Module_3: *(Template)*

Team Members:

Kayla and Marielle

Project Title:

Investigation into genetic causes for metastasis in uterine carcinosarcoma (UCS)

Project Goal:

This project seeks to better understand the genetic mutations and pathways that lead to the development of UCS and how to find potential drug therapies that can reduce the effects of metastasis that is prevalent in this type of cancer.

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.

- Cancer hallmark focus: tissue invasion and metastasis
- Overview of hallmark:
  - metastasis is defined as when tumor cells migrate and develop in a different region than which they originated from
  - this can make cancer significantly more dangerous as it allows mutations to take more control of the body
- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate):
  - ID1 has a strong correlation with angiogenesis and cell growth/ proliferation
    - often seen in metastasized tumors because it helps transcription factors travel through the blood stream.
    - this has been seen in a diverse set of cancer types, but not specifically UCS due to the limited nature of research on uterine cancers.
  - ANGPTL4 enhances tumor migration and invasion
    - promotes angiogenesis -> cancer cells are typically rich in ANGPTL4 so they can take control of the bodies blood supply

sources:

- NIH - ID1 contributes to cell growth invasion and migration in salivary adenoid cystic carcinoma
- National Cancer Institute - Metastasis
- NIH - Potential role of ANGPTL4 in cancer progression, metastasis, and metabolism

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
  - incidence: 1.36/100,000 women (very rare, leading to it being understudied)
  - more commone in older women, with the most frequent diagnosis being in lower-mid 70s
  - UCS makes up less than 5% of all uterine malignancies
  - Roughly only 35% of patients survive five years after diagnosis
- Risk factors (genetic, lifestyle) & Societal determinants
  - there is little data for what causes/ increases chances of getting this specific form of uterine cancer
  - women who are at higher risk for general uterine cancer are those from ages 50-70, overweight or obese, and have higher level of estrogen or take supplements to increase estrogen levels.
- Standard of care treatments (& reimbursement)
  - there is no standard course of treatment of UCS
  - the best procedure/ option given is surgical removal of tumors
    - chemo is recommeneded to due the high chance of metastasis
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)
  - the pathology of UCS is largely unknown however animal testing is being done to further understand the tumor enviornent of unterine cancer at large.
  - monoclonal origin theory states that UCS tumors likely started out as one type of tumer (carcinoma) and transformed/ mutated into a mix of both carcinoma and sarcomas
  - stems from epithelial precursor and quickly takes over and metastasizes

sources:

- NIH - Uterine Carcinosarcoma Study
- SGO - Uterine Cancer Risk Factors
- NIH - The biology of uterine sarcomas

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 contains comprehensive clinical and survival information for 1,802 cancer patients for 24 cancer types. It includes demographic data (age, gender, and ethnicity), tumor characteristics (stage, grade, and histology), treatment history, and survival outcomes measured in days. Furthermore, it includes the top variance genes and 50 to 100 tumors per cancer type.
- The data used for this project was collected by obtaining the raw sequencing data for 9264 tumor samples and 741 samples across 24 cancer types, which were then reprocessed using a Subread algorithm. Then, from there, the data was cleaned up by using pandas to create a TCGA pan-cancer subset
- There is a mix of units, such as time (i.e. days and years) and counts per microliter.
- source(s): *Cancer Genome Atlas RNA-seq dataset re-processed by Rahman et Al and Module 3 - Lecture 2 - 2025.pptx*

Data Analyis:

Methods

The machine learning technique I am using is: dimensionality reduction

*What is this method optimizing? How does the model decide it is "good enough"?*We are using this in order to find patterns in the data between UCS and the prescense of genes related to angiogenesis. This technicque is best for this particular data set because it is able to identify clusters of large sets of data (for example all of the genes in the data set) which can allow us to identify patterns. This model is also able to simplify the data while preserving the importance of key trends/ dimensions.

**

Analysis

(Describe how you analyzed the data. This is where you should intersperse your Python code so that anyone reading this can run your code to perform the analysis that you did, generate your figures, etc.)

