VIDO BIOINFO Bioinformatics and Systems Biology Lab

Canadian Bioinformatics Hub

Canadian Bioinformatics Workshops

Introduction to R Programming for Bioinformatics

Day 2- Module 4A: Biological Data Preprocessing and Visualization

Mohamed Helmy, PhD
Principal Scientist and Adjunct Professor
Bioinformatics and Systems Biology Lab
VIDO, University of Saskatchewan

6-7 October 2025, VIDO, Saskatoon

Learning Objectives

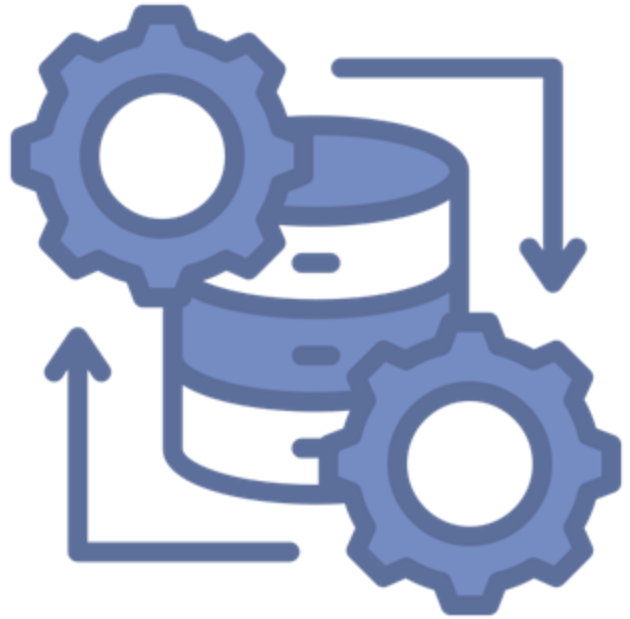
By the end of this module, we should have knowledge on:

- Why preprocessing and normalization matter
- DESeq2 workflow for preprocessing
- Exploratory visualization: PCA, clustering, heatmaps

Why preprocessing and normalization matter

- **Raw RNA-seq data is not ready for analysis**
  - Counts are affected by sequencing depth, gene length, and other technical factors.
  - Raw values can’t be compared directly across samples.

- **Preprocessing ensures data quality**
  - Detects and handles missing or low-quality values.
  - Makes downstream analysis more robust and reproducible.



- **Normalization is critical**
  - Adjusts for sequencing depth and composition bias.
  - Makes biological differences stand out from technical noise.

- **Takeaway**
  - Without preprocessing and normalization, results may reflect artifacts, not biology.

Normalization with DESeq2

- **Raw counts depend on:**
  - Sequencing depth
  - Gene length
  - Technical biases

- **Normalization makes samples comparable.**

- **To normalize using DESeq2**
  - Creates a DESeq2 object from airway data.
  - Normalizes counts for sequencing depth.

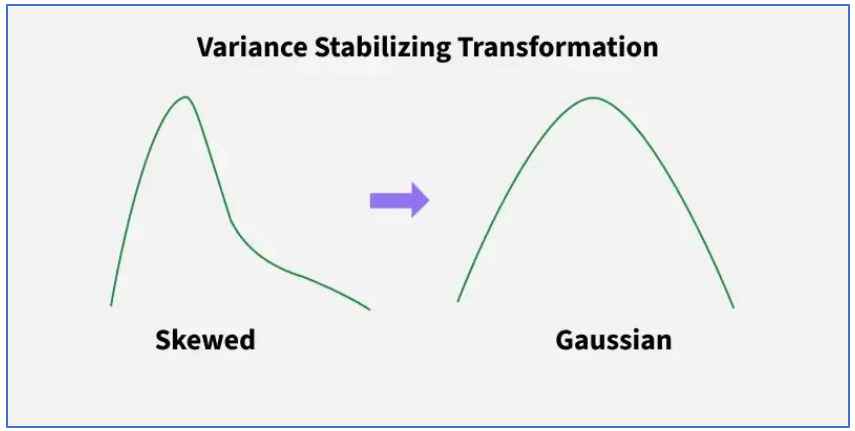
	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516	SRR1039517	SRR1039520	SRR1039521
ENSG00000000003	679	448	873	408	1138	1047	770	572
ENSG00000000005	0	0	0	0	0	0	0	0
ENSG00000000419	467	515	621	365	587	799	417	508
ENSG00000000457	260	211	263	164	245	331	233	229
ENSG00000000460	60	55	40	35	78	63	76	60
ENSG00000000938	0	0	2	0	1	0	0	0

