Module_3:

Team Members:

Neil Shroff and Senada Mujanovic

Project Title:

Investigating the occurence of angiogenic markers in patients with breast cancer via VEGF and FGF concentrations

Project Goal:

This project seeks to analyze the impact that angiogenesis has on breast cancer pathogenesis via measuring the expression of genes that code for VEGF and FGF.

Disease Background:

- Cancer hallmark focus:
    - Angiogenesis
- Overview of hallmark:
    - When new tissues are developing in the body, they heavily rely on nearby capillaries to receive nutrients and expel waste. Thus, new capillaries can be stimulated to grow via angiogenic initiating signals such as vascular endothelial growth factor (VEGF) and fibroblast growth factor (FGF1/2) to ensure that vasculature does not continually grow, angiogenesis inhibitors such as thrombospondin-1 will bind to respective receptors to prevent uncontrolled growth. During cancer pathogenesis there is an alteration in transcription leading to an upregulation of VEGF and FGF and a downregulation of thrombospondin-1 and other regulator proteins such as p53. Therefore, this leads to an uncontrolled growth of blood vessels near the tumor, allowing for increased nutrient uptake to supplement the uncontrolled growth of the tissue. For future developments, questions remain as to how these promoters and inhibitors interact with one another and how to influence these

interactions. Moreover, questions about whether antiangiogenic therapeutics could be translated to every cancer type still remain.

- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate):
    - VEGFA and FGF-1 will be studied in this analysis of breast cancer. These genes encode for VEGF and FGF which are key growth factors for angiogenesis. In the pathogenesis of cancer, these genes are mutated and thus the VEGF and FGF are upregulated, in turn leading to an uncontrolled growth of blood vessels.
- Prevalence & incidence
    - Breast cancer is the most common cancer type for women in the United States, with about 4 million women living with female breast cancer in the United States in 2022.
        - https://seer.cancer.gov/statfacts/html/breast.html
    - It is projected that ~316,950 new cases of invasive breast cancer will be diagnosed in women in 2025.
        - https://www.cancer.org/cancer/types/breast-cancer/about/how-common-is-breast-cancer.html
- Risk factors (genetic, lifestyle) & Societal determinants
    - Genetic factors: Mutations in BRCA1 and BRCA2 are the most well-known genetic contributors, significantly raising the lifetime risk. Other genes like TP53, PTEN, CHEK2, and HER2 amplifications also influence tumor development and aggressiveness.
    - Lifestyle factors: A high-fat diet, alcohol intake, obesity, prolonged hormone replacement therapy, and lack of physical activity increase risk. Early menstruation, late menopause, and nulliparity also contribute due to prolonged estrogen exposure.
    - Societal determinants: Socioeconomic barriers, healthcare access, and racial disparities play a major role in outcomes. Black women, for instance, are more likely to develop triple-negative breast cancer, an aggressive subtype, and have higher mortality rates.
    - Environmental exposures—such as endocrine-disrupting

chemicals—are emerging as possible contributors to breast cancer risk.

    - https://us.kisqali.com/metastatic-breast-cancer/proven-results/results-with-
      kisqali?site=BST-1238433GK100421&utm_source=g
      dtc-unbranded-overall-survival-treatment-
      exact%3Bs%3Bph%3Bbr%3Bonc%3Bdtc%3Btre_sep
      survival&utm_term=metastatic%20breast%20cance
      _bvtC1QzDHF5XObsrnryES3oFsBoCRVMQAvD_BwE

- Standard of care treatments (& reimbursement)

    - Localized and early-stage disease: Surgery (lumpectomy or mastectomy) followed by radiation therapy remains the standard approach to remove and control localized tumors.
    - Systemic therapies: Include chemotherapy, endocrine therapy for hormone receptor-positive cancers (e.g., tamoxifen, aromatase inhibitors), and targeted therapies for HER2-positive subtypes (e.g., trastuzumab, pertuzumab).
    - Anti-angiogenic therapy: Drugs like Bevacizumab (Avastin) target VEGF signaling to inhibit new blood vessel growth. However, their effectiveness in breast cancer has been debated due to limited improvement in overall survival and significant side effects such as hypertension and clotting risks.
    - Reimbursement: Most standard treatments are covered by private and public insurance in the U.S., but access can vary globally. Cost and insurance disparities can limit the availability of advanced targeted therapies and clinical trial participation.
        - https://www.mayoclinic.org/diseases-conditions/breast-cancer/diagnosis-treatment/drc-20352475

- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)

    - Anatomy & organ physiology: The breast is composed of lobules (milk-producing glands) and ducts (milk-carrying channels), surrounded by fatty and connective tissue. Cancer often originates in ductal or lobular epithelial cells, where mutations drive uncontrolled proliferation.
    - Cell & molecular physiology: In hypoxic (low-oxygen) tumor

regions, cells stabilize HIF-1α, which upregulates VEGF and FGF, stimulating endothelial cells to form new blood vessels. These new tumor vessels are often disorganized, leaky, and inefficient, leading to uneven oxygen distribution and facilitating metastasis by allowing tumor cells to enter circulation.

- FGF and VEGF signaling pathways also interact, creating redundancy in angiogenic signaling, which can make tumors resistant to single-agent anti-angiogenic drugs.
- This aberrant angiogenesis not only supports tumor growth but also contributes to the tumor’s ability to evade immune surveillance and develop drug resistance.
  - https://pmc.ncbi.nlm.nih.gov/articles/PMC8582527/

Data-Set:

- The subset of data that will be analyzed will be the subsample of data that contains the RNA sequence of the VEGF and FGF genes, as this directly correlates with the concentration of VEGF and FGF.

