Module_3: *Cancer*

Team Members:

Gabby Holohan & Meredith Lineweaver

Project Title:

Impact of E-Cadherin gene on Breast Cancer Prognosis

Project Goal:

This project seeks to analyze groupings of EMT-related gene expression in breast cancer patients, with the goal of identifying biological or clinical relationships.

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: Activating Invasion and Metastasis
- Overview of hallmark: Invasion and metastasis are categorized in a series of steps beginning with local invasion, followed by expansion to blood and lymphatic vessels, escape of cancer cells into tissues, and formation of tumors. In carcinoma cells, it is often found that a molecule responsible for shaping endothelial sheets, E-cadherin, is decreased, allowing cancer cells to migrate and the cancer to metastasize. Some transcription factors such as Snail, Slug, Twist, and Zeb1/2 help regulate the migratory process of cells, and some are even involved in the E-cadherin gene expression. The exact interactions and roles of these transcription factors on E-cadherin expression remain largely unknown. The general consensus, however, is clear that increasing expression of molecules aiding the assembly of endothelial sheets or structures like E-cadherin decreases the ability of cancer to metastasize while decreasing expression of molecules like E-cadherin enhances the cancer's ability to spread.
- Genes associated with hallmark to be studied (describe the role of each gene, signaling pathway, or gene set you are going to investigate): The EDH1 gene transcribes E-cadherin, a protein that allows endothelial cells to adhere to one another. Deficiency in E-cadherin results in Epithelial-mesenchymal transition (EMT): a condition which causes cells to become depolarized and unable to adhere to one another.
  https://www.sciencedirect.com/science/article/abs/pii/S0924224424000748?via%3Dihub

- Mutations, proteolytic cleavage, chromosomal deletions, epigenetic regulation and transcriptional silencing of CDH1 promoter may limit the functionality of E-cadherin, especially in gastric, breast, liver, pancreatic, and skin cancer
- Loss of E-cadherin activates EMT transcription factors --> metastasis/invasion
- E-cadherin regulates receptor tyrosine kinase (RTK) and tyrosine kinase Src
- E-cadherin-expressing cell line w/ increased nuclear factor kappa-light-chain-enhancer of activated B cells (NF-κB) activity and increased c-Myc expression promote cell proliferation when adenosine triphosphate (ATP) production increases, increasing glycolysis and oxidative phosphorylation rates
- N-cadherin is another protein in the cadherin family which contributes to EMT at increased levels. EMT is found to inhibit apoptosis via the death receptor 4 TNF-related apoptosis-inducing ligand-receptor 1 (TRAIL-R1) and/or death receptor 5 TRAIL-R2 https://pmc.ncbi.nlm.nih.gov/articles/PMC6830116/
- CDH2 is the gene related to N-cadherin expression, often involved in neurocognitive disease.
- Elevated expression of CDH2 is tied to decreased expression in EDH1 and the development of EMT https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.972059/fu

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

- Breast cancer is second most common cancer for women in US, 2nd leading cause of cancer death, and leading cause of cancer death for Hispanic and Black women https://www.cdc.gov/breast-cancer/statistics/index.html
- 279,731 new breast cancers in women in the US were reported in 2022, and 42,213 women in the US died from breast cancer in 2023
- Incidence is 132.9 per 100,000 in US annually https://www.cdc.gov/united-states-cancer-statistics/publications/breast-cancer-stat-bite.html

Risk factors (genetic, lifestyle) & Societal determinants

- being female
- older age
- family history
- personal history of cancer

- dense breasts
- physical inactivity
- being overweight or obese
- alcohol consumption
- hormone use/oral contraceptive use
- societal determinants: poverty, lack of education, unemployment, lower health literacy, lack of health insurance, living in disadvantaged neighborhoods, housing and food insecurity, delayed childbearing
- genetic mutations in BRCA1, BRCA2, ATM, BARD1, BRIP1, CHEK2, CDH1 https://www.cdc.gov/breast-cancer/risk-factors/index.html

Standard of care treatments (& reimbursement)

