AOI: 10.100.234512.00026
ACT2024: Special Proceedings of Applied Computer Technology.

A Survey On Various Computational Tools And Steps In RNA-Seq

Prayakhi Emee Dutta, Nihar Jyoti Boro, Dr. Rosy Sarmah

Department of Computer Science and Engineering
Tezpur University, Assam – 784028, INDIA

ABSTRACT
RNA-Seq is a technique that uses Next-Generation Sequencing technology to comprehensively profile gene expression patterns. This survey provides a comprehensive overview of the computational pipelines and tools used in RNA-seq from preprocessing to differential expression till pathway enrichment. The emerging role of Machine learning and Deep learning in RNA-seq is highlighted offering new perspectives to feature extraction, predictive modeling and multi-omics integration. With the increasing scale and complexity of transcriptomic data, the need for efficient, accurate and interpretable analytical framework has become essential. This review addresses the need by mapping the computational landscape from foundational preprocessing tools to advanced ML/DL methods serving as a practical guide for researchers aiming to derive meaningful insights from RNA-seq studies across diverse biological contexts.

Keywords: RNA seq, Differential Gene Expression, Machine Learning, Deep Learning

I. INTRODUCTION

The emergence of high-throughput sequencing technologies has transformed molecular biology and genomics. In particular, RNA sequencing (RNA-seq) has become a vital tool for studying gene expression, offering comprehensive and highly accurate transcriptome analysis [1]. Unlike conventional microarray methods that depend on predesigned probes, RNA-seq allows for an unbiased exploration of novel transcripts, precise expression quantification, and in-depth examination of alternative splicing patterns. RNA serves as a vital intermediary between DNA and proteins, playing a key role in regulating cellular activities and biological processes. The ability to sequence RNA molecules has greatly expanded our understanding of gene expression patterns in both health and disease. RNA-Seq, a method categorized under Next-Generation Sequencing (NGS) technologies, offers significant improvements over earlier approaches like Sanger sequencing and hybridization-based microarrays. It provides enhanced sensitivity, a wider dynamic range, and the capability to detect novel, previously unannotated transcripts.

Different variants of RNA-Seq have been developed to cater to specific research needs like Bulk RNA-Seq, scRNA and Targeted RNA-Seq [2]. Bulk RNA-Seq provides an overall view of gene expression across tissues or cell populations whereas scRNA-Seq looks at the expression patterns at the individual cell level revealing cellular heterogeneity and rare cell types. In this paper, we will focus on RNA-seq analysis. RNA-seq analysis integrates both experimental and computational workflows to extract biologically meaningful information. A crucial computational step here is differential gene expression (DGE) , aiming to identify genes with significant expression changes between experimental conditions. Statistical tools such as DESeq2, edgeR, etc. [3] are widely used to model the discrete and overdispersed nature of RNA-Seq data. Beyond traditional statistical frameworks, the integration of machine learning(ML) and Deep learning(DL) methods into RNA-Seq analysis [4][5] has opened new ways for classification, feature selection, dimensionality reduction and multi-omics integration, enabling deeper exploration of complex biological systems.
In this survey, we comprehensively review the computational steps and tools involved in RNA-seq analysis, highlighting both foundational techniques and emerging trends that advance the robustness and interpretability of transcriptomic studies.

II. BACKGROUND

Sequencing of DNA fragments was discovered in the 20th century, leading to widespread changes and introduction of new technologies in the field of bioinformatics and computational biology. RNA which is synthesized from DNA became a very important component when it was discovered that RNA is the messenger between DNA and protein synthesis. This led to research in trying to discover new RNA transcripts and measuring Gene Expression levels [6]. These two studies became the backbone of Molecular Biology.
Among high-throughput sequencing methods, RNA-Seq stands out as a widely used approach in various computational analyses related to genome and transcriptome research. RNA-Seq is part of the broader category known as Next-Generation Sequencing (NGS), which encompasses a range of techniques designed to sequence millions of DNA or RNA fragments

simultaneously within a short timeframe [7]. NGS is the direct successor of Microarrays, where expression of gene levels are studied using the concept of hybridisation and fluorescent illumination. Microarrays use chips embedded with thousands of predefined gene probes but do not involve actual sequencing [8]. The origins of sequencing techniques trace back to Sanger sequencing, developed by Frederick Sanger in 1977. Often referred to as the "father" of sequencing, the Sanger method provided a foundation for modern sequencing technologies, offering high accuracy and long read lengths. However, it was limited to determining the nucleotide order of specific gene sequences without offering insights into gene expression levels [6]. RNA-Seq consists of many different types based on the required study of genes. The most common type is 'Bulk RNA-Seq' which is also just called RNA-seq in general. It involves sequencing of gene fragments derived from tissue or cell population which provides an overall picture of gene expression. Other techniques are specializations like Single Cell RNA-Seq (scRNA), Targeted RNA-Seq, Small RNA-Seq and other methods [2]. RNA-Seq can be used in various types of gene and transcriptome analysis including both statistical and predictive approaches. Many different statistical and machine learning tools are applied for a complete pipeline study including both statistical and predictive analysis of gene information. The workflow pipeline of RNA-Seq as illustrated in Fig.1 includes many tools and techniques [6]. In the complete workflow, the computational pipeline is a separate step after retrieving gene expression information from the sequences. As can be seen in Fig. 1, the very first step in the workflow involves sample collection from tissues or cells. RNA is extracted from the sample and synthesized to cDNA. After cDNA synthesis library preparation is done. After a decent library is prepared, the workflow moves into the computational pipeline. The information collected from the library is stored digitally thanks to modern sequencing platforms. One such widely used platform is Illumina Sequencer. The digital information collected from the library is stored in raw files typically 'FASTQ' files. These files contain the raw information of the gene library. These raw files are then read, quality controlled, trimmed, aligned, quantified, normalized and various analyses are performed as depicted in Fig.2. The computational pipeline involves the use of many tools including statistical and raw file processing tools. These steps can be done individually in traditional computers or everything can be processed in sequencing platforms itself when a huge number of genes are concerned, typically in numbers of millions and needed to be processed in a short period of time. One of the important analyses that are almost always performed before any type of biological interpretation is done is Differential Expression (DE) Analysis of the sequenced genes. In differential expression, the gene expressions among different samples or conditions are analysed and how significantly different the expression of a gene is among the conditions is analysed. This is done through tools like DESeq2 and edgeR. After finding differentially expressed genes, functional interpretation is done to gain biological inference.

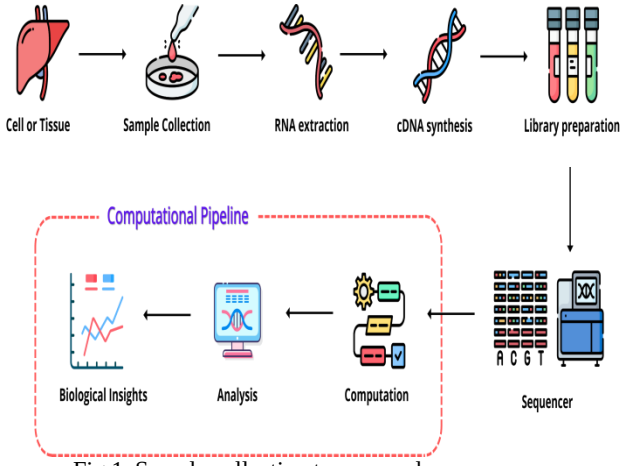

Fig.1. Sample collection to raw reads

After performing statistical and biological interpretation of the genes, one can go for machine learning implementation depending on the type of work or study involved. Appropriate machine learning tools are used as required. Machine Learning role is becoming an important end to end implementation step in RNA-Seq. Both traditional and advanced ML approaches can be used including supervised, unsupervised and deep learning.

In the upcoming sections we talk in detail about the computational steps involved in RNA-Seq.

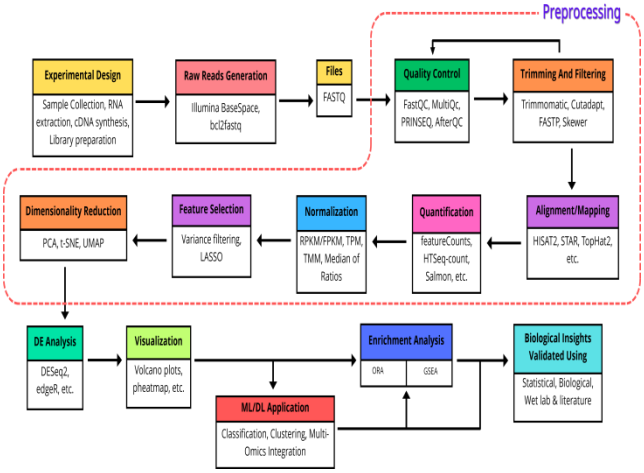

Fig 2. A computational analysis roadmap of RNA-seq data

III. PREPROCESSING

Preprocessing is a critical phase in RNA sequencing (RNA-Seq) data analysis, ensuring that subsequent

analyses are based on clean, normalized, and biologically meaningful input. The preprocessing pipeline typically includes five major steps: quality control, read trimming and filtering, normalization, feature selection, and dimensionality reduction [6]. Each step plays a vital role in minimizing technical biases, reducing data complexity, and preserving the underlying biological signal.

Quality Control: The first step in preprocessing is quality control [9] involves assessing the quality of raw sequencing reads to identify potential issues that could compromise downstream analyses. Tools like FastQC assess key metrics such as per-base quality, Guanine-cytosine(GC) content, sequence duplication and adapter contamination [6].This step helps detect and remove low-quality reads and artifacts early improving data reliability. Recent advances incorporate ML for automated quality checks [4], enabling efficient and objective identification of problematic samples.

Trimming and filtering: After quality control preprocessing involves trimming unwanted sequences and filtering low-quality reads using tools like Trimmomatic, Cutadapt, fastp, etc. Trimming removes adapters and low-quality bases while filtering excludes reads below a specific quality score or minimum length thresholds. This improves alignment accuracy and minimizes technical artifacts in gene expression analysis [7].

Normalization: Normalization [6] is a crucial preprocessing step in RNA-seq that corrects for systematic variations such as sequencing depth, library size and RNA composition. Without it, technical biases may distort biological interpretation. Common methods include Transcripts Per Million(TPM) [6], which accounts for gene length and sequencing depth, Reads Per Kilobase per Million mapped reads(RPKM)/ Fragments Per Kilobase per Million mapper reads(FPKM) [10], used for within sample comparisons and Trimmed Mean of M-values(TMM) [10], employed in edgeR to adjust for compositional differences. For scRNA-Seq, approaches like size factor normalization and scran's deconvolution are used to handle cell-specific biases. Overall, normalization ensures that downstream analysis reflects true biological variation.