We looked into which angiogenisis related genes were present in the data. Then we used the machine learning technique of dimensionality reduction and PCA

```
# Exploratory data analysis (EDA) on a cancer dataset
# Loading the files and exploring the data with pandas
# %%
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# %% Load the data
```

```
#####################################################
data = pd.read_csv(
    '/Users/kaylac/Downloads/GSE62944_subsample_topVar_log2TPM.csv',
index_col=0, header=0)  # can also use larger dataset with more genes
metadata_df = pd.read_csv(
    '/Users/kaylac/Downloads/GSE62944_metadata.csv', index_col=0,
header=0)
#print(data.head())

'''
# %% Explore the data
#####################################################
print(data.shape)
print(data.info())
print(data.describe())

# %% Explore the metadata
#####################################################
print(metadata_df.info())
print(metadata_df.describe())
'''
# %% Subset the data for a specific cancer type
#####################################################
cancer_type = 'UCS'  # Uterine Carcinosarcoma

# From metadata, get the rows where "cancer_type" is equal to the
specified cancer type
# Then grab the index of this subset (these are the sample IDs)
cancer_samples = metadata_df[metadata_df['cancer_type'] ==
cancer_type].index
print(cancer_samples)
# Subset the main data to include only these samples
# When you want a subset of columns, you can pass a list of column
names to the data frame in []
UCS_data = data[cancer_samples]



# %% Subset by index (genes)
#####################################################
desired_gene_list = ['ID1', 'ANGPTL4']
gene_list = [gene for gene in desired_gene_list if gene in
UCS_data.index]
for gene in desired_gene_list:
    if gene not in gene_list:
        print(f"Warning: {gene} not found in the dataset.")

# .loc[] is the method to subset by index labels
# .iloc[] will subset by index position (integer location) instead
```

```
UCS_gene_data = UCS_data.loc[gene_list]
print(UCS_gene_data.head())
```

```
Index(['TCGA-N5-A4R8-01A-11R-A28V-07', 'TCGA-N9-A4PZ-01A-22R-A28V-07',
       'TCGA-N6-A4VC-01A-11R-A28V-07', 'TCGA-N8-A4PP-01A-11R-A28V-07',
       'TCGA-N7-A59B-01A-11R-A28V-07', 'TCGA-N9-A4Q4-01A-11R-A28V-07',
       'TCGA-NF-A4X2-01A-11R-A28V-07', 'TCGA-NF-A4WU-01A-11R-A28V-07',
       'TCGA-N9-A4Q8-01A-31R-A28V-07', 'TCGA-ND-A4WA-01A-12R-A28V-07',
       'TCGA-N5-A4RS-01A-11R-A28V-07', 'TCGA-N8-A4PM-01A-11R-A28V-07',
       'TCGA-NA-A4QY-01A-11R-A28V-07', 'TCGA-N6-A4VE-01A-11R-A28V-07',
       'TCGA-NG-A4VU-01A-11R-A28V-07', 'TCGA-N8-A56S-01A-11R-A28V-07',
       'TCGA-NA-A4QW-01A-11R-A28V-07', 'TCGA-N7-A4Y0-01A-12R-A28V-07',
       'TCGA-N9-A4Q3-01A-11R-A28V-07', 'TCGA-N8-A4PQ-01A-11R-A28V-07',
       'TCGA-N9-A4Q7-01A-11R-A28V-07', 'TCGA-NF-A5CP-01A-12R-A28V-07',
       'TCGA-NG-A4VW-01A-11R-A28V-07', 'TCGA-N8-A4PI-01A-21R-A28V-07',
       'TCGA-N5-A4RU-01A-31R-A28V-07', 'TCGA-ND-A4WF-01A-11R-A28V-07',
       'TCGA-N8-A4PL-01A-11R-A28V-07', 'TCGA-N6-A4V9-01A-11R-A28V-07',
       'TCGA-N5-A59E-01A-11R-A28V-07', 'TCGA-N6-A4VG-01A-31R-A28V-07',
       'TCGA-NF-A4WX-01A-11R-A28V-07', 'TCGA-NA-A4QV-01A-11R-A28V-07',
       'TCGA-N5-A59F-01A-11R-A28V-07', 'TCGA-NA-A4QX-01A-11R-A28V-07',
       'TCGA-NA-A4R0-01A-11R-A28V-07', 'TCGA-N8-A4PN-01A-11R-A28V-07',
       'TCGA-NA-A4R1-01A-11R-A28V-07', 'TCGA-N7-A4Y8-01A-11R-A28V-07',
       'TCGA-N8-A4PO-01A-11R-A28V-07', 'TCGA-N9-A4Q1-01A-11R-A28V-07',
       'TCGA-N7-A4Y5-01A-12R-A28V-07', 'TCGA-N5-A4RM-01A-11R-A28V-07',
       'TCGA-N5-A4RA-01A-11R-A28V-07', 'TCGA-NA-A5I1-01A-21R-A28V-07',
       'TCGA-QN-A5NN-01A-11R-A28V-07', 'TCGA-ND-A4W6-01A-11R-A28V-07',