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516	SRR1039517	SRR1039520	SRR1039521
ENSG00000000003	663	500	740	609	966	748	836	606
ENSG00000000005	0	0	0	0	0	0	0	0
ENSG00000000419	456	575	527	545	498	571	453	538
ENSG00000000457	254	235	223	245	208	237	253	242
ENSG00000000460	59	61	34	52	66	45	83	64
ENSG00000000938	0	0	2	0	1	0	0	0

```r
# install DESeq2
BiocManager::install("DESeq2")
# Load DESeq2
library(DESeq2)

# Create a DESeq2 dataset object
dds <- DESeqDataSet(airway, design = ~ dex)

# Run the DESeq pipeline
dds <- DESeq(dds)

# Raw and ormalized counts
# Normalized
norm_counts <- counts(dds, normalized=TRUE)
head(norm_counts)

# Raw
norm_counts <- counts(dds, normalized=FALSE)
head(norm_counts)
```

Variance Stabilizing Transformation (VST)

- **Why do we need it?**
  - RNA-seq data often shows mean–variance dependence:
    - Genes with high counts have higher variance.
    - Lowly expressed genes show noisy variability.
  - This makes downstream analyses (like PCA, clustering) biased by a few highly expressed genes.



- **What does VST do?**
  - Applies a mathematical transformation to make variance more uniform across expression levels.
  - Similar to log2 transform, but handles low counts more gracefully.
  - Produces values that are easier to interpret in exploratory visualizations.

Source: https://www.geeksforgeeks.org/data-science/variance-stabilizing-transformation/

How to use VST in DESeq2

- **Normalization adjusts for sequencing depth → makes samples comparable.**
- **VST adjusts variance → makes genes comparable.**
- **Together, they prepare data for exploratory visualization and robust downstream analysis.**

```
# How to use VST in DESeq2
# creates a new transformed dataset
vsd <- vst(dds, blind=FALSE) # takes into account the experimental design
# Us the transformed expression matrix
assay(vsd)[1:5, 1:5]
```

	SRR1039508	SRR1039509	SRR1039512	SRR1039513	SRR1039516
ENSG00000000003	9.554898	9.199610	9.695680	9.446121	10.044345
ENSG00000000005	5.517877	5.517877	5.517877	5.517877	5.517877
ENSG00000000419	9.087438	9.373174	9.263794	9.306185	9.195987
ENSG00000000457	8.407043	8.324342	8.265830	8.366469	8.192285
ENSG00000000460	7.073199	7.106145	6.723655	6.993144	7.162000

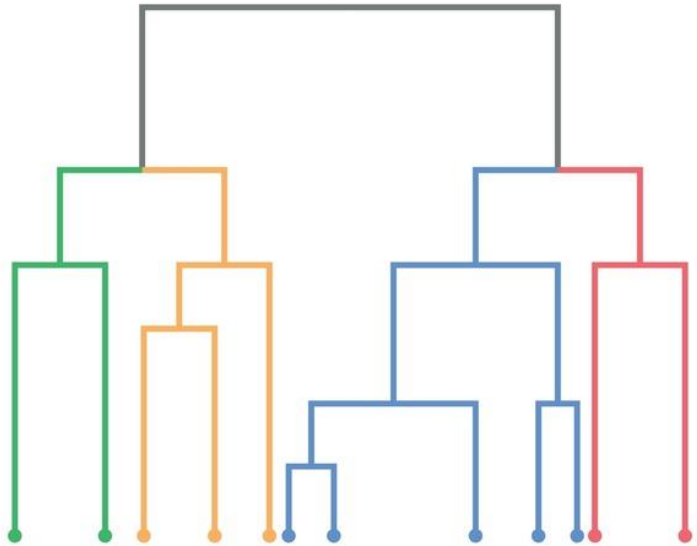
Sample Clustering (Dendrogram/Heatmap)

Why do we do sample clustering?

- **Helps evaluate similarities between samples.**
- **Reveals whether samples group by biological condition (treated vs untreated) or by technical effects (batch, sequencing run).**
- **Identifies potential outliers that don't cluster with their expected group.**

How does it work?

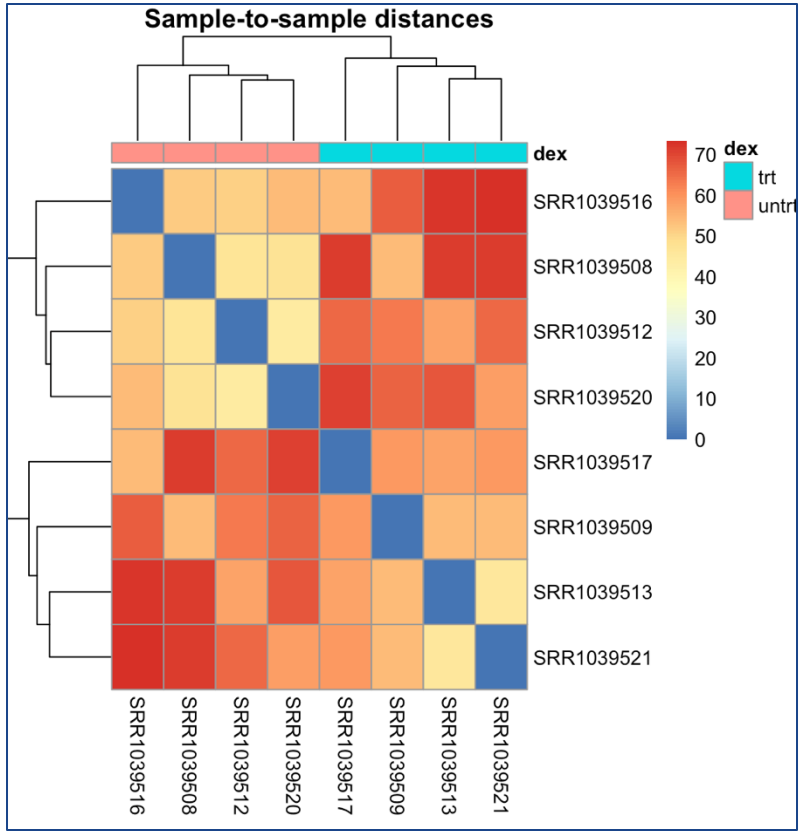
- **Compute distances between samples (e.g., Euclidean).**
- **Cluster samples based on their expression profiles.**
- **Visualize using a heatmap or dendrogram.**



Demo: Sample Clustering

- **Install the *pheatmap* package**
  - Use *BiocManager::install()*
- **Calculate sample-to-sample distances**
  - Use *dist()* and *t()*
- **Convert distances into a matrix**
  - Use *as.matrix()*
- **Heatmap of distances between samples**
  - Use *pheatmap()*

```
# install pheatmap
BiocManager::install("pheatmap", force = T)

# Load pheatmap
library(pheatmap)

# Calculate sample-to-sample distances
sampleDists <- dist(t(assay(vsd)))

# Convert distances into a matrix
sampleDistMatrix <- as.matrix(sampleDists)

# Heatmap of distances between samples
pheatmap(sampleDistMatrix,
         annotation_col = as.data.frame(colData(vsd)[, "dex", drop=FALSE]),
         main = "Sample-to-sample distances")
```



Principal Component Analysis (PCA) Plot

Why use PCA?

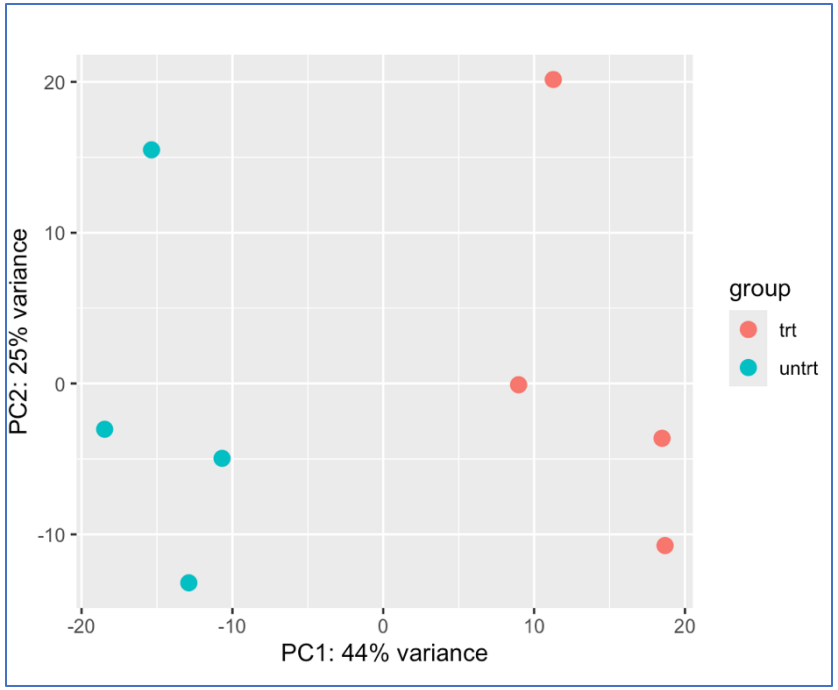
- **High-dimensional data (e.g., 20k genes) is hard to interpret.**
- **PCA reduces the data to a few dimensions (principal components)**
- **It captures the major sources of variation.**
- **Helps visualize how samples relate to each other in a 2D space.**

What can PCA tell us?

- **Whether treated vs untreated samples separate clearly.**
- **If there are batch effects driving variation.**
- **If outliers exist that don’t cluster with their group.**

Demo: PCA with DESeq2 VST Data

- **Use the VST object that we created earlier**
- **Use *plotPCA()***




```
# PCA with DESeq2 VST Data colored by treatment (dex)
plotPCA(vsd, intgroup="dex")
```

Hands-on Exercise

Student Tasks:

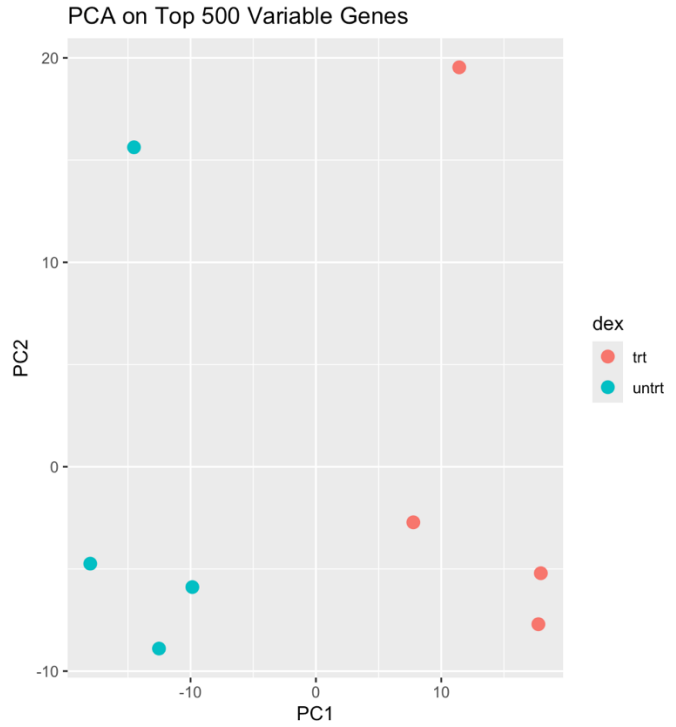
- **Variance check:**
  - Calculate the variance for each gene in the vsd object.
  - Identify the top 10 most variable genes.

- **Custom PCA:**
  - Perform PCA manually on the top 500 most variable genes (instead of all genes).
  - Plot the first 2 principal components using ggplot2, coloring samples by treatment (dex).

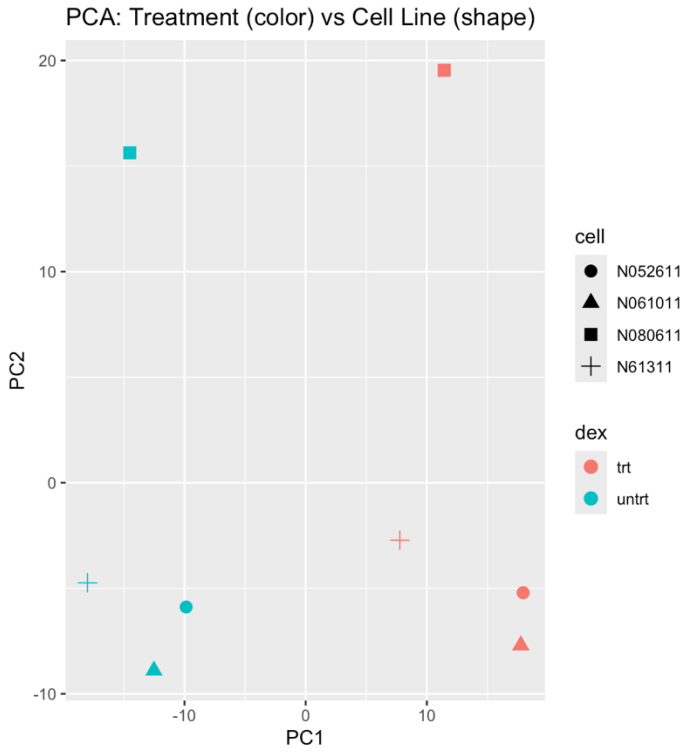
- **Challenge (optional):**
  - Add sample cell line (cell) as a shape aesthetic in the PCA plot.
  - Do treated vs untreated separate more clearly than the cell lines?

Hands-on Exercise

- **Top 10 genes**
  - "ENSG00000129824","ENSG00000229807","ENSG00000067048","ENSG00000114374","ENSG00000123243"
    "ENSG00000262902","ENSG00000012817","ENSG00000211445", "ENSG00000101347","ENSG00000109906"



Custom PCA on top 500 genes



Add cell line

THANK YOU

VIDO

VACCINE AND INFECTIOUS DISEASE ORGANIZATION

VIDO.ORG

UNIVERSITY OF SASKATCHEWAN