Feature Selection: Feature selection reduces RNA-Seq data dimensionality by identifying informative genes relevant to the biological context, improving model performance, interpretability and computational efficiency. Common methods include: (i) Variance filtering to retain high variable genes (ii) univariate statistical tests like t-tests or ANOVA and (iii) machine learning techniques such as random forest, LASSO and mutual information gain [11]. This step is especially critical for ML applications to prevent overfitting and enhance generalizability.

Dimensionality reduction: Even after feature selection, RNA-Seq data often remain high-dimensional. Dimensionality reduction techniques [12] help project this data into lower-dimensional spaces for better visualization, clustering and interpretation. Common methods include Principal Component Analysis (PCA) which captures maximum variance linearly [13], t-distributed Stochastic Neighbor Embedding (t-SNE), which preserves local structure [13] and UMAP which maintains both local and global relationships and scales well to large datasets [13]. Deep learning approaches like autoencoders and Variational autoencoders (VAE) are also used to learn compact representations, particularly in scRNA [5]. These techniques enable more effective downstream analysis such as classification and trajectory inference.

The mathematical foundation and key formulas associated with each preprocessing step is provided in Table I.

TABLE I: PREPROCESSING STEPS AND THEIR ASSOCIATED MATHEMATICAL FOUNDATIONS

Step	Mathematical Basis	Formula
Quality Control	Phred score calculation, GC content estimation	$Q = -10log_{10}(p)$ $GC = \frac{G+C}{A+T+G+C}$
Trimming and filtering	Thresholding rules, sliding window averages	$\underline{Q_W} = \frac{1}{k}\sum Q_i$, discard if $\underline{Q_W} < Q_{min}$
Normalization	Scaling for sequencing depth and gene length	TPM, RPKM, TMM equations
Feature selection	Variance calculation, hypothesis testing, regularized regression	$Var(X_i)$, LASSO objective
Dimensionality reduction	Projection onto lower dimensions preserving variance or topology	PCA, t-SNE, UMAP, Autoencoder losses

(where, Q is Phred score, p is probability, GC is guanine cytosine, $\underline{Q_W}$ is Phred quality score, $Q_{min}$is Threshold, k is size of sliding window, Q_i is Phred quality score of the i-th base in the sliding window, $Var(X_i)$ is Variance of gene iii, measuring how much expression fluctuates).

IV. TOOLS IN RNA-Seq

In the previous section, Preprocessing, we discussed the steps involved in the computational pipeline for making the data ready for analysis. In this section, various tools involved in the pipeline are discussed along with a summarized tabular version presented in Table IV..

RAW READS

After the sequencer finishes the sequencing process, the data which is produced are raw reads. Raw reads are sequences generated from the sequencer which are not long and are associated with some quality scores. Quality values are given to each nucleotide base. Raw reads are stored in FASTQ files [14].

1. **Illumina BaseSpace:** Most commonly used analysis software designed for Illumina sequencers. It converts fluorescence signals to base calls (FASTQ) [14]. Phred score and Bayesian models are used for base calling.
2. **bcl2fastq:** This tool converts raw Illumina base call (BCL) files into FASTQ files [14]. Also uses Phred score and error probability estimation.

QUALITY CONTROL (QC)

In this step the reads which are generated and stored in FASTQ files **[6]** are analyzed and visualized to check for various problems like low quality bases, adapter contamination, overrepresented sequences, GC-content bias, sequence duplication, K-mer enrichment anomalies. Tools used for Quality Control include:

1. **FastQC:** This is a widely used quality analysis tool. FastQC generates a report summary of various quality modules like per base quality, GC content, sequence duplication, etc [10]. It uses descriptive statistics, boxplots and histograms.
2. **MultiQC:** It provides a summary of multiple FastQC reports by using aggregation [10].
3. **PRINSEQ:** This tool also does filtering and trimming along with performing quality checks. It uses statistical thresholds for read quality scores [10].
4. **AfterQC:** This tool automates filtering, trimming, quality checks and error removal. It uses statistical statistical models for error correction [10]**.**

One of the modules that FastQC uses is K-mer Enrichment check. Here the enrichment score of k-mer frequency is checked. It is expected that the sequences which are created are random and for random reads it should have fairly random k-mer frequencies. If there is a strong enrichment of k-mer, it may suggest adapter contamination, bias in amplification and poor primer performance. Enrichment score is given by:

$$enrichment = \frac{Observed\ frequency}{Expected\ frequency}$$

Here, if enrichment >> 1, a warning flag is given indicating some form of contamination.

READ TRIMMING

Some parts of the reads may have low quality bases, contain adapter sequences and have ambiguous bases (N's). During read trimming it is ensured that these types or errors are trimmed off from the sequencing reads and only high quality bases are present for further analysis. Some read trimming tools are:

1. **Trimmomatic:** Most commonly used tool for trimming. It removes low-quality bases and adapters. It works using a sliding window technique based on mean quality [15].
2. **Cutadapt :** This tool detects and removes adapter sequences. It uses pattern matching and string matching algorithms [15].
3. **FASTP:** It is a filtering tool which uses quality score-based filtering for trimming [15].
4. **Skewer:** It uses maximum likelihood estimation (MLE) and alignment scoring to trim adapters [15].

In the sliding window algorithm [21], a window of size *k* is taken. this k can be user defined (e.g., 4 bases). The window is slid across the read from 5' to 3' end. Then average quality is computed inside the window. If average quality < a given threshold (e.g., Q=20), then trimming is done from that point onward.

Example of parameters that are set in trimming to control trimming aggressiveness are mentioned in Table II.

TABLE II: TRIM CONTROLLING PARAMETERS EXAMPLE

Parameter	Typical Value	Meaning
Sliding window size	4	Bases to average
Quality threshold	20	Minimum acceptable quality
Leading quality	3	Minimum starting base quality
Trailing quality	3	Minimum ending base quality
Minimum read length	30	After trimming

READ ALIGNMENT/MAPPING

After trimming the high quality sequences are aligned/mapped to a reference genome. The reference can be a genome or transcriptome. Some of the commonly used tools such as **STAR** align spliced transcripts to a reference. Very fast aligner tool. Uses Maximal Mappable Prefix (MMP) and seed search techniques [1]. **HISAT2** is a splice aware aligner. Uses FM-index (compressed BWT - Burrows-Wheeler Transform) and graph traversal techniques [1]. **TopHat2** [1] tool aligns by finding exon-exon junctions. Uses statistical junction discovery approach [1]. **Subread** is a high-speed aligner that uses seed-and-vote strategy for accurate alignments along with the usage of hashing and voting on candidate mappings. **Bowtie2** aligns short reads and can align with mismatches and gaps present. User BWT and FM-index with dynamic programming for local alignment [1].

QUANTIFICATION

Quantification is the next step, which gives the count of how many reads are mapped to each gene or transcript. The quantification process gives a matrix of genes and conditions (samples). The records are the genes that are mapped and the columns are the conditions. This matrix is called the count matrix. The count matrix is the starting point of various biological analyses of the gene dataset. Some commonly used tools are: (i) **featureCounts** also known as classical counting uses interval trees for efficient read-feature overlap [1]**. HTSeq-count** comes under classical counting and uses an overlap counting technique to count reads [1]. **Salmon** tool [1] is lightweight and is based on quasi-mapping quantification. Uses k-mer matching, Expectation-Maximization (EM). **Kallisto** [1] is a pseudo alignment tool which is ultra-fast in performance. Also uses k-mer indexing, EM algorithm. Salmon/kallisto uses statistical inference to know which transcript each read most likely came from. Some key differences between featureCounts/HTSeq-count and Salmon/kallisto are summarized in Table III.

TABLE III: KEY DIFFERENCES IN QUANTIFICATION TOOLS

Feature	**featureCounts/HTSeq-count**	**Salmon/kallisto**
Input	Aligned reads (BAM)	Raw or quasi-aligned reads
Speed	Slower (needs alignment)	Very fast
Math	Set-theory overlap counting	EM-based abundance estimation
Memory	Moderate	Higher
Output	Raw counts	Estimated TPM, counts

NORMALIZATION

Normalization is a key step in the RNA-Seq computational pipeline. Every dataset should be first processed with normalization before moving to analysis. Normalization method can be generally divided into two types: Classical Methods and using tools like DESeq2 [16][3], edgeR [17][3], limma-voom [3]**.**In Classical Methods the following types of normalizations are used:

1. **RPKM/FPKM (Reads/Fragments Per Kilobase Per Million)** [12] is used if we are trying to study gene expressions within the same sample. Given a sample/condition, we can tell how different genes are expressed in the same sample/condition. For FPKM, the formula remains the same and is used for pair-ended sequencing instead of single-ended sequencing. Fragments are used instead of reads in the above formula.
2. **TPM (Transcripts Per Million)** [12] is similar to RPKM/FPKM. It rescales its expression values to make the total sum consistent across all samples/conditions which equals to 1 million. It is also used for expression comparison within the same sample of different genes.
3. Using tools to perform normalization is more common in modern day RNA-Seq. Some tools are: **DESeq2** uses the Median-of-ratios method to normalize the reads. Here median scaling and size factors are taken into account. The **edgeR** tool uses TMM normalization for RNA-seq libraries. It is based on trimmed mean of log expression ratios. The **limma-voom** is a precision weight modeling technique for RNA-seq. It is based on Mean-variance modeling and empirical Bayes smoothing.

DE ANALYSIS TOOLS

For performing gene expression analysis to study differential expressions between genes and samples tools like DESeq2, edgeR, limma-voom are used. Performing differential expression with these tools also includes the normalization step described in the above section. **DESeq2** uses Negative binomial Generalized Linear Model (GLM), Wald Test, Likelihood Ratio Test for gene expression. **edgeR** also uses Negative binomial modeling, Exact test and GLM. **limma-voom** uses linear modeling

technique with the Bayes method. **NOISeq** [18] uses probabilistic noise modeling and ranking for non-parametric DE analysis.
Detail about the working of DESeq2, edgeR, limma-voom and NOISeq is given in Differential Analysis (DE) section.

VISUALIZATION

For visualization some commonly used tools are [19]: **EnhancedVolcano** use -log(p-value) vs log(fold-change) for volcano plots in DE visualization. **ComplexHeatmap** for visualizing multi-layered datasets is used. It uses clustering and dendrograms. The **pheatmap** uses distance matrices and hierarchical clustering to visualize gene expressions.

ML APPLICATION TOOLS

Nowadays, machine learning and deep learning neural networks are widely used for biological interpretation and predictive modeling of RNA-Seq pipeline, various tools can be helpful. The **scikit-learn** package contains all general purpose machine learning models and algorithms. Traditional approaches like SVM, Random Forests, PCA, KMeans, etc. can be applied with the help of this package [5]. **TensorFlow/Keras** [5] is used for deep learning and neural net applications. It contains Dense Neural Networks, Convolutional Neural Networks, Recurrent Neural Networks, etc . **MLSeq** [20] is an R package that acts as a framework for supervised classification. **PyTorch** [5] is an alternative DL library to TensorFlow/Keras developed by PyTorch. **TPOT/H2O.ai** [4] are AutoML used in pipeline optimization and can be used for Genetic programming, ensemble learning etc.

TABLE IV: Summary OF Tools In RNA-Seq

Steps	Tool	Technique
Raw Reads	Illumina BaseSpace	Phred score, Bayesian basecalling
	bcl2fastq	Error estimation, BaseCall to FASTQ
Quality Control	FastQC	Descriptive statistics
	MultiQC	Aggregation
	PRINSEQ	Quality filtering
	AfterQC	Error correction
Read Trimming	Trimmomatic	Sliding window mean quality
	Cutadapt	Pattern matching
	FASTP	Quality score filtering
	Skewer	MLE for adapter detection
Alignment/Mapping	STAR	MMP seeding
	HISAT2	FM-index, BWT
	TopHat2	Bowtie mapping and junction finding
	Subread	Seed-and-vote
	Bowtie2	BWT and dynamic programming
Quantification	featureCounts	Interval trees
	HTSeq-count	Overlap counting
	Salmon	k-mer matching, EM
	Kallisto	Pseudoalignment, EM
Normalization	DESeq2	Median of ratios
	edgeR	TMM normalization
	limma-voom	Precision weights
Differential Expression	DESeq2	Negative Binomial GLM
	edgeR	Negative Binomial model
	limma-voom	Linear model

	NOISeq	Noise modeling
Visualization	pheatmap	Clustering heatmaps
	EnhancedVolcano	Volcano plots
	ComplexHeatmap	Hierarchical clustering
ML/AI Applications	scikit-learn	SVMs, RF, PCA
	TensorFlow/Keras	DL library for neural networks
	MLSeq	Supervised classification framework
	Pytorch	DL library alternative to TF/Keras
	TPOT/H2O.ai	Genetic programming

V. DIFFERENTIAL EXPRESSION (DE) ANALYSIS

After normalization of the dataset using the techniques discussed above, differential expression analysis of the genes is done. It helps to statistically identify genes that are expressed at different levels between two or more biological samples/conditions. This is a primary step before studying for biological interpretations. RNA-Seq data has discrete counts instead of continuous values, thereby proving a challenge for expression and biological interpretation. Challenges in variance in counts (overdispersion) and different genes having different variances depending on expression levels are tackled in DE using statistical tools like DESeq2, edgeR, NOISeq and limma-voom. An overview of the statistical methods used by these tools given in Table V.

In differential expression (DE) analysis, statistical hypothesis testing [6] is fundamental. For each gene, a null hypothesis (H_0) is typically set, stating that there is no difference in expression between conditions. Tools such as DESeq2, edgeR, and limma-voom evaluate this hypothesis using p-values derived from statistical tests (e.g., Wald test, exact test, moderated t-test). A gene is considered significantly differentially expressed if the p-value falls below a set threshold (commonly 0.05). To control for multiple testing (thousands of genes tested), methods such as the Benjamini-Hochberg correction are applied to adjust p-values (yielding the FDR or adjusted p-value). Thresholds like adjusted p-value < 0.05 and minimum fold-change (e.g.,$|log_2 FC| > 1$) are used to define significant DEGs.

TABLE V: STATISTICAL METHODS OF DE TOOLS

Method	Processes	Tool
Negative Binomial distribution	Models overdispersed counts	DESeq2, edgeR
Quasi-likelihood Negative Binomial	More robust estimation of variability	edgeR (advanced GLM QL)
Shrinkage Estimation	Stabilizes fold changes and dispersions	DESeq2 (apeglm, ashr)
Non-parametric, Noise Modeling	Technical noise modeled without any parametric assumption	NOISeq
Mean-Variance Modeling (log-CPM transformation)	Mean-variance trend is modeled after log-transformation	limma-voom

DESeq2

DESeq2 [16] introduced by Love, Huber and Andres in 2014, is a widely used tool for differential gene expression analysis. It begins by normalizing raw counts using the Median of Ratios method and models gene counts with Negative Binomial distribution [16] to account for biological variability and overdispersion. Mean expression is modeled using size factors and true expression levels while dispersion is estimated with a smooth trend and stabilized using empirical Bayes shrinkage. The Wald Test is used to assess differential expression, with multiple testing controlled via the Benjamini-Hochberg method to limit the False discovery

Rate (FDR). For low-count genes, fold changes shrinkage methods like apeglm or ashr improve result reliability.

edgeR

edgeR [17], developed by Mark Robinson, Davis McCarthy, and Gordon Smyth in 2010, is a popular tool for differential expression analysis using a Negative Binomial model to account for biological and technical variability. It uses TMM normalization to correct for sequencing depth and composition biases, edgeR estimates three types of dispersion namely common, trended and tagwise, which is enhanced by an Empirical Bayes method [17] for stable estimates, especially with small sample sizes. For statistical testing, it offers the Exact Test for two-group comparisons and a Generalized Linear Model (GLM) framework for complex designs. The Quasi-Likelihood F-test (QLF) further improves robustness and control over false positives making it a preferred method in RNA-Seq workflows.

NOISeq

NOISeq was introduced as a non-parametric approach to detect differentially expressed genes, developed by Tarazona, Furió-Tarí, Conesa, et al [18]. Unlike DESeq2 and edgeR, it does not assume a particular underlying distribution (like Negative Binomial) but instead, it models technical and biological noise directly from the data using resampling. It typically applies Upper Quartile or TMM normalization and builds a noise distribution by comparing genes within or across randomized conditions. Instead of p-values, NOISeq computes a probability score(q), where higher q-values (e.g., > 0.8) indicate stronger evidence of differential expression. It evaluates both fold-change and absolute expression differences, making it suitable when parametric assumptions may not hold.

Limma-voom

The limma-voom method introduced by Charity W Law, Yunshun Chen, Wei Sei and Gordon K Smyth in 2014 [21], adapts the limma framework (originally designed for microarrays) to RNA-seq data by transforming count data into a form suitable for linear modeling. The term "voom" stands for "variance modeling at the observational level, where precision weights are estimated from the mean-variance trend. After normalization, typically using TMM or Upper Quartile methods, voom generates a logCPM matrix and a weight matrix. These weights are incorporated into the linear model and empirical Bayes moderation is applied to stabilized variance estimates similar to microarray analysis [21].

A comparison table across the methods DESeq2, edgeR, Limma-voom and NOISeq is given in Table VI.

VI. Pathway Analysis:

After identifying differentially expressed genes (DEGs), pathway enrichment analysis [22] was conducted to determine which biological pathways were most significantly affected by the experimental conditions. This helped interpret large DEG lists by identifying overrepresented biological processes, molecular functions, and signaling pathways. Two main strategies were used: Overrepresentation Analysis (ORA) and Gene Set Enrichment Analysis (GSEA) [22] ORA assessed whether pathways contained more DEGs than expected by chance, while GSEA analyzed all ranked genes to detect enriched pathways without relying on arbitrary cutoffs.

The DEGs were analyzed for enrichment using curated databases such as KEGG, Reactome, and Gene Ontology (GO: (Biological Process (BP), Molecular Function (MF), Cellular Component (CC)). Tools like g:Profiler, clusterProfiler, and Enrichr [4] were employed. Statistical significance was determined via hypergeometric testing, with Benjamini-Hochberg correction applied to control the false discovery rate (FDR). Pathways with FDR-adjusted p-values < 0.05 were considered significantly enriched. In GSEA, all genes were ranked by fold change and statistical significance, and enrichment scores were calculated to identify pathways showing coordinated expression shifts, without using a predefined significance threshold. This enabled detection of subtle but consistent changes across pathways. A comparison of ORA and GSEA is presented in Table VII. Enrichment results were visualized using bar and dot plots to display top pathways based on gene ratios and adjusted p-values. Network-based visualizations with Cytoscape and EnrichmentMap [23] clustered related pathways, revealing broader biological themes and reducing redundancy. These insights highlighted key molecular mechanisms and potential targets for further investigation.

TABLE VI: METHOD COMPARISON TABLE

Feature	DESeq2	edgeR	Limma-voom	NOISeq
Primary Statistical Test	Wald test, LRT	Exact test, GLM + QL F-test	Moderated t-test (Emprical Bayes)	Empirical probability (q-score)
Data transformation	None (modeled directly)	None(modeled directly)	log2-counts per million (logCPM) via voom	None(non-parametric approach)
Distribution assumption	Negative Binomial	Negative Binomial	Log-normal after voom transformation	None(learns noise empirically)
Variance modelling	Gene-specific dispersion + shrinkage	Tagwise + common dispersion	Observation-level weights + EB moderation	Empirical noise modeling
Normalization method	Median of RAtios	TMM normalization	TMM(with voom) or quantile (microarray)	Upper quantile or TMM normalization
Fold-change shrinkage	Yes (aseglm, ashr)	No direct shrinkage (robust tests)	VAriance moderation improves FC stability	No shrinkage
Handling of complex designs	Good(with LRT)	Excellent (GLM, QL F-test)	Excellent(multifactor designs, batches)	Limited(basic comparisons preferred)
When preferred	Small datasets, need strong shrinkage	Complex designs, small samples robust	Very large datasets, complex experimental designs	Datasets where parametric assumptions are questionable
Multiple testing correction	Benjamini-Hochberg FDR correction	Benjamini-Hochberg FDR correction	Benjamini-Hochberg FDR correction	Not required(uses threshold on q scores)

TABLE VII: COMPARISION ACROSS ORA AND GSEA

	ORA	GSEA
When done?	After selecting significant DEGs (padj <0.05, etc)	Directly from all genes ranked by fold-change (without cutoff)
Input	A subset of genes (DEGs only)	A ranked list of all genes.
Detects	Overrepresented pathways in DEGs	Enriched pathways even if gene changes are small but coordinated.
Tools	gProfiler, Enrichr, clusterProfiler (R package), DAVID	GSEA software, DAVID, Ingenuity

VII. ADVANCED ANALYSIS (ML/DL)

Over the past two decades, analyses such as differential expression (DE) and pathway enrichment (PE) have yielded profound biological insights, underpinning numerous landmark studies [24]. Beyond these foundational approaches, investigations into alternative splicing, novel transcript discovery, allele-specific expression, and single-cell RNA-Seq have further expanded our understanding of transcriptomic complexity [2]. Concurrently, biomarker discovery has emerged as a powerful strategy for characterizing disease presence, monitoring progression, and predicting treatment response. By integrating metabolomics and immunohistochemistry (IHC) data with RNA-Seq results, researchers can obtain a more comprehensive view of how gene expression relates to downstream metabolite production and protein abundance, thereby enhancing biomarker identification [25]. Dimensionality-reduction techniques such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), and uniform manifold approximation and projection (UMAP) reveal latent structures in high-dimensional data, highlighting clusters or patterns that may correspond to distinct biological states [1]. Network-based approaches, including gene co-expression networks and multi-omics integration, further enable the discovery of phenotype-linked genes and candidate biomarkers [26]. Survival analysis methods can then correlate expression levels with time-to-event outcomes, facilitating prognostic biomarker identification. Finally, machine-learning (ML) and deep-learning (DL) models employing feature-selection algorithms alongside

classifiers such as support vector machines, random forests, and XGBoost, or neural-network architectures offer predictive frameworks for prioritizing key genes or metabolites [4][5]. A comprehensive summary of these analytical strategies is presented in Table VIII.

Despite these advances, applying ML and DL to RNA-Seq data presents several challenges. The intrinsically high dimensionality and sparsity of gene-expression matrices can impede model training, particularly when sample sizes are limited, leading to overfitting and unreliable predictions. Moreover, the scarcity of richly annotated, real-world datasets hampers supervised learning, while class-imbalance issues further complicate classifier performance. Deep-learning approaches, in particular, demand large volumes of data to achieve generalizable models, and even extensive RNA-Seq cohorts may fall short of this requirement. These factors coupled with the difficulty of interpreting complex model outputs and the absence of standardized benchmarking platforms pose significant hurdles for robust, reproducible ML/DL applications in transcriptomics.

TABLE VIII: ML/DL MODELS IN RNA-SEQ

Advanced Analyses	Analysis Features
Metabolomics with RNA-Seq	Metabolite abundance or concentration identification
IHC (Immunohistochemistry) with RNA-Seq	Studies presence or specific proteins within tissue samples
Dimensionality Reduction	Reveals patterns or clusters of various biological states
Network Analysis	Studies gene co-expression networks
Survival Analysis	Identification of prognostic biomarkers
Machine Learning	Designing predictive models for biological insights

VIII. CONCLUSION

Understanding the complexities of transcriptomes is now advanced by RNA-seq. A robust computational pipeline is thereby necessary to extract meaningful insights from sequencing data. Traditional statistical methods supported by tools like DESeq2,edgeR, etc have backboned the RNA-seq pipeline. But with the increasing data dimensionality and complexity, the integration of Machine learning and Deep learning is becoming indispensable. Nevertheless, challenges such as overfitting, data sparsity and interpretability need to be addressed. Advancement in RNA-seq pipeline will increasingly rely on hybrid pipelines that blend traditional statistical ways with ML/DL's predictive strength, unlocking deeper biological insights.

ACKNOWLEDGEMENT

This publication is an outcome of the R&D work undertaken under the Visvesvaraya PhD Scheme Phase II of Ministry of Electronics & Information Technology, Government of India, being implemented by Digital India Corporation. The work is undergoing at the BioNet Lab of Tezpur University.

REFERENCES

[1] L. A. Corchete, E. A. Rojas, D. Alonso-López, J. De Las Rivas, N. C. Gutiérrez, and F. J. Burguillo, "Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis," Sci. Rep., vol. 10, no. 1, p. 19737, 202. DOI: https://doi.org/10.1038/s41598-020-76881-x

[2] Li, X., Wang, CY. From bulk, single-cell to spatial RNA sequencing. *Int J Oral Sci* 13, 36 2021. DOI: https://doi.org/10.1038/s41368-021-00146-0

[3] Liu, Shiyi, Zitao Wang, Ronghui Zhu, Feiyan Wang, Yanxiang Cheng, and Yeqiang Liu. "Three differential expression analysis methods for RNA sequencing: limma, EdgeR, DESeq2." *Journal of Visualized Experiments (JoVE)* 175 2021: e62528. DOI: https://doi.org/10.3791/62528

[4] Khalsan, Mahmood, et al. 'A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction'. IEEE Access, vol. 10, 2022, pp. 27522–34. DOI.org (Crossref), https://doi.org/10.1109/ACCESS.2022.3146312.

[5] Pandey, Diksha, and P. Onkara Perumal. 'A Scoping Review on Deep Learning for Next-Generation RNA-Seq. Data Analysis'. *Functional & Integrative Genomics*, vol. 23, no. 2, June 2023, p. 134. *DOI.org (Crossref)*,https://doi.org/10.1007/s10142-023-01064-6.

[6] Conesa, Ana, et al. 'A Survey of Best Practices for RNA-Seq Data Analysis'. *Genome Biology*, vol. 17, no. 1, Jan. 2016, p. 13. *BioMed Central*, https://doi.org/10.1186/s13059-016-0881-8.

[7] Deshpande, Dhrithi, et al. 'RNA-Seq Data Science: From Raw Data to Effective Interpretation'. *Frontiers in Genetics*, vol. 14, Mar. 2023. *Frontiers*, https://doi.org/10.3389/fgene.2023.997383.

[8] K. Mandal, R. Sarmah, and D. K. Bhattacharyya, "POPTric: Pathway-based Order Preserving Triclustering for gene sample time data analysis," *Expert Syst. Appl.*, vol. 192, p. 116336, 2022. DOI: https://doi.org/10.1109/tcbb.2020.2980816

[9] Bray, Nicolas L., et al. 'Near-Optimal Probabilistic RNA-Seq Quantification'. *Nature Biotechnology*, vol. 34, no. 5, May 2016, pp. 525–27. *www.nature.com*, https://doi.org/10.1038/nbt.3519.

[10] Koch, Clarissa M., et al. 'A Beginner's Guide to Analysis of RNA Sequencing Data'. *American Journal of Respiratory Cell and Molecular Biology*, vol. 59, no. 2, Aug. 2018, pp. 145–57. *PubMed Central*, https://doi.org/10.1165/rcmb.2017-0430TR.

[11] Tadist, Khawla, et al. 'Feature Selection Methods and Genomic Big Data: A Systematic Review'. *Journal of Big Data*, vol. 6, no. 1, Aug. 2019, p. 79. *BioMed Central*, https://doi.org/10.1186/s40537-019-0241-0.

[12] Zhao, Yingdong, et al. 'TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-Seq Data from the NCI Patient-Derived Models Repository'. *Journal of Translational Medicine*, vol. 19, no. 1, June 2021, p. 269. *BioMed Central*, https://doi.org/10.1186/s12967-021-02936-w.

[13] Arowolo, Micheal Olaolu, et al. 'A Survey of Dimension Reduction and Classification Methods for RNA-Seq Data on Malaria Vector'. *Journal of Big Data*, vol. 8, no. 1, Mar. 2021, p. 50. *BioMed Central*, https://doi.org/10.1186/s40537-021-00441-x.

[14] Ewing, Brent, et al. 'Base-Calling of Automated Sequencer Traces Using *Phred.* I. Accuracy Assessment'. *Genome Research*, vol. 8, no. 3, Mar. 1998, pp. 175–85. *DOI.org (Crossref)*, https://doi.org/10.1101/gr.8.3.175.

[15] Bolger, Anthony M., et al. 'Trimmomatic: A Flexible Trimmer for Illumina Sequence Data'. *Bioinformatics*, vol. 30, no. 15, Aug. 2014, pp. 2114–20. *DOI.org (Crossref)*, https://doi.org/10.1093/bioinformatics/btu170.

[16] Love, Michael I., et al. 'Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2'. *Genome Biology*, vol. 15, no. 12, Dec. 2014, p. 550. *BioMed Central*, https://doi.org/10.1186/s13059-014-0550-8.

[17] Robinson, Mark D., et al. 'edgeR : A Bioconductor Package for Differential Expression Analysis of Digital Gene Expression Data'. *Bioinformatics*, vol. 26, no. 1, Jan. 2010, pp. 139–40. *DOI.org (Crossref)*, https://doi.org/10.1093/bioinformatics/btp616.

[18] Tarazona, Sonia, Pedro Furió-Tarí, David Turrà, Antonio Di Pietro, María José Nueda, Alberto Ferrer, and Ana Conesa. "Data quality aware analysis of differential expression in RNA-seq with NOISeq R/Bioc package." *Nucleic acids research* 43, no. 21 (2015): e140-e140.

[19] Griffith, Malachi, et al. "Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud." *PLOS Computational Biology*, vol. 11, no. 8, Aug. 2015, p. e1004393. *PLoS Journals*, https://doi.org/10.1371/journal.pcbi.1004393.

[20] Goksuluk, Dincer, et al. 'MLSeq: Machine Learning Interface for RNA-Sequencing Data'. *Computer Methods and Programs in Biomedicine*, vol. 175, July 2019, pp. 223–31. *DOI.org (Crossref)*, https://doi.org/10.1016/j.cmpb.2019.04.007.

[21] Law, Charity W., Yunshun Chen, Wei Shi, and Gordon K. Smyth. "voom: Precision weights unlock linear model analysis tools for RNA-seq read counts." *Genome biology* 15 (2014): 1-17.

[22] D. Chicco and G. Agapito, "Nine quick tips for pathway enrichment analysis," *PLoS Computational Biology*, vol. 18, no. 8, p. e1010348, 2022.

[23] J. Reimand, R. Isserlin, V. Voisin, M. Kucera, C. Tannus-Lopes, A. Rostamianfar, L. Wadi, M. Meyer, J. Wong, and C. Xu, "Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap," *Nature Protocols*, vol. 14, no. 2, pp. 482–517, 2019.

[24] K. Mandal, R. Sarmah, and D. K. Bhattacharyya, "Biomarker identification for cancer disease using biclustering approach: An empirical study," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 2, pp. 490–509, 2018.

[25]Meller, S., et al., "Integration of tissue metabolomics, transcriptomics and immunohistochemistry reveals ERG-and gleason score-specific metabolomic alterations in prostate cancer," *Oncotarget*, vol. 7, no. 2, pp. 1421, 2015.

[26] Argelaguet, Ricard, et al. 'Multi-Omics Factor Analysis—a Framework for Unsupervised Integration of Multi-omics Data Sets'. *Molecular Systems Biology*, vol. 14, no. 6, June 2018, p. e8124. *DOI.org (Crossref)*, https://doi.org/10.15252/msb.20178124.