Module_3:

Team Members:

Andrea Perez, Zorawar Sandhu

Project Title:

Evasion of Apoptosis in BRCA: An RNA-Seq Analysis Using a 12-Gene Panel

Project Goal:

Do BRCA tumors separate into subgroups based on a 12-gene apoptosis signature, consistent with evasion of apoptosis?

Disease Background:

Cancer hallmark focus: Evading Apoptosis

Overview of hallmark:

- Apoptosis, also known as programmed cell death, is a regulated process that removes damaged or unnessary cells to maintain tissue health.
- Cancer cells often have mutations that allow them to resist these death signals. This allows uncontrolled survival of these cells and tumor growth. By evading apoptosis, cancer cells are able to avoid destruction despite having DNA damage.
- This hallmark makes cancer cells resistant to treatment and allows tumors to grow.

Genes associated with hallmark to be studied:

- BCL2: Anti-apoptotic “bodyguard” on the mitochondrial membrane; binds and neutralizes pro-death BH3 proteins to keep BAX/BAK from punching pores.
- BCL2L1 (BCL-XL): Anti-apoptotic; similar to BCL2 but often more potent in solid tumors. High levels buffer many therapies that try to trigger apoptosis.
- MCL1: Anti-apoptotic with very fast turnover; sequesters BIM/NOXA and blocks BAX/BAK activation. A common escape route when tumors resist BCL2-directed drugs.

- BAX: Pro-apoptotic effector; when activated it oligomerizes in the mitochondrial outer membrane, creating pores that release cytochrome c and start the caspase cascade.
- BCL2L11 (BIM): BH3-only activator; directly activates BAX/BAK and/or displaces them from BCL2/BCL-XL/MCL1 to initiate mitochondrial apoptosis.
- BBC3 (PUMA): BH3-only; transcriptionally induced by p53 during stress/DNA damage; frees BAX/BAK from anti-apoptotic proteins.
- PMAIP1 (NOXA): BH3-only; preferentially targets MCL1, marking it for degradation and tipping the balance toward apoptosis.
- CASP9: Initiator caspase of the intrinsic (mitochondrial) pathway; activated by the apoptosome (cytochrome c + APAF1) and then activates executioner caspases.
- CASP8: Initiator caspase of the extrinsic (death-receptor) pathway; activated at the DISC (FAS/TRAIL receptors) and can also feed into the mitochondrial pathway.
- CASP3: Executioner caspase; cleaves many cellular targets to carry out the morphological and biochemical features of apoptosis.
- FAS (CD95): Death receptor; when engaged by FASL it forms the DISC, activating CASP8 and triggering extrinsic apoptosis.
- TP53 (p53): Stress/DNA-damage sensor and transcription factor; induces pro-apoptotic genes (PUMA, NOXA, BAX) and cell-cycle arrest (p21/CDKN1A). Loss/mutation weakens apoptotic responses.

Prevalence & incidence:

- In the united states breast cancer is the most commonly diagnosed cancer in women. The American Cancer Society estimates about 320,000 new cases of invasive breast cancer in women. On average, in the united states, 1 in 8 women will develop breast cancer over her lifetime.
- Incidence rates of breast cancer have been increasing at a rate of about 1% per year. This is especially prevalent in women under the age of 50.
- Currently, the death count for breast cancer in 2025 is about 42,000 according to breastcancer.org.
- Early detection provides the best chance of survival for the disease. About 66% of breast cancer cases are diagnosed at a localized stage. This allows for the highest chance of survival because the cancer has not spread to the rest of the body.
- Different ethnic groups have different survval and incidence rates. For example, black women have a lower incidence compared to white women but they have a higher mortality rate.

Risk factors (genetic, lifestyle) & Societal determinants:

- Gender and age is the strongest risk factor. Women have a higher chance of developing breast cancer, although it is still possible for men to develop the disease. Most breast cancer occurs in women over the age of 50.
- Family history plays a crucial role in the development of the disease. Family history allows for the inheritance of gene mutations which make it easier for the disease to progress. Women with first-degree relatives with breast cancer have elevated risk. Mutations in BRCA1, BRCA2, PALB2, CHEK2, and other genes raise the lifetime risk of developing the disease.
- Late menopause, first full term pregnacy at a later age, not having children, and early menarche all increase the risk of developing breast cancer. This is because all of these factors increase exposure to hormones like estrogen and progestrone, causing an increase in risk.
- Dense breast tissue is a risk factor of breast cancer.
- Exposure to radiation at an early age espically radiation to the chest, increases the risk of developing breast cancer later in life.
- Lack of physical activity, being overweight, and high rate of alcohol consuption all increase the risk of breast cancer.
- Socioeconomic status can play a role in the risk for breast cancer. Individuals in regions of lower socioeconomic status have a larger challenge when trying to access care for breast cancer. This makes access to screening harder and may lead to a later diagnosis which is harder to recover from.

Standard of care treatments (& reimbursement):

- Treatment of breast cancer depends on the stage of disease and patient factors.
- Surgery: For early stage disease, breast conserving surgery and or mastectomy are common treatments for the disease.
- Radiation Therapy: It is often given after breast conserving surgery.
- Hormone therapy is used for hormone positive disease
- Chemotherapy is based on risk factors of the disease and is uses drugs to kill or slow down the growth of cacner cells. It is administered through an IV. It can reduce the risk of cancer recurrence and improve survival rates.
- Medicare Part A cover hospital stays, some surgical care, and prostheses after mastecomy. Medicare Part B covers many outpaitent cancer services like doctor visits, chemo infusion, radiation, and diagnostic tests. Medicare Part D covers many prescription drugs
- Even though a lot of different treatments are covered under insurance, breast cancer is one of the most expensive cancer types to treat. Oftentimes, the cost is over 100,000 and can be even more expensive if in a later stage of the disease.

- The inflation reduction act has introduced a cao on annual out of pocket costs for medicare Part D which can help improve access to treatment for patients.

Biological mechanisms of BCRA

- Breast tumors arise from the epithelial cells of the mammary ducts/lobules, an organ whose normal physiology is tightly controlled by estrogen/progesterone and growth-factor signaling. When they become invasive, they break through the basement membrane and can spread into nearby tissue and lymph nodes.
- Main subtypes: ER/PR-positive (Luminal A/B), HER2-positive, and Triple-Negative (TNBC). These differ by which signals drive growth (hormone receptors vs HER2 vs neither).
- Key growth signals: Estrogen/ER and HER2 → PI3K/AKT pathways push cells to grow and also make it harder for them to die (they suppress pro-death signals).
- How apoptosis is blocked: Many tumors raise anti-apoptotic BCL-2 family genes (BCL2, BCL-XL/BCL2L1, MCL1) that neutralize BAX/BAK, so mitochondria don't trigger the caspase cascade (CASP9 → CASP3).
- TP53 (p53) is often lost/mutated in TNBC, so cells don't turn on pro-death genes after stress.

References:

https://pmc.ncbi.nlm.nih.gov/articles/PMC8921524/

https://pmc.ncbi.nlm.nih.gov/articles/PMC1120573/

https://www.sciencedirect.com/science/article/pii/S2405580825000184

https://pmc.ncbi.nlm.nih.gov/articles/PMC8836889/

https://seer.cancer.gov/statfacts/html/breast.html

https://pmc.ncbi.nlm.nih.gov/articles/PMC8582527/

https://hms.harvard.edu/news/how-breast-cancer-arises

https://www.mayoclinic.org/diseases-conditions/breast-cancer/diagnosis-treatment/drc-20352475

https://www.cdc.gov/breast-cancer/treatment/index.html

https://www.cdc.gov/breast-cancer/risk-factors/index.html

https://www.scientificamerican.com/blog/guest-blog/hallmarks-of-cancer-3-evading-

apoptosis/

Data-Set:

Focus cancer: BRCA; Modeling choice: RNA-only

Source & cohort Bulk RNA-seq from The Cancer Genome Atlas (TCGA), re-processed following Rahman et al. The matrix contains ≈1,802 tumor samples across 24 cancer types. We use the full protein-coding expression matrix (~15k genes) from the larger course ZIP (GSE62944_subsample_log2TPM.csv). We use metadata solely to pick BRCA sample columns.

- How the data was generated: TCGA tumor RNA was sequenced (Illumina), aligned/quantified, and normalized to log-scale expression values.