- The expression of these genes will aid in the understanding of how angiogenesis impacts breast cancer pathogenesis

- The data was collected via the RNA gene sequencing of various breast invasive carcinoma tissue samples.

- The data is found on Canvas in Module 3 Data: https://canvas.its.virginia.edu/courses/153653/modules

Data Analyis:

Methods

The machine learning technique I am using is: a K-means clustering to find a correlation between VEGFA and FGF1 expression and breast cancer incidence. This was chosen, as the data does not contain a control group to compare the gene expression in breast cancer to. Thus, by clustering known trends in the data with two features, FGF1 and VEGFA, a K-means clustering allows for the most optimal learning method to observe a correlation in the dataset.

This method attempts to minimize the total distance between each sample and the center of its assigned cluster. This algorithm stops when the centers of the clusters stop moving significantly, thus making the clusters as compact as possible, increasing the accuracy of the model.

Analysis

In [28]:

```
#Import Necessary Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import pairwise_distances
from sklearn import datasets
from sklearn import metrics
from sklearn.metrics import silhouette_score
```

In [2]:

```
#Configure Variables
file_path = "GSE62944_subsample_topVar_log2TPM.csv"
feature_genes = ['VEGFA', 'FGF1']
n_clusters = 3

#Load/Process Data
try:
    df = pd.read_csv(file_path, index_col=0)
    X_raw = df.T
    X = X_raw[feature_genes].copy()
    X.dropna(inplace=True)
    print(f'Total Usable Smaples for Clustering: {len(X)}')
    if len(X) == 0:
        raise ValueError("No usable samples found after dropping missing value
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
except Exception as e:
    print(f"Error during data loading/processing: {e}")
    print("\nCheck raw data snippet:")
    print(df.head())
    exit()
```

```
Total Usable Smaples for Clustering: 1802
```

In [3]:

```
#Perform Clustering
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)
X['Cluster'] = kmeans.labels_
print('\n --- Clustering Results --- \n')
print(X['Cluster'].value_counts())
```

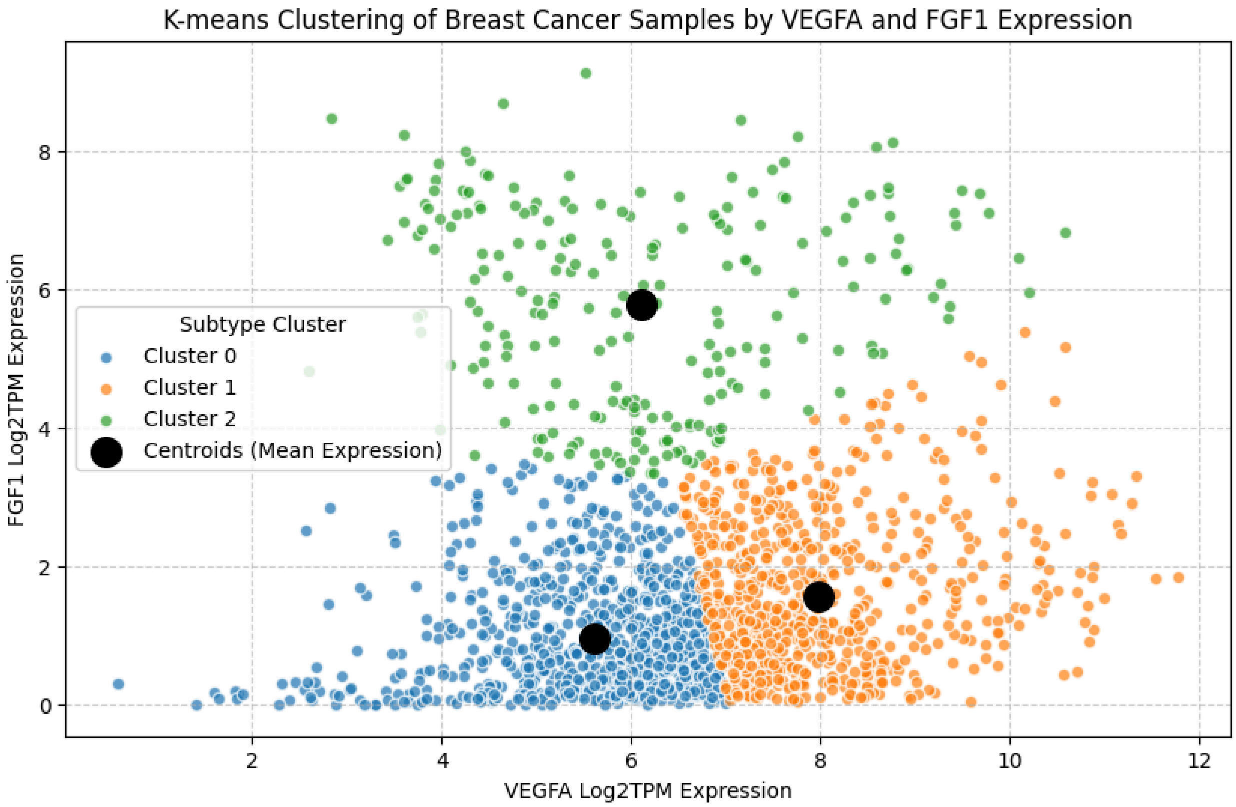
```
 --- Clustering Results ---

Cluster
0    855
1    718
2    229
Name: count, dtype: int64
```

In [4]:
```
#Analyze Cluster Centers
cluster_means = X.groupby('Cluster')[feature_genes].mean()
print('\nAverage Log2TPM Expression by Cluster (Cluster Center):')
print(cluster_means.round(3))
```

```
Average Log2TPM Expression by Cluster (Cluster Center):
         VEGFA   FGF1
Cluster
0        5.610  0.944
1        7.970  1.566
2        6.113  5.781
```

