

How to Analyze RNA-Sequencing Data Generated using KAPA RNA Products: A Comprehensive Bioinformatics Guide

AUTHOR

Jieqiong Dai

Senior Bioinformatics Scientist

Roche Diagnostics · Wilmington, MA, USA

For Research Use Only. Not for use in diagnostic procedures.

Overview

RNA-Sequencing (RNA-seq) has become an essential tool for transcriptome profiling, enabling researchers to investigate gene expression patterns, identify novel transcripts, detect alternative splicing events, and discover gene fusions. The analysis of RNA-seq data involves complex computational processes that require comprehensive quality control and robust analytical methods.

This document provides practical bioinformatics guidance for analyzing RNA-seq data generated using [KAPA RNA EvoPrep Kits](#) for whole transcriptome analysis, or targeted with [KAPA HyperCap Probes](#). It presents comprehensive instructions organized around three core capabilities:

- 1. Complete Analysis Coverage** — Step-by-step methods spanning the core RNA-seq workflow (data processing and quality assessment, read alignment, and gene expression quantification) with specialized extensions for advanced applications including fusion detection and splicing analysis. These allow researchers to select analyses relevant to their biological questions.
- 2. Multi-Stage Quality Control** — Guidance on the various quality metrics to monitor at primary and secondary analysis stages. This helps researchers identify data characteristics and technical considerations.

For Research Use Only. Not for use in diagnostic procedures.

3. Comprehensive Methodological Guidance — Detailed command-line procedures with explanations of key parameters, outputs, and practical recommendations. This provides researchers with thorough understanding of each analysis stage.

This guidance is designed for bioinformaticians, computational biologists, and researchers with experience in command-line tools and RNA-seq analysis. Familiarity with Unix/Linux command-line operations, NGS data formats, and RNA-seq concepts is assumed.

For a broader educational overview of RNA-seq data analysis, including quality control metrics interpretation and downstream applications, see the accompanying webinar presented through Labroots: [How to Analyze RNA-Sequencing Data: A Bioinformatics Overview](#).

Workflow Overview

The RNA-seq bioinformatics workflow described in this guidance spans primary, secondary, and tertiary analysis stages. Key functionalities are illustrated in Figure 1 and include the following, with the corresponding methodology steps noted for each:

- Raw data processing and quality assessment (FastQC, Fastp, Seqtk) — Steps 1–4
- Read alignment and expression quantification (STAR, Samtools, Salmon) — Steps 5–7, 11
- Multi-stage quality control (Qualimap, RNA-SeQC) — Steps 8–10
- Fusion detection (STAR-Fusion) — Step 12
- Transcript assembly and annotation comparison (StringTie, GffCompare, GffRead) — Steps 13–15
- Integrated quality reporting (MultiQC) — Step 16

