



Master's thesis

Master's Programme in Life Science Informatics

Fusion transcript detection with long RNA reads

Taria Solasuo

April 9, 2026

Supervisor: Dr. Andrey Przhibelskiy

Examiners: Dr. Andrey Przhibelskiy

Prof. Alexandru Tomescu

FACULTY OF SCIENCE
UNIVERSITY OF HELSINKI

Contact information

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki, Finland

Email address: info@cs.helsinki.fi

URL: <http://www.cs.helsinki.fi/>

Tiedekunta — Fakultet — Faculty		Koulutusohjelma — Utbildningsprogram — Study programme	
Faculty of Science		Master's Programme in Life Science Informatics	
Tekijä — Författare — Author			
Taria Solasuo			
Työn nimi — Arbetets titel — Title			
Fusion transcript detection with long RNA reads			
Ohjaajat —Handledare — Supervisors			
Dr. Andrey Przhibelskiy			
Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages	
Master's thesis	April 9, 2026	39 pages	
Tiivistelmä — Referat — Abstract			
Gene fusions are important cancer drivers, biomarkers and drug targets. Reliable detection of fusion events from patient samples is challenging with traditional short-read sequencing because transcripts must be assembled from short fragments. Long-read RNA sequencing produces reads than can span entire fusion transcript, providing more sequence context around the fusion breakpoint. This enables more reliable identification of fusion partners and evaluation of the biological significance of fusion events. Several mechanisms can lead to a fusion event at the DNA level but these events are not necessarily transcribed. In addition, genes can be fused during transcription to produce a fusion transcript without underlying genetic rearrangement. For these reasons, RNA sequencing is commonly used for fusion detection. Several fusion callers have been developed for long-read RNA data, but there are no published bioinformatics tools that can detect both gene isoforms and fusions. During this thesis project I developed a fusion detection algorithm as an extension of IsoQuant, a gene isoform detection tool. Most queries in the pipeline are implemented using interval trees, which allows for fast lookups of the genomic context surrounding candidate breakpoints. I benchmarked IsoQuant-fusion against three existing long-read RNA fusion callers using real tumor datasets and synthetic long-read data simulated with Badread. The algorithm performed clearly better than JAFFAL and FusionSeeker, and comparably to LongGF. It became evident that many validation mechanisms employed by short-read fusion callers are redundant when applied to long-read data. Fusion detection itself is relatively straightforward; however, results from real datasets often contain numerous artifacts and biologically meaningless fusions. Incorporating gene biotype information helps filtering these events. ACM Computing Classification System (CCS) Applied computing → Life and medical sciences → Computational biology → Computational transcriptomics			
Avainsanat — Nyckelord — Keywords			
Gene fusions, long-read RNA sequencing, interval tree			
Säilytyspaikka — Förvaringsställe — Where deposited			
Helsinki University Library			
Muita tietoja — övriga uppgifter — Additional information			
Bioinformatics and Systems Medicine study track			

Contents

1	Introduction	1
1.1	Background	1
1.2	Types of fusion events	1
1.3	Mechanisms of action of fusion genes	5
1.4	Detection of fusion events	6
1.5	Parallels of isoform detection and fusion detection	10
2	Methods	12
2.1	Input file processing	12
2.2	Achitectural constraints	13
2.3	Genomic indexing and context queries	14
2.4	Read processing and candidate fusion detection	14
2.5	Breakpoint clustering and metadata construction	17
2.6	Validation, filtering, and fusion reporting	18
3	Results	24
3.1	Simulated data	25
3.2	Real data	27
4	Discussion	30
4.1	Alternative approaches to long-read fusion detection	30
4.2	Advantages of IsoQuant-fusion	31
4.3	Further development	32
5	Conclusions	34
	Bibliography	35

1 Introduction

Gene fusion, the joining of genetic material from two or more genes, is an important mechanism for generating novel genes and contributing to genomic diversity. Signs of such events were observed as early as the 1960s, but their reliable detection has only become achievable with modern sequencing technologies.

1.1 Background

The existence of a fusion gene was first proved in 1973, when cells from nine patients with chronic myelogenous leukaemia (CML) were analyzed with quinacrine fluorescence and Giemsa staining techniques (Rowley, 1973). The long arm of the chromosome 22 was missing a part that was fused in the long arm of chromosome 9. This translocation later came to be known as BCR-ABL1 fusion gene (Shtivelman et al., 1985) and it is recognized to be the key factor causing CML. Other gene fusions have been found to play critical roles in tumorigenesis in several types of cancer. Gene fusion detection is used to confirm cancer diagnosis (Annala et al., 2013). Conversely, fusions are often the target of cancer therapy and the detections of a certain fusion in a patient's sample is sometimes essential for planning the patient's treatment. Cancer medication specifically targeting fusion genes include imatinib for BCR-ABL1, crizotinib for ALK fusions and larotrectinib designed for NTRK fusions (Jin et al., 2022). Sequencing reads comprising of two or more genes are sometimes referred to as chimeric, derived from the Greek mythological character *Chimera*, which was a mixture of three animal species (Rogers, 2026).

1.2 Types of fusion events

Although the term fusion gene may seem more concrete than fusion transcript, many biologically relevant fusions do not occur at the DNA level. A classification system of fusions by (Calabrese et al., 2020) separates them based on the dependence of genomic rearrangement.

Fusions involving genetic rearrangement can form through multiple mechanisms (Annala et

al., 2013). The four basic types of these are deletions, tandem duplications, inversions and translocations. A loss of genomic region between two genes in the same strand can result in the genes to be fused together. In tandem duplication a genomic region is duplicated one or more time and this can occasionally result in a fusion gene at the breakpoint of the original and copied region. Sometimes a fusion involves two genes two genes at the opposite strands of a chromosome, as a result of an inversion of chromosomal segments. In addition to changes occurring within a chromosome, many known fusions involve genes located in different chromosomes. Formation of such a fusion requires a translocation of a part of a chromosome into another one or a reciprocal swapping of genetic code between two chromosome. Table 1.1 lists a few well known oncogenic fusions.

Table 1.1: Fusions resulting from chromosomal rearrangements.

Type	Example	Disease
Deletion	TMPRSS2–ERG	Prostate cancer
Deletion	IGH–MYC	Burkitt lymphoma
Tandem duplication	FGFR3–TACC3	Glioblastoma
Tandem duplication	BRAF–KIAA1549	Pilocytic astrocytoma
Inversion	CBFB–MYH11	Acute myeloid leukemia
Inversion	EML4–ALK	Non–small cell lung cancer
Translocation	ETV6–NTRK3	Secretory breast carcinoma
Translocation	BCR–ABL1	Chronic myeloid leukemia
Translocation	EWSR1–FLI1	Ewing sarcoma
Translocation	PML–RARA	Acute promyelocytic leukemia
Translocation	RUNX1–RUNX1T1	Acute myeloid leukemia
Translocation	NTRK1–TPM3	Papillary thyroid carcinoma

Gene fusion events at DNA level can be further classified into direct and composite (Calabrese et al., 2020). In a direct case, a fusion is formed as a result of a single event, while a composite fusion forms after multiple structural rearrangements. A subclass of composite fusions is bridged fusions, a where third genomic location bridges two genes, though the bridge section is generally not included the transcribed RNA. In 2-hop fusions the final fusion product is formed through two sequential fusion events, and the final transcript contains exons from three genes. 2-hop fusions can, in addition to DNA-level rearrangements, contain transcriptional or splicing-level events.

The genomic position where a gene is fused into another is called breakpoint. The breakpoints often occur in non-coding regions, not interrupting exons (Annala et al., 2013). The likely reason for intronic breakpoints is simply the small percentage of the exonic areas in the human genome. In DNA-level fusions, genes are joined directly at the breakpoint, unlike transcription-mediated fusions where the fusion occurs only at the RNA level without genomic rearrangement.

Fusions formed as genomic rearrangements can bring together genes that are located far apart in the genome, while fusions occurring during transcription often involve genes in close proximity to one another. Figure 1.1 illustrates the diversity of fusion transcript formation.

The Pan-Cancer Analysis of Whole Genomes Consortium, which analyzed RNA samples from tumors of 1188 individuals, revealed that 18 % of fusion transcripts displayed no evidence of genomic rearrangement (Dorney et al., 2023). Fusions at RNA level can occur through several mechanisms: trans-splicing, transcriptional readthrough and cis-splicing between adjacent genes (cis-SAGe). During transcription and splicing, exons from different genes may be joined. Transcriptional readthrough happens when RNA polymerase continues transcribing beyond the normal stop site of one gene into the next, though some argue these are not genuine fusion RNAs. The biological function of fusions occurring during transcription is not always clear.

Fusions with breakpoints in intronic or intergenic areas can produce functional transcripts but more often they alter gene expression. An analysis of 268 whole-genome sequencing samples consisting of 40 breast invasive carcinomas, 61 colon and rectum adenocarcinomas, 47 thyroid carcinomas and 120 prostate adenocarcinomas revealed that fusions including intergenic and intragenic regions are common in cancer tissues (Yun et al., 2020). Intronic breakpoints can lead to regulatory changes, such as enhancer hijacking or promoter swapping, which can upregulate oncogenes.

Fusion events are directional. The 5' partner contributes the promoter and N-terminal portion of a protein coding fusion and the 3' partner contributes the C-terminal protein domain. In a canonical fusion, both the 5' partner and the 3' contribute its sense (coding) strand, creating a transcript that follows the normal coding direction. Sense-antisense (SAS) fusions, also known as cross-strand chimeras or intragenic trans-splicing, one partner in the sense orientation and the other in the antisense orientation (Dorney et al., 2025). This makes them non-canonical and often harder to detect and interpret. SAS fusions are typically tissue-specific, although many have been found recurrently across different

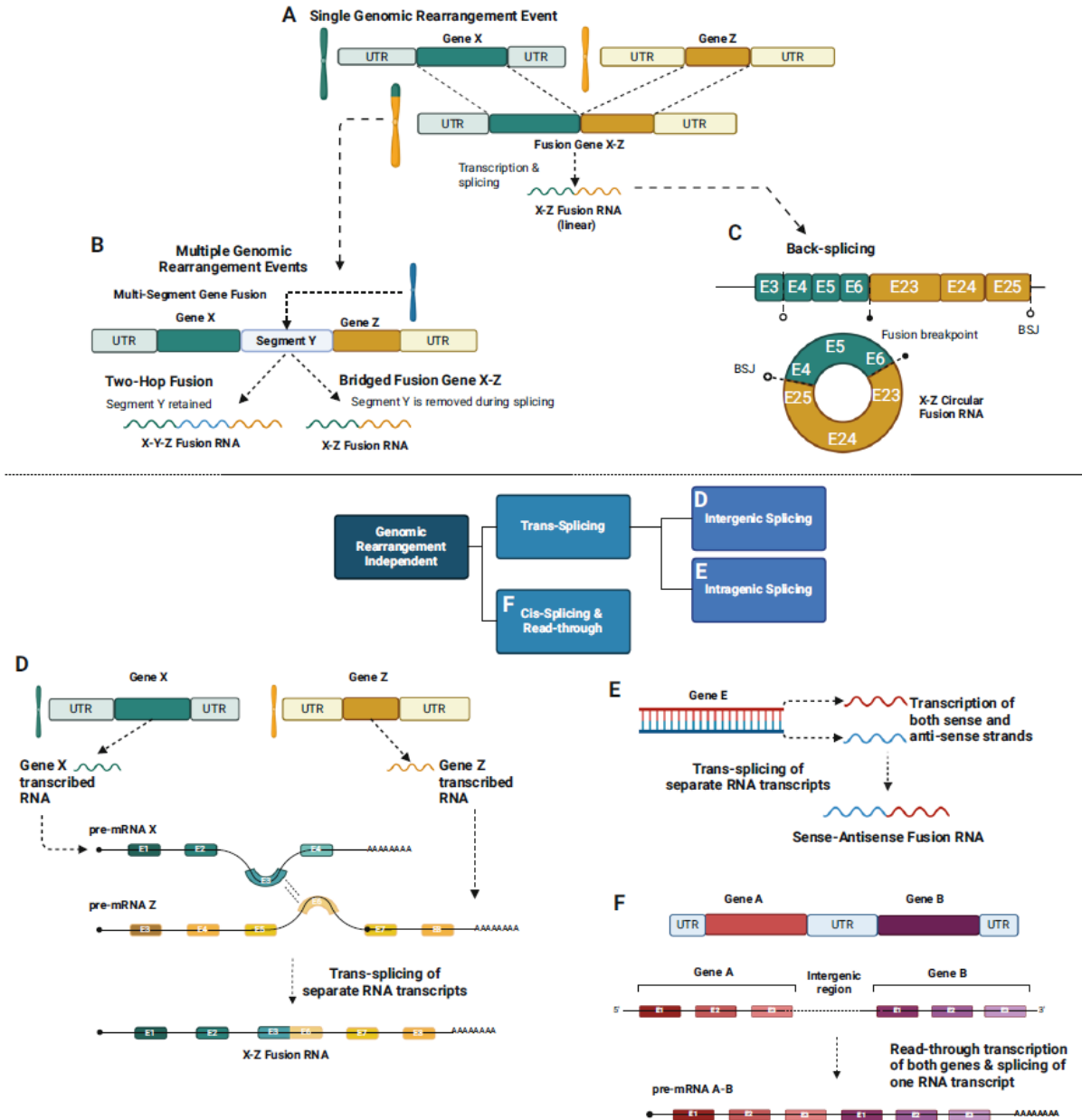


Figure 1.1: Classification of fusion transcripts (Dorney et al., 2025).

tissue types. A study of 9000 tumors found that sense-antisense (SAS) fusions that they were actually more prevalent than canonical fusions, but most of them were novel and previously unreported (Vellichirammal et al., 2020).

Fusion circular RNAs (f-circRNAs) are a newly discovered class of fusion transcripts (Dorney et al., 2025). Unlike canonical circular RNAs, which arise from back-splicing within a single gene, f-circRNAs originate from exons of two distinct genes brought together by chromosomal translocations or other structural variations. These molecules are covalently

closed loops, making them highly stable compared to linear RNAs, and they often exhibit tissue-specific or disease-specific expression patterns. Recent studies suggest that f-circRNAs can play functional roles in tumorigenesis by acting as microRNA sponges, interacting with RNA-binding proteins, or modulating transcriptional networks. Their stability and potential regulatory functions make them promising biomarkers and therapeutic targets. However, detecting f-circRNAs remains challenging due to their complex structure. Identifying fusion transcripts acting as gene fusion initiator is challenging. One approach is to use any fusion caller to detect putative gene fusions and try if they align with any ncRNA in the reference.

1.3 Mechanisms of action of fusion genes

Cell function can be affected by fusion genes in multiple ways. Activation of an oncogene can happen through promoter exchange, where the 3' gene participating in the fusion is overexpressed when a chromosomal rearrangement brings the 3' gene's expression under the control of the 5' gene's promoter (Annala et al., 2013). For example, in IGH-MYC fusion found in Burkitt lymphoma, MYC is placed under the control of the highly active immunoglobulin heavy-chain promoter, leading to strong oncogenic overexpression.

Alternatively, fusion genes may produce proteins with altered structure. In CML, the BCR-ABL1 fusion leads to a constantly active tyrosine kinase that drives uncontrolled cell division.

There is even evidence that a fusion RNA genesis can precede the fusion gene and guide the genomic rearrangement (Gupta et al., 2018). In an experimental study, researchers expressed a short fusion RNA consisting of two exons, TMPRSS2 exon-1 joined to ERG exon-4, mimicking the TMPRSS2-ERG fusion common in prostate cancer, leading to the formation of the corresponding fusion gene in DNA. This worked only with the antisense RNA because it could bind the non-template DNA strand not occupied by RNA polymerase.

Despite the connection between fusion genes and cancer, some fusion genes are also found in germline cells of healthy humans (Annala et al., 2013). An example is TFG-GPR128 fusion, formed as a result of tandem duplication and found in about 2 % of humans. Moreover, fusion transcripts resulting from transcription readthrough are frequently found in various tissues (Caldas et al., 2024). A study that analysed 2778 samples from 23 healthy tissue types found that 34 % of the expressed genes produced readthrough transcripts. The

authors noted that these transcript contained several microRNA binding sites and could thus act as miRNA sponges, affecting gene regulation. Another study, analyzing 236 biologically meaningful, recurrent oncogenic fusions found that the fusion KANSL-ARL17 was found in almost one third of individuals and was slightly more prevalent in normal tissues than in tumor tissues.

1.4 Detection of fusion events

The first known fusion genes were discovered through wet laboratory experiments. When high throughput sequencing became available, several tools were developed for detection of fusion events using DNA or RNA sequences. Most such algorithms designed for short RNA reads are based on the idea of using paired end reads to identify cDNA fragments that combine parts from two genes (Annala et al., 2013). Tools utilizing DNA reads also exist, but fusion detection through DNA sequencing has several shortcomings. It does not reveal to which extent the fused genes are actually been transcribed, and the fusions resulting from splicing and transcription readthrough go unnoticed. Cancer genomes are often rearranged and contain randomly fused genes, which are not transcribed due to an inactive 5' promoter or loss of function.

Traditional short-read RNA-sequencing has several limitations that make it poorly suited for discovering fusion transcripts, especially novel, complex, or unconventional ones (Dorney et al., 2023). Short-read RNA-seq requires RNA fragmentation during library preparation. It breaks apart full-length fusion RNAs and makes it unlikely that full-length SAS fusions or other long fusion events survive intact, reducing the chance of detecting them at all. The development of long-read sequencing has enabled more straightforward fusion detection (Zhao et al., 2025). Currently there are two main technologies for producing long reads, Pacific Biosciences (PacBio) and Oxford Nanopore Technology (ONT). Long reads can span the full length of a fusion transcript, bypassing the assembly step and increasing the accuracy of fusion analysis.

A handful of tools for fusion detection using long RNA reads have been published during 2020's: LongGF (Q. Liu et al., 2020), AERON (Rautiainen et al., 2020), JAFFAL (Davidson et al., 2022), FusionSeeker (Chen et al., 2023) and FUGAREC (Masuda et al., 2024). In addition, GFVoter (Zhao et al., 2025) introduces a voting system between two different sequence aligners and fusion detection results of JAFFAL and LongGF. GFHunter (Y. Liu et al., 2025) – not yet formally published – is among the newest additions. These tools

share the approach of finding split reads – transcripts that map to two or more locations in the reference genome – but the filtering and validation techniques vary. For this study I got acquainted with the principles of three of these fusion detection tools.

LongGF LongGF, a fusion detection tool implemented in C++ was among the first designed for long RNA reads. It does not have a built-in alignment method but assumes that the reads are aligned to a reference genome with a long-read aligner. The input is a BAM file, sorted by read name, and a Genome annotation file (GTF) containing the definition of known genes and their transcriptional isoforms (Q. Liu et al., 2020). The fusion detection process starts with finding reads that map to multiple positions. Candidate gene pairs are obtained by comparing mapped genomic positions against a corresponding gene definition with genomic coordinates of exon for each transcript of genes and finding gene pairs with non-random supporting reads. Alignments are discretized into windows and fusion points are identified when breakpoints from different alignments fall into consistent window pairs. The final output is a list of candidate gene fusions ranked by the number of supporting reads.

LongGF compares mapped positions to pre-defined genomic coordinates of exons of known genes (Dorney et al., 2023). It does not detect fusions involving more than two genes, or breakpoints in intronic regions.

JAFFAL JAFFAL is an extension to the short-read fusion detection tool JAFFA written in C++. Input of JAFFAL is the long reads in FASTQ format – BAM files are not accepted as input. Distinct to other similar tools, JAFFAL pipeline includes two rounds of alignments (Davidson et al., 2022). Fusions are first detected aligning the reads to a reference transcriptome using minimap2 . Reads that have sections aligning to two different genes are selected to further analysis and aligned to the reference genome to remove possible false positives. Fusion breakpoints are determined using the end position of reference genome alignments. The breakpoints are then classified into "High confidence", "Low Confidence" and "Potential Trans-Splicing" classes.

JAFFAL identifies the 5' and 3' partner genes of found fusions and annotates their strands and in-frame or out-of-frame status. By searching for reads with two or more breakpoints, JAFFAL is able to detect multi-segment fusions. For example, it identified fusion pairs BMPR2-TYW5 and TYW5-ALS2CR11 belonging to the same transcript, forming three-gene fusion BMPR2-TYW5-ALS2CR11. Since the initial alignment is done to the reference

transcriptome, fusions involving intronic or intergenic regions are not detected.

Among the tools tested, JAFFAL is the most complex to use. The pipeline expects the user to download a set of around 20 reference files, some of which need to be converted into BLAST databases before running fusion detection. The names of the required files are hard-coded, making the use of custom reference files difficult.

FusionSeeker FusionSeeker, written in Python, is designed to detect fusions involving both exonic and intronic breakpoints (Chen et al., 2023). The input of FusionSeeker is a BAM file and a GTF file, which are scanned for split-read patterns chromosome by chromosome. Candidate fusions from the same pair of genes are clustered based on the breakpoint locations using a density-based spatial clustering of applications with noise algorithm (DBSCAN). Candidate fusions of each cluster are merged into a gene fusion call and filtered based on the number of fusion-supporting reads. Finally, the pipeline reconstructs the transcripts of the found fusions and aligns them to the reference genome with minimap2 to find the exact breakpoint of each gene fusion.

Reconstruction of the fusion transcripts reveals which exons from each partner gene are involved and whether the fusion is in-frame or frameshift. With the full transcript it is also possible to quantify the expression levels. In case of novel fusions, full transcripts are required for fusion database submission.

Specialized methods have been developed for the detection of complex fusion structures. These predominantly rely on short-read sequencing. For cis-SAGE events in particular, long-read data would not provide a meaningful advantage, since the Cis-SAGE junctions are typically only a few hundred bases apart. CIRI-long (Zhang et al., 2021) is an example of long-read fusion caller capable of detecting circular fusion RNAs. The developers highlight that short-read fusion callers fail largely because they cannot distinguish circular reads from linear ones that share the same exons. Circular fusion RNA detectors need to identify back-splice junctions, reconstruct full circular isoforms and utilize specific enrichment and reconstruction methods, such as rolling-circle reverse transcription, to distinguish circular fusion RNA transcripts.

The roles of non-coding RNA (ncRNA) fusions in cancer are largely undocumented and most fusion calling algorithms are designed to discard ncRNA fusion transcripts (Dorney et al., 2025). Addressing this bias, Somatic Fusions involving Non-Coding Sequences (SFyNCS) is a computational method specifically designed to detect gene fusions involving both protein-coding and non-coding sequences (Zhong et al., 2023). The method performs

raw fusion detection without gene annotation so that fusion candidates are considered without the knowledge of their biotype. The validation is done by comparing the fusion transcripts with somatic structural variations detected from genomic data. SFyNCS uses short-read RNA-seq data.

The main issue with non-coding fusion RNA transcripts is that they are plentiful in real tumor samples but presumably only a small fraction is biologically relevant. FusionInspector (Haas et al., 2023) is an attempt to classify fusion transcripts. The pipeline evaluates candidate fusion transcripts by examining expression strength, fusion allelic imbalance, splice-junction fidelity, and the presence or absence of sequence microhomologies between partner genes. Biologically relevant fusions are characterized by high fusion transcript expression, canonical splice patterns, strong allelic imbalance, and low microhomology, whereas artifact-like fusions show low expression, irregular or non-canonical splicing, and elevated sequence microhomology.

Another project focused on the clinical relevancy of fusion transcripts is by Hafstað et al., 2023. They first constructed a large training set of validated fusions by matching RNA-seq fusion calls to supporting structural variants in paired whole-genome sequencing, treating unsupported calls as likely artifacts. Using these validated events, they trained a machine-learning classifier that distinguishes true biological fusions from false positives based on features such as read support patterns, breakpoint consistency, gene expression, and annotation-derived properties. When applied to new RNA-seq samples, the classifier effectively filtered out technical artifacts and enriches for clinically relevant fusions, including kinase fusions of therapeutic interest.

Most fusion detection tools are designed for bulk sequencing data. Single-cell data typically have much lower read depth and higher noise, making fusion detection less reliable (Jin et al., 2022). Fusion callers developed for single-cell RNA sequencing can help identifying which cells carry the fusion and compare expression profiles between fusion-positive and fusion-negative cells. Single-cell fusion detection can also be better at detecting lowly expressed fusions. In theory, long-read fusion callers can also support barcoded single-cell long reads (Davidson et al., 2022), but it was not attempted during this project.

1.5 Parallels of isoform detection and fusion detection

IsoQuant Gene isoforms are distinct RNA transcripts produced from the same gene through alternative transcription start sites, splicing, or polyadenylation (Prjibelski et al., 2023). These isoforms can encode different protein variants or have different regulatory roles. Long-read sequencing has enabled more accurate detection of gene isoforms. Isoquant is a graph-based isoform discovery tools that takes PacBio or ONT reads and a reference genome as input and constructs a directed intron graph where vertices are pairs of splice sites and edges represent sequential relationships between splice junctions. This graph is used to construct paths representing full-length transcripts. If user has included a GTF file with the input, the reads are first mapped to the known isoforms and the assignments are used for reference transcript quantification and error correction. IsoQuant is able to discover novel isoforms by identifying new paths through known and unknown splice junctions.

A comparison between nine isoform detection tools using long RNA reads concluded IsoQuant’s guided mode has the best performance for alternative splicing detection with good precision and sensitivity (Su et al., 2024).

While isoform detection aims to identify alternative splicing events that produce multiple transcript variants from a single gene, fusion detection identifies chimeric transcripts formed by the joining of exons from two different genes. Current bioinformatics tools for both tasks utilize PacBio and ONT long reads and require accurate sequence alignment. Both have significant implications for functional genomics, disease mechanisms, and clinical diagnostics. However, there are currently no tools that can detect both gene isoforms and fusion events from long RNA reads. The developers of the isoform discovery tool FLAIR have introduced also FLAIR-fusion (Felton et al., 2023), but these are two different tools.

IsoQuant users have requested a fusion detection functionality from the developers. It is often useful to test a sample for both gene fusions and for certain isoforms since both can affect protein function. Gene fusions often involve a certain isoform of a gene (Haas et al., 2023), and some fusions have several known isoforms (Davidson et al., 2022).

The goal of my master’s thesis project was to extend IsoQuant with a fusion detection module, enabling more comprehensive transcriptome analysis. I implemented the algorithm

in Python, like the main program. It is primarily designed to detect canonical, protein-coding fusions. Bulk long-read RNA sequence data was used during the development and benchmarking.

2 Methods

Although fusion events are distinct from splicing events, parts of IsoQuant’s existing architecture was utilized in the development of the new functionality. In this chapter I explain how fusion detection is integrated into IsoQuant, and the workflow and logic of the pipeline.

2.1 Input file processing

In the guided mode of IsoQuant, user provides a FASTQ file or a BAM file of long RNA reads, a reference genome in fasta format, and a Gene Transfer format (GTF) file as input (Prjibelski et al., 2023). If a FASTQ is given as input, the IsoQuant pipeline performs the alignment with minimap2 in splice-aware mode and consequently sorts and indexes the resulting BAM file. A reference genome is required even when input reads are already aligned, because IsoQuant accesses genomic sequence information throughout isoform detection and validation. In fusion transcript detection, the reference genome is additionally used for realigning the reconstructed fusion sequences. If any required input files are missing, the logger reports this to the user.

IsoQuant manages the reference FASTA file using the `Fasta` class from the `pyfaidx` library. The resulting `Fasta` object is stored as a dictionary-like structure, enabling efficient, random access to chromosome sequences during all stages of the pipeline. Retrieved chromosome records are used in several components of the workflow—for example, to extract reference sequence context needed for identifying canonical splice sites.

The GTF file is converted into an SQLite feature database using `gffutils` library (Pertea and Pertea, 2020). This database allows for fast lookups of genetic features at specific genomic positions. For subsequent runs with the same genome, user is supposed to provide the database as input to avoid time-consuming conversion.

2.2 Architectural constraints

In IsoQuant, transcript model is a data structure that represents a validated transcript isoform, containing its complete genomic structure and classification information. It serves as the final output of the isoform transcript discovery and construction process. Initially, it seemed appealing to build a similar structure to validate fusion transcripts.

Transcript models reveal the complete exon-intron architecture of each isoform, showing exactly how the pre-mRNA is spliced. The `exon_blocks` field stores the genomic coordinates of each exon and `intron_path` represents the splicing junction pattern used during model construction. By comparing multiple transcript models from the same gene locus, it is possible to identify alternative splice and transcription start/end sites or exon skipping and intron retention events. Each transcript model is associated with supporting read counts stored in `transcript_read_ids`, which maps transcript IDs to the reads that support them. An isoform is linked to their parent gene through the `gene_id` field.

Transcript models are built through a graph-based approach that creates an intron graph from all observed splicing patterns in the input reads, identifies complete paths through the graph representing full-length transcripts, validates each path based on coverage thresholds and quality metrics, classifies transcripts as known or novel and filters redundant or low-quality models.

The core problem with integrating fusion detection into IsoQuant was that the program is designed to work with single chromosomes. All data structures, starting from `ReadAssignment` to `TranscriptModel` use a single `chr_id` field. IsoQuant processes data chromosome-by-chromosome in complete isolation, making it almost impossible to construct transcript models spanning multiple chromosomes. Attempts were made to modify the components compatible with fusion reads, but for them to function would have required major architectural changes with potentially little gain.

Therefore, the fusion detection feature is largely separate from the main IsoQuant pipeline. The GTF and FASTQ processing are the same as in the isoform detection, but if the user sets the `-fusion` flag, the feature database and the FASTQ and BAM files are passed onto the `FusionDetector` module, after which logger will notify about the beginning of fusion detection.

2.3 Genomic indexing and context queries

To support efficient coordinate-to-feature mapping, chromosome-specific interval trees are constructed for genes and exons. These structures are created from a `gffutils` FeatureDB and store only feature identifiers for memory efficiency, while full annotation objects are queried from `gffutils` database when information such as biotype is needed. All database queries are cached. The `GenomicIntervalIndex` provides both point and windowed lookups, enabling fast retrieval of the genomic context surrounding candidate breakpoints. Exon enumeration relies on the interval tree, implemented with the `IntervalTree` library. Interval trees are specifically designed to support efficient identification of all intervals overlapping a given query interval and are widely used in tools for genomic region querying (Schäfer and Yang, 2025). This structure enables lookup times of $O(\log n)$. In figure 2.1 each gene is a subtree and exons are depicted in green.

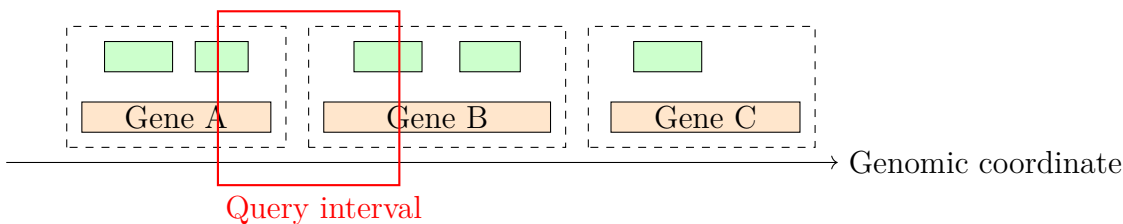


Figure 2.1: Conceptual representation of chromosome-specific interval trees. A query interval overlaps multiple interval tree nodes, genes and exons.

2.4 Read processing and candidate fusion detection

The events potentially representing a fusion are collected from supplementary alignments and soft-clipped areas.

1. Supplementary alignments

When a fusion RNA transcript is aligned to the reference genome, it typically maps to two or more distinct locations, often on different chromosomes. Splice-aware aligners such as `minimap2` report supplementary alignments (SA) in the SAM file when alignments are performed in spliced mode (Li, 2018 Li et al., 2009). Each SAM entry consists of 11 or more tab-separated fields. For fusion detection, the most relevant elements are the supplementary alignment flag, the presence of soft clips, and the read quality.

Reads with supplementary alignments are marked with the 0x800 flag in the second column of the SAM entry. The SA tag appears when a read spans structural variants or fusion breakpoints. If a read is split across multiple regions, all segments are reported. This supplementary alignment information is essential for reconstructing the full alignment of a read.

SA entries are parsed and filtered by per-entry aligned length L_{sa} and mapping quality Q_{sa} (defaults $L_{sa} \geq 50$ bp, $Q_{sa} \geq 10$). To bound computation and reduce noise, at most the top three entries ranked by (Q_{sa}, L_{sa}) are kept.

2. **Large soft-clips:** In sequence alignment, soft-clipped regions are portions of a read that do not align to the reference but remain in the sequence. In contrast, hard-clipped regions are trimmed bases that are completely removed from the sequence and do not appear in the SAM file. Soft clips are indicated in the CIGAR string (Samtools developers, 2024). For fusion transcripts, the junction often appears as a soft-clipped region because it does not align as expected. The part of the read before the breakpoint aligns normally, and the part after the breakpoint has nowhere to align and becomes soft-clipped. When searching for fusion breakpoints, it is important to identify soft clips of sufficient length and correctly oriented relative to the fusion partner. There is only one possible side for each partner, as illustrated in Figure 2.2 (Hafstað et al., 2023). The algorithm detects a clip of length S_c (default ≥ 40 bp) at either read end. The clipped sequence is realigned to the reference genome using `mappy`. A valid hit requires a minimum match length and mapping quality (defaults: match ≥ 50 bp, $Q \geq 20$).

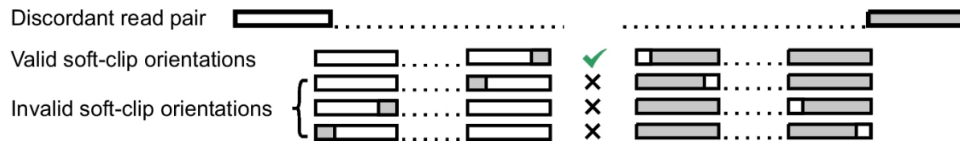


Figure 2.2: Soft-clipped ends that support a fusion breakpoint (Hafstað et al., 2023).

While true fusion transcripts almost always carry an SA tag, not all SA-tagged reads arise from true fusions. Adjacent genes on the same strand can produce transcripts spanning both genes naturally. SA tags can arise from normal alignment artifacts, repetitive sequences, or structural features of the genome, not just true gene fusions. Especially long reads often contain sequencing errors that can mimic fusions (Zhao et al., 2025). Both

PacBio and ONT cDNA use reverse transcription to convert RNA into cDNA before sequencing, and this process, fragments from different transcripts may join artificially. Template switching refers to a case where reverse transcriptase jumps from an RNA molecule to another when copying RNA and produces a single cDNA that does not reflect any real molecule in the sample. Numerous extra chimeric reads have been found in datasets sequenced with cDNA, which are absent when sequenced with ONT direct technology (Davidson et al., 2022).

The default number of supporting reads required to report a fusion varies between existing tools. JAFFAL does not set a strict limit but instead reports reads that have just one supporting read complemented by other evidence as low-confidence (Davidson et al., 2022). In the LongGF CLI, user needs to set the number of supporting reads, the recommendation being 2 (Q. Liu et al., 2020). Other criteria for fusion validation include the number of fusion-supporting reads, mapping quality, breakpoint proximity to exon boundaries, length of the aligned parts and successful reconstruction of transcripts and realignment. Strictening of these parameters and increasing the number of required supporting reads are heuristic methods to limit the amount of false positives which, inevitably, also leads to missing true positives. Running fusion detection with a small dataset including known fusions demonstrates how tweaking one criteria might reveal one more real fusion but drop another one out.

Whenever a potential fusion breakpoint is identified, the workflow invokes `record_fusion`. This function first performs an initial gene assignment for both sides of the breakpoint. If this context indicates that the event would involve an intergenic region, the same gene on both sides (intragenic), or the mitochondrial genome, the fusion is not recorded, as such events are considered non-informative in downstream analysis. Fusions involving partners representing other than protein-coding genes and immune receptor genes (IG and TR) are discarded before further processing. Because the raw assignment uses unclustered per-read breakpoints, it may temporarily map to non-coding elements even when the true fusion partner is the adjacent coding gene. In practice, however, trying to map non-coding fusion partners to nearby protein coding variants produced more false positives than it salvaged true positives.

2.5 Breakpoint clustering and metadata construction

Once all reads have been processed, the pipeline advances to the metadata-construction phase. Before any clustering is performed, an early filtering step removes candidate fusions involving non-protein-coding or non-whitelisted gene biotypes at the raw-assignment stage. These early eliminations prevent downstream computation on biologically implausible events. For each remaining candidate fusion key, the algorithm aggregates all breakpoint observations and applies a two-dimensional sliding-window clustering procedure. The breakpoint clustering procedure identifies the most strongly supported fusion junction by applying a two-dimensional sliding-window search over the observed breakpoint coordinates.

First, all breakpoint observations are grouped by chromosome pair and sorted by the first breakpoint coordinate p_1 . A one-dimensional sliding window of width w is moved along p_1 , retaining at each step only those observations whose p_1 values fall within the window. For every such p_1 -window, the enclosed points are then locally sorted by the second breakpoint coordinate p_2 . A second sliding window of the same width w is applied to p_2 , yielding a set of points whose breakpoints fall within an axis-aligned $w \times w$ region. The total weight of the points inside this region is computed, and the region with the highest cumulative weight is selected. The final breakpoint estimate is obtained as a weighted mean of the coordinates (p_1, p_2) within the best-scoring window, as shown in figure 2.3. This process is repeated independently for each chromosome pair, and the pair yielding the largest total support is chosen as the final consensus breakpoint.

Metadata for each fusion key is then initialized. This includes the set of supporting reads, gene assignments, per-gene assignment scores, and the consensus breakpoint. With these values defined, the metadata layer retrieves gene biotypes for each partner and prepares the fusion record for downstream validation and classification.

Gene identifiers are normalized resolving database IDs such as ENS- and RP11- to canonical gene symbols and removing suffixes (-AS, -DT that denote antisense or divergent transcript variants. Divergent transcription produces numerous short, unstable antisense RNAs that are not reliably annotated and rarely represent independent genes, making their suffixes unsuitable for representing biologically meaningful fusion partners. SAS fusions, on the other hand, are not considered in IsoQuant-fusion’s logic. Normalizing identifiers to gene symbols avoids inflating the number of distinct ”genes”, stabilizes fusion clustering, and ensures fusion events are represented at the biologically relevant gene

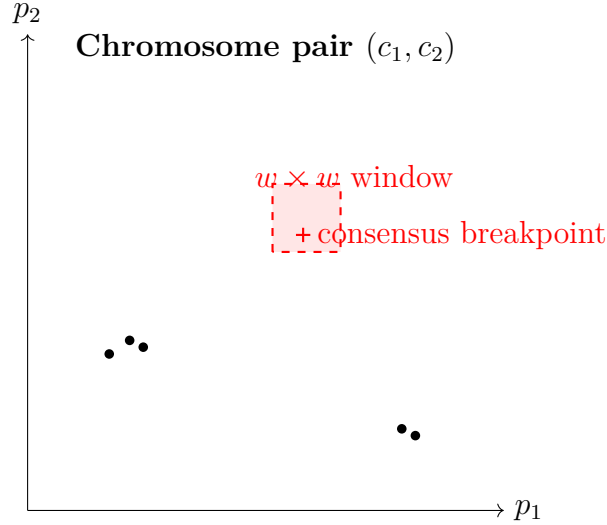


Figure 2.3: Consensus breakpoint determination in the (p_1, p_2) plane. Observed breakpoint pairs are shown as points. A $w \times w$ sliding window identifies the region with maximal cumulative support, and the final breakpoint is computed as the weighted mean of points within this region.

level, which aligns with how fusion transcripts are described in large pan-cancer studies.

2.6 Validation, filtering, and fusion reporting

After metadata construction, the pipeline proceeds through a tiered validation framework implemented by the **FusionValidator**. First, remaining non-coding partners are filtered out using biotype rules applied to the final gene assignments. Next, the system performs frequency-based filtering that removes recurrent multicopy artifacts.

Fusion detection on real data often reveals "promiscuous" genes, that appear as fusion partners in multiple fusions, most of which are not biologically relevant. Distinguishing artifacts from biologically relevant fusions is not straightforward. For example, the KMT2A gene is heavily rearranged in AML, acute lymphoblastic leukemia (ALL) and mixed-lineage leukemia (MLL) and has been documented to fuse with 135 different fusion partners (Engvall et al., 2020). None of the long-read fusion callers describe any explicit filtering rules targeting biologically irrelevant gene families such as ribosomal proteins, immunoglobulin genes, or HLA. Short-read fusion caller Arriba uses position-specific blacklist, trained with real sample data, to remove recurrent artifacts (Uhrig et al., 2021). According to Haas et al., 2023, prefixes of fusion partners, that are likely not biologically relevant, include:

- Ribosomal proteins: RPL (large subunit), RPS (small subunit)
- Mitochondrial genes: MT-
- Immunoglobulin genes: IGH, IGK, IGL
- T-cell receptor genes: TRA, TRB, TRD, TRG
- HLA genes: HLA-, highly polymorphic
- Small nuclear/nucleolar RNAs: SNORD, SNORA, SCARNA, RNU, U (e.g., U1, U2)
- Y RNAs: RNY
- Pseudogene indicators: suffix "P", LOC
- Long non-coding RNA prefixes: MIR, LINC, RP11-, AC, AL, AP
- Histone gene clusters: HIST1, HIST2, HIST3
- Olfactory receptors: OR
- Structural protein families prone to artifacts: KRT

FusionValidator removes fusion candidates where a partner gene is partnering with more than four different genes. In case of ribosomal and histone genes, the threshold is more than two different partners. Fusions where both partner genes represent multicopy families or ribosomal or histone genes are excluded.

For validated candidates, exons within ± 100 bp of each consensus breakpoint are extracted from the reference genome and concatenated to form a putative fusion transcript. This sequence is realigned to assess alignment quality, with thresholds applied to match length and MAPQ. Reconstruction and realignment success earns the fusion candidate a realignment bonus which contributes towards confidence scoring of fusion candidates.

The goal of this scoring system is to summarize, in a single value between 0 and 1, the overall plausibility that a reported fusion represents a genuine biological event rather than an artefact of sequencing, alignment, or genomic annotation. The scoring normalizes the number of supporting reads and up to ten reads contribute linearly to the base confidence, reflecting diminishing returns beyond modest coverage. Breakpoints located close to annotated exon boundaries receive a positive bonus, reflecting that most biologically meaningful fusions preserve exon structure. A set of biological priors then penalizes fusion

events involving multicopy gene families and conversely awards a bonus if either gene is a known recurrent driver (e.g., BCR, ABL1, PML, ETV6). All evidence components are summed and scaled between 0 and 1, yielding a calibrated fusion confidence measure.

The final report is generated in a tab-separated format, including gene names, chromosome number, consensus breakpoints, number of supporting reads, classification (canonical or cis-SAGe), confidence score and fusion key. Candidates earning a confidence score lower than 0.3 are excluded from the report. The confidence score is included in the final report to guide further analysis.

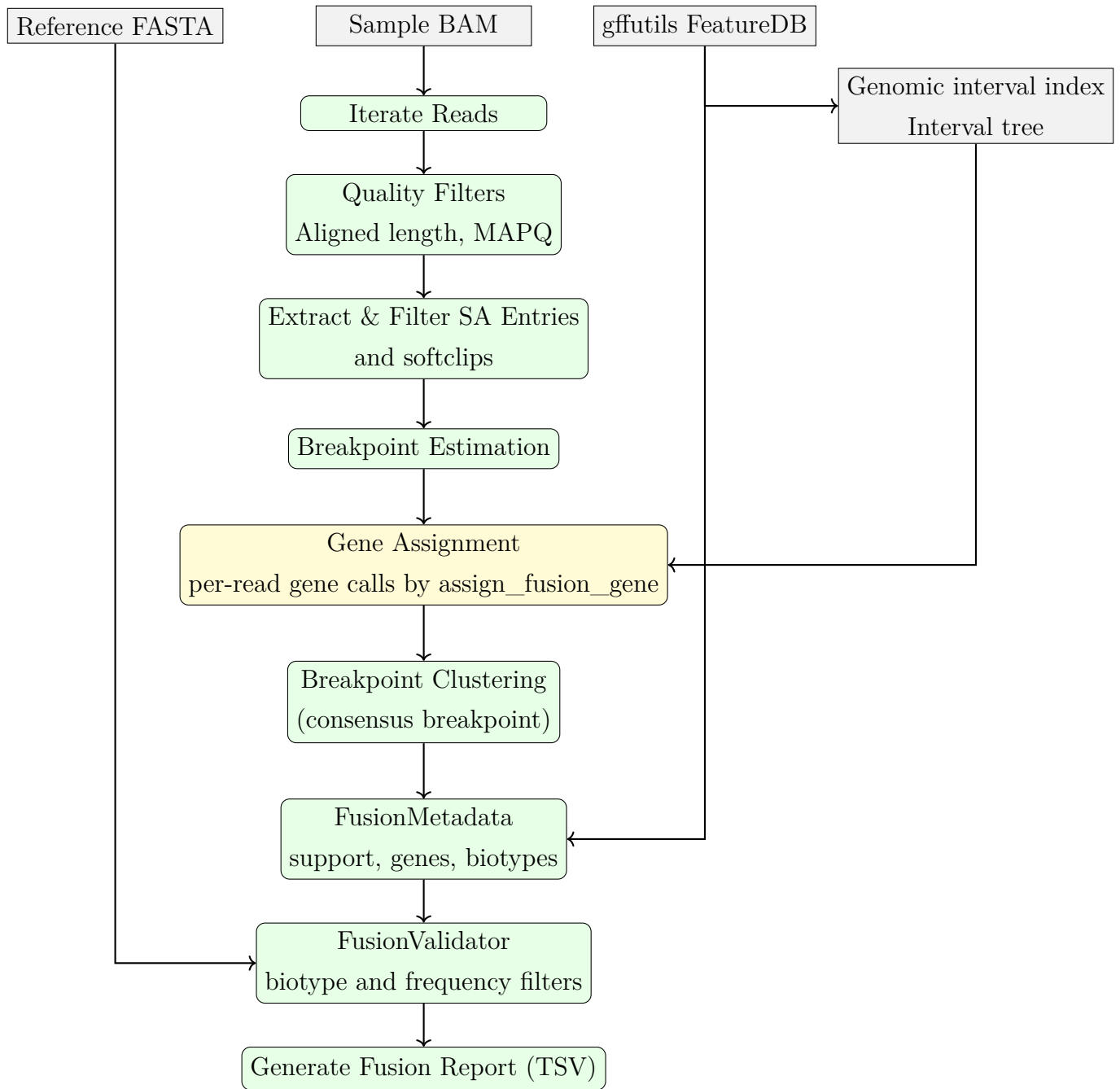
The pipeline is illustrated in Figure 2.4.

During the development process I tried out several additional steps in scoring the fusion genes and filtering fusion candidates, such as:

- In case of non-protein coding fusion partner genes, I applied a function to look for a nearby protein coding gene which might be the correct fusion partner, and especially tried to look for parent genes of pseudogenes
- Verifying that the candidate is represented in correct 5' to 3' orientation
- Ensuring correct orientation of soft-clipped parts
- A two-stage fusion gene assignment. First, `assign_fusion_gene` was called for individual chimeric reads. After breakpoint clustering and determining of a consensus breakpoint, fusion gene was re-assigned based on the consensus breakpoint, in case the assignment was not confident enough
- An extensive scoring system, which calculated whether the candidate fusion event was in-frame or out-of-frame and how was the breakpoint located in relation to exon boundaries

None of these attempts had any meaningful effect on the fusion detection results. Eventually, I begun deleting these additional steps from the pipeline one by one, and the metrics improved every time. This process highlights the nature of long reads and their advantages in fusion detection. The whole fusion transcript being presented in a single read eliminates the need for validation techniques used by short-read fusion callers.

Computational and space complexity The construction of the genomic interval index constitutes the most computationally expensive initialization step of the pipeline.

**Figure 2.4:** IsoQuant-fusion pipeline.

During this phase, all gene and exon features from the annotation database are inserted into chromosome-specific interval trees. Let V and E denote the number of gene and exon features, respectively. Each insertion into an interval tree incurs a $O(\log N)$ cost, where N is the number of intervals already stored for the corresponding chromosome, resulting in overall construction complexity of $O(V \log V + E \log E)$. In practice, since $E > G$, exon indexing dominates runtime. The memory footprint of the index scales linearly with the number of stored features, $O(V + E)$, as each interval stores only basic coordinate data and a feature identifier. Although costly to build, this interval index significantly reduces the time complexity of subsequent gene and exon queries, enabling efficient logarithmic-time lookups during fusion detection.

As for the rest of the pipeline, let

- G denote the number of gene features in the annotation database,
- E denote the number of exon features in the annotation database,
- C denote the number of chromosomes represented in the annotation,
- N_{chrom} denote the number of intervals stored for a given chromosome,
- G_{chrom} denote the number of gene features on a given chromosome,
- E_{chrom} denote the number of exon features on a given chromosome.

The per-read processing performed in `detect_fusions()` is the most time-consuming. Each of the N reads is evaluated together with an average of S supplementary alignments. After filtering, each SA entry triggers breakpoint estimation and a gene-assignment query on an interval tree, yielding an overall cost of $O(N \cdot S \cdot \log G)$, where G denotes the number of genes in the annotation database. Following read processing, all breakpoint observations associated with each fusion candidate undergo a two-dimensional sliding-window clustering procedure whose cost is $O(K \log K)$ per fusion, where K is the number of breakpoints for that specific candidate; over all candidates this amounts to $O(B \log K_{\text{max}})$, with B being the total number of collected breakpoints. Metadata construction and subsequent validation steps, including biotype filtering, frequency-based artifact removal, classification, and confidence scoring, operate once per fusion candidate and therefore run in $O(F)$ time. Combining these contributions gives a total time complexity of $O(N \cdot S \cdot \log G) + O(B \log K_{\text{max}}) + O(F)$, typically dominated by the read-level

gene-assignment stage. In practice, analyzing a 5 GB BAM file with Gencode v49 annotation takes roughly 10 minutes with a T490 Linux device.

The peak memory usage arises from storing breakpoint observations, raw per-read assignments, the interval-tree gene index, and fusion-level metadata. These structures require $O(B)$, $O(N_f)$, and $O(G)$ space respectively, where N_f is the number of reads participating in any fusion event. After metadata construction, most per-read structures are cleared, leaving a final memory footprint of $O(G) + O(B) + O(F)$, which is generally modest compared to the input size and stable across datasets.

Table 2.1: Space complexity of major data structures used in the fusion detection pipeline.

Data structure	Space complexity	Description
Gene interval trees	$O(G)$	Gene-level genomic intervals
Exon interval trees	$O(E)$	Exon-level genomic intervals (dominant)
Fusion breakpoint store	$O(B)$	Raw breakpoint observations
Per-read fusion assignments	$O(N_f)$	Temporary per-read gene assignments
Fusion metadata	$O(F)$	Final fusion-level records

3 Results

Benchmark results of long-read fusion callers in published papers are inconclusive. F1 scores of FusionSeeker, Jaffal and LongGF were 0.93, 0.66 and 0.89, respectively, when benchmarked with simulated data in Chen et al., 2023. Real data runs were excluded from the overall metrics. Meanwhile, according to GFVoter developers, their tool achieved an average F1 score of 0.569, and JAFFAL, LongGF and Fusionseeker scored 0.386, 0.407 and 0.291, respectively (Zhao et al., 2025). Included in the metrics was both real and simulated data and the results varied significantly by dataset. Genion developers found that their tool performed better than LongGF (Karaoglanoglu et al., 2022), but the result hasn't been replicated in other studies. Differences in the metrics may arise from development practices in which the software is optimized specifically for the data used during testing.

IsoQuant's fusion detector was tested with both real and simulated PacBio and ONT data. The reference files were downloaded from <https://www.encodegenes.org/human/>. The GTF file used with real data and my own simulation was

gencode.v49.chr_patch_hapl_scaff.annotation.gtf and the reference genome was GRCh38.primary.assembly.genome.fa. This version of gene annotation includes alternate haplotypes, patches, and scaffolds where many fusion events occur. Simpler versions of annotation files didn't provide decent results. During the testing it also became evident that the GTF version needs to be compatible with the test data. Older simulated data, such as the datasets used in validating JAFFAL, worked well with Gencode version 22 GTF, but the accuracy dropped to around 50 % when fusion detection was run with the newer version 49 GTF. Gencode updates continuously: gene models move exon boundaries, transcripts are added and removed, biotypes change, and gene names get revised. The gencode.v22.chr_patch_hapl_scaff.annotation.gtf file is 1.3GB in size, whereas version 49 of the same annotation is 3.5GB. This substantial increase reflects the extent of changes and additions introduced between the two releases.

LongGF was run using the recommended settings 100 50 200 0 0 2, which define the minimum overlap length with a gene, discretized bin size, minimum length of an alignment record against the reference genome, pseudogene inclusion, secondary alignment inclusion and number of supported reads, respectively. The other tested tools don't have input parameters.

I excluded JAFFAL’s ‘potential’ fusions—those supported by only a single read—to ensure comparability with the output of the other tools. Unfortunately, I was unable to run JAFFAL with the Gencode v49 annotation, as the most recent publicly available release supports only v22. Consequently, the JAFFAL results for both the real and the simulated datasets may not fully reflect the tool’s optimal performance with updated annotations. The gene names are a bit messy in the older annotation, which can also be seen in table 3.2.

3.1 Simulated data

Bioinformatics tools are typically tested with simulated data mimicking the characteristics and errors of real sequencing techniques. Knowing the ground truth allows for calculating exact metrics for each tool run with the same dataset.

For benchmarking, I first generated a synthetic ONT dataset using Badread (Wick, 2019). Badread is a long-read simulator which can, in addition to modeling read lengths and sequencing errors, include glitches, random reads, adapters and chimeras. I combined the input reference using transcripts from Gencode v49 reference transcriptome and 493 fusion transcripts of known, validated fusions. These fusion pairs (and a few triplets) were randomly selected from the `known_fusions.txt` within JAFFA reference files. Reads were simulated to a total depth of $10\times$ across the reference.

Badread was configured to introduce ONT-like sequencing artefacts through the `--glitches` parameter (10,000 substitutions, 10 insertions, and 10 deletions per million bases), as well as additional sources of noise commonly observed in real datasets. Specifically, 0.1 % of reads were designated as “junk” (low-quality or truncated reads), 0.1 % as random sequences, and 0.1 % as chimeric reads. These settings were chosen to approximate the heterogeneity and error characteristics of real ONT RNA-seq experiments.

The dataset was generated using the following command:

```
badread simulate \  
  --reference combined_transcripts.fa \  
  --quantity 10x \  
  --glitches 10000,10,10 \  
  --junk_reads 0.1 \  
  --random_reads 0.1 \  

```

--chimeras 0.1

To simulate fusion transcripts in PacBio data, I simulated another dataset with Badread using PacBio2023 model. This dataset also included 493 randomly selected real fusions, and was otherwise simulated with the same settings as above but included no other than fusion transcripts.

For further comparison, the tools were run with Pac_fus_sim95 and Ont_fus_sim95 which was downloaded from Davidson and Oshlack, 2021. They used 2,500 predefined fusion transcripts from Haas et al., 2019 and simulated reads at varying coverage with Badread. Junk reads, random reads and chimeras were set at 0. Metrics were better when run with an older annotation file (v22), since the data was simulated a few years ago when the more comprehensive annotation was not yet available.

Metrics for all four simulated datasets are presented in 3.1

Table 3.1: Comparison of fusion detection performance for four tools across three simulated datasets. Best values per dataset are **bolded**.

Tool	Ont49			Pac49		
	Sensitivity	Precision	F1	Sensitivity	Precision	F1
IsoQuant	0.923	0.934	0.929	0.852	0.955	0.900
LongGF	0.925	0.996	0.959	0.817	0.995	0.898
Jaffal	0.716	0.878	0.789	0.609	0.893	0.724
FusionSeeker	0.807	0.838	0.822	0.633	0.852	0.726
Tool	Pac22			Ont22		
	Sensitivity	Precision	F1	Sensitivity	Precision	F1
IsoQuant	0.818	0.879	0.848	0.798	0.876	0.835
LongGF	0.749	0.888	0.813	0.722	0.888	0.796
Jaffal	0.841	0.823	0.832	0.729	0.742	0.735
FusionSeeker	0.655	0.721	0.686	0.616	0.698	0.654

The results on these datasets indicate that IsoQuant-fusion is clearly better at recovering fusions from simulated data than the existing tools. It also achieved the best F1 in three

out of four datasets.

3.2 Real data

All existing tools output an unknown number of false positives, when applied to real data (Dorney et al., 2023, Hafstað et al., 2023). While the complete ground truth for real datasets cannot be fully established, cancer samples with experimentally validated gene fusions offer a benchmark against which fusion-detection results can be evaluated. Mitelman Database of Chromosome Aberrations and Gene Fusions in Cancer (Mitelman et al., 2026) and COSMIC database (Tate et al., 2025) are widely-used resources for curated fusion gene cases across multiple cancer types.

It is important to note that reported fusion events are often cancer-type specific, and real sequencing data generally exhibit a higher prevalence of false positives and technical artifacts compared with simulated data. In addition, databases include a vast amount on fusions that have been confirmed to exist but have no biological relevance.

Two PacBio sequenced datasets, MCF7 breast cancer cell line and HCT116 color cancer cell line were downloaded from NCBI Sequence Read Archive (SRA) under accessions SRR1853200 and SRR5009523, respectively. The ONT dataset of A549 acute myeloid leukemia cell line was also retrieved from SRA under accession SRR12048357.

The PacBio MCF7 dataset carries several known and verified fusion transcripts and has been widely used in comparison between fusion detection tools (Dorney et al., 2023, Chen et al., 2023). Half of IsoQuant-fusion’s results were known fusions, including RPS6KB1:VMP1, which was missed by the other tools. Table 3.2 presents the found fusions by four tools. In this case I have included also *potential* fusions by JAFFAL. The tool reported the well-known BCAS3–BCAS4 fusion as potential, supported by just one read, because it was reported as two separate – identical – entry rows. In **FusionValidator**, identical fusion candidates are merged at the end of the pipeline to avoid such double entries.

Curiously, FusionSeeker run for almost an hour with this dataset and found only one fusion when run with Gencode v49 annotation. With v22, FusionSeeker found also SLC25A24–NBPF6 and a likely artifact FBXO42–SZRD1. It is also worth noting that, according to IGV genome browser, the RP3-430N8.11:ARMCX3 and ARMCX3–TUG1 refer to the same event

HCT116 is a small dataset from which IsoQuant-fusion, LongGF and FusionSeeker didn’t

Table 3.2: Comparison of detected fusion genes across four fusion detection tools.

Gene Fusion	IsoQuant	LongGF	JAFFAL	FusionSeeker	Known
KRT8-SPTSSB	x	—	—	—	—
ARMCX3-TUG1	x	x	—	—	—
NBPF6-SLC25A24	x	x	x	—	yes
SPTSSB-H3-3B	x	—	—	—	—
VMP1-RPS6KB1	x	—	—	—	yes
BCAS3-BCAS4	x	x	x	—	yes
RPS6KB1-DIAPH3	—	—	x	x	yes
C16orf62-IQCK	—	—	x	—	—
DPH7-PNPLA7	—	—	x	—	yes
HIPK1-DENND2C	—	—	x	—	—
CCNI-C11orf80	—	—	—	—	—
NCOA3-ACTL6A	—	—	x	—	—
PMPCA-C1orf95	—	—	x	—	—
VPS35L-IQCK	—	x	—	—	yes
RP3-430N8.11:ARMCX3	—	—	x	—	—
AC099850.1:VMP1	—	—	x	—	—

find any fusions with default settings. JAFFAL found 10 fusions with 1 supporting read, labeled as potential fusions.

Running fusion detection on large real tumor datasets is essential for algorithm development. Simulated data rarely represents the full diversity of false positives that we observe in real samples. In the early stages of fusion detector development, the AML dataset output over 1400 fusions, of which a vast majority were multicopy fusions of promiscuous genes partnering with several random partners.

Acute myeloid leukemia is a genetically diverse disease, and gene fusions play a central role in its diagnosis and treatment (Engvall et al., 2020). Fusions we can expect to find in AML data include PML-RARA, CBFB-MYH11, DEK-NUP214, KAT6A-CREBBP, NPM1-MLF1, KMT2A rearrangements and NUP98 fusions.

For comparing the recovered true fusions between LongGF, JAFFAL, FusionSeeker and IsoQuant-fusion, I utilized the file `fusion_db.txt` from fusion_db repository Shiraishi, 2017. The file includes merged verified fusions lists from The Mitelman and COSMIC

databases and other widely cited sources. I compared the fusion detection results against the gene pairs and triples in this file. The results, were not quite what expected. All four tools output the RUNX1–RUNX1T1 fusion with high confidence. FusionSeeker’s result file, actually, featured exactly the same fusion three times. This fusion is a plausible finding in AML but the other expected fusions were not recovered with any of the tools. LongGF and Jaffal didn’t find any other validated fusions. FusionSeeker found RPS11–RPS6 – a ribosomal fusion than can be either a passenger fusion or a long-read specific artifact. IsoQuant-fusion found the largest amount of previously validated fusions: RUNX1–RUNX1T1, CHD8-RAB2B, ARF4-PDE12, ISM2-SPTLC2, BCAS3-MED13, HNRNPA3-NFE2L2, IP6K2-PRKAR2A, RAD52-WNK1, RFC5-WSB2, IGFBP7-POLR2B, DYNC1H1-HSP90AA1, RAI1-SREBF1. Most of these are likely passenger fusions but this would require more investigation. Table 3.3 lists the numbers of found fusions

Table 3.3: Found fusions in AML data by four fusion detection tools.

	IsoQuant	LongGF	JAFFAL	FusionSeeker
Found fusions	397	88	504	90
Known fusions	12	1	1	2

Even if a fusion caller predominantly outputs true positives with simulated data, results with real data are likely noisy and require further analysis. Most ideas for further development of IsoQuant-fusion stem from the real data runs. In future, I will test the algorithm with more cancer types and non-cancer data.

4 Discussion

FusionDetector demonstrates precision and F1 metrics similar to those of LongGF and may surpass it in sensitivity. In this same evaluation, JaffaL and FusionSeeker did not perform as strongly. The Precision of IsoQuant-fusion leaves some room for improvement.

4.1 Alternative approaches to long-read fusion detection

The pipeline of LongGF, FusionSeeker and IsoQuant’s fusion detector starts with aligning the sample fastq to reference genome. JAFFAL and AERON align the reads to reference transcriptome (Davidson et al., 2022, Rautiainen et al., 2020). This is a computationally effective solution since transcriptomes, especially human, are much smaller than genomes. Mapping directly to known transcripts simplifies gene name retrieval. However, fusions involving unannotated regions, novel breakpoints or rare isoforms are missed. Similar transcripts may be mapped ambiguously and cause false positives. In my study, the transcriptome approach by JAFFAL didn’t give it any benefit over other tools.

Parsing for supplementary alignments is a typical starting point in finding fusion candidates from long reads. GFHunter (Zhao et al., 2025) has a reverse approach in the sense that it utilizes SA tag scanning only at the end of the pipeline to validate fusion candidates. GFHunter further optimizes the alignment to the reference transcriptome by reconstructing the reference transcriptome using only protein-coding transcripts. The initial screening of fusion candidates is done using sparse matrix multiplication, where alignments are represented as a sparse 0-1 matrix representing the co-occurrence patterns of reads spanning different gene pairs. The developers state that GFHunter is 2–3 times faster and uses only 16–50 % of the memory required by other long-read fusion detection tools. GFHunter was not tested during this thesis project but the idea to use a minimal version of reference transcriptome to reduce time and space complexity is interesting.

GFVoter’s (Zhao et al., 2025) strategy of integrating results from two aligners and two fusion callers has the potential to reduce false positives. The authors report that GFVoter produces substantially fewer fusion calls on real datasets than LongGF, JAFFAL, or Fu-

sionSeeker individually. However, the installation process is relatively complex, and the tool currently hard-codes the Gencode v22 annotation, making it difficult to use more recent reference versions.

Precision is considered an important metric in fusion detection, and, as a result, most fusion callers prefer to filter out non-coding fusion partners, transcription readthrough, fusions of multiple genes and intergenic and intragenic fusions. However, recent research has identified several roles for long non-coding RNAs, enhancer RNAs (eRNAs), circular RNAs, and sense-antisense fusions in the formation, regulation, and downstream consequences of gene fusion events (Dorney et al., 2025 Gupta et al., 2018). Detection of these events is not straightforward because of their structural complexities such as circularity or antisense orientation and mapping biases introduced by repetitive or ambiguously mappable non-coding regions. Specialized methods exist for the detection of non-canonical fusions, such as the circular fusion caller for long reads by Zhang et al., 2021.

Fusion callers usually rely on gene transfer format files to annotate the genes in fusion events. This means only fusions with known genes can be detected (Dorney et al., 2023). SFyNCS is an annotation-free fusion caller that, in benchmarking, was able to recover over 5,000 fusion events missed by other short-read fusion callers (Zhong et al., 2023). An annotation-free approach can detect fusions involving novel genes and can be useful when working with genomic areas that are not fully annotated.

4.2 Advantages of IsoQuant-fusion

With the added fusion detector module, IsoQuant is presumably the first tool capable of detecting both gene isoforms and fusions from long-read RNA data. Although the pipelines are separate, the input file requirements, file preprocessing and event logging are unified. The fusion detection outputs an easily interpretable TSV file of fusion candidates.

Maintenance of fusion detection tools, whether for short or long reads, varies a lot. For example, the public repositories of LongGF and FusionSeeker haven't been updated for years. IsoQuant is actively maintained and the code has been updated to match the advancements in sequencing technologies.

4.3 Further development

IsoQuant’s isoform detection has recently been extended to support barcoded single-cell datasets. In the future, the fusion detection module could also be adapted to operate on single-cell data. Isoform detection has an annotation-free mode which allows for searching gene isoforms even from unannotated genomes. Fusion detection, however, cannot work properly without gene annotation file. Accessing the canonical gene names and their biotypes is crucial for filtering biologically irrelevant candidates from the results.

Correctly outputting fusions consisting of more than two genes is a challenge for all fusion detection tools. IsoQuant-fusion is able to recover such events, but not necessarily in the correct biological order. The whole architecture is currently designed for gene pairs.

Currently, fusion detection does not have any input parameters, but they could be added into the command-line interface. As of now, parameters can be modified in the source code, if needed. In some cases it might be useful to modify the required number of supporting reads to record a fusion, which is now hard-coded to be 2.

IsoQuant requires a one-time preprocessing step in which the input GTF file is converted into an indexed database. While performed only once per annotation, this operation is computationally expensive. Database queries also add to the computational burden. Currently, fusion detection performs most queries using interval tree which is much faster. It could make sense to discard database completely and store all gene and exon information in the interval tree. I have tried this, but the results were not ideal. There should also be an option to store the tree structure in some form since interval tree conversion can take several minutes of time for a large annotation.

Alternative splicing mechanisms applied to the fused gene structure can generate multiple alternative transcript isoforms of fusion genes (Felton et al., 2023). Alternative isoforms can have different biological and clinical consequences. IsoQuant-fusion pipeline does not yet account for differential use of exons or other possible isoforms. Detecting and annotating such cases would be a logical next step since IsoQuant’s isoform detection does not consider fusion transcripts.

A machine learning model, trained with validated real fusions and simulated fusions could help in detecting less typical types of fusion transcripts or aid in validation and filtering of fusion candidates. One of the few published fusion detection tools utilizing machine learning is scFusion (Jin et al., 2022). It detects fusions from short single-cell RNA reads

with the help of a statistical model and a deep learning model. Machine learning aided approach could be suitable especially when working with single-cell data since it is more prone to produce false positives than bulk data. The most useful point of IsoQuant-fusion to use machine learning would probably be at the validation stage of the pipeline, where we can formulate a classification case: "Given a candidate fusion with biological and technical attributes, predict whether it is real."

Fusions output by a fusion detection pipeline can be classified using machine learning, as was done in the works by Hafstað et al., 2023 and Haas et al., 2023. Considering clinical settings, users might have the need to detect special types of fusion transcripts. A possible classification for the output could be:

- Artifact / Non-biological
- Biological but likely non-pathogenic
- Potentially pathogenic
- Clinically relevant
- Therapeutically actionable

The last class would include kinase fusions and fusions involving genes with documented sensitivity or resistance to targeted inhibitors, such as ALK, LTK, FGFR, RET and TRK, for example. Kinase fusions often constitute actionable driver events and represent good candidates for targeted therapy (Vellichirammal et al., 2020).

I plan to continue developing fusion detector to implement the aforementioned features.

5 Conclusions

Long-read RNA sequencing is highly effective for fusion detection because individual reads span entire transcripts, allowing the full fusion junction – including exonic structure and isoform context – to be captured in a single uninterrupted sequence. This eliminates the ambiguity and assembly errors common in short-read methods. Provided that long reads are aligned in a splice-aware mode, detecting chimeric reads from the resulting alignment file is relatively straightforward.

I developed a fusion detection module into isoform detection tool IsoQuant to effectively detect fusion transcripts from long RNA sequencing data. The method achieves higher sensitivity than existing long-read fusion callers, while preserving high precision.

A central, unresolved issue in fusion analysis is whether computational tools should report every technically valid fusion event or restrict the output to those that are likely to be biologically meaningful. In many cases a fusion event may not result in a functional protein but it might still be a valuable biomarker. This thesis work focuses on computer science, with the aim of producing easily interpretable results that can subsequently be filtered and analyzed by experts in biology or medicine.

Fusion detection module consists of four classes `GenomicIntervalIndex`, `FusionDetector`, `FusionMetadata` and `FusionValidator`. It is used via command-line interface of IsoQuant and requires the same input files - the sample FASTQ or BAM, reference FASTA and a FeatureDB or a GTF file. The `requirements_fusion.txt` file includes the libraries that need to be installed in addition to IsoQuant libraries – `mappy` and `intervaltree`. To detect fusions instead of isoforms, user needs to enter `-fusion` in the input command.

Bibliography

- Annala, M. J., Parker, B. C., Zhang, W., and Nykter, M. (2013). “Fusion genes and their discovery using high throughput sequencing”. In: *Cancer Letters*, 340(2), pp. 192–200. DOI: [10.1016/j.canlet.2013.01.011](https://doi.org/10.1016/j.canlet.2013.01.011).
- Calabrese, C., Davidson, N. R., Demircioğlu, D., Fonseca, N. A., He, Y., Kahles, A., Lehmann, K.-V., Liu, F., Shiraishi, Y., Soulette, C. M., Urban, L., Greger, L., Li, S., Liu, D., Perry, M. D., Xiang, Q., Zhang, F., Zhang, J., Bailey, P., and Zhang, Z. (2020). “Genomic basis for RNA alterations in cancer”. In: *Nature*, 578(7793), pp. 129–136. DOI: [10.1038/s41586-020-1970-0](https://doi.org/10.1038/s41586-020-1970-0).
- Caldas, P., Luz, M., Baseggio, S., et al. (2024). “Transcription readthrough is prevalent in healthy human tissues and associated with inherent genomic features”. In: *Communications Biology*, 7, p. 100. DOI: [10.1038/s42003-024-05779-5](https://doi.org/10.1038/s42003-024-05779-5).
- Chen, Y., Wang, Y., Chen, W., Tan, Z., Song, Y., Chen, H., and Chong, Z. (2023). “Gene fusion detection and characterization in long-read cancer transcriptome sequencing data with FusionSeeker”. In: *Cancer Research*, 83(1), pp. 28–33. DOI: [10.1158/0008-5472.CAN-22-1628](https://doi.org/10.1158/0008-5472.CAN-22-1628).
- Davidson, N. and Oshlack, A. (2021). *JAFFA version 2.2*. figshare. Software. Available online. DOI: [10.6084/m9.figshare.17272499.v1](https://doi.org/10.6084/m9.figshare.17272499.v1).
- Davidson, N. M., Chen, Y., Sadras, T., Ryland, G. L., Blombery, P., Ekert, P. G., Göke, J., and Oshlack, A. (2022). “JAFFAL: detecting fusion genes with long-read transcriptome sequencing”. In: *Genome Biology*, 23(1), p. 10. DOI: [10.1186/s13059-021-02588-5](https://doi.org/10.1186/s13059-021-02588-5).
- Dorney, R., Dhungel, B. P., Rasko, J. E. J., Hebbard, L., and Schmitz, U. (2023). “Recent advances in cancer fusion transcript detection”. In: *Briefings in Bioinformatics*, 24(1). DOI: [10.1093/bib/bbac519](https://doi.org/10.1093/bib/bbac519).
- Dorney, R., Reis-das-Mercês, L., and Schmitz, U. (2025). “Architects and partners: the dual roles of non-coding RNAs in gene fusion events”. In: *Computational Biology of Non-Coding RNA*. Ed. by X. Lai, S. Gupta, and J. Vera Gonzalez. Vol. 2883. Methods in Molecular Biology. New York, NY: Humana, pp. 231–255. DOI: [10.1007/978-1-0716-4290-0_10](https://doi.org/10.1007/978-1-0716-4290-0_10).
- Engvall, M., Cahill, N., Jönsson, B. I., Höglund, M., Hallböök, H., and Cavelier, L. (July 2020). “Detection of leukemia gene fusions by targeted RNA-sequencing in routine diag-

- nostics”. In: *BMC Medical Genomics*, 13(1), p. 106. DOI: [10.1186/s12920-020-00739-4](https://doi.org/10.1186/s12920-020-00739-4).
- Felton, C., Tang, A. D., Knisbacher, B. A., Wu, C. J., and Brooks, A. N. (2023). “Detection of alternative isoforms of gene fusions from long-read RNA-seq with FLAIR-fusion”. In: *bioRxiv*. DOI: [10.1101/2022.08.01.502364](https://doi.org/10.1101/2022.08.01.502364).
- Gupta, S. K., Luo, L., and Yen, L. (2018). “RNA-mediated gene fusion in mammalian cells”. In: *Proceedings of the National Academy of Sciences of the United States of America*, 115(52), E12295–E12304. DOI: [10.1073/pnas.1814704115](https://doi.org/10.1073/pnas.1814704115).
- Haas, B. J., Dobin, A., Ghandi, M., Gillani, R., Kasif, S., Regev, A., et al. (2023). “Targeted in silico characterization of fusion transcripts in tumor and normal tissues via FusionInspector”. In: *Cell Reports Methods*, 3(5), p. 100467. DOI: [10.1016/j.crmeth.2023.100467](https://doi.org/10.1016/j.crmeth.2023.100467).
- Haas, B. J., Dobin, A., Li, B., Stransky, N., Pochet, N., and Regev, A. (2019). “Accuracy assessment of fusion transcript detection via read-mapping and de novo fusion transcript assembly-based methods”. In: *Genome Biology*, 20(1), p. 213. DOI: [10.1186/s13059-019-1842-9](https://doi.org/10.1186/s13059-019-1842-9).
- Hafstað, V., Häkkinen, J., and Persson, H. (2023). “Fast and sensitive validation of fusion transcripts in whole-genome sequencing data”. In: *BMC Bioinformatics*, 24(1), p. 359. DOI: [10.1186/s12859-023-05489-5](https://doi.org/10.1186/s12859-023-05489-5).
- Jin, Z., Huang, W., Shen, N., Li, J., Wang, X., Dong, J., Park, P. J., and Xi, R. (2022). “Single-cell gene fusion detection by scFusion”. In: *Nature Communications*, 13(1), p. 1084. DOI: [10.1038/s41467-022-28661-6](https://doi.org/10.1038/s41467-022-28661-6).
- Karaoglanoglu, F., Chauve, C., and Hach, F. (2022). “Genion, an accurate tool to detect gene fusion from long transcriptomics reads”. In: *BMC Genomics*, 23(1), p. 129. DOI: [10.1186/s12864-022-08339-5](https://doi.org/10.1186/s12864-022-08339-5).
- Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and Subgroup, 1. G. P. D. P. (2009). “The Sequence Alignment/Map format and SAMtools”. In: *Bioinformatics*, 25(16), pp. 2078–2079. DOI: [10.1093/bioinformatics/btp352](https://doi.org/10.1093/bioinformatics/btp352).
- Li, H. (2018). “Minimap2: pairwise alignment for nucleotide sequences”. In: *Bioinformatics*, 34(18), pp. 3094–3100. DOI: [10.1093/bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191).
- Liu, Q., Hu, Y., Stucky, A., Fang, L., Zhong, J. F., and Wang, K. (2020). “LongGF: computational algorithm and software tool for fast and accurate detection of gene fusions by long-read transcriptome sequencing”. In: *BMC Genomics*, 21(Suppl 11), p. 793. DOI: [10.1186/s12864-020-07207-4](https://doi.org/10.1186/s12864-020-07207-4).

- Liu, Y., Lu, Z., Wang, Y., and Jiang, T. (2025). “GFHunter enables accurate and efficient gene fusion detection in long-read cancer transcriptomes”. In: *bioRxiv*. Preprint; not certified by peer review. DOI: [10.1101/2025.02.23.639788](https://doi.org/10.1101/2025.02.23.639788). URL: <https://doi.org/10.1101/2025.02.23.639788>.
- Masuda, K., Sota, Y., and Matsuda, H. (2024). “Gene fusion detection in long-Read transcriptome datasets from multiple cancer cell lines”. In: *Frontiers in Bioscience (Landmark. Print)*, 29(12), p. 413. DOI: [10.31083/j.fbl2912413](https://doi.org/10.31083/j.fbl2912413). URL: <https://doi.org/10.31083/j.fbl2912413>.
- Mitelman, F., Johansson, B., and Mertens, F. (2026). *Mitelman Database of Chromosome Aberrations and Gene Fusions in Cancer*. <https://mitelmandatabase.isb-cgc.org>. Edited by Mitelman F, Johansson B and Mertens F.
- Pertea, G. and Pertea, M. (2020). “GFF Utilities: GffRead and GffCompare [version 2; peer review: 3 approved]”. In: *F1000Research*, 9, p. 304. DOI: [10.12688/f1000research.23297.2](https://doi.org/10.12688/f1000research.23297.2).
- Prijibelski, A. D., Mikheenko, A., Joglekar, A., Smetanin, A., Jarroux, J., Lapidus, A. L., and Tilgner, H. U. (2023). “Accurate isoform discovery with IsoQuant using long reads”. In: *Nature Biotechnology*, 41(7), pp. 915–918. DOI: [10.1038/s41587-022-01565-y](https://doi.org/10.1038/s41587-022-01565-y).
- Rautiainen, M., Durai, D., Chen, Y., Xin, L., Low, H. M., Goeke, J., Marschall, T., and Schulz, M. (2020). “AERON: Transcript quantification and gene-fusion detection using long reads”. In: *bioRxiv*. Preprint. DOI: [10.1101/2020.01.27.921338](https://doi.org/10.1101/2020.01.27.921338).
- Rogers, K. (2026). “Chimera”. In: *Encyclopaedia Britannica*. Accessed 6 April 2026. URL: <https://www.britannica.com/science/chimera-genetics>.
- Rowley, J. D. (1973). “A new consistent chromosomal abnormality in chronic myelogenous leukaemia identified by quinacrine fluorescence and giemsa staining”. In: *Nature*, 243, pp. 290–293. DOI: [10.1038/243290a0](https://doi.org/10.1038/243290a0).
- Samtools developers (2024). *Sequence Alignment/Map Format Specification (Version 1)*. Accessed: 2025-11-20. URL: <https://samtools.github.io/hts-specs/SAMv1.pdf>.
- Schäfer, R. A. and Yang, R. (2025). “A comprehensive benchmark of tools for efficient genomic interval querying”. In: *Briefings in Bioinformatics*, 26(4), bbaf379. DOI: [10.1093/bib/bbaf379](https://doi.org/10.1093/bib/bbaf379).
- Shiraishi, Y. (2017). *fusion_db: Database for fusion genes*. https://github.com/friend1ws/fusion_db. Accessed: 2026-03-02.
- Shtivelman, E., Lifshitz, B., Gale, R. P., and Canaani, E. (1985). “Fused transcript of ABL and BCR genes in chronic myelogenous leukaemia”. In: *Nature*, 315, pp. 550–554. DOI: [10.1038/315550a0](https://doi.org/10.1038/315550a0).

- Su, Y., Yu, Z., Jin, S., Ai, Z., Yuan, R., Chen, X., Xue, Z., Guo, Y., Chen, D., Liang, H., Liu, Z., and Liu, W. (2024). “Comprehensive assessment of mRNA isoform detection methods for long-read sequencing data”. In: *Nature Communications*, 15(1), Article 3972. DOI: [10.1038/s41467-024-48117-3](https://doi.org/10.1038/s41467-024-48117-3).
- Tate, J. G., Bamford, S., Jubb, H. C., Sondka, Z., Beare, D. M., Bindal, N., Boutselakis, H., Cole, C. G., Creatore, C., Dawson, E., Fish, P., Harsha, B., Hathaway, C., Jupe, S. C., Kok, C. Y., Noble, K., Ponting, L., Ramshaw, C. C., Rye, C. E., Speedy, H. E., Stefancsik, R., Thompson, S. L., Wang, S., Ward, S., Campbell, P. J., and Forbes, S. A. (2025). “COSMIC: the Catalogue Of Somatic Mutations In Cancer”. In: *Nucleic Acids Research*, 47(D1), pp. D941–D947. DOI: [10.1093/nar/gky1015](https://doi.org/10.1093/nar/gky1015). URL: <https://www.cosmicdb.org>.
- Uhrig, S., Ellermann, J., Walther, T., Burkhardt, P., Fröhlich, M., Hutter, B., Toprak, U. H., Neumann, O., Stenzinger, A., Scholl, C., Fröhling, S., and Brors, B. (2021). “Accurate and efficient detection of gene fusions from RNA sequencing data”. In: *Genome Research*, 31(3), pp. 448–460. DOI: [10.1101/gr.257246.119](https://doi.org/10.1101/gr.257246.119).
- Vellichirammal, N. N., Albahrani, A., Banwait, J. K., Mishra, N. K., Li, Y., Roychoudhury, S., Kling, M. J., Mirza, S., Bhakat, K. K., Band, V., Joshi, S. S., and Guda, C. (2020). “Pan-cancer analysis reveals the diverse landscape of novel sense and antisense fusion transcripts”. In: *Molecular Therapy. Nucleic Acids*, 19, pp. 1379–1398. DOI: [10.1016/j.omtn.2020.01.023](https://doi.org/10.1016/j.omtn.2020.01.023).
- Wick, R. R. (2019). “Badread: simulation of error-prone long reads”. In: *Journal of Open Source Software*, 4(36), p. 1316. DOI: [10.21105/joss.01316](https://doi.org/10.21105/joss.01316). URL: <https://doi.org/10.21105/joss.01316>.
- Yun, J. W., Yang, L., Park, H. Y., and al., et (2020). “Dysregulation of cancer genes by recurrent intergenic fusions”. In: *Genome Biology*, 21, p. 166. DOI: [10.1186/s13059-020-02076-2](https://doi.org/10.1186/s13059-020-02076-2).
- Zhang, J., Hou, L., Zuo, Z., Ji, P., Zhang, X., Xue, Y., and Zhao, F. (2021). “Comprehensive profiling of circular RNAs with nanopore sequencing and CIRI-long”. In: *Nature Biotechnology*, 39(7), pp. 836–845. DOI: [10.1038/s41587-021-00842-6](https://doi.org/10.1038/s41587-021-00842-6). URL: <https://doi.org/10.1038/s41587-021-00842-6>.
- Zhao, X., Ren, Z., Qi, J., et al. (2025). “Gene fusion detection in long-read transcriptome sequencing data with GFvoter”. In: *BMC Genomics*, 26, p. 670. DOI: [10.1186/s12864-025-11866-6](https://doi.org/10.1186/s12864-025-11866-6).

Zhong, X., Luan, J., Yu, A., Lee-Hassett, A., Miao, Y., and Yang, L. (2023). “SFyNCS detects oncogenic fusions involving non-coding sequences in cancer”. In: *Nucleic Acids Research*, 51(18), e96. DOI: [10.1093/nar/gkad705](https://doi.org/10.1093/nar/gkad705).