- surgery: mastectomies and lumpectomies
- radiation therapy
- chemotherapy
- targeted therapy drugs
- hormone therapy
- reimbursement depends on insurance type (ex. private, Medicare, or Medicaid) https://www.breastcancer.org/managing-life/covering-cost-of-care/cost-of-care-report https://www.cancer.gov/types/breast/hp/breast-treatment-pdq

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

- most breast cancers originate in the epithelial cells lining the ducts or the lobules
- genetic mutations accumulate in the breast epithelial cells: spontaneous or inherited mutations
- chromosomal rearrangements can activate oncogenes/silence tumor supressor genes - these rearrangements can be triggered by estrogen
- if a breast cancer is hormone receptor positive that means they grow in response to estrogen or progesterone, they can also overexpress HER2
- additional mutations can lead to invasion, angiogenesis, and metastasis https://link.springer.com/chapter/10.1007/978-3-319-21683-6_9

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.

The data analyzed for this project comes from a paper published in Bioinformatics titled "Alternative preprocessing of RNA-Sequencing data in The Cancer Genome Atlas leads to improved analysis results". These scientists collected data for the 9265 tumor and 741

normal samples across 24 cancer types using the Rsubread package. To obtain the RNA-Sequencing and clinical data, the scientists compared TCGA samples that were processed using pipelines and determined the Rsubread pipeline produced fewer errors and more consistent expression levels. For this project we only used data that contained information about the CDH1 gene.

Data Analyis:

Methods

The machine learning technique used in this dataset analysis is clustering. This method looks for groups within the dataset using EMT-related genes. Clustering is used to find patterns and groupings in expression of EMT-related genes and potential relation to patient survival within the breast cancer portion of the data set. We used PCA to improve identification of any possible clustering. To further analyze contributions of specific genes, we found the loadings of each gene using components in scikit learn. To analyze how well our trained model was performing we evaluated the ROC curve and AUC (more about this in the conclusion).

**

Analysis

Analysis of PCA: The PCA shows two distinct clusters, labeled 1 and 0 in the key. Patients' vital statuses are also noted. PC1 shows the most variance with 43.2% while PC2 has a variance of 14.1%. This means that PC1 contributes more heavily to the variance of the data set. Silhoutte coefficient: Our average silhoutte coefficient was 0.479. This value indicates that the clusters were moderately separated but had some overlap. Silhoutte coefficient by cluster: The average silhoutte coefficient per cluster is slightly higher than .5 with a couple of outliers excluded. This indicates good clustering. Gene contribution: This section indicates how much each gene contributes to each principal component. Positive loading indicates a positive relationship between the principal component and expression of the gene while negative loading shows the opposite. Additional analysis is included in the conclusion.

In [4]:
```
import pandas as pd

# === Step 1: Load files ===
# Replace these with your actual file paths
metadata = pd.read_csv("/Users/meredithlineweaver/Desktop/Computational BME/
expr = pd.read_csv("/Users/meredithlineweaver/Desktop/Computational BME/Modu

# === Step 2: Filter metadata for BRCA samples ===
brca_meta = metadata[metadata["cancer_type"] == "BRCA"]
```

```
# === Step 3: Match sample IDs between datasets ===
# Expression files often use TCGA sample barcodes (e.g., TCGA-XX-XXXX-01)
# Metadata uses bcr_patient_barcode (e.g., TCGA-XX-XXXX)
# So we'll shorten the expression column names to match the metadata barcode

expr.columns = expr.columns.str.slice(0, 12)  # keep first 12 chars for matc
expr_brca = expr.loc[:, expr.columns.isin(brca_meta["bcr_patient_barcode"])]

# === Step 4: Align metadata and expression data ===
# Keep only samples that appear in both
shared_samples = set(expr_brca.columns).intersection(set(brca_meta["bcr_pati
expr_brca = expr_brca.loc[:, list(shared_samples)]
brca_meta = brca_meta[brca_meta["bcr_patient_barcode"].isin(shared_samples)]

# === Step 5: Optional — reorder metadata to match expression order ===
brca_meta = brca_meta.set_index("bcr_patient_barcode").loc[expr_brca.columns

# === Step 6: Confirm shapes ===
print("Expression data shape:", expr_brca.shape)
print("Metadata shape:", brca_meta.shape)

# === Step 7: (Optional) Save cleaned BRCA datasets ===
expr_brca.to_csv("BRCA_log2TPM_filtered.csv")
brca_meta.to_csv("BRCA_metadata_filtered.csv")
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns



# === Step 4: Focus on EMT-related genes ===
# You can modify this list as needed
emt_genes = [
    "CDH1", "VIM", "SNAI1", "SNAI2", "TWIST1", "TWIST2",
    "ZEB1", "ZEB2", "FN1", "MMP2", "MMP9", "CLDN1", "CDH2"
]

emt_expr = expr_brca.loc[expr_brca.index.intersection(emt_genes)]

if emt_expr.empty:
    print("⚠️ None of the EMT genes were found in your dataset. Check gene r
else:
    print(f"Found {emt_expr.shape[0]} EMT genes in dataset.")

# === Step 5: PCA ===
# Transpose so samples are rows and genes are columns
X = emt_expr.T
X_scaled = StandardScaler().fit_transform(X)

pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_scaled)
```

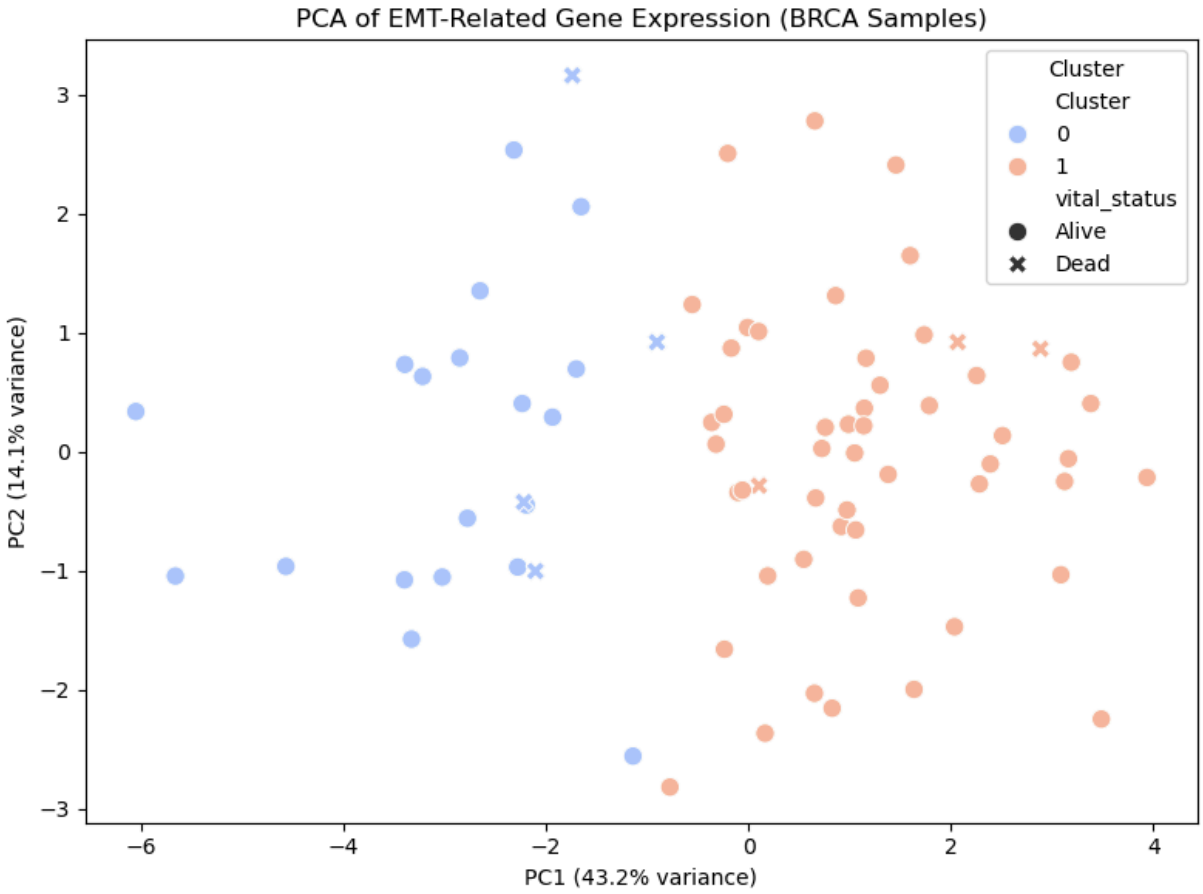
```python
pca_df = pd.DataFrame(pca_result, columns=["PC1", "PC2"], index=X.index)
pca_df = pca_df.join(brca_meta)

# === Step 6: K-means clustering (optional) ===
kmeans = KMeans(n_clusters=2, random_state=42, n_init=10)
pca_df["Cluster"] = kmeans.fit_predict(pca_result)

# === Step 7: Visualization ===
plt.figure(figsize=(8, 6))
sns.scatterplot(
    data=pca_df,
    x="PC1", y="PC2",
    hue="Cluster",
    style="vital_status" if "vital_status" in pca_df.columns else None,
    palette="coolwarm",
    s=80
)
plt.title("PCA of EMT-Related Gene Expression (BRCA Samples)")
plt.xlabel(f"PC1 ({pca.explained_variance_ratio_[0]*100:.1f}% variance)")
plt.ylabel(f"PC2 ({pca.explained_variance_ratio_[1]*100:.1f}% variance)")
plt.legend(title="Cluster", loc="best")
leg = plt.legend(title="Cluster")
plt.tight_layout()
plt.show()
from sklearn.metrics import silhouette_score, silhouette_samples

# === Step 7b: Evaluate clustering quality ===
silhouette_avg = silhouette_score(pca_result, pca_df["Cluster"])
print(f"Average silhouette score: {silhouette_avg:.3f}")

# Optional: look at silhouette per sample
silhouette_vals = silhouette_samples(pca_result, pca_df["Cluster"])
pca_df["Silhouette"] = silhouette_vals

# === Optional visualization ===
plt.figure(figsize=(6, 4))
sns.boxplot(data=pca_df, x="Cluster", y="Silhouette", palette="coolwarm")
plt.title("Silhouette Coefficient per Cluster")
plt.tight_layout()
plt.show()


# === Step 8: (Optional) Save outputs ===
pca_df.to_csv("BRCA_EMT_PCA_clusters.csv")
emt_expr.to_csv("BRCA_EMT_genes_expression.csv")
# === Step 9: Analyze Gene Contributions (Loadings) ===
loadings = pd.DataFrame(
    pca.components_.T,
    columns=["PC1_loading", "PC2_loading"],
    index=emt_expr.index
)

# Compute contribution magnitude
loadings["PC1_contrib_abs"] = np.abs(loadings["PC1_loading"])
loadings["PC2_contrib_abs"] = np.abs(loadings["PC2_loading"])
loadings["Total_contribution"] = loadings["PC1_contrib_abs"] + loadings["PC2
```

```python

# Sort genes by overall contribution
loadings_sorted = loadings.sort_values("Total_contribution", ascending=False
print("\nTop contributing genes to PCA components:")
print(loadings_sorted.head(10))

# === Step 10: Visualize gene importance ===
plt.figure(figsize=(10, 6))
sns.barplot(
    x=loadings_sorted.index,
    y="Total_contribution",
    data=loadings_sorted,
    palette="viridis"
)
plt.xticks(rotation=45, ha="right")
plt.title("Gene Contribution to PCA (EMT Genes)")
plt.ylabel("Total |Loading| (PC1 + PC2)")
plt.tight_layout()
plt.show()

# === Step 11: Save outputs ===
pca_df.to_csv("BRCA_EMT_PCA_clusters.csv")
emt_expr.to_csv("BRCA_EMT_genes_expression.csv")
loadings_sorted.to_csv("BRCA_EMT_PCA_loadings.csv")

# == Compare in-sample vs out-of-sample data ==
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, roc_auc_score

# === Step 1: Load BRCA training data ===
X_in = pd.read_csv("BRCA_EMT_genes_expression.csv", index_col=0).T
meta_in = pd.read_csv("BRCA_metadata_filtered.csv", index_col=0)
y_in = meta_in["vital_status"].map({"Alive": 0, "Dead": 1})