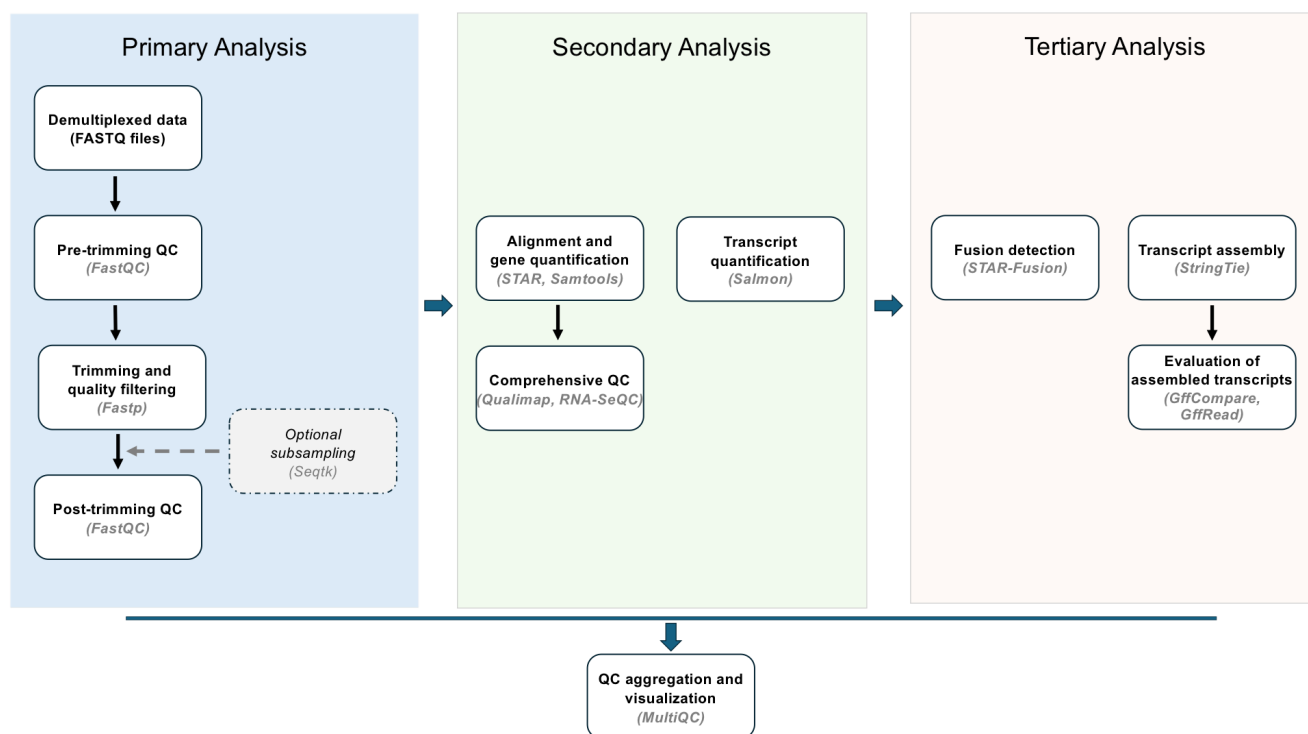


Figure 1. Overview of the RNA-seq bioinformatics workflow described in this guidance, spanning primary, secondary, and tertiary analysis stages.

A comprehensive guide to interpreting key quality control metrics, including what each metric measures and what it tells you, is provided at the end of this document.

Overview of Bioinformatics Tools

This guidance leverages various peer-reviewed open-source bioinformatics tools selected for their robustness, accuracy, and widespread adoption in the research community. The software versions used in developing this guidance are listed below.

- [Fastp v1.0.1](#)
- [FastQC v0.12.1](#)
- [GffCompare v0.12.10](#)
- [GffRead v0.12.7](#)
- [MultiQC v1.33](#)
- [QualiMap v2.3](#)
- [RNA-SeQC v2.4.2](#)
- [Salmon v1.10.3](#)
- [Samtools v1.6](#)
- [Seqtk v1.5](#)
- [STAR v2.7.11b](#)
- [STAR-Fusion v1.15.1](#)
- [StringTie v3.0.3](#)

NOTE: Open-source software tools may change over time, and different software versions may require different function calls and/or flags. Roche cannot guarantee or be held responsible for the results obtained using third-party tools described herein. Users should refer to the tool authors for support and documentation.

Methodology

The following sections provide detailed commands and instructions for each step of the RNA-seq bioinformatics workflow.

NOTICE: All code indicated in **ITALICS** requires user input and cannot be simply copied and pasted. This is indicated where relevant.

Required input files and resources:

- Paired-end reads in compressed FASTQ format: `{SAMPLE_ID}_R1.fastq.gz`, `{SAMPLE_ID}_R2.fastq.gz`. The data should be demultiplexed and concatenated across lanes for each sample.
- Reference genome FASTA file (e.g., `GRCh38.fa`) and annotation GTF file (e.g., GENCODE, Ensembl, RefSeq). Ensure compatibility with the reference genome build.
- Collapsed GTF file: Required for rRNA rate estimation using RNA-SeQC. Must be collapsed to one transcript per gene. Generate using [collapse_annotation.py](#).
- Salmon transcriptome index: Required for transcript quantification. Pre-built indices for some common organisms are available [here](#).
- STAR-Fusion CTAT resource genome library: Required for fusion detection. Libraries for some common organisms are available [here](#).
- Optional: Target region BED file for targeted enrichment analysis.

