Task
For: Project Research Scientist-II post

Submitted by: Sarita Maurya
Date- 27/2/2025

- Data summary (sample count, gene statistics, variability)
- Differential expression analysis (Wilcoxon/Kruskal-Wallis)
- Heatmap visualization
- Hierarchical clustering
- PCA with explained variance
- Machine learning model (Random Forest)
- Gene-gene co-expression network

Install and load required packages in Jupyter Noteook

- import pandas as pd
- import numpy as np
- import seaborn as sns
- import matplotlib.pyplot as plt
- from scipy.stats import kruskal, ttest_ind
- from sklearn.decomposition import PCA
- from sklearn.cluster import KMeans
- from sklearn.manifold import TSNE
- from sklearn.ensemble import RandomForestClassifier
- from sklearn.model_selection import train_test_split
- from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
- from scipy.stats import spearmanr
- import seaborn as sns
- import networkx as nx
- import warnings

Data Loading and Summary

1. Load the TCGA dataset and provide a summary including:

The number of samples per cancer type

The mean and standard deviation of expression levels for each gene

The top 5 most variable genes across all samples

The number of samples per cancer type:

Cancer type: ACC BLCA BRCA CESC COAD DLBC GBM HNSC KICH KIRC KIRP LAML LGG LIHC LUAD LUSC OV PRAD READ SKCM STAD THCA UCEC UCS

Numer of sample:79 414 1119 306 483 48 170 504 66 542 291 178 532 374 541 502 430 501 167 472 420 513 554 57

S.N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Cancer type	ACC	**BLCA**	BRCA	CESC	COAD	DLBC	GBM	HNSC	KICH	KIRP	KIRC	LAML	LGG	LIHC	LUAD	LUSC	OV	PRAD	READ	SKCM	**STAD**	THCA	UCEC	UCS
Number of samples	79	414	1119	306	483	48	170	504	66	542	291	178	532	374	541	502	430	501	167	472	420	513	554	57

Summary of Datasets

```
]: # Step 1: Summary of Dataset
print("Number of samples per cancer type:")
print(data['Cancertype'].value_counts())
```

```
Number of samples per cancer type:
Cancertype
BRCA    1119
UCEC     554
KIRC     542
LUAD     541
LGG      532
THCA     513
HNSC     504
LUSC     502
PRAD     501
COAD     483
SKCM     472
OV       430
STAD     420
BLCA     414
LIHC     374
CESC     306
KIRP     291
LAML     178
GBM      170
READ     167
ACC       79
KICH      66
UCS       57
DLBC      48
Name: count, dtype: int64
```

The mean and standard deviation(sd) of expression levels for each gene

```
# Compute mean and standard deviation for each gene
gene_stats = data.drop(columns=['Sample', 'Cancertype']).agg(['mean', 'std']).T
gene_stats = gene_stats.sort_values(by='std', ascending=False)
top5_variable_genes = gene_stats.head(5)
print("Top 5 most variable genes:")
print(top5_variable_genes)
```

```
Top 5 most variable genes:
                 mean          std
UBB       7330.653614  4205.813034
HIST1H1C  1955.368684  3579.582470
UBC       5190.153905  2556.609357
PFN1      3298.034454  2039.389104
DAXX      4895.071901  1628.726865
```

- > print(top5_variable_genes)
- Gene mean sd
- <chr> <dbl> <dbl>
- 1) UBB 7331 4206.
- 2) HIST1H1C 1955 3580
- 3) UBC 5190 255
- 4) PFN1 3298 2039
- 5) DAXX 4895 1629
- Variance Explained by First Two PCs:"
- > print(explained_variance)
- PC1 PC2
- 0.37367 0.18524

2. Identify the top 10 differentially expressed genes between 5 cancer types using a t-test or Wilcoxon rank-sum test. Visualize the expression of these genes with a heatmap. (you can make multiple visualization plots)

Load and preprocess the dataset

Check for missing values & filter genes

Perform statistical tests (t-test & Wilcoxon)

Identify the top 10 differentially expressed genes

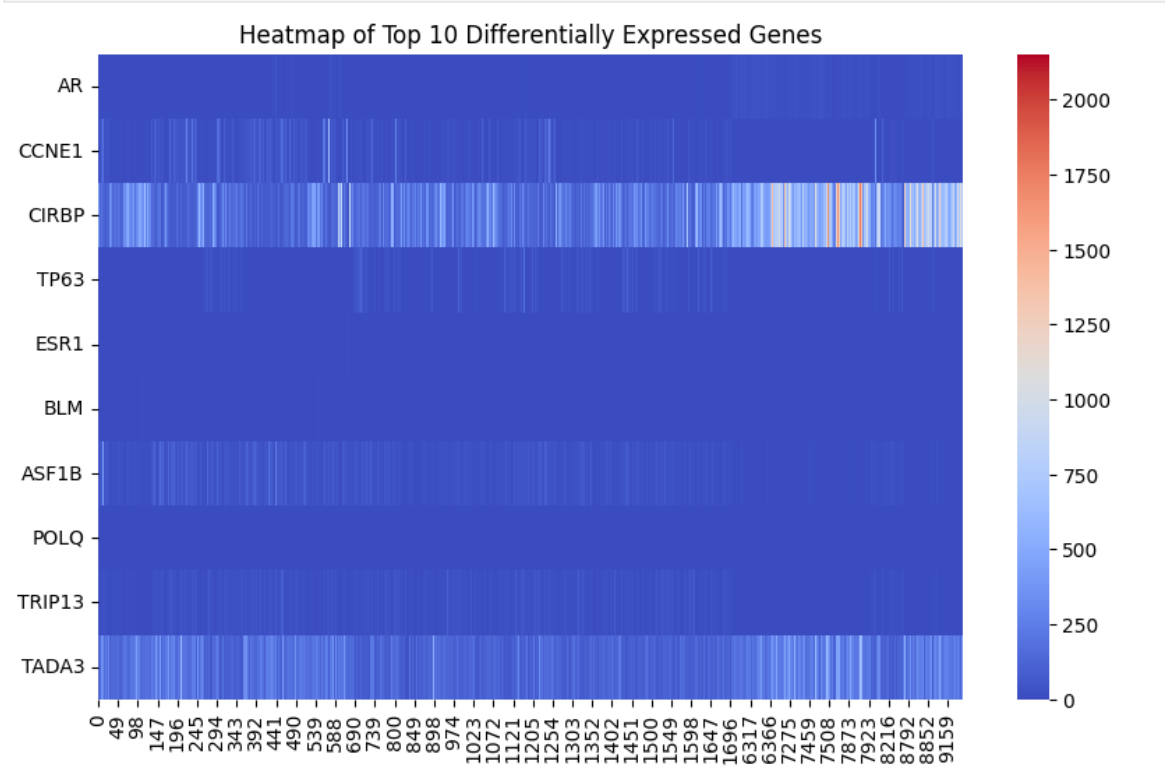
Visualize expression levels with a heatmap

- **The top 5 most variable genes across all samples:** "BRCA" "UCEC" "KIRC" "LUAD" "LGG"

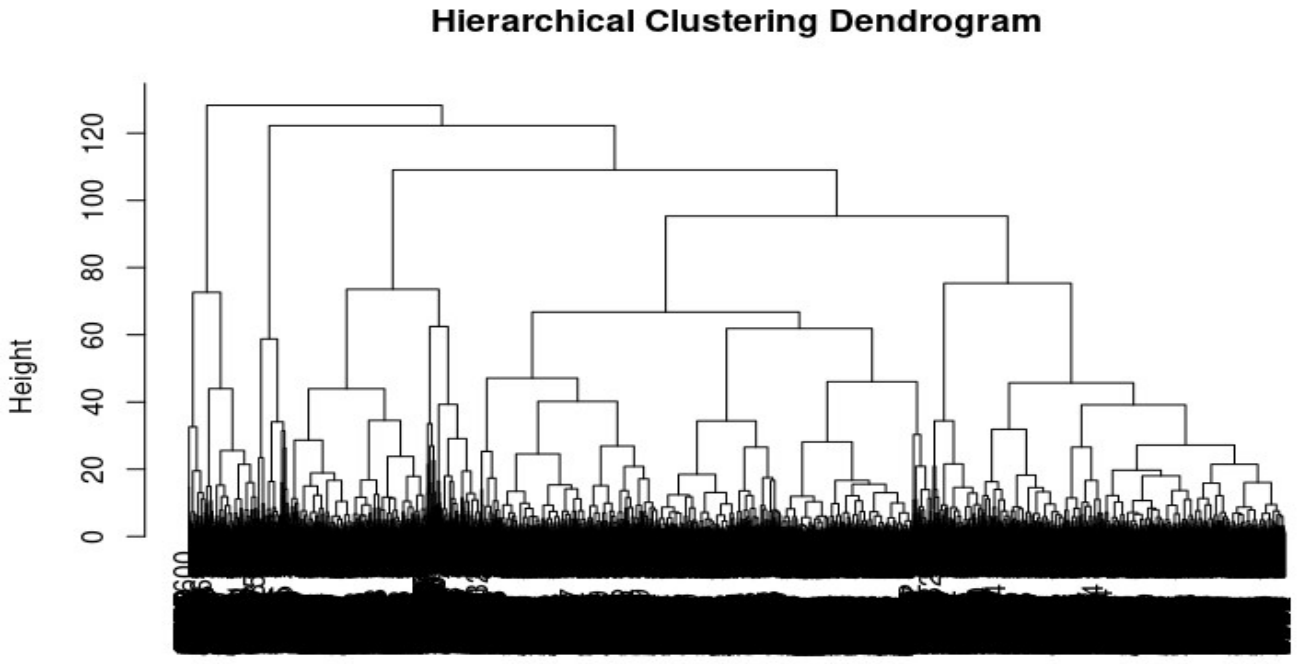
2. Identify the top 10 differentially expressed genes between 5 cancer types using a t-test or Wilcoxon rank-sum test. Visualize the expression of these genes with a heatmap. (you can make multiple visualization plots)

top 10 differentially expressed genes between 5 cancer types: ABCC1" "ACTL6A" "ANKRD44" "ASF1B" "BCL2" "BRCC3" "C19orf40" "CALM3" "CCNB1" "CCNB2"

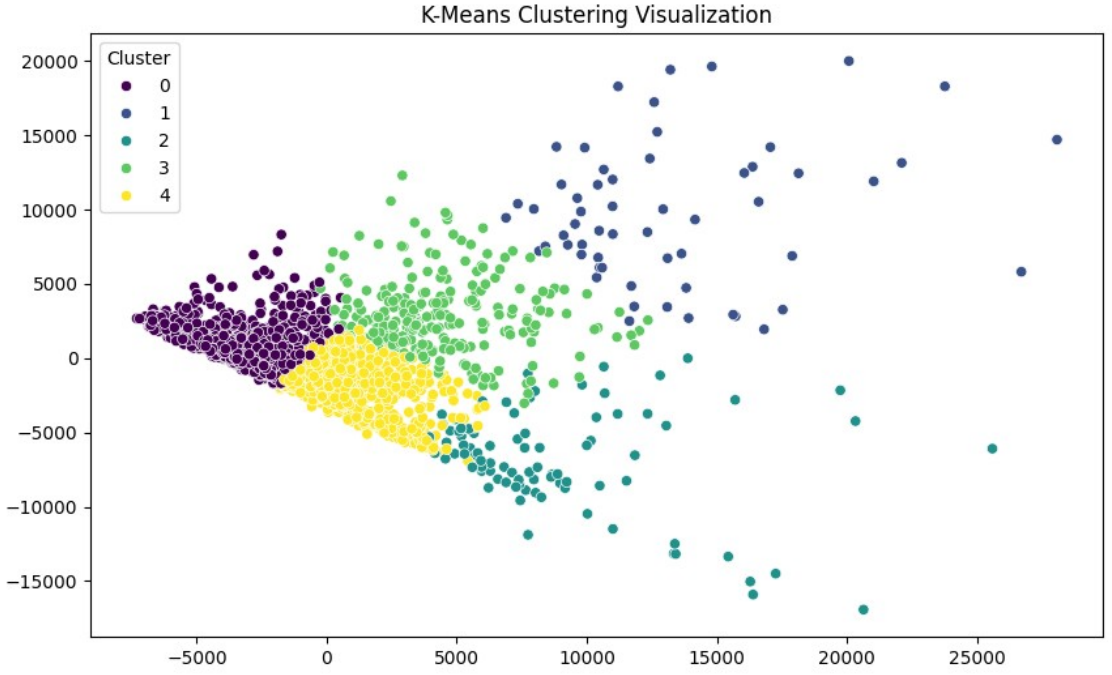
Visualize expression levels with a heatmap



Hierarchical Clustering Dendrogram



K-Means Clustering Visualization



PCA Plot of Gene Expression

Step 5: Machine Learning Model

```python
# Step 5: Machine Learning Model
X_train, X_test, y_train, y_test = train_test_split(expr_matrix, filtered_data['Cancertype'], test_size=0.2, random_state=42)
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred = rf_model.predict(X_test)


print("Classification Report:")
print(classification_report(y_test, y_pred))
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Collapse Output

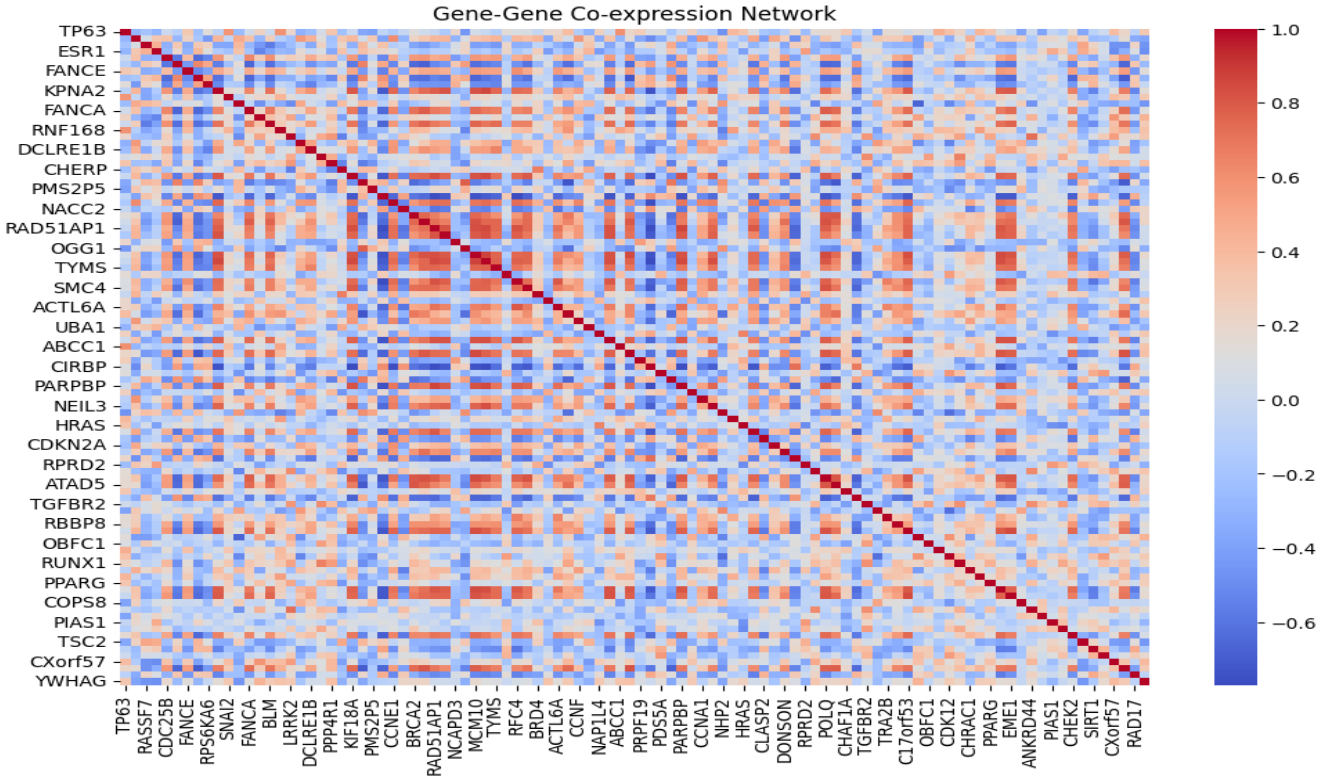
```
Classification Report:
              precision    recall  f1-score   support

         GBM       1.00      0.97      0.99        40
        LUAD       0.97      0.93      0.95       120
        LUSC       0.90      0.96      0.93        81
          OV       1.00      1.00      1.00        88
        PRAD       1.00      0.99      0.99       100

    accuracy                           0.97       429
   macro avg       0.97      0.97      0.97       429
weighted avg       0.97      0.97      0.97       429

Confusion Matrix:
[[ 39   0   1   0   0]
 [  0 112   8   0   0]
 [  0   3  78   0   0]
 [  0   0   0  88   0]
 [  0   1   0   0  99]]
```

Step 6: Feature Importance

```
# Step 6: Feature Importance
gene_importance = pd.Series(rf_model.feature_importances_, index=expr_matrix.columns)
top_features = gene_importance.nlargest(10)
print("Top Predictive Genes:")
print(top_features)
```

```
Top Predictive Genes:
TP63       0.030070
RBM38      0.017062
RASSF7     0.016649
ESR1       0.016091
CDC25B     0.013162
SETMAR     0.013042
FANCE      0.013026
AR         0.012858
RPS6KA6    0.012529
KPNA2      0.012270
dtype: float64
```

Gene-Gene Co-expression Network



BonusQuestion:

If you had access to mutation, methylation, and copy number variation data, how would you integrate these with gene expression to better classify cancer types? Design a workflow for this.

- Workflow for Integrating Multi-Omics Data to Classify Cancer Types:
- Step 1: Data Collection & Preprocessing
- Obtain Data:
- Gene Expression: RNA-Seq counts or normalized TPM/FPKM/RPKM.
- Mutation Data: Variant allele frequencies (VAF), mutation types (missense, nonsense, etc.).
- Methylation Data: β-values from Illumina Infinium 450K or EPIC arrays.
- CNV Data: Log2 fold change values from SNP arrays or sequencing.

- **2) Quality Control (QC):**

- Remove low-quality samples and batch effects (ComBat, SVA).
- Normalize gene expression (log2(TPM + 1) or z-score transformation).
- Filter lowly expressed genes and hypermethylated promoters.
- **3) Feature Selection:**
- Gene Expression: Select differentially expressed genes (DEGs) using Kruskal-Wallis or LIMMA.
- Mutation Data: Keep genes with high mutation frequency (e.g., TP53, KRAS, PIK3CA).
- Methylation Data: Identify differentially methylated regions (DMRs) using DSS or ChAMP.
- CNV Data: Extract high-amplitude gains/losses (GISTIC2.0).

Step 2: Multi-Omics Integration

- Create a Multi-Modal Feature Matrix
- Dimensionality Reduction (Optional)

Apply Principal Component Analysis (PCA) or t-SNE/UMAP to visualize data.

Feature selection via LASSO, Random Forest, or Boruta.

Step 3: Machine Learning Model Training

- Train a Classifier to Predict Cancer Types

- Traditional ML: Random Forest, XGBoost, SVM.
- Deep Learning: Multi-omics Graph Neural Networks (GNNs) or Multi-Modal Transformers.
- Cross-Validation & Hyperparameter Tuning

- Use stratified k-fold cross-validation.
- Tune hyperparameters with GridSearchCV or Bayesian Optimization.

Step 4: Model Evaluation & Interpretation

- Performance Metrics

- Accuracy, F1-score, AUC-ROC, Precision-Recall Curve.
- Biological Interpretation

- Use SHAP (SHapley Additive exPlanations) to interpret feature importance.
- Identify driver genes that contribute most to classification.

Step 5: Network Analysis & Biomarker Discovery

- Gene Regulatory Networks

- Use WGCNA (Weighted Gene Co-expression Network Analysis) to find co-expressed modules.
- Build protein-protein interaction (PPI) networks with STRING or BioGRID.
- Pathway Enrichment Analysis

- GO/KEGG Pathway Analysis (using clusterProfiler in R).
- Find significant pathways related to cancer subtypes.

Step 6: Validation & Clinical Applications

-
- Validate Model on Independent Cohorts

- Use external datasets (TCGA, METABRIC, ICGC, GEO).
- Check performance in pan-cancer analysis.
- Clinical Utility

- Develop a multi-omics biomarker panel for precision oncology.
- Predict therapy response (e.g., Immunotherapy, Chemotherapy).

Final Thoughts

This workflow ensures a comprehensive multi-omics approach to cancer classification, capturing genetic, epigenetic, and expression-level variations. By integrating mutation, methylation, and CNV data, we can improve subtype identification, biomarker discovery, and precision medicine applications.

Thank You