DESeq2 Usage: PCA Analysis on RNA-seq Data

Oğuzhan Işılay

2025-12-30

What is PCA?

PCA (Principal Component Analysis) is a dimensionality reduction method used to make complex datasets easier to interpret. In RNA-seq experiments, thousands of genes are measured for each sample, making direct interpretation of the data difficult.

PCA summarizes this complex structure by projecting samples into a lower-dimensional space. Samples with similar gene expression profiles are positioned closer to each other, while samples with different profiles are separated. This allows us to visually assess whether groups such as control and treatment samples show overall differences in their expression patterns.

PCA is also useful for detecting unexpected patterns in the data. For example, if a sample appears far away from all others, this may indicate a technical issue, batch effect, or experimental problem.

It is important to note that PCA is not a statistical test. Instead, it is an exploratory and visualization tool that helps us understand the structure of the data before performing downstream analyses.

On Which Types of Data Can PCA Be Performed?

PCA cannot be applied meaningfully to every type of data. For a PCA analysis to be valid and interpretable, the dataset must meet several conditions:

The data must be numerical.

There must be multiple samples.

The data structure must be correct, where rows represent samples and columns represent variables (genes or measurements).

RNA-seq Data and PCA

Raw RNA-seq data consist of read counts, which show large differences in variance across genes. Because of this, RNA-seq count data are not directly suitable for PCA. Before applying PCA, the data must be transformed using an appropriate normalization or variance-stabilizing method.

What is DESeq2 and Why Is It Used for PCA?

DESeq2 is an R/Bioconductor package designed to analyze RNA-seq count data. Its primary purpose is to identify genes that are differentially expressed between experimental conditions.

However, DESeq2 is also widely used in PCA-based analyses because it provides transformations that make RNA-seq data suitable for visualization. One of the most important of these is the Variance Stabilizing Transformation (VST).

VST helps to:

balance variance between low and high count genes,

prevent PCA from being dominated by a small number of highly expressed genes,

allow more reliable comparisons between samples.

In this study, PCA is performed on RNA-seq data after applying the VST provided by DESeq2.

Can PCA Be Performed Without DESeq2?

Technically, PCA can be applied to any numerical dataset. However, when working with RNA-seq data, performing PCA without RNA-seq–specific preprocessing methods (such as those provided by DESeq2) often highlights technical variation rather than true biological differences.

For this reason, PCA in RNA-seq studies is typically performed after appropriate transformations using DESeq2, and is mainly used for quality control and sample similarity assessment.

#Dataset Used in This Study: airway

In this analysis, we use the airway dataset because:

it is derived from a real RNA-seq experiment,

it is small and well-structured, making it suitable for teaching,

both the count matrix and sample metadata are provided in a single package,

it includes clear control and treatment groups, making PCA interpretation straightforward.

Many bioinformatics curricula introduce the DESeq2 workflow using small example datasets like airway. This approach helps students focus on understanding core concepts such as counts, metadata, transformations, and PCA interpretation without being overwhelmed by large datasets.

