Gene Expression Analysis Script

Programmer: Puja Sanjay Thorat
Language: Python 3.10+
Date Submitted: 2025-10-10
Script Version: 1.0

Description

This Python script performs a complete analysis of gene expression data, from loading raw data files to identifying differentially expressed genes (DEGs) and generating exploratory visualizations. The script is designed to handle typical file format inconsistencies, compute fold changes between tumor and normal samples, and generate QC plots including chromosome distribution, DEG trends, and heatmaps/clustermaps of selected probes.

Required Files

The following input files must be uploaded to the working directory:

1. **Sample_Information.tsv** – Contains sample metadata, including sample IDs, phenotypes, and patient IDs.
2. **Gene_Expression_Data.xlsx** – Contains gene expression data, with probes as rows and sample IDs as columns.
3. **Gene_Information.csv** – Contains gene annotation data, including probe IDs and optional chromosome information.

Required Libraries / Packages

The script requires the following Python libraries:

- `numpy`
- `pandas`
- `matplotlib`
- `seaborn`
- `openpyxl`
- `sklearn` (for StandardScaler)
- `warnings` (standard Python library)

Make sure all packages are installed. You can install missing packages using `pip`, for example:

```
pip install numpy pandas matplotlib seaborn openpyxl scikit-learn
```

Usage Instructions

1. Open Google Colab or any Python environment.
2. Upload the three required files (`Sample_Information.tsv`, `Gene_Expression_Data.xlsx`, `Gene_Information.csv`).
3. Run the script. The script will automatically:
    - Load and QC the input files.
    - Map sample IDs to phenotypes and rename expression columns.
    - Split expression data into tumor and normal groups.
    - Compute average expression per probe for each group.
    - Calculate fold-change for tumor vs. normal.
    - Merge fold-change values with gene annotation.
    - Identify differentially expressed genes (DEGs) with absolute fold change > 5.
    - Generate QC plots including chromosome distributions, DEG trends, heatmaps, and clustermaps.
4. Monitor outputs printed to the console for warnings or QC information.

Note: Ensure that sample IDs in `Gene_Expression_Data.xlsx` match those in `Sample_Information.tsv`. Extra whitespace will be automatically handled.

Output Files and Objects

While this script mainly generates in-memory data frames and plots, the following outputs are created during execution:

1. **DataFrames:**
    - `sample_info` – Cleaned sample metadata.
    - `gene_expr` – Gene expression matrix with renamed columns.
    - `tumor_df` and `normal_df` – Split expression matrices.
    - `tumor_avg` and `normal_avg` – Average expression per probe.
    - `fold_change` – Computed fold-change values.
    - `merged` – Fold-change merged with gene annotation.
    - `deg` – Differentially expressed genes with fold-change > 5 and expression trend.
2. **Plots:**
    - Chromosome distribution of DEGs (if chromosome info is available).

    - Stacked histogram of DEGs by expression trend.
    - Bar plot of percentage of DEGs upregulated in tumor vs. normal.
    - Heatmap and clustermap of top 50 DEGs (or top 50 most variable probes if <50 DEGs).

Optional: Users may export `deg` or `merged` to CSV for downstream analysis.

Additional Notes

- The script includes robust handling for missing or malformed headers, missing chromosome data, and zero values in normal samples.
- If chromosome information is missing, alternative plots based on gene symbols will be used.
- Designed to run in Google Colab, but should work on any local Python environment with the required packages installed.