Comparative Analysis of RNA-seq Alignment Softwares and Algorithms

In partial fulfillment of CMSC244

Mark Cyril R. Mercado

December 12, 2025

Contents

1 Introduction

1.1 Background

The flow of genetic information is starts from DNA to RNA and to protein. The DNA serve as the low-level code and molecular repository that gets keep and replicated, DNA then is transcribe into RNA when needed, and RNA is translated to protein to function to various biological processes [Crick, 1970]. One technology to quantify the gene expression is RNA-sequencing technology that measures RNA abundance to estimate gene or transcript expression and then the expression patterns across tissues, organs, or developmental stages is compared.

In a typical condition, not all genes are expressed uniformly: expression is context-dependent and specific to tissue/organ. But one major processes that add complexity is the presence of alternative (differential) splicing, where a same gene can produce multiple transcript isoforms by including or excluding specific exons (coding sequences of the gene) while introns (the noncoding sequences) are removed during RNA processing. As a classical example, the Dscam locus of *Drosophila melogenogaster*, common fly, can generate tens of thousands of isoforms through alternative splicing, illustrating how isoform diversity can be large even for a single gene [Wang et al., 2012]. Because splicing alters which sequences appear in the mature RNA, splice-aware alignment and transcript-level quantification strategies can produce different expression estimates for the same dataset.

1.2 Research Problems and Challenges

In the computational side, handling of large file sizes output of transcriptome-scale RNA-seq data, optimizing CPU/thread utilization, and ensuring accuracy and sensitivity of expression profiles while accounting for differential splicing posed as a challenge to post-processing of the data. A good test research questions for this context is the identification of *SmelDMP* gene(s) which are expected to be differentially expressed in floral buds providing a single target organ and determining which *SmelGRFs* and *SmelGIFs* genes exhibit characteristically high expression in young developing organs, providing multiple target organs.

1.3 RNA-seq Data Analysis Workflow and Alignment/Quantification Programs

A typical RNA-seq workflow starts with raw read-level preprocessing (quality control and trimming), followed by read mapping and expression quantification. At the mapping/quantification stage, the central operation in the pipeline, tool choice matters: different alignment and quantification programs can produce meaningfully different expression estimates from the same reads, which can affect downstream biological interpretation [Fonseca et al., 2014]. Moreover, systematic assessments have reported that there is no single universally "best" pipeline across datasets and goals [Corchete et al., 2020].

Generally, there are three ways or approach to quantify gene expression:

(i) Alignment against a genome (e.g., HISAT2),

(ii) Alignment/mapping against a transcriptome reference (e.g., Bowtie2), and

(iii) Pseudoalignment approaches for rapid transcript quantification (e.g., Salmon SAF).

HISAT2 is software that can performs splice-aware genome alignment tailored to the complex RNA-seq data, enabling accurate detection of exon–intron junctions and alternative splicing events [Kim et al., 2014, Kim et al., 2019]. Bowtie2 functions as a fast, sensitive, and flexible gapped aligner that is widely incorporated into RNA-seq processing pipelines to provide reliable and efficient read mapping across diverse genomic contexts [Langmead & Salzberg, 2012]. Salmon, in contrast, is designed for rapid and statistically rigorous transcript abundance estimation, leveraging the lightweight mapping and selective alignment strategies [Patro et al., 2015].

1.4 Considerations

For large-scale RNA-seq studies, runtime and memory complexity are critical determinants of computational efficiency, particularly when working with large reference genomes or transcriptome, numerous samples, or deep sequencing coverage. Performance is also sensitive to the degree of threading or parallelization. Because tissue-specific ground-truth expression data for the selected gene families and species are scarce, evaluation of correctness relies on concordance with established biological expectations, replication, and on cross-referencing with other method for consistency in the resulting expression profiles.

2 Methodology

2.1 RNA-seq Pipeline Overview

As shown in Figure 1, the RNA-seq pipeline generally comprises of preprocessing (quality control and trimming, with a strategy of compressing trimmed SRR files), main operation (alignment as the central step, with a strategy of deleting transient files such as SAM/BAM after extracting expression data to reduced space), and post-processing (quantification followed by statistical normalization).

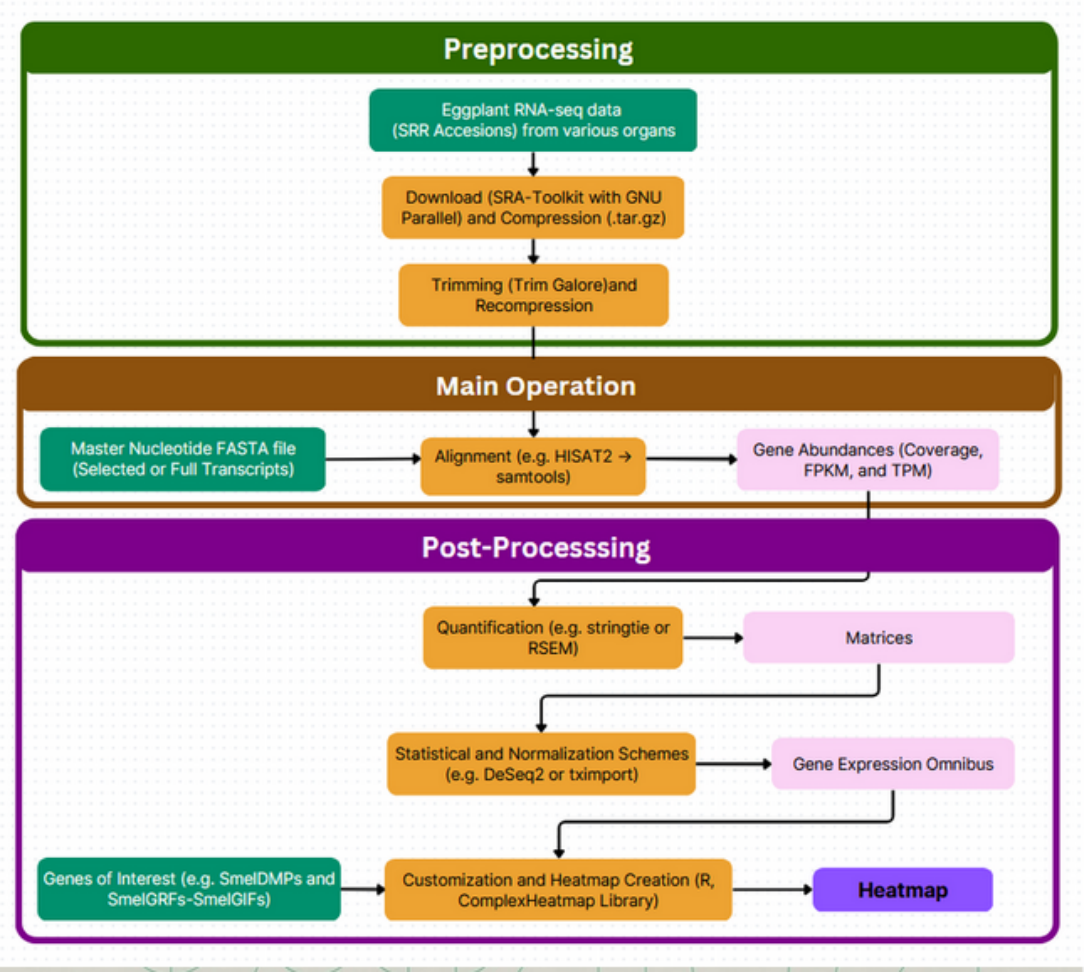

Figure 1: General Overview of the Gene Expression Analysis Pipeline.

2.2 Running of the Alignment Programs and Reimplemented Algorithms per Alignment Programs

Papers, lectures, and repositories were used as guide for the Python reimplementations of the algorithms, with each approaches decomposed into steps to analyze behavior and complexity.

The reimplemented alignment algorithms were saved in:

`CMSC244_Aln_Algo_for_RNA-seq/test_modules/Aln_Algorithm_Functions/`

For HISAT, the hisat2-build software and step focuses on FM-index pattern finding (reference code: gfm.cpp), while the hisat software performs seed-and-extend with mismatches (reference code used: aligner_seed.cpp) and splice-aware alignment (reference code used: hi_aligner.h);

Reimplementation Filepath: `hisat_alignment.py`,

Reference Repository Path: `Actual_Softwares/hisat2`.

For Bowtie, rsem-prepare-reference sofware performs the FM-index pattern finding (ref: aligner_seed.cpp), and rsem-calculate-expression with Bowtie2 software aligns using Needleman–Wunsch and seed extraction (ref: aligner_sw.cpp);

Reimplement path: `bowtie_alignment.py`,

Ref_Repo_Path: `Actual_Softwares/bowtie2`.

For Salmon_SAF, salmon-index software builds the index and salmon-quant software performs quasi-mapping/selective alignment (ref: SalmonQuantify.cpp) with EM for quantification (ref: CollapsedEMOptimizer.cpp);

Reimplementation filepath: `salmon_saf_alignment.py`.

Ref_Repo_Path: `Actual_Softwares/salmon`.

3 Results

3.1 Actual Runtime and Memory Comparison

3.1.1 Actual Alignment Programs

Runtime Complexity Average user time (actual runtime) of HISAT2, Bowtie2, and Salmon SAF at increasing CPU cores.

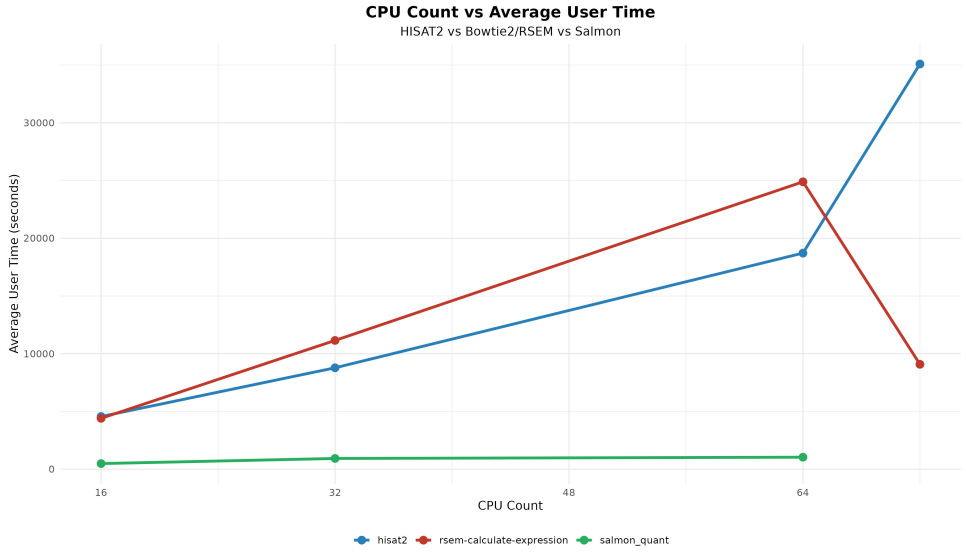

Figure 2: CPU count vs average user time for HISAT2, Bowtie2, and Salmon SAF.

In Figure 2, Salmon (green) yields the lowest user time, HISAT2 (blue) is moderate, and Bowtie2-RSEM (red) is highest. The advantage of Salmon's quasi-mapping increases with core count, while threading gains across programs taper beyond ∼32 to 64 cores.

Space Complexity Memory usage comparison of the Alignment Programs at 32 CPU cores.

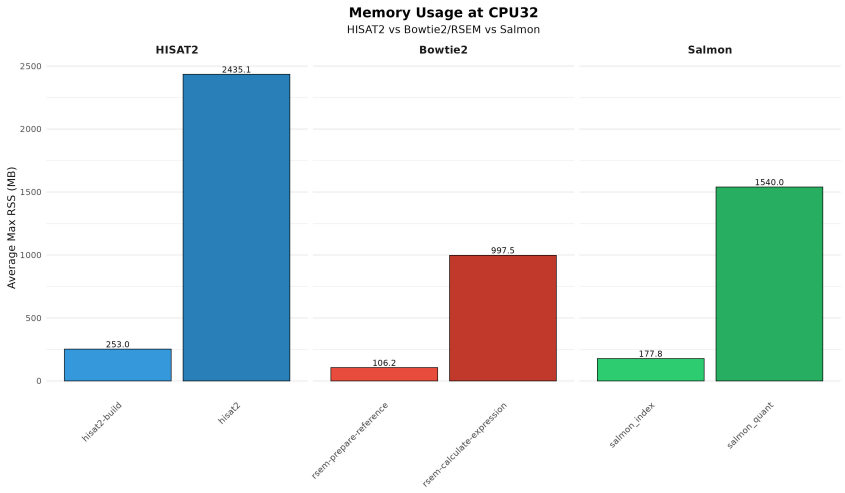

Figure 3: Average max RSS (memory usage) of HISAT2, Bowtie2, and Salmon SAF at 32 CPU cores.

HISAT2 has the highest memory usage (∼24MB); Bowtie2 and Salmon are ∼9–15MB (Figure 3).

3.1.2 Recreated Algorithms

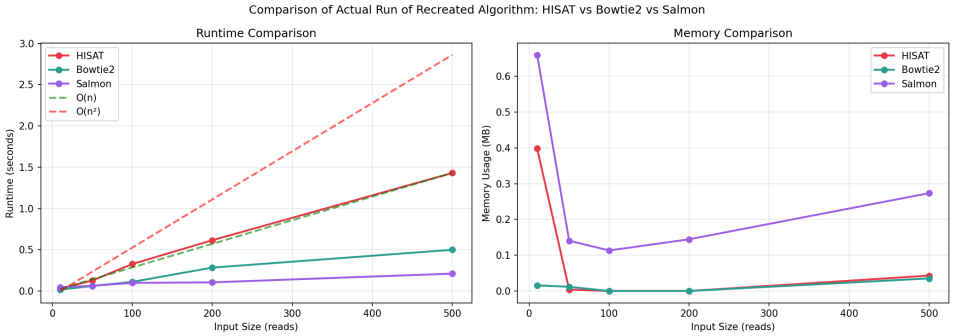

Figure 4: Runtime and memory comparison of recreated HISAT2, Bowtie2, and Salmon algorithms.

Runtime Complexity HISAT scales near-linearly ($O(r)$ dominant), Bowtie increases more steeply ($O(c \cdot r \cdot w)$ due to extensions), and Salmon is flattest ($O(m \cdot h)$ quasi-mapping). At 500 reads: Bowtie ∼0.5s, HISAT ∼1.4s, Salmon ∼0.2s, consistent with the theoretical (Figure 4).

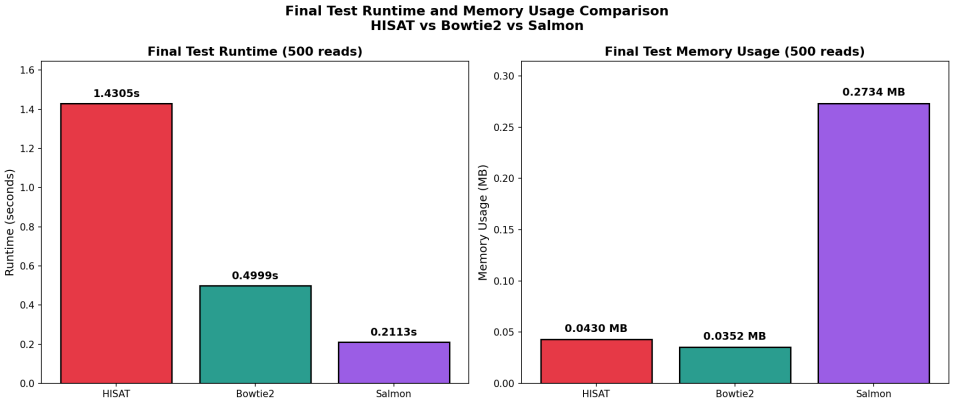

Figure 5: Performance benchmarks of recreated alignment algorithms.

Space Complexity Figure 5 summarizes performance benchmarks at final test on 500 reads. For runtime, HISAT is slowest (∼1.4s) due to its additional processing steps, followed by Bowtie (∼0.5s), while Salmon is fastest (∼0.2s) at 500 reads. For memory, all remain <1MB on the small test reference: HISAT and Bowtie use 0.04MB and 0.03MB, respectively, whereas Salmon exhibits the highest memory usage due to direct transcript-level alignment and EM quantification overhead.

3.2 Analysis of the Algorithm

HISAT2 Python Reimplementation: `hisat_alignment.py` Overall Algorithm Analysis:

- Indexing $O(n \log n)$
- Steps or Tiers:
  - Per-Read $O(r + k)$ exact – FM-index search scales with read length plus matches found
  - $O(s \cdot h \cdot r)$ approx – seeds × hits per seed × mismatch counting across read
  - $O(r \cdot L \cdot R)$ spliced – split positions × left anchor hits × right anchor hits

Bowtie2-RSEM Python Reimplementation: `bowtie_alignment.py`
Overall Algorithm Analysis:

- Indexing $O(n \log n)$ – suffix array construction is the key determinant for runtime complexity; dominates index building; presence of the sorting function
- Per-Read $O(s \cdot m + c \cdot r \cdot w)$ – seed lookups + Smith-Waterman extensions per candidate
- Space $O(n + r \cdot w)$ – FM-index storage plus DP matrices for extension

Salmon SAF Python Reimplementation: `salmon_saf_alignment.py`
Overall Algorithm Analysis:

- Index $O(T \cdot L)$ – minimizer extraction (transcripts × length)
- Mapping $O(R \cdot m \cdot h)$ – minimizer lookups (reads × minimizers × hits)
- Quant $O(I \cdot R \cdot A)$ – EM iterations over reads and multi-mappings
- Space $O(M + T + R \cdot A)$ – index plus abundances and assignments

HISAT2 and Bowtie2 share $O(n \log n)$ genome indexing but differ in per-read strategy: HISAT's tiered exact→approximate→spliced flow reduces expensive extensions, whereas Bowtie2's uniform seed-and-extend increases per-candidate cost via Smith–Waterman. Salmon avoids genome alignment with $O(T \cdot L)$ transcript indexing and $O(m \cdot h)$ quasi-mapping, trading precise genomic positions for speed. Space is dominated by $O(n)$ indexing (FM-based methods), with Salmon adding $O(R \cdot A)$ state for EM.

3.2.1 Complexity Ranking

Table 1: Runtime and space complexity ranking of alignment algorithms

Complexity Metric	Algorithm	Complexity
Runtime (per read)	Salmon (fastest)	$O(m \cdot h)$
	HISAT2 (moderate)	$O(r + k)$ to $O(r \cdot L \cdot R)$
	Bowtie2 (slowest)	$O(c \cdot r \cdot w)$
Space (overall)	Bowtie2 (lowest)	$O(n)$ (index-dominant)
	Salmon (moderate)	$O(M + T + R \cdot A)$
	HISAT2 (highest)	$O(n \log n)$ indexing

Summary: Salmon achieves the lowest runtime via lightweight quasi-mapping ($O(m \cdot h)$), relying on minimizer lookups rather than full alignment extensions. HISAT2's tiered approach (exact, approximate, spliced) yields moderate runtime by deferring expensive operations to later tiers. Bowtie2's uniform seed-and-extend strategy incurs the highest per-read cost due to frequent Smith–Waterman extensions ($O(c \cdot r \cdot w)$). For space, FM-index footprints dominate; Bowtie2 exhibits the tightest indexing, whereas HISAT2's hierarchical indexing and associated metadata increase memory overhead, consistent with the observed benchmarks (Figure 3).

3.3 Correctness and Validation

3.3.1 In the case of *SmelDMPs*

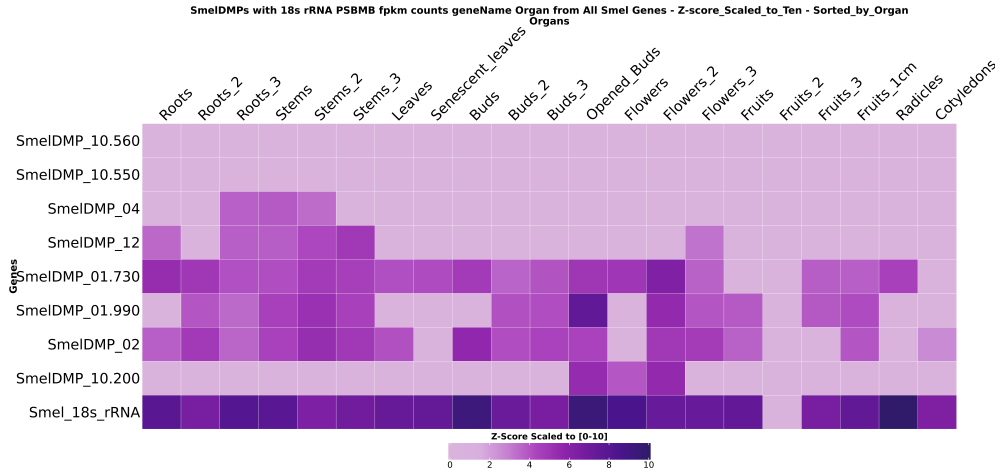

Figure 6: *SmelDMPs* heatmap (HISAT2, gene-level FPKM).

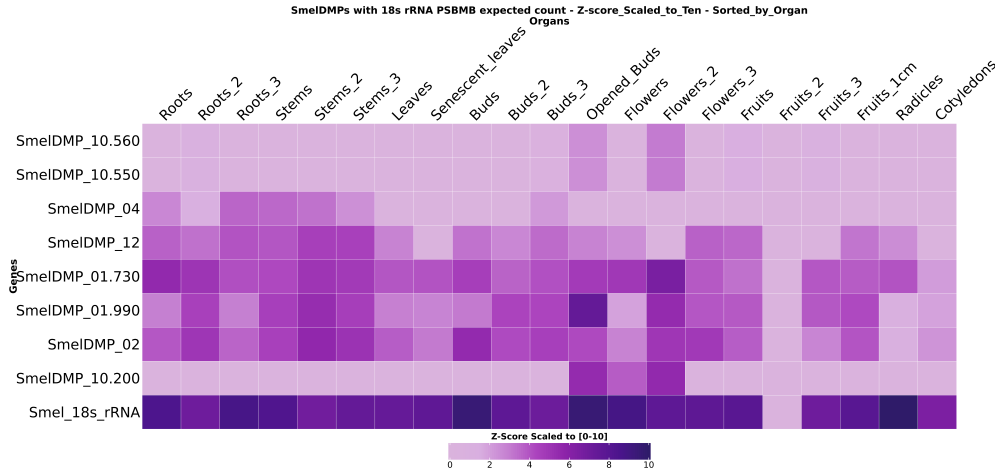

Figure 7: *SmelDMPs* heatmap (Bowtie2-RSEM, gene-level expected counts).

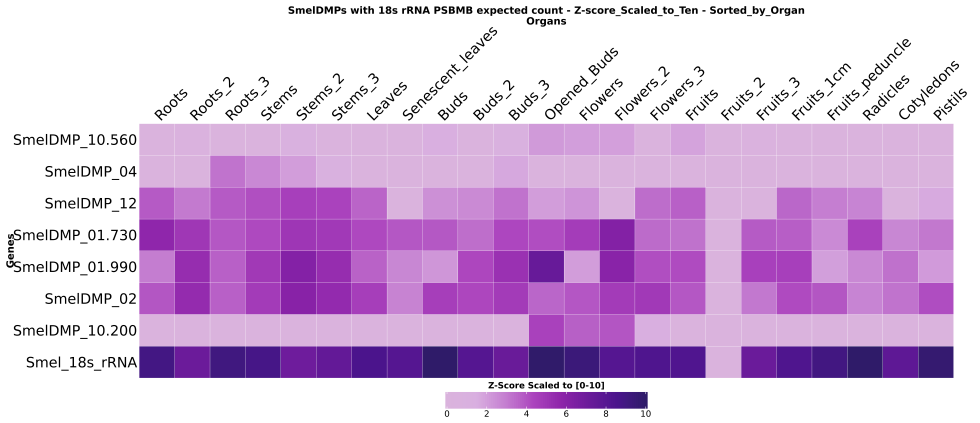

Figure 8: *SmelDMPs* heatmap (Salmon SAF, isoform-level expected counts).

Expectation and Results: *SmelDMP10.200* has the highest expression in the floral buds across methods (Figures 6–8), matching expectations from orthology in the phylogenetic tree against well-studied DMP from other crop species also expressed in floral buds. Profiles across other organs are broadly concordant. HISAT2 is most specific; Bowtie2 and Salmon show additional low-level signal, reflecting sensitivity and read assignment differences.

3.3.2 In the case of *SmelGRFs* and *SmelGIFs*

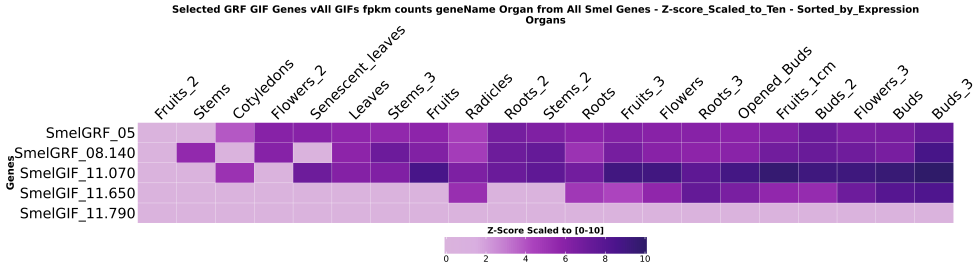

Figure 9: Selected GRF/GIF heatmap (HISAT2, gene-level FPKM).

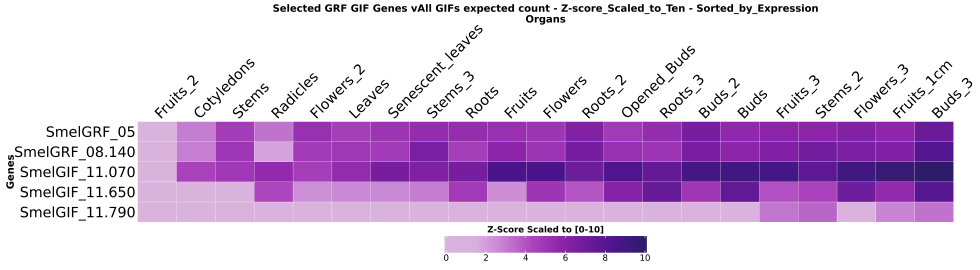

Figure 10: Selected GRF/GIF heatmap (Bowtie2-RSEM, gene-level expected counts).

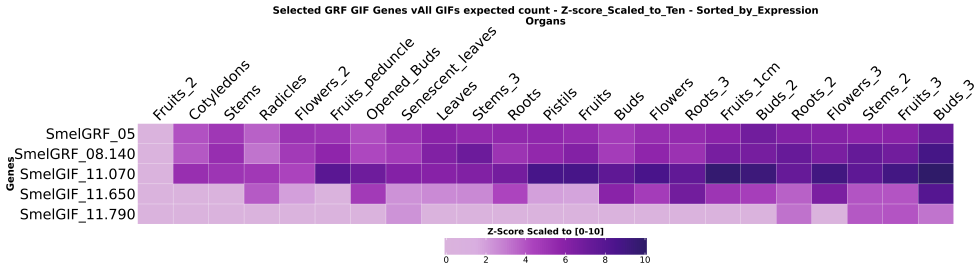

Figure 11: Selected GRF/GIF heatmap (Salmon SAF, isoform-level expected counts).

Expectation and Results: *SmelGRFs* and *SmelGIFs* peak in buds and remain strong in young/meristematic organs across methods (Figures 9–11). Differences are mainly quantitative (ranking and background), driven by sensitivity and treatment of multi-mapping reads.

4 Discussions

4.1 Pipeline Design Considerations

Differential Splicing of Query Sequences Bowtie2 and Salmon Saf are not designed to discover novel splice junctions or explicitly model spliced alignments to the genome. Only HISAT2 is designed to be splice-aware for genome alignment and to support detection of spliced reads (including junction-spanning reads) as shown in the sharply and specific expression results detected in the heatmap. Salmon SAF selectively maps/assigns reads to transcript sequences for quantification, so it can reflect annotated isoform usage but does not perform genome-level splice-junction discovery.

Alignment Sensitivity For basic characterization of expression of genes through heatmap, small sensitivity differences are often negligible in effect at the qualitative pattern level (see Figures 6–11).

But for determination of coexpression that relies on statistical correlation (especially across many tissues/organs) and for differential expression testing, sensitivity and consistent handling of multi-mapping/ambiguous reads become important.

For Bowtie2, a commonly used high-sensitivity setting for RNA-seq-like short reads is `-very-sensitive-local` (with appropriate paired-end and insert-size constraints if applicable).

4.2 Algorithmic Considerations

Actual Runtime and Memory Comparison of the Actual Program and Recreated Algorithm The recreated algorithm preserve the same performance ranking (Salmon $<$ HISAT2 $<$ Bowtie2) seen in the actual softwares. However, the actual tools are 1000–10000$\times$ faster due to optimized C++ implementations, mature FM-index code, and effective multi-threading, whereas the recreated versions are simply proof-of-concepts written in python. Memory usage differs as well: actual programs show higher baselines from indexing and threading overhead, while the recreated algorithms remain $<$1MB on the

small test reference. Bowtie support $O(n)$ space dominance, with Salmon's quasi-mapping outperforming alignment-based approaches (Figures 2–5). Threading gains taper beyond ∼32 cores as I/O and synchronization overhead dominate (Figure 2)

Actual Runtime and Algorithm Analysis The observed runtime ranking (Salmon < HISAT2 < Bowtie2) matches the theoretical per-read complexities in Table 1: Salmon's $O(m \cdot h)$ quasi-mapping is cheapest, HISAT2's tiered $O(r + k)$-to-$O(r \cdot L \cdot R)$ approach is moderate, and Bowtie2's extension-heavy $O(c \cdot r \cdot w)$ is most expensive. Scaling trends in the recreated benchmarks reinforce this ordering—Salmon's runtime curve is flattest, HISAT2 near-linear, and Bowtie2 steepest (Figure 4). Memory usage likewise reflects predictions: HISAT2's $O(n \log n)$ hierarchical indexing yields the highest footprint (∼24MB), while Bowtie2's $O(n)$-dominant FM-index and Salmon's quasi-mapping index remain lower (∼9–15MB), consistent with Figure 3. Thus, the strong concordance between algorithm analysis and actual benchmarks validates the algorithm and supports validity of the result for tool selection.

Correctness and Validation Validation against expected tissue expression supports overall correctness: *SmelDMP10.200* is strongly expressed in floral buds across HISAT2, Bowtie2, and Salmon, and *SmelGRFs/SmelGIFs* show peak expression in buds with consistently strong signal in young/meristematic organs. Differences are mainly quantitative (relative ranking and low-level/background signal). HISAT2 produces the most specific profiles, while Bowtie2/Salmon show additional low-level expression consistent with higher sensitivity and differences in read assignment (Figures 6–11).

5 Conclusion

The actual and recreated results consistently show a performance ranking of Salmon < HISAT2 < Bowtie2 in terms of runtime, driven by quasi-mapping's approach (Figures 2, 4) and alignment-tier differences. Memory profiles are modest and dominated by index footprints (Figures 3, 5). Algorithmically, HISAT2's splice-aware, tiered strategy balances specificity with runtime; Bowtie2's seed-and-extend yields strong sensitivity at higher computational cost; Salmon's transcript-centric selective alignment maximizes throughput with acceptable ambiguity handled via its EM.

For studies emphasizing junction discovery or genome-level splicing, HISAT2 is recommended. For fast, annotation-driven quantification across many samples, Salmon is the most efficient. Bowtie2-RSEM remains a viable option when sensitive local alignment is needed, for example in RNA-seq, accepting higher runtime. For biological validation, there is degree of concordance across *SmelDMPs* and *SmelGRFs/GIFs*: expected peak expression in buds and meristematic tissues, respectively, is reproduced across methods, with minimal differences (Figures 6–11).

Selection of an alignment or quantification tool should be guided by the analytical objective: HISAT2 is preferable when splice resolution and mapping specificity are priorities; Salmon is advantageous when rapid processing and accurate isoform-level abundance estimates are required; and Bowtie2 remains useful when sensitivity to sequence variation is central to the analysis. Multi-mapping artifacts can be mitigated through rigorous normalization strategies. Threading beyond roughly 32 threads often provides limited additional speed, so resource allocation should be planned with these constraints

in mind (Figure 2). Overall, pipeline design should reflect the tissue or organ specificity of the study while maintaining an appropriate balance among sensitivity, specificity, and computational efficiency.

References

[Corchete et al., 2020] Corchete, L. A., Rojas, E. A., Alonso-López, D., De Las Rivas, J., Gutiérrez, N. C., & Burguillo, F. J. (2020). Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis. *Scientific Reports*, 10(1), 1–15.

[Crick, 1970] Crick, F. (1970). Central Dogma of Molecular Biology. *Nature*, 227(5258), 561–563.

[Fonseca et al., 2014] Fonseca, N. A., Marioni, J., & Brazma, A. (2014). RNA-Seq gene profiling - A systematic empirical comparison. *PLoS ONE*, 9(9).

[Kim et al., 2014] Kim, D., Langmead, B., & Salzberg, S. (2014). HISAT: Hierarchical Indexing for Spliced Alignment of Transcripts. *BioRxiv*, 012591.

[Kim et al., 2019] Kim, D., Paggi, J. M., Park, C., Bennett, C., & Salzberg, S. L. (2019). Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nature Biotechnology*, 37(August).

[Langmead & Salzberg, 2012] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*.

[Patro et al., 2015] Patro, R., Duggal, G., & Kingsford, C. (2015). Salmon: Accurate, Versatile and Ultrafast Quantification from RNA-seq Data using Lightweight-Alignment. *BioRxiv*, 021592.

[Wang et al., 2012] Wang, X., Li, G., Yang, Y., Wang, W., Zhang, W., Pan, H., Zhang, P., Yue, Y., Lin, H., Liu, B., Bi, J., Shi, F., Mao, J., Meng, Y., Zhan, L., & Jin, Y. (2012). An RNA architectural locus control region involved in Dscam mutually exclusive splicing. *Nature Communications*, 3.

Repository References

The following repositories and software sources were used as guides for algorithm reimplementation and analysis:

Source Repositories

[9] **HISAT2**: `https://github.com/DaehwanKimLab/hisat2`

[10] **Bowtie2**: `https://github.com/BenLangmead/bowtie2`

[11] **Salmon**: `https://github.com/COMBINE-lab/salmon`

Local Project Directories

[12] Repository copies: `CMSC244_Aln_Algo_for_RNA-seq/test_modules/Actual_Softwares/`

[13] Reimplemented algorithms: `CMSC244_Aln_Algo_for_RNA-seq/test_modules/Aln_Algorithm_F`

Code Repository and Supplementaries

[14] Code Repository and Supplementaries of this Report: `https://github.com/markcyril28/CMSC244_Aln_Algo_for_RNA-seq.git`