue. In particular, gene fusions have been implicated as playing an important and complex role in both tumorigenesis and tumors (Hanahan and Weinberg, 2000) (Stratton, Campbell, and Futreal, 2009) (Mertens et al., 2015). These gene fusions are formed through chromosomal rearrangements and other aberant rearrangements of the genome (Figure 5.1). Long read RNA sequencing in cancer cells and immune cells is a powerful technique to detect chromosomal rearrangements and immune sequence rearrangements, allowing for *de novo* discovery of actively expressed fusion genes and immune cell receptor sequences, which have important implications for personalizing medicine and understanding the fragile sites in the genome that may be particular susceptible to causing cancer (Yunis and Soreng, 1984) (Rowley, 1973) (Chin, Andersen, and Futreal, 2011). Here we focus on the first step of analyzing these important RNA sequences: detecting gene fusions and immune receptor sequences from raw long read RNA sequenc-

ing data. We perform an initial benchmark of gene fusion detection which needs expanded within pizzly (Melsted et al., 2017), initially designed for short reads, to determine associated breakpoints. However, from the initial benchmarking of fusion gene pairs, lr-kallisto+pizzly+extensions (termed fugi for **fu**sion **g**enes and **i**mmune sequence discovery with lr-kallisto) performed 100% accurately with respect to spike-ins, which placed it alongside CTAT-LR-Fusion and pbfusion as the best at detecting gene fusions in ground truth spike-ins (Qin et al., 2025) (Miller et al., 2022). Moreover, when neglecting order of pairs (which should be resolved with full implementation of the workflow within pizzly), fugi achieved greater than .9 mean F1 score, making it comparable to leading gene fusino tools, in simulation data for gene fusion pairs with at least 2% sequencing error.

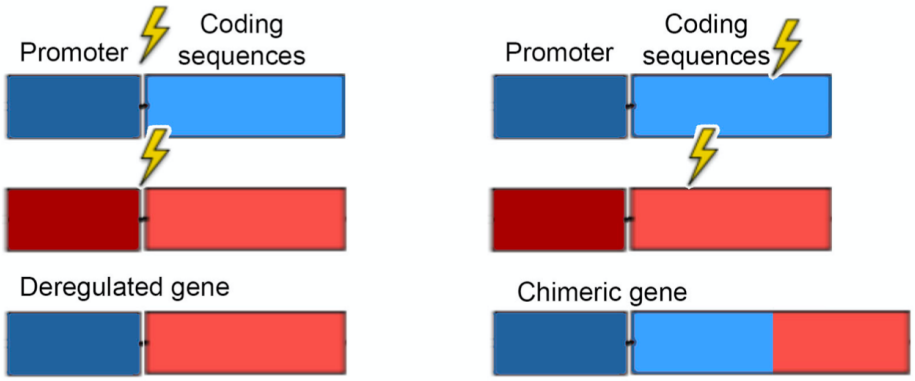


Figure 1: Fusion genes resulting from chromosomal rearrangements.
If the chromosomal rearrangement is made at coding and/or regulatory regions, a deregulated or chimeric gene can occur. Lightening bolt indicates chromosomal breakpoint.

Figure 5.1: pizzly Figure 1 used under CC-BY 4.0 Interntional license.

pizzly and lr-kallisto together then allow for the accurate quantifying of fusion genes abundances. Building on the pseudoalignment idea that simplifies and accelerates transcript quantification and extended both accuracy and efficiency to long read RNA-seq quantification with lr-kallisto, we introduce a novel approach to fusion detection where we inspect reads that do not pseudoalign due to conflicting matches of *k*-mers. We further extend this method to also retrieve immune sequences and immune sequence rearrangements from immune receptors. The method and software, called fugi, filters false positives, assembles new transcripts from the fusion and immune sequence reads, and reports candidate fusions and immune receptor sequences. With fugi, fusion detection and immune sequence analysis with long read RNA-Seq reads can scale to the state-of-the-art datasets. fugi's scalability and accuracy make it suitable for the analysis of large cancer gene expression databases

and for clinical use (Li et al., 2009) (Stransky et al., 2014).

5.3 Results

We first performed benchmarking with positive controls, using the high-quality PacBio spike-in control with SeraCare v4 Fusion Reference Control, fugi detected all spike-in gene fusions in the PacBio monomer dataset, joining only two other tools which achieved this, CTAT-LR-Fusion and pbfusion (Qin et al., 2025) (Miller et al., 2022).

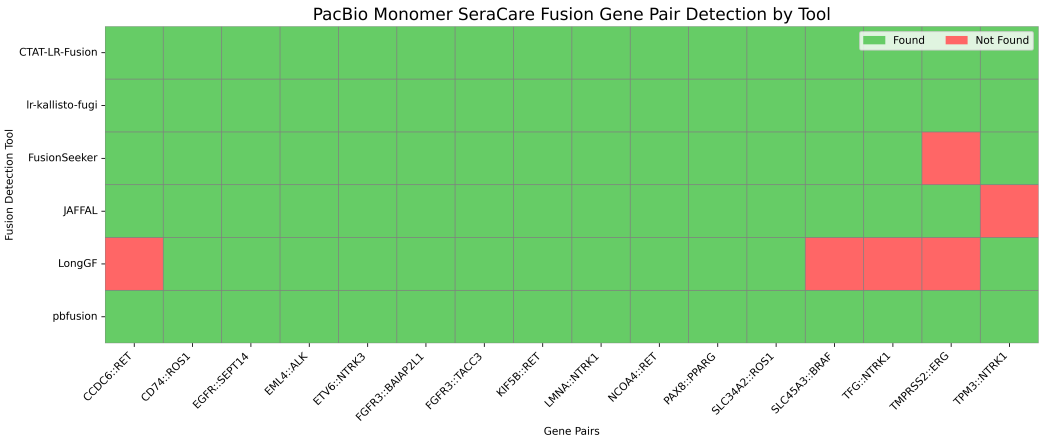


Figure 5.2: Comparison of fusion detection in SeraCare gene fusion spike-in PacBio Monomer dataset. CTAT-LR-Fusion, lr-kallisto-fugi, and pbfusion are able to detect all synthetic fusions. However, JAFFAL and FusionSeeker each miss a single synthetic fusion, while LongGF misses four fusion gene pairs.

We performed benchmarking with data from the CTAT-LR-Fusion paper, including pbsim simulations of HiFi PacBio data and 98% sequencing accuracy ONT data. lr-kallisto `-fusion`, fugi, detected gene fusions with a mean F1 score greater than .9 in ONT and PacBio simulations using gencode annotation version 22 (Figures 5.2 and 5.3).

Candidate TCRs in PacBio HiFi Kinnex PBMCs

Here we include preliminary results for the discovery of T-cell receptor rearrangements in PacBio HiFi Kinnex PBMCs, where we discovered 17,000 molecules with TCR-related regions. The alignments clearly indicate the presence of β molecules and γ molecules with 75% matching in variable regions and 98% in joining regions with the constant regions between the variable and the joining regions not annotated but a gap indicated for them. If a read contains an unannotated splice junction

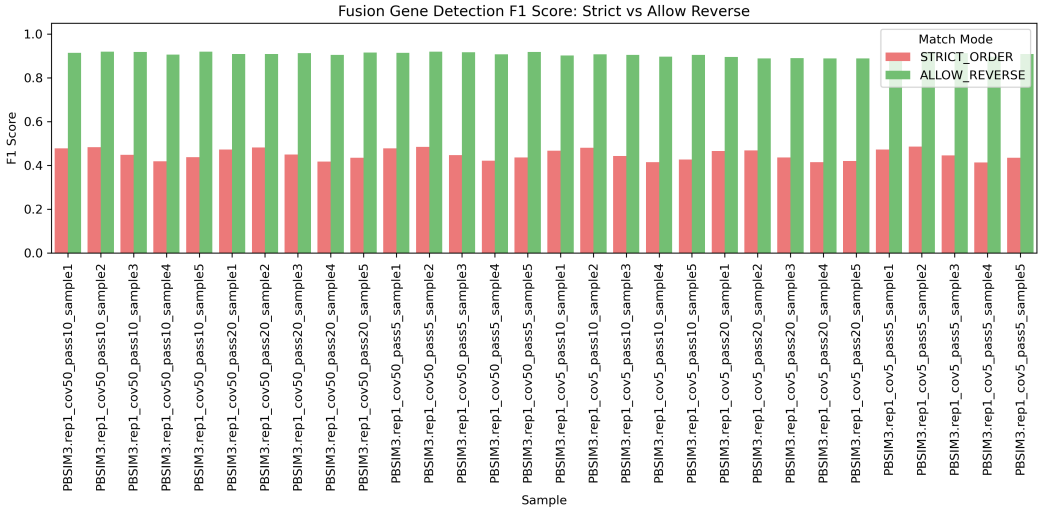


Figure 5.3: Comparison of fusion detection in pbsim simulated gene fusions in HiFi PacBio datasets. lr-kallisto-fugi is able to detect gene fusions with a mean F1 score greater than .9 when allowing reverse; with the addition of pizzly to the workflow, the mean score for strict ordering of gene pairs should increase to greater than .9 mean F1 score as well.