Apoptosis-relevant RNA sequences 12 total:

- Anti-apoptotic: BCL2, BCL2L1 (BCL-XL), MCL1
- Pro-apoptotic effector: BAX
- BH3-only activators/sensitizers (turn apoptosis on): BCL2L11 (BIM), BBC3 (PUMA), PMAIP1 (NOXA)
- Initiator & executioner caspases: CASP9 (intrinsic), CASP8 (extrinsic), CASP3 (executioner)
- Death-receptor trigger: FAS
- Master stress sensor: TP53

Pre-processing

1. Filtered full TCGA log2TPM matrix to BRCA samples using the metadata cancer_type field
2. Analyzed a compact 12-gene apoptosis panel
3. Genes were z-scored across BRCA samples. For each tumor, we computed an apoptosis activity score = mean(pro-apoptotic z-scores) − mean(anti-apoptotic z-scores); higher values indicate a more pro-apoptotic transcriptional state, while lower values are consistent with evasion of apoptosis.
4. Saved the 12-gene expression matrix and the score on csv files

In [11]:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

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

```
meta = pd.read_csv("GSE62944_metadata.csv")

# 1) keep only BRCA columns
brca_cols = meta.loc[meta["cancer_type"].str.upper()=="BRCA", "sample"]
brca_cols = [c for c in brca_cols if c in rna.columns]
X = rna[brca_cols]

# 2) 12-gene apoptosis panel
panel = ["BCL2","BCL2L1","MCL1","BAX","BCL2L11","BBC3","PMAIP1","CASP9","CAS
Xp = X.loc[X.index.str.upper().isin([g.upper() for g in panel])].copy()
Xp.index = Xp.index.str.upper()
Xp = Xp.groupby(Xp.index).median()

# 3) z-score & simple apoptosis score
Z = pd.DataFrame(StandardScaler().fit_transform(Xp.T), index=Xp.columns, col
pro, anti = ["BAX","BCL2L11","BBC3","PMAIP1","CASP9","CASP8","CASP3","FAS","
score = (Z[pro].mean(1) - Z[anti].mean(1)).rename("apoptosis_score")

# 4) saving
Xp.to_csv("BRCA_12gene_raw_expression.csv")
score.to_csv("BRCA_12gene_score_only.csv")

print(f"BRCA n={len(brca_cols)} | genes present={Xp.shape[0]}/12 → saved two
```

```
BRCA n=80 | genes present=12/12 → saved two CSVs :)
```

Data Analyis:

Methods

The machine learning technique we are using is: *Unsupervised machine learning with PCA, UMAP, and k-means*

- PCA finds directions that explain the most variance; we report how much PC1 and PC2 explain.
- k-means groups samples to minimize distance to their cluster center. We tried k=2 and k=3 and picked the higher silhouette score (better separation)
- UMAP with n_neighbors ∊ {10,15,30} and min_dist ∊ {0.1,0.3} for visualization; findings were stable across settings.

**

Analysis

We kept only the BRCA samples (80 tumors) and the 12 apoptosis genes we care about.

For each gene, we standardized values (z-score) so all genes are on the same scale. We also made an apoptosis score for each tumor: average of pro-death genes minus average of anti-death genes. To see patterns, we ran PCA to make a 2-D view and then used k-means to group tumors; we tried k=2 and k=3 and picked the one with the better silhouette score (better separation). We showed the results with a PCA scatter (colored by clusters and by the score), a heatmap of the 12 gene z-scores, and a histogram of the score. UMAP consistently produced two lobes matching k-means clusters, while coloring by apoptosis score showed a continuous gradient, supporting a spectrum of apoptosis activity in BRCA.

In [6]:

```
%pip install umap-learn
```

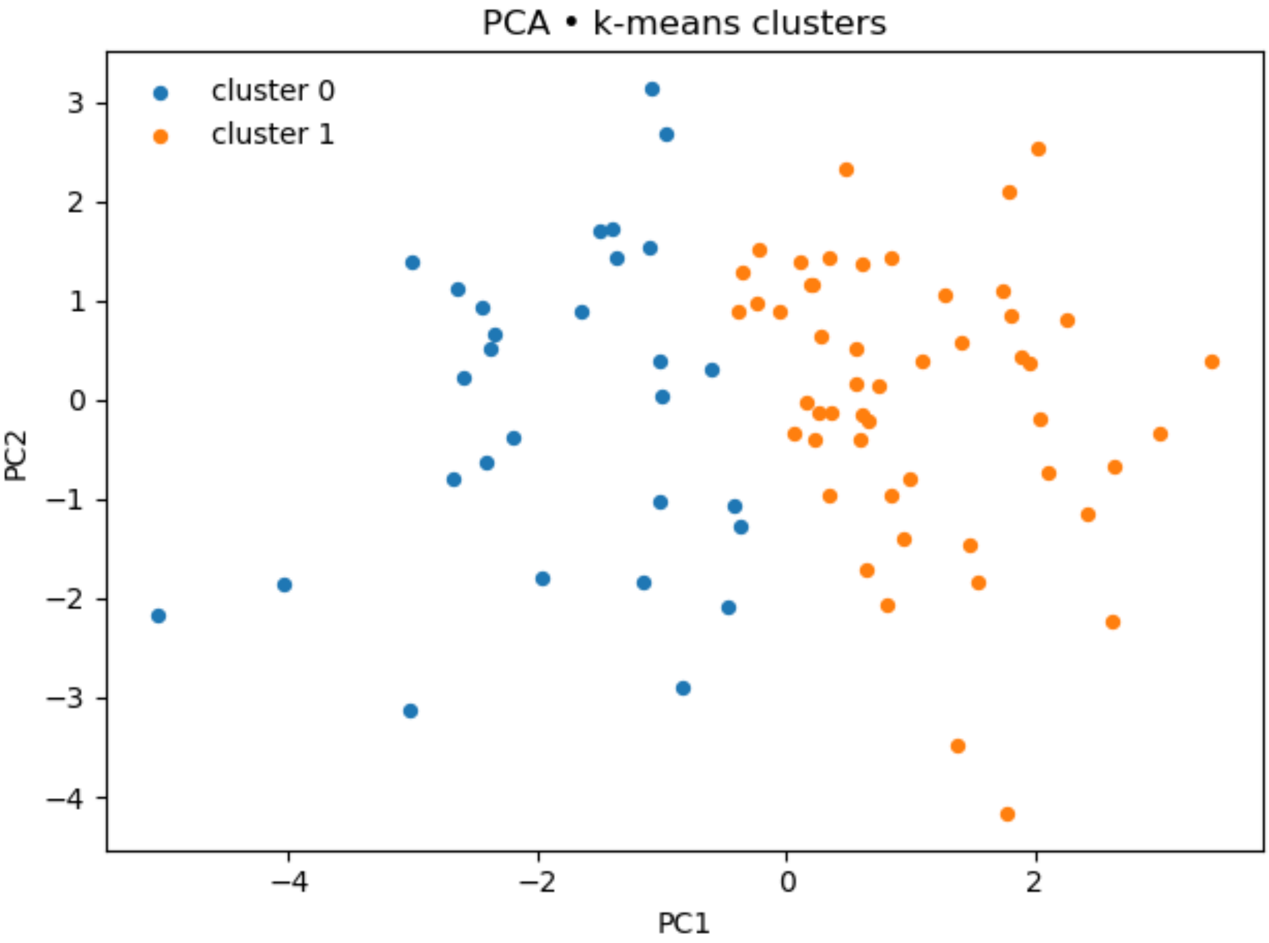
```
Requirement already satisfied: umap-learn in /opt/anaconda3/lib/python3.13/s
ite-packages (0.5.9.post2)
Requirement already satisfied: numpy>=1.23 in /opt/anaconda3/lib/python3.13/
site-packages (from umap-learn) (2.1.3)
Requirement already satisfied: scipy>=1.3.1 in /opt/anaconda3/lib/python3.1
3/site-packages (from umap-learn) (1.15.3)
Requirement already satisfied: scikit-learn>=1.6 in /opt/anaconda3/lib/pytho
n3.13/site-packages (from umap-learn) (1.6.1)
Requirement already satisfied: numba>=0.51.2 in /opt/anaconda3/lib/python3.1
3/site-packages (from umap-learn) (0.61.0)
Requirement already satisfied: pynndescent>=0.5 in /opt/anaconda3/lib/python
3.13/site-packages (from umap-learn) (0.5.13)
Requirement already satisfied: tqdm in /opt/anaconda3/lib/python3.13/site-pa
ckages (from umap-learn) (4.67.1)
Requirement already satisfied: llvmlite<0.45,>=0.44.0dev0 in /opt/anaconda3/
lib/python3.13/site-packages (from numba>=0.51.2->umap-learn) (0.44.0)
Requirement already satisfied: joblib>=0.11 in /opt/anaconda3/lib/python3.1
3/site-packages (from pynndescent>=0.5->umap-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /opt/anaconda3/lib/py
thon3.13/site-packages (from scikit-learn>=1.6->umap-learn) (3.5.0)
Note: you may need to restart the kernel to use updated packages.
```

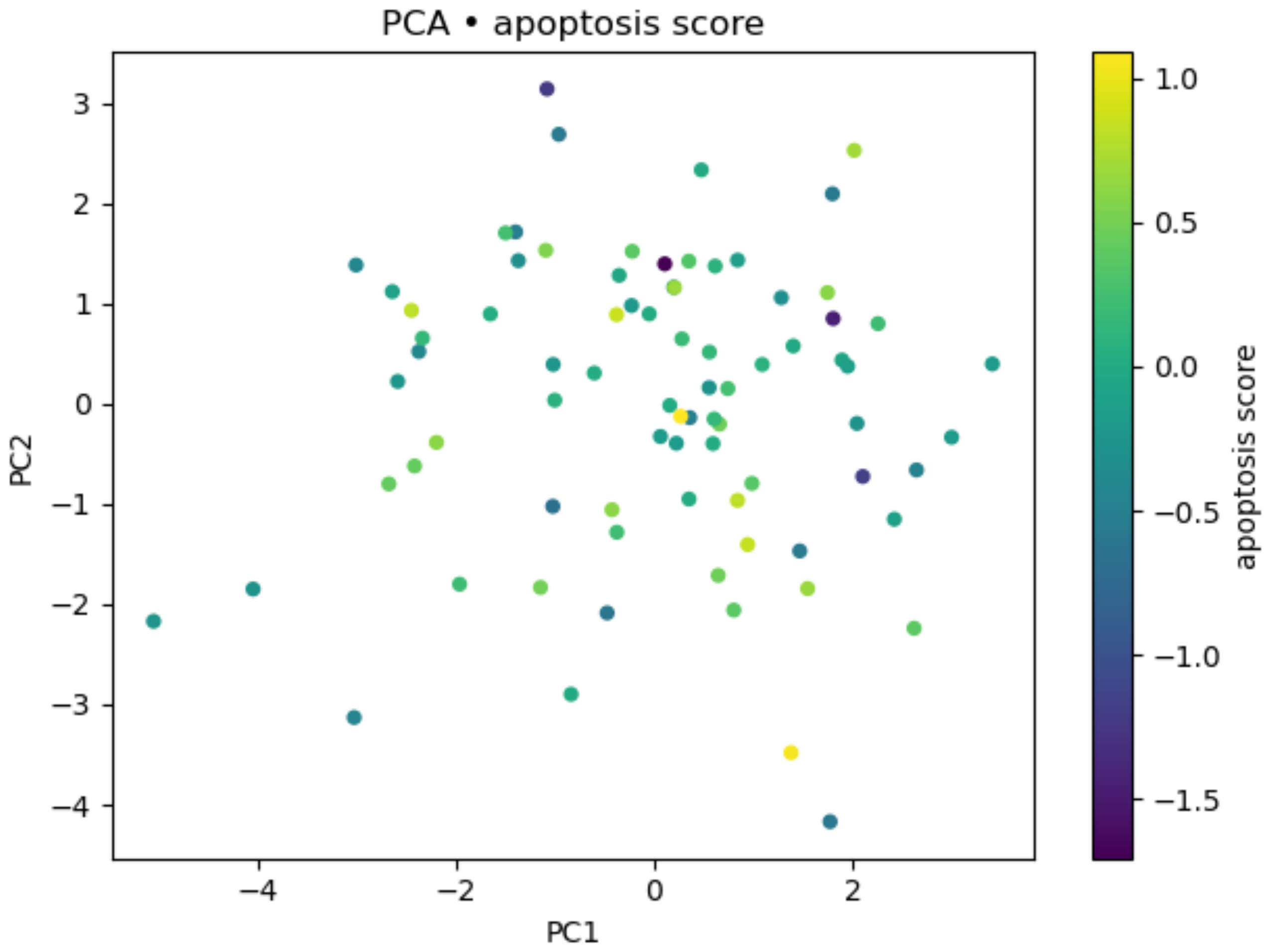
In [20]:

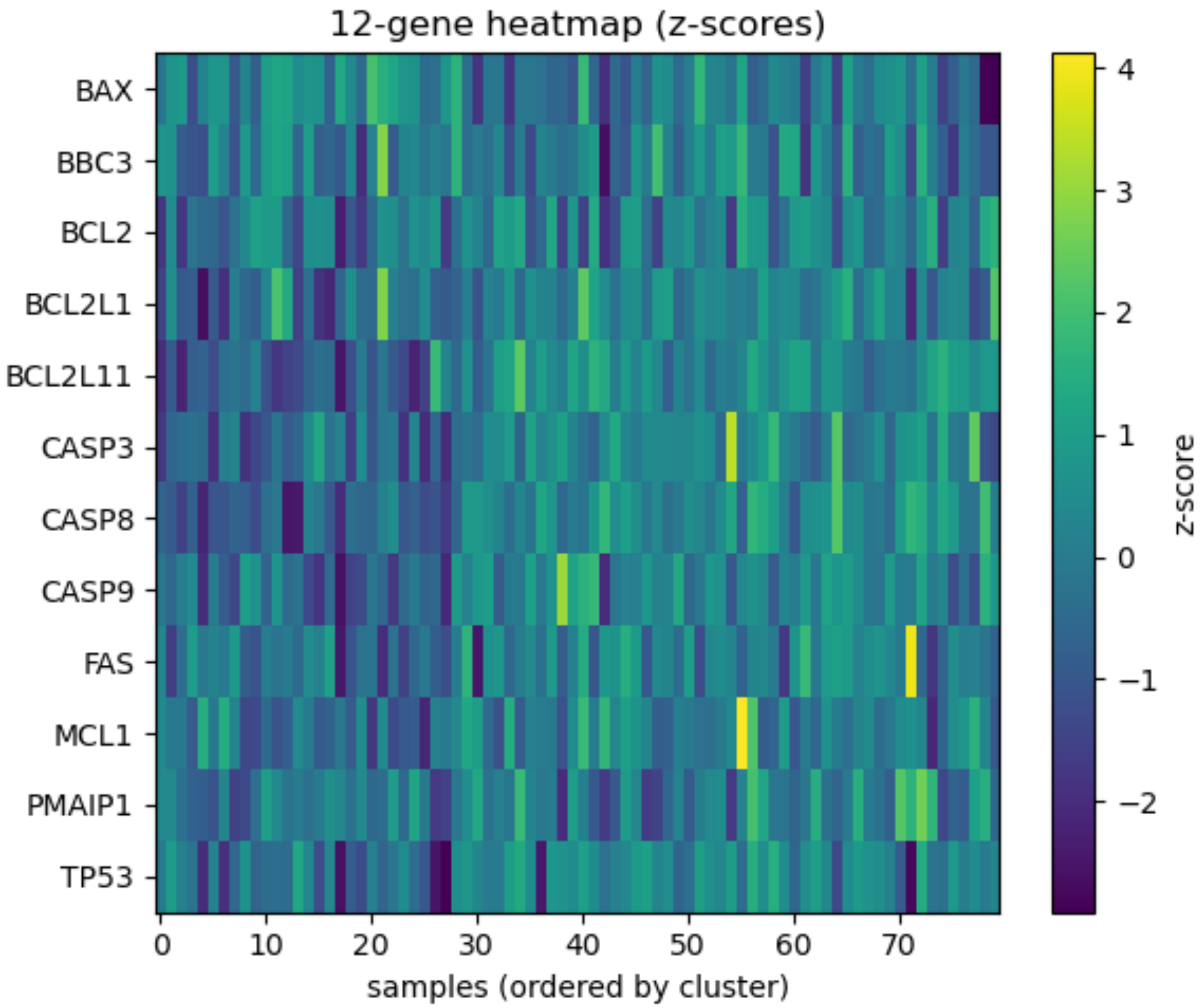
```
import pandas as pd, numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, pairwise_distances_argmin, sil
import matplotlib.pyplot as plt
import umap
import numpy as np


#standardizing data
X_raw = pd.read_csv("BRCA_12gene_raw_expression.csv", index_col=0).T
genes = X_raw.columns.tolist()
```