Important Note on UMI Processing: If KAPA Universal UMI Adapters were used during library preparation, follow the detailed methods described in the **KAPA Universal UMI Adapter Instructions for Use (IFU)** for processing. UMI-specific procedures are not covered in this workflow.

1. Assess sequencing quality of raw reads, using FastQC

Command:

```
fastqc {SAMPLE_ID}_R1.fastq.gz \  
      {SAMPLE_ID}_R2.fastq.gz \  
      -f fastq \  
      --noextract \  
      -t {threads} \  
      -o {output_directory}/pretrim_qc
```

Purpose: Evaluate raw sequencing data quality.

Major Output:

- `{SAMPLE_ID}_R1_fastqc.html` : Interactive HTML QC report for read 1
- `{SAMPLE_ID}_R2_fastqc.html` : Interactive HTML QC report for read 2
- `{SAMPLE_ID}_R1_fastqc.zip` : Data files for read 1
- `{SAMPLE_ID}_R2_fastqc.zip` : Data files for read 2

2. Perform adapter trimming and quality filtering, using Fastp

Command:

```
fastp -i {SAMPLE_ID}_R1.fastq.gz \  
      -I {SAMPLE_ID}_R2.fastq.gz \  
      -o {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
      -O {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
      -w {threads} \  
      -j {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimming.fastp.json \  
      -h {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimming.fastp.html \  
      {additional_parameters}
```

Purpose: Remove sequencing adapters, trim low-quality bases, and filter low-quality reads to improve downstream analysis accuracy.

Recommended Additional Parameters:

- `-q 20`: Minimum base quality score (Phred scale) for quality filtering
- `-u 40`: Maximum percentage of bases allowed to be unqualified (default value)
- `-l 50`: Minimum read length
- `-c`: Enable base correction in overlapped regions for paired-end reads
- `-3`: Enable 3' end sliding window quality trimming
- `-g`: Enable poly-G tail trimming
- `-x`: Enable poly-X trimming in 3' ends
- `-p`: Enable overrepresented sequence analysis

Major Output:

- `{SAMPLE_ID}_trimmed_R1.fastq.gz`: Trimmed read 1 FASTQ file
- `{SAMPLE_ID}_trimmed_R2.fastq.gz`: Trimmed read 2 FASTQ file
- `{SAMPLE_ID}_trimming.fastp.json`: JSON report with detailed QC metrics
- `{SAMPLE_ID}_trimming.fastp.html`: Interactive HTML QC report

3. Optional: Subsample reads, using Seqtk

Command:

```
seqtk sample -s {random_seed} \  
    {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
    {fraction_or_number} \  
    | gzip > \  
    {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R1.fastq.gz  
seqtk sample -s {random_seed} \  
    {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
    {fraction_or_number} \  
    | gzip > \  
    {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R2.fastq.gz
```

Purpose: Downsample reads to a desired depth for testing, normalization, or computational efficiency. Use the same random seed for paired-end files to maintain read pairing.

Key Parameters:

- `-s`: Random seed for reproducible subsampling (use identical seed for R1 and R2)
- `{fraction_or_number}`: Either fraction (e.g., 0.5 for 50%) or absolute number of reads (e.g., 10000000 for 10M reads)

Major Output:

- `{SAMPLE_ID}_subsampled_R1.fastq.gz`: Subsampled read 1 file
- `{SAMPLE_ID}_subsampled_R2.fastq.gz`: Subsampled read 2 file

4. Assess sequencing quality of reads post-trimming and after subsampling (if performed), using FastQC

Command:

```
fastqc {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
      {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
      -f fastq \  
      --noextract \  
      -t {threads} \  
      -o {output_directory}/posttrim_qc
```

NOTE: If subsampling was performed in Step 3, use the subsampled files (`{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R1.fastq.gz` and `{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R2.fastq.gz`) instead of the trimmed files.

Purpose: Verify quality improvements after trimming and assess the effectiveness of the trimming process.

Major Output:

- `{SAMPLE_ID}_trimmed_R1_fastqc.html`: Interactive HTML QC report for trimmed read 1
- `{SAMPLE_ID}_trimmed_R2_fastqc.html`: Interactive HTML QC report for trimmed read 2

- `{SAMPLE_ID}_trimmed_R1_fastqc.zip` : Data files for trimmed read 1
- `{SAMPLE_ID}_trimmed_R2_fastqc.zip` : Data files for trimmed read 2

5. Generate genome index for STAR (if not already available)

Command:

```
STAR --runMode genomeGenerate \  
    --runThreadN {threads} \  
    --genomeDir {output_index_directory} \  
    --genomeFastaFiles {reference_genome_fasta} \  
    --sjdbGTFfile {GTF_annotation_file} \  
    --sjdbOverhang {read_length_minus_1}
```

Purpose: Generate a genome index for STAR aligner, incorporating splice junction annotations for RNA-seq alignment. This is a one-time setup task per genome build.

Key Parameter:

- `--sjdbOverhang` : Read length minus 1 (e.g., 99 for 100bp reads, 149 for 150bp reads). This value optimizes the splice junction database for your read length. In most cases, the default value of 100 will work as well as the ideal value.

Major Output:

- STAR genome index files in the specified directory

6. Map reads to the reference genome, using STAR

Command:

```
STAR --runThreadN {threads} \  
    --genomeDir {STAR_index_directory} \  
    --sjdbGTFfile {GTF_annotation_file} \  
    --readFilesIn \  
        {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
        {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
    --readFilesCommand zcat \  
    --outFileNamePrefix {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_ \  
    --outSAMtype BAM SortedByCoordinate \  
    --quantMode GeneCounts \  
    --twopassMode Basic \  
    {additional_parameters}
```

NOTE:

- If subsampling was performed in Step 3, use the subsampled files (`{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R1.fastq.gz` and `{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R2.fastq.gz`) instead of the trimmed files.
- Ensure STAR index was built with the same STAR version used for alignment to avoid compatibility issues.
- For downstream gene expression analysis, combine the `ReadsPerGene.out.tab` files from all samples using the relevant column according to your library's strand specificity into a gene count matrix using your preferred method. KAPA RNA EvoPrep Kits produce strand-specific libraries, typically using the 4th column "Stranded reverse".

Purpose: Align trimmed reads to the reference genome using splice-aware alignment.

Key Parameters:

- `--readFilesCommand zcat` : Decompression command for gzipped FASTQ files
- `--outSAMtype BAM SortedByCoordinate` : Output format (sorted BAM for downstream tools)
- `--quantMode GeneCounts` : Enable gene-level read counting during alignment
- `--twopassMode Basic` : Two-pass alignment for improved junction detection

Recommended Additional Parameter:

- `--outSAMstrandField intronMotif`: Add XS strand attribute to spliced alignments by inferring the strand from genomic motifs. This is required by transcript assemblers such as StringTie and Cufflinks

Major Output:

- `{SAMPLE_ID}_Aligned.sortedByCoord.out.bam`: Sorted BAM file with aligned reads
- `{SAMPLE_ID}_ReadsPerGene.out.tab`: Gene-level read counts (tab-delimited with 4 columns)
 - Column 1: Gene ID
 - Column 2: Unstranded count
 - Column 3: Stranded forward count (1st read strand aligned with RNA)
 - Column 4: Stranded reverse count (2nd read strand aligned with RNA)
- `{SAMPLE_ID}_SJ.out.tab`: Splice junction information
- `{SAMPLE_ID}_Log.final.out`: Alignment statistics summary

7. Index the aligned BAM file, using Samtools

Command:

```
samtools index \  
    {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_Aligned.sortedByCoord.out.bam
```

Purpose: Create an index file for the aligned BAM file to enable efficient random access for downstream analysis tools.

Major Output:

- `{SAMPLE_ID}_Aligned.sortedByCoord.out.bam.bai`: BAM index file

8. Assess alignment quality, using Qualimap BAMQC

Command:

```
qualimap bamqc -bam \
    {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_Aligned.sortedByCoord.out.bam \
    --java-mem-size={memory_allocation} \
    -c \
    -nt {threads} \
    -outdir {output_directory}/qualimap_bamqc/{SAMPLE_ID} \
    -p {strand_specificity} \
    {additional_parameters}
```

NOTE:

- The strand specificity setting should match the library preparation protocol. KAPA RNA EvoPrep Kits produce strand-specific libraries, typically using `strand-specific-reverse`.
- For targeted enrichment applications, use `-gff` to provide feature file with regions of interest in a GFF/GTF or BED format. When provided, the QualiMap BAMQC HTML report will include target enrichment performance metrics such as on-target read counts and percentages.