       'TCGA-N5-A4RD-01A-11R-A28V-07', 'TCGA-N5-A4RF-01A-11R-A28V-07',
       'TCGA-N6-A4VF-01A-31R-A28V-07', 'TCGA-N5-A4RN-01A-12R-A28V-07',
       'TCGA-QM-A5NM-01A-11R-A28V-07', 'TCGA-N5-A4RJ-01A-11R-A28V-07',
       'TCGA-N5-A4RO-01A-11R-A28V-07', 'TCGA-N5-A4RV-01A-21R-A28V-07',
       'TCGA-N6-A4VD-01A-11R-A28V-07', 'TCGA-N5-A4RT-01A-11R-A28V-07',
       'TCGA-ND-A4WC-01A-21R-A28V-07'],
      dtype='object', name='sample')
Warning: SP1 not found in the dataset.
         TCGA-N5-A4R8-01A-11R-A28V-07  TCGA-N9-A4PZ-01A-22R-A28V-07  \
ID1                          6.087906                      8.722320
ANGPTL4                      4.924785                      4.640932

         TCGA-N6-A4VC-01A-11R-A28V-07  TCGA-N8-A4PP-01A-11R-A28V-07  \
ID1                          7.312388                      5.937803
ANGPTL4                      5.936389                      6.917569

         TCGA-N7-A59B-01A-11R-A28V-07  TCGA-N9-A4Q4-01A-11R-A28V-07  \
ID1                          9.094339                      5.842666
ANGPTL4                      5.694396                      3.694251

         TCGA-NF-A4X2-01A-11R-A28V-07  TCGA-NF-A4WU-01A-11R-A28V-07  \
ID1                          7.589301                      4.789746
ANGPTL4                      3.608729                      3.094208
```

```
         TCGA-N9-A4Q8-01A-31R-A28V-07  TCGA-ND-A4WA-01A-12R-A28V-
07  ...  \
ID1                          6.159638
10.137546  ...
ANGPTL4                      6.229458
2.883275  ...

         TCGA-N5-A4RF-01A-11R-A28V-07  TCGA-N6-A4VF-01A-31R-A28V-07  \
ID1                          6.212293                      7.338210
ANGPTL4                      7.444103                      3.391924

         TCGA-N5-A4RN-01A-12R-A28V-07  TCGA-QM-A5NM-01A-11R-A28V-07  \
ID1                          5.677914                      9.509561
ANGPTL4                      5.092576                      5.698711

         TCGA-N5-A4RJ-01A-11R-A28V-07  TCGA-N5-A4RO-01A-11R-A28V-07  \
ID1                          8.429906                      8.904862
ANGPTL4                      4.143875                      4.769700

         TCGA-N5-A4RV-01A-21R-A28V-07  TCGA-N6-A4VD-01A-11R-A28V-07  \
ID1                          7.763700                      8.266253
ANGPTL4                      6.048413                      5.395814

         TCGA-N5-A4RT-01A-11R-A28V-07  TCGA-ND-A4WC-01A-21R-A28V-07
ID1                          7.380625                      8.906793
ANGPTL4                      3.616816                      7.179886

[2 rows x 57 columns]
```

```
#This is code in order to run PCA Analysis for UCS (Uterine
Carcinosarcoma) to help find key gene expression patterns and pathways
that may drive UCS development/metastasis

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

#Load the data
data = pd.read_csv(
    '/Users/mariellemiranda/Downloads/Module
3/GSE62944_subsample_topVar_log2TPM.csv',
    index_col=0, header=0
)
metadata_df = pd.read_csv(
    '/Users/mariellemiranda/Downloads/Module 3/GSE62944_metadata.csv',
    index_col=0, header=0
)
```

```
#Filter for UCS samples only
cancer_type = 'UCS'  # Uterine Carcinosarcoma
cancer_samples = metadata_df[metadata_df['cancer_type'] ==
cancer_type].index
UCS_data = data[cancer_samples]

print(f"\nLoaded UCS subset: {UCS_data.shape[0]} genes ×
{UCS_data.shape[1]} samples")

#Transpose data: rows = samples, columns = genes
UCS_data_T = UCS_data.transpose()

# Standardize data (mean = 0, variance = 1)
scaler = StandardScaler()
UCS_scaled = scaler.fit_transform(UCS_data_T)

#Perform PCA (2 components)
pca = PCA(n_components=2)
pca_result = pca.fit_transform(UCS_scaled)

#Create PCA result DataFrame
pca_df = pd.DataFrame(
    data=pca_result,