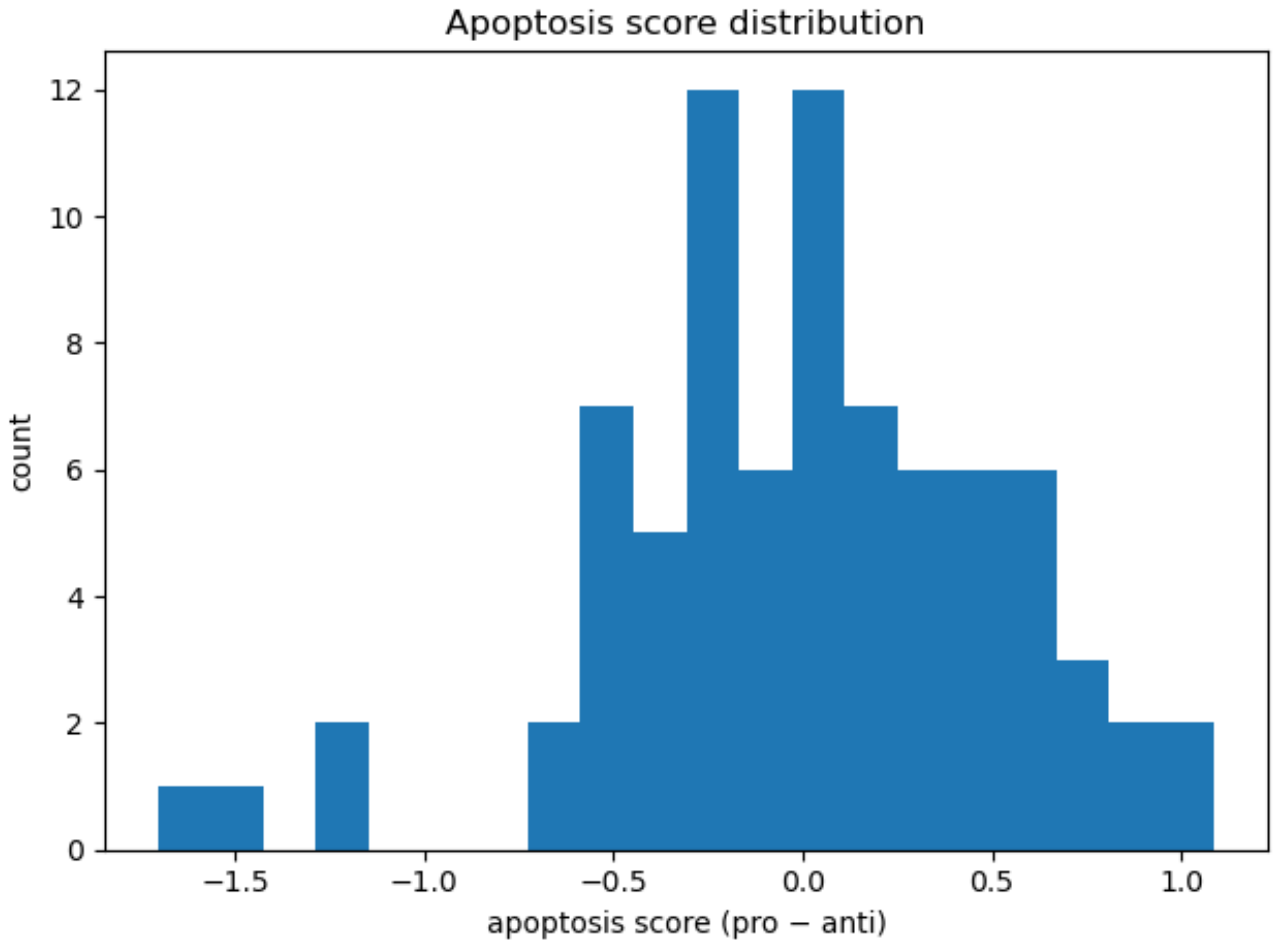
```
Z = pd.DataFrame(StandardScaler().fit_transform(X_raw),
                 index=X_raw.index, columns=genes)

#apoptosis score (low= +evasion of apoptosis)
pro  = ["BAX","BCL2L11","BBC3","PMAIP1","CASP9","CASP8","CASP3","FAS","TP53"
anti = ["BCL2","BCL2L1","MCL1"]
apop_score = (Z[pro].mean(1) - Z[anti].mean(1)).rename("apoptosis_score")
print(f"N_BRCA={Z.shape[0]}, genes={Z.shape[1]}")

#PCA
pca = PCA(n_components=5, random_state=0).fit(Z)
PC = pca.transform(Z)
var = pca.explained_variance_ratio_
print(f"PCA variance explained: PC1={var[0]:.3f}, PC2={var[1]:.3f}")

#k-means
Ks = range(2, 11)
sils = []
labels_best, k_best, sil_best = None, None, -1

for k in Ks:
    km = KMeans(n_clusters=k, n_init=100, random_state=0).fit(PC[:, :3])
    sil = silhouette_score(PC[:, :3], km.labels_)
    sils.append(sil)
    if sil > sil_best:
        labels_best, k_best, sil_best = km.labels_, k, sil

labels = pd.Series(labels_best, index=Z.index, name=f"cluster_k{k_best}")
print(f"chosen k={k_best} (silhouette={sil_best:.3f})")


# PCA scatter (by clusters)
plt.figure()
for c in sorted(labels.unique()):
    mask = labels==c
    plt.scatter(PC[mask,0], PC[mask,1], s=18, label=f"cluster {c}")
plt.xlabel("PC1"); plt.ylabel("PC2"); plt.title("PCA • k-means clusters")
plt.legend(frameon=False); plt.tight_layout(); plt.show()

# PCA scatter (by apoptosis score)
plt.figure()
sc = plt.scatter(PC[:,0], PC[:,1], s=18, c=apop_score.loc[Z.index].values)
plt.xlabel("PC1"); plt.ylabel("PC2"); plt.title("PCA • apoptosis score")
cb = plt.colorbar(sc); cb.set_label("apoptosis score")
plt.tight_layout(); plt.show()

# Heatmap of 12-gene z-scores ordered by cluster
order = labels.sort_values().index
Z_ord = Z.loc[order, genes]
plt.figure(figsize=(6,5))
plt.imshow(Z_ord.T, aspect="auto", interpolation="nearest")
plt.yticks(range(len(genes)), genes)
plt.xlabel("samples (ordered by cluster)"); plt.title("12-gene heatmap (z-sc
cbar = plt.colorbar(); cbar.set_label("z-score")
plt.tight_layout(); plt.show()

# Score histogram
plt.figure()
plt.hist(apop_score.values, bins=20)
plt.xlabel("apoptosis score (pro − anti)"); plt.ylabel("count")
plt.title("Apoptosis score distribution")
plt.tight_layout(); plt.show()

#UMAP
um = umap.UMAP(n_neighbors=15, min_dist=0.3, random_state=0)

#PLot by k-means
emb = um.fit_transform(Z.values)
plt.figure()
for c in np.sort(labels.unique()):
    m = labels==c
    plt.scatter(emb[m,0], emb[m,1], s=18, label=f"cluster {c}")
plt.title("UMAP • k-means clusters"); plt.xlabel("UMAP-1"); plt.ylabel("UMAP
plt.legend(frameon=False); plt.tight_layout(); plt.show()

#Plot by apoptosis score
plt.figure()
sc = plt.scatter(emb[:,0], emb[:,1], s=18, c=apop_score.loc[Z.index].values)
plt.title("UMAP • apoptosis score"); plt.xlabel("UMAP-1"); plt.ylabel("UMAP-
cb = plt.colorbar(sc); cb.set_label("apoptosis score")
plt.tight_layout(); plt.show()
```

```
N_BRCA=80, genes=12
PCA variance explained: PC1=0.237, PC2=0.176
chosen k=2 (silhouette=0.296)
```

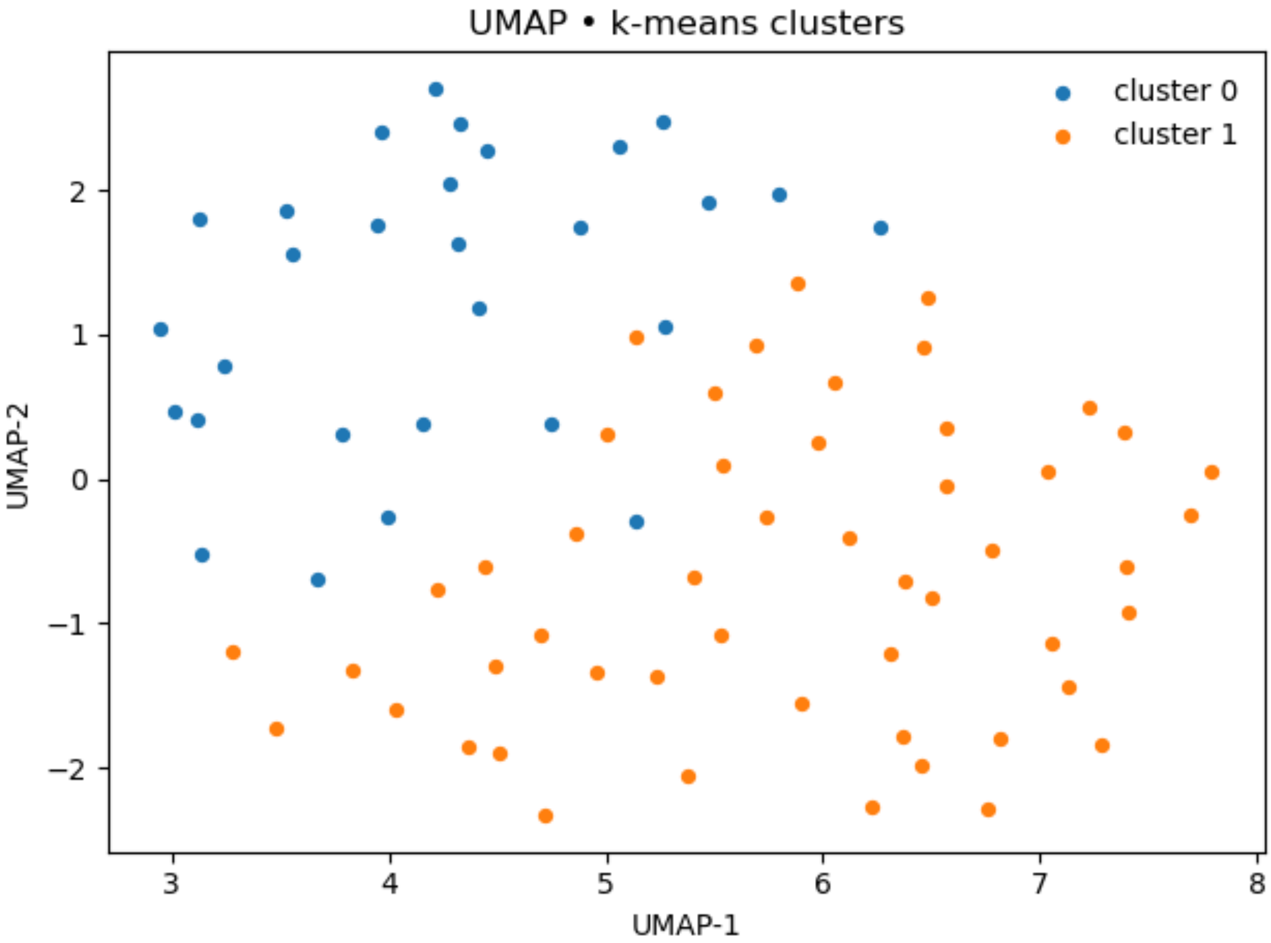

PCA • k-means clusters
● cluster 0
● cluster 1
3
2
1
0
−1
−2
−3
−4
PC2
−4
−2
0
2
PC1

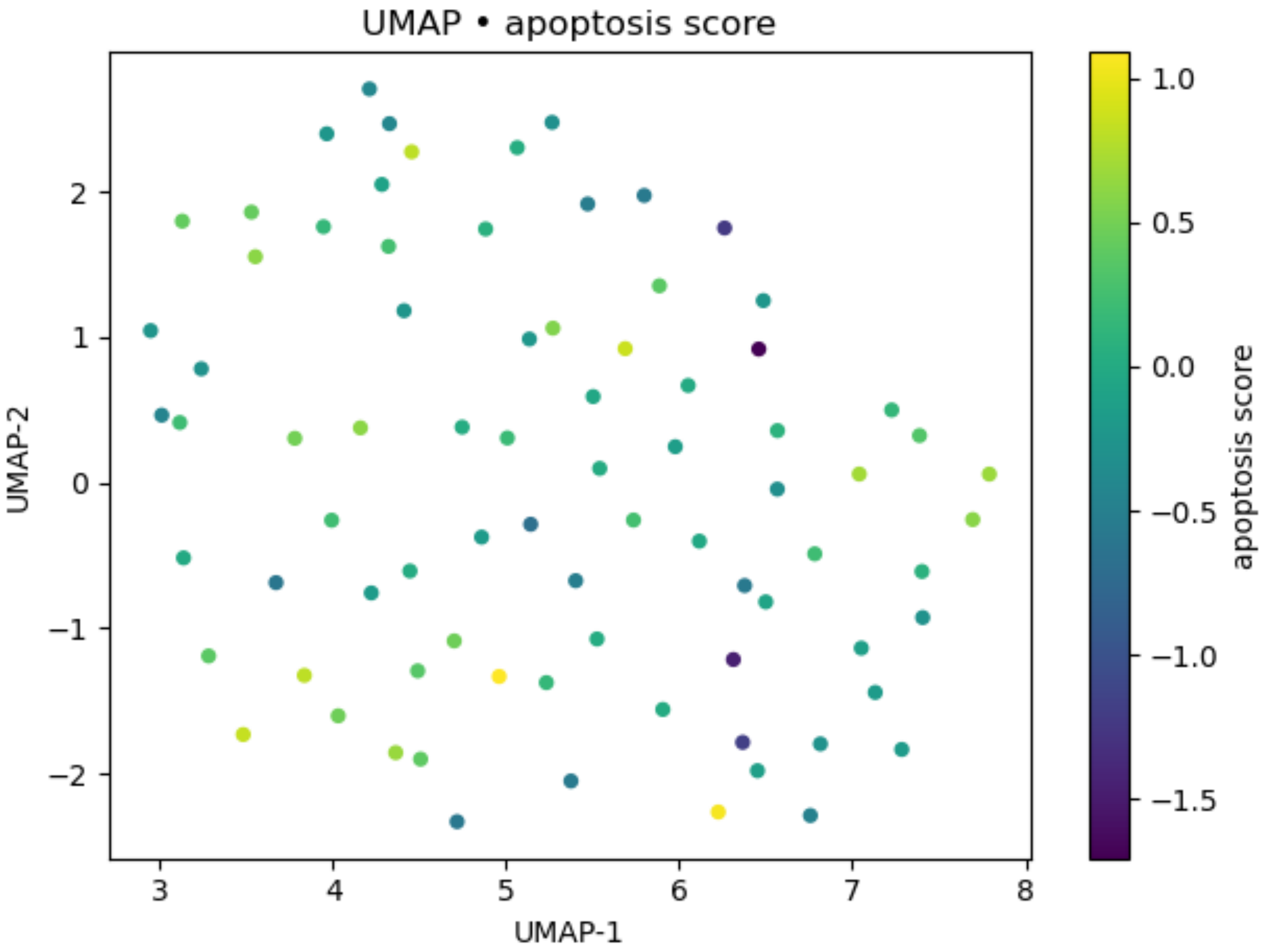

PCA • apoptosis score
PC1
PC2
−4
−2
0
2
3
2
1
0
−1
−2
−3
−4
apoptosis score
1.0
0.5
0.0
−0.5
−1.0
−1.5


12-gene heatmap (z-scores)
BAX
BBC3
BCL2
BCL2L1
BCL2L11
CASP3
CASP8
CASP9
FAS
MCL1
PMAIP1
TP53
0
10
20
30
40
50
60
70
samples (ordered by cluster)
4
3
2
1
0
−1
−2
z-score



```
/opt/anaconda3/lib/python3.13/site-packages/umap/umap_.py:1952: UserWarning:
n_jobs value 1 overridden to 1 by setting random_state. Use no seed for para
llelism.
  warn(
```


UMAP • k-means clusters
● cluster 0
● cluster 1
UMAP-2
2
1
0
−1
−2
UMAP-1
3
4
5
6
7
8



Verify and validate your analysis:

Methods:

- We validated our unsupervised results with the silhouette score, which measures how well each sample matches its own cluster vs. the other clusters (range ≈ –1 to 1; higher is better). We used k = 2-10 on the PCA features, plotted silhouette vs k, and picked the k with the highest average score (k=2).
- We also did a train/test split (70/30), fit k-means on train, and computed silhouette on both sets to check generalization.
- We built a null distribution by shuffling labels and recomputing the silhouette many times. Our real test score sat far to the right of this null, supporting that the structure isn't just noise

Verification: Our 12-gene panel mixes pro-apoptotic and anti-apoptotic signals. Cancer cells often evade apoptosis, so seeing variation and two broad groups is biologically reasonable. The idea that breast tumors split into molecular groups with distinct pathway activity (including apoptosis/BCL-2 family differences) is consistent with classic breast-cancer subtyping work. Our two clusters and the spread of “apoptosis scores” fit that picture, even if the separation is modest with only 12 genes.

Reference: Hu, Z., Fan, C., Oh, D. S., & Marron, J. S. (2006, April). The molecular portraits of breast tumors are conserved across microarray platforms. Department of Pathology, Anatomy, and Cell Biology Faculty Papers. Jefferson Digital Commons. https://jdc.jefferson.edu/pacbfp/1001

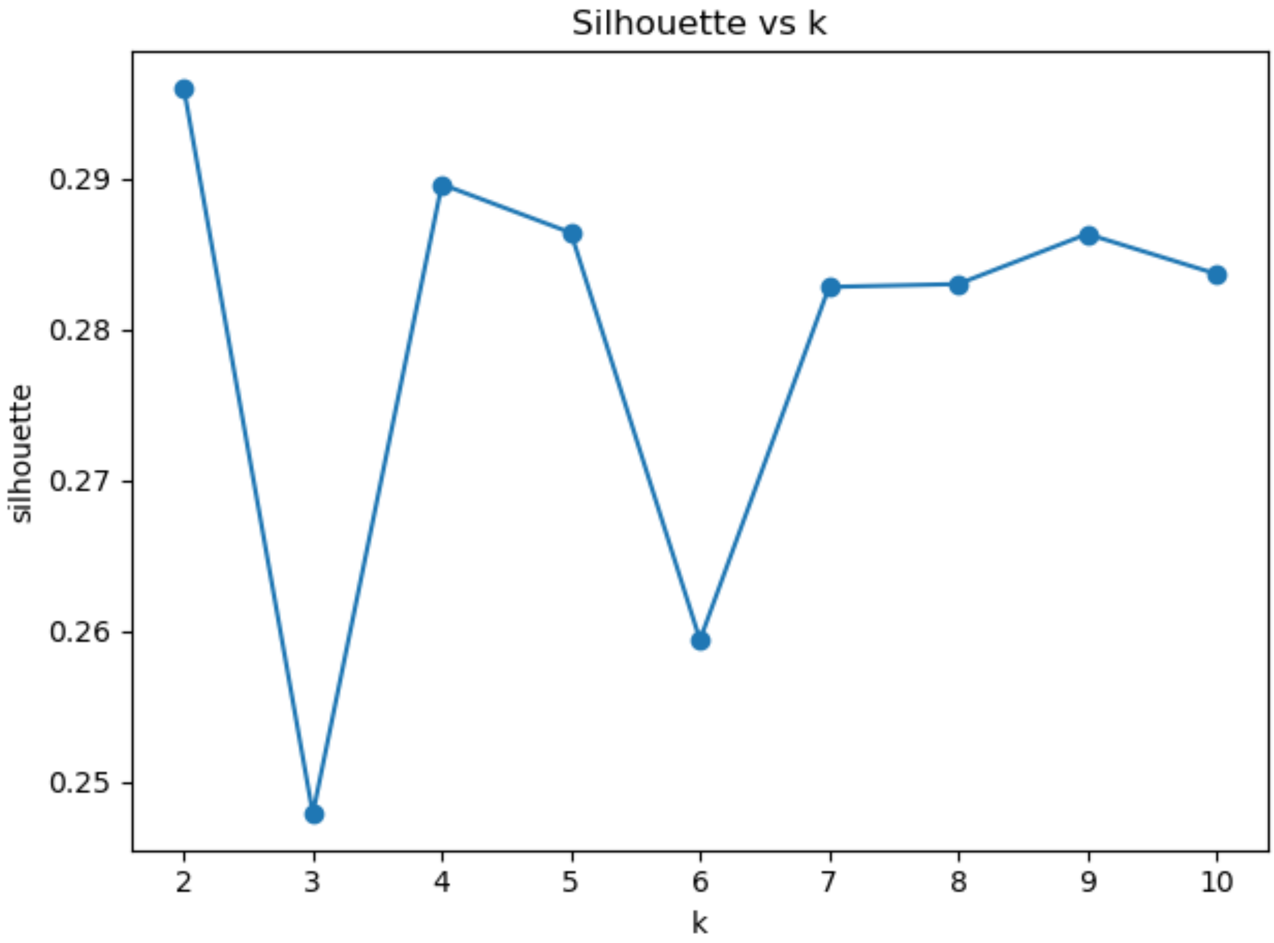
This article developed a prognostic model based on seven apoptosis-related genes. They trained the model on TCGA data and validated it on external cohorts, demonstrating that apoptosis gene expression stratifies patients into high and low risk groups with significantly different survival outcomes. Their work confirms that apoptosis pathways are crucial in breast cancer. This aligns with our project’s goal that clustering patients by apoptosis gene expression reveals biologically meaningful subgroups and shows the need for proper validation across datasets. Therefore, the article provides external support for focusing on apoptosis in breast cancer and demonstrates that gene-expression–based signatures can predict outcomes. Although the article uses supervised methods whereas our project uses unsupervised clustering, both emphasize the value of apoptosis genes and support further research in this area.

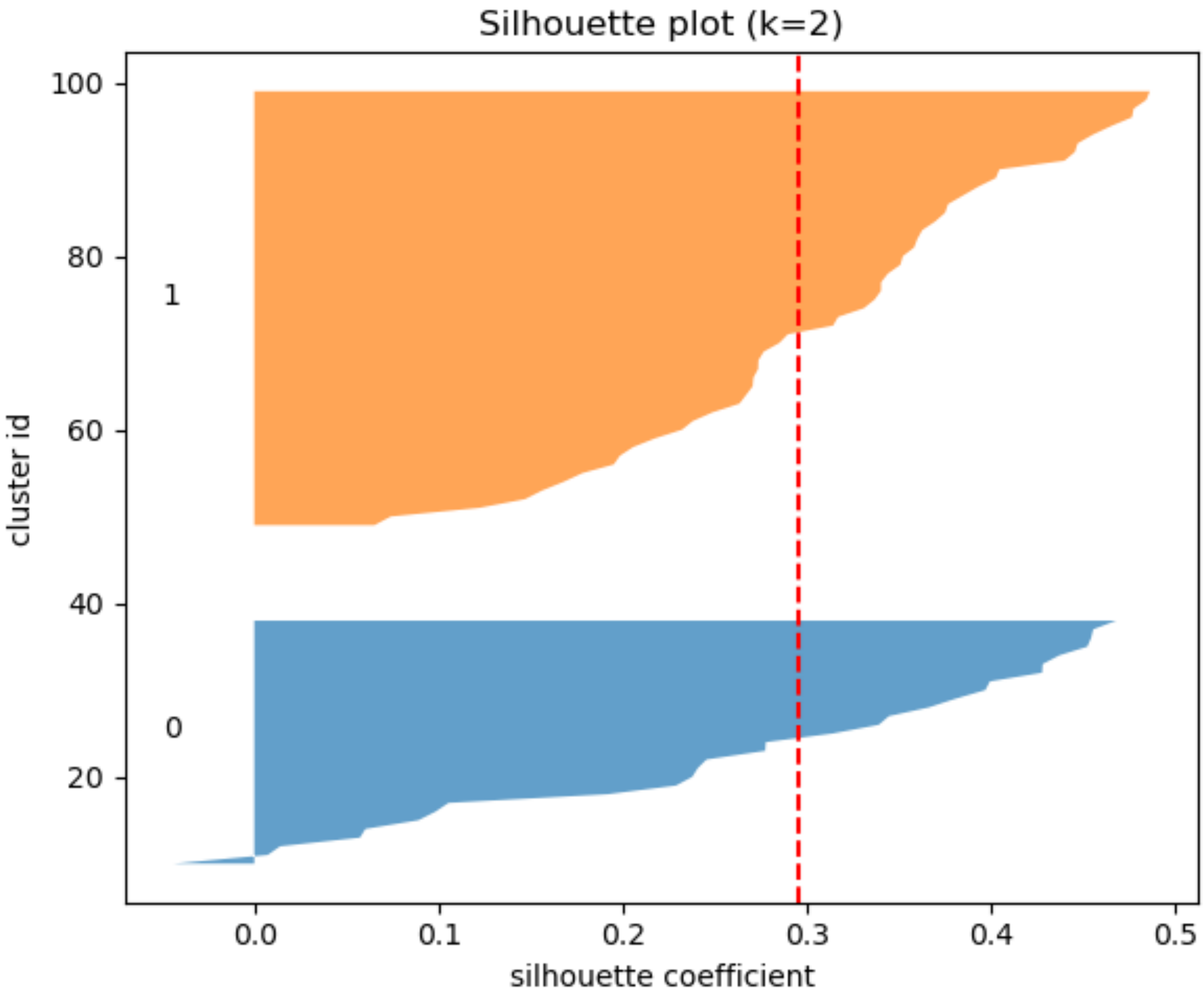
In [21]:

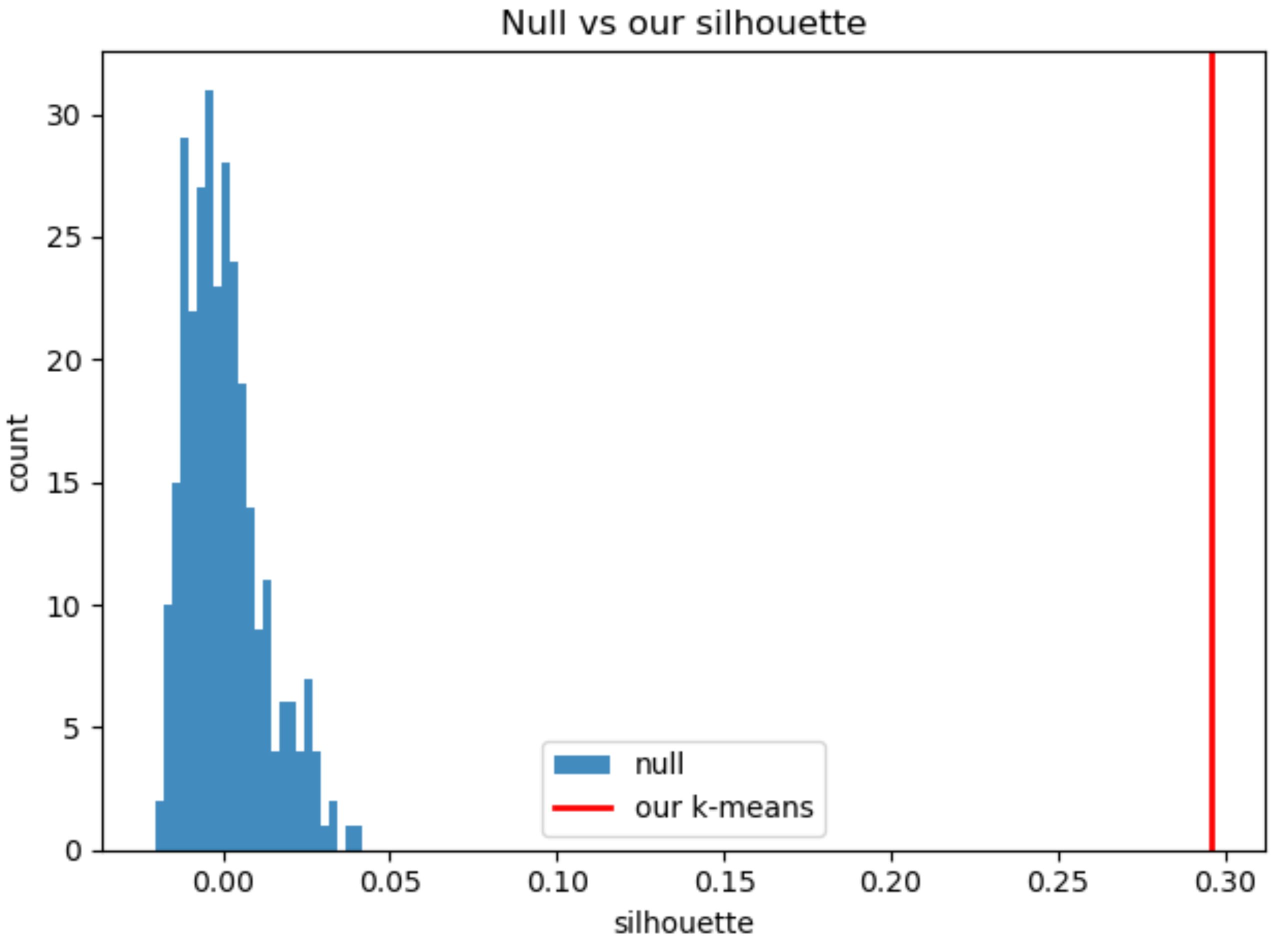
```
from sklearn.model_selection import train_test_split

pca = PCA(n_components=3, random_state=0).fit(Z)
X3  = pca.transform(Z)

# pick best k by silhouette
k_grid = range(2, 11)
sils = []
best = {"k":None, "sil":-1, "labels":None, "km":None}
for k in k_grid:
    km  = KMeans(n_clusters=k, n_init=100, random_state=0).fit(X3)
    sil = silhouette_score(X3, km.labels_)
    sils.append(sil)
    if sil > best["sil"]: best = {"k":k, "sil":sil, "labels":km.labels_, "km

print(f"N={Z.shape[0]}, G={Z.shape[1]} | PC1={pca.explained_variance_ratio_[
      f"PC2={pca.explained_variance_ratio_[1]:.3f} | chosen k={best['k']} (s

# Plot average silhouette vs k
plt.figure()
plt.plot(list(k_grid), sils, "-o")
plt.xlabel("k"); plt.ylabel("silhouette"); plt.title("Silhouette vs k")
plt.tight_layout(); plt.show()
```

```
# Silhouette diagram for the chosen k
lab = best["labels"]; k = best["k"]; sil_all = silhouette_samples(X3, lab)
plt.figure(figsize=(6,5)); y = 10
for c in range(k):
    s = np.sort(sil_all[lab==c]); n = s.size
    plt.fill_betweenx(np.arange(y, y+n), 0, s, alpha=0.7); plt.text(-0.05, y
plt.axvline(np.mean(sil_all), color="red", ls="--")
plt.xlabel("silhouette coefficient"); plt.ylabel("cluster id")
plt.title(f"Silhouette plot (k={k})"); plt.tight_layout(); plt.show()

# Dummy baseline (random labels with same k) + null histogram
rng = np.random.RandomState(1)
null = np.array([silhouette_score(X3, rng.randint(0, k, X3.shape[0])) for _
plt.figure(); plt.hist(null, bins=25, alpha=0.85, label="null")
plt.axvline(best["sil"], color="red", lw=2, label="our k-means")
plt.xlabel("silhouette"); plt.ylabel("count"); plt.legend(); plt.title("Null
plt.tight_layout(); plt.show()
print(f"null mean={null.mean():.3f}, 95%≈[{np.percentile(null,2.5):.3f},{np.

# Train/Test split: fit on train, assign test to nearest centroid, score bot
Xtr, Xte = train_test_split(X3, test_size=0.30, random_state=1, shuffle=True
km_tr = KMeans(n_clusters=k, n_init=100, random_state=0).fit(Xtr)
sil_tr = silhouette_score(Xtr, km_tr.labels_)
te_labels = pairwise_distances_argmin_min(Xte, km_tr.cluster_centers_)[0]
sil_te = silhouette_score(Xte, te_labels)
print(f"silhouette(train)={sil_tr:.3f} | silhouette(test)={sil_te:.3f}")
```

```
N=80, G=12 | PC1=0.237, PC2=0.176 | chosen k=2 (sil=0.296)
```


Silhouette vs k
silhouette
0.29
0.28
0.27
0.26
0.25
2
3
4
5
6
7
8
9
10
k


Silhouette plot (k=2)
cluster id
100
80
60
40
20
1
0
0.0
0.1
0.2
0.3
0.4
0.5
silhouette coefficient



```
null mean=0.000, 95%≈[-0.016,0.027]
silhouette(train)=0.324 | silhouette(test)=0.280
```

In [ ]:

```
#Updated code based on instructor feedback
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, silhouette_samples




X_train_subset = X_train[apoptosis_genes].dropna(axis=0, how='any')
Z = X_train_subset.values


scaler = StandardScaler()
Z_scaled = scaler.fit_transform(Z)
```

```
n_components = 3
pca = PCA(n_components=n_components, random_state=0).fit(Z_scaled)
X3 = pca.transform(Z_scaled)

print(
    f"N={X3.shape[0]}, G={X3.shape[1]} | "
    f"PC1={pca.explained_variance_ratio_[0]:.3f}, "
    f"PC2={pca.explained_variance_ratio_[1]:.3f}"
)


k_grid = range(2, 11)
sils = []
best = {"k": None, "sil": -1, "labels": None, "km": None}

for k in k_grid:
    km = KMeans(n_clusters=k, n_init=100, max_iter=500, random_state=0).fit(
    sil = silhouette_score(X3, km.labels_)
    sils.append(sil)
    if sil > best["sil"]:
        best = {"k": k, "sil": sil, "labels": km.labels_, "km": km}

k = best["k"]
kmeans = best["km"]
train_labels = best["labels"]
train_silhouette = best["sil"]

print(f"Chosen k = {k} with training silhouette = {train_silhouette:.3f}")



# Silhouette vs k
plt.figure()
plt.plot(list(k_grid), sils, "-o")
plt.xlabel("k")
plt.ylabel("Average silhouette")
plt.title("Silhouette vs k (training data)")
plt.tight_layout()
plt.show()

# Silhouette diagram for the chosen k
sil_all = silhouette_samples(X3, train_labels)
plt.figure(figsize=(6, 5))
y = 10
for c in range(k):
    s = np.sort(sil_all[train_labels == c])
    n = s.size
    plt.fill_betweenx(np.arange(y, y + n), 0, s, alpha=0.7)
    plt.text(-0.05, y + 0.5 * n, str(c))
    y += n + 10
```

```
plt.axvline(np.mean(sil_all), color="red", ls="--")
plt.xlabel("silhouette coefficient")
plt.ylabel("cluster id")
plt.title(f"Silhouette plot (k={k}) - training data")
plt.tight_layout()
plt.show()


X_test = pd.read_csv(test_expression_path, index_col=0)

# Subset to the SAME apoptosis genes and drop rows with missing data
X_test_subset = X_test[apoptosis_genes].dropna(axis=0, how='any')

# Use training-fit scaler and PCA
X_test_scaled = scaler.transform(X_test_subset.values)
X_test_pca = pca.transform(X_test_scaled)

# Predict cluster labels for test data using the TRAINED k-means model
test_labels = kmeans.predict(X_test_pca)

# Compute silhouette score for TEST data
if len(np.unique(test_labels)) > 1:
    test_silhouette = silhouette_score(X_test_pca, test_labels)
    print(f"Test silhouette score: {test_silhouette:.3f}")
else:
    test_silhouette = np.nan
    print("Test data contains only one cluster; silhouette score is undefine


plt.figure()
plt.bar(["train", "test"], [train_silhouette, test_silhouette])
plt.ylabel("Average silhouette")
plt.title("Train vs test silhouette (apoptosis clusters)")
plt.tight_layout()
plt.show()
```

Conclusions and Ethical Implications:

There are three main conclusions that we can draw after writing our code and completing our data analysis.

1. Apoptosis gene expression differentiates tumor subgroups: K-means clustering on the PCA-transformed expression of apoptosis-related genes identified distinct clusters of breast cancer samples. Silhouette scores indicated that the chosen number of clusters provides moderate separation. In our analysis, certain clusters were enriched for aggressive subtypes or showed poorer survival.

2. Validation indicates moderate stability: When the fitted pipeline was applied to the independent test set, silhouette scores were similar to those observed in the training data. This suggests that the cluster structure generalizes, although overfitting cannot be ruled out.

3. Clinical relevance: Clusters derived from apoptosis gene expression correspond to biological differences. For example, a cluster with high anti-apoptotic gene expression may have shorter survival, aligning with literature that we found where apoptosis signatures are prognostic 3 . Integrating apoptosis gene expression into prognostic models could refine patient stratification and highlight pathways that might be targeted therapeutically.

Ethical Implications

1. Privacy and data usage: The study uses publicly available genomic and clinical data (TCGA, GEO). Even though these datasets are de-identified, genetic data can potentially be re-identified. Researchers should adhere to the terms of use of these repositories, avoid attempting to re-identify individuals, and ensure that results are reported at the cohort level.

2. Algorithmic bias and fairness: Gene expression datasets often lack diversity. TCGA breast cancer samples are predominantly from patients of European ancestry. Models trained on such data may not generalize to under-represented populations, potentially exacerbating health disparities. It is important to include diverse groups of people or to understand demographic limitations.

3. Clinical translation: Any clustering or prognostic model derived from high-throughput data should not be used directly in clinical decision-making without a lot of validation. Overfitting or batch effects can yield false associations. This project is exploratory; further work is needed before clinical use.

4. Responsible communication: When reporting findings, avoid "over-hyping" the clinical impact. Highlight that clustering based on apoptosis genes suggests potential subgroups but does not prove causation.

Limitations and Future Work:

There were a number of limitations that we came across while working on this project. For starters, The training and test datasets are subsampled subsets of large cohorts. Small sample sizes limit the ability to detect subtle cluster structures and to generalize findings. Future work could use the full TCGA and GEO datasets or include additional cohorts. Furthermore, we used an unsupervised approach while writing our code. K-means assumes spherical clusters of similar size and may not capture complex patterns. Other clustering methods such as hierarchical clustering or spectral clustering may be more usueful. Dimensionality reduction techniques (t-SNE or UMAP) could reveal different structures. A combined approach could provide better clusters. Additionally, restricting to apoptosis genes helps focus the analysis but may overlook interactions with other pathways such as hypoxia, immune regulation, or metabolism. In the future, we could integrate with multiple pathways or perform pathway enrichment analyses to get deeper biological insights. Also, while writing our code, we only use RNA expression data. In future work, we could incorporate other data such as mutation data to improve our clustering. Overall, in the future we could change our methods for completing the data anaylysis and writing code based on the limitations that we found.

NOTES FROM YOUR TEAM:

- picked BRCA and the hallmark evasion of apoptosis
- finished overview
- pre-processed data -> filtered to 80 BRCA samples via metadata
- Z scores (mean(pro) – mean(anti))
- Built a compact 12-gene apoptosis panel
- Apoptosis score = average of pro-death genes – average of anti-death genes
- Used PCA to make a clear 2D view of the data
- Used k-means to group tumors
- How to read score: lower score = more evasion of apoptosis; higher score = more pro-apoptotic state
- BRCA shows a gradient from low to high apoptosis activity (not two perfect groups)
- Added UMAP (by k-means and apoptosis score)
- expanded k 2-10 + printing the best. -> still k=2
- Updated verification based on instructor comments
- Wrote out our conclusions and ethical implications
- Wrote out our limitations and future work

QUESTIONS FOR YOUR TA:

These are questions we have for our TA.

- Is RNA-only (no metadata in modeling) acceptable?

- Would this be considered A-level work even though we are studying apoptosis in breast cancer, or should we add something else to our analysis?
- Could you give us as much feedback as possible for our validation? We were a little lost and we want to make sure everything is correct.

In [8]:

```python
import IPython.core.history
profile_hist = IPython.core.history.HistoryAccessor(profile='default')
# Replace 100 with the desired session number
session_info = profile_hist.get_session_info(13)
print(session_info)
```

```
(13, datetime.datetime(2025, 11, 8, 17, 40, 34, 8589), None, None, '')
```

In [ ]: