Author: Puja Sanjay Thorat
Language: Python 3.10+
Version: 1.0
Date Submitted: 12 October 2025

Description

This script performs a **complete differential gene expression (DGE) analysis** pipeline for tumor vs. normal samples using gene expression and annotation data.
It automates loading, cleaning, merging, and visualizing gene-level fold changes and chromosome distributions.
The script includes **robust QC checks, visualization of DEGs by chromosome**, and **heatmaps/clustermaps** for exploratory insights.

Input files required

a) Sample_Information.tsv
b) Gene_Expression_Data.xlsx
c) Gene_Information.csv

Dependencies

The following Python libraries are required:

Library	Purpose
numpy	Numerical computations
pandas	Data handling and manipulation
matplotlib	Data visualization

seaborn	Advanced plotting with aesthetic styles
openpyxl	Reading Excel (.xlsx) files
scikit-learn	Standardization for heatmap visualization (StandardScaler)

Install them via:

- pip install numpy pandas matplotlib seaborn openpyxl scikit-learn

---

How to Run the Script

Step 1: Organize Your Files

Place the following files in the **same directory** as the script:

- Sample_Information.tsv
- Gene_Expression_Data.xlsx
- Gene_Information.csv
- assignment_3.py

Step 2: Run the Script

Run the script using Python

The script automatically:

1. Loads all required data.
2. Maps sample IDs to phenotypes (tumor/normal).
3. Computes average expression per probe.
4. Calculates fold change between tumor and normal groups.
5. Identifies DEGs (|fold change| > 5).
6. Merges DEGs with gene annotation data.
7. Generates QC visualizations (chromosome distribution, DEG trends, heatmaps).

Output Files and Visualizations

The script generates the following **figures and outputs** during execution:

Output Type	Description
Chromosome Histogram	Distribution of DEGs across chromosomes.
Stacked Bar Chart (Tumor vs Normal)	Chromosomal breakdown of DEGs by expression trend.
Percentage Bar Plot	Percentage of DEGs upregulated in tumor vs normal samples.
Heatmap	Scaled heatmap of top variable or DEG probes.
Clustermap	Hierarchical clustering of selected probes across samples.
QC Console Logs	Detailed step-by-step outputs verifying data integrity and summary metrics.

(Optional: You can modify the script to save these plots as .png files or export tables as .csv.)

Notes & Recommendations

- Ensure that column names in all input files match exactly (Probe_ID, sample_id, etc.).
- The fold-change threshold for DEG detection (|fold_change| > 5) can be changed in **Step 7**.
- If no DEGs are detected, try lowering the threshold (e.g., |fold_change| > 2).
- Chromosome-based visualizations require a valid chromosome column in Gene_Information.csv.

Example Output Summary (Console)

After a successful run, you'll see a summary like this:

- FINAL SUMMARY & QC METRICS
- ----------
- Sample info shape: (30, 3)
- Expression matrix shape: (50000, 30)
- Gene info shape: (50000, 4)
- Common probes used for analysis: 49875
- DEGs (|FC|>5): 612
- Top chromosomes with DEGs:
- 1 84
- 12 52
- X 39
- DEG trend distribution:
- Tumor 340
- Normal 272

Files Created During Execution

By default, the script only produces **visualizations and console outputs**.
If you wish to save outputs, uncomment or add:

- deg.to_csv("Differentially_Expressed_Genes.csv", index=True)
- plt.savefig("DEG_Chromosome_Distribution.png")

This will save the DEG table and generated figures to your working directory.
