Contents

1 Introduction

RNA sequencing (RNA-seq) has become an indispensable technology in molecular biology, enabling researchers to explore and quantify the transcriptome of an organism under various conditions. It provides critical insights into gene expression dynamics, alternative splicing, and the discovery of novel transcripts. However, the analysis of RNA-seq data involves a multi-step computational process that can be complex, time-consuming, and prone to errors if not managed properly. Ensuring the reproducibility and scalability of these analyzes is a major challenge in modern bioinformatics.

To address these challenges, we have developed the **RNA-seq Analysis Pipeline**, a systematic and automated workflow built on the Nextflow [3] framework following nf-core [14] good practices. With the use of Nextflow, the pipeline adheres to the principles of FAIR [18] by ensuring that workflows are findable, accessible, interoperable and reproducible in diverse computational environments. This pipeline is designed to provide a comprehensive, end-to-end solution for RNA-seq analysis, from raw sequencing reads to quantified gene and transcript expression matrices. By encapsulating a suite of state-of-the-art bioinformatics tools within a portable and scalable environment using Docker, the pipeline guaranties reproducibility and simplifies the execution of complex analyzes.

This report details the pipeline's architecture and the methods employed at each stage. A key feature of this workflow is its flexibility, offering users the choice between two distinct and widely used quantification strategies: a traditional alignment-based approach using Hisat2 [9] or Star [4], and a rapid pseudo-alignment approach with Salmon. This dual-path design makes the pipeline adaptable to a wide range of research questions, from standard differential gene expression analysis to more detailed transcript-level investigations.

2 Methods

2.1 Pipeline Overview

The pipeline is structured as a series of modular processes, where the output of one step serves as the input for the next. The overall workflow is managed by Nextflow, which handles parallelization, resource management, and error reporting. A visual representation of the workflow is shown in Figure 1.

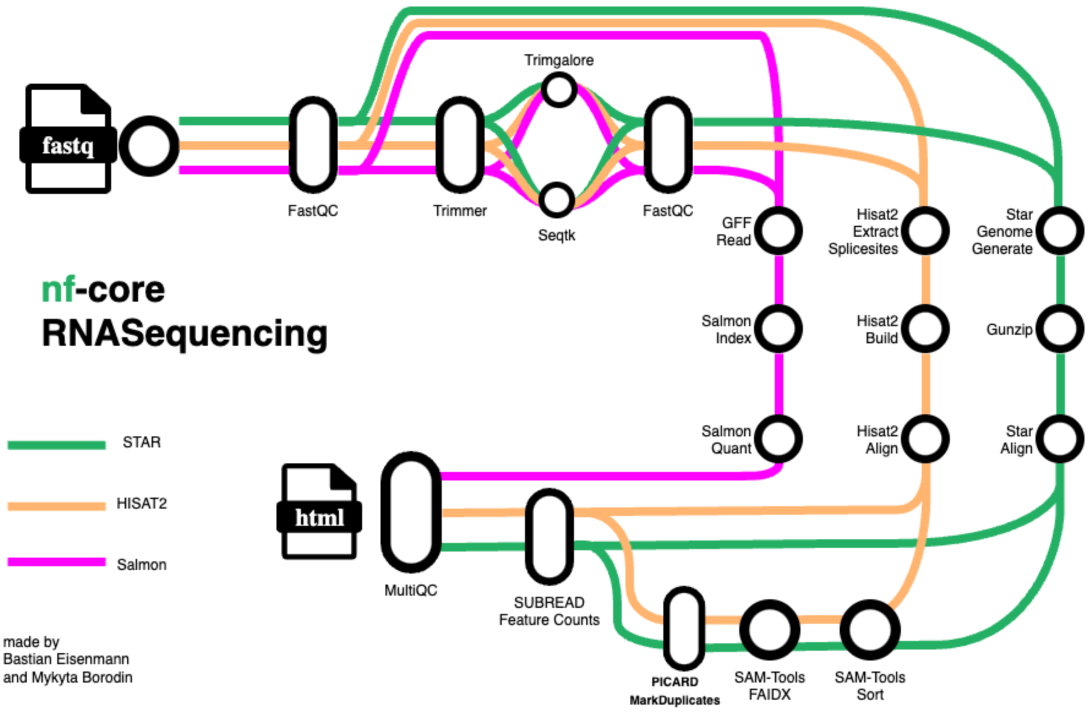

Figure 1: Schematic overview of the RNA-seq analysis pipeline. The workflow begins with raw read processing and then splits into two parallel paths for quantification: an alignment-based path (orange / green) and a pseudo-alignment path (pink). All quality metrics are aggregated at the end.

2.2 Used software

The two core branches of the pipeline use either Salmon for pseudo-alignment and quantification, or a combination of Hisat2 or Star for alignment and Subread [12] FeatureCounts for gene-level counting, with optional duplicate removal using Picard. Additionally, the pipeline incorporates FASTQC [2] and MultiQC [5] for quality assessment, with a choice of TrimGalore [10] or Seqtk [8] for adapter trimming. Other utilities, such as gunzip [7] and samtools [11], are required for pre-processing. A complete overview of all software tools and their versions is provided in Table 1.

Module	**Tool**	**Version**	**Description**
Quality Control	FASTQC [2]	0.12.1	Quality control of raw sequencing reads.
Quality Control	MultiQC [5]	1.31	Aggregates and summarizes results from multiple QC tools.
Adapter-Trimming	cutadapt [13]	4.9	Adapter trimming and quality filtering of reads.
Adapter-Trimming	trimgalore [10]	0.6.10	Wrapper for Cutadapt and FASTQC for trimming and QC.
Adapter-Trimming	seqtk [8]	1.4-r122	Lightweight toolkit for processing FASTA/FASTQ sequences.
Pre-Processing	pigz [1]	2.8	Parallel compression tool for faster gzip operations.
Pre-Processing	gffread [16]	0.12.7	Processes and converts GFF/GTF annotation files.
Pre-Processing	samtools [11]	1.2, 1.22.1, 1.21	Utilities for manipulating alignments in SAM/BAM format.
Pre-Processing	gawk [6]	5.1.0	Text processing utility used in various workflow steps.
Pseudo-Alignment, Quantification	salmon [15]	1.10.3	Pseudo-alignment and transcript quantification from RNA-Seq reads.
Alignment	hisat2 [9]	2.2.1	Aligns RNA-Seq reads to a reference genome.
Alignment	star [4]	2.7.11b	Ultrafast RNA-Seq read aligner suitable for large datasets.
Duplicate Removal	picard [17]	3.4.0	Identifies and marks duplicate reads in alignment files.
Quantification	subread [12]	2.0.6	Generates gene-level read count summaries from alignments.
Pipeline-Management	nf-core [14]/ rnasequencing	v1.0.0dev	Current development version of the nf-core RNA-Seq pipeline.
Pipeline-Management	Nextflow [3]	25.04.2	Workflow management system used for pipeline orchestration.

Table 1: Tools and versions used in the RNA-Seq pipeline.

2.3 Pipeline Input

The pipeline requires a samplesheet in CSV format, which must include the columns sample, fastq_1, fastq_2, strandedness, library, and seq_center. This file can be passed via the *–input* parameter. The sample column serves as the unique identifier for each sample, while the fastq_1 and fastq_2 columns specify the paired-end sequencing reads. The strandedness and library columns provide additional information to the aligner. The seq_center column is currently prepared for future use with the Star aligner but has no effect in the current version. A sample file is provided in the pipeline repository at *./assets/samplesheet.csv.* Using the *–outdir* parameter, all output files can be written directly to a specified directory. The *–genome* parameter is the recommended way to provide a reference genome, which must correspond to the genome used for the input samples. By selecting the *–mode* parameter, the pipeline will either perform pseudo-alignment and quantification with Salmon, or run Hisat2 or Star alignment followed by read counting with Subread FeatureCounts. In the alignment-based mode, duplicate removal with Picard can optionally be enabled. Adapter trimming can also be optionally performed using TrimGalore or Seqtk, and FASTQC can be optionally executed for quality control. A complete overview of all parameters is provided in Table 2.

Parameter	**Default**	**Description**
`--input`	-	Path to the input samplesheet in CSV format.
`--outdir`	-	Directory where all pipeline output files are written.
`--genome`	-	Reference genome identifier (e.g., `sacCer3`, `hg38`).
`--mode`	`hisat2`	RNA-Seq analysis mode; available options are `salmon`, `hisat2`, or `star`.
`--trimmer`	`trimgalore`	Read trimming tool; choose between `none`, `trimgalore` or `seqtk`.
`--mark_duplicates`	`true`	Enables marking of duplicate reads in alignment files. (Only for mode star and hisat2)
`--run_fastqc_at_start`	`true`	Performs quality control (FASTQC) on raw sequencing reads.
`--run_fastqc_after_trim`	`true`	Performs quality control (FASTQC) on trimmed reads.

Table 2: Key parameters used in the RNA-Seq pipeline.

2.4 Pipeline Output

As the primary outputs, the pipeline generates transcript abundance files from Salmon, located at *salmon/*/quant.sf*, or gene-level feature count tables from Subread, located at *subread/*.featureCounts.tsv.* Additionally, the MultiQC report can be found at *multiqc/multiqc_report.html*, and the pipeline execution report is available at *pipeline_info/execution_report.html.* A concise overview of the key output files is provided in Table 3.

File	Description
`multiqc/multiqc_report.html`	Main QC report — provides a comprehensive overview of all quality control results.
`subread/combined_counts.txt`	Gene-level count matrix containing all processed samples.
`salmon/*/quant.sf`	Transcript abundance files per sample generated in the *salmon* quantification mode.
`subread/id.featureCounts.tsv`	Gene-level count output used when running in *hisat2* or *star* alignment mode.
`pipeline_info/execution_report.html`	Summary report of pipeline execution metrics and runtime information.

Table 3: Key output files generated by the RNA-Seq pipeline.

3 Results

3.1 Workflow Management and Reproducibility

The pipeline is written in Nextflow, a workflow management system that enables scalable and reproducible scientific workflows. It uses a series of independent processes for each bioinformatic task. To ensure full reproducibility and portability, all software dependencies are managed through Docker containers, which encapsulates each tool and its required libraries into a self-contained unit. This approach guarantees that the analysis can be run on any system with Nextflow and Docker installed, yielding identical results. The pipeline reads the provided samplesheet and executes all necessary computational tasks for each sample according to the additional run configurations specified as input. A sample run can be performed using the command *nextflow run . -profile docker,test –outdir test-out*, which executes an initial quality control step with FASTQC, followed by Salmon quantification based on four yeast sequencing samples. In this chapter, we provide a more detailed overview of the different components of the pipeline.

3.2 Input Data and Quality Control

The pipeline accepts single-end or paired-end sequencing reads in FASTQ format. A samplesheet in CSV format is used to specify the location of the files and associated metadata for each sample. The initial step in the workflow is a quality assessment of the raw FASTQ files using FASTQC. This tool generates a comprehensive report for each sample, evaluating key metrics such as per-base quality scores (Phred scores), GC content, sequence duplication levels, and the presence of overrepresented sequences or adapter contamination. This initial QC is crucial for identifying potential issues with the sequencing data before proceeding with downstream analysis. It is recommended to keep the initial FASTQC execution enabled. However, if only the post-trimmed reads are of interest, this step can be disabled to save computational resources.

3.3 Adapter Trimming

Based on the initial QC reports, pre-processing is performed to clean the raw reads. The pipeline employs TrimGalore, a wrapper for Cutadapt and FASTQC, to automatically detect and remove adapter sequences from the 3' ends of reads. Additionally, low-quality bases (typically with a Phred score below 20) are trimmed from both ends of the reads. An alternative trimmer, Seqkt, is also available. After trimming, reads that become too short are discarded. Optionally, a second round of FASTQC is performed on the cleaned reads to verify that quality issues have been resolved.

3.4 Quantification Pathways

The pipeline supports two distinct methodologies for quantifying expression. Both pathways can be switched through the use of the *–mode (salmon)/(hisat2/star)* parameter, per default the salmon pathway is used.

3.4.1 Pseudo-Alignment-Based Quantification (Salmon)

Prior to alignment, reference indices must be built for the chosen aligner. The pipeline automatically generates these indices from a user-provided reference genome (FASTA format) and gene annotation file (GTF format). For common genomes, matching reference genome and gene annotations file can be passed by using the *–genome (genome-name)* in a convient way. For the pseudo-alignment path, a transcriptome reference is first generated from the genome and GTF files using GFFREAD, and this is then used to build the Salmon index.

For a faster, alignment-free approach, the pipeline uses Salmon. Instead of performing a computationally intensive base-by-base alignment, Salmon utilizes a quasi-mapping algorithm to rapidly determine the likely transcript of origin for each read. It quantifies the abundance of each transcript, providing outputs in Transcripts Per Million (TPM) and estimated read counts. This method is highly efficient and particularly well-suited for transcript-level analyses.

3.4.2 Alignment-Based Quantification (Hisat2 / Star + Subread)

This traditional approach involves aligning the trimmed reads to the reference genome. The reference genome is also passed as in salmon pathway and requiresequire a tool specific indexing step.

Users can choose between two splice-aware aligners:

- **Hisat2**: A fast and memory-efficient aligner that uses a graph-based indexing scheme to handle spliced alignments of reads spanning multiple exons.
- **Star**: An ultra-fast aligner known for its high accuracy and sensitivity in detecting spliced transcripts.

The resulting alignments are stored in SAM/BAM format. These files are then sorted and indexed using SAMtools. To mitigate biases from PCR amplification during library preparation, duplicate reads can optionally be identified and marked using Picard MarkDuplicates.

Finally, gene-level expression is quantified using Subread FeatureCounts. This tool counts the number of reads that map to the exons of each gene as defined in the GTF annotation file, generating a raw count matrix where rows represent genes and columns represent samples.

3.5 Aggregate Reporting

To provide a comprehensive overview of the entire analysis, the pipeline uses MultiQC. This tool parses the log files and output from all other tools (e.g., FASTQC, TrimGalore, Star, Hisat2, Salmon, featureCounts) across all samples and compiles them into a single, interactive HTML report. This report allows for easy comparison of quality metrics and alignment/quantification statistics across the entire project, facilitating the identification of outliers or batch effects.

4 Discussion

Using Nextflow, the pipeline can be easily shared and collaboratively developed between multiple researchers, enabling reuse of the implemented RNA-sequencing and quantification workflows in their own projects. Since the module versions are pinned and corresponding Docker containers are provided, users can rely on consistent, reproducible results—as long as container support is available on their computing infrastructure.

The pipeline currently supports workflows based on Salmon, Hisat2, or Star. However, the Star workflow lacks support for cases where a read trimmer was used prior. Additionally, parameters specific to the sequencer and sequencing center are currently hard-coded rather than dynamically extracted from the input sample sheet, as would be preferable for flexibility and automation.

At present, the pipeline outputs a gene count table, which already provides valuable quantitative data. In typical RNA-Seq analyses, however, researchers often perform differential expression analysis (DEA) on these count tables to compare gene expression levels across different experimental conditions (e.g., disease state, age, or gender). Incorporating such functionality could be a valuable enhancement for future versions of the pipeline. As further improvement also pre-available genome indexes could be used in the case of Hisat2 and Star alignment to reduce required computation tasks.

References

[1] Mark Adler. pigz: Parallel implementation of gzip. https://github.com/madler/pigz, 2022. Accessed: 2025-10-19.

[2] S. Andrews. Fastqc: A quality control tool for high throughput sequence data. *Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/*, 2010. Accessed: 2025-10-19.

[3] Paolo Di Tommaso, Maria Chatzou, Evan Floden, Pablo P. Barja, Emilio Palumbo, and Cedric Notredame. Nextflow enables reproducible computational workflows. *Nature Biotechnology*, 35:316–319, 2017. doi:10.1038/nbt.3820.

[4] Alexander Dobin, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, January 2013. doi:10.1093/bioinformatics/bts635.

[5] Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19):3047–3048, October 2016. doi:10.1093/bioinformatics/btw354.

[6] Free Software Foundation, Inc. *GNU Awk (gawk): The GNU implementation of the AWK programming language*. Free Software Foundation, 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA, 2023. Copyright © 2008, 2020, 2022, 2023 Free Software Foundation, Inc. URL: https://www.gnu.org/software/gawk/.

[7] Jean-loup Gailly and Mark Adler. *GNU gzip – A file compression utility*. Free Software Foundation, 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA, 2023. Copyright © 2008, 2020, 2022, 2023 Free Software Foundation, Inc. URL: https://www.gnu.org/software/gzip/.

[8] Li Heng. Seqtk: Toolkit for processing sequences in fasta/q format. https://github.com/lh3/seqtk, 2023. Accessed: 2025-10-19.

[9] Daehwan Kim, Jeremiah M. Paggi, Chanhee Park, Christopher Bennett, and Steven L. Salzberg. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*, 37:907–915, 2019. doi:10.1038/s41587-019-0201-4.

[10] Felix Krueger. Trim galore! https://github.com/FelixKrueger/TrimGalore, 2023. Accessed: 2025-10-19.

[11] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nicholas Homer, Gad Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, August 2009. doi:10.1093/bioinformatics/btp352.

[12] Yang Liao, Gordon K. Smyth, and Wei Shi. The subread aligner: fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, 41(10):e108, May 2013. doi:10.1093/nar/gkt214.

[13] Marcel Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17(1):10–12, May 2011. Accessed: 2025-10-19. URL: https://journal.embnet.org/index.php/embnetjournal/article/view/200, doi:10.14806/ej.17.1.200.

[14] Eisenmann Bastian Mykyta Borodin. nf-core/rnasequencing: nf-core rna sequencing pipeline. https://github.com/eisenmann-bastian/rnasequencing, 2025. Accessed: 2025-10-19.

[15] Rob Patro, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, 14:417–419, 2017. doi:10.1038/nmeth.4197.

[16] Geo Pertea and Mihaela Pertea. Gff utilities: Gffread and gffcompare. *F1000Research*, 9:ISCB Comm J–304, April 2020. doi:10.12688/f1000research.23297.2.

[17] Picard Development Team. Broad institute. https://broadinstitute.github.io/picard/, 2015. Accessed: 2025-10-19.

[18] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Barend Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, René van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi:10.1038/sdata.2016.18.