print("Training data:", X_in.shape, "Labels:", y_in.shape)

# === Step 2: Load TEST data ===
X_out = pd.read_csv("/Users/meredithlineweaver/Desktop/Computational BME/Moc
meta_out = pd.read_csv("/Users/meredithlineweaver/Desktop/Computational BME/

# Keep only BRCA samples if needed
if "cancer_type" in meta_out.columns:
    meta_out = meta_out[meta_out["cancer_type"] == "BRCA"]

# Keep only Alive/Dead
meta_out = meta_out[meta_out["vital_status"].isin(["Alive", "Dead"])]
meta_out = meta_out.set_index("bcr_patient_barcode")

# Match sample IDs between metadata + expression
X_out.columns = X_out.columns.str.slice(0, 12)
shared_samples = X_out.columns.intersection(meta_out.index)
X_out = X_out.loc[:, shared_samples]
meta_out = meta_out.loc[shared_samples]
```

```python
y_out = meta_out["vital_status"].map({"Alive": 0, "Dead": 1})

print("Test data:", X_out.shape, "Labels:", y_out.shape)

# === Step 3: Align gene names ===
X_in.columns = X_in.columns.str.upper().str.replace(r"\.\d+$", "", regex=Tru
X_out.index = X_out.index.str.upper().str.replace(r"\.\d+$", "", regex=True)

if X_out.shape[0] > X_out.shape[1]:  # genes are rows → transpose
    X_out = X_out.T

X_out.columns = X_out.columns.str.upper().str.replace(r"\.\d+$", "", regex=T

# Intersect genes
common_genes = X_in.columns.intersection(X_out.columns)
print("Common genes:", len(common_genes))

X_in_aligned = X_in[common_genes]
X_out_aligned = X_out[common_genes]

print("X_in_aligned:", X_in_aligned.shape)
print("X_out_aligned:", X_out_aligned.shape)

# === Step 4: Scale ===
scaler = StandardScaler()
X_in_scaled = scaler.fit_transform(X_in_aligned)
X_out_scaled = scaler.transform(X_out_aligned)

# === Step 5: Train ONE logistic regression ===
model = LogisticRegression(C=1.0, max_iter=1000)
model.fit(X_in_scaled, y_in)

# === Step 6: Compute in-sample accuracy & AUC ===
y_in_pred = model.predict(X_in_scaled)
in_acc = accuracy_score(y_in, y_in_pred)
in_auc = roc_auc_score(y_in, model.predict_proba(X_in_scaled)[:, 1])

# === Step 7: Compute out-of-sample accuracy & AUC ===
y_out_pred = model.predict(X_out_scaled)
out_acc = accuracy_score(y_out, y_out_pred)
out_auc = roc_auc_score(y_out, model.predict_proba(X_out_scaled)[:, 1])

# === Step 8: Print results ===
print("\n=== FINAL MODEL PERFORMANCE ===")
print(f"In-sample Accuracy: {in_acc:.4f}")
print(f"Out-of-sample Accuracy: {out_acc:.4f}")
print(f"In-sample AUC:      {in_auc:.4f}")
print(f"Out-of-sample AUC:   {out_auc:.4f}")

import matplotlib.pyplot as plt
import numpy as np

metrics = ["Accuracy", "AUC"]
in_vals = [in_acc, in_auc]
out_vals = [out_acc, out_auc]
```

```
x = np.arange(len(metrics))
width = 0.35

plt.figure(figsize=(7,5))
plt.bar(x - width/2, in_vals, width, label="In-sample")
plt.bar(x + width/2, out_vals, width, label="Out-of-sample")

plt.xticks(x, metrics)
plt.ylabel("Score")
plt.title("In-sample vs Out-of-sample Performance")
plt.legend()
plt.ylim(0, 1)
plt.tight_layout()
plt.show()
```

```
Expression data shape: (3000, 76)
Metadata shape: (76, 71)
Found 11 EMT genes in dataset.
```




```
Average silhouette score: 0.479
```

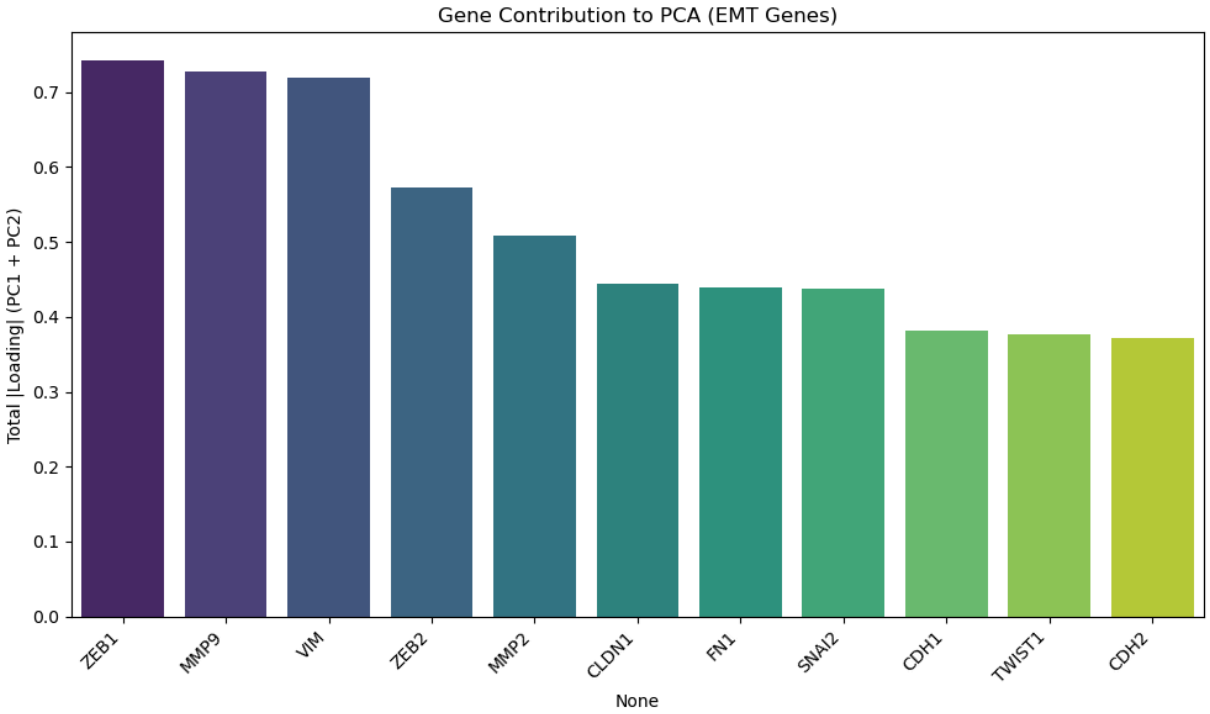
```
/var/folders/x5/0yw9gcvx1zzflc909kgzqbb00000gn/T/ipykernel_39125/1574380889.
py:103: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.boxplot(data=pca_df, x="Cluster", y="Silhouette", palette="coolwarm")
```

Top contributing genes to PCA components:

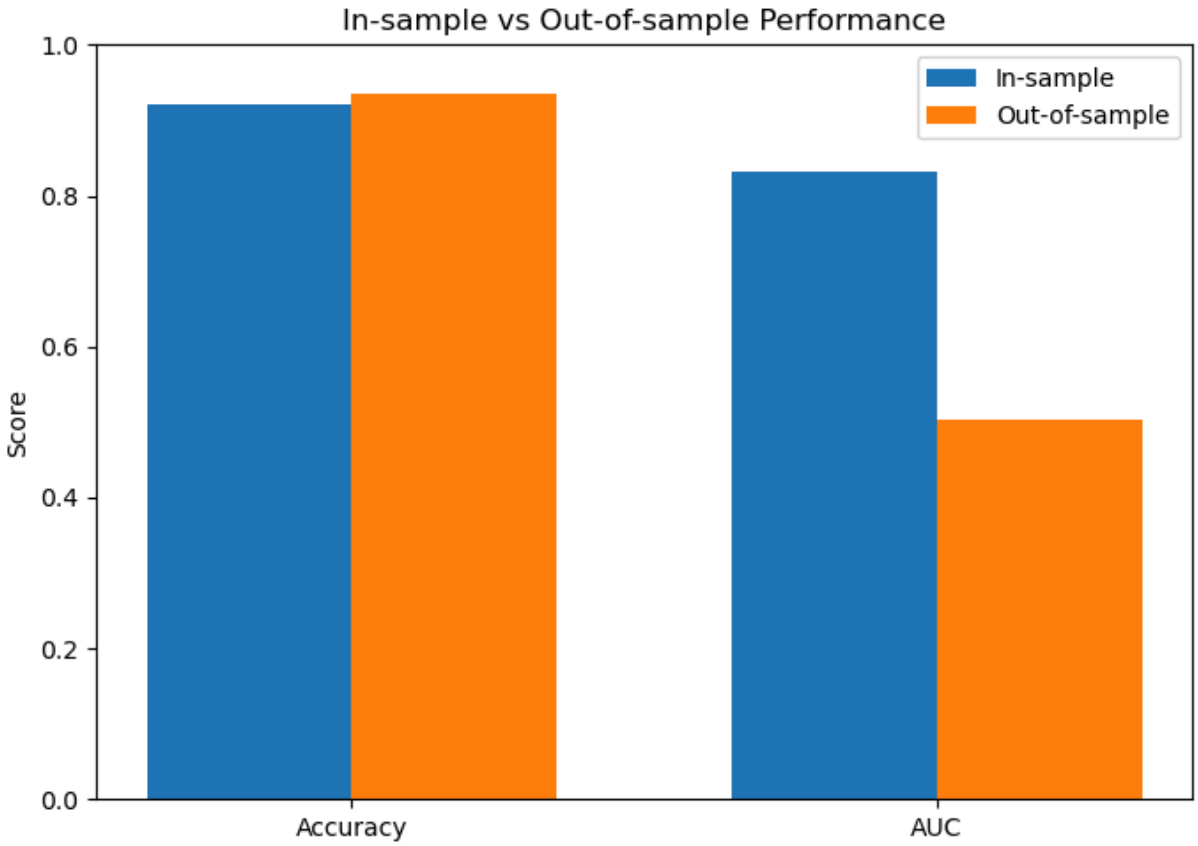
	PC1_loading	PC2_loading	PC1_contrib_abs	PC2_contrib_abs	\
ZEB1	0.371545	-0.371197	0.371545	0.371197	
MMP9	0.063668	0.664391	0.063668	0.664391	
VIM	0.322009	0.397236	0.322009	0.397236	
ZEB2	0.391333	-0.181713	0.391333	0.181713	
MMP2	0.402433	-0.105726	0.402433	0.105726	
CLDN1	0.190344	0.254228	0.190344	0.254228	
FN1	0.338272	-0.101816	0.338272	0.101816	
SNAI2	0.356892	0.081426	0.356892	0.081426	
CDH1	-0.036766	0.344881	0.036766	0.344881	
TWIST1	0.307035	0.069984	0.307035	0.069984	

	Total_contribution
ZEB1	0.742742
MMP9	0.728059
VIM	0.719246
ZEB2	0.573047
MMP2	0.508159
CLDN1	0.444571
FN1	0.440088
SNAI2	0.438318
CDH1	0.381647
TWIST1	0.377019

```
/var/folders/x5/0yw9gcvx1zzflc909kgzqbb00000gn/T/ipykernel_39125/1574380889.
py:131: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed
in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the
same effect.

  sns.barplot(
```




```
Training data: (76, 11) Labels: (76,)
Test data: (15716, 77) Labels: (77,)
Common genes: 11
X_in_aligned: (76, 11)
X_out_aligned: (77, 11)

=== FINAL MODEL PERFORMANCE ===
In-sample Accuracy: 0.9211
Out-of-sample Accuracy: 0.9351
In-sample AUC:      0.8323
Out-of-sample AUC:   0.5028
```



Verify and validate your analysis:

We used the receiver operating characteristic curve and the area under the ROC curve. The AUC values are both above 0.5 but less than 1.0 which means they fall somewhere between random guessing and perfect separation. This mewans the model is performing relatively well. This model works by evaluating the model's performance across all decisions thresholda and the AUC summarizes the entire crve into a single value.

To verify our analysis we first made sure the data was aligned by confirming the training and test datasets contained the same set of genes and the metadata labels correctly matched the expression data. Both the training and test sets were transformed using the StandardScalar object which prevents data leakage.

Initial validation concluded that the out-of-sample performance did not collapse and the ROC curve shape made sense because it was smooth and monotonic. We used a paper published in the Frontiers in Genetics journal titled "Feature selection for Breast Cancer Classification by Integrating Somatic Mutation and Gene Expression" written by Qin Jiang and Min Jin. This study focuses on a TCGA-BRCA gene-expression dataset and applies machine learning to their analysis. Their results highlight how important it is to validate classification accuracy properly to determing the effectiveness of your machine learning model. This supports the focus our analysis made on AUC rather than just accuracy.

https://www.frontiersin.org/journals/genetics/articles/10.3389/fgene.2021.629946/full

Conclusions and Ethical Implications:

We found that the genes ZEB1, MMP9, and VIM had the highest loading contribution to our PCA, indicating that these genes are driving the clustering indicated by our PCA. Additionally, PC1 contributed 43.2% variance to our PCA, indicating that PC1 is significant in dividing the dataset into clusters. Biologically, ZEB1, MMP9, and VIM are all genes involved in the activation of the EMT pathways, revealing that the primary variation in our dataset was EMT program activation. PC1's main contributors were ZEB1, ZEB2, Vimentin, Fibronectin, MMP2, TWIST1, and SNAI2, all with positive loadings. The heavy concentration of mesenchymal EMT factors indicates that PC1 is representative of the degree of mesenchymal EMT activation. PC2 was more variable with positive loading from VIM, MMP9, CDH1, and CLDN1, all top contributors, but negative loading from ZEB1, ZEB2, MMP2, FN1. This difference in sign indicates that the EMT process may be activated by different components along PC2. Overall, we can determine two biologically distinct clusters with an average silhoutte score of .479. Due to PC1's significant variance, we can infer that clustering may be due to differences in EMT activation. The model we generated demonstrates good predicitve ability and generalizes reasonably well to new data. The in-sample performance is consistent with the out-of-sample accuracy and AUC which indicates the model is not overfitting or underfitting the dataset.

With this conclusion, we must consider the ethical implications of our model. This model could inform further research into different EMT signaling promoting metastasis in breast cancer which could inform treatment and prevention of metasized breast cancer. However, the sensitive nature of this data requires protection of patient data; our model must ensure patient data is protected. Additionally, the model must be tested on more diverse samples of people to improve its accuracy and limit biases. In future work, if a distinction between EMT activation pathways is verified, researchers must consider the access to genomic testing and socioeconomic gaps that may impact treatment variations between patients.

Limitations and Future Work:

Our model was only tested on 75 samples of breast cancer patients, therefore the scope of our data was relatively small compared to the actual population. Small samples contributes to inaccuracy in a machine learning model, therefore, our model would need to be given an expanded data set to improve accuracy and to help mitigate the impact of outliers or a bias in the dataset.

Future work could include expanding this model on EMT related genes to all cancer types, expanding the sample for machine learning, or further inquiry into the clustering we identified. A future inquiry could investigate other genes or metadata associated with our clusters and expand our data analysis with k-means clustering or classification.

NOTES FROM YOUR TEAM:

- We originally identified the CDH1 gene as a gene of interest as its expression relates to EMT pathways that influence metastasis in cancers.
- After the company meeting on Thursday, we decided to broaden our genes of interest to other genes related to EMT pathways including CDH1, SNAI2, TWIST1, ZEB1, ZEB2, TGFB1, WNT5A, AXIN2, LEF1, GRHL2, and OVOL2.
- We decided to use clustering to show any potential patterns emerging when visualizing gene expression
- PCA was used to better visualize any potential clustering.
- We intend to further analyze the clusters and maybe compare the genes in the future.
- Look up loading and components in scikit learn.
- Try classification with alive vs dead with genes.
- Write why using each method in markdown
- If using classification, make decision boundary plot & ROC plot
- for clustering: silhoutte coefficient
- in sample vs out of sample error --> test model with out of sample data (posted on canvas) and in sample data and compare errors (underfit: low complexity, overfit: high complexity, well-fit: medium complexity) for M3 JNC3

QUESTIONS FOR YOUR TA:

- Still slightly confused about what PCA actually does and how to analyze the graph?

In [ ]: