Statistical Testing in RNA-seq Analysis

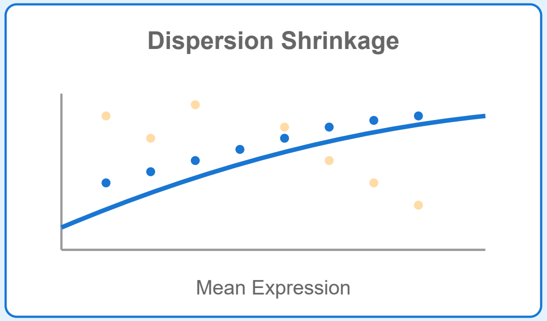
RNA-seq Statistical Methods Comparison

DESeq2

$Y \sim NB(\mu, \alpha)$

Key Features:

- Shrinkage estimation of dispersion
- Size factor normalization
- Wald test / LRT



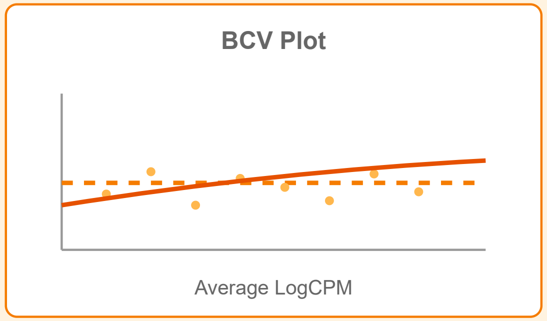
Dispersion Shrinkage

edgeR

$Y \sim NB(\mu, \varphi)$

Key Features:

- Empirical Bayes methods
- TMM normalization
- Quasi-likelihood F-test



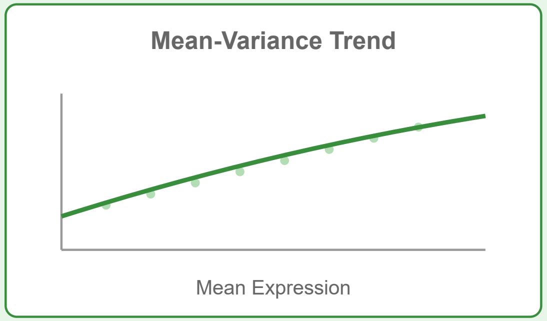
BCV Plot

limma-voom

$\log(Y) \sim N(\mu, \sigma^2)$

Key Features:

- Transform to log-space
- Precision weights (voom)
- Linear modeling



Mean-Variance Trend

Performance: All methods perform similarly with proper use • Choose based on experimental design and analysis goals

💡 DESeq2 and edgeR are most widely used and well-validated

📊 DESeq2 - Detailed Analysis

Overview

DESeq2 is a statistical method for differential gene expression analysis based on the negative binomial distribution. It uses shrinkage estimation to improve dispersion estimates, especially for genes with low counts or small sample sizes. DESeq2 is particularly robust for experiments with small sample sizes (3-5 replicates per condition).

Statistical Model

DESeq2 models RNA-seq count data using a negative binomial (NB) distribution:

$$K \sim NB(\mu, \alpha)$$

NB

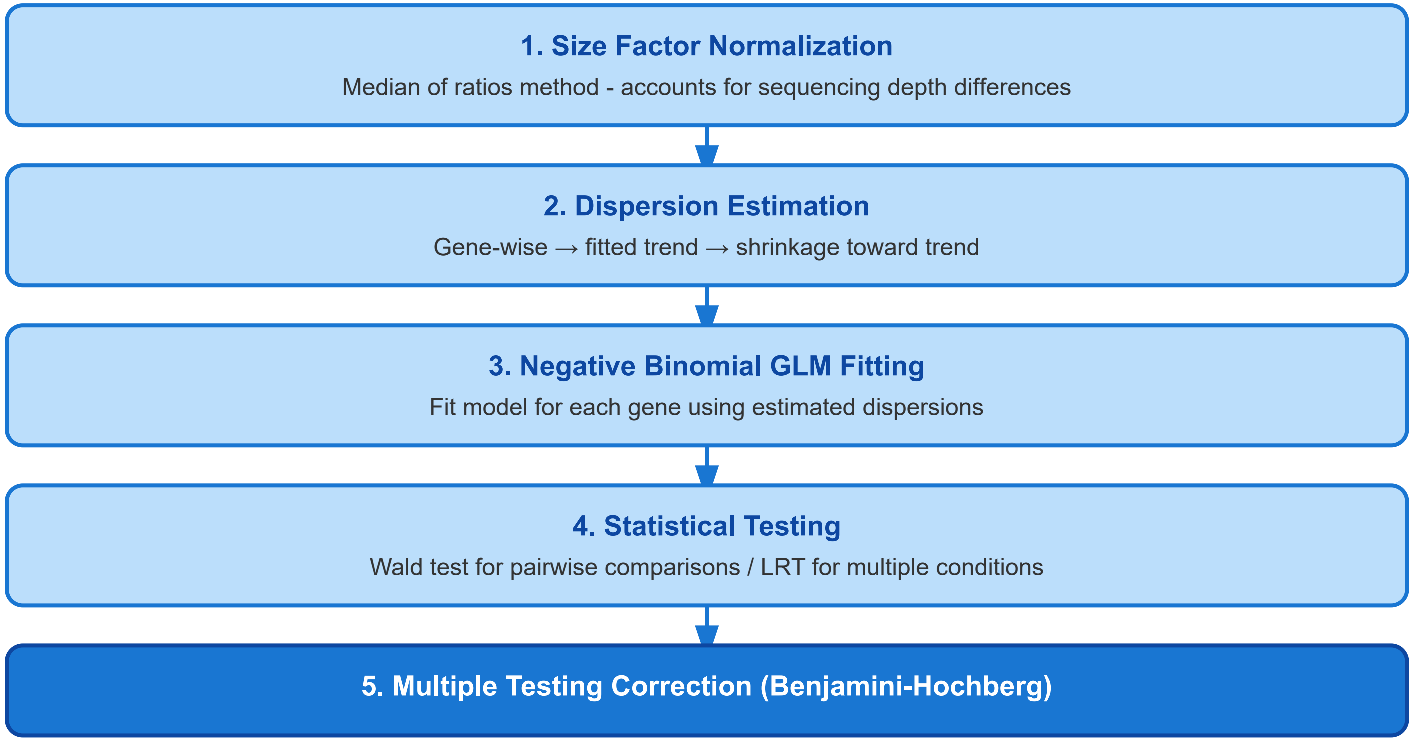
K = Read counts for a gene

μ = Mean expression level (affected by sequencing depth and biological condition)

α = Dispersion parameter (controls variance beyond mean)

The mean μ is modeled as: $\mu = s \times q$, where s is the size factor (normalization) and q is the expected count related to the biological condition.

Analysis Workflow



1. Size Factor Normalization

Median of ratios method - accounts for sequencing depth differences

2. Dispersion Estimation

Gene-wise → fitted trend → shrinkage toward trend

3. Negative Binomial GLM Fitting

Fit model for each gene using estimated dispersions

4. Statistical Testing

Wald test for pairwise comparisons / LRT for multiple conditions

5. Multiple Testing Correction (Benjamini-Hochberg)

Key Features & Advantages

- **Shrinkage Estimation:** Borrows information across genes to improve dispersion estimates, particularly beneficial for genes with low counts. This reduces false positives while maintaining sensitivity.
- **Size Factor Normalization:** Uses geometric mean of ratios method, which is robust to outliers and doesn't assume most genes are unchanged between conditions.

- **Small Sample Performance:** Performs well with as few as 2-3 replicates per condition, though 3-5 is recommended for reliable results.

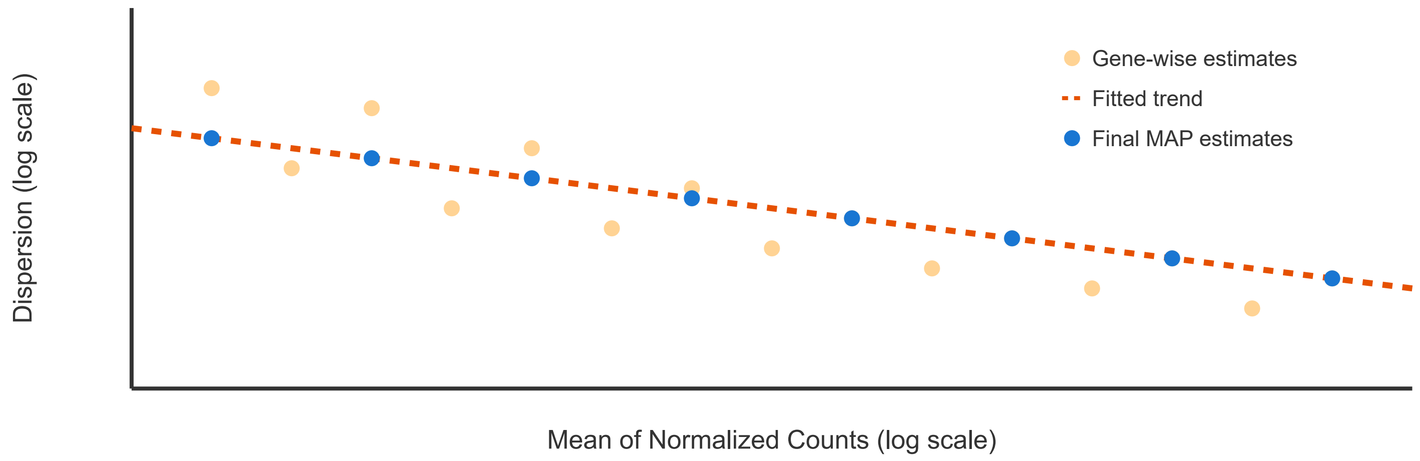
- **Independent Filtering:** Automatically filters out low-count genes to increase statistical power for detecting truly differentially expressed genes.

Example R Code

```
# Load library library(DESeq2) # Create DESeq2 dataset dds <- DESeqDataSetFromMatrix( countData = counts,
colData = metadata, design = ~ condition ) # Run DESeq2 analysis (all steps in one) dds <- DESeq(dds) #
Extract results results <- results(dds, contrast = c("condition", "treated", "control")) # Apply shrinkage to
log2 fold changes resultsLFC <- lfcShrink(dds, coef = "condition_treated_vs_control", type = "apeglm") # View
summary summary(results)
```

Understanding Dispersion Shrinkage

Dispersion Plot: Before and After Shrinkage



Orange points represent initial gene-wise dispersion estimates (noisy, especially for low-count genes). **Red dashed line** shows the fitted trend across all genes. **Blue points** are the final maximum a posteriori (MAP) estimates after shrinkage toward the trend, providing more stable estimates.

📈 edgeR - Detailed Analysis

Overview

edgeR (empirical analysis of digital gene expression data in R) uses empirical Bayes methods to estimate biological variability. It's particularly well-suited for experiments with multiple factors and complex experimental designs. edgeR is known for its speed and flexibility in handling various study designs.

Statistical Model

edgeR also uses the negative binomial distribution but parameterizes it slightly differently:

$$Y \sim NB(\mu, \varphi)$$

NB

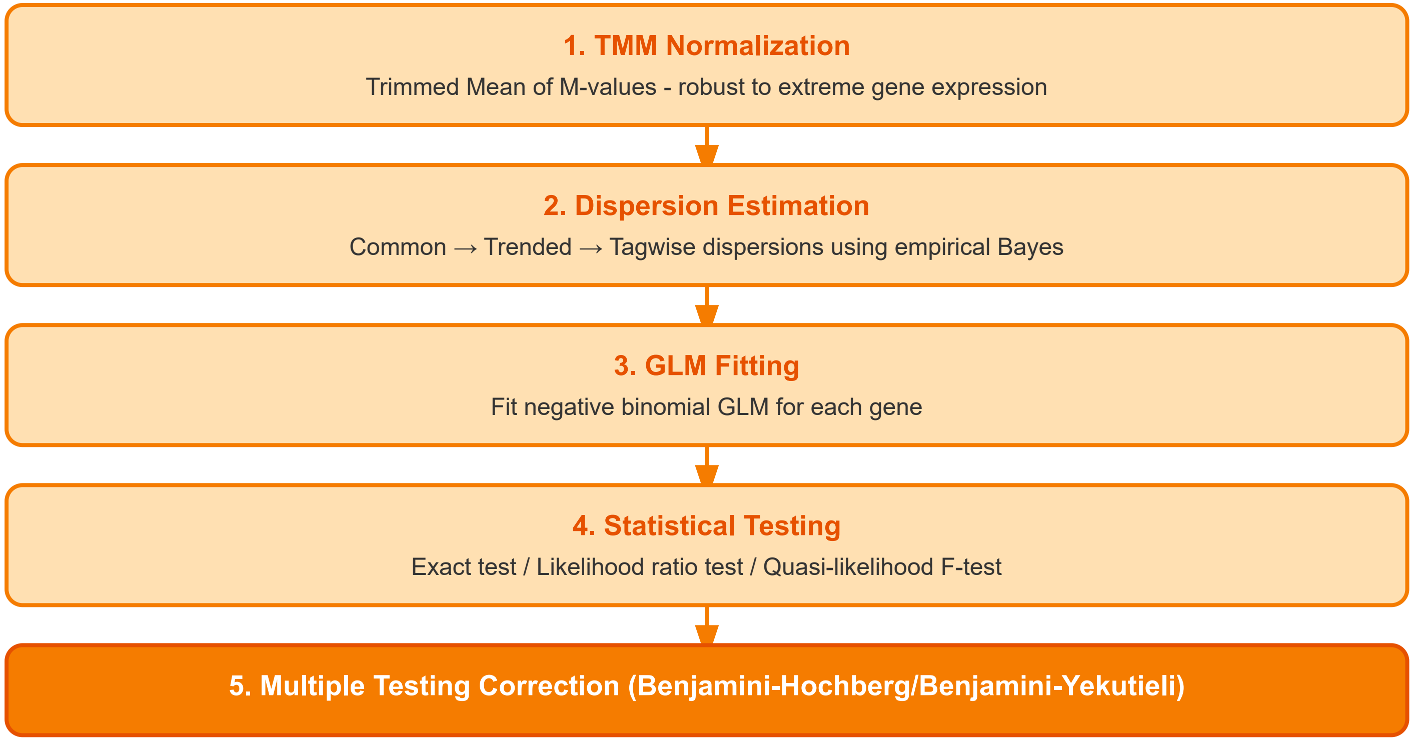
Y = Read counts for a gene

μ = Mean expression level

φ = Dispersion parameter (Variance = $\mu + \varphi\mu^2$)

edgeR models dispersion as a combination of common, trended, and tagwise (gene-specific) components, providing flexibility in capturing biological variability.

Analysis Workflow



1. TMM Normalization

Trimmed Mean of M-values - robust to extreme gene expression

2. Dispersion Estimation

Common → Trended → Tagwise dispersions using empirical Bayes

3. GLM Fitting

Fit negative binomial GLM for each gene

4. Statistical Testing

Exact test / Likelihood ratio test / Quasi-likelihood F-test

5. Multiple Testing Correction (Benjamini-Hochberg/Benjamini-Yekutieli)

Key Features & Advantages

- **TMM Normalization:** Trimmed Mean of M-values is particularly robust when a small proportion of genes are very highly expressed or when there are compositional differences between samples.
- **Flexible Dispersion Estimation:** Three-tier approach (common, trended, tagwise) allows capturing different levels of biological variability across genes.
- **Quasi-likelihood Framework:** The QL F-test provides better type I error control than likelihood ratio tests, especially for small sample sizes.

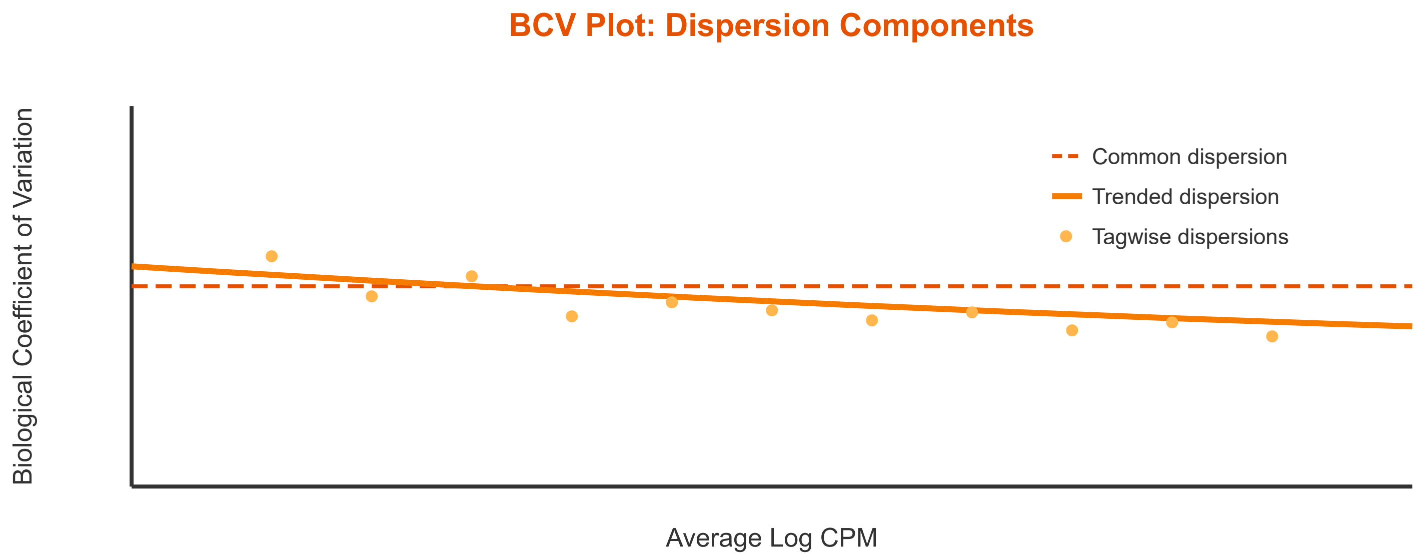
- **Exact Test:** For simple two-group comparisons, edgeR offers an exact test analogous to Fisher's exact test, which doesn't require asymptotic approximations.

- **Complex Design Support:** Excellent support for multi-factor experiments, paired designs, and batch effect correction through its GLM framework.

Example R Code

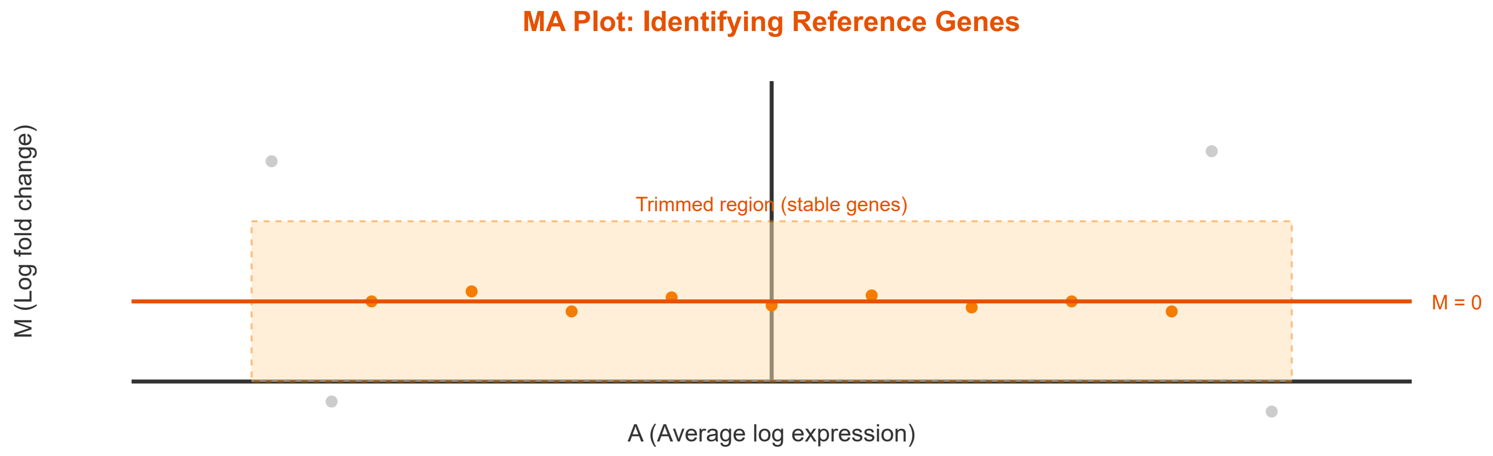
```
# Load library library(edgeR) # Create DGEList object y <- DGEList(counts = counts, group = group) # TMM
normalization y <- calcNormFactors(y, method = "TMM") # Design matrix design <- model.matrix(~ group) #
Estimate dispersions y <- estimateDisp(y, design) # Fit GLM fit <- glmQLFit(y, design) # Conduct quasi-
likelihood F-test qlf <- glmQLFTest(fit, coef = 2) # Extract results results <- topTags(qlf, n = Inf)
```

Understanding BCV (Biological Coefficient of Variation)



Red dashed line represents common dispersion (average across all genes). **Orange curve** shows trended dispersion (expression-dependent). **Orange points** are tagwise (gene-specific) dispersions that are moderated toward the trend using empirical Bayes.

TMM Normalization Principle



TMM trims extreme M-values and A-values (typically 30% and 10% respectively) before calculating the normalization factor. This makes it robust to genes with very high expression or extreme differential expression that could bias the normalization.

📉 limma-voom - Detailed Analysis

Overview

limma-voom transforms RNA-seq count data to log-counts per million (log-CPM) with associated precision weights. This allows the use of limma's linear modeling framework, which was originally developed for microarray data. The voom transformation estimates the mean-variance relationship and computes precision weights for each observation.

Statistical Model

Unlike DESeq2 and edgeR, limma-voom operates in log-space with a normal distribution:

$$\log_2(\text{CPM} + 0.5) \sim N(\mu, \sigma^2_i)$$

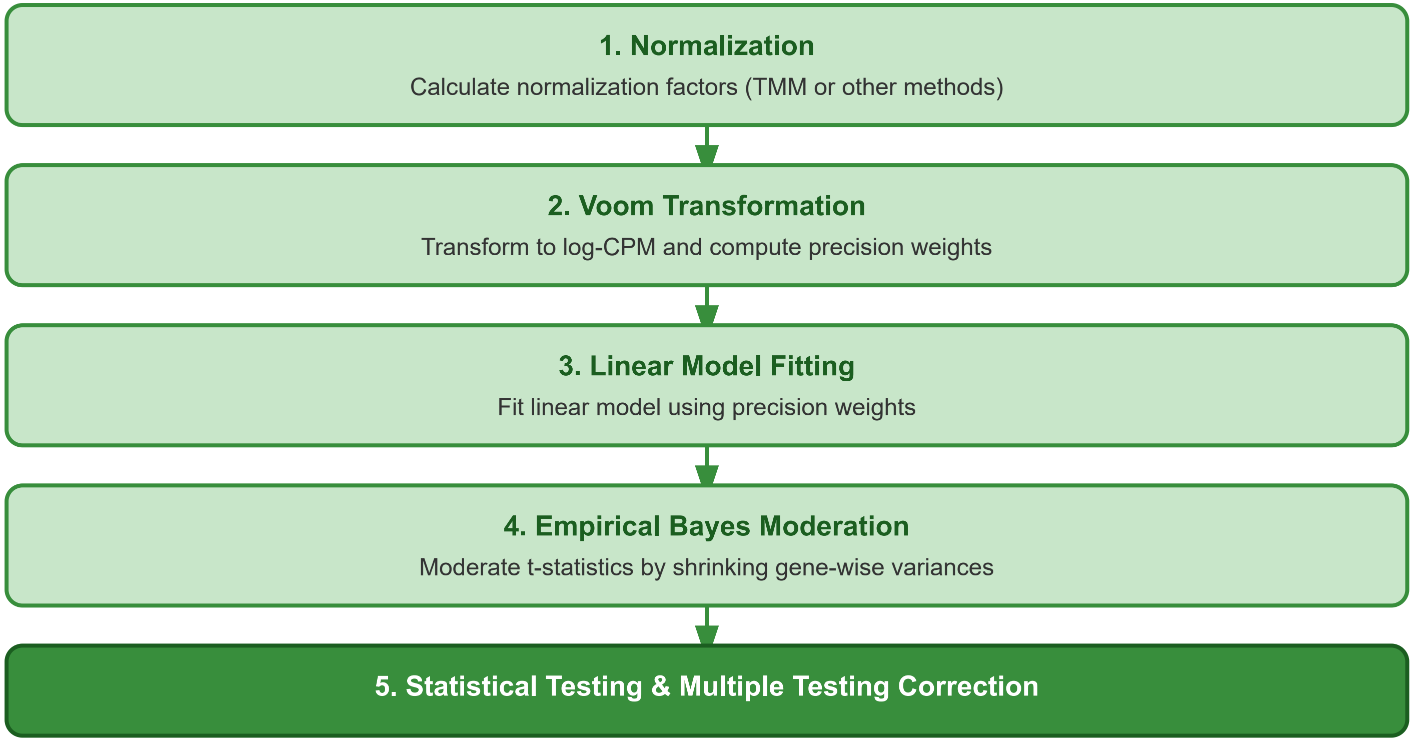
CPM = Counts per million (normalized counts)

μ = Mean log-expression for the gene

σ^2_i = Variance for observation i (varies by expression level)

Precision weights $w_i = 1/\sigma^2_i$ capture mean-variance relationship

Analysis Workflow



1. Normalization

Calculate normalization factors (TMM or other methods)

2. Voom Transformation

Transform to log-CPM and compute precision weights

3. Linear Model Fitting

Fit linear model using precision weights

4. Empirical Bayes Moderation

Moderate t-statistics by shrinking gene-wise variances

5. Statistical Testing & Multiple Testing Correction

Key Features & Advantages

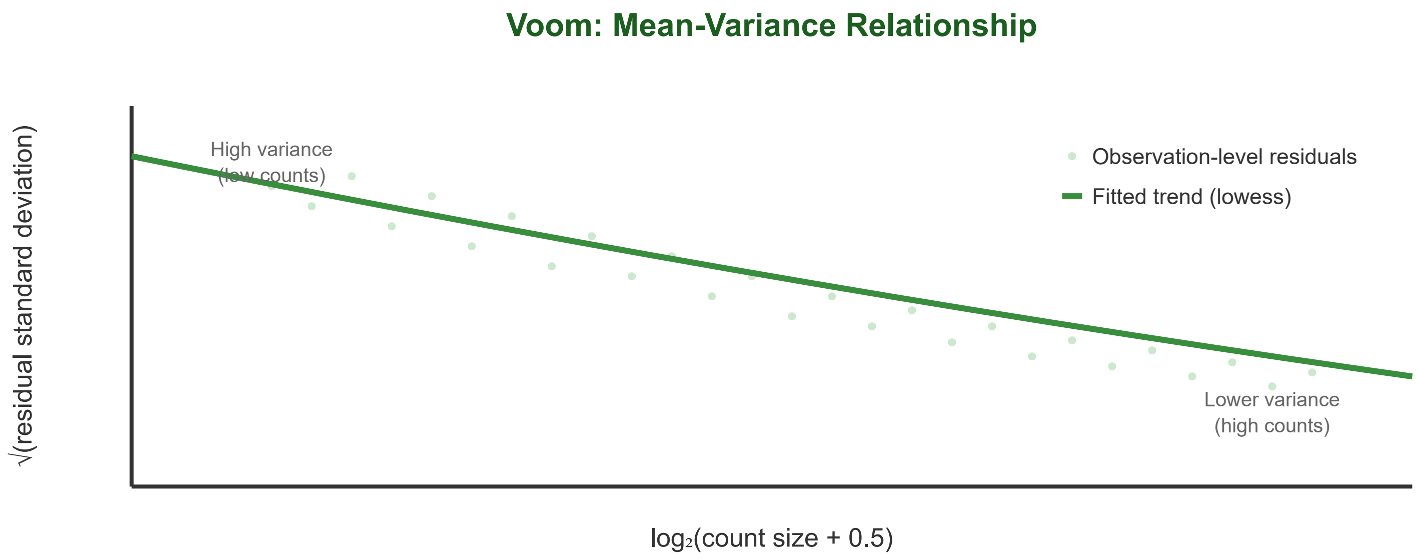
- **Log-space Analysis:** Working in log-space makes the data more normally distributed and allows using the mature statistical framework of linear models.
- **Precision Weights:** voom accurately models the mean-variance relationship by computing observation-specific precision weights, addressing heteroscedasticity in the log-transformed data.
- **Speed:** Linear modeling is computationally faster than iterative GLM fitting, making limma-voom particularly efficient for large datasets.

- **Empirical Bayes Moderation:** Borrows information across genes to moderate standard errors, improving statistical power especially with small sample sizes.
- **Flexible Contrasts:** Easy specification of complex contrasts and multiple comparisons within a single modeling framework.
- **Quality Weights:** Can incorporate sample-level quality weights (voomWithQualityWeights) to down-weight poor-quality samples automatically.

Example R Code

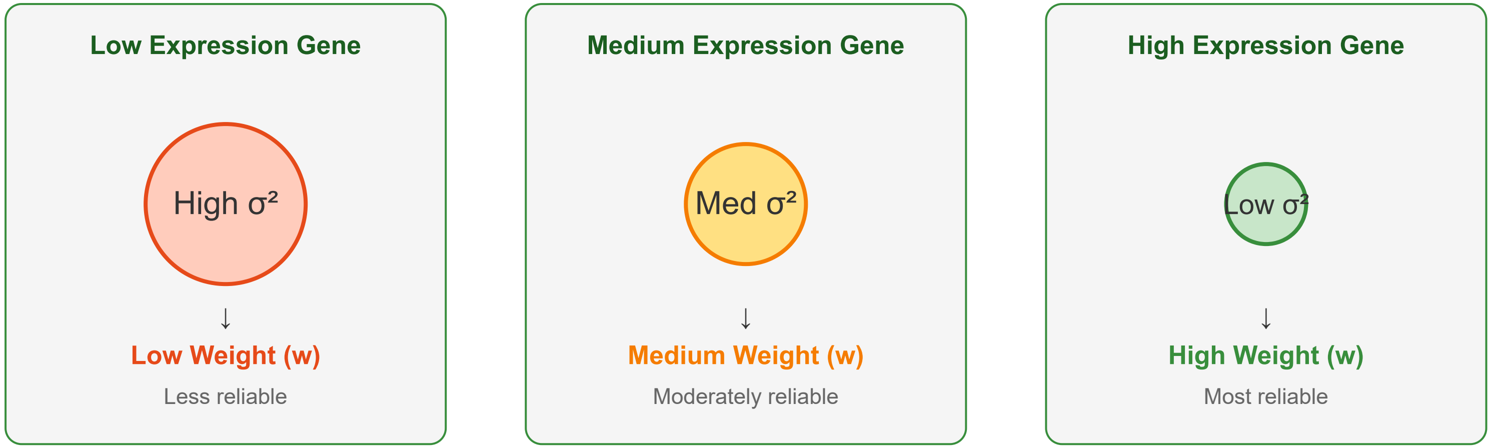
```
# Load libraries library(limma) library(edgeR) # Create DGEList and normalize dge <- DGEList(counts = counts)
dge <- calcNormFactors(dge, method = "TMM") # Design matrix design <- model.matrix(~ group) # Voom
transformation v <- voom(dge, design, plot = TRUE) # Fit linear model fit <- lmFit(v, design) # Empirical
Bayes moderation fit <- eBayes(fit) # Extract results results <- topTable(fit, coef = 2, number = Inf)
```

Understanding the Voom Mean-Variance Trend



The voom plot shows the square root of residual standard deviation versus average $\log_2$ count. The trend captures the mean-variance relationship, which is then used to calculate precision weights. **Low-count genes** (left) have higher variance and lower weights, while **high-count genes** (right) have lower variance and higher weights in the analysis.

How Precision Weights Work



Low Expression Gene

High σ²

↓

Low Weight (w)

Less reliable

Medium Expression Gene

Med σ²

↓

Medium Weight (w)

Moderately reliable

High Expression Gene

Low σ²

↓

High Weight (w)

Most reliable

Precision weight w = 1/σ^2. Genes with low expression have high variance (large circle) and receive low weights, meaning they contribute less to the statistical inference. High-expression genes have low variance (small circle) and receive high weights, making them more influential in the analysis. This appropriately accounts for the different reliability of measurements across the expression range.

📋 Method Comparison Summary

Feature	DESeq2	edgeR	limma-voom
Statistical Model	Negative Binomial GLM	Negative Binomial GLM	Linear model (log-space)
Normalization	Size factors (geometric mean)	TMM (Trimmed Mean of M-values)	TMM or other methods

Feature	DESeq2	edgeR	limma-voom
Dispersion Estimation	Shrinkage toward fitted trend	Common, trended, tagwise	Precision weights from mean-variance
Statistical Test	Wald test / LRT	Exact test / LRT / QL F-test	Moderated t-test
Best For	Small sample sizes (2-5 per group)	Complex designs, many samples	Large datasets, speed priority
Computation Speed	Moderate	Moderate	Fast
Memory Usage	Moderate	Low-Moderate	Low
Handling of Low Counts	Excellent (shrinkage)	Very good (empirical Bayes)	Good (precision weights)
Complex Designs	Good	Excellent	Excellent
Documentation	Extensive	Extensive	Extensive
Typical Use Case	Standard DE analysis, medical research	Multi-factor experiments, large studies	Large-scale screens, time-course

💡 Choosing the Right Method

Use DESeq2 if: You have small sample sizes (2-5 replicates), prioritize conservative estimates, or are following standard genomics workflows.

Use edgeR if: You have complex experimental designs with multiple factors, need flexible statistical tests, or want the most established NB-based approach.

Use limma-voom if: You have large datasets requiring fast computation, are comfortable with log-transformation assumptions, or want easy integration with microarray analysis pipelines.

General advice: All three methods produce similar results when used appropriately. The most important factors are proper experimental design, adequate biological replicates (≥3 per group), and appropriate quality control. Choose based on your specific needs and computational resources.

These methods are continuously updated. Always refer to the official documentation and recent literature for the latest recommendations.

Key References: Love et al. (2014) - DESeq2 | Robinson et al. (2010) - edgeR | Law et al. (2014) - limma-voom