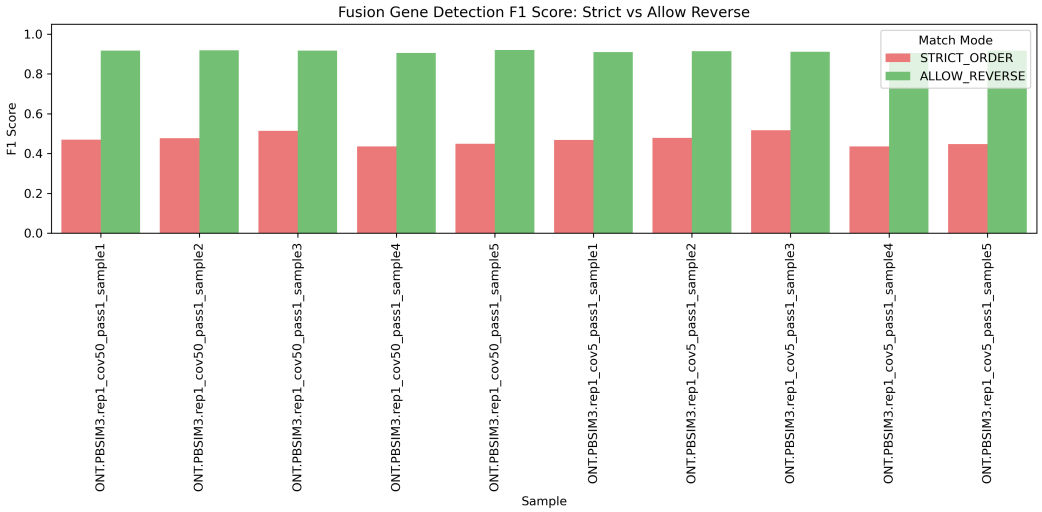


Figure 5.4: Comparison of fusion detection in pbsim simulated gene fusions 98% sequencing accuracy ONT datasets. lr-kallisto-fugi is able to detect gene fusions with a mean F1 score greater than .9 when allowing reverse; with the addition of pizzly to the workflow, the mean score for strict ordering of gene pairs should increase to greater than .9 mean F1 score as well.

it is displayed in read for that pseudoalignment, reads that contain only annotated regions and splices of the genome are shown in blue. On the left hand side of each

plot, we include the transcript name (gene name). On the right hand side of the plot, we include the percentage of the covered portion of the read which is an exact match to the transcript. Thus, the 75% match in variable regions is consistent with these regions being highly recombined for immune function. Further, we also include the length of portion of the read consumed by this match as well as starting position in the read corresponding to matching with the transcript.

Method for Candidate TCRs We begin by extracting the TCR gene sequences from the reference annotation. We then build an index from the TCR gene sequences which also d-lists the rest of the genome. We then extract the sequences which pseudoalign to the TCR gene index.

```
1
2 cat gencode.v48.chr_patch_hapl_scaff.annotation.gtf | grep "
    gene_type \"TR_" > TR_genes.gtf
3
4 kb ref -k 47 -i TR_genes_k-47.idx -g TR_genes_k-47.t2g -f1
    TR_genes.cdna.fa GRCh38.p14.genome.fa.gz TR_genes.gtf --opt-
    off --overwrite
5
6 kb extract --extract_all_fast -i TR_genes_k-47.idx -g TR_genes_k
    -47.t2g -o PBMC_HiFi_3prime_TR_genes_k-47 PBMC_HiFi_3prime.
    fastq -k 47 --opt-off
```

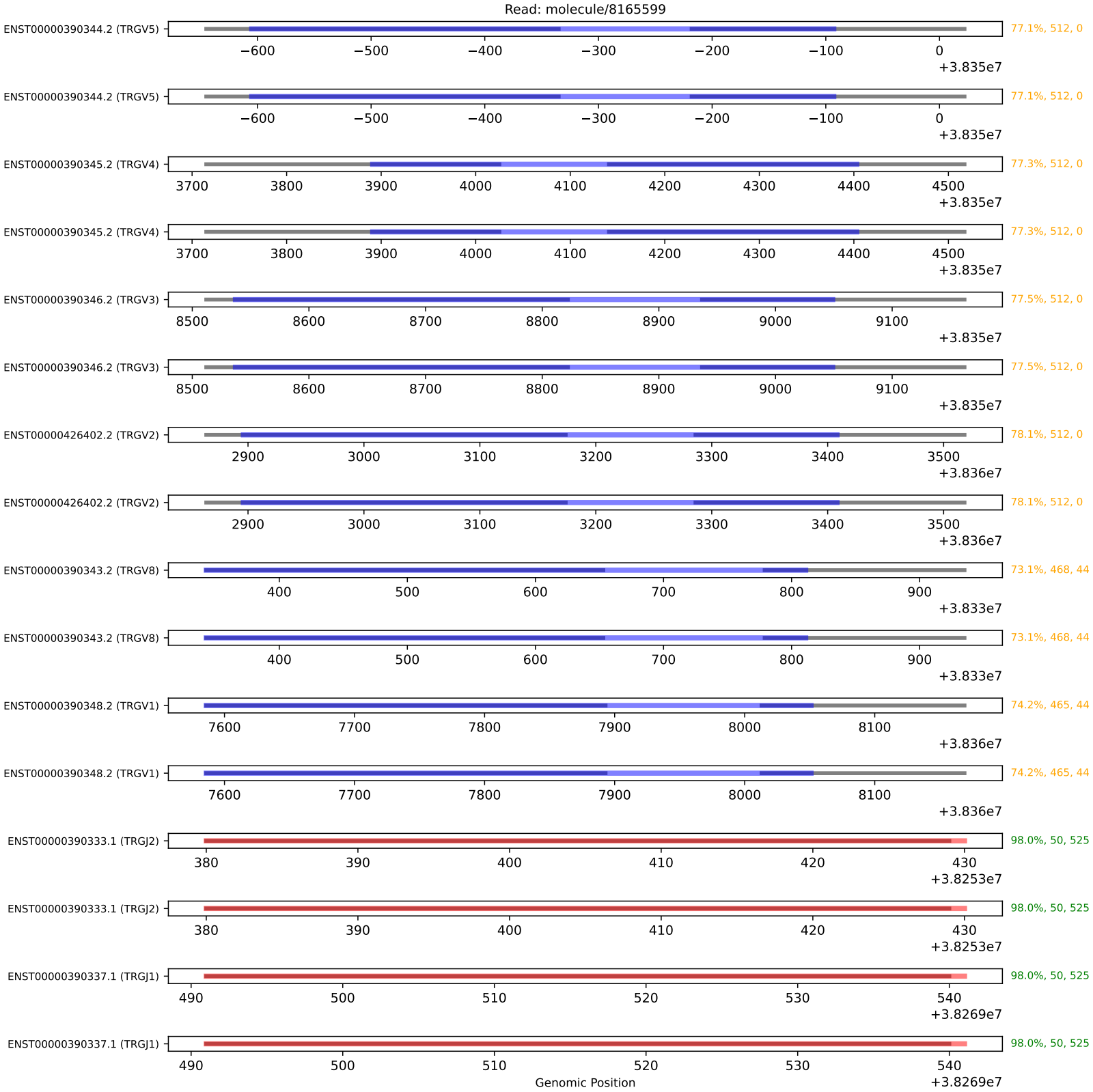
Listing 5.1: building TCR genes indices and extracting TCR sequences

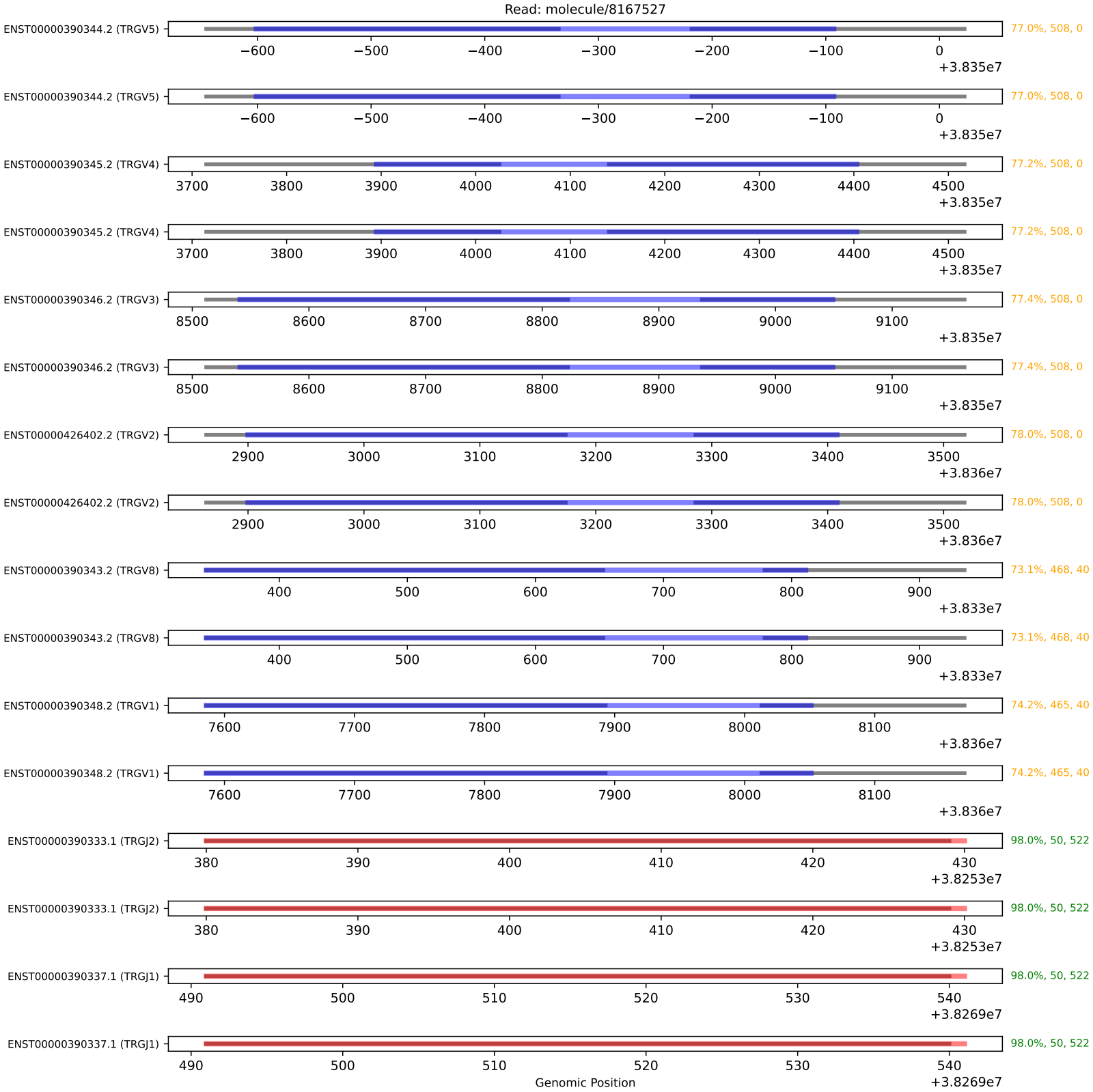
We can now pseudoalign with pseudobam and fusion options the sequences that were extracted as pseudoaligning to the TCR genes.

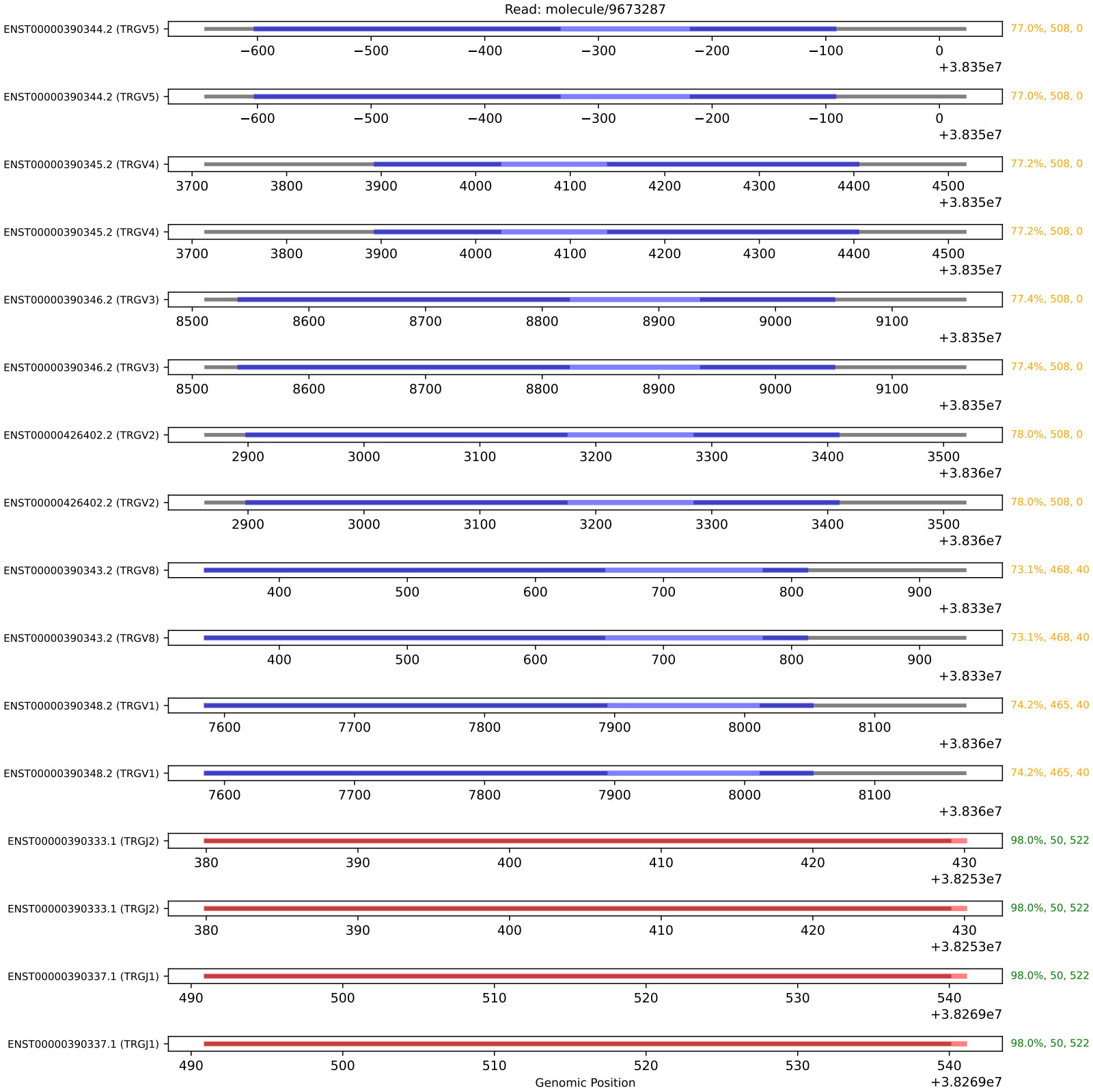
```
1 kallisto_long_pseudobam/kallisto/build/src/kallisto quant -i
    TR_genes_k-11.idx --long --fusion --union --pseudobam -o
    PBMC_HiFi_3prime_TCR PBMC_HiFi_3prime_TR_genes_k-45/all/1.
    fastq.gz
```

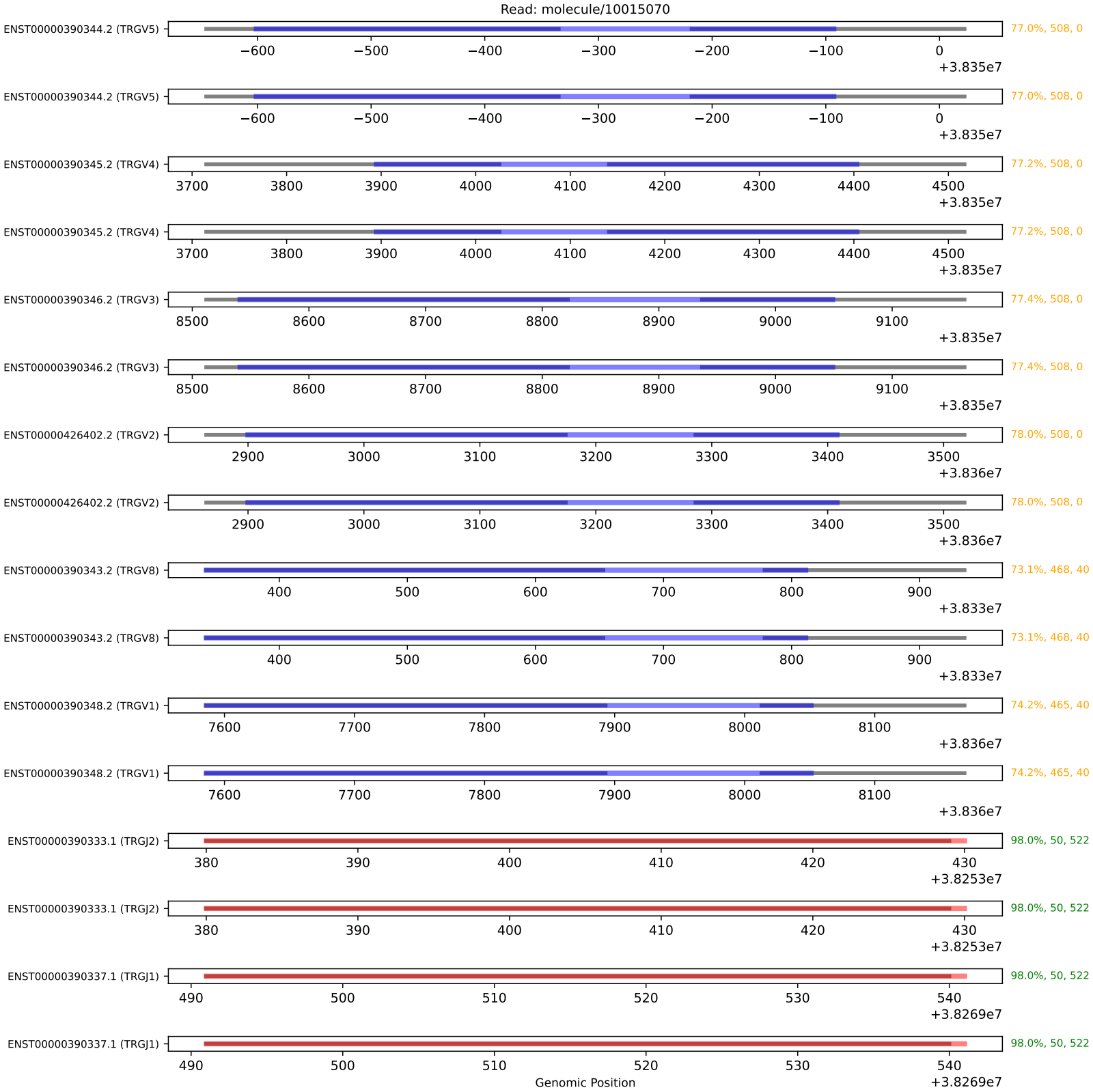
Listing 5.2: using lr-kallisto + pseudobam + fusion to map TCRs

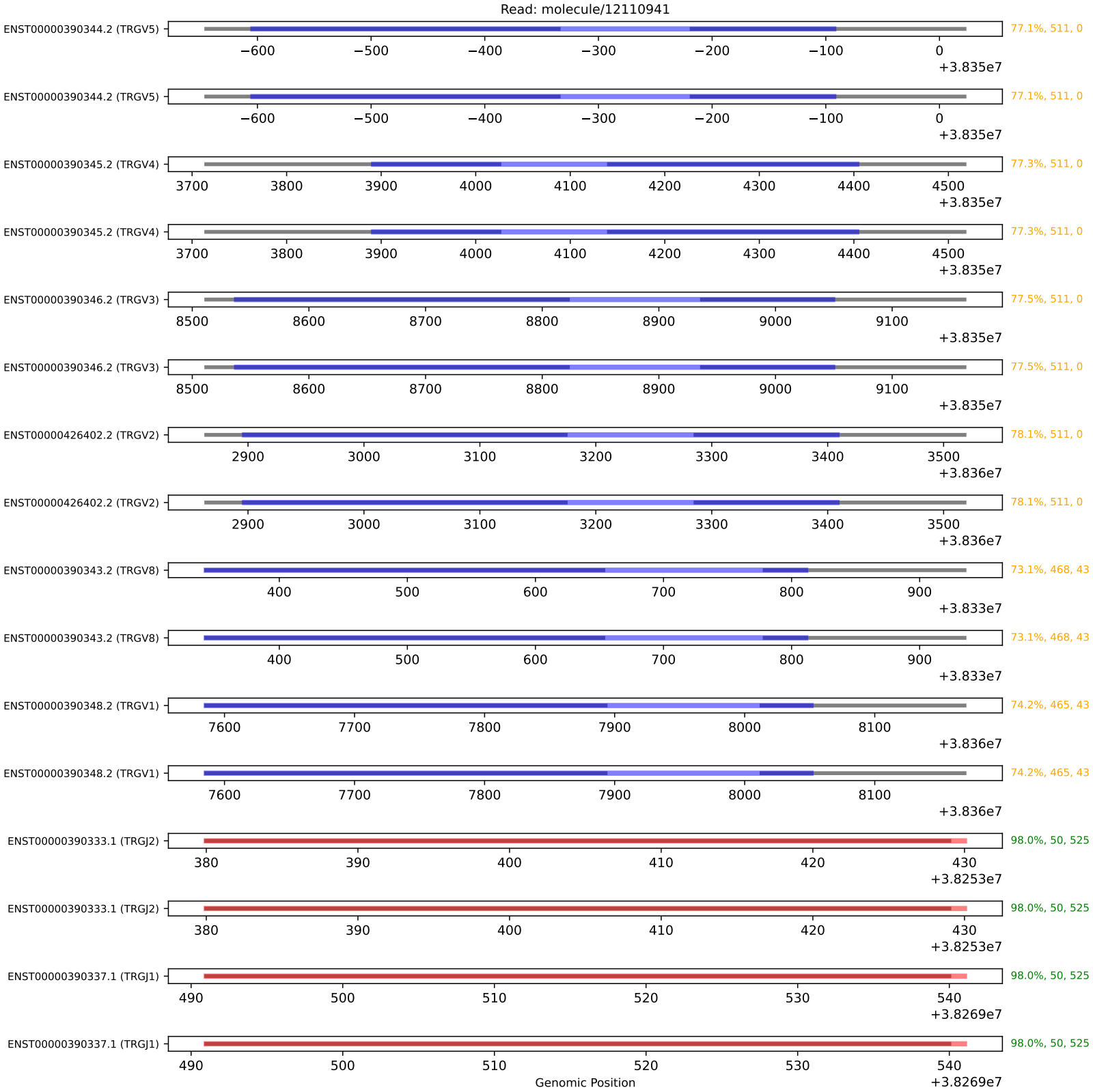
Below we include a subset of the pages of TCR results generated by the workflow included in the code at `https://github.com/bound-to-love/fugi.git` for the PacBio HiFi Kinnex data we analyzed (see Data and Code Availability).

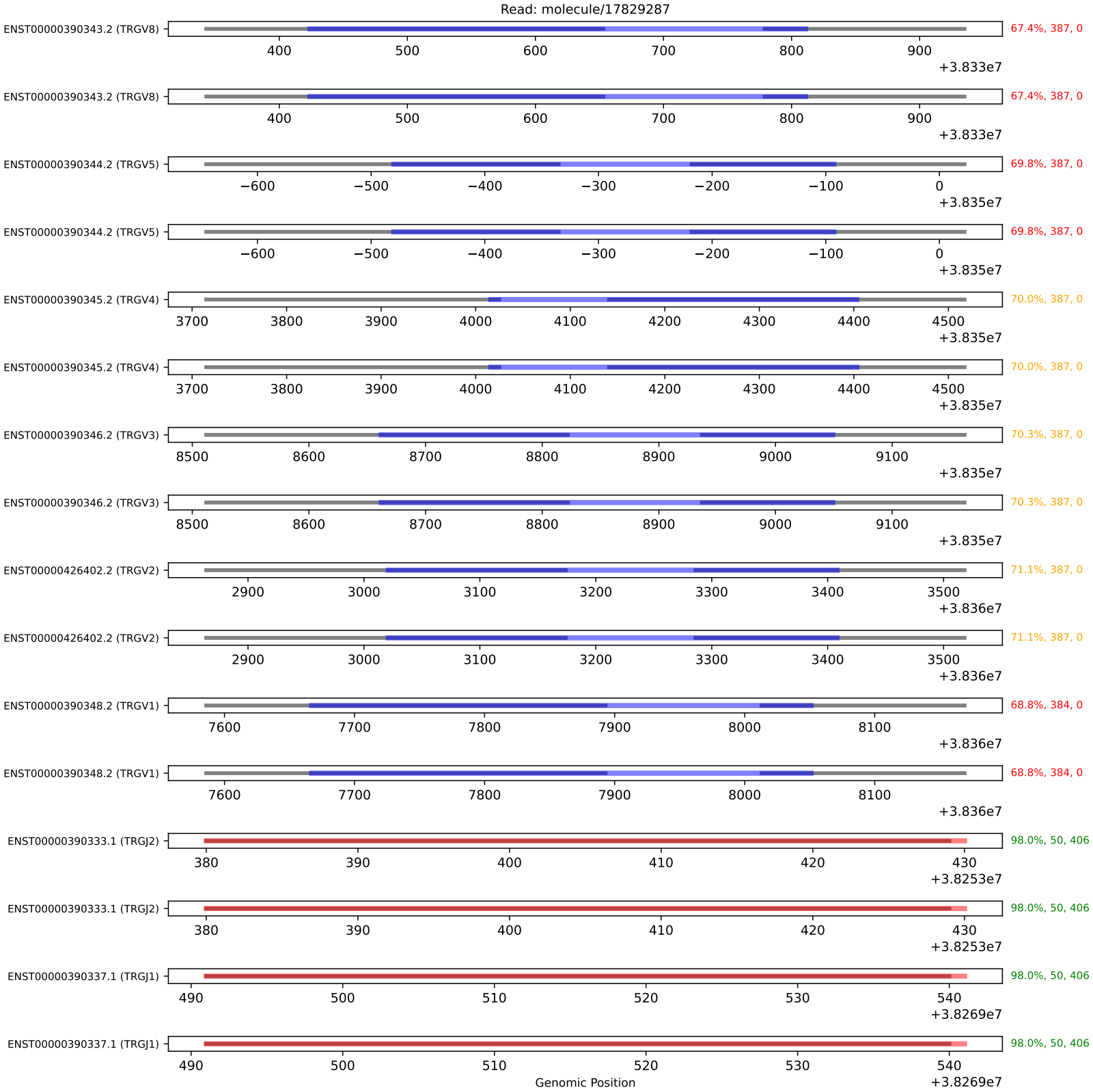

Read: molecule/8165599
ENST00000390344.2 (TRGV5)
77.1%, 512, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.1%, 512, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 512, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 512, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 512, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 512, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.1%, 512, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.1%, 512, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 44
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 44
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 44
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 44
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 525
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 525
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 525
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 525
490
500
510
520
530
540
+3.8269e7
Genomic Position

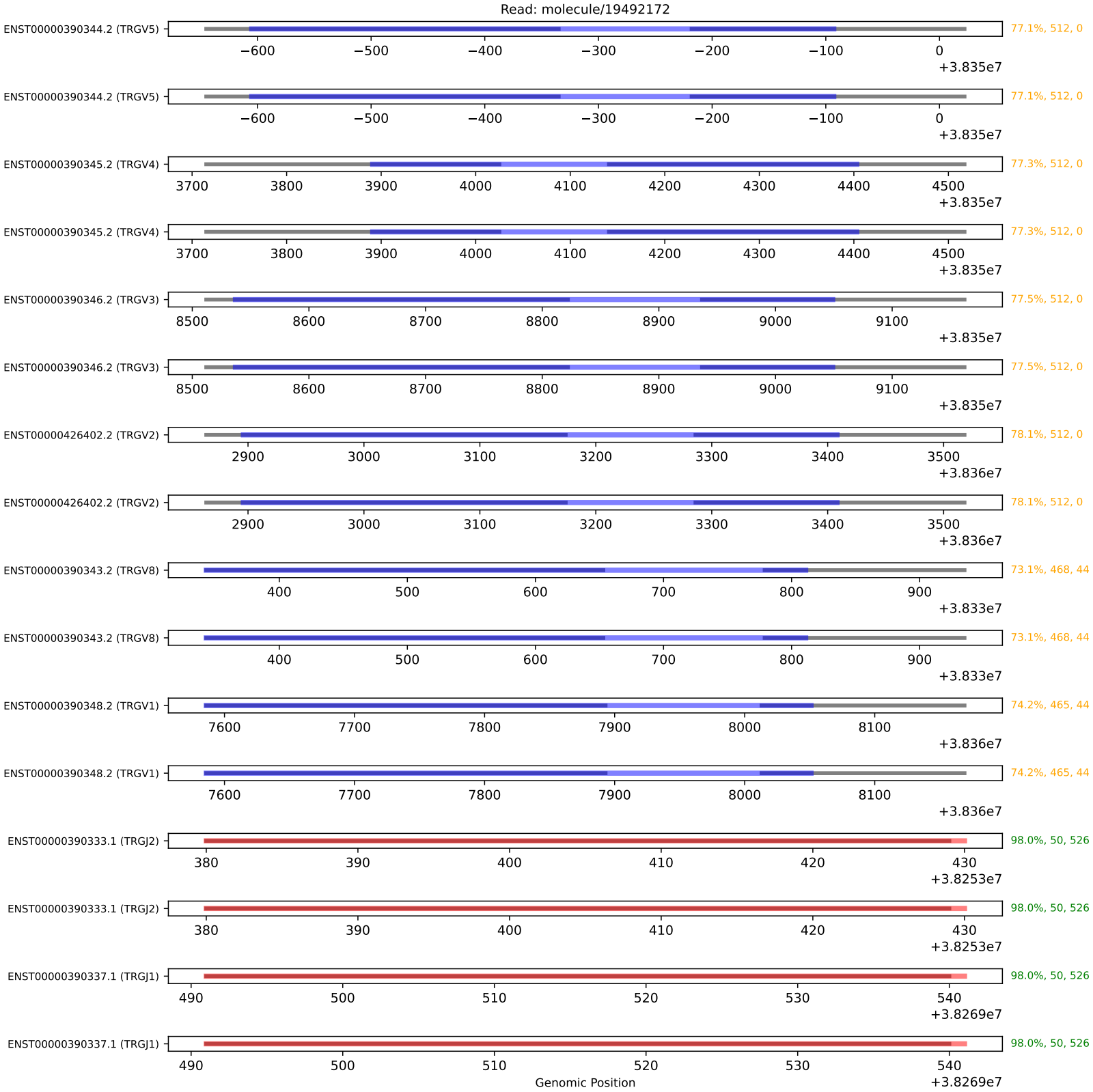

Read: molecule/8167527
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
Genomic Position

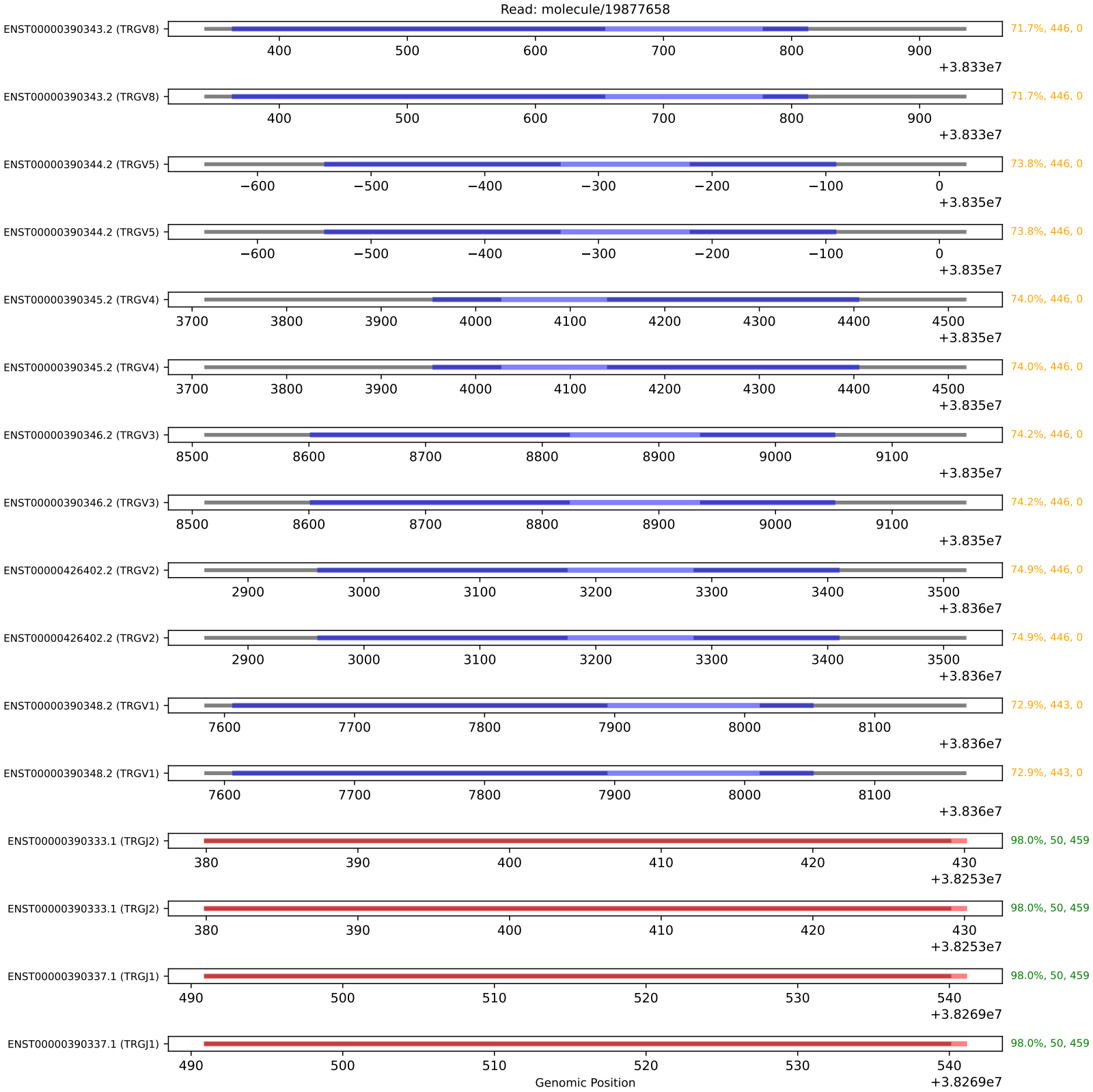

Read: molecule/9673287
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
Genomic Position

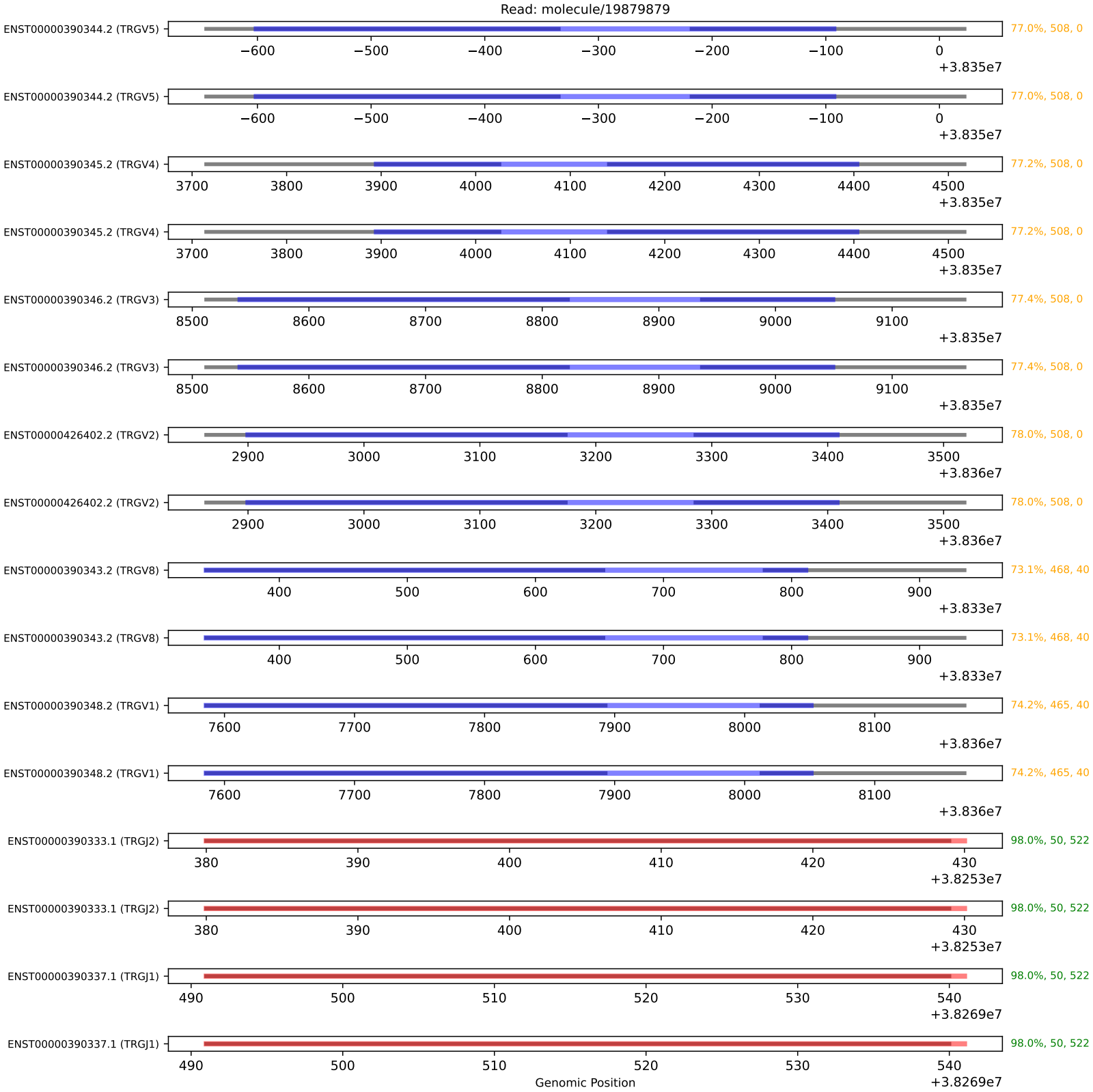

Read: molecule/10015070
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
Genomic Position

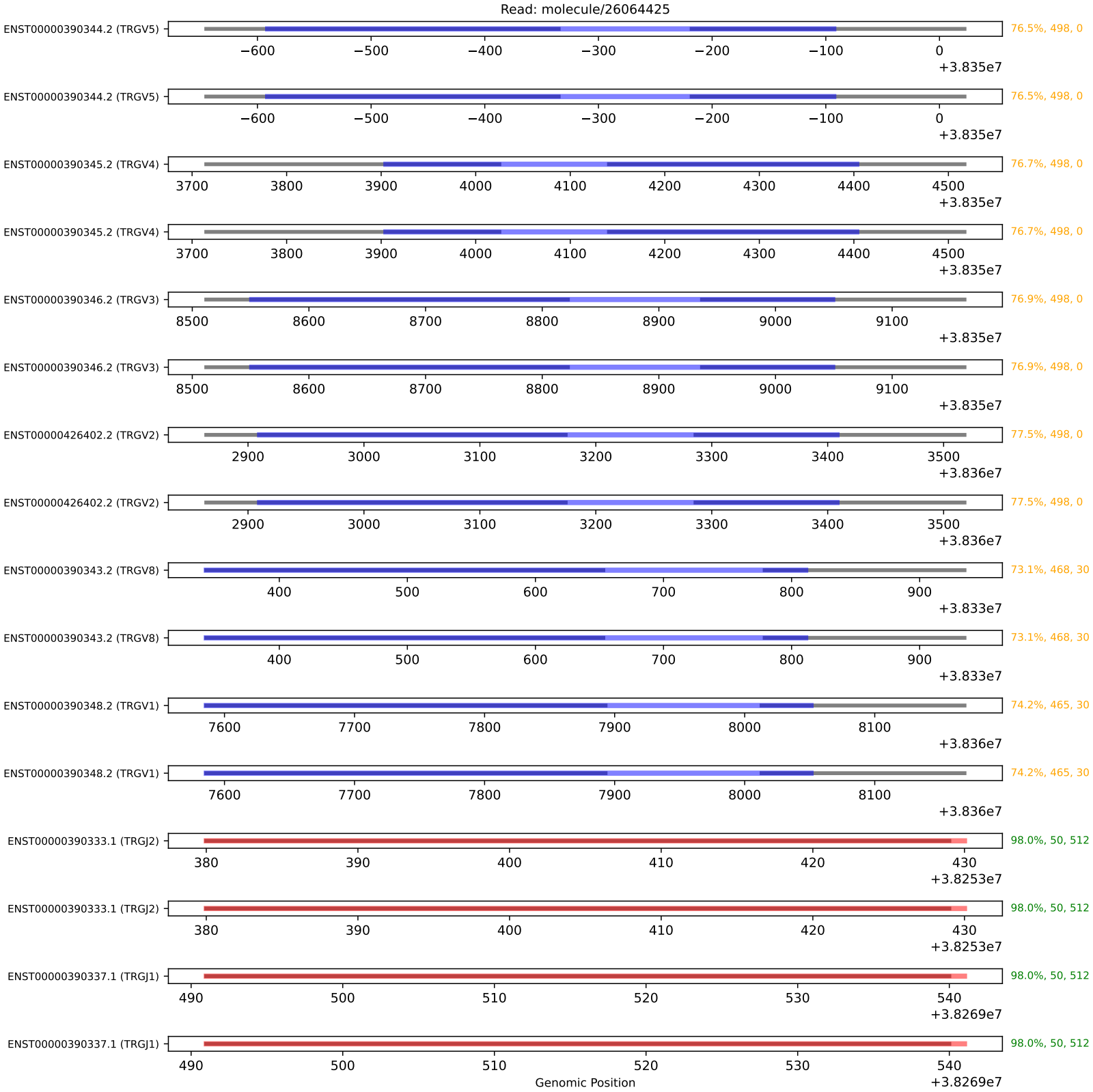

Read: molecule/12110941
ENST00000390344.2 (TRGV5)
77.1%, 511, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.1%, 511, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 511, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 511, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 511, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 511, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.1%, 511, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.1%, 511, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 43
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 43
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 43
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 43
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 525
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 525
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 525
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 525
490
500
510
520
530
540
+3.8269e7
Genomic Position


Read: molecule/17829287
ENST00000390343.2 (TRGV8)
67.4%, 387, 0
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
67.4%, 387, 0
400
500
600
700
800
900
+3.833e7
ENST00000390344.2 (TRGV5)
69.8%, 387, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
69.8%, 387, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
70.0%, 387, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
70.0%, 387, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
70.3%, 387, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
70.3%, 387, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
71.1%, 387, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
71.1%, 387, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390348.2 (TRGV1)
68.8%, 384, 0
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
68.8%, 384, 0
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 406
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 406
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 406
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 406
490
500
510
520
530
540
+3.8269e7
Genomic Position


Read: molecule/19492172
ENST00000390344.2 (TRGV5)
77.1%, 512, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.1%, 512, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 512, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.3%, 512, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 512, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.5%, 512, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.1%, 512, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.1%, 512, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 44
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 44
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 44
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 44
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 526
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 526
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 526
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 526
490
500
510
520
530
540
+3.8269e7
Genomic Position


Read: molecule/19877658
ENST00000390343.2 (TRGV8)
71.7%, 446, 0
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
71.7%, 446, 0
400
500
600
700
800
900
+3.833e7
ENST00000390344.2 (TRGV5)
73.8%, 446, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
73.8%, 446, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
74.0%, 446, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
74.0%, 446, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
74.2%, 446, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
74.2%, 446, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
74.9%, 446, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
74.9%, 446, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390348.2 (TRGV1)
72.9%, 443, 0
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
72.9%, 443, 0
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 459
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 459
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 459
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 459
490
500
510
520
530
540
+3.8269e7
Genomic Position


Read: molecule/19879879
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
77.0%, 508, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
77.2%, 508, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
77.4%, 508, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
78.0%, 508, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 40
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 40
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 522
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 522
490
500
510
520
530
540
+3.8269e7
Genomic Position


Read: molecule/26064425
ENST00000390344.2 (TRGV5)
76.5%, 498, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390344.2 (TRGV5)
76.5%, 498, 0
−600
−500
−400
−300
−200
−100
0
+3.835e7
ENST00000390345.2 (TRGV4)
76.7%, 498, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390345.2 (TRGV4)
76.7%, 498, 0
3700
3800
3900
4000
4100
4200
4300
4400
4500
+3.835e7
ENST00000390346.2 (TRGV3)
76.9%, 498, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000390346.2 (TRGV3)
76.9%, 498, 0
8500
8600
8700
8800
8900
9000
9100
+3.835e7
ENST00000426402.2 (TRGV2)
77.5%, 498, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000426402.2 (TRGV2)
77.5%, 498, 0
2900
3000
3100
3200
3300
3400
3500
+3.836e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 30
400
500
600
700
800
900
+3.833e7
ENST00000390343.2 (TRGV8)
73.1%, 468, 30
400
500
600
700
800
900
+3.833e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 30
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390348.2 (TRGV1)
74.2%, 465, 30
7600
7700
7800
7900
8000
8100
+3.836e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 512
380
390
400
410
420
430
+3.8253e7
ENST00000390333.1 (TRGJ2)
98.0%, 50, 512
380
390
400
410
420
430
+3.8253e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 512
490
500
510
520
530
540
+3.8269e7
ENST00000390337.1 (TRGJ1)
98.0%, 50, 512
490
500
510
520
530
540
+3.8269e7
Genomic Position

5.4 Methods

Here we describe the implementation of fusion detection in lr-kallisto and pizzly for long reads, which we term `fugi` (Loving et al., 2024; Melsted et al., 2017). First the fusions are detected with lr-kallisto. Since a fusion presents in a long read as possessing ends mapping to different genes, lr-kallisto can detect these fusions when the segment of the read belonging to each gene map but the intersection is empty. The occurrence of all mapping *k*-mers overlapping between the two genes in the fusion gene is possible, but highly unlikely. pizzly then takes the fusions output by lr-kallisto in the `fusion.txt` and performs an annotation and genome aware analysis of the associated reads by performing alignment of the read across the detected junctions. The annotation aware aspect allows pizzly to identify potential false positives detected from repetitive regions in the genome.

lr-kallisto –fusion stage

The fugi algorithm - fugi is built within the lr-kallisto framework and, thus, builds on the transcript de Bruijn Graph (t-DBG) and pseudoalignment methods within kallisto, extending the enhancements already put in place during the development of pizzly for fusion detection via pseudoalignment. In the development of pizzly, kallisto's k-mer based index for the reference transcriptome was exploited for the first time to detect candidate fusions when computing a pseudoalignment for reads, if a read did not pseudoalign (Figure 5.6). In kallisto, for each *k*-mer, the index is a hash map that records the set of transcripts containing this *k*-mer, called the transcript compatibility class (TCC). In general, matching *k*-mers are supported by at least one transcript which is the intersection of the TCCs for the read, causing the intersection to be non-empty and contain the true transcript for the read. In the case of long reads, this is not always true, due to the higher prevalence of sequencing errors. Thus, when the intersection is empty, we use the mode of mapping *k*-mers instead of the intersection.

However, for long reads when we are in fusion detection mode, we will again use the intersection instead of the mode for the fusion detection stage. To perform fusion detection with lr-kallisto, we must again handle reads where the intersection of mapping k-mers is empty. However, just as in lr-kallisto where the mapping needed handled slightly differently for long reads vs for short reads, with long read fusion detection the mapping also must be handled slightly differently. In these instances, we must search the entirety of the read for mapping *k*-mers instead of using the jumping procedure used in short reads, searching the read for disjoint regions which

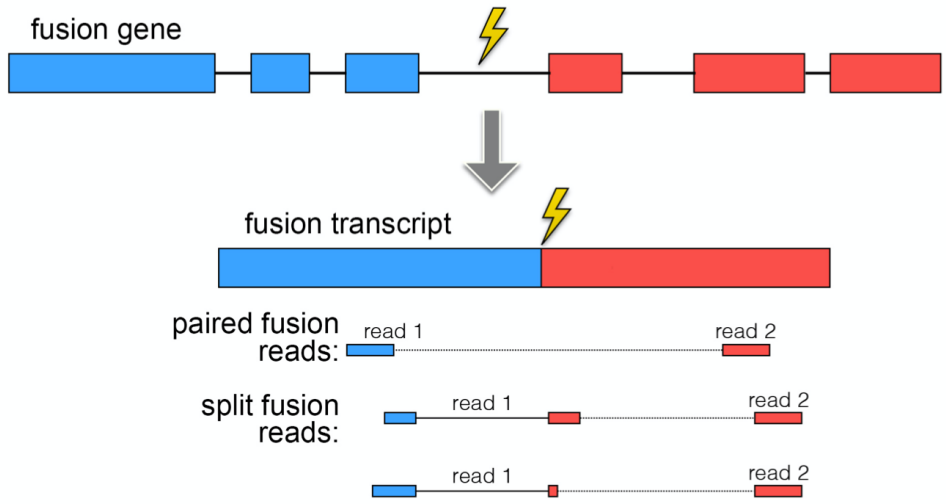


Figure 2: Reads meeting specific criteria are passed to pizzly.
A fusion gene resulting from chromosomal rearrangement, is transcribed into a fusion transcript. Standard kallisto does not pseudoalign reads spanning the fusion junction to the transcriptome. In the fusion-modified kallisto, reads meeting specific criteria are reported as candidate fusions for use with pizzly.

Figure 5.5: pizzly Figure 2 used under CC-BY 4.0 Interntional license.

map with non-empty intersections. We accomplish this by using subsequences of the read, jumping by the length of the subsequences which map to a contig. We then restart the search at the end of the mapping region until we reach the end of the read. In the case of fusion searches, we are looking for a single break in the read which divides it into two mapping regions, mapped to disjoint transcript compatibility classes. After the initial mapping of the unmapped reads to find fusion reads, we can then filter the produced pseudobam to increase our true positive rate and decrease our false positive rate or we can apply pizzly, which we describe next. In the case of pizzly, for reads or read pairs that span a fusion breakpoint, the TCCs from each side of the breakpoint will again have a non-empty TCC intersection, but the intersection of these non-empty TCCs would be empty since they are TCCs for different genes. Therefore, these reads would typically be discarded from further consideration by kallisto. However, when running lr-kallisto or kallisto in fusion finding mode, kallisto identifies reads and read pairs whose intersection of TCCs is empty, where there are mapping k-mers. These reads become fusion candidates which are output to `fusion.txt` when either one of the following holds:

- (1, for paired-end reads): each read has a non-empty TCC intersection separately, but combined the intersection of the paired-end reads is empty (as

shown in Figure “paired fusion reads,” where the reads come from opposite sides of the fusion junction) (Figure 5.6)

- (2, for long read or split case): one of the reads or the long read can be split into two parts such that the first part of the read has a non-empty TCC intersection and the remainder of the read, along with the other read from the pair, has a non-empty EC intersection (this is consistent with Figure “split fusion reads,” where one of the reads spans the fusion junction). When a potential split of the reads has been identified, kallisto checks all matching *k*-mers and requires that the intersection of the union of TCCs on either side is empty. This last step lowers the false positive rate that can be increased by reads from unannotated transcripts resembling fusions between related transcripts. (Figure 5.6)

All read pairs matching these criteria are saved along with supporting information about the matching transcripts.

pizzly processing of `fusion.txt`

pizzly can now be applied to fusion.txt to perform a genome annotation aware analysis of the candidate fusion breakpoints. pizzly uses a transcriptome annotation in the form of a GTF, which includes functional annotation of genes. For paired-end reads, the input to pizzly is the set of read pairs that kallisto detected as candidate fusions and, for long reads or single-end reads, the input is the read.

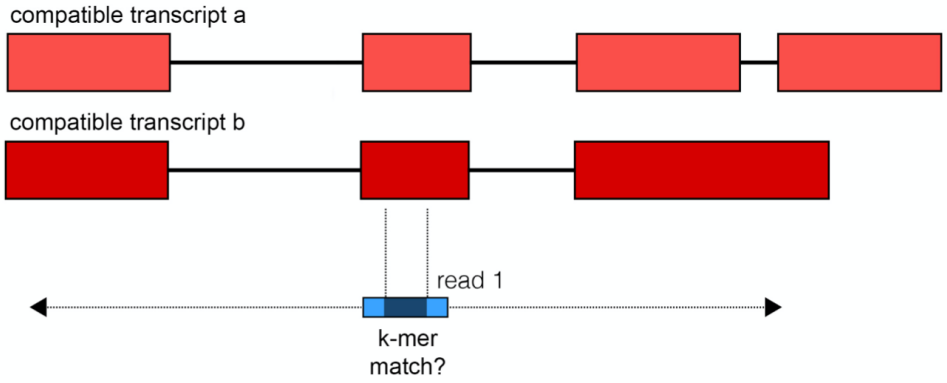


Figure 3: A check for alignment to partner transcript ensures correct fusion.
The *k*-mers from the first part of the candidate fusion, the blue “read 1”, are aligned with mismatch allowance to all compatible transcripts from the second part of the candidate fusion. If a match is found, this false positive is discarded. The same is repeated for the other end of the candidate fusion.

Figure 5.6: pizzly Figure 3 used under CC-BY 4.0 Interntional license.

The first step pizzly performs is evaluating each read independently and rejecting false positives, including reads that map to multiple genomic locations (Figure 5.7). kallisto and lr-kallisto can also produce false positives that are due to incomplete annotations, where two distinct transcripts from the same gene are mapped to by kallisto in a disjoint manner that is due to a novel transcript in the data which alternatively splices from what is recorded in the annotation. These reads are discarded from the pizzly analysis. Another class of false positive candidate fusion can be created by sequencing error (especially in the case of long reads) or by SNPs or other mutations that create small mismatches which result in the mapping of a *k*-mer to a similar gene that is disjoint in TCC from the rest of the read. pizzly filters these reads not through annotated gene families or groupings, but through approximate sequence alignment filtering. Matching *k*-mers from each end of the read are considered, where instead of considering approximate matches to the whole genome, only the listed TCCs for each end of the read need to be considered. If approximate matches are found between the sequences of the transcripts within the TCCs between the two ends, then the read is discarded as a false positive.

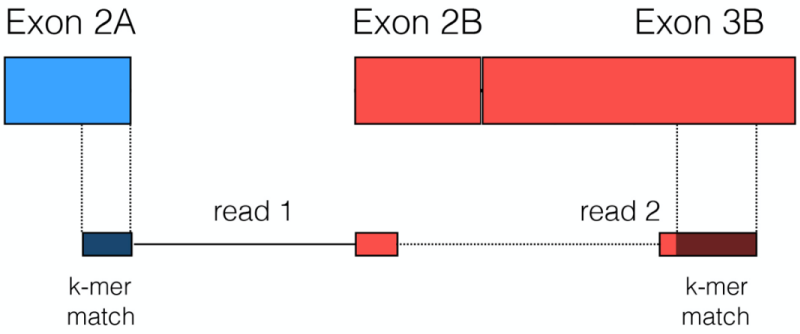


Figure 4: A read can be split-aligned between the two fusion transcripts.
Split-read alignments provide breakpoint sequence resolution.

Figure 5.7: pizzly Figure 4 used under CC-BY 4.0 Interntional license.

Next, on either side of the candidate fusion breakpoint, the read or reads must fully align with their transcript or gene of origin.

After this filtering, the candidate fusions are aggregated on a gene-to-gene fusion level. The predicted fusions are filtered according to the number of reads supporting the fusion breakpoint. For split reads (which includes all long reads), the fusion breakpoint must be within 10 bp of an exon boundary in both genes (Figure 5.8). For fusion breakpoints that are supported by pairs in paired-end reads, the distance to the nearest internal exon boundary must be consistent with the insert length provided

to pizzly. After this final filtering, pizzly reports the number of paired and split reads supporting each fusion breakpoint as well as each potential transcript fusion breakpoint, the number of reads supporting the transcript-level fusion as well as the sequence of the fused transcripts and the original reads which support it.

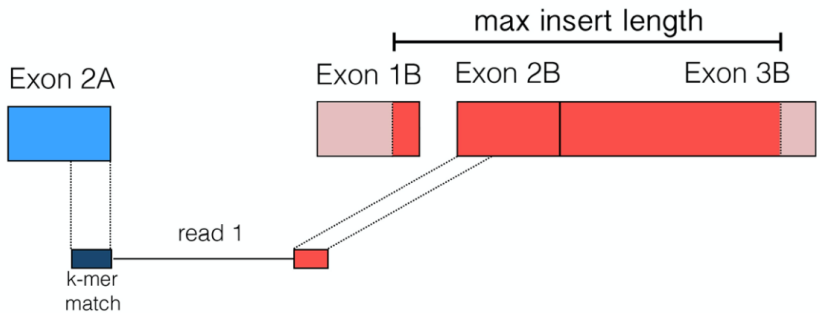


Figure 5: Split alignment search.
The search space for split alignments is limited to the matching part of the compatible transcript up to the fragment insert length.

Figure 5.8: pizzly Figure 5 used under CC-BY 4.0 Interntional license.

In the case of immune cell receptor sequence discovery, we are taking reads which map to the immune cell receptor sequences and then mapping them with pseudobam to find the rearrangements. Here we programmatically look for repeated breaks that divide the receptor sequence into the rearrangement pattern that occurred to create the unique receptor sequence using the union of disjoint regions of *k*-mer TCCs.

5.5 Discussion

The accurate and efficient search for immune receptor sequences and gene fusions in long reads is still a significantly unsolved challenge with CTAT-LR-fusion (Qin et al., 2025) coming the closest thus far. While fugi is a work in progress, we have clearly demonstrated its utility to the problem of gene fusions and immune cell receptor sequence discovery. Moreover, the same methodological approach within *k*-mer matches is proving widely useful in a previously unexploited algorithmic approach to mapping long reads across *de novo* transcript discovery, fusion gene discovery, and immune cell receptor discovery. This simple algorithm maps using the non-empty intersections of “region" compatible segments of the molecule to find the sequences of origin for V-D-J combinations, unannotated transcripts, or other genomic rearrangements. While some of the non-empty intersections may be spurious, spurious non-empty intersections will not have the read count support for further analysis.

5.6 Future Directions and Development

While this work needs further development, the current results demonstrate state-of-the-art performance in gene fusion detection and promise in retrieving and annotating immune receptor sequences, especially with the extension of improved immune sequence annotation and addition of `klue`, an unpublished tool for distinguishing unique *k*-mers and *de novo* assembly of t-DBG, to the workflow for improved clonality analysis. Due to the building of `fugi` on lr-kallisto and pizzly, this work will naturally extend to single-cell resolution. Thus, the immune sequence discovery application will provide an unbiased, broad analysis workflow for immune response in healthy and diseased scenarios to increase our knowledge of building immunity and healthy immune response as well as the drivers behind autoimmune diseases and cancers with joint analysis of gene fusions and immune response facilitated by `fugi` with non-specific RNA-seq library preparation of liquid or tissue biopsies.

5.7 Data and Code Availability

```
README  (Last Updated 02/20/2024)
from \hyperref{https://downloads.pacbcloud.com/public/dataset/
   Kinnex-single-cell-RNA/}
********************
INTRODUCTION
********************

This README file describes the contents in this directory.

The dataset generated here contains single-cell RNA-Seq data
   generated
using the MAS-Seq for 10x Single Cell 3' kit ("MAS") [1] and the
KinnexTM single-cell RNA kit ("Kinnex") [2].

The MAS-Seq libraries were sequenced on the Sequel  II/IIe and
   Revio
systems and processed using SMRT  Link v11.1 [3] or BioConda [4].

The Kinnex libraries were sequenced on the Revio system and
   processed using
SMRT Link v13.1 [5].

To learn more about Kinnex, visit: https://pacb.com/kinnex


********************
SAMPLE
********************

All PBMC samples were purchased from BioIVT. Either fresh or
   cryopreserved.

All HG002/GM24385 10k cells were purchased from Coriell.

All cDNA libraries were generated using the 10x Chromium Next GEM
Single Cell 3    kit (v3.1) or Single Cell 5' kit (v2) with a 10x
   Chromium
Next GEM Chip G on a 10x Chromium X system.

Below is a description of the kits, systems, samples used for each
    directory.

```

```
DATA-Revio-Kinnex-PBMC-10x3p : Kinnex kit, Revio, PBMC, 10x 3' kit

https://downloads.pacbcloud.com/public/dataset/Kinnex-single-cell-
    RNA/DATA-Revio-Kinnex-PBMC-10x3p/2-DeduplicatedReads/scisoseq
    .5p--3p.tagged.refined.corrected.sorted.dedup.bam

bedtools bamtofastq -i scisoseq.5p--3p.tagged.refined.corrected.
    sorted.dedup.bam -fq PBMC_HiFi_3prime.fastq
```

Listing 5.3: Description of PBMC PacBio HiFi Kinnex data and bedtools conversion command for bamtofastq

The code for `fugi` is available at `https://github.com/bound-to-love/fugi.git`.

References

Chin, Lynda, Jason N Andersen, and P Andrew Futreal (2011). "Cancer genomics: from discovery science to personalized medicine". In: *Nature Medicine* 17.3, pp. 297–303.

Hanahan, Douglas and Robert A Weinberg (2000). "The Hallmarks of Cancer". In: *Cell* 100.1, pp. 57–70.

Li, Hao et al. (2009). "Gene fusions and RNA trans-splicing in normal and neoplastic human cells". In: *Cell Cycle* 8.2, pp. 218–222.

Loving, Rebekah K et al. (2024). "Long-read sequencing transcriptome quantification with lr-kallisto". In: *bioRxiv v1*, pp. 2024–07.

Melsted, Páll et al. (July 2017). "Fusion detection and quantification by pseudoalignment".

Mertens, F. et al. (2015). "The emerging complexity of gene fusions in cancer". In: *Nature Reviews Cancer* 15.6, pp. 371–381. DOI: 10.1038/nrc3947. URL: https://doi.org/10.1038/nrc3947.

Miller, Anthony R. et al. (2022). "Pacific Biosciences Fusion and Long Isoform Pipeline for Cancer Transcriptome–Based Resolution of Isoform Complexity". In: *The Journal of Molecular Diagnostics* 24.12, pp. 1292–1306. ISSN: 1525-1578. DOI: https://doi.org/10.1016/j.jmoldx.2022.09.003. URL: https://www.sciencedirect.com/science/article/pii/S1525157822002653.

Qin, Qian et al. (2025). "CTAT-LR-fusion: accurate fusion transcript identification from long and short read isoform sequencing at bulk or single cell resolution". In: *Genome Research* 35.4. Originally published as a bioRxiv preprint: doi:10.1101/2024.02.24.581862, pp. 967–986. DOI: 10.1101/gr.279200.124. URL: https://doi.org/10.1101/gr.279200.124.

Rowley, Janet D (1973). “A new consistent chromosomal abnormality in chronic myelogenous leukaemia identified by quinacrine fluorescence and Giemsa staining”. In: *Nature* 243.5405, pp. 290–293.

Stransky, N. et al. (2014). “The landscape of kinase fusions in cancer”. In: *Nature Communications* 5, p. 4846. DOI: 10.1038/ncomms5846. URL: https://doi.org/10.1038/ncomms5846.

Stratton, Michael R, Peter J Campbell, and P Andrew Futreal (2009). “The cancer genome”. In: *Nature* 458.7239, pp. 719–724.

Yunis, John J and Alan L Soreng (1984). “Constitutive fragile sites and cancer”. In: *Science* 226.4673, pp. 1199–1204.

Chapter 6

SUMMARY AND FUTURE DIRECTIONS

6.1 Summary

In this thesis, we have introduced three tools for the analysis of long read RNA-seq transcriptomics: `lr-kallisto` for mapping and quantification, `dn-kallisto` for *de novo* transcriptomics, and lr-kallisto+pizzly (`fugi`) for fusion detection and immune receptor sequence discovery in long read RNA-seq. Each of these tools offers a scalable and high fidelity approach to a challenge in long read RNA-seq analysis. In each case, we have leveraged the advantages of pseudoalignment with adaptations for long reads, providing innovations to the methodology that provide solutions to important biological problems. We have confirmed that, just as the application of pseudoalignment in short reads leads to scalability and efficiency without sacrificing accuracy, with long-reads scalability is achieved and accuracy is equivalent, and in some cases even improved in the new lower sequencing error rate generation of newer long read RNA-sequencing technology.

6.2 Discussion

Recently, there have been significant technological advances in long read RNA-sequencing increasing the length and quality of reads and improving the machine learning basecalling algorithms leading to higher sequencing accuracy in addition to improvements due to advances in chemistry used and consensus sequencing. The application of these workflows presented in this thesis to both existing data and new data will have broad impacts across different fields of study as long read RNA-sequencing is used across various settings from clinical research to developmental biology to metagenomics.

6.3 Future directions

Future directions for `lr-kallisto`, `dn-kallisto`, and `fugi` include:

1. Applying lr-kallisto to analysis of both single-cell and bulk datasets to perform more sensitive and accurate differential transcript usage analysis than was previously possible. This has already started to occur with long read transcriptomics and the application of transcript discovery tools and `lr-kallisto`

revealing ancestral bias in the genome annotation (Clavell-Revelles et al., 2025). `lr-kallisto` will be highly useful in the reanalysis of long read RNA-seq in a uniform workflow for true comparison of differential transcript usage and for transcript discovery in large datasets that many long read tools have difficulty scaling to.

2. While `dn`-kallisto requires further benchmarking, we have, however, shown the usefulness of pseudoalignment for reference-guided transcript discovery. This approach yields both efficiency and confidence advantages to traditional approaches to transcript discovery. After a thorough benchmarking is performed, the work we have done will make it possible to perform personalized transcriptome discovery, which may have significant medical and health applications.

3. `fugi` is the first tool analyzing both gene fusions and immune cell receptor sequencing with pseudoalignment. We have demonstrated its ability to detect and align gene fusions and T-cell receptor sequences. While more benchmarking is needed to fully quantify it's performance across the spectrum of challenges in these areas, our current results show the feasibility and advantages of the approach. This will allow for a thorough benchmark of various applications of this tool to gene fusion detection and immune cell receptor detection with simulations and real data. We foresee that a careful analysis of the biases that may be present in TCR-seq and other methods that select for T-cell receptor sequences prior to sequencing, will provide further improvements in accuracy beyond those we have already achieved.

References

Clavell-Revelles, Pau et al. (2025). "Long-read transcriptomics of a diverse human cohort reveals widespread ancestry bias in gene annotation". In: *bioRxiv*. DOI: 10.1101/2025.03.14.643250. eprint: https://www.biorxiv.org/content/early/2025/03/17/2025.03.14.643250.full.pdf. URL: https://www.biorxiv.org/content/early/2025/03/17/2025.03.14.643250.

INDEX

[1]"Human Genome Project Completion: Frequently Asked Questions". National Human Genome Research Institute (NHGRI).

[2]"CHM13 T2T v1.1 – Genome – Assembly – NCBI". www.ncbi.nlm.nih.gov.

[3]"T2T-CHM13v2.0 – Genome – Assembly – NCBI". www.ncbi.nlm.nih.gov.