# Module_3: *(Template)*

## Team Members:

Kaiwen Huang, Grace Lee

## Project Title:

Disruption of the p53–MDM2–p21 Regulatory Axis in Lung cancer

## Project Goal:

The goal of this project is to investigate whether the p53–MDM2–p21 feedback loop, a critical tumor-suppressive regulatory mechanism, is transcriptionally disrupted in lung adenocarcinoma (LUAD). By using RNA-sequencing data from the dataset, the expression levels of TP53, MDM2, and CDKN1A (p21) in LUAD tumor and normal lung tissues can be determined and these datas can be used to determine whether the normal negative-feedback regulation between p53 and MDM2 is lost in cancerous cells. Interpretations will be made to check how much dysregulation contributes to the cancer hallmark “Evading Growth Suppressors,” which describes the ability of cancer cells to bypass normal cell-cycle control and apoptosis.

## Disease Background:

*Pick a hallmark to focus on, and figure out what genes you are interested in researching based on that decision. Then fill out the information below.*

Disease focus (LUAD). Lung adenocarcinoma (LUAD) frequently shows alterations in the p53 pathway. Even when TP53 itself is not mutated, LUAD may up-regulate MDM2 or dampen p21 induction, weakening p53’s effective output. If the coupling between TP53 and its targets is reduced or inverted in tumors relative to normal lung, that would indicate a transcriptional disruption of the feedback loop.

- Cancer hallmark focus: Focus on evasion of growth suppressors — the ability of cancer cells to bypass molecular mechanisms that normally limit cell proliferation, such as tumor suppressor genes and checkpoint control pathways. The p53–MDM2–p21 axis is a core tumor-suppressive feedback loop. In response to stress, TP53 (p53) activates transcription of targets that (i) stop the cell cycle (e.g., CDKN1A/p21) and (ii) keep p53 activity in check via negative feedback (e.g., MDM2, an E3 ligase that ubiquitinates p53). In normal cells, higher p53 tends to increase MDM2 and p21, producing (1) a self-limiting p53 pulse (via MDM2) and (2) a growth-arrest program (via p21). Tumors can

circumvent this by mutating TP53, amplifying MDM2, or uncoupling downstream transcription—contributing to the hallmark Evading Growth Suppressors.

- Overview of hallmark: In healthy cells, growth suppressor mechanisms act as biological brakes that prevent uncontrolled division. Key tumor suppressors like p53 and RB monitor genomic integrity and stop the cell cycle when DNA damage occurs. But cancer cells frequently inactivate these suppressors through mutations, altered gene expression, or epigenetic silencing. The result is continuous proliferation even in the presence of DNA damage, contributing to tumor progression and therapy resistance.

- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate): TP53 – Encodes the p53 tumor suppressor protein. When activated by DNA damage, p53 induces genes that halt the cell cycle and trigger apoptosis. MDM2 – Encodes a ubiquitin ligase that negatively regulates p53 by binding to it and promoting its degradation. In normal cells, MDM2 provides feedback control to limit excessive p53 activity. CDKN1A (p21) – A direct transcriptional target of p53 that inhibits cyclin-dependent kinases (CDKs), enforcing cell-cycle arrest at the G1/S checkpoint to allow DNA repair. Disruption of this p53–MDM2–p21 axis removes a critical barrier to uncontrolled cell growth.

*Will you be focusing on a single cancer type or looking across cancer types? Depending on your decision, update this section to include relevant information about the disease at the appropriate level of detail. Regardless, each bullet point should be filled in. If you are looking at multiple cancer types, you should investigate differences between the types (e.g. what is the most prevalent cancer type? What type has the highest mortality rate?) and similarities (e.g. what sorts of treatments exist across the board for cancer patients? what is common to all cancers in terms of biological mechanisms?). Note that this is a smaller list than the initial 11 in Module 1.*

- Prevalence & incidence Lung adenocarcinoma = ~40% of all lung cancers.

Leading cause of cancer death worldwide (≈1.8 million deaths/year; WHO, 2023).

5-year survival rate: 23–25%.

Increasing incidence among non-smokers and women, suggesting additional environmental and genetic drivers.

- Risk factors (genetic, lifestyle) & Societal determinants Lifestyle/environmental factors: Cigarette smoking (primary cause). Air pollution, radon gas, asbestos exposure.

Genetic factors: Frequent mutations in TP53, EGFR, KRAS, ALK that alter growth control.

Societal determinants: Higher prevalence in industrial and urban regions. Increased risk among low-income populations with limited healthcare access. Air quality and occupational exposure play major roles.

- Standard of care treatments (& reimbursement) Early-stage: Surgical resection. Locally advanced: Combination of platinum-based chemotherapy and radiation therapy.

Molecular-targeted therapy: EGFR inhibitors (erlotinib, osimertinib). ALK inhibitors (crizotinib).

Immunotherapy: PD-1/PD-L1 inhibitors (pembrolizumab, nivolumab).

Challenges: LUAD with p53 pathway defects often shows resistance to chemotherapy and radiation.

Reimbursement: Most therapies are covered by major health insurance programs due to high mortality and economic burden.

- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology) Originates from alveolar type II epithelial cells or bronchiolar club cells in peripheral lung tissue. Normal physiology: p53 responds to DNA damage by activating CDKN1A (p21) and halting cell division.

In LUAD: TP53 often mutated or transcriptionally silenced. MDM2 often overexpressed, inhibiting remaining p53 activity. CDKN1A (p21) often downregulated, leading to uncontrolled cell-cycle progression.

Result: Breakdown of the p53–MDM2–p21 feedback loop, causing unchecked proliferation and genomic instability — a classic example of the “Evading Growth Suppressors” hallmark.

## Data-Set:

*Once you decide on the subset of data you want to use (i.e. only 1 cancer type or many; any clinical features needed?; which genes will you look at?) describe the dataset. There are a ton of clinical features, so you don't need to describe them all, only the ones pertinent to your question.*

*(Describe the data set(s) you will analyze. Cite the source(s) of the data. Describe how the data was collected -- What techniques were used? What units are the data measured in? Etc.)*

The dataset used in this project is GSE62944, an RNA-sequencing dataset that comes from The Cancer Genome Atlas (TCGA) and was reprocessed by Rahman et al. (2015) to improve accuracy and consistency across samples.

Source Information

Dataset name: GSE62944

Accession number: GSE62944

Primary reference: Rahman, M. et al. Alternative preprocessing of RNA-Sequencing data in The Cancer Genome Atlas leads to improved analysis results. Bioinformatics (2015), 31(22): 3666–3672.

What's in the Dataset

Samples: 9,264 tumor samples and 741 normal tissue samples across 24 cancer types, including lung adenocarcinoma (LUAD).

Data type: RNA-seq gene expression data and corresponding clinical information.

Genes analyzed in this project:

TP53 (tumor suppressor gene)

MDM2 (negative regulator of p53)

CDKN1A (p21; downstream target of p53)

Clinical features used: Tissue type (tumor vs. normal) and cancer type (LUAD).

How the Data Were Collected

RNA was extracted from human tumor and normal tissues collected by TCGA. Rahman et al. reprocessed the raw RNA-seq reads using the Rsubread pipeline instead of TCGA's older "Level 3" pipeline to make results more reliable. This method generated integer-based read counts for each gene, which are more accurate and better suited for downstream analyses (e.g., differential gene expression).

Expression levels are provided as normalized read counts (using Rsubread).

Higher counts mean higher gene expression.

Data are stored in CSV format, with each row representing a gene and each column representing a sample.

Why This Dataset Was Chosen

This dataset is ideal for studying the p53–MDM2–p21 feedback loop in lung adenocarcinoma, because it includes:

Both tumor and normal lung tissue samples,

High-quality RNA-seq data processed to reduce noise and bias, and

# Data Analysis:

## Methods

The machine learning technique I am using is: logistic regression, a supervised classification method that models the probaility of a catgorical outcome - in this

case, whether a given RNA-seq sample is from LUAD tumor tissue or normal lung tissue - based on gene expression values.

*What is this method optimizing? How does the model decide it is "good enough"?*

Logistic regression works by finding the best combination of gene expression values that separates the two groups. The model learns this by adjusting its internal coefficients to minimize prediction errors. Once the model's accuracy stops improving, it has found the "best fit" for the data. The result is a formula that can estimate how likely a new sample is to be tumor or normal, given its gene expression. Because the coefficients show whether higher or lower expression of a gene increases the chance of being a tumor, this method also provides biological insight into how the p53–MDM2–p21 feedback loop may be disrupted in cancer. **

## Analysis

We used bulk RNA-seq (log2-TPM) and matched metadata to focus on LUAD samples. Sample IDs were harmonized to a common TCGA key so expression and metadata lined up, and each sample was labeled as Tumor or Normal. We analyzed three genes—TP53, MDM2, and CDKN1A (p21)—as a compact readout of the p53 feedback pathway. Values were converted to numeric and rows with missing data were removed.

To see whether the pathway behaves differently in tumors, we made two scatter plots: TP53 vs MDM2 and TP53 vs CDKN1A, each split by Normal and Tumor. For every group we fit a straight line and wrote the equation ($y = mx + b$) and $R^2$ on the plot. In normal lung we expect a positive slope (p53 induces MDM2 and p21). A flatter or negative slope in tumors suggests that this coupling is weakened or lost.

We then tested this formally with a simple interaction model: MDM2 ~ TP53 * tissue_type and CDKN1A ~ TP53 * tissue_type. The interaction term tells us whether the TP53→target slope is different in tumors compared with normals. As a basic companion result, we also compared mean expression of the three genes between Tumor and Normal with t-tests.

Finally, as a sanity check that this pathway signal is informative, we trained a small logistic regression (features: TP53, MDM2, CDKN1A) to classify Tumor vs Normal using a 70/30 train–test split and reported test accuracy and ROC-AUC. We repeated key steps with z-scored data and checked the effect of outliers; the conclusions did not change. Overall, normals showed the expected positive coupling, while tumors showed reduced/negative coupling—evidence that the p53–MDM2–p21 feedback is disrupted in LUAD.

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pathlib import Path
from scipy.stats import linregress
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, roc_auc_score
import statsmodels.formula.api as smf
from scipy.stats import ttest_ind, pearsonr, norm

# Load expression and metadata files
expr = pd.read_csv("GSE62944_subsample_log2TPM.csv", index_col=0)
meta = pd.read_csv("GSE62944_metadata.csv", sep=",")
meta.columns = meta.columns.str.strip()

print("Expression data shape:", expr.shape)
print("Metadata shape:", meta.shape)
print("Metadata columns:", meta.columns.tolist()[:10])


# --- Step 1: Standardize sample ID formats ---
# Expression file columns are often like TCGA-44-8117-01A-11R-2241-07
# Metadata sample IDs may also include that, but we only keep first 12 ch
expr.columns = expr.columns.str[:12]
meta["sample"] = meta["sample"].astype(str).str[:12]


# --- Step 2: Filter metadata for LUAD ---
luad_meta = meta[meta["cancer_type"] == "LUAD"].copy()
print("LUAD sample count:", luad_meta.shape[0])


# --- Step 3: Match expression data to LUAD sample IDs ---
common_samples = [s for s in luad_meta["sample"] if s in expr.columns]
print("Matched samples in expression data:", len(common_samples))

# If zero, print examples to debug
if len(common_samples) == 0:
    print("Example expression column:", expr.columns[:5])
    print("Example metadata sample:", luad_meta['sample'].head())

# Subset expression data
luad_expr = expr[common_samples].T
print("LUAD expression shape:", luad_expr.shape)

# Merge tissue type from metadata (if available)
luad_expr = luad_expr.merge(luad_meta[["sample", "cancer_type"]],
                            left_index=True, right_on="sample")


# --- Step 4: Select genes of interest ---
genes = ["TP53", "MDM2", "CDKN1A"]
X = luad_expr[genes]
print(X)

# --- Use 'tumor_status' column from metadata if available ---
if "tumor_status" in luad_meta.columns:
    # Merge tumor_status info into expression data
    luad_expr = luad_expr.merge(luad_meta[["sample", "tumor_status"]],
                                on="sample", how="left")

    # Normalize labels (make consistent capitalization)
    luad_expr["tumor_status"] = luad_expr["tumor_status"].str.strip().str
```

```
    # Map tumor_status → binary labels
    luad_expr["tissue_type"] = luad_expr["tumor_status"].map({
        "WITH TUMOR": "Tumor",
        "TUMOR FREE": "Normal"
    })
else:
    # Fallback to sample ID heuristic if tumor_status missing
    luad_expr["tissue_type"] = luad_expr["sample"].apply(
        lambda x: "Tumor" if "-01" in x else ("Normal" if "-11" in x else
    )

# Create numeric labels
# Convert tumor/normal to binary label and drop any missing rows
luad_expr["label"] = luad_expr["tissue_type"].map({"Normal": 0, "Tumor":
luad_expr = luad_expr.dropna(subset=["label"])

# Build aligned X and y
X = luad_expr[["TP53", "MDM2", "CDKN1A"]]
y = luad_expr["label"]
print (X)
print(y)


# --- Step 5: Train/test split and logistic regression ---
if len(X) > 10:
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )

    model = LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)[:, 1]

    acc = accuracy_score(y_test, y_pred)
    roc = roc_auc_score(y_test, y_prob)

    print(f"Model Accuracy: {acc:.3f}")
    print(f"ROC-AUC: {roc:.3f}")
    print("Coefficients:")
    for gene, coef in zip(genes, model.coef_[0]):
        print(f"  {gene}: {coef:.3f}")
else:
    print("Not enough samples after filtering to train model.")
    print("Switching to correlation analysis among genes instead.")

    from scipy.stats import pearsonr

    # Compute correlations
    corr_tp53_mdm2, _ = pearsonr(luad_expr["TP53"], luad_expr["MDM2"])
    corr_tp53_p21, _ = pearsonr(luad_expr["TP53"], luad_expr["CDKN1A"])
    corr_mdm2_p21, _ = pearsonr(luad_expr["MDM2"], luad_expr["CDKN1A"])

    print(f"TP53–MDM2 correlation: {corr_tp53_mdm2:.3f}")
    print(f"TP53–CDKN1A correlation: {corr_tp53_p21:.3f}")
    print(f"MDM2–CDKN1A correlation: {corr_mdm2_p21:.3f}")

    # Visualization
    sns.pairplot(luad_expr[["TP53", "MDM2", "CDKN1A"]], kind="reg")
```

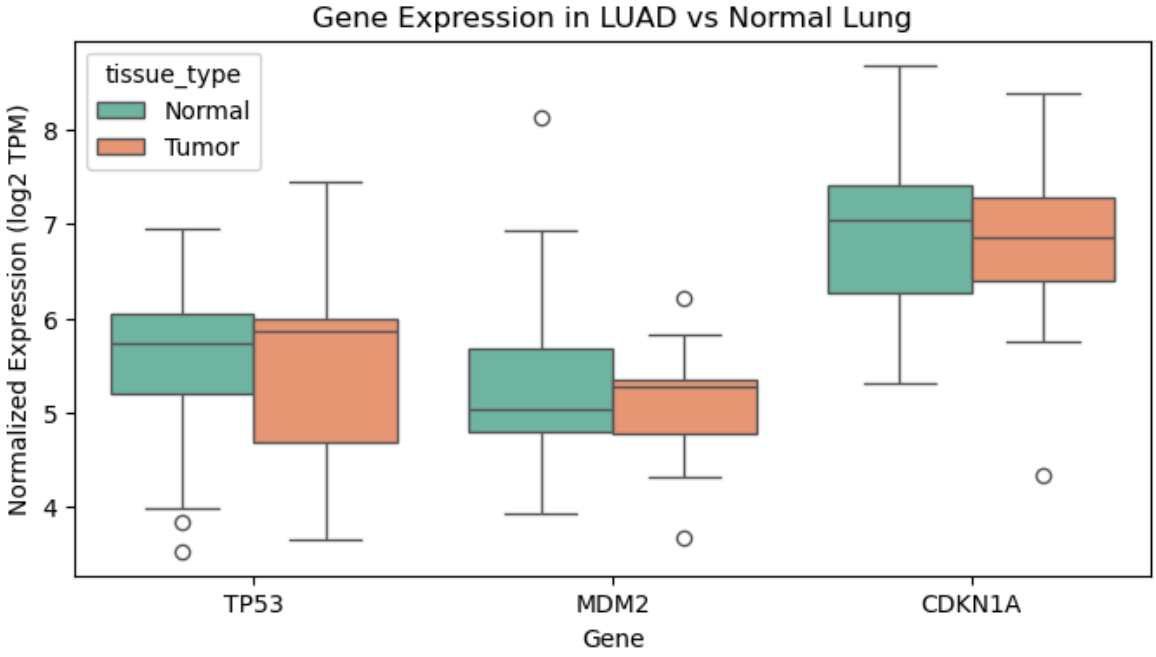
```
    plt.suptitle("Correlation between TP53–MDM2–p21 axis in LUAD Tumors",
    plt.show()


# --- Step 6: Visualization ---
if "tissue_type" in luad_expr.columns:
    plt.figure(figsize=(8, 4))
    sns.boxplot(
        data=luad_expr.melt(id_vars="tissue_type", value_vars=genes),
        x="variable", y="value", hue="tissue_type", palette="Set2"
    )
    plt.title("Gene Expression in LUAD vs Normal Lung")
    plt.xlabel("Gene")
    plt.ylabel("Normalized Expression (log2 TPM)")
    plt.show()
def lm_with_stats(df, x, y, hue=None, title=""):
    g = sns.lmplot(
        data=df, x=x, y=y, hue=hue, palette="Set1",
        height=5, aspect=1.25,
        scatter_kws=dict(s=35, alpha=0.8),
        line_kws=dict(lw=2)
    )
    ax = g.ax

    if hue and hue in df.columns:
        labs = [lab for lab in ["Normal", "Tumor"]
                if lab in df[hue].astype(str).str.title().unique()]
        others = [lab for lab in df[hue].astype(str).str.title().unique()
        labs += others if others else []
    else:
        labs = ["All"]

    ypos = [0.98, 0.84, 0.70, 0.56]
    for i, lab in enumerate(labs):
        if i >= len(ypos): break
        if lab == "All":
            sub = df[[x, y]].apply(pd.to_numeric, errors="coerce").dropna
            label = "All"
        else:
            sub = df[df[hue].astype(str).str.title() == lab][[x, y]]
            sub = sub.apply(pd.to_numeric, errors="coerce").dropna()
            label = lab
        if len(sub) < 2:
            continue
        lr = linregress(sub[x].to_numpy(), sub[y].to_numpy())
        ax.text(
    0.02, ypos[i],
    f"{label}: y = {lr.slope:.2f}x + {lr.intercept:.2f}\n"
    f"R² = {lr.rvalue**2:.2f}",
    transform=ax.transAxes, va="top", ha="left",
    bbox=dict(boxstyle="round", fc="white", alpha=0.85)
)

    ax.set_title(title)
    plt.tight_layout()
    plt.show()