Code: PCA Using DESeq2 and VST on the airway Dataset

The following code represents the core DESeq2 workflow. The same structure applies to other RNA-seq datasets: counts + coldata + design → DESeq2 → VST → PCA

```
#install.packages("BiocManager")

#BiocManager::install(c("airway", "DESeq2"))

# Load required packages
suppressPackageStartupMessages({
  library(airway)
  library(DESeq2)
})

# Load the airway dataset
data(airway)

# Extract count matrix (genes × samples)
counts <- assay(airway)

# Extract sample metadata
coldata <- colData(airway)

# Create DESeq2 dataset
dds <- DESeqDataSetFromMatrix(
  countData = counts,
  colData   = coldata,
  design    = ~ dex
)

# Run DESeq2 (normalization and model fitting)
dds <- DESeq(dds)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

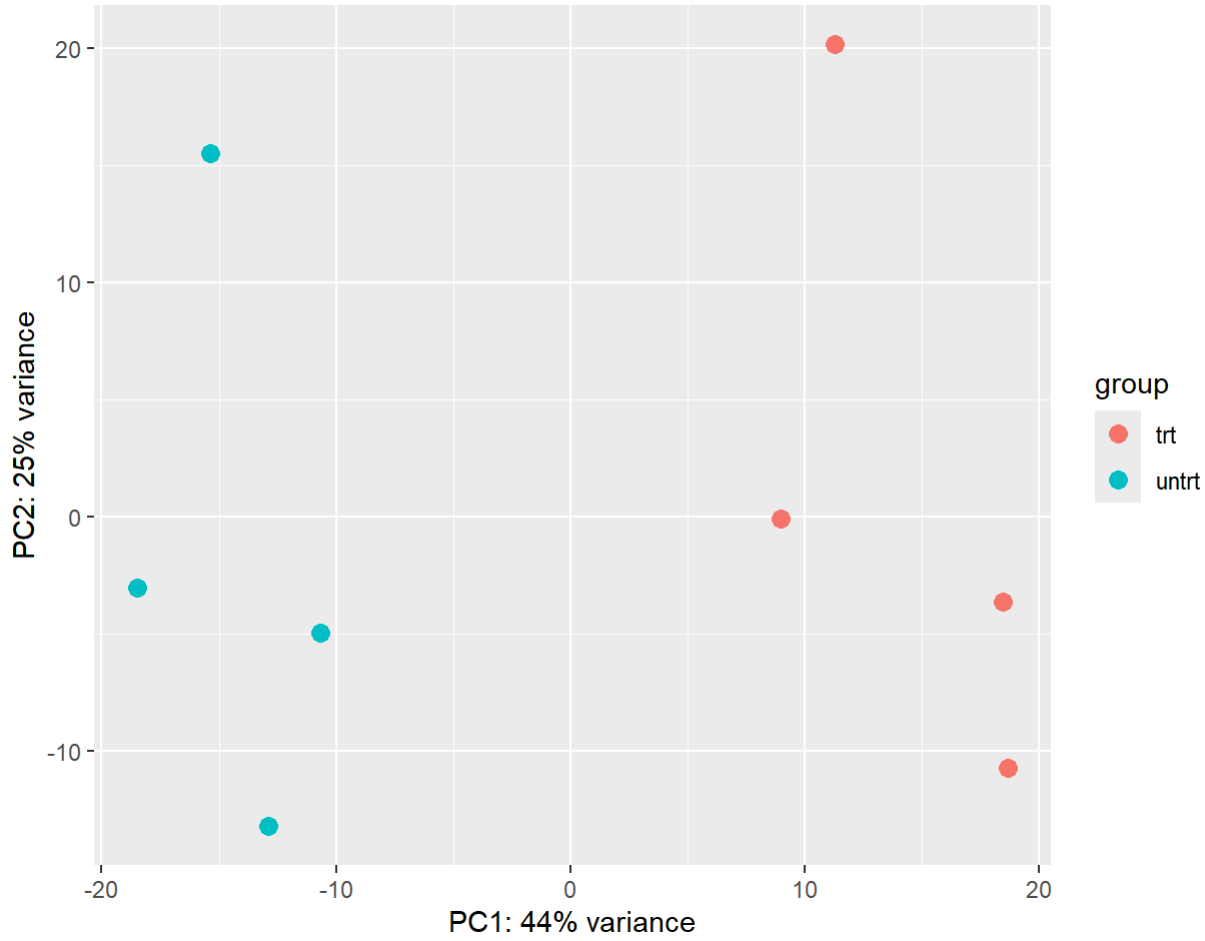
```
## final dispersion estimates
```

```
## fitting model and testing
```

```
# Apply variance stabilizing transformation
vsd <- vst(dds, blind = FALSE)

# Generate PCA plot
plotPCA(vsd, intgroup = "dex")
```

```
## using ntop=500 top features by variance
```



How to Interpret the PCA Plot

In the PCA plot, each point represents a single RNA-seq sample. The position of each point reflects the overall gene expression profile of that sample relative to others.

Understanding the Axes

PC1 (Principal Component 1) explains approximately 44% of the total variance in the data. This axis captures the largest source of variation between samples.

PC2 (Principal Component 2) explains about 25% of the variance and represents the second most important source of variation.

Together, PC1 and PC2 summarize roughly 69% of the total variance, which is sufficient for meaningful visualization.

Group Separation

Two groups are visible in the plot:

trt (treatment)

untrt (untreated / control)

Samples within the same group cluster closely together, indicating consistency among replicates. The separation between control and treatment samples suggests that the treatment induces a global change in gene expression.

What If We Used a Different Dataset?

If a different RNA-seq dataset were used instead of airway, the overall DESeq2 workflow would remain the same. The main difference would be the effort required to prepare the input data.

In real-world datasets, researchers often need to:

extract or construct the count matrix,

organize sample metadata,

define appropriate experimental groups.

For teaching purposes, using airway allows students to first understand the logic of the analysis before moving on to more complex and less curated datasets.