Normalization Methods in RNA-Seq Analysis

RPKM/FPKM Issues

Reads/Fragments Per Kilobase Million - biased by composition

TPM Calculation

Transcripts Per Million - better for comparison

DESeq2 Normalization

Median-of-ratios method for differential expression

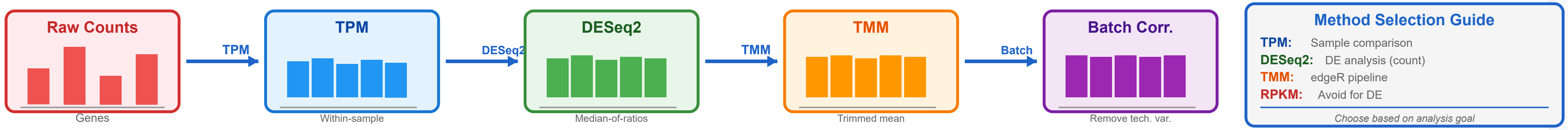
TMM Method

Trimmed Mean of M-values - robust to outliers

Batch Correction

ComBat, limma removeBatchEffect for technical variation

> 💡 Choose normalization method based on your downstream analysis goals



Method Selection Guide

TPM:	Sample comparison
DESeq2:	DE analysis (count)
TMM:	edgeR pipeline
RPKM:	Avoid for DE

Choose based on analysis goal

Detailed Normalization Methods

1 RPKM/FPKM - Reads/Fragments Per Kilobase Million

RPKM (Reads Per Kilobase Million) and FPKM (Fragments Per Kilobase Million) were among the first normalization methods developed for RNA-seq data. They normalize for both sequencing depth and gene length.

Formula

```
RPKM = (Number of reads mapped to gene × 10⁹) /
(Total mapped reads × Gene length in bp)

FPKM uses fragments instead of reads (for paired-
end)
```

Key Concept

Normalizes by scaling counts to account for differences in library size and gene length. The method assumes that most genes are not differentially expressed.

✓ Advantages

- Intuitive and simple
- Accounts for gene length
- Early standard method

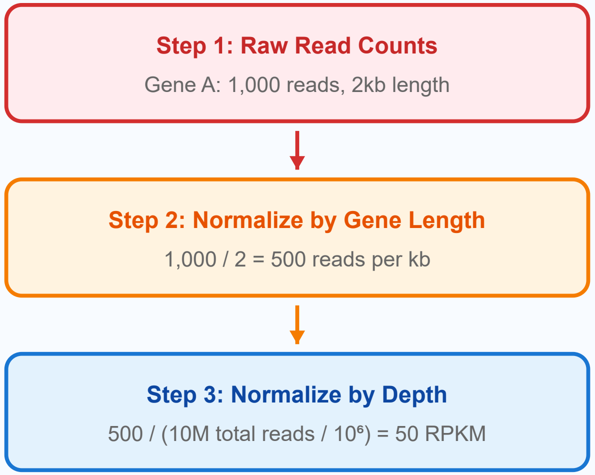
✗ Disadvantages

- Composition bias issue
- Not suitable for DE analysis
- Sample-specific biases
- Values don't sum consistently

⚠ Usage Recommendation

Avoid RPKM/FPKM for differential expression analysis. Better alternatives like TPM, DESeq2, and edgeR are now standard. May still be used for within-sample comparisons or legacy pipelines.

RPKM Calculation Steps



Step 1: Raw Read Counts

Gene A: 1,000 reads, 2kb length

Step 2: Normalize by Gene Length

1,000 / 2 = 500 reads per kb

Step 3: Normalize by Depth

500 / (10M total reads / 10^6) = 50 RPKM

⚠ Composition Bias Problem

If a few highly expressed genes dominate,
they artificially reduce RPKM of other genes
→ Not comparable across samples!

2 TPM - Transcripts Per Million

TPM (Transcripts Per Million) is an improved normalization method that addresses the composition bias issues of RPKM/FPKM. The key difference is the order of normalization steps.

Formula

```
Step 1: RPK = Reads / (Gene length in kb)
Step 2: Sum all RPK values in sample
Step 3: TPM = (RPK / Sum of all RPK) × 10⁶
```

Key Advantage

TPM values always sum to the same total (1 million) in each sample, making cross-sample comparisons more meaningful. The sum of all TPM values in a sample is constant, unlike RPKM.

✓ Advantages

- Better for cross-sample comparison
- Sums to same value per sample
- Reduces composition bias
- Interpretable proportions

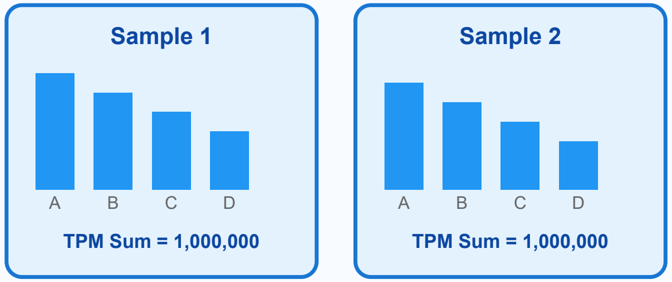
✗ Disadvantages

- Still not ideal for DE testing
- Assumes constant RNA output
- Requires gene length info

✓ Usage Recommendation

Excellent for comparing expression levels across samples, visualization, and reporting relative abundances. Preferred over RPKM/FPKM for most visualization purposes.

TPM vs RPKM Comparison



✓ Key Benefit: Consistent Totals

TPM always sums to 1M, enabling direct comparison

TPM Calculation Order

1. Normalize by gene length → RPK
2. Calculate sum of all RPK values
3. Scale to 1 million → TPM

(Order matters! Different from RPKM)

3 DESeq2 - Median-of-Ratios Normalization

DESeq2 uses a sophisticated median-of-ratios method to estimate size factors for normalization. It's specifically designed for differential expression analysis with count data and handles biological variability effectively.

Algorithm Steps

```
1. Calculate geometric mean for each gene across
samples
2. For each sample, calculate ratio to geometric
mean
3. Take median of ratios → size factor
4. Divide raw counts by size factor
```

Key Innovation

Uses a negative binomial model to account for biological variability. Robust to genes with zero counts and doesn't require gene length information. Automatically filters out genes with all zeros or extreme outliers.

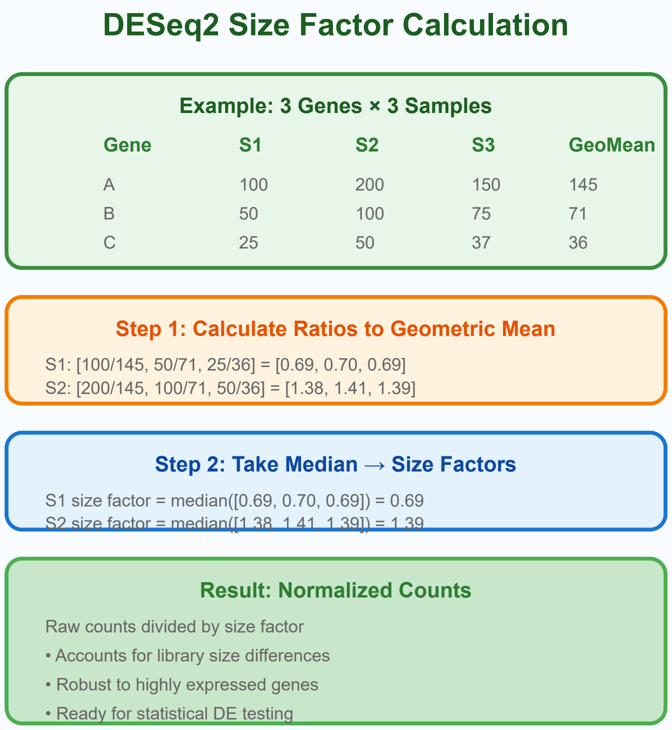
✓ Advantages

- Gold standard for DE analysis
- Handles biological replicates well
- Robust to outliers
- Built-in statistical testing
- No gene length needed

✗ Disadvantages

- Requires raw count data
- Computationally intensive
- Needs biological replicates

✓ Usage Recommendation



DESeq2 Size Factor Calculation

Example: 3 Genes × 3 Samples

Gene	S1	S2	S3	GeoMean
A	100	200	150	145
B	50	100	75	71
C	25	50	37	36

Step 1: Calculate Ratios to Geometric Mean

S1: [100/145, 50/71, 25/36] = [0.69, 0.70, 0.69]
S2: [200/145, 100/71, 50/36] = [1.38, 1.41, 1.39]

Step 2: Take Median → Size Factors

S1 size factor = median([0.69, 0.70, 0.69]) = 0.69
S2 size factor = median([1.38, 1.41, 1.39]) = 1.39

Result: Normalized Counts

Raw counts divided by size factor
• Accounts for library size differences
• Robust to highly expressed genes
• Ready for statistical DE testing

The preferred method for differential gene expression analysis. Use when you have raw count data with biological replicates and want to identify statistically significant changes between conditions.

4 TMM - Trimmed Mean of M-values

TMM (Trimmed Mean of M-values) is the normalization method used by edgeR. It calculates scaling factors based on the weighted trimmed mean of log-ratios between samples, making it highly robust to outliers and composition biases.

Algorithm

```
M-values = $\log_2$(Sample / Reference)
A-values = $\frac{1}{2} \times \log_2$(Sample × Reference)
Trim extreme M and A values (default: 30% & 5%)
Calculate weighted mean → TMM factor
```

Key Features

The method trims both extreme fold-changes (M) and extreme absolute expression levels (A), then calculates a weighted mean. This makes it exceptionally robust to outliers and genes with very high or low expression.

✓ Advantages

- Very robust to outliers

✗ Disadvantages

- Needs reference sample selection

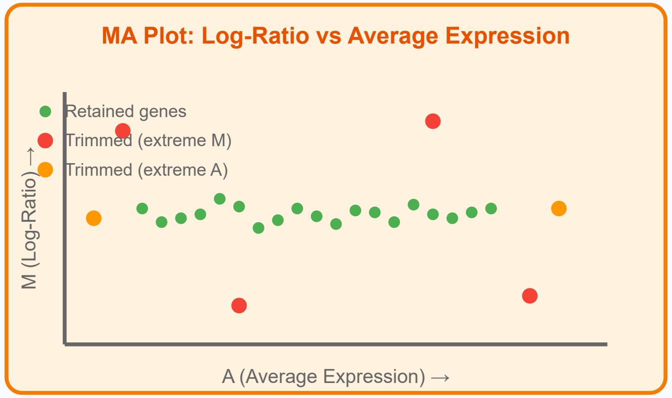
- Handles composition bias well
- Works with edgeR pipeline
- Fast computation
- Effective for many scenarios

- Assumptions may fail with extreme DE
- Less intuitive than DESeq2

✓ Usage Recommendation

Excellent choice for differential expression with edgeR. Particularly good when you have concerns about outliers or composition bias. Often produces similar results to DESeq2 but with faster computation.

TMM Trimming Strategy


MA Plot: Log-Ratio vs Average Expression

Trimming Parameters

- M-trim: Remove top/bottom 30% by fold-change
- A-trim: Remove top/bottom 5% by expression
- Calculate weighted mean of remaining genes
- Weighting reduces influence of low-count genes

Result: Robust normalization factors resistant to outliers

5 Batch Effect Correction

Batch effects are systematic non-biological variations in data caused by technical factors like different sequencing runs, dates, operators, or reagent lots. Batch correction methods remove these technical artifacts while preserving biological variation.

Common Methods

```
ComBat (sva package):
Empirical Bayes method to adjust for known batches
```

```
limma::removeBatchEffect:
Linear model-based batch removal

SVA (Surrogate Variable Analysis):
Identifies and removes unknown batch effects
```

Key Considerations

Batch correction should be applied AFTER normalization but BEFORE differential expression testing. Be careful not to remove biological signal along with batch effects. Always visualize data before and after correction using PCA plots.

✓ Advantages

- Removes technical variation
- Improves signal detection
- Enables multi-cohort analysis
- Essential for meta-analysis

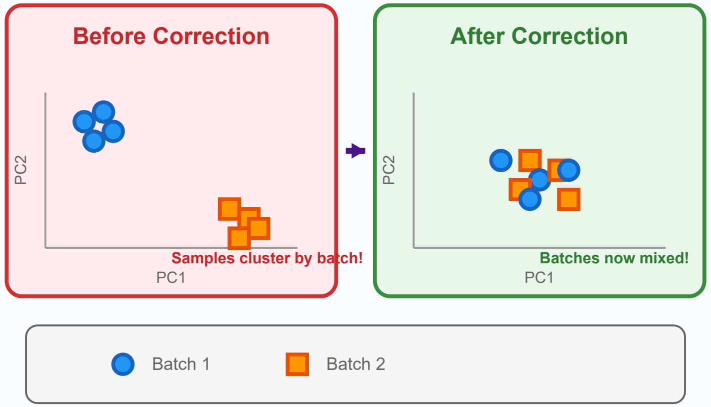
✗ Disadvantages

- Can remove biological signal
- Requires careful validation
- Batch info must be known (ComBat)
- May overcorrect

⚠ Usage Recommendation

Use batch correction when you have clear technical batches (different sequencing runs, dates, labs). Always validate with PCA plots. For DE analysis, consider including batch as a covariate in the model instead of pre-correcting.

Batch Effect Correction



Common Batch Correction Methods

ComBat:	Empirical Bayes, needs batch labels
limma:	Linear model, fast and simple
SVA:	Finds unknown batch effects (surrogate variables)
RUVSeq:	Uses control genes to estimate variation

⚠ *Always validate with PCA before and after correction*

📊 Quick Selection Guide

For Visualization & Comparison:

For Differential Expression:

Use **TPM** - consistent totals enable direct comparison between samples

Use **DESeq2** or **TMM/edgeR** - statistical models for count data

Avoid for DE Analysis:

RPKM/FPKM - composition bias makes them unsuitable for statistical testing

For Multi-Batch Experiments:

Apply **batch correction** after normalization, validate with PCA