    columns=['PC1', 'PC2'],
    index=UCS_data_T.index
)
pca_df['cancer_type'] = cancer_type

#PCA scatter plot of UCS samples
plt.figure(figsize=(8,6))
sns.scatterplot(
    x='PC1', y='PC2',
    data=pca_df,
    hue='cancer_type',
    palette='Reds',
    s=100, edgecolor='k', alpha=0.8
)
plt.title('PCA of UCS Gene Expression Patterns')
plt.xlabel(f'PC1 ({pca.explained_variance_ratio_[0]*100:.2f}%
variance)')
plt.ylabel(f'PC2 ({pca.explained_variance_ratio_[1]*100:.2f}%
variance)')
plt.grid(True)
plt.tight_layout()
plt.show()

print("\nExplained variance ratio:", pca.explained_variance_ratio_)
print(f"Total variance captured by PC1 + PC2:
{pca.explained_variance_ratio_.sum()*100:.2f}%")
```

```
#Identifying key genes contributing to PC1 and PC2
# Create a loadings DataFrame
loadings = pd.DataFrame(
    pca.components_.T,           # transpose: rows = genes, columns =
PCs
    index=UCS_data_T.columns,    # gene names
    columns=['PC1', 'PC2']
)

#Top 15 genes contributing to each component
#PC1: mostly measured epithelial organization and polarity
#PC2: measured ECM remodeling and Mesenchymal activation
top_PC1_genes =
loadings['PC1'].abs().sort_values(ascending=False).head(15)
top_PC2_genes =
loadings['PC2'].abs().sort_values(ascending=False).head(15)

print("\nTop genes driving PC1 (main UCS expression pattern):")
print(top_PC1_genes)
print("\nTop genes driving PC2 (secondary UCS pattern):")
print(top_PC2_genes)

#Visualize top PC1 and PC2 genes (barplots)
fig, axes = plt.subplots(1, 2, figsize=(14,5))
sns.barplot(x=top_PC1_genes.index, y=top_PC1_genes.values, ax=axes[0],
palette='coolwarm')
axes[0].set_title('Top Genes Driving PC1')
axes[0].set_ylabel('Absolute PCA Loading')
axes[0].tick_params(axis='x', rotation=75)

sns.barplot(x=top_PC2_genes.index, y=top_PC2_genes.values, ax=axes[1],
palette='viridis')
axes[1].set_title('Top Genes Driving PC2')
axes[1].set_ylabel('Absolute PCA Loading')
axes[1].tick_params(axis='x', rotation=75)

plt.tight_layout()
plt.show()

#Loadings scatter plot (each point = gene)
#which genes drive the main behaviors of UCS tumors (PC1 and PC2 and
their genes (in red) and the other possible genes)
plt.figure(figsize=(8,6))
sns.scatterplot(
    x='PC1', y='PC2',
    data=loadings,
    alpha=0.7
)
plt.title("PCA Loadings for UCS Genes (PC1 vs PC2)")
plt.xlabel("PC1 Loading")
```

```
plt.ylabel("PC2 Loading")
plt.grid(True)

#Label top genes for clarity
for gene in top_PC1_genes.index.union(top_PC2_genes.index):
    x = loadings.loc[gene, 'PC1']
    y = loadings.loc[gene, 'PC2']
    plt.text(x, y, gene, fontsize=8, color='red')

plt.tight_layout()
plt.show()



############################################################
#Checking ID1 and ANGPTL4 (they are both not top contributors for PC1
or PC2, so I wanted to see how they affect/correlate UCS)
corr = UCS_data_T[['ID1', 'ANGPTL4']].corrwith(pca_df['PC2'])
print(corr)




#The following is code being used in order to view and visualize PCA
sample scores for PC1 and PC2
#lower the PC1 = more invasive and agressive
#higher the PC2 = more invasive, agressive, and mesenchymal
#pca_df contains columns ['PC1', 'PC2'] and index = UCS sample IDs

#Display first few samples with their PC1 and PC2 scores
print("\nUCS sample PCA scores (first 10):")
print(pca_df.head(10))

#Sort samples by PC1 (epithelial axis)
sorted_PC1 = pca_df.sort_values(by='PC1', ascending=True)
print("\nSamples ranked by PC1 (lowest = less epithelial, more
invasive):")
print(sorted_PC1[['PC1']].head(10))

#Sort samples by PC2 (invasion axis)
sorted_PC2 = pca_df.sort_values(by='PC2', ascending=False)
print("\nSamples ranked by PC2 (highest = more
mesenchymal/invasive):")
print(sorted_PC2[['PC2']].head(10))

#Bar plot of PC1 scores (Epithelial axis)
plt.figure(figsize=(10,5))
sns.barplot(
    x=sorted_PC1.index,
```

```
    y=sorted_PC1['PC1'],
    palette="coolwarm"
)
plt.xticks(rotation=90)
plt.title("UCS Tumor Samples Ranked by PC1 (Epithelial Structure
Axis)")
plt.xlabel("Sample ID")
plt.ylabel("PC1 Score (Epithelial → Less Epithelial)")
plt.tight_layout()
plt.show()

#Bar plot of PC2 scores (Invasion axis)
plt.figure(figsize=(10,5))
sns.barplot(
    x=sorted_PC2.index,
    y=sorted_PC2['PC2'],
    palette="viridis"
)
plt.xticks(rotation=90)
plt.title("UCS Tumor Samples Ranked by PC2 (Mesenchymal/Invasion
Axis)")
plt.xlabel("Sample ID")
plt.ylabel("PC2 Score (Less Invasive → More Invasive)")
plt.tight_layout()
plt.show()

# Use PCA results (PC1 and PC2)
X_pca = pca_df[['PC1', 'PC2']]

silhouette_scores = {}

# Try multiple cluster counts (k = 2 to 8)
for k in range(2, 9):
    kmeans = KMeans(n_clusters=k, random_state=42)
    labels = kmeans.fit_predict(X_pca)
    score = silhouette_score(X_pca, labels)
    silhouette_scores[k] = score
    print(f"k = {k}, silhouette score = {score:.3f}")

# Find the best number of clusters
best_k = max(silhouette_scores, key=silhouette_scores.get)
print(f"\nBest number of clusters: k = {best_k} (Silhouette =
{silhouette_scores[best_k]:.3f})")

# Visualize silhouette scores across k values
plt.figure(figsize=(8,5))
plt.plot(list(silhouette_scores.keys()),
list(silhouette_scores.values()), marker='o', linewidth=2)
plt.title("Silhouette Scores for Different Cluster Numbers (k)")
plt.xlabel("Number of Clusters (k)")
```

```
plt.ylabel("Silhouette Score")
plt.grid(True)
plt.tight_layout()
plt.show()

# Fit the final KMeans model with the best k
final_kmeans = KMeans(n_clusters=best_k, random_state=42)
pca_df['Cluster'] = final_kmeans.fit_predict(X_pca)

# Visualize clusters in PCA space
plt.figure(figsize=(8,6))
sns.scatterplot(
    x='PC1', y='PC2',
    data=pca_df,
    hue='Cluster',
    palette='Set2',
    s=100, edgecolor='k', alpha=0.8
)
plt.title(f'K-Means Clustering on UCS PCA Data (k = {best_k})')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend(title='Cluster')
plt.grid(True)
plt.tight_layout()
plt.show()



#VALIDATION: APPLYING PCA + K-MEANS TO TEST SET AND COMPUTE SILHOUETTE
SCORE
#Load the test set
test_data = pd.read_csv(
    '/Users/mariellemiranda/Downloads/Module
3/GSE62944_subsample_topVar_log2TPM.csv',
    index_col=0, header=0
)

#Match genes between training and test sets
common_genes = UCS_data.index.intersection(test_data.index)
test_data = test_data.loc[common_genes]

#Transpose to samples × genes (like training data)
test_data_T = test_data.transpose()

#Apply the same standardization and PCA transform as the training set
test_scaled = scaler.transform(test_data_T)
test_pca_result = pca.transform(test_scaled)

#Create DataFrame of test PCA results
test_pca_df = pd.DataFrame(
```

```
    data=test_pca_result,
    columns=['PC1', 'PC2'],
    index=test_data_T.index
)

#Apply trained KMeans (from training PCA)
test_labels = final_kmeans.predict(test_pca_df[['PC1', 'PC2']])
test_pca_df['Cluster'] = test_labels

#Compute Silhouette Score for test set
test_silhouette = silhouette_score(test_pca_df[['PC1', 'PC2']],
test_labels)
print(f"Test Set Silhouette Score: {test_silhouette:.3f}")

#Visualize test set clusters in PCA space
plt.figure(figsize=(8,6))
sns.scatterplot(
    x='PC1', y='PC2',
    data=test_pca_df,
    hue='Cluster',
    palette='Set2',
    s=100, edgecolor='k', alpha=0.8
)
plt.title(f'K-Means Clustering on UCS Test Set (k =
{final_kmeans.n_clusters})')
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend(title='Cluster')
plt.grid(True)
plt.tight_layout()
plt.show()
```

Verify and validate your analysis:

Pick a SPECIFIC method to determine how well your model is performing and describe how it works here.

(Describe how you checked to see that your analysis gave you an answer that you believe (verify). Describe how your determined if your analysis gave you an answer that is supported by other evidence (e.g., a published paper).

To verify and validate the PCA and clustering model, the silhouette score was used to measure how well the samples were grouped into distinct clusters after dimensionality reduction. The Silhouette Score ranges from -1 to +1, where higher values indicate better-defined and more distinct clusters. After testing multiple cluster counts (k = 2–8), the model achieved its highest score at k = 2, with a Silhouette Score of 0.441 on the training dataset. This suggests that the UCS tumor samples can be moderately separated into two biologically meaningful subgroups based on their gene expression patterns.

To assess the model's generalizability, the same PCA transformation and clustering pipeline were applied to an independent test set, which produced a Silhouette Score of 0.398. The close agreement between the training and test scores demonstrates that the clustering structure is consistent and not overfitted to the original dataset. This validates that the PCA (while there is no correlation between PC1 and PC2) captured robust biological signals underlying UCS variability.

Therefore, with the use of the k-cluster to run a silhouette score that gave a score of 0.398 this showed that the PCA separated the UCS tumor samples into biologically distinct subgroups instead of random scatter. With this in mind, then the PCA showed the following for PC1 and PC2. For PC1, epithelial organization was the main grouping and suggested that a lower PC1 score reflected a loss of epithelial traits associated with early tumor invasion. The four genes that were the lowest were: GRHL2, KRT8, CX3CL1, and CRB3. Whereas, PC2 grouped mesenchymal and angiogenesis-related genes and suggested that a higher PC2 score corresponded to an increased vascular remolding tissue invasion, and metastasis potential. The four genes that were the highest were: FBN1, ZEB1, ZCCHC24, and LIFR. Overall, this demonstrates that these genes and pathways are the main driving forces behind tissue invasion and metastasis in UCS.

This conclusion is supported by a variety of papers. FBN1, for example, is highlighted to lead to poor survival rates and high metastasis when there are high levels of expression in cases of ovarian cancer. Our conclusion extrapolates this paper to cases of UCS because even though research is limited, the patterns between more studied cancers and UCS should not be ignored. ZEB1 promotes metastasis because it is a transcription factor that leads to the promotion of EMT, which helps uterine cancer cells migrate to other parts of the body. ZEB1 can also contribute to the positive feedback loop that leads to more EMT production and thus more metastasis. There is little to no data that directly relates ZCCHC24 to UCS; however, it is related to metastasis in many other types of cancers due to its high expression in mesenchymal-like cancer stem cells. Similarly, LIFR is currently understudied with its relationship to UCS but is frequently present in aggressive types of endometrial cancers and is shown to help cancer cells survive and travel within the body/ immune system.

Work cited:

- NIH - The extracellular matrix glycoprotein fibrillin-1 in health and disease
- NIH - ZEB1 Expression in Endometrial Biopsy Predicts Lymph Node Metastases in Patient with Endometrial Cancer
- NIH - LIF/LIFR oncogenic signaling is a novel therapeutic target in endometrial cancer

Conclusions and Ethical Implications:

(Think about the answer your analysis generated, draw conclusions related to your overarching question, and disc uss the ethical implications of your conclusions.

Conclusion:

- The results of the analysis support the overarching goal of identifying the key genetic pathways involved in UCS development and metastasis. From the PCA, it revealed two major biological axes underlying tumor variability: an epithelial-loss axis (PC1) and a mesenchymal/invasion axis (PC2). The genres most strongly driving these axes are

GRHL2, KRT8, CX3CL1, and CRB3 for epithelial structure, and FBN1, ZEB1, ZCCHC24, and LIFR for invasion and vascular remodeling. These highlight the transition from stable epithelial states to aggressive mesenchymal phenotypes that facilitate metastasis. These findings align with published evidence showing that genes like ZEB1 and FBN1 actively promote epithelial-mesenchymal transition (EMT), extracellular matrix remodeling, and dissemination in gynecologic cancers. Together this suggests that UCS tumors are not just becoming more invasive (through EMT pathways) but are also increasing their ability to grow and spread by forming new blood vessels (through angiogenesis pathways). However, because PCA is not a mechanistic test and the dataset is limited, these results cannot establish causation but instead point to possible pathways and gene targets that future research could explore for therapeutic development.

Ethical Implications:

- For ethical limitation, two factors must be considered. First, PCA and clustering only show correlation, but cannot prove causation. Therefore, clinical recommendations cannot be made without further biological validation. Furthermore, since UCS is rare and understudied, applying insights from other cancers must be done cautiously in order to avoid overstating their relevance. Overall, the research has to be transparent about uncertainty, not overstating what the results can prove, and making sure that any potential drug targets are thoroughly tested in the lab and in clinical studies before they are considered for use in patients.

Limitations and Future Work:

(Think about the answer your analysis generated, draw conclusions related to your overarching question, and discuss the ethical implications of your conclusions.

The limitation of this project is the small data set and the overall lack of research on UCS. We also don't have the background on how this data was collected, so the conclusions we can draw are very limited. Additionally, PCA is not a mechanistic test and therefore cannot make conclusions on HOW a gene influences something. With this being said, we cannot say that we know what causes metastasis, but there is a relationship between the expression of these genes and metastasis.

Future work can provide a more explicit derivation of samples in order to better understand if these patterns of gene expression are in every tissue or are locally regulated. We can also look at genes outside of this gene set and across other types of cancer that have known metastasis-associated genes. This could include endometrial and uterine cancers. We could additionally look into unsupervised learning to see if there are ANY up-/down regulated genes in tumors of this kind, and then further look into what those genes are known for.

NOTES FROM YOUR TEAM:

- the research on this cancer is limited, so some inference are being made fro gernal uterine cancer studies

QUESTIONS FOR YOUR TA:

- what should we do if other genes that aren't in dataset have large effect on this cancer?