Module_3: First Jupyter Notebook Check-In (Pair Assignment)

Team Members:

Jaya Kinley and Sydney Schrage

Project Title:

Investigating RAS/MAPK Pathway Activity and Tumor Proliferation for Uterine Carsinosarcoma

Project Goal:

This project seeks to investigate whether activity of the RAS/MAPK signaling pathway is associated with tumor proliferation for Uterine Carsinosarcoma by analyzing MKI67 expression.

Disease Background:

- **Cancer Hallmark Focus:** Self-Sufficiency in Growth Signals
- **Overview of Hallmark:** *(Source: Hallmarks of Cancer by Douglas Hanahan and Robert A. Weinberg)*
    - Normal cells require external signals to regulate growth and maintain tissue homeostasis.
    - Cancer cells gain the ability to produce their own growth signals or proliferate independently, called **self-sufficiency in growth signals**.
    - Mechanisms by which cancer cells achieve this:
        - **Autocrine signaling:** Produce their own growth factors (e.g., PDGF, TGF-α) to stimulate continuous division.
        - **Overexpression or mutation of receptors:** EGFR, HER2/neu overexpression or mutation causes constant activation of proliferative pathways without ligand binding.
        - **Constitutive activation of intracellular signaling pathways:** Mutations in downstream components (Ras, Raf, PI3K) lead to persistent MAPK and PI3K-AKT signaling, stimulating growth and survival.
    - Outcome: Tumor cells proliferate independently of surrounding tissue, breaking normal homeostatic controls.

- **Genes Associated with the Hallmark to Be Studied:** *(Source: Hallmarks of Cancer)*
    - **EGFR (Epidermal Growth Factor Receptor):** Overexpression/mutation drives continuous proliferation; linked to tumor aggressiveness and poor prognosis in UCS.
    - **ERBB2 / HER2 (Human Epidermal Growth Factor Receptor 2):** Amplification or overexpression activates MAPK and PI3K-AKT pathways; HER2-targeted therapies show potential in UCS.
    - **KRAS / NRAS / HRAS (Ras Family Oncogenes):** Mutations lock Ras proteins in active state, stimulating downstream growth pathways; KRAS mutations detected in UCS.
    - **PIK3CA and PTEN (PI3K/AKT Pathway Regulators):** Mutations or loss lead to hyperactivation of PI3K-AKT cascade, promoting survival and growth; common in UCS.
    - **Integrins:** Altered integrin expression promotes survival and proliferation in abnormal microenvironments, supporting growth signaling and invasion.
    - **MAP2K1 (MEK1):** Kinase in the MAPK pathway, phosphorylates ERK/MAPK.
    - **MAPK1 (ERK2):** Regulates transcription to promote proliferation.
    - **MKI67:** Encodes Ki-67, a protein indicating active cell division.
- **Prevalence and Incidence:**
    - UCS is rare, <5% of all uterine cancers; ~2 per 100,000 women develop UCS annually in the U.S.; 5-year survival ~35%.
    - 20-year limited-duration prevalence increased from 0.47 to 3.36 per 100,000 (1999–2018).
    - 2017 incidence per 100,000 by race:
        - Black women: 3.16
        - Hispanic women: 1.27
        - White women: 1.11
        - Asian women: 1.05
    - Sources: NCI TCGA UCS, Frontiers in Public Health, ScienceDirect
- **Risk Factors (Genetic, Lifestyle, & Societal Determinants):** *(source: ChatGPT)*
    - **Genetic:** Mutations in TP53, PIK3CA, PTEN, KRAS; HER2 amplification; EGFR overexpression.
    - **Lifestyle:** Obesity, unopposed estrogen exposure, nulliparity, diabetes mellitus.
    - **Previous treatments:** Prior pelvic radiation therapy.
    - **Societal determinants:** Limited access to gynecologic care, socioeconomic barriers, delayed awareness of postmenopausal bleeding.
- **Standard of Care Treatments (& Reimbursement):**
    - **Targeted therapies for RAS/MAPK-driven cancers:**
        - **RAF inhibitors** (e.g., vemurafenib, dabrafenib)

        - Directly target RAF kinases in the RAS/MAPK pathway.
        - Designed to block aberrant signaling caused by activating **BRAF mutations**, which drive uncontrolled tumor growth.
    - **MEK inhibitors** (e.g., trametinib)
        - Act downstream of RAF in the MAPK cascade.
        - Often used **in combination with RAF inhibitors** to enhance effectiveness.
        - Combination prevents bypass of the blocked RAF signal via downstream pathway reactivation.
        - This approach improves tumor response rates and **delays resistance** compared to single-agent therapy.
  - Source: Bahar, M. E., Kim, H. J., & Kim, D. R. (2023, December 18). Targeting the RAS/RAF/MAPK pathway for cancer therapy: From mechanism to clinical studies. Nature News. https://www.nature.com/articles/s41392-023-01705-z
- **Biological Mechanisms (Anatomy, Organ Physiology, Cell & Molecular Physiology):**
  - **Autocrine/Paracrine Signaling:**
    - Cancer cells produce their own growth factors (e.g., PDGF, TGF-α, IGF) or stimulate neighboring cells to do so.
    - Sustains proliferation without reliance on external signals.
  - **Constitutive Receptor Activation:**
    - Mutations in growth factor receptors (e.g., EGFR) enable continuous signaling independent of ligand binding.
  - **RAS/RAF/MAPK Pathway Activation:**
    - Oncogenic mutations in RAS genes (KRAS, NRAS, HRAS) activate the cascade: **RAF → MEK → ERK**.
    - Promotes transcription of genes that drive cell proliferation.
  - **Tumor Suppressor Inactivation:**
    - Loss of **TP53** or **Rb** function removes regulatory “brakes” on cell division, enabling uncontrolled growth.
  - **Resistance to Inhibitory Signals:**
    - Cancer cells ignore growth-inhibitory cues, allowing persistent proliferation.

Data-Set:

This study uses data from the GSE62944 dataset in the Gene Expression Omnibus (GEO), which contains gene expression information from The Cancer Genome Atlas (TCGA). The dataset includes $\log_2$-transformed Transcripts Per Million (TPM) values that

show how strongly each gene is expressed in tumor and normal samples from many types of cancer. The goal of this project is to see whether RAS pathway activity is related to tumor proliferation across cancers. To do this, we plan to analyze six genes—KRAS, NRAS, HRAS, RAF1, MAP2K1, and MAPK1—which are part of the RAS/MAPK signaling pathway, a system that sends growth signals within cells to promote division and survival. We also plan to analyze MKI67, a well-established marker of cellular proliferation, to test whether changes in pathway activity were reflected in tumor growth behavior. We manually searched the expression data csv to confirm that all seven of these genes were present in the dataset.

The data for this project come from The Cancer Genome Atlas (TCGA) and were reprocessed as part of the GSE62944 dataset (Rahman et al., 2015). Tumor and normal tissue samples from many cancer types were analyzed using Illumina HiSeq 2000 RNA sequencing. RNA was extracted from each sample, converted to cDNA, and sequenced to measure how much each gene was expressed. The sequencing reads were aligned to the human genome (hg19) using Rsubread, a computational tool that accurately maps short sequencing reads to their positions in the genome. After alignment, the number of reads mapped to each gene was counted to estimate expression levels. These counts were converted to Transcripts Per Million (TPM) to account for differences in gene length and sequencing depth, and then $\log_2$-transformed so that expression levels could be compared across samples. Clinical information such as cancer type and survival data came from the TCGA Pan-Cancer Clinical Data Resource (Liu et al., 2018), which standardized patient data across TCGA projects.

In addition to gene expression values, we used a metadata file containing clinical and sample-level annotations for every TCGA sample in the dataset. This metadata includes information such as cancer type, patient barcode, tissue source site, sample ID, tumor/normal status, and various clinical attributes (e.g., stage, grade, treatment history, and survival outcomes). These annotations allow us to filter the dataset to specific cancers—in this case, uterine carcinosarcoma (UCS)—and to link gene expression with biological and clinical variables such as tumor type or patient survival.

By focusing on these genes and expression measures, this study tests whether higher RAS pathway activity is associated with increased tumor proliferation across different cancer types, reflecting the role of RAS signaling in promoting uncontrolled cell growth.

Sources:

Liu J, Lichtenberg T, Hoadley KA, Poisson LM, Lazar AJ, Cherniack AD, Kovatich AJ, Benz CC, Levine DA, Lee AV, Omberg L, Wolf DM, Shriver CD, Thorsson V; Cancer Genome Atlas Research Network; Hu H. An Integrated TCGA Pan-Cancer Clinical Data Resource to Drive High-Quality Survival Outcome Analytics. Cell. 2018 Apr 5;173(2):400-416.e11. doi: 10.1016/j.cell.2018.02.052. PMID: 29625055; PMCID: PMC6066282.

Rahman M, Jackson LK, Johnson WE, Li DY, Bild AH, Piccolo SR. Alternative preprocessing of RNA-Sequencing data in The Cancer Genome Atlas leads to improved analysis results. Bioinformatics. 2015 Nov 15;31(22):3666-72. doi: 10.1093/bioinformatics/btv377. Epub 2015 Jul 24. PMID: 26209429; PMCID: PMC4804769.

Data Analysis

Principal Component Analysis (PCA)

What is this method? Principal Component Analysis (PCA) is an unsupervised learning technique used for dimensionality reduction. It identifies new axes, called principal components, that summarize patterns in the data by capturing the directions of greatest variation. In our project, PCA was applied to reduce a complex, multi-variable dataset into two main components (PC1 and PC2) that represent the dominant trends across all samples. PCA optimizes for maximum variance, meaning it finds the directions in which the data vary the most. Each principal component is a linear combination of the original variables and is orthogonal to the others, ensuring that each captures unique information. PC1 explained the largest proportion of variance, while PC2 represented a smaller, independent source of variation. The PCA model is considered sufficient when a small number of components explain most of the total variance. Once additional components add only minimal variance, the reduction is effective. This allows for simpler visualization and interpretation without major loss of information.

Visualization using color coding

To further interpret the results, a MKI67 was color-coded across the PCA plot. Samples with higher values of MKI67 appeared in warmer colors (yellow/ green), revealing visible clustering or gradients along PC1 and PC2. This visualization is used in an attempt to identify how variation in the dataset related to meaningful differences across samples. Regression Line Analysis

Linear Regression Analysis

A simple linear regression was applied to explore potential relationships between the reduced dimensions (PC1 and PC2) and MKI67. This supervised learning method models how a dependent variable changes in response to one or more predictors. The regression model optimizes by minimizing the sum of squared errors between predicted and observed values. This ensures the fitted line best represents the overall trend in the data, providing a quantitative measure of association between the variables.

Model performance was then evaluated using the coefficient of determination (R^2), which indicates how much of the variation in the dependent variable is explained by the

model. In general, R^2 (the coefficient of determination) measures how well the independent variables account for variability in the dependent variable — with values closer to 1 representing a strong fit and values near 0 indicating a weak or no linear relationship. Using this method will help use determine MKI67 relationship with PC1 and PC2, and if the PCA explains variation in MKI67.

Importing Libraries

In [1]:
```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
```

run_pca_ucs Function Definition

In [ ]:
```
def run_pca_ucs():
    # --- Step 1. Load metadata ---
    meta = pd.read_csv("GSE62944_metadata.csv")
    print("Cancer types available:", meta["cancer_type"].unique())

    # --- Step 2. Filter UCS samples ---
    ucs_meta = meta[meta["cancer_type"].str.contains("UCS", case=False, na=F
    print(f"Found {ucs_meta.shape[0]} UCS samples.")

    # --- Step 3. Load expression data ---
    expr = pd.read_csv("GSE62944_subsample_log2TPM.csv", index_col=0)
    print(f"Expression data shape: {expr.shape}")

    # --- Step 4. Match samples ---
    common_samples = expr.columns.intersection(ucs_meta["sample"])
    expr_ucs = expr[common_samples].T
    print(f"Matched {expr_ucs.shape[0]} UCS samples.")

    # --- Step 5. Select KRAS–MAPK genes ---
    genes = ["KRAS", "NRAS", "HRAS", "RAF1", "MAP2K1", "MAPK1"]
    subset = expr_ucs[[g for g in genes if g in expr_ucs.columns]]

    #  Extract MKI67 for coloring + regression
    if "MKI67" in expr_ucs.columns:
        mk_expr = expr_ucs["MKI67"]
        print("MKI67 found in dataset. Will use for regression.")
    else:
        mk_expr = None
        print("⚠ MKI67 not found in dataset. Skipping regression step.")

    # --- Step 6. Scale gene expression ---
    X = StandardScaler().fit_transform(subset)

    # --- Step 7. PCA ---
    pca = PCA(n_components=2)
```

```
Z = pca.fit_transform(X)
evr = pca.explained_variance_ratio_
print("\nExplained variance ratio:", evr)

# --- Step 8. Loadings ---
loadings = pd.DataFrame(
    pca.components_.T,
    index=subset.columns,
    columns=["PC1","PC2"]
)
print("\n=== PCA Loadings ===\n")
print(loadings.round(3))

# --- Step 9. PCA Scatter (colored by MKI67) ---
plt.figure(figsize=(8,6))
if mk_expr is not None:
    sc = plt.scatter(Z[:,0], Z[:,1], c=mk_expr, cmap="viridis", edgecolo
    plt.colorbar(sc, label="MKI67 expression (log2 TPM)")
else:
    plt.scatter(Z[:,0], Z[:,1], edgecolor='k')

plt.xlabel(f"PC1 ({evr[0]*100:.1f}% var)")
plt.ylabel(f"PC2 ({evr[1]*100:.1f}% var)")
plt.title("PCA of KRAS Pathway Genes in Uterine Carcinosarcoma (UCS)")
plt.grid(True)
plt.show()

# === Linear Regression — MKI67 vs PC1 ===
# if mk_expr is not None:
#     X1 = Z[:, [0]]   # PC1 only
#     y = mk_expr.values
#     reg1 = LinearRegression().fit(X1, y)

#     print("\n=== Regression: MKI67 ~ PC1 ===")
#     print(f"Intercept: {reg1.intercept_:.3f}")
#     print(f"PC1 slope: {reg1.coef_[0]:.3f}")
#     print(f"R^2: {reg1.score(X1, y):.3f}")

#     # Plot regression line
#     plt.figure(figsize=(7,5))
#     plt.scatter(X1, y, edgecolor='k')
#     x_line = np.linspace(X1.min(), X1.max(), 100).reshape(-1,1)
#     plt.plot(x_line, reg1.predict(x_line), color='red')
#     plt.xlabel("PC1")
#     plt.ylabel("MKI67 (log2 TPM)")
#     plt.title("Linear Regression: MKI67 ~ PC1")
#     plt.grid(True)
#     plt.show()

#     # === Regression 2 — MKI67 vs PC2 ===
#     X2 = Z[:, [1]]   # PC2 only
#     reg2 = LinearRegression().fit(X2, y)

#     print("\n=== Regression: MKI67 ~ PC2 ===")
#     print(f"Intercept: {reg2.intercept_:.3f}")
#     print(f"PC2 slope: {reg2.coef_[0]:.3f}")
```

```
#     print(f"R^2: {reg2.score(X2, y):.3f}")

#     # Plot regression line
#     plt.figure(figsize=(7,5))
#     plt.scatter(X2, y, edgecolor='k')
#     x_line = np.linspace(X2.min(), X2.max(), 100).reshape(-1,1)
#     plt.plot(x_line, reg2.predict(x_line), color='red')
#     plt.xlabel("PC2")
#     plt.ylabel("MKI67 (log2 TPM)")
#     plt.title("Linear Regression: MKI67 ~ PC2")
#     plt.grid(True)
#     plt.show()
    # === Regression 1 — MKI67 vs PC1 (Quadratic fit) ===
    # === Quadratic Regression 1 — MKI67 vs PC1 ===
# if mk_expr is not None:
#     y = mk_expr.values

#     # --- PC1 quadratic regression ---
#     X1 = Z[:, [0]]
#     poly1 = PolynomialFeatures(degree=2, include_bias=False)
#     X1_poly = poly1.fit_transform(X1)
#     reg1 = LinearRegression().fit(X1_poly, y)

#     print("\n=== Quadratic Regression: MKI67 ~ PC1 + PC1² ===")
#     print(f"Intercept: {reg1.intercept_:.3f}")
#     print(f"PC1 coefficient: {reg1.coef_[0]:.3f}")
#     print(f"PC1² coefficient: {reg1.coef_[1]:.3f}")
#     print(f"R²: {reg1.score(X1_poly, y):.3f}")

#     # Plot MKI67 vs PC1 (quadratic)
#     plt.figure(figsize=(7,5))
#     plt.scatter(X1, y, edgecolor='k')
#     x_line = np.linspace(X1.min(), X1.max(), 200).reshape(-1,1)
#     x_line_poly = poly1.transform(x_line)
#     plt.plot(x_line, reg1.predict(x_line_poly), color='red')
#     plt.xlabel("PC1")
#     plt.ylabel("MKI67 (log2 TPM)")
#     plt.title("Quadratic Regression: MKI67 ~ PC1 + PC1²")
#     plt.grid(True)
#     plt.show()

#     # --- PC2 quadratic regression ---
#     X2 = Z[:, [1]]
#     poly2 = PolynomialFeatures(degree=2, include_bias=False)
#     X2_poly = poly2.fit_transform(X2)
#     reg2 = LinearRegression().fit(X2_poly, y)

#     print("\n=== Quadratic Regression: MKI67 ~ PC2 + PC2² ===")
#     print(f"Intercept: {reg2.intercept_:.3f}")
#     print(f"PC2 coefficient: {reg2.coef_[0]:.3f}")
#     print(f"PC2² coefficient: {reg2.coef_[1]:.3f}")
#     print(f"R²: {reg2.score(X2_poly, y):.3f}")

#     # Plot MKI67 vs PC2 (quadratic)
#     plt.figure(figsize=(7,5))
#     plt.scatter(X2, y, edgecolor='k')
```