Purpose: Perform comprehensive BAM file quality assessment including mapping quality, insert size distribution, coverage statistics, and GC content analysis.

Key Parameters:

- `-c`: Paint chromosome limits inside charts for better visualization
- `-p`: Strand specificity protocol (`non-strand-specific`, `strand-specific-forward`, `strand-specific-reverse`)

Recommended Additional Parameter:

- `-gd`: Species to compare with genome GC distribution (HUMAN: hg19, hg38; MOUSE: mm9, mm10)

Major Output:

- `qualimapReport.html`: Interactive HTML report with plots and summary statistics
- `genome_results.txt`: Plain-text summary statistics

9. Assess RNA-seq specific quality metrics, using Qualimap RNAseq

Command:

```
qualimap rnaseq -bam \  
    {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_Aligned.sortedByCoord.out.bam \  
    --java-mem-size={memory_allocation} \  
    -gtf {GTF_annotation_file} \  
    -outdir {output_directory}/qualimap_rnaseq/{SAMPLE_ID} \  
    -p {strand_specificity} \  
    -pe
```

NOTE: The strand specificity setting should match the library preparation protocol. KAPA RNA EvoPrep Kits produce strand-specific libraries, typically using `strand-specific-reverse`.

Purpose: Generate RNA-seq specific quality metrics including genomic origin of reads, junction analysis, transcript coverage and 5'-3' bias computation.

Key Parameters:

- `-p`: Strand specificity protocol (`non-strand-specific`, `strand-specific-forward`, `strand-specific-reverse`)
- `-pe`: Paired-end sequencing data

Major Output:

- `qualimapReport.html`: Interactive HTML report with RNA-seq specific plots and statistics
- `rnaseq_qc_results.txt`: Plain-text summary statistics

10. Perform additional RNA-seq quality assessment, using RNA-SeQC

Command:

```
rnaseqc {collapsed_GTF_file} \  
    {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_Aligned.sortedByCoord.out.bam \  
    {output_directory}/rnaseqc/{SAMPLE_ID} \  
    -s {SAMPLE_ID} \  
    --stranded {strand_specificity} \  
    --coverage
```

NOTE: The strand specificity setting should match the library preparation protocol. KAPA RNA EvoPrep Kits produce strand-specific libraries, typically using **RF**.

Purpose: Conduct additional RNA-seq quality assessment such as rRNA rate estimation and strand specificity verification.

Key Parameters:

- **{collapsed_GTF_file}**: Collapsed annotation file (see **Required input files and resources** section for details)
- **--stranded**: Strand specificity setting. Use **RF** for reverse-stranded libraries, **FR** for forward-stranded libraries. Omit this parameter for non-stranded libraries.
- **--coverage**: Generate per-transcript coverage statistics table in addition to summary coverage metrics

Major Output:

- **{SAMPLE_ID}.metrics.tsv**: Summary QC metrics

11. Perform transcript-level quantification, using Salmon

Command:

```
salmon quant \  
  -i {Salmon_index_directory} \  
  -p {threads} \  
  -l A \  
  -1 {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
  -2 {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
  --validateMappings \  
  -o {output_directory}/transcript/{SAMPLE_ID}
```

NOTE: If subsampling was performed in Step 3, use the subsampled files (`{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R1.fastq.gz` and `{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R2.fastq.gz`) instead of the trimmed files.

Purpose: Quantify transcript-level expression using Salmon mapping-based mode.

Key Parameters:

- `-i` : Path to Salmon transcriptome index (see `Required input files and resources` section for details)
- `-l A` : Automatic detection of library type (strand specificity)
- `--validateMappings` : Perform selective alignment for more accurate quantification

Major Output:

- `quant.sf` : Transcript quantification file with effective length of the target transcript, TPM (Transcripts Per Million), and number of reads mapping to the transcript
- `lib_format_counts.json` : Library format (strand specificity) detection results
- `aux_info/meta_info.json` : Meta information about the run, including stats such as the number of observed and mapped fragments, details of the bias modeling etc

12. Perform gene fusion detection, using STAR-Fusion

Command:

```
STAR-Fusion \  
  --left_fq {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R1.fastq.gz \  
  --right_fq {output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_trimmed_R2.fastq.gz \  
  --genome_lib_dir {CTAT_resource_library} \  
  --CPU {threads} \  
  --output_dir {output_directory}/fusion/{SAMPLE_ID} \  
  --FusionInspector validate \  
  --examine_coding_effect \  
  --denovo_reconstruct
```

NOTE: If subsampling was performed in Step 3, use the subsampled files (`{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R1.fastq.gz` and `{output_directory}/trimmed/{SAMPLE_ID}/{SAMPLE_ID}_subsampled_R2.fastq.gz`) instead of the trimmed files.

Purpose: Identify gene fusion events from RNA-seq data using STAR-Fusion through two primary evidence types: split reads that span the fusion junction, and spanning fragments — read pairs whose two mates map to two different genes.

Key Parameters:

- `--genome_lib_dir` : Path to STAR-Fusion CTAT resource genome library
- `--FusionInspector validate` : Run FusionInspector to validate and visualize fusion evidence
- `--examine_coding_effect` : Predict impact of the fusion event on coding regions
- `--denovo_reconstruct` : Perform de novo assembly of fusion transcripts for full-length reconstruction

Major Output:

- `star-fusion.fusion_predictions.tsv` : Main fusion predictions file with detailed information on each detected fusion event
- `FusionInspector/` : Directory containing validation results and visualizations

Key Columns of `star-fusion.fusion_predictions.tsv`:

- `#FusionName`: Gene pair involved (`GeneA--GeneB`)
- `JunctionReadCount`: Number of split reads spanning the putative fusion junction
- `SpanningFragCount`: Number of read pairs spanning the fusion junction where paired mates map to different genes
- `SpliceType`: Whether the proposed breakpoint occurs at reference exon junctions as provided by the reference transcript structure annotations (ONLY_REF_SPLICE, INCL_NON_REF_SPLICE)
- `LeftGene`, `RightGene`: Gene symbols of fusion partners
- `LeftBreakpoint`, `RightBreakpoint`: Genomic coordinates of the breakpoints
- `FFPM`: Fusion Fragments Per Million total reads (normalized abundance)
- `LargeAnchorSupport`: Whether there are split reads that provide 'long' (at least 25 bases) alignments on both sides of the putative breakpoint (YES_LDAS or NO_LDAS)
- `annots`: Annotations for fusion transcripts through Fusion Annotator, which leverages resources including known cancer-relevant fusions and common false positives

13. Perform transcript assembly, using StringTie

Command:

```
stringtie {strand_parameters} \  
  -G {GTF_annotation_file} \  
  -p {threads} \  
  -A {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}_gene_abund.tab \  
  {additional_parameters} \  
  -o {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}.gtf \  
  {output_directory}/align/{SAMPLE_ID}/{SAMPLE_ID}_Aligned.sortedByCoord.out.bam
```

NOTE: The strand specificity setting should match the library preparation protocol. KAPA RNA EvoPrep Kits produce strand-specific libraries, typically using `--rf`.

Purpose: Assemble transcripts from aligned reads using reference-guided assembly to identify both known and novel transcripts and estimate gene abundances.

Key Parameters:

- `{strand_parameters}`: Library strand specificity. Use `--rf` for fr-firststrand stranded libraries, `--fr` for fr-secondsd stranded libraries, or omit for non-stranded libraries.
- `-A`: Gene abundance estimation output file
- `-o`: Output file for assembled transcripts in GTF format

Recommended Additional Parameters:

- `-m 180`: Minimum assembled transcript length (default: 200). A slightly lower threshold can help retain shorter but biologically relevant transcripts.
- `-j 3`: Minimum junction coverage (default: 1)
- `-c 1.5`: Minimum reads per bp coverage to consider for multi-exon transcript (default: 1)

Major Output:

- `{SAMPLE_ID}.gtf`: Assembled transcript structures in GTF format
- `{SAMPLE_ID}_gene_abund.tab`: Gene abundances in tab-delimited format

14. Compare assembled transcripts by StringTie with reference annotation, using GffCompare

Command:

```
gffcompare -r {GTF_annotation_file} \  
           -R -Q \  
           -s {reference_genome_fasta} \  
           {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}.gtf \  
           -o {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}_gffcmp
```

Purpose: Compare assembled transcripts from StringTie with a reference annotation to identify the relationships of the transcripts to the known annotation and assess the accuracy of the assembly.

Key Parameters:

- `-r`: Reference annotation used for comparison

- `-R`: Ignore reference transcripts not overlapped by assembled transcripts
- `-Q`: Ignore assembled transcripts not overlapped by reference transcripts

Major Output:

- `{SAMPLE_ID}_gffcmp.annotated.gtf`: Assembled transcripts with comparison annotations
- `{SAMPLE_ID}_gffcmp.stats`: Summary statistics of transcript comparison

15. Extract StringTie assembled transcript sequences, using GffRead

Command:

```
gffread {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}.gtf \
  -g {reference_genome_fasta} \
  -w {output_directory}/splicing/{SAMPLE_ID}/{SAMPLE_ID}_gffread.fa
```

Purpose: Convert assembled transcript GTF files to FASTA format for downstream sequence analysis and validation.

Major Output:

- `{SAMPLE_ID}_gffread.fa`: Transcript sequences in FASTA format

16. Summarize QC metrics of all samples into an interactive report, using MultiQC

Command:

```
multiqc {directory_containing_all_QC_metrics}/ \
  --title {your_title} \
  -o {output_directory}/multiqc \
  -f
```

Purpose: Integrate all quality control metrics from various analysis stages into a comprehensive, interactive HTML report for easy review and interpretation.

Major Output:

- `{your_title}_multiqc_report.html`: Integrated interactive QC report

Guidelines for Reviewing the MultiQC Report

- If you're unfamiliar with the MultiQC report, access the `Watch a tutorial video` button located at the top of the report, just above the `General Statistics` section. This video tutorial, provided by `MultiQC`, will guide you through navigating and comprehending the report.
- Prioritize your attention on the metrics presented within the `General Statistics` section. This section consolidates key quality control metrics from various tools. These metrics offer a comprehensive overview of the sequencing data quality and alignment performance.
- For a more in-depth exploration of additional metrics, you can access further information by clicking on the link associated with each bioinformatics tool. This link will direct you to the tool's documentation, offering comprehensive insights into the tool's functionalities and output metrics. Within certain sections, you will also come across a `Help` button positioned at the upper right corner of the pertinent metric. Clicking the `Help` button will reveal detailed information about the specific metric.

Key Quality Control Metrics: Interpretation Guide

Understanding key QC metrics is essential for assessing the quality and characteristics of RNA-seq data. The following guide describes what each metric measures, what it tells you, and common patterns observed in RNA-seq data. Acceptable ranges vary by experiment type, organism, tissue source, and research objectives, so metrics should always be evaluated in the context of your specific study design.

Sequencing Quality Metrics

1. Total Reads

- **What it measures:** Number of reads generated per sample
- **What it tells you:**
 - Sequencing depth achieved for each sample
 - Adequate depth depends on experimental goals (e.g., differential expression analysis, transcript discovery, fusion detection)

2. Read Length Distribution

- **What it measures:** Length distribution of sequencing reads
- **What it tells you:**
 - Confirms expected read length from sequencing protocol
 - Should match what was specified for the sequencing run

3. Insert Size Distribution (Paired-End)

- **What it measures:** Length distribution of the inserts located between the paired sequencing adapters
- **What it tells you:**
 - Library fragment characteristics
 - Should match library preparation protocol expectations

4. GC Content Distribution

- **What it measures:** Overall GC composition across reads
- **What it tells you:**
 - In RNA-seq, GC content distribution is expression-driven, reflecting which genes are expressed
 - Should be reasonably consistent across biological replicates

5. Percent Reads Passing Filter (Fastp)

- **What it measures:** Proportion of total raw reads that meet minimum quality and length requirements
- **What it tells you:**
 - Overall data quality based on filtering criteria
 - Low percentages indicate potential sequencing quality issues

6. Base Quality Scores

- **What it measures:** Per-base quality (Phred score) across read length, showing the confidence of each base call
- **What it tells you:**
 - Q30 indicates 99.9% accuracy; Q20 indicates 99% accuracy
 - Slight quality drop at read ends is normal for Illumina sequencing chemistry
 - Sudden mid-read drops may indicate sequencing issues

7. N Content (Ambiguous Base Calls)

- **What it measures:** Percentage of bases called as 'N' (undetermined)
- **What it tells you:**
 - Should be very low in good quality data
 - High N content indicates severe sequencing quality issues

8. Adapter Content

- **What it measures:** Percentage of reads containing adapter sequences

- **What it tells you:**

- Indicates fragments shorter than read length, causing the sequencer to read into adapter sequences
- Adapter sequences interfere with alignment and should be removed during preprocessing

Alignment Quality Metrics

9. Mapping Rate

- **What it measures:** Proportion of reads successfully aligned to the reference genome
- **What it tells you:**
 - Overall alignment success for the sample
 - Mapping rate depends on sample type, RNA quality, organism, and reference genome completeness

10. Uniquely Mapped Rate

- **What it measures:** Proportion of reads mapping to a single location in the reference genome
- **What it tells you:**
 - Specificity of read alignment
 - Multi-mapping reads occur naturally for gene families, paralogs, and repetitive elements
 - Most downstream analysis tools by default use only uniquely mapped reads to avoid ambiguity in gene assignment

RNA-Specific Quality Metrics

11. Transcript Mapping Rate

- **What it measures:** Proportion of reads successfully mapped to annotated transcripts
- **What it tells you:**
 - How well reads align to the transcriptome reference
 - Different from genome mapping — focuses on known transcript sequences
 - Lower rates may indicate incomplete annotation, DNA contamination, or quality issues

12. rRNA Rate

- **What it measures:** Percentage of reads mapping to ribosomal RNA sequences
- **What it tells you:**
 - Efficiency of rRNA removal or poly(A) selection during library preparation
 - Higher rRNA rates reduce effective sequencing depth for mRNAs

13. Library Strandedness

- **What it measures:** Whether reads align to the expected strand relative to gene annotation
- **What it tells you:**
 - Confirms strand-specific library preparation worked correctly
 - Strand information is essential for accurate gene quantification, fusion analysis, and isoform identification
 - Different tools use different methods for strandedness evaluation, so interpretation should be tool-specific
 - Unexpected results may indicate genomic DNA contamination or library prep failure

14. Genomic Origin of Reads

- **What it measures:** Proportion of reads mapping to exonic, intronic, and intergenic regions
- **What it tells you:**
 - High exonic: mature mRNA (expected for poly-A selected libraries)
 - High intronic: nascent transcripts, pre-mRNA, or potential genomic DNA contamination
 - High intergenic: potential genomic DNA contamination or alignment issues

15. Number of Detected Genes

- **What it measures:** Number of genes with detectable expression above threshold
- **What it tells you:**
 - Reflects both library complexity and sequencing depth
 - Biological sample characteristics (cell type, tissue complexity) naturally affect gene counts

16. Gene Body Coverage Plot

- **What it measures:** Coverage profile along gene bodies from 5' to 3' end, visualized as a plot
- **What it tells you:**
 - Uniform coverage generally indicates good RNA quality
 - Bias patterns depend on library prep method (some 3' bias is normal for poly-A selection)
 - Severe bias may indicate RNA degradation or library preparation issues

17. 5' to 3' Bias Ratio

- **What it measures:** Numerical ratio summarizing the balance of coverage at 5' end versus 3' end of transcripts
- **What it tells you:**
 - Values close to 1 indicate uniform coverage
 - Values significantly below 1 indicate 3' bias
 - Values significantly above 1 indicate 5' bias

Target Enrichment Metrics (If Applicable)

18. On-Target Rate

- **What it measures:** Percentage of reads mapping to targeted regions
- **What it tells you:**
 - Capture efficiency for targeted enrichment panels
 - On-target rates depend on panel design and capture efficiency