In [5]:
```
#Visualize Clusters
print('\n --- Data Visualization ---')
plt.figure(figsize = (10,6))
for cluster_label in range(n_clusters):
    cluster_data = X[X['Cluster'] == cluster_label]
    plt.scatter(cluster_data['VEGFA'], cluster_data['FGF1'], label=f'Cluster {
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='black', label='Centroi
plt.title('K-means Clustering of Breast Cancer Samples by VEGFA and FGF1 Expre
plt.xlabel('VEGFA Log2TPM Expression')
plt.ylabel('FGF1 Log2TPM Expression')
plt.legend(title = 'Subtype Cluster')
plt.grid(True, linestyle = '--', alpha = 0.6)
plt.savefig('VEGFA_FGF!_Clustering_Scatter .png')
print("Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png")

#Output Classification Results
X.to_csv('tumor_samples_clustered_subtypes.csv')
print("Saved cluster assignments for each sample to 'tumor_samples_clustered_s
```

```
 --- Data Visualization ---
Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png
Saved cluster assignments for each sample to 'tumor_samples_clustered_subtype
s.csv'
```



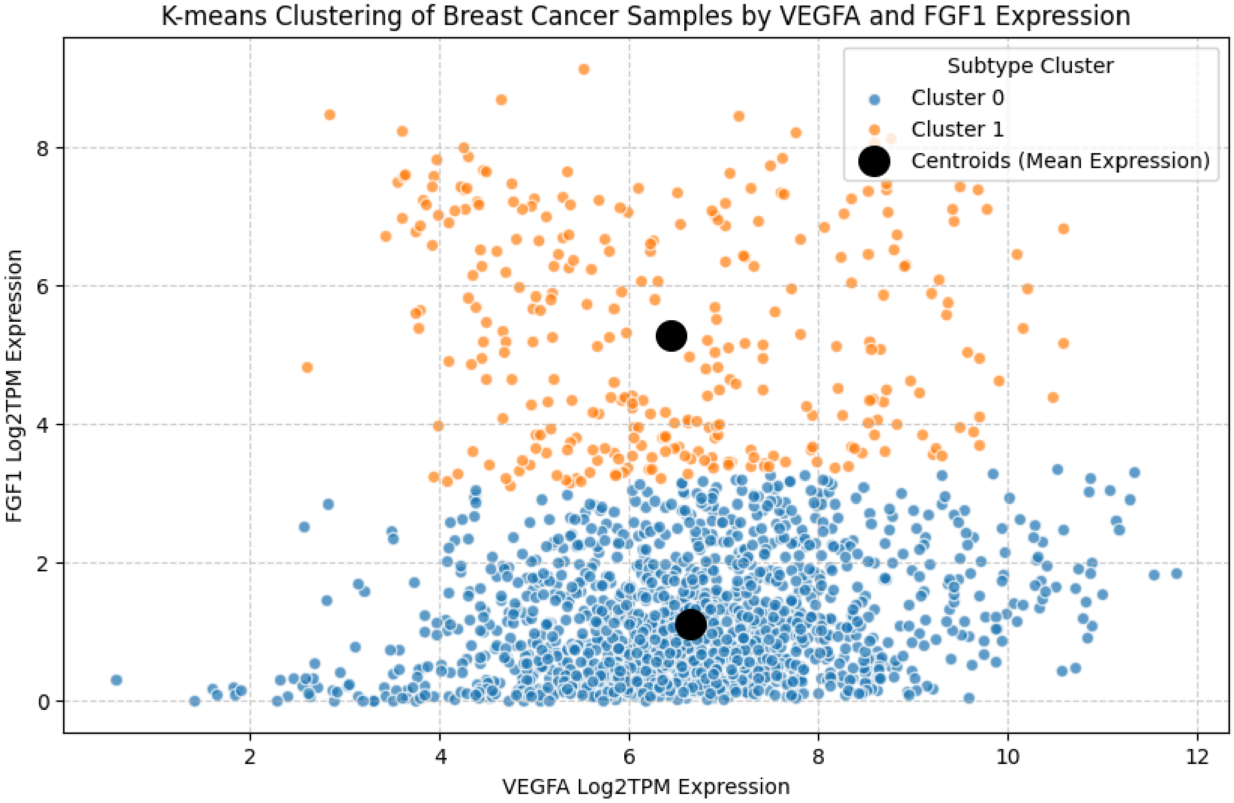
In [21]:

```
#Verify Clustering using Silhouette Score
X, y = datasets.load_iris(return_X_y=True)
kmeans_model = KMeans(n_clusters=3, random_state=1).fit(X)
labels = kmeans_model.labels_
metrics.silhouette_score(X, labels, metric='euclidean')
print ('Silhouette Score: ', metrics.silhouette_score(X, labels, metric='eucli
```

```
Silhouette Score:  0.551191604619592
```

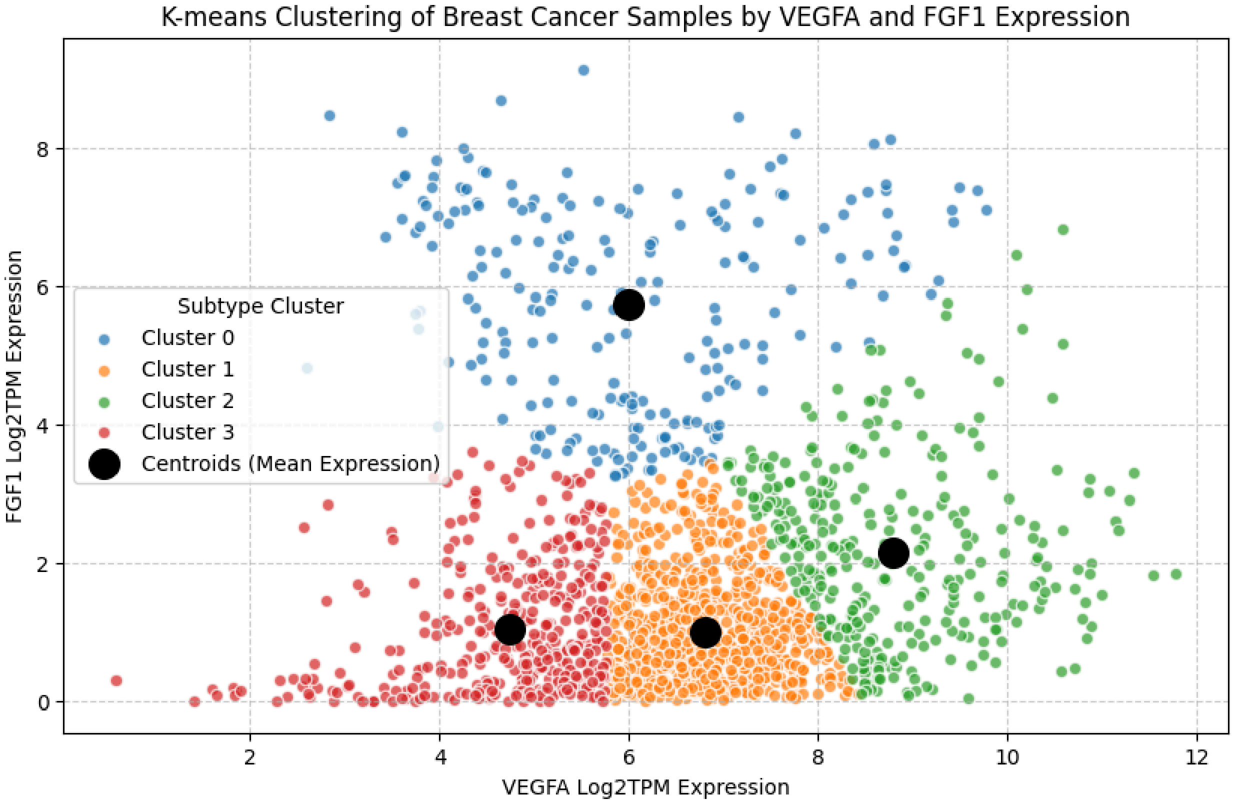
In [31]:

```
# Repeat code with 2 clusters
#Configure Variables
file_path = "GSE62944_subsample_topVar_log2TPM.csv"
feature_genes = ['VEGFA', 'FGF1']
n_clusters = 2

#Load/Process Data
try:
    df = pd.read_csv(file_path, index_col=0)
    X_raw = df.T
    X = X_raw[feature_genes].copy()
    X.dropna(inplace=True)
    print(f'Total Usable Smaples for Clustering: {len(X)}')
    if len(X) == 0:
        raise ValueError("No usable samples found after dropping missing value
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
except Exception as e:
    print(f"Error during data loading/processing: {e}")
    print("\nCheck raw data snippet:")
    print(df.head())
    exit()

#Perform Clustering
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)
X['Cluster'] = kmeans.labels_
print('\n --- Clustering Results --- \n')
print(X['Cluster'].value_counts())

#Analyze Cluster Centers
cluster_means = X.groupby('Cluster')[feature_genes].mean()
print('\nAverage Log2TPM Expression by Cluster (Cluster Center):')
print(cluster_means.round(3))

#Visualize Clusters
print('\n --- Data Visualization ---')
plt.figure(figsize = (10,6))
for cluster_label in range(n_clusters):
    cluster_data = X[X['Cluster'] == cluster_label]
    plt.scatter(cluster_data['VEGFA'], cluster_data['FGF1'], label=f'Cluster {
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='black', label='Centroi
plt.title('K-means Clustering of Breast Cancer Samples by VEGFA and FGF1 Expre
plt.xlabel('VEGFA Log2TPM Expression')
plt.ylabel('FGF1 Log2TPM Expression')
plt.legend(title = 'Subtype Cluster')
plt.grid(True, linestyle = '--', alpha = 0.6)
plt.savefig('VEGFA_FGF!_Clustering_Scatter .png')
print("Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png")

#Output Classification Results
X.to_csv('tumor_samples_clustered_subtypes.csv')
print("Saved cluster assignments for each sample to 'tumor_samples_clustered_s

#Verify Clustering using Silhouette Score
X, y = datasets.load_iris(return_X_y=True)
kmeans_model = KMeans(n_clusters=2, random_state=1).fit(X)
labels = kmeans_model.labels_
metrics.silhouette_score(X, labels, metric='euclidean')
print ('Silhouette Score: ', metrics.silhouette_score(X, labels, metric='eucli
Total Usable Smaples for Clustering: 1802

 --- Clustering Results ---

Cluster
0    1503
1     299
Name: count, dtype: int64

Average Log2TPM Expression by Cluster (Cluster Center):
         VEGFA   FGF1
Cluster
0        6.649  1.112
1        6.437  5.297

 --- Data Visualization ---
Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png
Saved cluster assignments for each sample to 'tumor_samples_clustered_subtype
s.csv'
Silhouette Score:  0.6810461692117462
```



In [24]:

```
# Repeat code with 4 clusters
#Configure Variables
file_path = "GSE62944_subsample_topVar_log2TPM.csv"
feature_genes = ['VEGFA', 'FGF1']
n_clusters = 4

#Load/Process Data
try:
    df = pd.read_csv(file_path, index_col=0)
    X_raw = df.T
    X = X_raw[feature_genes].copy()
    X.dropna(inplace=True)
    print(f'Total Usable Smaples for Clustering: {len(X)}')
    if len(X) == 0:
        raise ValueError("No usable samples found after dropping missing value
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
except Exception as e:
    print(f"Error during data loading/processing: {e}")
    print("\nCheck raw data snippet:")
    print(df.head())
    exit()

#Perform Clustering
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)
X['Cluster'] = kmeans.labels_
print('\n --- Clustering Results --- \n')
print(X['Cluster'].value_counts())

#Analyze Cluster Centers
cluster_means = X.groupby('Cluster')[feature_genes].mean()
print('\nAverage Log2TPM Expression by Cluster (Cluster Center):')
print(cluster_means.round(3))

#Visualize Clusters
print('\n --- Data Visualization ---')
plt.figure(figsize = (10,6))
for cluster_label in range(n_clusters):
    cluster_data = X[X['Cluster'] == cluster_label]
    plt.scatter(cluster_data['VEGFA'], cluster_data['FGF1'], label=f'Cluster {
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='black', label='Centroi
plt.title('K-means Clustering of Breast Cancer Samples by VEGFA and FGF1 Expre
plt.xlabel('VEGFA Log2TPM Expression')
plt.ylabel('FGF1 Log2TPM Expression')
plt.legend(title = 'Subtype Cluster')
plt.grid(True, linestyle = '--', alpha = 0.6)
plt.savefig('VEGFA_FGF!_Clustering_Scatter .png')
print("Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png")

#Output Classification Results
X.to_csv('tumor_samples_clustered_subtypes.csv')
print("Saved cluster assignments for each sample to 'tumor_samples_clustered_s

#Verify Clustering using Silhouette Score
X, y = datasets.load_iris(return_X_y=True)
kmeans_model = KMeans(n_clusters=4, random_state=1).fit(X)
labels = kmeans_model.labels_
metrics.silhouette_score(X, labels, metric='euclidean')
print ('Silhouette Score: ', metrics.silhouette_score(X, labels, metric='eucli
Total Usable Smaples for Clustering: 1802

 --- Clustering Results ---

Cluster
1    847
3    399
2    332
0    224
Name: count, dtype: int64

Average Log2TPM Expression by Cluster (Cluster Center):
         VEGFA   FGF1
Cluster
0        6.000  5.748
1        6.811  0.988
2        8.785  2.155
3        4.736  1.042

 --- Data Visualization ---
Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png
Saved cluster assignments for each sample to 'tumor_samples_clustered_subtype
s.csv'
Silhouette Score:  0.49535632852884987
```



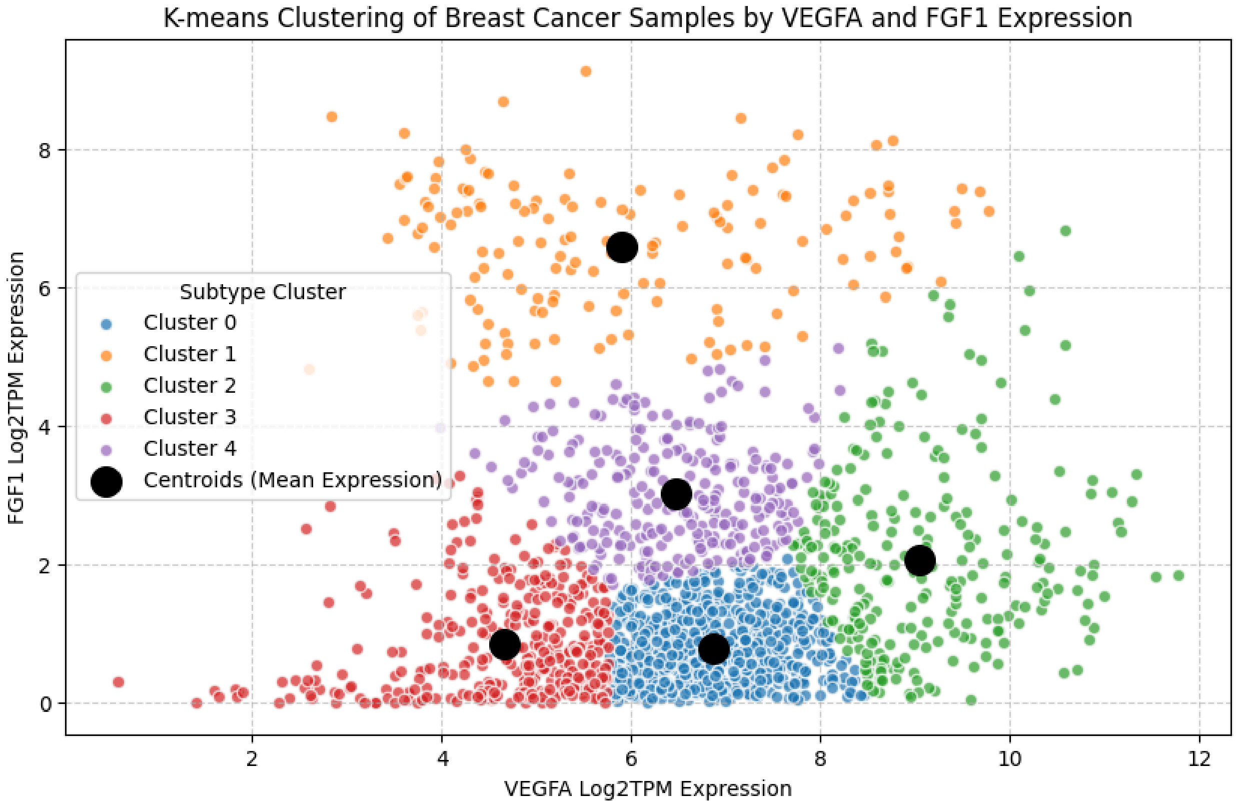
In [25]:

```
# Repeat code with 5 clusters
#Configure Variables
file_path = "GSE62944_subsample_topVar_log2TPM.csv"
feature_genes = ['VEGFA', 'FGF1']
```

```
n_clusters = 5

#Load/Process Data
try:
    df = pd.read_csv(file_path, index_col=0)
    X_raw = df.T
    X = X_raw[feature_genes].copy()
    X.dropna(inplace=True)
    print(f'Total Usable Smaples for Clustering: {len(X)}')
    if len(X) == 0:
        raise ValueError("No usable samples found after dropping missing value
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
except Exception as e:
    print(f"Error during data loading/processing: {e}")
    print("\nCheck raw data snippet:")
    print(df.head())
    exit()

#Perform Clustering
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)
X['Cluster'] = kmeans.labels_
print('\n --- Clustering Results --- \n')
print(X['Cluster'].value_counts())

#Analyze Cluster Centers
cluster_means = X.groupby('Cluster')[feature_genes].mean()
print('\nAverage Log2TPM Expression by Cluster (Cluster Center):')
print(cluster_means.round(3))

#Visualize Clusters
print('\n --- Data Visualization ---')
plt.figure(figsize = (10,6))
for cluster_label in range(n_clusters):
    cluster_data = X[X['Cluster'] == cluster_label]
    plt.scatter(cluster_data['VEGFA'], cluster_data['FGF1'], label=f'Cluster {
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='black', label='Centroi
plt.title('K-means Clustering of Breast Cancer Samples by VEGFA and FGF1 Expre
plt.xlabel('VEGFA Log2TPM Expression')
plt.ylabel('FGF1 Log2TPM Expression')
plt.legend(title = 'Subtype Cluster')
plt.grid(True, linestyle = '--', alpha = 0.6)
plt.savefig('VEGFA_FGF!_Clustering_Scatter .png')
print("Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png")

#Output Classification Results
X.to_csv('tumor_samples_clustered_subtypes.csv')
print("Saved cluster assignments for each sample to 'tumor_samples_clustered_s

#Verify Clustering using Silhouette Score
X, y = datasets.load_iris(return_X_y=True)
```

```python
kmeans_model = KMeans(n_clusters=5, random_state=1).fit(X)
labels = kmeans_model.labels_
metrics.silhouette_score(X, labels, metric='euclidean')
print ('Silhouette Score: ', metrics.silhouette_score(X, labels, metric='eucli
```

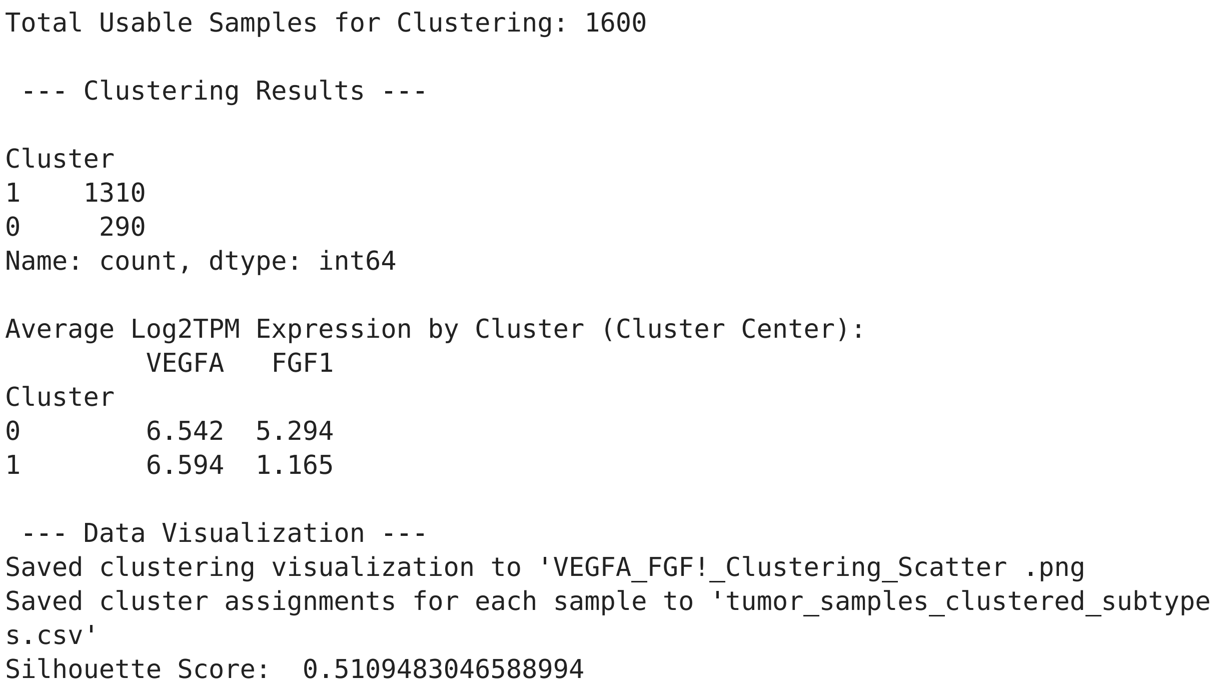
```
Total Usable Smaples for Clustering: 1802

 --- Clustering Results ---

Cluster
0    752
3    359
2    271
4    269
1    151
Name: count, dtype: int64

Average Log2TPM Expression by Cluster (Cluster Center):
         VEGFA   FGF1
Cluster
0        6.865  0.774
1        5.893  6.591
2        9.047  2.071
3        4.666  0.853
4        6.465  3.012

 --- Data Visualization ---
Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png
Saved cluster assignments for each sample to 'tumor_samples_clustered_subtype
s.csv'
Silhouette Score:  0.44207674329916946
```



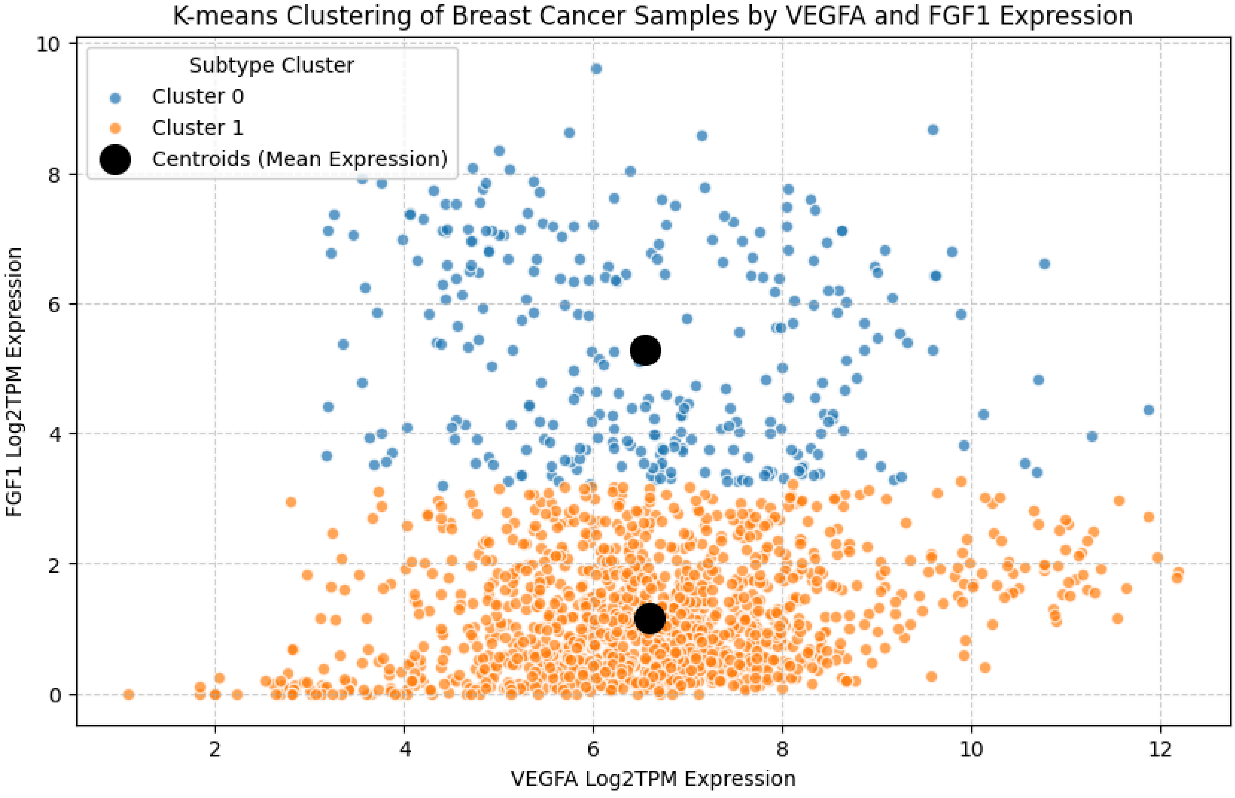
Observing the silhouette scores of the model with varying clusters, the two cluster model has the highest score, and thus will be used to further validate the model with test data.

```
In [33]: #Configure Variables
file_path = "TEST_SET_GSE62944_subsample_log2TPM.csv"
feature_genes = ['VEGFA', 'FGF1']
n_clusters = 2

#Load/Process Data
try:
    df = pd.read_csv(file_path, index_col=0)
    X_raw = df.T
    X = X_raw[feature_genes].copy()
    X.dropna(inplace=True)
    print(f'Total Usable Samples for Clustering: {len(X)}')
    if len(X) == 0:
        raise ValueError("No usable samples found after dropping missing value
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
except Exception as e:
    print(f"Error during data loading/processing: {e}")
    print("\nCheck raw data snippet:")
    print(df.head())
    exit()

#Perform Clustering
```

```
kmeans = KMeans(n_clusters=n_clusters, random_state=42, n_init=10)
kmeans.fit(X_scaled)
X['Cluster'] = kmeans.labels_
print('\n --- Clustering Results --- \n')
print(X['Cluster'].value_counts())

#Analyze Cluster Centers
cluster_means = X.groupby('Cluster')[feature_genes].mean()
print('\nAverage Log2TPM Expression by Cluster (Cluster Center):')
print(cluster_means.round(3))

#Visualize Clusters
print('\n --- Data Visualization ---')
plt.figure(figsize = (10,6))
for cluster_label in range(n_clusters):
    cluster_data = X[X['Cluster'] == cluster_label]
    plt.scatter(cluster_data['VEGFA'], cluster_data['FGF1'], label=f'Cluster {
centroids = scaler.inverse_transform(kmeans.cluster_centers_)
plt.scatter(centroids[:, 0], centroids[:, 1], s=200, c='black', label='Centroi
plt.title('K-means Clustering of Breast Cancer Samples by VEGFA and FGF1 Expre
plt.xlabel('VEGFA Log2TPM Expression')
plt.ylabel('FGF1 Log2TPM Expression')
plt.legend(title = 'Subtype Cluster')
plt.grid(True, linestyle = '--', alpha = 0.6)
plt.savefig('VEGFA_FGF!_Clustering_Scatter .png')
print("Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png")

#Output Classification Results
X.to_csv('tumor_samples_clustered_subtypes.csv')
print("Saved cluster assignments for each sample to 'tumor_samples_clustered_s

#Verify Clustering using Silhouette Score
kmeans_model = KMeans(n_clusters=2, random_state=1).fit(X)
labels = kmeans_model.labels_
metrics.silhouette_score(X, labels, metric='euclidean')
print ('Silhouette Score: ', metrics.silhouette_score(X, labels, metric='eucli
```



```
Total Usable Samples for Clustering: 1600

 --- Clustering Results ---

Cluster
1    1310
0     290
Name: count, dtype: int64

Average Log2TPM Expression by Cluster (Cluster Center):
         VEGFA   FGF1
Cluster
0        6.542  5.294
1        6.594  1.165

 --- Data Visualization ---
Saved clustering visualization to 'VEGFA_FGF!_Clustering_Scatter .png
Saved cluster assignments for each sample to 'tumor_samples_clustered_subtype
s.csv'
Silhouette Score:  0.5109483046588994
```



Verify and validate your analysis:

K-Means clustering was used as a learning method in this project. Thus, to verify that the model works sufficiently to analyze the expression of VEGFA and FGF1 in relation to breast cancer, a silhouette score was employed. The closer that the silhouette score is to one, the better the clustering of the model is. This is because, the silhouette score measures how close one point in a cluster is close to another point in the same cluster and how far apart that point is from a point in another cluster. A score close to 1 shows that the learning model properly grouped the data into appropriate clusters that are different from one another.

The silhouette score is calculated by comparing how similar a data point is to its own cluster versus other clusters using the formula s = (b-a)/max(a,b), where a is the average distance to points in the same cluster, b is the minimum average distance to points in a different cluster.

Initially, to determine which would be the optimal amount of clusters, 2, 3, 4, and 5 clusters were chosen on the original data set and their silhouette scores were calculated. This showed that two clusters yielded the most reliable model, as the silhouette score was the highest for this. Then, this was employed on the test data and the silhouette score for that was calculated as well.

When comparing the expression of VEGFA and FGF1 via clustering, in all models, it is shown that the cluster with the most samples has a much higher VEGFA expression in comparison to FGF1. Thus, one could conclude in patients with breast cancer, it is likely that the VEGFA gene expression is more expressed compared to the FGF1 gene. This implies that while FGF may serve a role in angiogenesis of breast cancer tissue, VEGF is a much better indicator to measure the progression of the tumor(Johnson & Wilgus, 2014). A higher concentration of VEGF indicating a greater rate of angiogenesis, and thus a more advanced/agressive type of breast cancer. Additionally, as VEGFA expression increases, the pathogenesis of breast cancer is impacted, as VEGFA experession promotes vascular development near the site of the tumor, thus increasing blood flow and providing the tumor cells with excess nutrients allowing for rapid proliferation. Additionally, the production of VEGF has immune impacts as it suppresses immune function near the tumor site, thus furthering the development of breast cancer (Sharma et al., 2025).

Johnson, K. E., & Wilgus, T. A. (2014). Vascular Endothelial Growth Factor and Angiogenesis in the Regulation of Cutaneous Wound Repair. Advances in wound care, 3(10), 647–661. https://doi.org/10.1089/wound.2013.0517

Sharma, P., Chida, K., Wu, R., Tung, K., Hakamada, K., Ishikawa, T., & Takabe, K. (2025). VEGFA Gene Expression in Breast Cancer Is Associated With Worse Prognosis, but Better Response to Chemotherapy and Immunotherapy. World journal of oncology, 16(1), 120–130. https://doi.org/10.14740/wjon1993

Conclusions and Ethical Implications:

From the clustering results, it was clear that VEGFA was expressed at higher levels than FGF1 in most breast cancer samples. This suggests that VEGF plays a bigger role in promoting angiogenesis and tumor growth compared to FGF. It also supports what other research has found that VEGF is one of the main signals driving blood vessel formation in cancer and thus contributes to the pathogenesis of breast cancer. These results point toward VEGF as a stronger target for anti-angiogenic treatment in breast cancer. Ethically, working with genetic data like this means protecting patient privacy and making sure the data is used responsibly. Researchers have to make sure samples are anonymized and used only for scientific purposes. It’s also important to remember that genetic and molecular differences don’t tell the whole story things like access to care, race, and income can also influence who develops breast cancer and how it’s treated, so that should be kept in mind when interpreting the results.

Limitations and Future Work:

One limitation of this analysis is that there wasn’t a control group of healthy tissue to compare to, so the differences in gene expression are based only on cancer samples. Also, the model only looked at two genes, VEGFA and FGF1, which simplifies a much more complex process. K-means clustering is useful for spotting trends, but it doesn’t fully capture the overlap or relationships between tumor subtypes. In the future, it would be helpful to include more genes involved in angiogenesis, like THBS1 or p53, and possibly clinical data such as tumor stage or patient survival. Using other methods beyond clustering, like supervised learning, could also make the analysis more accurate. Finally, future studies should focus on how treatments that target angiogenesis could be made more effective and accessible for all patients, not just certain groups.

NOTES FROM YOUR TEAM:

No notes at this time.

QUESTIONS FOR YOUR TA:

No questions at this time.