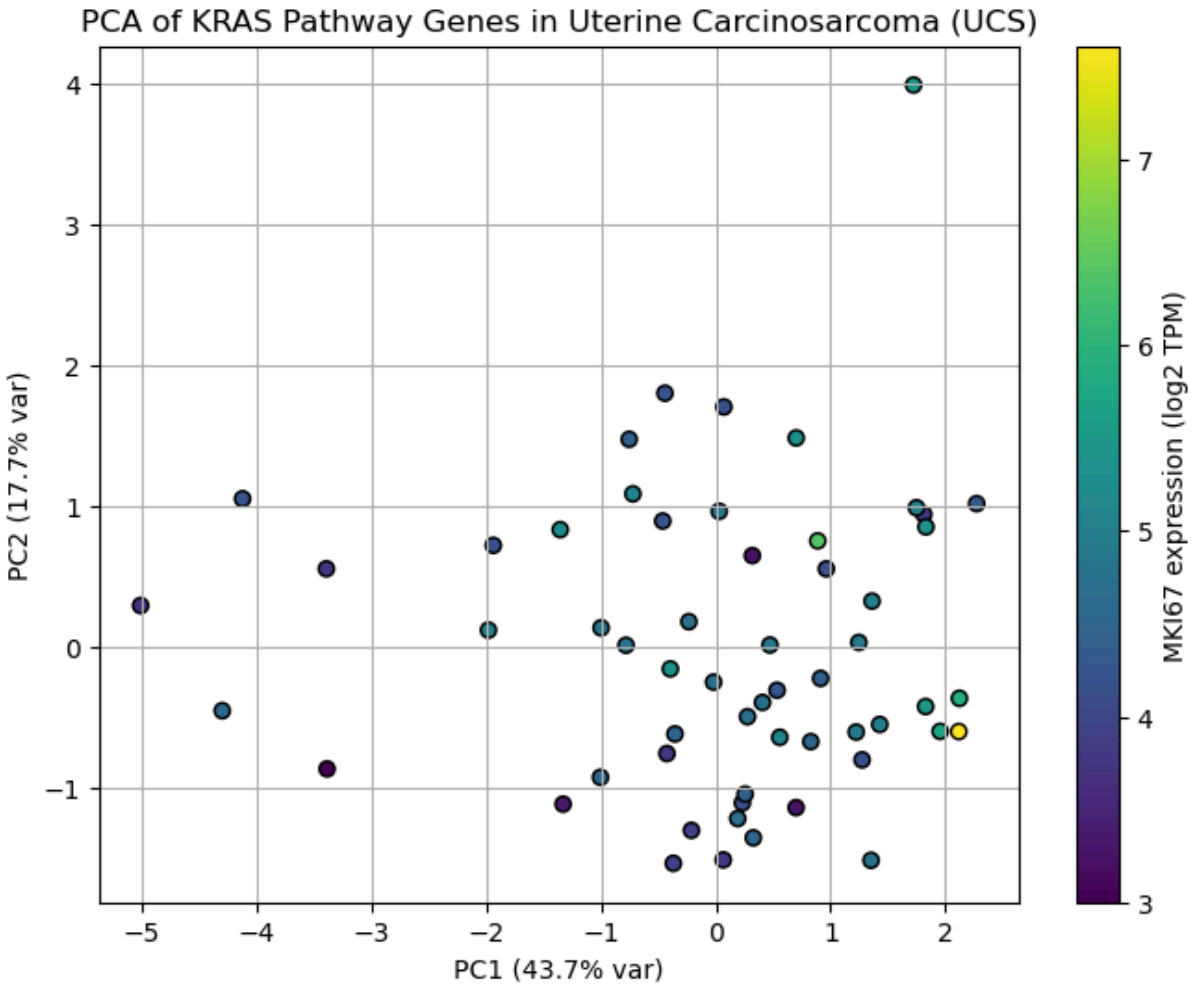
```
#     x_line = np.linspace(X2.min(), X2.max(), 200).reshape(-1,1)
#     x_line_poly = poly2.transform(x_line)
#     plt.plot(x_line, reg2.predict(x_line_poly), color='red')
#     plt.xlabel("PC2")
#     plt.ylabel("MKI67 (log2 TPM)")
#     plt.title("Quadratic Regression: MKI67 ~ PC2 + PC2²")
#     plt.grid(True)
#     plt.show()

    # === Cubic Regression: MKI67 vs PC1 and PC2 ===
if mk_expr is not None:
    y = mk_expr.values

    # --- PC1 cubic regression ---
    X1 = Z[:, [0]]
    from sklearn.preprocessing import PolynomialFeatures
    poly1 = PolynomialFeatures(degree=3, include_bias=False)
    X1_poly = poly1.fit_transform(X1)
    reg1 = LinearRegression().fit(X1_poly, y)

    print("\n=== Cubic Regression: MKI67 ~ PC1 + PC1² + PC1³ ===")
    print(f"Intercept: {reg1.intercept_:.3f}")
    print(f"PC1 coefficient: {reg1.coef_[0]:.3f}")
    print(f"PC1² coefficient: {reg1.coef_[1]:.3f}")
    print(f"PC1³ coefficient: {reg1.coef_[2]:.3f}")
    print(f"R²: {reg1.score(X1_poly, y):.3f}")

    # Plot cubic regression curve (PC1)
    plt.figure(figsize=(7,5))
    plt.scatter(X1, y, edgecolor='k')
    x_line = np.linspace(X1.min(), X1.max(), 200).reshape(-1,1)
    x_line_poly = poly1.transform(x_line)
    plt.plot(x_line, reg1.predict(x_line_poly), color='red')
    plt.xlabel("PC1")
    plt.ylabel("MKI67 (log2 TPM)")
    plt.title("Cubic Regression: MKI67 ~ PC1 + PC1² + PC1³")
    plt.grid(True)
    plt.show()

    # --- PC2 cubic regression ---
    X2 = Z[:, [1]]
    poly2 = PolynomialFeatures(degree=3, include_bias=False)
    X2_poly = poly2.fit_transform(X2)
    reg2 = LinearRegression().fit(X2_poly, y)

    print("\n=== Cubic Regression: MKI67 ~ PC2 + PC2² + PC2³ ===")
    print(f"Intercept: {reg2.intercept_:.3f}")
    print(f"PC2 coefficient: {reg2.coef_[0]:.3f}")
    print(f"PC2² coefficient: {reg2.coef_[1]:.3f}")
    print(f"PC2³ coefficient: {reg2.coef_[2]:.3f}")
    print(f"R²: {reg2.score(X2_poly, y):.3f}")

    # Plot cubic regression curve (PC2)
    plt.figure(figsize=(7,5))
    plt.scatter(X2, y, edgecolor='k')
    x_line = np.linspace(X2.min(), X2.max(), 200).reshape(-1,1)
```

```
        x_line_poly = poly2.transform(x_line)
        plt.plot(x_line, reg2.predict(x_line_poly), color='red')
        plt.xlabel("PC2")
        plt.ylabel("MKI67 (log2 TPM)")
        plt.title("Cubic Regression: MKI67 ~ PC2 + PC2² + PC2³")
        plt.grid(True)
        plt.show()



    return loadings
```

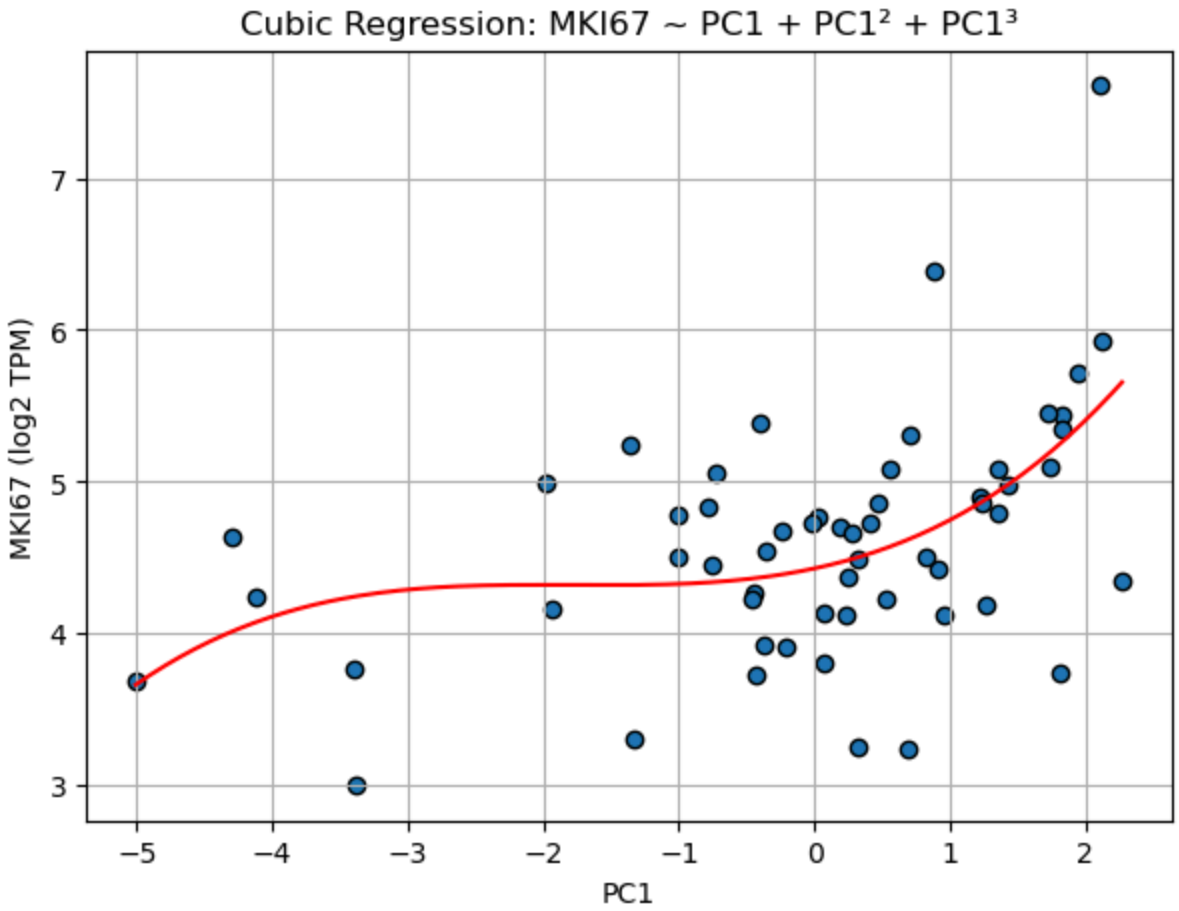
Execute PCA Function

In [30]:
```
loadings = run_pca_ucs()
```

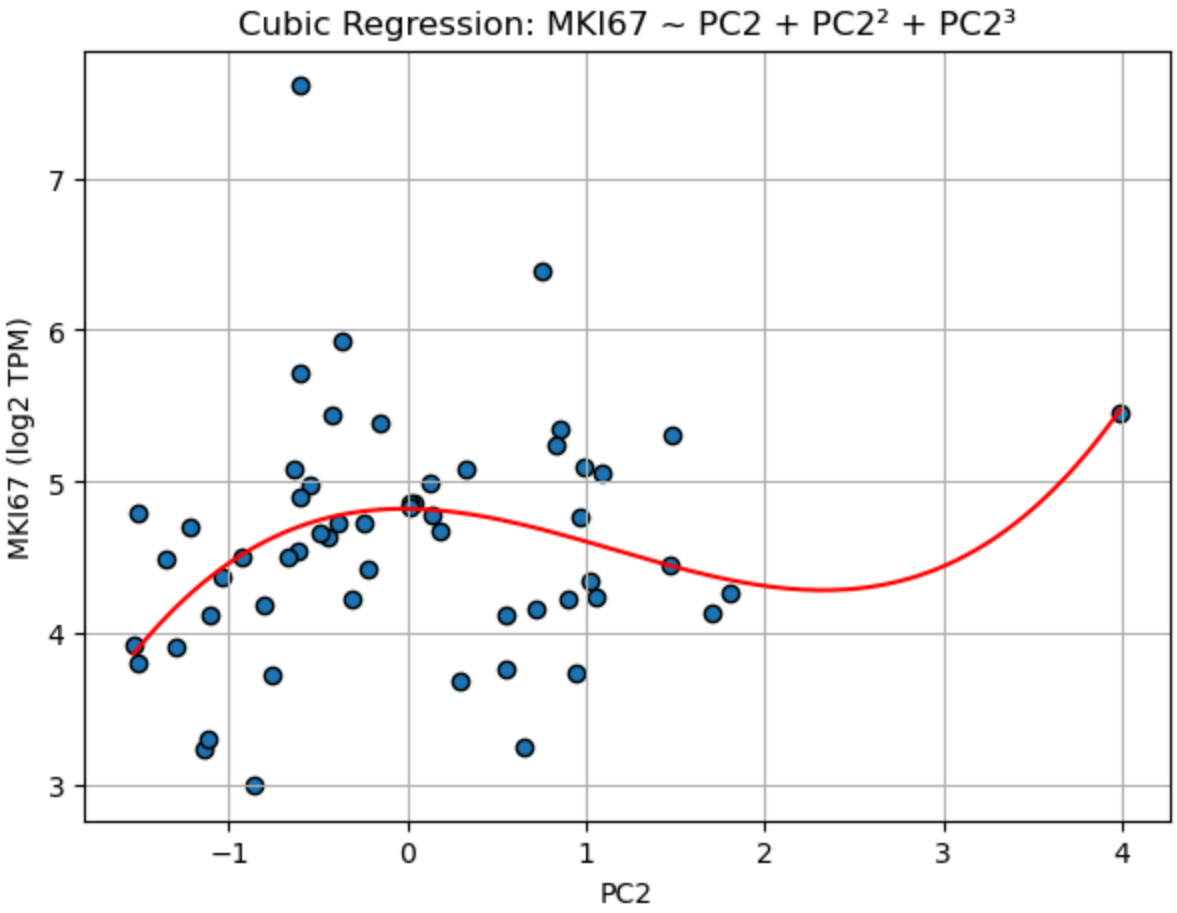
```
Cancer types available: ['BRCA' 'UCEC' 'KIRC' 'LUAD' 'LGG' 'THCA' 'HNSC' 'LU
SC' 'PRAD' 'COAD'
 'SKCM' 'OV' 'STAD' 'BLCA' 'LIHC' 'CESC' 'KIRP' 'LAML' 'GBM' 'READ' 'ACC'
 'KICH' 'UCS']
Found 57 UCS samples.
Expression data shape: (15716, 1802)
Matched 57 UCS samples.
MKI67 found in dataset. Will use for regression.

Explained variance ratio: [0.43749203 0.17694263]

=== PCA Loadings ===

          PC1    PC2
KRAS    0.261  0.406
NRAS    0.451  0.094
HRAS   -0.141  0.857
RAF1    0.484  0.196
MAP2K1  0.469 -0.012
MAPK1   0.504 -0.232
```




```
=== Cubic Regression: MKI67 ~ PC1 + PC1² + PC1³ ===
Intercept: 4.429
PC1 coefficient: 0.190
PC1² coefficient: 0.108
PC1³ coefficient: 0.020
R²: 0.260
```




```
=== Cubic Regression: MKI67 ~ PC2 + PC2² + PC2³ ===
Intercept: 4.822
PC2 coefficient: -0.012
PC2² coefficient: -0.287
PC2³ coefficient: 0.083
R²: 0.116
```



Analysis of Results

The PCA revealed that PC1 explains 43.7% of variation in pathway gene expression and reflects a coordinated pattern of RAS/MAPK activation across tumors. When MKI67 expression was overlaid on the PCA plot, tumors with higher proliferation (yellow/green points) showed slight clustering toward the higher end of PC1, suggesting a relationship between pathway activation and tumor growth.

The cubic regression of MKI67 on PC1 showed a weak positive association, explaining about 26% of the variation in proliferation ($R^2 = 0.26$). The regression using PC2 was even weaker, accounting for only about 11.6% of variation ($R^2 = 0.116$), indicating that PC2 had little meaningful relationship to MKI67 expression.

These results indicate that RAS/MAPK signaling makes only a slight contribution to proliferative behavior in UCS tumors, as reflected by the weak correlations observed. The low R^2 values suggest that this pathway is not a dominant predictor of proliferation, and that additional molecular or regulatory factors are likely playing a larger role.

Verify and validate your analysis:

To evaluate how well the model performed, we used external validation with an independent dataset and assessed performance using the coefficient of determination (R^2).

The original regression models were trained on uterine carcinosarcoma (UCS), where we found that:

- PC1 explained 26% of variation in MKI67 expression
- PC2 explained very little variation ($R^2 \approx 0.116$)

To test whether these relationships were generalizable outside of this sample of UCS, we applied the same PCA loadings and same regression coefficients to a new dataset. However, data for UCS was not included in this new sample so we were forced to validate our findings using UCEC (Uterine Corpus Endometrial Carcinoma) samples.

This ultimately allowed us to test whether the MAPK–proliferation relationship generalized to a different uterine cancer subtype without retraining, which provided a strong test of model validity.

Performance was evaluated by computing R^2 between the predicted and true MKI67 expression in the new dataset.

Validation Code

In [31]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score

# --- Coefficients from UCS cubic regression (training) ---
intercept_PC1 = 4.429
coef_PC1 = 0.190
coef_PC1_sq = 0.108
coef_PC1_cu = 0.020

intercept_PC2 = 4.822
coef_PC2 = -0.012
coef_PC2_sq = -0.287
coef_PC2_cu = 0.083

# --- Load TEST dataset ---
meta_test = pd.read_csv("TEST_SET_GSE62944_metadata.csv")
expr_test = pd.read_csv("TEST_SET_GSE62944_subsample_log2TPM.csv", index_col

print("Expression data shape (test set):", expr_test.shape)
print("Metadata rows (test set):", meta_test.shape[0])

# --- Filter for UCEC samples ---
ucec_samples_test = meta_test[meta_test["cancer_type"].str.contains("UCEC",
print(f"UCEC samples in TEST metadata: {len(ucec_samples_test)}")
```

```
# --- Match expression samples ---
common_samples = expr_test.columns.intersection(ucec_samples_test)
expr_ucec = expr_test[common_samples].T
print(f"Matched UCEC samples in TEST expression matrix: {expr_ucec.shape[0]}")

# --- Extract genes used for PCA ---
genes = ["KRAS", "NRAS", "HRAS", "RAF1", "MAP2K1", "MAPK1"]
subset_test = expr_ucec[[g for g in genes if g in expr_ucec.columns]]

# --- Scale expression data ---
X_test = StandardScaler().fit_transform(subset_test)

# --- Project onto PCA loadings from UCS training set ---
loadings = pd.DataFrame(
    [[0.261, 0.406],
     [0.451, 0.094],
     [-0.141, 0.857],
     [0.484, 0.196],
     [0.469, -0.012],
     [0.504, -0.232]],
    index=genes,
    columns=["PC1","PC2"]
)

Z_test = X_test.dot(loadings.values)
PC1_test = Z_test[:,0]
PC2_test = Z_test[:,1]

# --- Get true MKI67 values ---
mki67_test = expr_ucec["MKI67"]

# --- Predict MKI67 using cubic UCS model ---
y_pred_PC1 = (intercept_PC1
              + coef_PC1 * PC1_test
              + coef_PC1_sq * (PC1_test**2)
              + coef_PC1_cu * (PC1_test**3))

y_pred_PC2 = (intercept_PC2
              + coef_PC2 * PC2_test
              + coef_PC2_sq * (PC2_test**2)
              + coef_PC2_cu * (PC2_test**3))

# --- Compute validation R² scores ---
r2_PC1 = r2_score(mki67_test, y_pred_PC1)
r2_PC2 = r2_score(mki67_test, y_pred_PC2)

print("\n=== VALIDATION RESULTS on UCEC dataset (Cubic Regression) ===")
print(f"R² (MKI67 ~ PC1 + PC1² + PC1³): {r2_PC1:.3f}")
print(f"R² (MKI67 ~ PC2 + PC2² + PC2³): {r2_PC2:.3f}")
```

```
Expression data shape (test set): (15716, 1600)
Metadata rows (test set): 1600
UCEC samples in TEST metadata: 80
Matched UCEC samples in TEST expression matrix: 80

=== VALIDATION RESULTS on UCEC dataset (Cubic Regression) ===
R² (MKI67 ~ PC1 + PC1² + PC1³): 0.119
R² (MKI67 ~ PC2 + PC2² + PC2³): -0.163
```

Implementation of Elastic Net Regression

When we attempted to validate our cubic regression on the validation dataset of UCEC samples, it was found that the R^2 values for both PC1 and PC2 were markedly lower or even the opposite sign as the original correlation on the training dataset. R^2 for PC1: 0.119 R^2 for PC2: -0.163 Because of this, we were able to conclude that the cubic regression was overfit to the training data. In order to combat the effects of overfitting, an elastic net regression was implemented. Elastic Net regression builds a model that combines Lasso (L1) and Ridge (L2) methods to prevent overfitting and deal with features that carry overlapping or redundant information. Below are the new R^2 values for both the training and validation data after implementing the elastic net regression on the cubic fit.

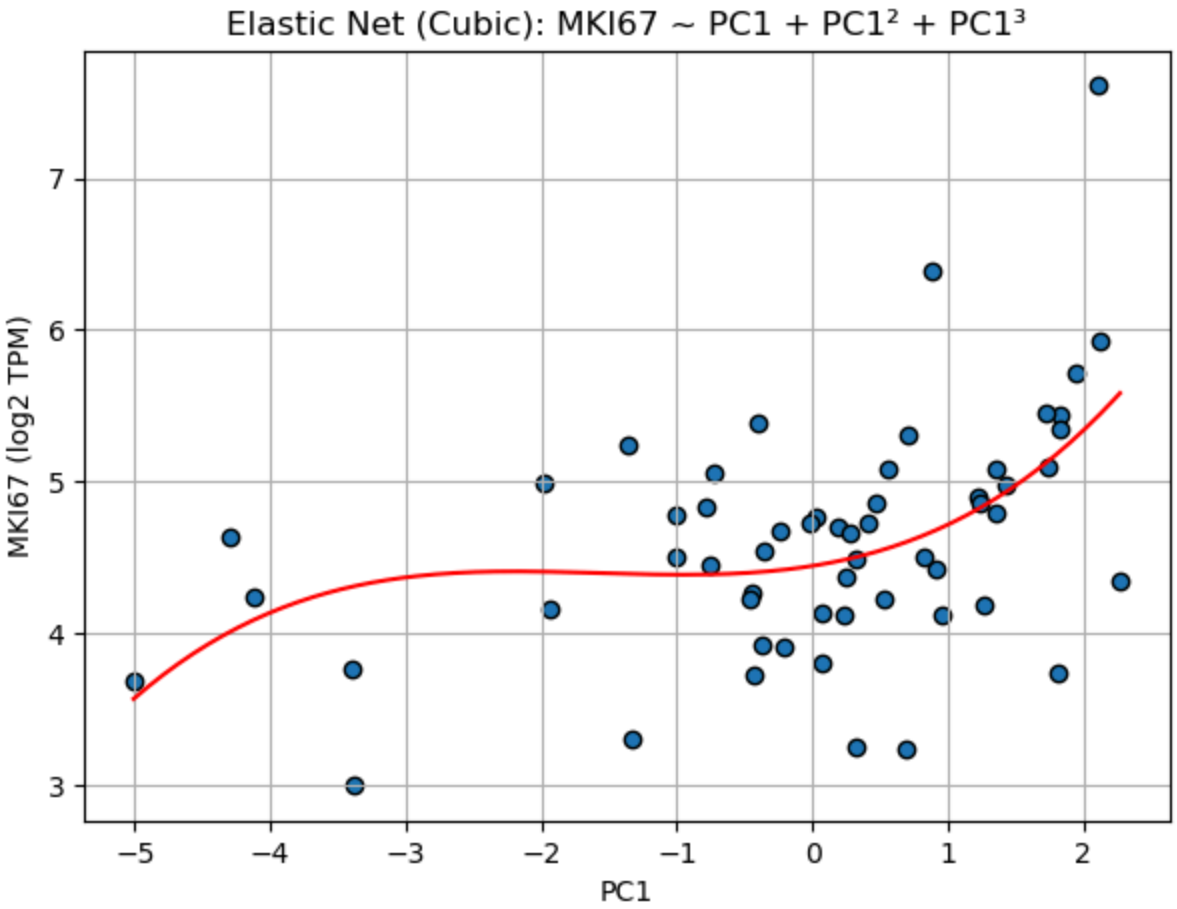
In [35]:
```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import ElasticNetCV
from sklearn.metrics import r2_score
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

def run_pca_ucs_elasticnet():
    # --- Step 1. Load metadata ---
    meta = pd.read_csv("GSE62944_metadata.csv")
    print("Cancer types available:", meta["cancer_type"].unique())

    # --- Step 2. Filter UCS samples ---
    ucs_meta = meta[meta["cancer_type"].str.contains("UCS", case=False, na=F
    print(f"Found {ucs_meta.shape[0]} UCS samples.")

    # --- Step 3. Load expression data ---
    expr = pd.read_csv("GSE62944_subsample_log2TPM.csv", index_col=0)
    print(f"Expression data shape: {expr.shape}")

    # --- Step 4. Match samples ---
    common_samples = expr.columns.intersection(ucs_meta["sample"])
    expr_ucs = expr[common_samples].T
    print(f"Matched {expr_ucs.shape[0]} UCS samples.")

    # --- Step 5. Select KRAS–MAPK genes ---
    genes = ["KRAS", "NRAS", "HRAS", "RAF1", "MAP2K1", "MAPK1"]
    subset = expr_ucs[[g for g in genes if g in expr_ucs.columns]]
```

```
    # Extract MKI67 for regression
    if "MKI67" in expr_ucs.columns:
        mk_expr = expr_ucs["MKI67"]
        print("MKI67 found in dataset. Will use for regression.")
    else:
        print("⚠ MKI67 not found in dataset. Skipping regression.")
        return

    # --- Step 6. Scale gene expression ---
    from sklearn.preprocessing import StandardScaler
    X = StandardScaler().fit_transform(subset)

    # --- Step 7. PCA ---
    from sklearn.decomposition import PCA
    pca = PCA(n_components=2)
    Z = pca.fit_transform(X)
    evr = pca.explained_variance_ratio_
    print("\nExplained variance ratio:", evr)

    loadings = pd.DataFrame(
        pca.components_.T,
        index=subset.columns,
        columns=["PC1","PC2"]
    )
    print("\n=== PCA Loadings ===\n")
    print(loadings.round(3))

    # --- Step 8. Elastic Net (Cubic Regression) for PC1 and PC2 ---
    y = mk_expr.values
    from sklearn.linear_model import ElasticNetCV

    for i, pc_label in enumerate(["PC1", "PC2"]):
        X_pc = Z[:, [i]]
        poly = PolynomialFeatures(degree=3, include_bias=False)
        X_poly = poly.fit_transform(X_pc)

        # Elastic Net with cross-validation
        model = ElasticNetCV(
            l1_ratio=[.1, .3, .5, .7, .9, 1.0],
            alphas=np.logspace(-3, 1, 50),
            cv=5,
            max_iter=5000,
            random_state=42
        ).fit(X_poly, y)

        print(f"\n=== Elastic Net (Cubic) Regression: MKI67 ~ {pc_label} + {
        print(f"Best alpha: {model.alpha_:.4f}")
        print(f"Best l1_ratio: {model.l1_ratio_:.2f}")
        print(f"Coefficients: {model.coef_.round(3)}")
        print(f"R² (on training data): {model.score(X_poly, y):.3f}")

        # Plot prediction curve
        plt.figure(figsize=(7,5))
        plt.scatter(X_pc, y, edgecolor='k')
        x_line = np.linspace(X_pc.min(), X_pc.max(), 200).reshape(-1,1)
        x_line_poly = poly.transform(x_line)
```

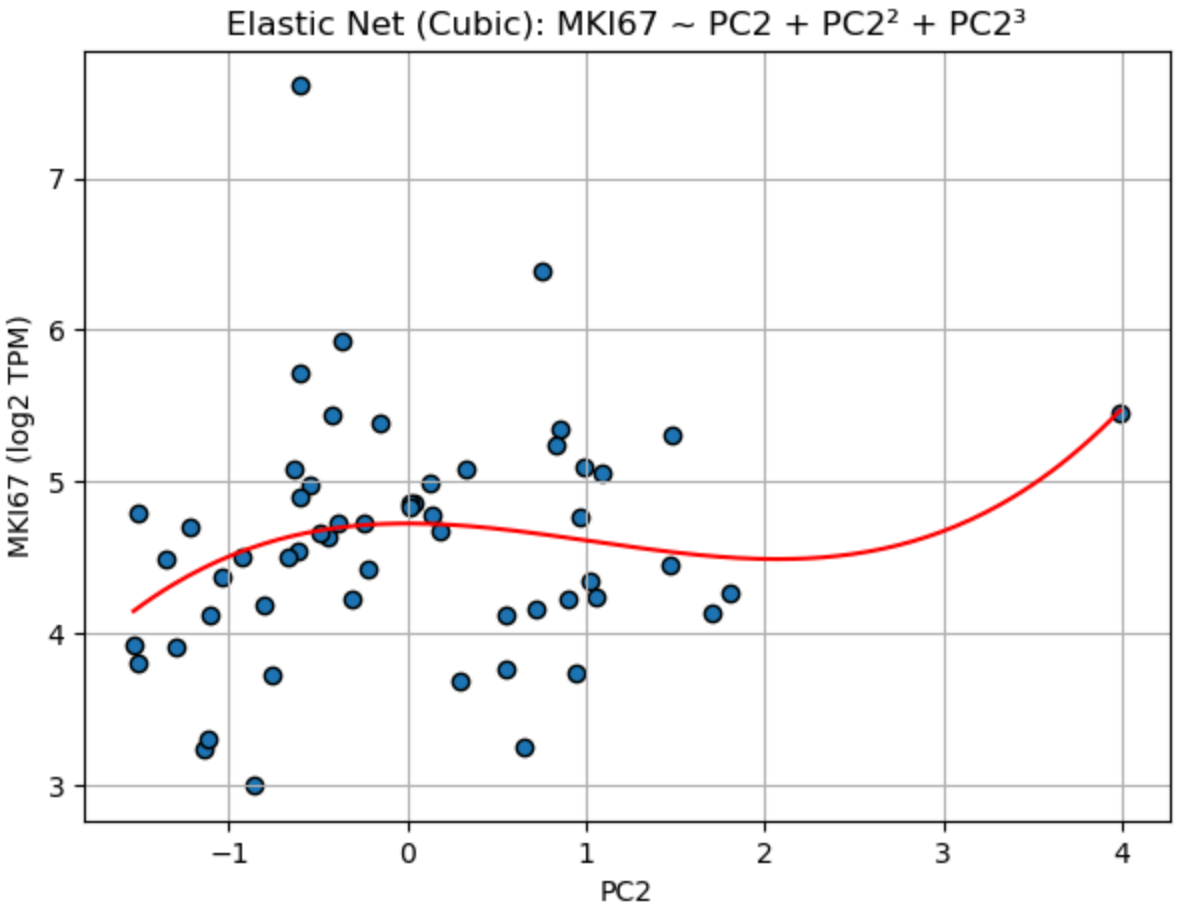
```
        plt.plot(x_line, model.predict(x_line_poly), color='red')
        plt.xlabel(pc_label)
        plt.ylabel("MKI67 (log2 TPM)")
        plt.title(f"Elastic Net (Cubic): MKI67 ~ {pc_label} + {pc_label}² +
        plt.grid(True)
        plt.show()

    return loadings
```

In [36]:
```
loadings = run_pca_ucs_elasticnet()
```

```
Cancer types available: ['BRCA' 'UCEC' 'KIRC' 'LUAD' 'LGG' 'THCA' 'HNSC' 'LU
SC' 'PRAD' 'COAD'
 'SKCM' 'OV' 'STAD' 'BLCA' 'LIHC' 'CESC' 'KIRP' 'LAML' 'GBM' 'READ' 'ACC'
 'KICH' 'UCS']
Found 57 UCS samples.
Expression data shape: (15716, 1802)
Matched 57 UCS samples.
MKI67 found in dataset. Will use for regression.

Explained variance ratio: [0.43749203 0.17694263]

=== PCA Loadings ===

          PC1    PC2
KRAS    0.261  0.406
NRAS    0.451  0.094
HRAS   -0.141  0.857
RAF1    0.484  0.196
MAP2K1  0.469 -0.012
MAPK1   0.504 -0.232

=== Elastic Net (Cubic) Regression: MKI67 ~ PC1 + PC1² + PC1³ ===
Best alpha: 0.1931
Best l1_ratio: 0.10
Coefficients: [0.142 0.107 0.023]
R² (on training data): 0.256
```




```
=== Elastic Net (Cubic) Regression: MKI67 ~ PC2 + PC2² + PC2³ ===
Best alpha: 0.2812
Best l1_ratio: 0.10
Coefficients: [ 0.    -0.164  0.053]
R² (on training data): 0.100
```



Adjusted Validation

When the elastic net regression was implemented, it yielded the following R^2 values. R^2 (on training data) for PC1: 0.256 R^2 (on training data) for PC2: 0.100 To validate the cubic elastic net regression, the fit was applied to the validation dataset. The results are shown below.

In [37]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score

# --- Coefficients from UCS Elastic Net (Cubic) regression ---
# Note: Elastic Net shrinks coefficients, but intercepts remain ~mean(MKI67)
intercept_PC1 = 4.429     # same as training intercept
coef_PC1 = 0.142
coef_PC1_sq = 0.107
coef_PC1_cu = 0.023

intercept_PC2 = 4.822     # same as training intercept
coef_PC2 = 0.000
coef_PC2_sq = -0.164
coef_PC2_cu = 0.053

# --- Load TEST dataset ---
```

```
meta_test = pd.read_csv("TEST_SET_GSE62944_metadata.csv")
expr_test = pd.read_csv("TEST_SET_GSE62944_subsample_log2TPM.csv", index_col

print("Expression data shape (test set):", expr_test.shape)
print("Metadata rows (test set):", meta_test.shape[0])

# --- Filter for UCEC samples ---
ucec_samples_test = meta_test[meta_test["cancer_type"].str.contains("UCEC",
print(f"UCEC samples in TEST metadata: {len(ucec_samples_test)}")

# --- Match expression samples ---
common_samples = expr_test.columns.intersection(ucec_samples_test)
expr_ucec = expr_test[common_samples].T
print(f"Matched UCEC samples in TEST expression matrix: {expr_ucec.shape[0]}

# --- Extract genes used for PCA ---
genes = ["KRAS", "NRAS", "HRAS", "RAF1", "MAP2K1", "MAPK1"]
subset_test = expr_ucec[[g for g in genes if g in expr_ucec.columns]]

# --- Scale expression data ---
X_test = StandardScaler().fit_transform(subset_test)

# --- Project onto PCA loadings from UCS training set ---
loadings = pd.DataFrame(
    [[0.261, 0.406],
     [0.451, 0.094],
     [-0.141, 0.857],
     [0.484, 0.196],
     [0.469, -0.012],
     [0.504, -0.232]],
    index=genes,
    columns=["PC1","PC2"]
)

Z_test = X_test.dot(loadings.values)
PC1_test = Z_test[:,0]
PC2_test = Z_test[:,1]

# --- Get true MKI67 values ---
mki67_test = expr_ucec["MKI67"]

# --- Predict MKI67 using Elastic Net (Cubic) UCS model ---
y_pred_PC1 = (intercept_PC1
              + coef_PC1 * PC1_test
              + coef_PC1_sq * (PC1_test**2)
              + coef_PC1_cu * (PC1_test**3))

y_pred_PC2 = (intercept_PC2
              + coef_PC2 * PC2_test
              + coef_PC2_sq * (PC2_test**2)
              + coef_PC2_cu * (PC2_test**3))

# --- Compute validation R² scores ---
r2_PC1 = r2_score(mki67_test, y_pred_PC1)
r2_PC2 = r2_score(mki67_test, y_pred_PC2)
```

```
print("\n=== VALIDATION RESULTS on UCEC dataset (Elastic Net Cubic Regressic
print(f"R² (MKI67 ~ PC1 + PC1² + PC1³): {r2_PC1:.3f}")
print(f"R² (MKI67 ~ PC2 + PC2² + PC2³): {r2_PC2:.3f}")
```

```
Expression data shape (test set): (15716, 1600)
Metadata rows (test set): 1600
UCEC samples in TEST metadata: 80
Matched UCEC samples in TEST expression matrix: 80

=== VALIDATION RESULTS on UCEC dataset (Elastic Net Cubic Regression) ===
R² (MKI67 ~ PC1 + PC1² + PC1³): 0.134
R² (MKI67 ~ PC2 + PC2² + PC2³): -0.168
```

Validation Results and Interpretation

The elastic net regression model built on UCS data showed only a modest association between PC1 (the MAPK pathway expression axis) and MKI67, with an R^2 of 0.26. When the same model was applied to an independent UCEC dataset, the R^2 dropped to 0.134, indicating that although the relationship remained positive, it explained only a small portion of proliferation behavior. The PC2-based model performed even worse, showing minimal predictive power in the training set (R^2 = 0.116) and a negative R^2 in the validation set (-0.168), confirming that PC2 does not generalize and likely reflects noise or subtype-specific variation rather than a meaningful proliferative signal.

These findings suggest that MAPK pathway expression alone is not a strong predictor of tumor proliferation in uterine cancers, and that additional pathways or regulatory mechanisms likely play a larger role. However, clinical and biological evidence still supports the presence of strong proliferative signaling in UCS tumors. A documented clinical case reported ~70% MKI67 staining in a UCS tumor, consistent with high growth activity despite the weak computational correlation. This aligns with the Hallmarks of Cancer concept in which sustained proliferative signaling is a defining feature of malignant cells, often involving—but not limited to—RAS/MAPK signaling.

Taken together, the results indicate that the weak statistical correlation reflects a limitation of the model rather than an absence of biological proliferation. The PCA-based approach likely failed to capture the full complexity of signaling interactions driving MKI67 expression, suggesting that future work should consider additional genes, nonlinear models, or pathway-level scoring rather than relying solely on variation in six MAPK genes.

Conclusions and Ethical Implications:

This study investigated whether variation in six RAS/MAPK pathway genes could predict tumor proliferation in uterine cancers using PCA and regression. The results showed only weak correlations between MAPK-related principal components and MKI67 expression in both the UCS training cohort and the UCEC validation set. While PC1 retained a small

positive association across datasets, the low R^2 values indicate that MAPK signaling alone explains only a limited portion of proliferative behavior, and that broader networks of pathways likely shape tumor growth. Clinical evidence of high MKI67 staining in UCS tumors further suggests that our simplified model does not fully capture the complexity of proliferation biology.

Ethically, these findings suggest several important considerations. First, simplified computational models can provide an incomplete or distorted view of complex biological systems, and it is essential to avoid overstating their accuracy or predictive power. Misinterpretation of weak correlations could lead to misguided scientific conclusions, especially in fields like oncology where research findings may eventually influence diagnostic or therapeutic approaches. Second, researchers have a responsibility to contextualize computational results with clinical evidence, acknowledging inconsistencies and uncertainty rather than forcing alignment between data and expectations. Third, this study highlights the ethical importance of model transparency: clearly communicating methodological limitations, dataset constraints, and analytical assumptions ensures that others can interpret the results responsibly. Finally, because cancer research ultimately impacts patient well-being, computational findings must be rigorously validated and integrated with biological understanding before being used to guide any real-world decision-making. These ethical considerations emphasize the need for careful, nuanced interpretation and further methodological refinement in future work.

Limitations and Future Work:

While our computational approach provided useful insights, there were several limitations that may have affected our results. We used principal component analysis (PCA) to reduce dimensionality and visualize trends before fitting a cubic regression model to test for nonlinear associations. However, the regression became overfit to the training data shown by a very low R2 value during validation. This indicates that the model failed to generalize and likely captured noise rather than meaningful biological relationships. Our dataset was also limited in size and diversity; as said in class, Professor Groves said she had limited our data from a very large dataset. For example, in the validation dataset, it did not include UCS samples, so we substituted UCEC data as the closest available alternative. While this allowed us to complete the analysis, UCEC may not accurately reflect the molecular or proliferative characteristics of UCS, introducing additional uncertainty into our findings. Another limitation was our reliance on PCA for dimensionality reduction, which only captures linear variance and may have overlooked complex nonlinear interactions between the RAS/MAPK pathway and MKI67 expression. For future work, expanding the analysis to include all available cancer types could better help determine whether MKI67 expression and RAS/MAPK signaling activity show stronger associations in other tumor contexts. Additionally, applying alternative modeling approaches, such as UMAP, or additional supervised learning methods could

improve our ability to detect subtle or indirect relationships. Lastly, seeing if UCS would appear different in our validation data in comparison to UCEC. Increasing the number and variety of samples would further strengthen the statistical power of the analysis and reduce overfitting, giving a clearer picture of whether MI67K channels have a measurable impact on cancer proliferation across different cancer types. Another direction for future research is to account for the progression of the tumor. Tumor progression over time can significantly alter both signaling pathway activity and proliferation markers like MKI67. Comparing early and late-stage tumors could reveal whether the relationship between RAS/MAPK signaling and proliferation strengthens, weakens, or changes direction as tumors evolve. Understanding how tumor age influences these pathways could provide valuable insight into when therapeutic interventions targeting RAS/MAPK might be most effective.

NOTES FROM YOUR TEAM:

10/23/25: The team did some baseline analysis of the data and thought about possible research questions based on growth signaling (our selected hallmark).

10/25/25: The team selected a research question, looked through the required data, and filled out the necessary checkpoint items.

11/03/25: The team ran PCA for Uterine Carcinosarcoma data and filled out the necessary checkpoint items.

11/07/25: Team furthered data analysis section and completed validation.

11/10/25: Team conducted quadratic and cubic regressions on the data, and furthered verification by implementing elastic net regression.

11/13/25: Team concluded project.

QUESTIONS FOR YOUR TA:

No further questions!