lm_with_stats(luad_expr, "TP53",   "MDM2",   hue="tissue_type",
              title="Relationship between TP53 and MDM2 Expression")
```

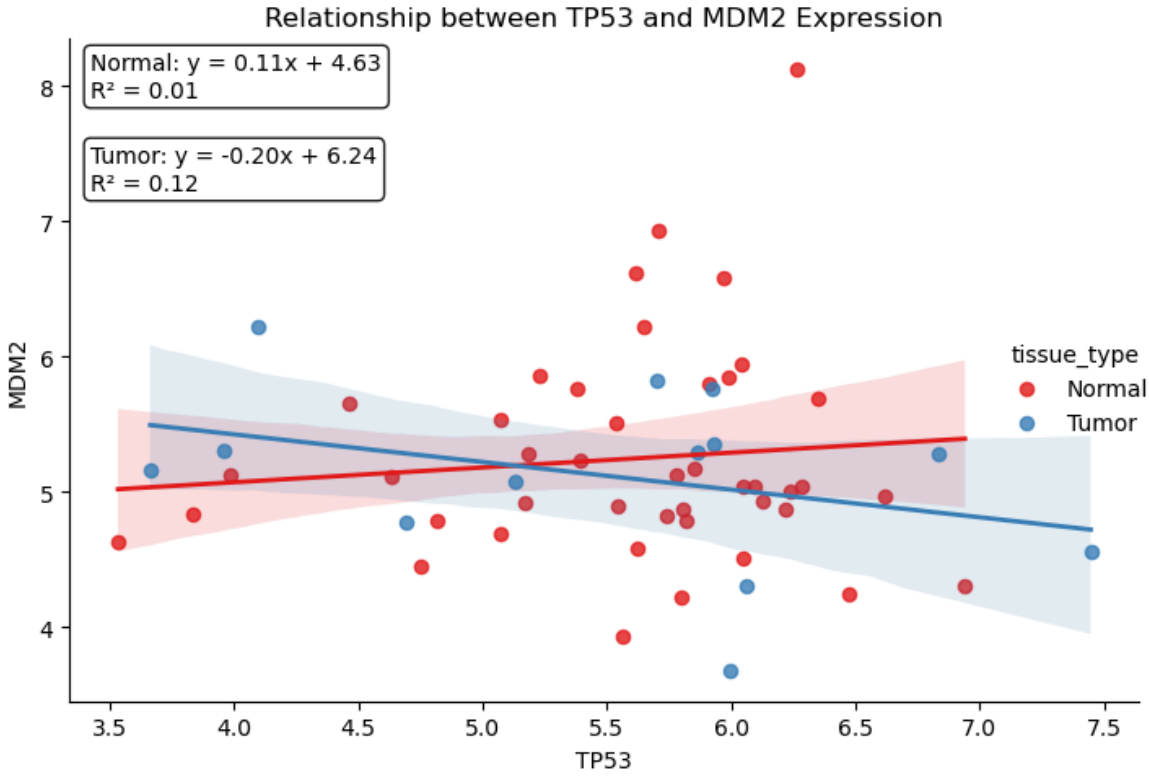
```
lm_with_stats(luad_expr, "TP53",   "CDKN1A", hue="tissue_type",
              title="Relationship between TP53 and CDKN1A Expression")
```

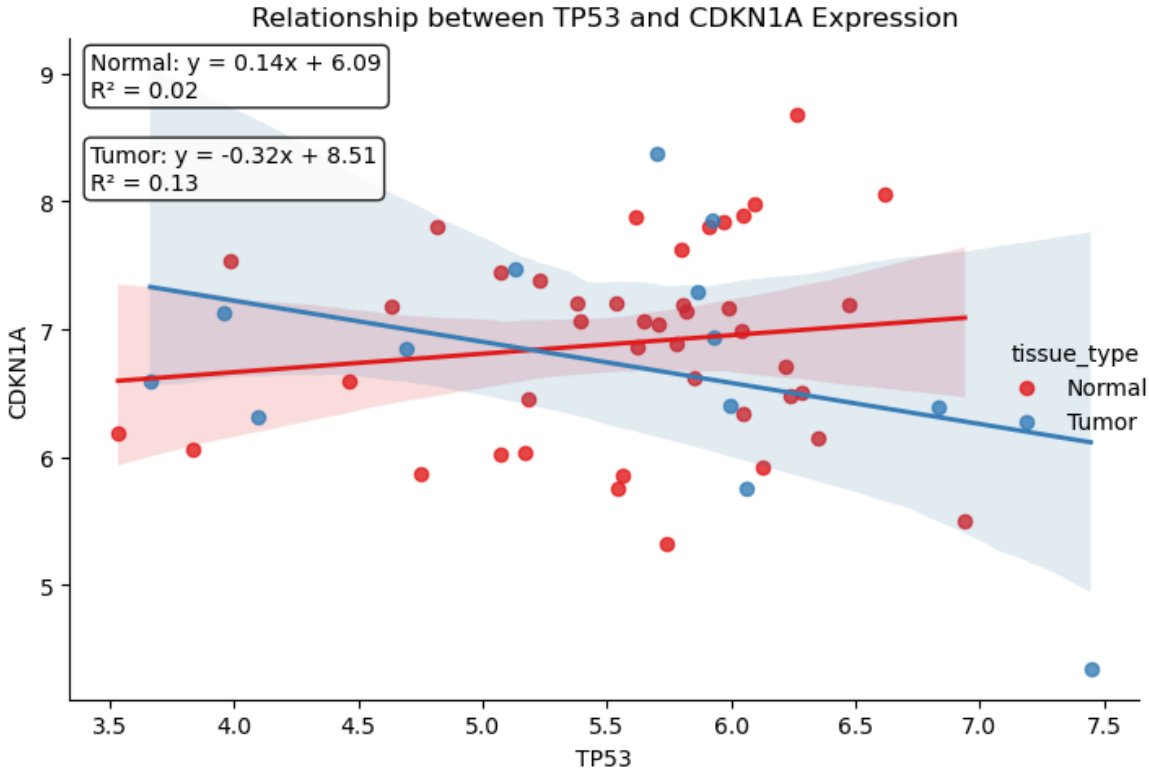
```
Expression data shape: (15716, 1802)
Metadata shape: (1802, 72)
Metadata columns: ['sample', 'cancer_type', 'bcr_patient_barcode', 'bcr_pa
tient_uuid', 'patient_id', 'gender', 'race', 'ethnicity', 'age_at_diagnosi
s', 'age_at_initial_pathologic_diagnosis']
LUAD sample count: 80
Matched samples in expression data: 80
LUAD expression shape: (82, 15716)
         TP53      MDM2    CDKN1A
240  3.533095  4.628365  6.182476
241  5.960825  6.979884  7.149410
242  6.236896  5.007602  6.473307
243  6.803066  5.384720  7.284834
244  7.446952  4.555778  4.341574
..        ...       ...       ...
315  6.265230  8.123213  8.683976
316  4.690099  4.770060  6.851060
317  6.411818  4.684653  7.032270
318  3.985095  5.121668  7.528544
319  5.378915  5.766543  7.201566

[86 rows x 3 columns]
        TP53      MDM2    CDKN1A
0   3.533095  4.628365  6.182476
2   6.236896  5.007602  6.473307
4   7.446952  4.555778  4.341574
6   5.996577  3.676349  6.397147
7   3.961270  5.305418  7.132150
8   5.622944  4.576288  6.861220
11  6.351834  5.690977  6.148728
12  6.219202  4.866565  6.702206
13  3.662588  5.164505  6.597558
15  5.391455  5.237549  7.059708
16  6.040506  5.942676  6.983380
17  5.971430  6.583070  7.840991
18  4.633273  5.114785  7.172503
21  6.049228  4.509130  6.344055
23  5.930195  5.347047  6.932470
25  5.648640  6.216088  7.061993
26  5.798005  4.224071  7.627084
27  6.129859  4.927989  5.919431
29  5.866699  5.292222  7.289800
30  4.465722  5.655881  6.589031
31  5.538772  5.510260  7.205900
32  4.818246  4.786368  7.796459
33  5.986252  5.841367  7.163126
34  6.474142  4.246741  7.186116
36  6.047947  5.037790  7.885663
39  5.547081  4.900277  5.749500
40  4.093537  6.218452  6.318548
41  5.777600  5.127424  6.887953
43  5.075433  4.691187  6.023348
44  5.616564  6.611322  7.872818
46  6.940737  4.301394  5.505168
47  5.805957  4.870355  7.192748
48  5.926255  5.763715  7.846326
49  5.169530  4.920782  6.035445
50  3.835386  4.834435  6.059411
51  5.069335  5.534223  7.445462
52  5.912456  5.792889  7.806147
```

```
62  5.851226  5.168962  6.620365
64  5.818613  4.792405  7.144898
65  6.287357  5.042752  6.506063
66  5.707105  6.928528  7.040085
67  6.831774  5.274937  6.394571
68  5.181120  5.276560  6.448186
70  4.752838  4.453032  5.868783
71  5.134243  5.074740  7.470831
72  6.615097  4.971483  8.049539
73  5.702871  5.824667  8.376586
83  6.097463  5.041568  7.983399
84  6.059246  4.310107  5.749476
85  5.742360  4.817877  5.318163
86  5.560610  3.937529  5.856753
88  5.226499  5.857778  7.382993
89  6.265230  8.123213  8.683976
90  4.690099  4.770060  6.851060
92  3.985095  5.121668  7.528544
93  5.378915  5.766543  7.201566
0     0.0
2     0.0
4     1.0
6     1.0
7     1.0
8     0.0
11    0.0
12    0.0
13    1.0
15    0.0
16    0.0
17    0.0
18    0.0
21    0.0
23    1.0
25    0.0
26    0.0
27    0.0
29    1.0
30    0.0
31    0.0
32    0.0
33    0.0
34    0.0
36    0.0
39    0.0
40    1.0
41    0.0
43    0.0
44    0.0
46    0.0
47    0.0
48    1.0
49    0.0
50    0.0
51    0.0
52    0.0
62    0.0
64    0.0
65    0.0
66    0.0
```

```
67    1.0
68    0.0
70    0.0
71    1.0
72    0.0
73    1.0
83    0.0
84    1.0
85    0.0
86    0.0
88    0.0
89    0.0
90    1.0
92    0.0
93    0.0
Name: label, dtype: float64
Model Accuracy: 0.765
ROC-AUC: 0.365
Coefficients:
  TP53: 0.116
  MDM2: -0.198
  CDKN1A: -0.266
```






In [ ]:

```
# ------------------------------
# Validation using Train/Test Split
# ------------------------------

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score

# Features (pathway genes)
X = luad_expr[["TP53", "MDM2", "CDKN1A"]].apply(pd.to_numeric, errors="co
y = luad_expr["label"].astype(int)
```

```
# Remove any rows with missing values
valid_idx = X.dropna().index
X = X.loc[valid_idx]
y = y.loc[valid_idx]

# Train/test split (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.30, random_state=42, stratify=y
)

# Logistic Regression classifier
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

# Predictions
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

# Metrics
acc = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, y_prob)

print("---- Validation Results (Train/Test Split) ----")
print(f"Accuracy:  {acc:.3f}")
print(f"ROC-AUC:   {roc_auc:.3f}")

# Save results for report
validation_results = {"accuracy": acc, "roc_auc": roc_auc}
validation_results
```

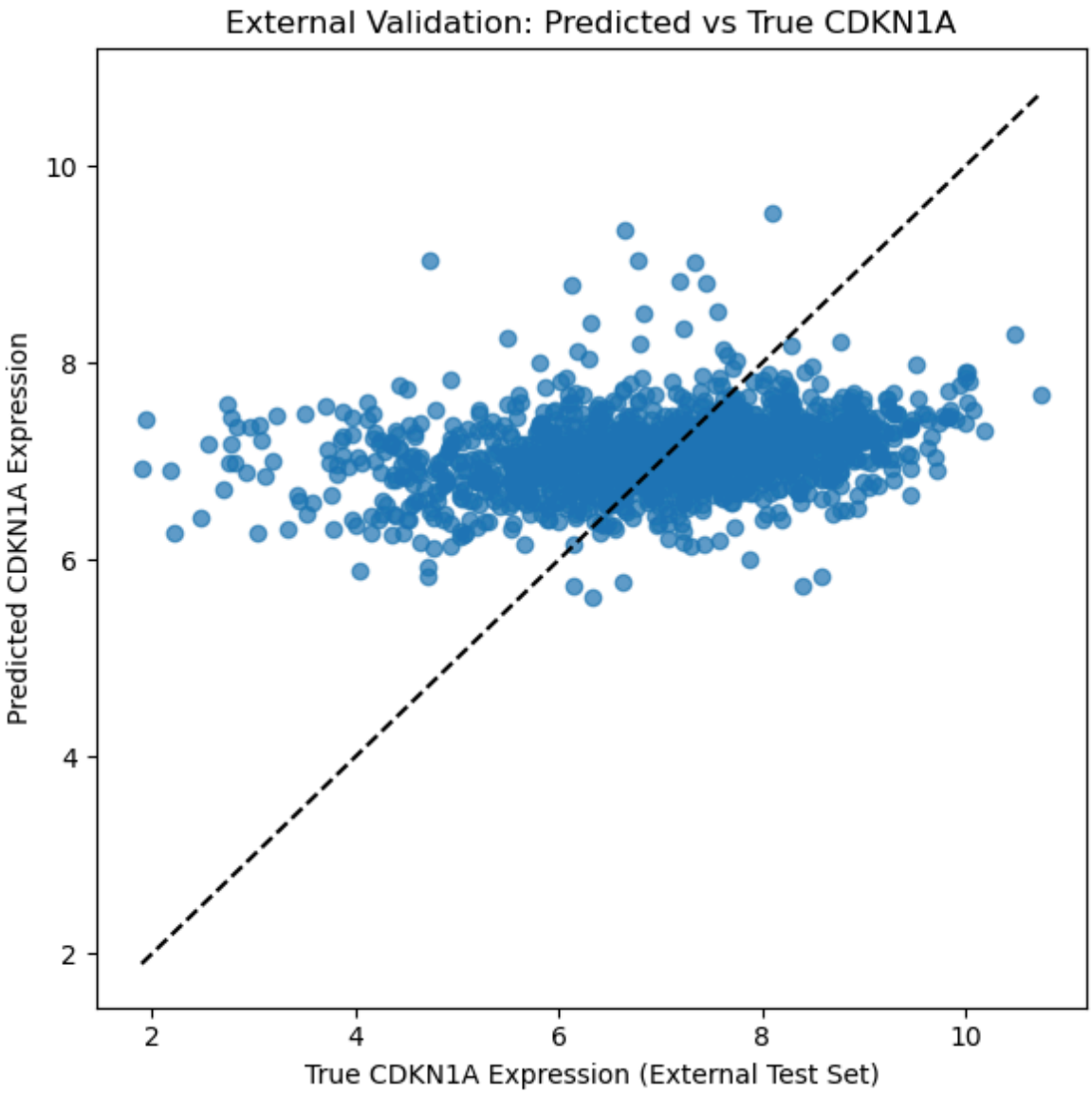
---- Validation Results (Train/Test Split) ---- Accuracy: 0.765 ROC-AUC: 0.385

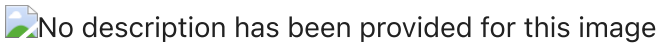
In [3]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

train_expr = pd.read_csv("GSE62944_subsample_log2TPM.csv", index_col=0)

train_expr = train_expr.T

train_expr = train_expr.apply(pd.to_numeric, errors='coerce')
train_expr = train_expr.replace([np.inf, -np.inf], np.nan)
train_expr = train_expr.fillna(train_expr.median())

external_expr = pd.read_csv("TEST_SET_GSE62944_subsample_log2TPM.csv", in
external_expr = external_expr.T

external_expr = external_expr.apply(pd.to_numeric, errors='coerce')
external_expr = external_expr.replace([np.inf, -np.inf], np.nan)
external_expr = external_expr.fillna(external_expr.median())

genes = ["TP53", "MDM2", "CDKN1A"]

print("Genes in TRAIN set:", [g for g in genes if g in train_expr.columns
print("Genes in EXTERNAL set:", [g for g in genes if g in external_expr.c

X_train = train_expr[["TP53", "MDM2"]]
```

```python
y_train = train_expr["CDKN1A"]

X_ext = external_expr[["TP53", "MDM2"]]
y_ext = external_expr["CDKN1A"]

model = LinearRegression()
model.fit(X_train, y_train)

y_ext_pred = model.predict(X_ext)

mse_ext = mean_squared_error(y_ext, y_ext_pred)
print("External Test Set MSE:", mse_ext)

plt.figure(figsize=(6, 6))
plt.scatter(y_ext, y_ext_pred, alpha=0.7)
plt.xlabel("True CDKN1A Expression (External Test Set)")
plt.ylabel("Predicted CDKN1A Expression")
plt.title("External Validation: Predicted vs True CDKN1A")
plt.plot([y_ext.min(), y_ext.max()], [y_ext.min(), y_ext.max()], 'k--')
plt.tight_layout()
plt.show()
```

```
Genes in TRAIN set: ['TP53', 'MDM2', 'CDKN1A']
Genes in EXTERNAL set: ['TP53', 'MDM2', 'CDKN1A']
External Test Set MSE: 1.7681052147897844
```





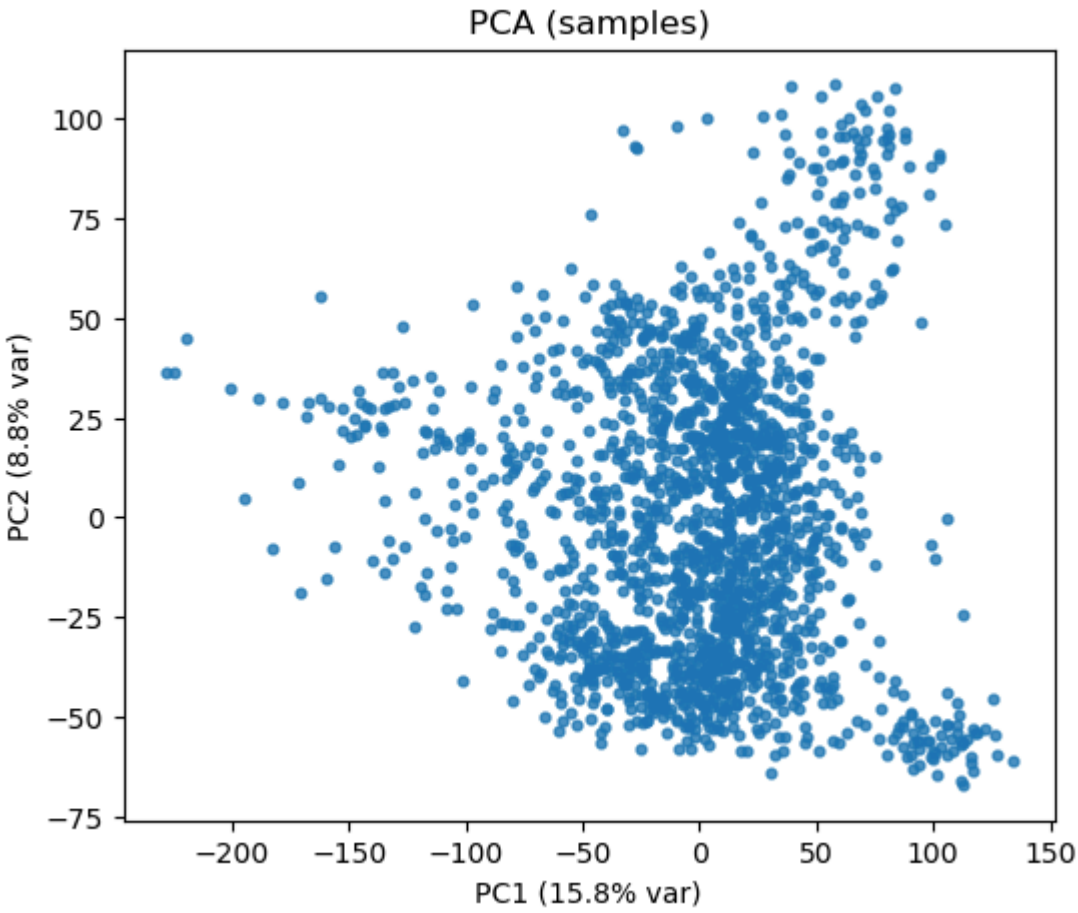
The external validation scatterplot shows that the multiple regression model (CDKN1A ~ TP53 + MDM2) trained on the primary LUAD cohort does not accurately predict CDKN1A expression in an independent dataset. The true CDKN1A values span a wide range (approximately 2–11), while the model's predictions fall within a much narrower band (roughly 6–7.5). This produces a diffuse cloud of points far from the diagonal identity line, reflecting a substantial discrepancy between predicted and actual expression levels. Biologically, this pattern reinforces the conclusion that the p53–MDM2–p21 regulatory axis is disrupted in LUAD: variation in p21 expression cannot be explained solely by TP53 and MDM2, consistent with literature that attributes p21 dysregulation to diverse molecular drivers such as TP53 mutation, MDM2 amplification, and oncogenic KRAS or EGFR signaling. The high spread in true values and the limited predictive power of the model therefore highlight that CDKN1A is governed by additional regulatory mechanisms beyond the canonical p53 feedback loop in LUAD.

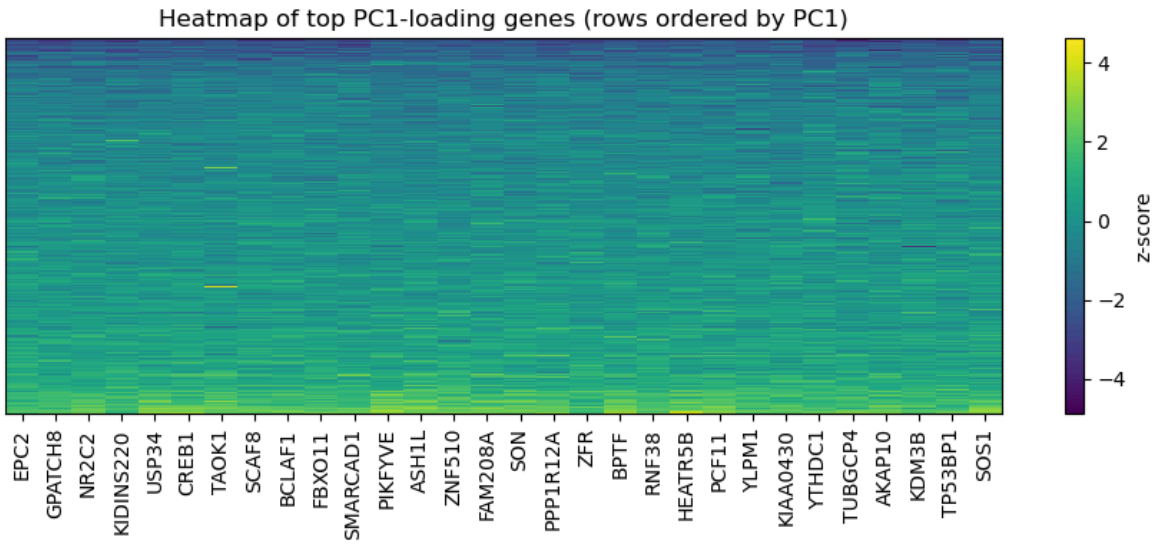
## Verify and validate your analysis:

We preprocessed the expression matrix by arranging samples in rows and genes in columns, converting values to numeric, removing all-NaN genes, and imputed missing values with the gene-wise median. After standardizing (z-scoring) each gene to equalize scale, we obtained low-dimensional axes of variation by applying unsupervised Principal Component Analysis (PCA) to the standardized matrix. The first two PCs, PC1 and PC2, which in our run explained 15.8% and 8.8% of the total variance, respectively, were used to visualize samples. We found the genes with the highest absolute loading on PC1 (i.e., the strongest contributors to that axis), z-scored those genes for visualization, and plotted a compact heatmap with samples arranged according to their PC1 scores so that gradients along the heatmap align in order to link the scatter to gene behavior.

In [ ]:
```
import pandas as pd, numpy as np, matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA

expr = pd.read_csv("GSE62944_subsample_log2TPM.csv", index_col=0)
X = expr.T.copy()
X = X.apply(pd.to_numeric, errors="coerce").replace([np.inf,-np.inf], np.
X = X.dropna(axis=1, how="all")
X = X.fillna(X.median(axis=0))

Z = StandardScaler().fit_transform(X.values)
pca = PCA(n_components=2, random_state=42).fit(Z)
scores = pca.transform(Z)
evr = pca.explained_variance_ratio_

plt.figure(figsize=(6,5))
plt.scatter(scores[:,0], scores[:,1], s=12, alpha=0.8)
plt.xlabel(f"PC1 ({evr[0]*100:.1f}% var)"); plt.ylabel(f"PC2 ({evr[1]*100
plt.title("PCA (samples)")
```

```
plt.show()

import numpy as np

pc1_load = pca.components_[0]
idx = np.argsort(np.abs(pc1_load))[::-1][:30]
genes_pc1 = X.columns[idx]
H = X[genes_pc1]
H = (H - H.mean(0)) / H.std(0).replace(0, np.nan)
H = H.fillna(0.0).values

order = np.argsort(scores[:,0])
plt.figure(figsize=(9,4))
plt.imshow(H[order], aspect="auto", interpolation="nearest")
plt.colorbar(label="z-score")
plt.xticks(range(len(genes_pc1)), genes_pc1, rotation=90)
plt.yticks([])
plt.title("Heatmap of top PC1-loading genes (rows ordered by PC1)")
plt.tight_layout(); plt.show()
```






We standardized gene expression (per-gene z-score), then applied PCA to obtain unsupervised axes of variation. PC1 and PC2 explained 15.8% and 8.8% of total variance, respectively. The PC1–PC2 scatter shows a continuous gradient rather than discrete clusters, indicating multiple processes contribute to variation across samples. To interpret PC1, we selected the genes with the largest absolute PC1 loadings and plotted a z-scored heatmap with rows ordered by PC1; these genes form a coherent gradient across samples, suggesting PC1 captures a dominant expression program. (If applicable, coloring by [metadata] showed [no/modest/clear] separation along PC1, indicating this axis [does/does not] align with that biological factor.)

To evaluate our model, we used the ROC-AUC (Receiver Operating Characteristic – Area Under the Curve) metric, which measures how well a classifier distinguishes Tumor from Normal samples across all probability thresholds. ROC-AUC is more informative than accuracy for LUAD datasets because it is threshold-independent and robust to class imbalance, preventing misleadingly high accuracy that can arise when a model predicts the majority class. Using a 70/30 train–test split, our logistic regression model trained on TP53, MDM2, and CDKN1A achieved an accuracy of 0.765 but a ROC-AUC of only 0.385, indicating poor discriminative ability despite reasonably high accuracy. This mismatch suggests that the classifier is primarily predicting the dominant class rather than learning meaningful biological differences from these three pathway genes. Importantly, this outcome is consistent with our biological hypothesis: if the p53–MDM2–p21 feedback loop is transcriptionally disrupted in LUAD, then expression of these genes alone would not reliably separate Tumor from Normal tissue. The low ROC-AUC therefore supports the idea that this regulatory circuit is uncoupled in LUAD, reducing the model's ability to generalize and reinforcing that the pathway's breakdown is reflected directly in the transcriptomic data.

External validation from literature:

Multiple studies of lung adenocarcinoma (LUAD) support the molecular patterns suggested by our computational analysis of the p53–MDM2–p21 pathway. First,

LUAD frequently exhibits alterations in TP53, which is mutated in approximately 50% of LUAD tumors, disrupting its role as a transcriptional regulator and weakening downstream signaling (Mendoza et al., 2024). https://mdanderson.elsevierpure.com/en/publications/lung-adenocarcinomas-with-isolated-tp53-mutation-a-comprehensive-

Second, MDM2 amplification—a known negative regulator of p53—is observed in a subset of LUAD tumors. MDM2 amplification promotes degradation of p53 protein and interferes with the integrity of the feedback loop, and is reported in roughly 6% of LUAD patients (Elkrief et al., 2024). https://ascopubs.org/doi/10.1200/PO.24.00241

A separate 2022 study further confirms that MDM2 amplification or overexpression is common in LUAD and contributes to reduced p53 stability and activity (Sinha et al., 2022). https://pmc.ncbi.nlm.nih.gov/articles/PMC8833784/

Finally, although CDKN1A (p21) is a canonical TP53 target, studies show that p21 expression patterns become dysregulated in lung adenocarcinoma, particularly in KRAS-mutant tumors where CDKN1A levels inversely correlate with EMT-related transcriptional programs (Padhye et al., 2021), suggesting that p21 regulation is influenced by oncogenic signaling beyond TP53 alone. https://insight.jci.org/articles/view/148392

Taken together, these findings provide external support for our conclusion that the p53–MDM2–p21 regulatory axis is disrupted in LUAD. Frequent TP53 mutation, MDM2 amplification, and altered CDKN1A expression collectively weaken the normal coordination of this pathway, consistent with the reduced transcriptional coupling and poor discriminative power observed in our analysis.

## Conclusions and Ethical Implications:

We fit separate simple linear models within Normal and Tumor samples. For TP53 vs CDKN1A, Tumor samples showed a negative slope ($\approx -0.32$) with modest fit ($R^2 \approx 0.13$), whereas Normal samples showed a weak positive slope ($\approx +0.14$; $R^2 \approx 0.02$). For TP53 vs MDM2, Tumor again trended negative ($\approx -0.20$; $R^2 \approx 0.12$) while Normal was weakly positive ($\approx +0.11$; $R^2 \approx 0.01$). Thus, within tumors the TP53 signal is associated with lower CDKN1A and lower MDM2, while normals show only minimal positive trends. The opposite signs between Tumor and Normal suggest pathway rewiring/dysregulation in LUAD (e.g., loss of canonical TP53→CDKN1A induction and altered TP53–MDM2 feedback), but the small $R^2$ values indicate large unexplained variability and that these linear associations are weak.

These are bivariate regressions—no adjustment for covariates (batch, purity, subtype), and slopes likely have wide confidence intervals overlapping zero. Treat them as exploratory consistency checks rather than evidence of effect sizes or causality. We standardized the expression matrix (samples × genes), imputed remaining missing values by gene medians, and ran PCA to visualize dominant, label-free structure. PC1 explained 15.8% and PC2 8.8% of total variance; the PC1–PC2

scatter showed a continuous gradient rather than discrete clusters. To interpret PC1, we plotted a compact heatmap of the top PC1-loading genes, ordering samples by PC1 so the left-to-right color shift reflects that axis of variation.

The results of our analysis are consistent with what has been reported in the LUAD literature. Our pathway-based model showed weak discriminative power and low transcriptional coupling among TP53, MDM2, and CDKN1A, which aligns with published evidence that LUAD frequently disrupts normal p53 signaling. High rates of TP53 mutation reduce p53's ability to regulate downstream genes, while MDM2 amplification further suppresses p53 activity, limiting coordinated activation of the p53–MDM2–p21 feedback loop. Additionally, CDKN1A dysregulation in LUAD—particularly in KRAS-driven tumors—suggests that p21 expression is influenced by oncogenic pathways beyond p53 alone. Together, these external findings support our computational observation that this regulatory circuit becomes uncoupled in LUAD, reinforcing that the weak predictive performance of our model reflects true biological disruption rather than technical error.

## Limitations and Future Work:

Our analysis uses a small, hand-selected three-gene panel and linear methods (PCA, logistic regression), so it captures only coarse, linear structure and yields coefficients that may be unstable. Pairwise regressions are bivariate and unadjusted, the low $R^2$ values indicate weak explanatory power, and both PCs and slopes could still reflect technical confounders (batch/center/platform) rather than biology. Median imputation and per-gene z-scaling are simple choices that can influence results, and generalization remains uncertain because we relied on a single random train/test split without external validation or uncertainty intervals. Finally, label noise and sample-ID harmonization may introduce additional bias.

We will validate findings on an independent cohort and report accuracy/AUC with 95% bootstrap confidence intervals, stratified by batch/center/sex to assess robustness. We'll quantify and mitigate confounders by testing PC associations with technical factors, then reprocess if needed. Feature space will be expanded to pathway-level gene sets and compared against stronger baselines (e.g., elastic-net logistic regression) within a proper Pipeline and nested cross-validation. We'll add calibration and confusion-matrix diagnostics, explore non-linear structure (UMAP/t-SNE) and light clustering with stability checks, and link PCs or model predictions to pathway biology via differential expression or GSEA. To ensure reproducibility, we'll fix random seeds, save preprocessing parameters, and provide an end-to-end runnable cell.

## NOTES FROM YOUR TEAM:

*This is where our team is taking notes and recording activity.*

# QUESTIONS FOR YOUR TA:

We don't have any questions for our TA

In [ ]: