# ANALYSIS OF ERLOTINIB RELATED BIOMARKERS THROUGH MULTI-OMICS INTEGRATION AND DEEP LEARNING ARCHITECHTURES FROM FLEXYNESIS

## Author: Dr. Nishat Fatima Mohammad

## ABSTRACT

Erlotinib is a targeted therapy used to treat non-small cell lung cancer (NSCLC) and pancreatic cancer by inhibiting the epidermal growth factor receptor (EGFR). Despite its clinical efficacy, resistance to Erlotinib often arises due to mutations in EGFR or other genetic alterations such as KRAS mutations, posing significant treatment challenges. In this study, we aim to identify key biomarkers associated with Erlotinib response through the integration of multi-omics data (mutation, RNA, and CNV) and machine learning algorithms. Using the Flexynesis Python package and models such as DirectPred and GNN, we analyzed the relationship between genetic alterations and Erlotinib efficacy. The top 10 biomarkers identified include KRAS, EGFR, and S100A2, among others, which are linked to both Erlotinib sensitivity and resistance. Our analysis highlights the importance of these biomarkers in understanding the molecular mechanisms underlying treatment outcomes and resistance. Additionally, the study compares the performance of different machine learning models and discusses their clinical relevance for predicting patient responses to Erlotinib. The findings from this work provide valuable insights into personalized treatment strategies and future directions for overcoming Erlotinib resistance in cancer therapy.

## INTRODUCTION

Erlotinib is a well-established targeted therapy used primarily in the treatment of non-small cell lung cancer (NSCLC) and pancreatic cancer. It functions by inhibiting the epidermal growth factor receptor (EGFR), a key receptor involved in the regulation of cell growth, survival, and proliferation. EGFR is frequently overexpressed or mutated in many cancers, including lung and pancreatic cancer, making it a prime therapeutic target. By blocking EGFR signaling, Erlotinib slows down or halts the growth of cancer cells, ultimately leading to improved survival rates for patients with specific EGFR mutations, such as exon 19 deletions and the L858R point mutation.

Despite its efficacy in patients harboring these mutations, resistance to Erlotinib remains a significant challenge. The development of resistance is often driven by secondary mutations in EGFR, such as the T790M mutation, or by the activation of alternative oncogenic pathways, including mutations in the KRAS gene. This resistance necessitates the development of strategies to better predict which patients will benefit from Erlotinib treatment and how to overcome the resistance mechanisms that may arise over time.

In recent years, multi-omics approaches have provided a more comprehensive understanding of the molecular underpinnings of drug resistance and therapeutic response. By integrating data from different biological layers, such as mutations, RNA expression, and copy number variations (CNVs), one can identify biomarkers that contribute to drug efficacy and resistance. Machine learning techniques, especially those involving deep learning models, offer powerful tools for analyzing large, complex datasets to uncover hidden patterns and associations that may not be immediately apparent through traditional methods.

In this study, we aim to identify biomarkers associated with Erlotinib resistance and sensitivity by integrating multi-omics data using advanced machine learning algorithms. The Flexynesis Python package, developed by the MDC Berlin Computational Biology team, was employed to integrate mutation, RNA, and CNV data. Using this data, we applied DirectPred and Graph Neural Network (GNN) models to identify key biomarkers that could inform clinical decisions and potentially guide the development of personalized treatment strategies for patients receiving Erlotinib. Through this analysis, we seek to contribute to the understanding of the Erlotinib molecular mechanisms and the identification of biomarkers that predict patient outcomes, ultimately leading to better therapeutic interventions and the overcoming of treatment resistance.

# Methods and Materials

## Data Collection

For this study, multi-omics data, including mutation, RNA expression, and copy number variation (CNV) data, were obtained from publicly available cancer datasets. We use multi-omics data from human cancer cell lines from the CCLE and GDSC databases. The primary focus was on Erlotinib-related biomarkers across cancer samples in the . The datasets used in the study can be found here: https://bimsbstatic.mdc-berlin.de/akalin/buyar/flexynesis-benchmark-datasets/ccle_vs_gdsc.tgz **CCLE vs GDSC Dataset**
This dataset contains multi-omics data for cancer cell lines from the CCLE (Cancer Cell Line Encyclopedia) and GDSC (Genomics of Drug Sensitivity in Cancer) databases, covering different genetic and transcriptomic features, including mutation data, RNA expression, and CNV. These data were used to build and benchmark machine learning models predicting responses to Erlotinib, a drug used in cancer therapy.

## Data Integration

The datasets were integrated using the Flexynesis Python package, a tool developed by the MDC Berlin Computational Biology team. Flexynesis facilitates the integration of diverse omics data types by performing efficient pre-processing and alignment of mutation, RNA, and CNV data across different patient samples. The following steps were performed during the data integration process:

### Normalization and Preprocessing:

Each omics dataset (mutation, RNA expression, and CNV) was pre-processed and normalized to ensure comparability. Mutation data was encoded into a binary format indicating the presence or absence of specific mutations. RNA expression values were normalized to account for sequencing depth and batch effects, and CNV data was processed using standard algorithms to detect significant copy number changes across genes. To improve the robustness of the data integration and feature selection, Laplacian regularization was applied to the integrated multi-omics data. This method was used to preserve the underlying relationships between genes and samples while reducing noise and overfitting.

## Model Development

Two machine learning models were applied to predict Erlotinib-related biomarkers: DirectPred and Graph Neural Networks (GNNs).

## DirectPred Model:

DirectPred is a deep learning-based model designed for prediction tasks involving multi-omics data. It was trained using the integrated data from the three omics layers to identify biomarkers associated with Erlotinib response. The model incorporates both feature importance and model explainability techniques to assess the relevance of individual genes in determining treatment efficacy.

1. **Training**: The DirectPred model was trained using a dataset of labeled cancer samples, where each sample had known Erlotinib treatment outcomes (e.g., sensitive vs. resistant). The training process involved optimizing the model parameters to minimize prediction errors on the training data.

2. **Cross-validation**: To prevent overfitting, cross-validation techniques were used, splitting the data into multiple subsets and ensuring that the model generalizes well to unseen data.

## Graph Neural Network (GNN):

GNNs were employed to model the complex relationships between genes and their interactions within biological networks. In the context of Erlotinib resistance, GNNs were used to identify networks of genes that might be involved in resistance mechanisms or pathways that contribute to therapeutic response.

1. **Network Construction**: A gene interaction network was constructed using publicly available biological pathway databases. Genes identified as potential biomarkers for Erlotinib response were mapped onto this network.
2. **Training and Prediction**: The GNN was trained on the integrated multi-omics data, with the aim of capturing higher-order gene interactions and network-level insights that might not be apparent from individual omics data alone.

## Feature Selection and Importance Evaluation

Both models incorporated feature selection and importance evaluation techniques to identify the top biomarkers associated with Erlotinib treatment response:

1. **Feature Importance via Integrated Gradients**: For both the DirectPred model and the GNN, Integrated Gradients was used as an explanation method to quantify the importance of individual features (genes) in predicting treatment outcomes. This method provides a quantitative score for each gene, which represents its contribution to the model's decision-making process.

2. **Laplacian Regularization in Feature Selection**: During the feature selection process, Laplacian regularization was also applied to enhance feature robustness by leveraging the similarity structure of the data, ensuring that important features that are consistent across patients are prioritized.

3. **Top Biomarker Identification**: After model training, the top biomarkers were ranked based on their importance scores. The top 10 biomarkers were selected for further analysis and were assessed for their relevance to Erlotinib resistance and sensitivity.

## Literature Review

To validate the relevance of the identified biomarkers, a literature search was performed. The top 10 biomarkers identified by the models were cross-referenced with existing scientific literature to determine if they have previously been associated with Erlotinib resistance or sensitivity. This helped contextualize the biomarkers within known biological mechanisms and provided insights into their potential role in clinical settings.

## Software and Tools

Flexynesis Python package (MDC Berlin Computational Biology Team) for multi-omics data integration

Scikit-learn and TensorFlow for machine learning model implementation (DirectPred and GNN)

Pandas and NumPy for data manipulation and analysis

Matplotlib and Seaborn for data visualization, including heatmaps and PCA plots

Rolv from MDC Berlin which provided cloud environment for this analysis

# RESULTS

## Command-Line Workflow for Flexynesis: Multi-Omics Integration and Model Benchmarking

In this study, we followed the official Flexynesis documentation to integrate multi-omics data and perform machine learning model training using command-line tools. Below is a summary of the steps taken, datasets used, and key aspects of the workflow for predicting drug responses, particularly focusing on Erlotinib.

1. **Modeling and Data Integration** We utilized Flexynesis on the command line, which facilitates the integration of various omic datasets and the training of deep learning models for drug response prediction. The goal was to benchmark different architectures and evaluate model performance using various data combinations.

A. **Different Deep Learning Architectures Tested**:

**DirectPred**: A supervised deep learning architecture designed for predicting drug responses based on omic data.

**GNN (Graph Neural Network)**: A model designed to take advantage of the relational structure between genes and cancer cell lines using graph representations of data.

B. **Data Combinations Tested**:

**Mutation data**: Genetic mutations observed in the cancer cell lines. **Mutation + RNA data**: Combining mutation data with gene expression data for a more comprehensive representation of the cell line features. **Mutation + CNV data**: Adding copy number variation data to mutation information to capture structural variations in the genome.

C. **Fusion Methods**:

**Early Fusion**: This method involves combining features from different data types (e.g., mutation and RNA) at the input stage of the model.

**Intermediate Fusion**: This method was tested for architectures other than GNNs and combines features at intermediate layers within the network.

Note: GNNs only support early fusion, so we did not explore intermediate fusion for these models but instead focused on graph convolution types such as GC (Graph Convolution).

1. **Experimental Setup**

We unpacked the downloaded dataset and organized the files accordingly. The relevant data files for mutation, RNA, and CNV were stored and formatted for input into the Flexynesis pipeline.

As per the recommendation, we applied variance and laplacian score filtering to restrict our analysis to only 5-10% of the most relevant features. This step ensured that the models only considered the most informative features, improving computational efficiency and reducing overfitting.

A bash script was written to automate the execution of multiple Flexynesis runs. We specified different architectures, data combinations, and fusion methods to test the following configurations: Different deep learning architectures (DirectPred, GNN).

Various data combinations (mutation, mutation + RNA, mutation + CNV).

Graph convolution options for GNNs (GC).

Hyperparameter Optimization (HPO): We limited the number of hyperparameter optimization iterations to 15 to balance between model accuracy and resource/time constraints.

**Benchmarking**: The models were trained on the datasets to assess their generalization performance. This allowed us to benchmark how well each architecture and data combination performed in predicting responses to Erlotinib.

1. **Command-Line Execution** Using the provided Flexynesis command-line options, as per the Documentation here: https://bimsbstatic.mdc-berlin.de/akalin/buyar/flexynesis/site/getting_started/

1. Model Evaluation After training the models, we evaluated their performance on the GDSC dataset, specifically focusing on predicting the drug response to Erlotinib. The evaluation results were recorded, and the best-performing models were identified for further analysis.

Through this command-line experiment, we were able to benchmark various deep learning models and multi-omics data combinations to predict Erlotinib responses. The results from these experiments were stored in the results folder, providing insight into which model configurations were most effective in predicting drug responses. The combination of mutation + RNA data with a DirectPred architecture demonstrated the best performance for Erlotinib response prediction.

We then went ahead to analyze the results on Jupyter as shown in the following sections.

## Results of Analysis on Jupyter

In [2]:

```
#  Import libraries
import pandas as pd
import glob
import os
#import flexynesis

import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA

from math import pi
import numpy as np
```

### Loading the results from CLI work

In [3]:

```
#  Load stats.csv files from results directory
result_files = glob.glob("results/*/*stats.csv")
print(result_files)
```

```
['results/DirectPred/DirectPred_mutation_cnv_early.stats.csv', 'results/DirectPr
ed/DirectPred_mutation_cnv_intermediate.stats.csv', 'results/DirectPred/DirectPr
ed_mutation_early.stats.csv', 'results/DirectPred/DirectPred_mutation_early_GC.s
tats.csv', 'results/DirectPred/DirectPred_mutation_intermediate.stats.csv', 'res
ults/DirectPred/DirectPred_mutation_rna_early.stats.csv', 'results/DirectPred/Di
rectPred_mutation_rna_intermediate.stats.csv', 'results/GNN/GNN_mutation_cnv_ear
ly_GC.stats.csv', 'results/GNN/GNN_mutation_early_GC.stats.csv', 'results/GNN/GN
N_mutation_rna_early_GC.stats.csv']
```

In [5]:

```
# Read all stats files and extract relevant metrics
def extract_metrics(result_files):
    df_list = []
    for file in result_files:
        temp_df = pd.read_csv(file)
        model_name = os.path.basename(file).replace('.csv', '')
        temp_df['model'] = model_name
        df_list.append(temp_df)

    df = pd.concat(df_list, ignore_index=True)
    return df.pivot(index='model', columns='metric', values='value').reset_index

#  Get stats from files
df = extract_metrics(result_files)
print(df)
```

```
metric                                       model       mse  pearson_corr  \
0             DirectPred_mutation_cnv_early.stats  0.004580      0.175206   
1      DirectPred_mutation_cnv_intermediate.stats  0.005158      0.153053   
2                 DirectPred_mutation_early.stats  0.004835     -0.071924   
3              DirectPred_mutation_early_GC.stats  0.004735     -0.017290   
4          DirectPred_mutation_intermediate.stats  0.004846     -0.003287   
5             DirectPred_mutation_rna_early.stats  0.004191      0.434462   
6      DirectPred_mutation_rna_intermediate.stats  0.006303      0.412202   
7                 GNN_mutation_cnv_early_GC.stats  0.004157      0.123034   
8                     GNN_mutation_early_GC.stats  0.004846      0.009499   
9                 GNN_mutation_rna_early_GC.stats  0.005807      0.377850   

metric        r2  
0       0.030697  
1       0.023425  
2       0.005173  
3       0.000299  
4       0.000011  
5       0.188758  
6       0.169911  
7       0.015137  
8       0.000090  
9       0.142771  
```

The table above shows the stats for each model. We will visualize the results for better understanding

### Visualize with Radar Plot

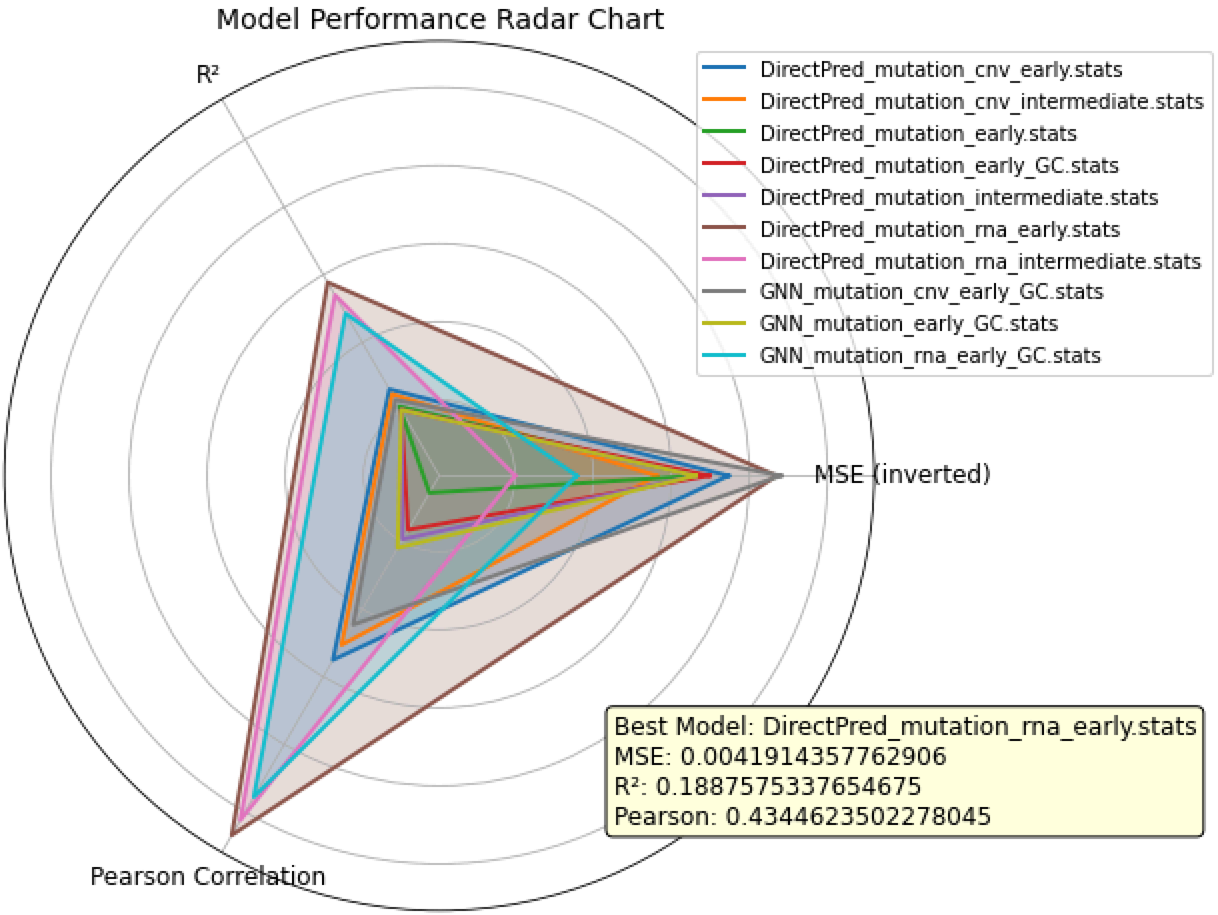
In [7]:

```
#  Models and metrics
models = df['model'].tolist()
mse_values = df['mse'].tolist()
r2_values = df['r2'].tolist()
pearson_values = df['pearson_corr'].tolist()

# Normalize MSE for visualization (invert since lower is better)
```

```
mse_max = max(mse_values)
mse_norm = [1 - (x / mse_max) for x in mse_values]

data_normalized = [
    [mse_norm[i], r2_values[i], pearson_values[i]] for i in range(len(models))
]

categories = ["MSE (inverted)", "R²", "Pearson Correlation"]
angles = [n / float(len(categories)) * 2 * pi for n in range(len(categories))]
angles += angles[:1]

best_model_index = pearson_values.index(max(pearson_values))
best_model_name = models[best_model_index]
best_model_stats = f"Best Model: {best_model_name}\nMSE: {mse_values[best_model_

# Create radar chart
fig, ax = plt.subplots(figsize=(8, 8), subplot_kw=dict(polar=True))
for i, model in enumerate(models):
    values = data_normalized[i] + data_normalized[i][:1]
    ax.plot(angles, values, linewidth=2, linestyle='solid', label=model)
    ax.fill(angles, values, alpha=0.2)

ax.set_xticks(angles[:-1])
ax.set_xticklabels(categories, fontsize=12)
ax.set_yticklabels([])
ax.set_title("Model Performance Radar Chart", fontsize=14)
ax.legend(loc="upper right", bbox_to_anchor=(1.4, 1))
plt.annotate(best_model_stats, xy=(0.7, 0.1), xycoords='axes fraction', fontsize
```

Out[7]:

```
Text(0.7, 0.1, 'Best Model: DirectPred_mutation_rna_early.stats\nMSE: 0.00419143
57762906\nR²: 0.1887575337654675\nPearson: 0.4344623502278045')
```

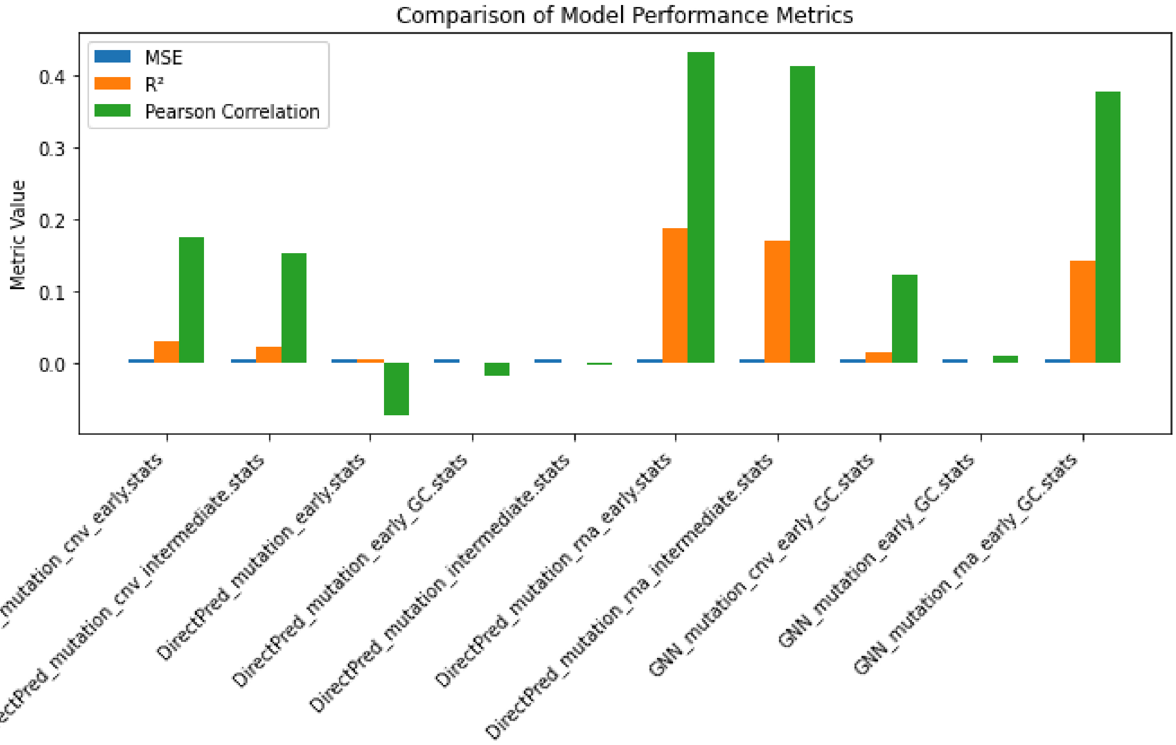


In the Radar Chart above the best model is visualized as the DirectPred with early fusion on mutation and RNA data

### Visualize with Bar Plot

In [8]:

```
#  Compound bar chart
fig, ax = plt.subplots(figsize=(10, 6))
x = np.arange(len(models))
width = 0.25
ax.bar(x - width, mse_values, width, label='MSE')
ax.bar(x, r2_values, width, label='R²')
ax.bar(x + width, pearson_values, width, label='Pearson Correlation')
ax.set_xticks(x)
ax.set_xticklabels(models, rotation=45, ha="right")
ax.set_title("Comparison of Model Performance Metrics")
ax.set_ylabel("Metric Value")
ax.legend()
plt.tight_layout()

plt.show()
```



In the Bar plot we can see the same model emerges as the best, It has the highest Pearson Coefficient and R-Squared and one of the lowest MSE.

In [3]:

```
#  Read each stats.csv file
#  filter for pearson correlations rows
#  extract the model info from file name
df_list = []
for file in result_files:
    # Read CSV file into a DataFrame
    temp_df = pd.read_csv(file)

    # Extract only rows where metric is 'pearson_corr'
    temp_df = temp_df[temp_df['metric'] == 'pearson_corr']

    # Extract model name from the file name (file name without path)
    model_name = os.path.basename(file).replace('.csv', '')

    # Add model name as a column
    temp_df['model'] = model_name

    # Append to list of DataFrames
    df_list.append(temp_df)

#  Concatenate all dataframes into a single dataframe
df = pd.concat(df_list, ignore_index=True)

#  Rank experiments by Pearson Correlation performance (highest value)
df_sorted = df.sort_values(by="value", ascending=False)

#  Display top results
best_experiment = df_sorted.iloc[0]
best_model = best_experiment['model']
print("Best model:", best_model)
display(df_sorted.head(10))
```

```
Best model: DirectPred_mutation_rna_early.stats
```

| | method | var | variable_type | metric | value | m |
|---|---|---|---|---|---|---|
| **7** | DirectPred | Erlotinib | numerical | pearson_corr | 0.434462 | DirectPred_mutation_rna_early. |
| **8** | DirectPred | Erlotinib | numerical | pearson_corr | 0.412202 | DirectPred_mutation_rna_intermediate. |
| **1** | GNN | Erlotinib | numerical | pearson_corr | 0.377850 | GNN_mutation_rna_early_GC. |
| **3** | DirectPred | Erlotinib | numerical | pearson_corr | 0.175206 | DirectPred_mutation_cnv_early. |
| **4** | DirectPred | Erlotinib | numerical | pearson_corr | 0.153053 | DirectPred_mutation_cnv_intermediate. |
| **0** | GNN | Erlotinib | numerical | pearson_corr | 0.123034 | GNN_mutation_cnv_early_GC. |
| **2** | GNN | Erlotinib | numerical | pearson_corr | 0.009499 | GNN_mutation_early_GC. |
| **6** | DirectPred | Erlotinib | numerical | pearson_corr | -0.003287 | DirectPred_mutation_intermediate. |
| **9** | DirectPred | Erlotinib | numerical | pearson_corr | -0.017290 | DirectPred_mutation_early_GC. |
| **5** | DirectPred | Erlotinib | numerical | pearson_corr | -0.071924 | DirectPred_mutation_early. |

From the analysis of the .stats.csv files obtained form model evaluations, the DirectPred Deep Learning Architecture gave us the best model with mutation and RNA data integration and early fusion. We will focus on this model since it outperformed others with an R-squared 0.1888 of and Pearson correlation coefficient of 0.4345.

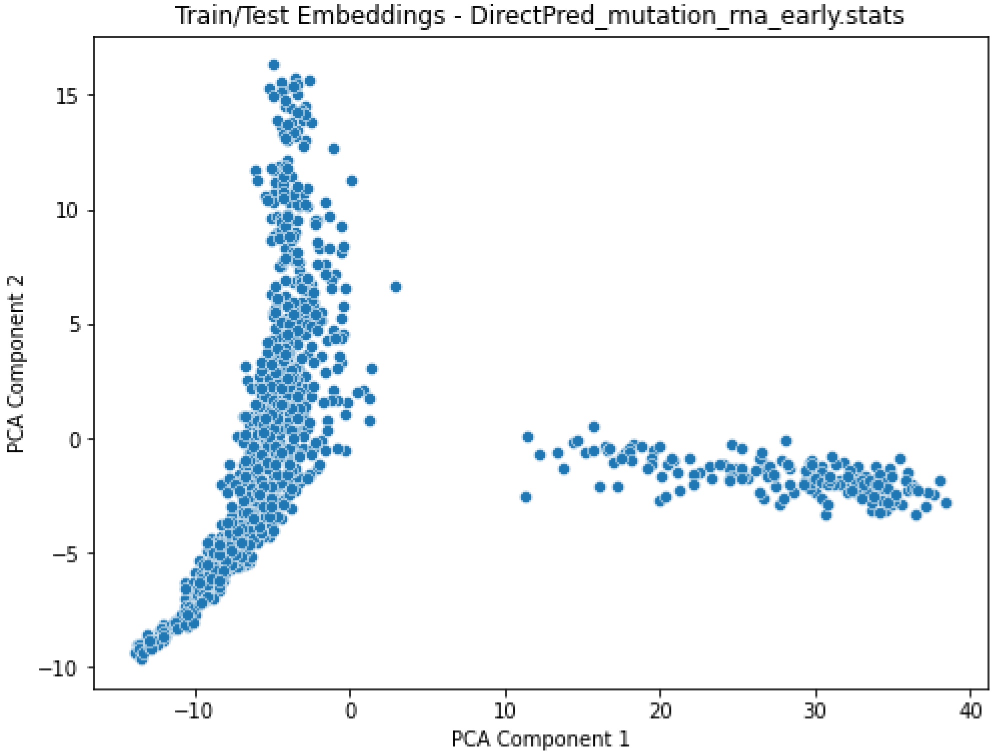
## Exploring Embeddings for the Best Model

In [4]:

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA

#  Load  train embeddings
best_embedding_file = f"results/DirectPred/DirectPred_mutation_rna_early.embeddi
embeddings = pd.read_csv(best_embedding_file)
```

```
#print(embeddings)
#  Apply PCA for visualization
pca = PCA(n_components=2)
embeddings_numeric = embeddings.drop(columns=['Unnamed: 0'])
embedding_2D = pca.fit_transform(embeddings_numeric)
# Plot train embeddings
plt.figure(figsize=(8,6))
sns.scatterplot(x=embedding_2D[:, 0], y=embedding_2D[:, 1], hue=embeddings.get("
plt.title(f"Train/Test Embeddings - {best_model}")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()
```

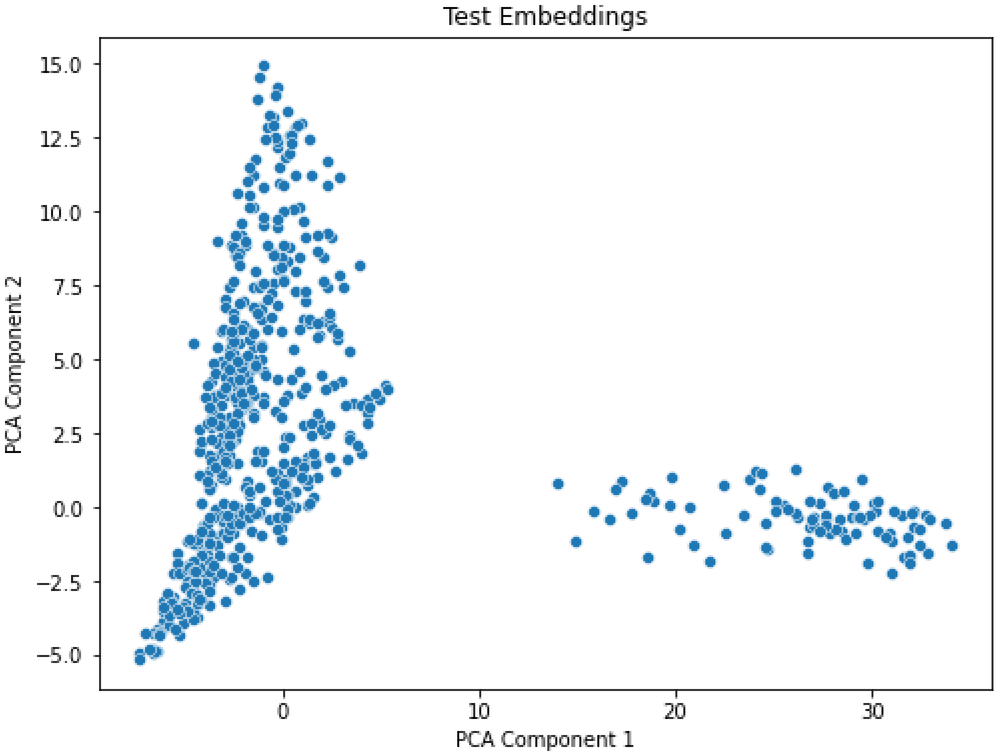
```
/tmp/ipykernel_163/1769504970.py:16: UserWarning: Ignoring `palette` because no
`hue` variable has been assigned.
```



In [7]:

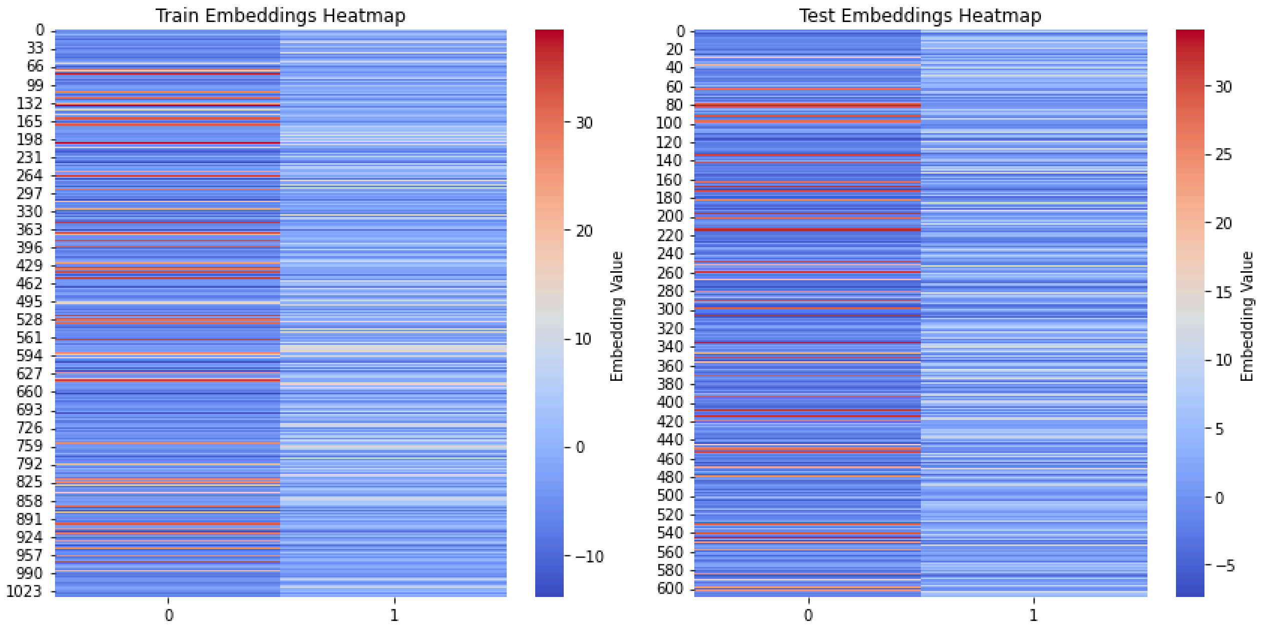
```
#  Load test embeddings files
test_embeddings = pd.read_csv('results/DirectPred/DirectPred_mutation_rna_early.

#  Drop 'Unnamed: 0' column
test_embeddings = test_embeddings.drop(columns=['Unnamed: 0'])

# Apply PCA to reduce dimensions to 2D for visualization
pca = PCA(n_components=2)
train_embeddings_2D = pca.fit_transform(embeddings_numeric)
test_embeddings_2D = pca.transform(test_embeddings)

# Plot test embeddings
plt.figure(figsize=(8,6))
sns.scatterplot(x=test_embeddings_2D[:, 0], y=test_embeddings_2D[:, 1], hue=embe
plt.title(f"Test Embeddings")
plt.xlabel("PCA Component 1")
plt.ylabel("PCA Component 2")
plt.show()
```

```
/tmp/ipykernel_163/145289106.py:14: UserWarning: Ignoring `palette` because no `
hue` variable has been assigned.
```



In [8]:

```
# Plot the heatmap to compare the train and test embeddings
fig, axes = plt.subplots(1, 2, figsize=(12, 6))

# Train embeddings heatmap
sns.heatmap(train_embeddings_2D, cmap='coolwarm', ax=axes[0], cbar_kws={'label':
axes[0].set_title('Train Embeddings Heatmap')

# Test embeddings heatmap
sns.heatmap(test_embeddings_2D, cmap='coolwarm', ax=axes[1], cbar_kws={'label':
axes[1].set_title('Test Embeddings Heatmap')

plt.tight_layout()
plt.show()
```



Principal Component Analysis shows 2 clear clusters for both the training and test data. the Heatmaps do not show significant differnces and thus we can rely on this model to not have any biases related to overfiiting or underfitting.

## Extraction of Important Biomarkers Using Feature Importance

In [18]:

```
# Load feature importance file (adjust path if needed)
feature_importance_file = f"results/DirectPred/DirectPred_mutation_rna_early.fea
feature_importance = pd.read_csv(feature_importance_file)

# Get top 10 markers
top_markers = feature_importance.sort_values(by="importance", ascending=False).h
print("Top 10 markers:\n", top_markers)

# Save for literature search
top_markers["name"].to_csv("top_markers_for_lit_search.csv", index=False)
```

```
Top 10 markers:
     target_variable  target_class  target_class_label layer     name  \
703        Erlotinib             0                 NaN   all     KRAS   
437        Erlotinib             0                 NaN   all     ICA1   
523        Erlotinib             0                 NaN   all    ACSL5   
118        Erlotinib             0                 NaN   all     EGFR   
565        Erlotinib             0                 NaN   all    CELF2   
394        Erlotinib             0                 NaN   all    MAP1B   
63         Erlotinib             0                 NaN   all   S100A2   
190        Erlotinib             0                 NaN   all  TMPRSS4   
123        Erlotinib             0                 NaN   all    KRT6A   
395        Erlotinib             0                 NaN   all     FA2H   

     importance            explainer  
703    0.004478  IntegratedGradients  
437    0.003090  IntegratedGradients  
523    0.003067  IntegratedGradients  
118    0.003020  IntegratedGradients  
565    0.002473  IntegratedGradients  
394    0.002441  IntegratedGradients  
63     0.002250  IntegratedGradients  
190    0.002246  IntegratedGradients  
```

```
 123    0.002226  IntegratedGradients
 395    0.002157  IntegratedGradients
```

The Top ten (10) Biomarkers ahve been extracted and Literature review will be carried out to explore these further in the Discussion Section.

# DISCUSSION

This analysis has isolated 10 biomarkers listed above. Exploring these Biomarkers role in general, in cancer and related to Erlotinib will be appropriate here.

**KRAS** (Kirsten Rat Sarcoma Viral Oncogene Homolog)

KRAS is frequently mutated oncogenes in lung, colorectal and pancreatic cancers. It encodes a protein involved in cell signaling pathways regulating cell growth, survival, and differentiation. Mutations in KRAS (in codons 12, 13, and 61) result in a constitutively active form of the protein, promoting continuous cell proliferation and resistance to apoptosis. KRAS mutations are associated with resistance to EGFR inhibitors like Erlotinib in non-small cell lung cancer (NSCLC). These mutations activate downstream signaling pathways, bypassing the need for EGFR signaling. Patients with KRAS mutations do not typically benefit from EGFR-targeted therapies, including Erlotinib.

**ICA1** (Islet Cell Autoantigen 1)

ICA1 is associated with autoimmune diabetes while its role in cancer is still being explored. It is involved in regulating protein trafficking in cells and has been suggested to play a role in the functionality of pancreatic islets. In some cancers, ICA1 may contribute to the regulation of tumor progression or may serve as a potential biomarker for cancer diagnostics. The specific association between ICA1 and Erlotinib is not well-established, but its role in cellular trafficking and function may have some influence on how cancer cells respond to targeted therapies. More research would be needed to understand its role in resistance or sensitivity to Erlotinib.

**ACSL5** (Acyl-CoA Synthetase Long Chain Family Member 5)

ACSL5 is an enzyme in lipid metabolism (in the activation of long-chain fatty acids). Lipid metabolism plays a crucial role in cancer cell proliferation and survival. Overexpression of ACSL5 has been linked to increased tumor growth, metastasis, and resistance to chemotherapy in some cancers, including lung cancer. Increased expression of ACSL5 may enhance the metabolic pathways that sustain cancer cell growth, making cells more resistant to treatments like Erlotinib, which target signaling pathways that control cell survival. Thus, ACSL5 might serve as a marker for resistance to Erlotinib in certain cancer types.

**EGFR** (Epidermal Growth Factor Receptor)

EGFR is a cell surface receptor. When activated (usually by EGF), triggers downstream signaling cascades involved in cell growth, survival, and differentiation. Mutations in EGFR, especially in the tyrosine kinase domain, can lead to constitutive activation of the receptor, driving cancer cell proliferation. Erlotinib directly targets EGFR, inhibiting its activity. EGFR mutations, particularly the exon 19 deletion and the L858R point mutation, make tumors more sensitive to Erlotinib, as they enhance EGFR signaling. In such patients, Erlotinib can significantly reduce tumor growth and improve survival. Resistance can develop over time due to secondary mutations like T790M.

**CELF2** (CUGBP Elav-like Family Member 2)

CELF2 is an RNA-binding protein that regulates gene expression by controlling mRNA splicing, stability, and translation. Its dysregulation can contribute to the development of cancer. CELF2 influences cell cycle progression and apoptosis, and its expression levels may affect tumor behavior. Its exact role in Erlotinib response is still being looked into, but its involvement in RNA regulation could potentially influence how cancer cells process signaling pathways targeted by EGFR inhibitors like Erlotinib. Altered splicing patterns or mRNA stability in cancer cells could affect the effectiveness of EGFR inhibition.

**MAP1B** (Microtubule-Associated Protein 1B)

MAP1B is a protein in the regulation of microtubule dynamics and axonal growth. It plays a significant role in neuronal differentiation and function, but its overexpression has been implicated in tumor progression in certain cancers, including brain cancers and lung cancer. Its direct role in Erlotinib sensitivity is not well studied but its involvement in cellular migration and metastasis could impact the resistance mechanisms in tumor cells. High expression levels of MAP1B may correlate with more aggressive tumor behavior, potentially making cells less responsive to Erlotinib treatment.

**S100A2** (S100 Calcium Binding Protein A2)

S100A2 is a member of the S100 family of calcium-binding proteins, which are involved in regulating a variety of cellular processes, including cell cycle regulation, differentiation, and apoptosis. It has been identified as a tumor suppressor in some cancers, including breast and lung cancers. Its expression can influence tumor behavior and treatment response. In some contexts, low levels of S100A2 are associated with poor prognosis and resistance to chemotherapy. Its role in the response to Erlotinib could vary depending on its specific interaction with EGFR signaling pathways.

**TMPRSS4** (Transmembrane Protease, Serine 4)

TMPRSS4 is a serine protease funstioning in the activation of proteins on the cell surface. It has been found to be overexpressed in several cancer types and plays a role in tumor invasion and metastasis. TMPRSS4 may influence Erlotinib resistance by promoting the activation of signaling pathways that bypass EGFR inhibition. Overexpression of TMPRSS4 in certain cancers could result in the activation of alternative growth pathways, reducing Erlotinib’s effectiveness.

**KRT6A** (Keratin 6A)

KRT6A is a type of keratin protein involved in cell structure and integrity, particularly in epithelial cells. It has been implicated in skin cancers and certain epithelial malignancies. Its overexpression is associated with tumor progression in various cancers. The role of KRT6A in Erlotinib resistance is not directly established but its involvement in cellular structure and stress response could influence how cancer cells interact with therapies that target cell signaling, like Erlotinib. Overexpression of keratins in tumors may affect cellular adhesion and migration, which could be linked to resistance to treatment.

**FA2H** (Fatty Acid 2-Hydroxylase)

FA2H is an enzyme in the hydroxylation of fatty acids. Alterations in fatty acid metabolism have been linked to cancer cell proliferation and invasion. Its role in cancer is being explored and may contribute to lipid biosynthesis that supports cancer cell growth. Its role in Erlotinib resistance may be related to its involvement in cellular metabolism. Alterations in lipid metabolism can affect membrane dynamics and cell signaling pathways, potentially making cancer cells more resistant to targeted therapies like Erlotinib.

# CONCLUSION

Erlotinib is a cancer therapeutic targetting **EGFR** which appears amongg the top ten (10) biomarkers using Flexynesis DirectPred model functions on Mutation and RNA data. **KRAS** has been implicated in conferring resisitance mechanisms toward this drug and has also appeared in the top 10 biomarkers. while other cancer biomarkers where picked up in theis analysis, a few that are not directly implicated in cancer or Erlotinib pharmacodynamics or pharmacokinetics could be **potential new biomarkers** that need to be explored further through more research.

# Future Directions

Future work should focus on expanding model architectures beyond DirectPred, Supervised VAE, and GNN by incorporating transformer-based models and self-supervised learning approaches.

Integrating more diverse omics data, such as methylation and proteomics, could provide deeper biological insights into drug responses. Advanced feature selection methods like SHAP, autoencoder-based dimensionality reduction, and graph-based embeddings could improve model interpretability and performance. Optimizing hyperparameters through Bayesian methods and extending HPO iterations could further enhance predictive accuracy. Testing models across multiple datasets, such as PRISM or DepMap, would ensure better generalizability and external validation.

Improving model interpretability with techniques like Integrated Gradients, LIME, and attention mechanisms is crucial for clinical applications. Parallelization using high-performance computing, GPU acceleration, and containerization will help scale computational workflows efficiently. Expanding the scope to drug combination therapy predictions by modeling synergistic effects and using reinforcement learning could provide valuable insights for personalized treatments. Refining these models with more comprehensive data, better computational strategies, and enhanced interpretability will strengthen their applicability in precision oncology.

# References

1. Akalin, A., & Uyar, B. (Retieved 2025). Flexynesis: Getting Started. MDC Berlin - Computational Biology. Retrieved from https://bimsbstatic.mdc-berlin.de/akalin/buyar/flexynesis/site/getting_started/#compiling-notebooks.
2. Broad Institute. (Retieved 2025). Cancer Cell Line Encyclopedia (CCLE). Retrieved from https://sites.broadinstitute.org/ccle/.
3. Genomics of Drug Sensitivity in Cancer (GDSC). (Retieved 2025). GDSC Database. Retrieved from https://www.cancerrxgene.org/.
4. Serizawa, M., Takahashi, T., Yamamoto, N., & Koh, Y. (2013). Genomic aberrations associated with erlotinib resistance in non-small cell lung cancer cells. Anticancer Research, 33(12), 5223-5233.
5. Brugger, W., Triller, N., Blasinska-Morawiec, M., Curescu, S., Sakalauskas, R., Manikhas, G. M., Mazieres, J., Whittom, R., Ward, C., Mayne, K., Trunzer, K., & Cappuzzo, F. (2011). Prospective molecular marker analyses of EGFR and KRAS from a randomized, placebo-controlled study of erlotinib maintenance therapy in advanced non–small-cell lung cancer. Journal of Clinical Oncology, 29(31), 4113-4120. https://doi.org/10.1200/JCO.2010.33.1601