Machine Learning-Based Multi-Omics Integration for Survival Prediction and Biomarker Discovery in Lung Adenocarcinoma (LUAD)

Project work report
Submitted
To

Visvesvaraya Technological University, Belgaum

In partial fulfillment for the award of

BACHELOR OF ENGINEERING
in
Biotechnology

By

Abhishek Manoj (1MS22BT001)
D Vaibhav (1MS22BT010)
Hridya Tessa (1MS22BT019)
Prajwal Kumar(1MS23BT400)

Guide
Dr. Lokesh. K. N
Assistant Professor
Department of Biotechnology
RIT, Bangalore

Department of Biotechnology
Ramaiah Institute of Technology
Bangalore-560 054

RAMAIAH Institute of Technology

DEPARTMENT OF BIOTECHNOLOGY

CERTIFICATE

Certified that the project work entitled **"Machine Learning-Based Multi-Omics Integration for Survival Prediction and Biomarker Discovery in Lung Adenocarcinoma (LUAD)"** Carried out by Abhishek Manoj, D.Vaibhav, Hridya Tessa, Prajwal Kumar bearing USN 1MS22BT001, 1MS22BT010, 1MS22BT019, 1MS23BT400 respectively are bonafide students of Department of Biotechnology, Ramaiah Institute of Technology, Bangalore, in partial fulfillment for the award of Bachelor of Engineering in Biotechnology of the Visvesvaraya Technological University, Belgaum during the year 2025. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

…… …….

Signature of the Guide (Internal) Signature of the HOD

……. ……..

Signature of the external examiner Signature of the Principal

RAMAIAH
Institute of Technology

Department of Biotechnology

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted by any other student/person or at any other institutions or for any other purpose.

Signature:

……

Name: Abhishek Manoj (1MS22BT001)

Signature:

……

Name: D Vaibhav (1MS22BT010)

Signature:

……

Name: Hridya Tessa (1MS22BT019)

Signature:

……

Name: Prajwal Kumar (1MS23BT400)

RAMAIAH
Institute of Technology

Department of Biotechnology

APPROVAL FOR SUBMISSION

I certify that this project report entitled**" Machine Learning-Based Multi-Omics Integration for Survival Prediction and Biomarker Discovery in Lung Adenocarcinoma (LUAD)"** was prepared by Abhishek Manoj, D Vaibhav, Hridya Tessa, Prajwal bonafide students of Department of Biotechnology, Ramaiah Institute of Technology, Bangalore in partial fulfillment of the requirements for the award of Bachelor of Engineering (B.E - Biotechnology) of the Visvesvaraya Technological University, Belgaum during the year 2022-2026. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements with respect to the Project Work prescribed for the degree.

.....

Signature

Guide (Internal)

Dr. Lokesh.K. N
Assistant Professor
Department of Biotechnology
RIT, Bangalore

Acknowledgements

Any achievement, be it scholastic or otherwise, does not depend solely on the individual efforts but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. A number of Personalities have helped us in carrying out this project. We would like to take this opportunity to thank them all.

We thank Dr. Chandraprabha M N, Head of the Biotechnology Department, RIT, for the encouragement in helping us pursue our mini project work.

We are highly indebted to Dr. Lokesh K N (internal guide), Assistant Professor, Department of Biotechnology RIT, for being kind, for his guidance, constant supervision, approachable, and for constant support all along till the completion, as well as for providing necessary information regarding the project work and for guiding us in carrying out the project. His motivation, valuable advice, and suggestions for corrections, modifications, and further refinements and improvements enhanced our perception of this project work.

We would like to express our gratitude towards our project coordinators: Dr. Abhijeet S R, Dr. Ravikumar Y S & the teaching and non- teaching staff of RIT for their kind co-operation and encouragement helping us in carrying out of the project work.

We would like to extend our gratitude to Dr. N V R Naidu, Principal, Ramaiah Institute of Technology, Bangalore, for providing us with the opportunities and resources that made this project possible

Table of Contents

List of Figures

List of Tables

Abstract

Lung Adenocarcinoma (LUAD) remains a leading cause of cancer-related mortality, highlighting the need for accurate survival prediction and identification of clinically relevant biomarkers. This mini project integrates multi-omics data, specifically RNA-seq expression profiles, mutation data, and clinical information from The Cancer Genome Atlas (TCGA) to develop a machine learning pipeline for predicting patient survival in LUAD.

After extensive preprocessing, including data cleaning, transposition, and merging, patients were labeled based on survival status. Machine learning models, including Random Forest (RF), Support Vector Machine (SVM), and a Voting Classifier, were trained and evaluated. The Random Forest model achieved the best accuracy (~66%), with both ROC-AUC and confusion matrices indicating moderate predictive capability. Feature importance scores extracted from RF highlighted key genes most associated with patient outcomes.

Box plots and a heatmap illustrated gene expression differences between survival groups. Functional enrichment using g:Profiler identified pathways such as immune regulation and apoptosis in survival-associated genes, and cell cycle and proliferation pathways in death-associated genes. Kaplan-Meier survival analysis further validated several genes, confirming their prognostic potential. Notably, genes like CTLA4 and BMP5 showed significant correlation with improved survival. For the deceased gene set, TLE1 and NEIL3 have strong and well-documented links to tumorigenesis.

This study not only demonstrates the power of machine learning in survival prediction using multi-omics data but also contributes potential prognostic biomarkers for further exploration in LUAD research.

1.Introduction

Lung Adenocarcinoma

Lung adenocarcinoma (LUAD) is the most prevalent histological subtype of non-small cell lung cancer (NSCLC), representing approximately 40% of all lung cancer cases worldwide1. Despite significant advances in diagnostic techniques and therapeutic interventions—including chemotherapy, immunotherapy, and targeted therapies—LUAD continues to exhibit poor long-term survival outcomes, with five-year survival rates remaining unacceptably low. The molecular heterogeneity of LUAD complicates effective prognostic assessment and limits the success of one-size-fits-all treatment strategies. This heterogeneity arises from diverse genetic mutations, epigenetic modifications, and altered gene expression profiles, which collectively challenge the identification of reliable biomarkers and the selection of optimal therapeutic regimens.

Traditional prognostic methods that rely on single-omics data—such as gene expression or mutation analysis alone—often fail to capture the full spectrum of biological complexity inherent in LUAD. In contrast, multi-omics integration, which combines data from multiple layers including transcriptomics, genomics, epigenomics, and proteomics, offers a more comprehensive understanding of tumor biology (Zhang, W., Zhao, L., Zheng, T. et al., 2024). Multi-omics approaches reveal the interplay between different molecular mechanisms and their collective impact on disease progression and patient outcomes. However, the high dimensionality and heterogeneity of multi-omics datasets pose significant analytical challenges, necessitating the use of advanced computational tools and machine learning algorithms.

Machine Learning and Its Bioinformatic Applications

Machine learning (ML) has emerged as a transformative force in bioinformatics, enabling the analysis of complex, high-dimensional biological datasets that are characteristic of cancer research. ML algorithms are capable of uncovering hidden patterns and relationships within multi-omics data that are often missed by traditional statistical methods (Haoyu Yang, Zheng An et al., 2018). The ability of ML to learn from data and make predictions without being explicitly programmed makes it particularly well-suited for addressing the challenges posed by cancer biology.

Key applications of machine learning in bioinformatics include sequence analysis, protein structure prediction, gene expression data analysis, classification and clustering of biological samples, and the prediction of biological networks (Haoyu Yang, Zheng An et al., 2018). In sequence analysis, ML algorithms help analyze DNA, RNA, and protein sequences to detect motifs, classify sequences, and predict functional sites. This enables a deeper understanding of gene regulation, the impact of mutations, and evolutionary relationships. In protein structure prediction, ML models assist in predicting the three-dimensional structure of proteins from amino acid sequences, which is crucial for understanding protein function and for drug design. Gene expression data analysis is another area where ML techniques have made significant contributions. By analyzing large-scale gene expression profiles, ML algorithms can identify biomarkers, classify diseases, and elucidate the mechanisms of gene regulation. Classification and clustering algorithms are widely used to categorize biological samples—such as distinguishing between cancerous and normal tissues—and to group similar data points, such as genes or proteins with comparable expression patterns. These capabilities are essential for diagnosis, prognosis, and the development of personalized medicine strategies.

Furthermore, ML plays a critical role in predicting biological networks, such as protein-protein interaction networks, gene regulatory networks, and metabolic pathways. By modeling these complex systems, ML enables researchers to reveal the underlying mechanisms of disease and to identify potential therapeutic targets (Haoyu Yang, Zheng An et al., 2018). The integration of ML with multi-omics data has the potential to revolutionize cancer research by providing a more holistic understanding of tumor biology and by enabling the discovery of novel biomarkers and therapeutic strategies.

Random Forest

Random Forest (RF) is a powerful ensemble machine learning method that constructs multiple decision trees during training and outputs the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees (Vladimir Svetnik et al., 2003). By aggregating the predictions of numerous trees, RF reduces the risk of overfitting and is robust to noise and outliers, making it particularly well-suited for analyzing complex and heterogeneous datasets such as those encountered in multi-omics studies.

The key advantages of Random Forest include its ability to handle high-dimensional data, its resistance to overfitting, and its capacity to provide estimates of feature importance. Feature importance scores generated by RF can highlight the most relevant genes or molecular features associated with patient outcomes, thereby aiding in the identification of potential biomarkers.

The robustness and accuracy of RF have made it a popular choice for survival prediction and biomarker discovery in cancer research (Vladimir Svetnik et al., 2003). In the context of LUAD, RF has been successfully applied to integrate multi-omics data and to identify genes and pathways that are most predictive of patient survival.

Support Vector Machine (SVM)

Support Vector Machine (SVM) is another widely used supervised learning algorithm, particularly effective for classification tasks. SVM seeks to find the optimal hyperplane that maximizes the margin between different classes in the feature space (Theodoros Evgeniou and Massimiliano Pontil, 2001). Its ability to handle high-dimensional data and its robustness to overfitting make SVM a popular choice for bioinformatics applications, especially when sample sizes are limited.

A distinctive advantage of SVM is that it relies on a subset of training samples called support vectors, which represent the critical elements defining the classification boundary. This property allows SVM to achieve strong generalization performance even with relatively small datasets (Jair Cervantes et al., 2020). SVM has been widely used in cancer research for tasks such as disease classification, biomarker identification, and survival prediction. In the context of LUAD, SVM has demonstrated the ability to effectively classify patients based on their molecular profiles and to identify genes associated with clinical outcomes.

Voting Classifier

A Voting Classifier is an ensemble method that combines the predictions of multiple machine learning models—such as Random Forest and SVM—to improve overall predictive performance (Zhou, Z.-H., 2009). By leveraging the strengths of each individual model, the Voting Classifier can enhance the robustness and accuracy of predictions, which is particularly important for complex biomedical datasets where no single model may capture all relevant patterns.

The use of ensemble methods like the Voting Classifier is especially valuable in the context of multi-omics data, where the integration of diverse data types and the identification of complex relationships between molecular features and clinical outcomes are critical. Ensemble methods help to reduce individual model biases and to better capture the intricate biological signals associated with patient survival and disease progression (Zhou, Z.-H., 2009).

Integration of Multi-Omics Data and Machine Learning in LUAD

The integration of multi-omics data with machine learning algorithms has emerged as a promising approach for improving survival prediction and biomarker discovery in LUAD. Recent studies have demonstrated the utility of combining transcriptomic, genomic, and clinical data to identify robust gene signatures that stratify patients into high- and low-risk groups (Zhang et al., 2023). For example, machine learning-based feature selection methods—such as LASSO regression and SVM-recursive feature elimination (SVM-RFE)—have been used to identify immune-related gene signatures that are predictive of patient survival and response to immunotherapy (Zhang et al., 2023).

These approaches have revealed that high-risk patients, as defined by specific gene signatures, exhibit significantly poorer overall survival and altered immune microenvironment profiles. Moreover, the identified gene signatures have been shown to have predictive power for response to immune checkpoint inhibitors, suggesting their potential utility in guiding personalized treatment decisions (Zhang et al., 2023). The validation of these models in independent patient cohorts has further confirmed their robustness and generalizability, highlighting the translational potential of machine learning-based prognostic models in LUAD.

In summary, the integration of multi-omics data with advanced machine learning algorithms offers a powerful framework for understanding the molecular basis of LUAD, predicting patient outcomes, and identifying novel biomarkers for personalized medicine. The continued development and refinement of these approaches hold great promise for improving the prognosis and treatment of lung adenocarcinoma.

2. Review of Literature

2.1 Identification of novel gene signature for lung adenocarcinoma by machine learning to predict immunotherapy and prognosis

In this article, the researchers aimed to identify a novel gene signature for lung adenocarcinoma (LUAD) using machine learning approaches to improve prognosis prediction and assess immunotherapy response. The primary motivation was to enhance personalized treatment strategies by integrating high-throughput gene expression data with computational algorithms to discover robust biomarkers related to tumor immunity and patient survival.

The theoretical framework of this study is based on the integration of bioinformatics and machine learning tools to process and analyze complex transcriptomic data. The goal was to extract meaningful patterns from large-scale datasets to identify gene signatures associated with immune activity and clinical outcomes.

The study used gene expression profiles from The Cancer Genome Atlas (TCGA) and Gene Expression Omnibus (GEO) databases. A set of immune-related genes was first curated, followed by feature selection using LASSO (Least Absolute Shrinkage and Selection Operator) regression and support vector machine-recursive feature elimination (SVM-RFE). These methods helped narrow down candidate genes. A multivariate Cox regression analysis was then applied to construct a prognostic model.

The results and findings of the article are as follows:

- A novel 10-gene signature was identified that effectively stratified LUAD patients into high- and low-risk groups. The high-risk group showed significantly poorer overall survival.

- This gene signature was found to be closely associated with the tumor immune microenvironment, particularly the infiltration of immune cells like macrophages and T cells. This was assessed using CIBERSORT and ESTIMATE algorithms.

- The model also demonstrated predictive power for response to immune checkpoint inhibitors (ICIs), suggesting its utility in guiding immunotherapy decisions.

- Validation in independent cohorts confirmed the robustness and generalizability of the gene signature across different datasets and populations.

This research successfully established a machine learning-based prognostic model for LUAD by identifying a novel immune-related gene signature. The study demonstrates how bioinformatics and computational techniques can provide valuable insights into tumor immunology and clinical outcomes, offering potential for improved prognostic tools and personalized treatment strategies in lung cancer management. **(Zhang et al., 2023)**

2.2 Bioinformatics analysis of an immunotherapy responsiveness-related gene signature in predicting lung adenocarcinoma prognosis

In this article, the researchers explored the development of an immunotherapy responsiveness-related gene signature for predicting prognosis in lung adenocarcinoma (LUAD). The primary motivation was to address the clinical variability in patient responses to immune checkpoint inhibitors (ICIs) by identifying molecular markers that could predict both prognosis and immunotherapy efficacy.

The study is grounded in the principles of precision oncology and bioinformatics, utilizing transcriptomic data to uncover gene expression patterns linked to immune response and patient survival. The authors aimed to integrate immune-related genomic features into a predictive model to improve therapeutic decision-making in LUAD.

The researchers collected LUAD gene expression and clinical data from The Cancer Genome Atlas (TCGA) and Gene Expression Omnibus (GEO). Differentially expressed genes (DEGs) between immunotherapy responders and non-responders were identified. These DEGs were then filtered using univariate Cox regression, LASSO regression, and multivariate Cox analysis to develop a prognostic gene signature.

The results and findings of the article are as follows:

- A six-gene signature was constructed that effectively stratified patients into high-risk and low-risk groups, with significant differences in overall survival.

- The model demonstrated strong prognostic value, as validated by Kaplan-Meier analysis and ROC curves, and was robust across both TCGA and external GEO datasets.

- The low-risk group was associated with a more active immune microenvironment, characterized by higher immune scores and greater infiltration of cytotoxic T cells, indicating potential for better immunotherapy response.

- Gene set enrichment analysis (GSEA) revealed that immune-related pathways, such as interferon gamma response and antigen processing, were significantly enriched in the low-risk group.

This study successfully established a bioinformatics-driven, immunotherapy-linked prognostic model for LUAD. The gene signature not only predicts overall survival but also provides insights into the tumor immune landscape, aiding in the identification of patients who are more likely to benefit from immune checkpoint therapy. The findings support the integration of transcriptomic biomarkers in personalizing cancer immunotherapy. **(Wang et al., 2024)**

2.3 Survival Analysis of Cancer Patients in North Eastern Nigeria from 2004 – 2017 – A Kaplan-Meier Method

In this article, the researchers performed a retrospective survival analysis of cancer patients treated in North-Eastern Nigeria over a 14-year period (2004–2017). The primary objective was to determine overall survival trends and the impact of different cancer types and patient demographics on survival outcomes, using the Kaplan-Meier statistical method. The study addresses the need for local cancer survival data in Nigeria to inform healthcare planning and improve cancer management strategies in resource-limited settings.

The theoretical foundation of the work is based on survival analysis, particularly the Kaplan-Meier method, which is widely used to estimate survival probabilities over time. The study aimed to identify which cancers had the poorest outcomes and how factors such as gender, cancer type, and follow-up duration affected survival rates.

Data was collected from patient records at the Federal Teaching Hospital Gombe and included 1,250 cancer patients with confirmed diagnoses. The survival time was calculated from the time of diagnosis to either death or last follow-up. Censoring was applied for patients lost to follow-up or still alive at the end of the study period.

The results and findings of the article are as follows:

- The **overall median survival time** for cancer patients in the cohort was approximately 13 months, with significant variation between cancer types.

- **Breast and cervical cancers** were the most commonly diagnosed, but breast cancer patients had better survival outcomes compared to cervical and gastrointestinal cancer patients.

- The **worst survival** was observed among patients with **gastrointestinal and liver cancers**, while **prostate and hematological cancers** had relatively better survival trends.

- Gender differences were observed, with female patients generally having longer survival times, possibly due to earlier detection and treatment of cancers like breast and cervical cancer.

- The study also highlighted challenges such as late presentation, lack of access to cancer care, and poor follow-up rates, which negatively impacted survival.

This research provides critical baseline data on cancer survival trends in a specific Nigerian population and emphasizes the urgent need for improved cancer diagnosis, treatment accessibility, and follow-up care. The use of the Kaplan-Meier method effectively demonstrated disparities in cancer outcomes, supporting future public health interventions aimed at cancer control and policy-making in low-resource regions. **(Ahmed et al., 2019)**

2.4 Integrating Omics Data and AI for Cancer Diagnosis and Prognosis

In this article, the authors explore how the integration of multi-omics data with artificial intelligence (AI) can significantly enhance cancer diagnosis and prognosis. The central

motivation behind this study is the growing need to utilize complex, high-dimensional biological data—such as genomics, transcriptomics, proteomics, and metabolomics—to develop more accurate and personalized approaches for cancer detection, subtyping, and outcome prediction.

The theoretical foundation lies in systems biology and machine learning, where multi-omics integration enables a holistic understanding of cancer mechanisms, and AI facilitates pattern recognition and predictive modeling. By combining these two domains, the study highlights how predictive accuracy and clinical relevance can be improved.

The review extensively summarizes recent advancements in omics technologies and machine learning methods applied to cancer datasets. It discusses a variety of algorithms, including Random Forest, Support Vector Machine (SVM), and deep learning frameworks used to process large omics datasets. The authors also emphasize the importance of feature selection, data normalization, and model validation techniques to avoid overfitting and ensure generalizability.

The results and findings of the article are as follows:

- Integrating **multi-omics data** leads to better prediction of cancer subtypes, patient stratification, and survival outcomes compared to single-omics approaches.

- AI models, particularly **deep learning architectures**, are capable of capturing nonlinear relationships in multi-omics data, leading to high diagnostic and prognostic accuracy across several cancer types.

- The study identifies significant challenges, including data heterogeneity, limited sample sizes, and the need for robust interpretability of AI models in clinical settings.

- Successful case studies in breast cancer, lung cancer, and glioblastoma are presented, demonstrating the clinical potential of AI-driven multi-omics integration.

This article underscores the transformative role of integrating omics data with AI in precision oncology. By leveraging the strengths of both biological data richness and computational intelligence, the approach offers promising tools for early detection, prognosis, and therapy

response prediction in cancer. The study concludes that while challenges remain, continued advancements in data standardization, model transparency, and cross-disciplinary collaboration are key to clinical translation. **(Stavrou et al., 2024)**

2.5 Comparisons of Forecasting for Survival Outcome for Head and Neck Squamous Cell Carcinoma by Using Machine Learning Models Based on Multi-omics

In this article, the researchers aimed to evaluate the predictive performance of multiple machine learning models in forecasting the survival outcomes of patients with head and neck squamous cell carcinoma (HNSCC), using integrated multi-omics data. The motivation behind the study lies in the complexity and heterogeneity of HNSCC, which necessitates comprehensive approaches for better prognosis and treatment planning.

The theoretical foundation of this work revolves around the use of multi-omics integration (genomics, transcriptomics, and epigenomics) and machine learning algorithms to build survival prediction models. The authors hypothesize that leveraging a broader molecular dataset through AI-based tools can improve survival forecasting accuracy in comparison to traditional statistical methods.

The study utilized publicly available datasets from The Cancer Genome Atlas (TCGA) for HNSCC patients. After preprocessing, multi-omics features were extracted and used to train various machine learning models, including Random Forest (RF), Support Vector Machine (SVM), and Cox proportional hazards models. These models were evaluated using cross-validation and multiple performance metrics.

The results and findings of the article are as follows:

- The Random Forest model outperformed other methods in terms of concordance index (C-index) and prediction accuracy, suggesting its robustness in handling multi-omics data for survival analysis.

- The integration of multi-omics features significantly improved model performance compared to models trained on single-omics data alone.

- Important features contributing to prediction included both gene expression and DNA methylation markers, underlining the complementary nature of different omics layers.

- The study also highlighted the potential of machine learning to stratify patients into high- and low-risk groups, which could aid in personalized treatment planning.

This research successfully demonstrates the power of machine learning and multi-omics integration for survival prediction in HNSCC. It provides evidence that combining diverse molecular data sources can enhance model performance and reliability, supporting the development of more effective prognostic tools in oncology. The study contributes to the growing body of literature advocating for AI-driven precision medicine. **(Chen et al., 2022)**

2.6 Bioinformatics and machine learning driven key genes screening for hepatocellular carcinoma

In this article, the researchers aimed to identify potential key genes involved in hepatocellular carcinoma (HCC) through an integrated approach involving bioinformatics analysis and machine learning algorithms. The main motivation was to uncover novel diagnostic and prognostic biomarkers that could enhance the early detection and therapeutic targeting of HCC, a highly malignant liver cancer with poor prognosis and high mortality.

The study builds on the foundations of transcriptomics and computational biology, using high-throughput gene expression data to identify differentially expressed genes (DEGs) and applying machine learning models to prioritize potential biomarkers. The integration of statistical filtering with algorithmic feature selection allows for more reliable identification of key regulatory genes in cancer progression.

The researchers obtained gene expression data from The Cancer Genome Atlas (TCGA) and Gene Expression Omnibus (GEO). DEGs between HCC and normal tissues were identified through statistical analysis. These DEGs were then subjected to machine learning-based selection using LASSO (Least Absolute Shrinkage and Selection Operator) and Random Forest (RF) to pinpoint the most influential genes.

The results and findings of the article are as follows:

- A total of 12 candidate genes were identified as significantly associated with HCC, based on the overlap between DEGs and genes selected by both LASSO and RF models.

- Among these, CYP26B1, TSPAN8, and KIF20A were highlighted as potential key genes due to their consistent overexpression in tumor samples and correlation with poor overall survival.

- Functional enrichment analysis (GO and KEGG) revealed that these genes were involved in key cancer-related pathways such as cell cycle regulation, retinoic acid metabolism, and immune response.

- The prognostic value of the selected genes was validated using Kaplan-Meier survival analysis, confirming their clinical relevance.

- The study also developed a risk score model based on the expression of the top genes, which effectively stratified HCC patients into high- and low-risk groups.

This research successfully combines bioinformatics and machine learning to screen for potential biomarkers in hepatocellular carcinoma. The identified key genes may serve as valuable tools for early diagnosis, prognosis prediction, and therapeutic targeting. The study demonstrates the utility of AI-based analytical pipelines in improving cancer biomarker discovery. **(Guo et al., 2023)**

3. Objectives

●To acquire and preprocess a multi-omics database from TCGA-LUAD linked-omics website.

●To implement and evaluate supervised machine learning algorithms such as Random Forest, SVM, and Voting Classifier (Hybrid).

●To identify key prognostic genes from both Survival-associated and Death-associated gene set.

●To perform functional enrichment analysis of prognostic genes using GO and pathway analysis.

●To conduct Kaplan-Meier survival analysis

Tools and Packages

1. Programming language: Python
2. Data Handling & Processing: Pandas, NumPy, os, csv
3. Machine Learning & Feature Selection: scikit-learn(sklearn)
4. Data Visualization: matplotlib, seaborn, plotly
5. Survival Analysis: lifelines
6. Functional Enrichment: g: Profiler
7. Data Source: LinkedOmics (https://www.linkedomics.org/data_download/TCGA-LUAD/)

4. Methodology

4.1 Data Acquisition and Processing

Acquiring RNA-seq expression(CCT format), mutation (CBT format), and clinical (TSV format) data from **The Cancer Genome Atlas (TCGA)**. (https://linkedomics.org/data_download/TCGA-LUAD/)Transposing and cleaning each dataset to ensure samples were aligned consistently using `attrib_name` as the common identifier.Handling missing values and ensuring compatibility of gene symbols and sample labels across datasets.Merging the mutation and RNA-seq files on the `attrib_name` column, followed by joining with transposed clinical data to create a comprehensive multi-omics dataset.Labeling each sample with a binary survival status (`status`: 0 = alive, 1 = deceased) for supervised classification. (As shown in Fig 4.1.1 & Fig 4.1.2)

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

# Defining file paths
mutation_path = r"C:\Users\ME\Desktop\min_proj\Datasets\mutation.cbt"
rnaseq_path = r"C:\Users\ME\Desktop\min_proj\Datasets\RNA.cct"
output_path = "../outputs/processed_luad.csv"

#Loading Mutation Data
print(" Loading mutation data...")
mutation_df = pd.read_csv(mutation_path, sep='\t')
print(f" Mutation data shape: {mutation_df.shape}")

#Loading RNA Data
print(" Loading RNA-seq data...")
rnaseq_df = pd.read_csv(rnaseq_path, sep='\t')
print(f" RNA-seq data shape: {rnaseq_df.shape}")

# Merging on gene column
print(" Merging on 'gene' column...")
merged_df = pd.merge(mutation_df, rnaseq_df, on='gene', how='inner')
print(f" Merged dataset shape: {merged_df.shape}")

# Filling any missing column
merged_df.fillna(0, inplace=True)

# Normalising feature column
print(" Normalizing features...")
gene_col = merged_df['gene']
features = merged_df.drop('gene', axis=1)

scaler = StandardScaler()
features_scaled = scaler.fit_transform(features)

# Reassembling Dataframe
processed_df = pd.DataFrame(features_scaled, columns=features.columns)
processed_df.insert(0, 'gene', gene_col)

# Save to output directory
print(" Saving preprocessed data to CSV...")
processed_df.to_csv(output_path, index=False)
print(f" Done! File saved at: {output_path}")
```

Fig 4.1.1: Converting the dataset to .csv file

```python
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns

#Setting Data Directory
data_dir = "../output"
mutation_path = os.path.join(data_dir, "mutation_cleaned.csv")
rnaseq_path = os.path.join(data_dir, "rnaseq_cleaned.csv")
clinical_path = os.path.join(data_dir, "clinical_cleaned.csv")

# Loading and transposing mutation & rnaseq
mutation_df = pd.read_csv(mutation_path, index_col=0).transpose()
rnaseq_df = pd.read_csv(rnaseq_path, index_col=0).transpose()

# Resetting index to make 'attrib_name' a column
mutation_df.reset_index(inplace=True)
rnaseq_df.reset_index(inplace=True)

# Renaming index column to match across datasets
mutation_df.rename(columns={"index": "attrib_name"}, inplace=True)
rnaseq_df.rename(columns={"index": "attrib_name"}, inplace=True)

# Merging omics data on sample ID
merged_df = pd.merge(mutation_df, rnaseq_df, on="attrib_name", how="inner")

# Loading and transposing clinical data
clinical_df = pd.read_csv(clinical_path, index_col=0).transpose().reset_index()
clinical_df.rename(columns={"index": "attrib_name"}, inplace=True)

# Merging clinical status
if "status" not in clinical_df.columns:
    raise ValueError("Status column not found in transposed clinical data!")

merged_df = merged_df.merge(clinical_df[["attrib_name", "status"]], on="attrib_name", how="inner")

# Dropping rows with missing labels
merged_df = merged_df.dropna(subset=["status"])

# Saving merged data
output_path = os.path.join(data_dir, "merged_labeled_data.csv")
merged_df.to_csv(output_path, index=False)
print(f"Saved merged data to {output_path}")

# Plotting label distribution
plt.figure(figsize=(6, 4))
sns.countplot(x="status", data=merged_df)
plt.title("Survival Label Distribution (status)")
plt.xlabel("Status (0 = Alive, 1 = Deceased)")
plt.ylabel("Count")
plt.tight_layout()
plot_path = os.path.join(data_dir, "label_distribution.png")
plt.savefig(plot_path)
print(f"Saved label distribution plot to {plot_path}")
plt.show()
```

Fig 4.1.2: Merging the converted .csv files to make it ready for model training

4.2. Feature Extraction

Training an initial **Random Forest (RF)** model to obtain feature importance scores. Selecting top features (genes) based on importance for downstream visualization and analysis. Developing a PCA plot analysis to make sure variance is observed and principal components 1 and 2 are obtained. Generating box plots comparing gene expression between alive and deceased groups. Constructing a clustered heatmap for the top 100 most informative genes to visualize expression trends across patients. (As shown in Fig 4.2.1, Fig 4.2.2 & Fig 4.2.3)

```python
import os
import sys
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Increase recursion depth for dendrograms
sys.setrecursionlimit(10000)

# Paths
data_dir = "../output"
output_dir = "../output"

# Load data
print("Loading merged labeled data...")
data = pd.read_csv(os.path.join(data_dir, "merged_labeled_data.csv"))
X = data.drop(columns=["attrib_name", "status"])
y = data["status"]

# Select top 100 most variable features
top_100 = X.var().sort_values(ascending=False).head(100).index
X_top = X[top_100].copy()
X_top["status"] = y

# Sort samples by class
X_top = X_top.sort_values("status")

# Drop labels for heatmap
X_top_plot = X_top.drop(columns=["status"])

# Generate clustered heatmap (uses SciPy by default)
print("Generating clustered heatmap using SciPy...")
sns.set(style="white")
clustermap = sns.clustermap(
    X_top_plot,
    cmap="vlag",
    figsize=(14, 10),
    yticklabels=False,
    xticklabels=True,
    row_cluster=True,
    col_cluster=True,   # Enable column clustering
    metric="euclidean", # Distance metric
    method="average"    # Linkage method
)

clustermap.fig.suptitle("Clustered Heatmap of Top 100 Features", fontsize=16)
clustermap.ax_heatmap.set_xlabel("Gene Features")
clustermap.ax_heatmap.set_ylabel("Samples")

# Save the plot
heatmap_path = os.path.join(output_dir, "clustered_heatmap_top100_scipy.png")
clustermap.savefig(heatmap_path)
print(f"Heatmap saved to {heatmap_path}")
```

Fig 4.2.1: Code Block for clustered heatmap for top 100 genes

```python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (classification_report, confusion_matrix, accuracy_score,
                             roc_curve, auc)
import plotly.graph_objects as go

# Set paths
data_dir = "../output"
output_dir = "../output"

# Load data
print("Loading labeled dataset...")
data = pd.read_csv(os.path.join(data_dir, "merged_labeled_data.csv"))

# Drop non-feature columns and set up X and y
X = data.drop(columns=["attrib_name", "status"])
y = data["status"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Random Forest
print("Training Random Forest...")
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
y_score_rf = rf_model.predict_proba(X_test)[:, 1]

# Evaluate and print metrics
def evaluate_model(name, y_true, y_pred):
    acc = accuracy_score(y_true, y_pred)
    report = classification_report(y_true, y_pred, zero_division=0)
    print(f"\n{name} Accuracy: {acc:.4f}")
    print(f"Classification Report:\n{report}")

    # Save confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
    plt.title(f"{name} Confusion Matrix")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f"{name.lower()}_confusion_matrix.png"))
    plt.close()

    # Save report
    with open(os.path.join(output_dir, f"{name.lower()}_report.txt"), "w") as f:
        f.write(f"{name} Accuracy: {acc:.4f}\n\n")
        f.write(report)

evaluate_model("RandomForest", y_test, y_pred_rf)
# Feature importance plot
print("Plotting feature importances...")
importances = rf_model.feature_importances_
feature_names = X.columns
top_indices = np.argsort(importances)[-10:]
top_features = feature_names[top_indices]
top_importances = importances[top_indices]

plt.figure(figsize=(8, 6))
sns.barplot(x=top_importances, y=top_features, orient="h")
plt.title("Top 10 Important Features (Random Forest)")
plt.xlabel("Feature Importance")
plt.ylabel("Gene")
plt.tight_layout()
plt.savefig(os.path.join(output_dir, "feature_importance_rf.png"))
plt.close()

# Box plot for top features
print("Generating box plots for top features...")
for feature in top_features:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x=y, y=data[feature])
    plt.title(f"Expression of {feature} by Survival Status")
    plt.xlabel("Survival Status (0=Deceased, 1=Alive)")
    plt.ylabel("Expression Level")
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f"boxplot_{feature}.png"))
    plt.close()
```

Fig 4.2.2: Code Block to train the Random Forest model to obtain feature importance scores and produce box plots.

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Setting path
data_dir = "../output"
merged_file = os.path.join(data_dir, "merged_labeled_data.csv")
df = pd.read_csv(merged_file)

# Separating features and label
X = df.drop(columns=["attrib_name", "status"])
y = df["status"]

# Standardizing features
X_scaled = StandardScaler().fit_transform(X)

# Performing PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)

# Getting feature names
feature_names = X.columns

# Getting loadings for PC1 and PC2
loadings = pd.DataFrame(pca.components_.T,
                        columns=['PC1', 'PC2'],
                        index=feature_names)

top_pc1 = loadings["PC1"].abs().sort_values(ascending=False).head(10)
top_pc2 = loadings["PC2"].abs().sort_values(ascending=False).head(10)

# Plotting combined figure
fig, axes = plt.subplots(1, 3, figsize=(20, 6))

# PCA Scatter plot
scatter = axes[0].scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='coolwarm', alpha=0.6, edgecolor='k')
axes[0].set_title("PCA of Samples by Survival Status")
axes[0].set_xlabel("Principal Component 1")
axes[0].set_ylabel("Principal Component 2")
legend_labels = ["Alive (0)", "Deceased (1)"]
legend = axes[0].legend(handles=scatter.legend_elements()[0], labels=legend_labels, title="Status")

# Top features PC1
axes[1].barh(top_pc1.index[::-1], top_pc1.values[::-1], color='skyblue')
axes[1].set_title("Top 10 Features Contributing to PC1")
axes[1].set_xlabel("Absolute Contribution")
axes[1].set_ylabel("Feature")

# Top features PC2
axes[2].barh(top_pc2.index[::-1], top_pc2.values[::-1], color='salmon')
axes[2].set_title("Top 10 Features Contributing to PC2")
axes[2].set_xlabel("Absolute Contribution")
axes[2].set_ylabel("Feature")

plt.tight_layout()
combined_path = os.path.join(data_dir, "pca_combined_analysis.png")
plt.savefig(combined_path)
plt.close()

print(f"PCA combined figure saved to: {combined_path}")
```

Fig 4.2.3: Code block for PCA plot

4.3 Machine Learning

A. Performing an 80/20 **train-test split** using the preprocessed dataset.

Training three supervised classifiers:

Random Forest

Support Vector Machine (SVM)

Voting Classifier combining RF and SVM (soft voting)

B. Evaluating models based on:

Accuracy

Confusion Matrix

Classification Report

ROC-AUC Score

Plotting ROC curves for all models to visualize performance comparison. (As shown in Fig 4.3.1 & Fig 4.3.2)

```python
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import (classification_report, confusion_matrix, accuracy_score,
                             roc_curve, auc)
import plotly.graph_objects as go

# Set paths
data_dir = "../output"
output_dir = "../output"

# Load data
print("Loading labeled dataset...")
data = pd.read_csv(os.path.join(data_dir, "merged_labeled_data.csv"))

# Drop non-feature columns and set up X and y
X = data.drop(columns=["attrib_name", "status"])
y = data["status"]

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Random Forest
print("Training Random Forest...")
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
y_score_rf = rf_model.predict_proba(X_test)[:, 1]

# SVM
print("Training SVM...")
svm_model = SVC(probability=True, random_state=42)
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict(X_test)
y_score_svm = svm_model.predict_proba(X_test)[:, 1]

# Evaluate and print metrics
def evaluate_model(name, y_true, y_pred):
    acc = accuracy_score(y_true, y_pred)
    report = classification_report(y_true, y_pred, zero_division=0)
    print(f"\n{name} Accuracy: {acc:.4f}")
    print(f"Classification Report:\n{report}")

    # Save confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(5, 4))
    sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
    plt.title(f"{name} Confusion Matrix")
    plt.xlabel("Predicted")
    plt.ylabel("Actual")
    plt.tight_layout()
    plt.savefig(os.path.join(output_dir, f"{name.lower()}_confusion_matrix.png"))
    plt.close()

    # Save report
    with open(os.path.join(output_dir, f"{name.lower()}_report.txt"), "w") as f:
        f.write(f"{name} Accuracy: {acc:.4f}\n\n")
        f.write(report)
```

FIG 4.3.1: Code block for machine learning model training and report generation

```python
evaluate_model("RandomForest", y_test, y_pred_rf)
evaluate_model("SVM", y_test, y_pred_svm)

# Interactive ROC curve plot
print("Generating interactive ROC curve...")
fpr_rf, tpr_rf, _ = roc_curve(y_test, y_score_rf)
fpr_svm, tpr_svm, _ = roc_curve(y_test, y_score_svm)
auc_rf = auc(fpr_rf, tpr_rf)
auc_svm = auc(fpr_svm, tpr_svm)

fig = go.Figure()
fig.add_trace(go.Scatter(x=fpr_rf, y=tpr_rf, mode='lines', name=f"Random Forest (AUC = {auc_rf:.2f})"))
fig.add_trace(go.Scatter(x=fpr_svm, y=tpr_svm, mode='lines', name=f"SVM (AUC = {auc_svm:.2f})"))
fig.add_trace(go.Scatter(x=[0, 1], y=[0, 1], mode='lines', name="Random Chance", line=dict(dash='dash')))

fig.update_layout(
    title="Interactive ROC Curve",
    xaxis_title="False Positive Rate",
    yaxis_title="True Positive Rate",
    width=800,
    height=600,
    template="plotly_white"
)

fig.write_html(os.path.join(output_dir, "interactive_roc_curve.html"))
print("ROC curve saved as interactive HTML in output directory.")
```

Fig 4.3.2: Code block for ROC curve and model comparison.

4.4. Functional and Survival Study

A.Dividing the top genes into:

Survival-associated genes (positive prognosis)

Death-associated genes (poor prognosis)

B.Uploaded gene sets to **g: Profiler** for functional enrichment:

Explored GO terms, Reactome, KEGG pathways

Exported results as bar plots and tables

C. Conducted **Kaplan-Meier survival analysis**:

Used expression data and clinical metadata (overall survival time and status)

Stratified patients into high- and low-expression groups per gene

Plotted survival curves and calculated p-values to assess significance (As shown in Fig 4.4.1 & Fig 4.4.2)

```python
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from imblearn.over_sampling import SMOTE
import urllib.parse
import os

# === Step 1: Load data ===
data_dir = "../output"
save_dir = "../test"
df = pd.read_csv(os.path.join(data_dir, "merged_labeled_data.csv"))
X = df.drop(columns=["attrib_name", "status"])
y = df["status"]

# === Step 2: Scale & Balance ===
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42, stratify=y)
sm = SMOTE(random_state=42)
X_train_bal, y_train_bal = sm.fit_resample(X_train, y_train)

# === Step 3: Train Random Forest ===
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train_bal, y_train_bal)

# === Step 4: Extract Top N Genes ===
importances = rf.feature_importances_
feature_names = X.columns
top_n = 20
top_indices = np.argsort(importances)[-top_n:][::-1]
top_genes = feature_names[top_indices]

# === Step 5: Split Genes by Outcome ===
death_genes = []
survival_genes = []

for gene in top_genes:
    mean_dead = df[df["status"] == 1][gene].mean()
    mean_alive = df[df["status"] == 0][gene].mean()
    if mean_dead > mean_alive:
        death_genes.append(gene)
    else:
        survival_genes.append(gene)

# === Step 6: Save gene lists ===
pd.Series(death_genes).to_csv(os.path.join(save_dir, "death_genes_auto.csv"), index=False)
pd.Series(survival_genes).to_csv(os.path.join(save_dir, "survival_genes_auto.csv"), index=False)
```

FIG 4.4.1: Code block for dividing gene sets into survival-associated and death-associated genes.

```python
import os
import pandas as pd
import matplotlib.pyplot as plt
from lifelines import KaplanMeierFitter
from lifelines.statistics import logrank_test

# Loading data
clinical = pd.read_csv("../output/clinical_cleaned.csv", header=None)
rnaseq = pd.read_csv("../output/rnaseq_cleaned.csv", header=None, low_memory=False)

# Transposing clinical and RNAseq dataframes (since rows are samples)
clinical = clinical.transpose()
rnaseq = rnaseq.transpose()

# Renaming columns (the first row of each dataframe should be the headers)
clinical.columns = clinical.iloc[0]
clinical = clinical.drop(0)  # Drop the first row which contains the column names
rnaseq.columns = rnaseq.iloc[0]
rnaseq = rnaseq.drop(0)  # Drop the first row which contains the column names

# Ensuring 'attrib_name' is present in both dataframes
if 'attrib_name' not in clinical.columns:
    raise KeyError("Clinical dataframe must contain 'attrib_name' column.")
if 'attrib_name' not in rnaseq.columns:
    raise KeyError("RNAseq dataframe must contain 'attrib_name' column.")

# Merging clinical and RNAseq data on 'attrib_name'
merged = pd.merge(clinical, rnaseq, on='attrib_name', how='inner')

# Making sure survival columns are numeric
merged['overall_survival'] = pd.to_numeric(merged['overall_survival'], errors='coerce')
merged['status'] = pd.to_numeric(merged['status'], errors='coerce')

# Dropping missing survival data
merged = merged.dropna(subset=['overall_survival', 'status'])

# Time and event columns
T = merged['overall_survival']
E = merged['status']

# Initialize Kaplan-Meier fitter
kmf = KaplanMeierFitter()

# Loop through each filtered gene in the CSV
filtered_genes = pd.read_csv("../test/survival_genes_auto.csv")

for gene in filtered_genes["Gene"]:  # Assuming column name is "Gene" in your CSV
    if gene not in merged.columns:
        print(f" skipping {gene} - not found in RNAseq data")
        continue

    # Ensure the gene expression column is numeric
    merged[gene] = pd.to_numeric(merged[gene], errors='coerce')
    merged = merged.dropna(subset=[gene])  # Drop rows with NaN values in the gene column

    # Group by median expression (high vs low)
    median_expr = merged[gene].median()
    merged['group'] = merged[gene].apply(lambda x: "High" if x > median_expr else "Low")

    # Plot KM curve
    plt.figure(figsize=(8, 6))

    for group in ['High', 'Low']:
        mask = merged['group'] == group
        kmf.fit(durations=T[mask], event_observed=E[mask], label=group)
        kmf.plot_survival_function(ci_show=False)

    plt.title(f"Survival Curve for {gene}")
    plt.xlabel("Days")
    plt.ylabel("Survival Probability")
    plt.grid(True)
    plt.legend(title="Expression Level")
    plt.tight_layout()

    # Save the plot
    plot_path = os.path.join(output_dir, f"{gene}_survival.png")
    plt.savefig(plot_path, dpi=300)
    plt.close()

    # Log-rank test
    mask_high = merged["group"] == "High"
    mask_low = merged["group"] == "Low"
    results = logrank_test(T[mask_high], T[mask_low], E[mask_high], E[mask_low])
    print(f"{gene}: log-rank p-value = {results.p_value:.4f}")

print("\n Survival analysis complete!")
print(f"Plots saved in: {output_dir}")
```

FIG 4.4.2: Code block for Kaplan Meier analysis plots.

5. Results and Discussion

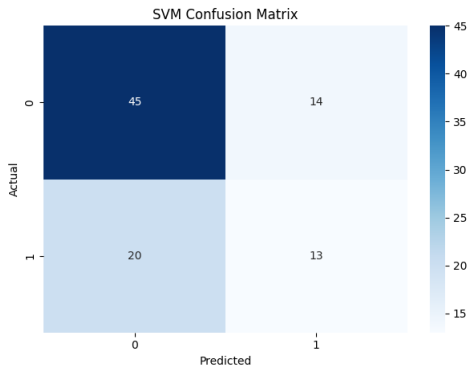

FIG 5.1: SVM Confusion matrix

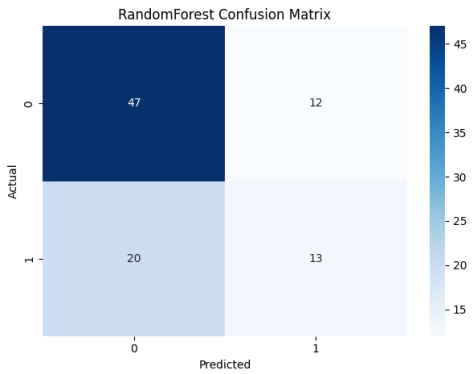

FIG 5.2: Random Forest Confusion matrix

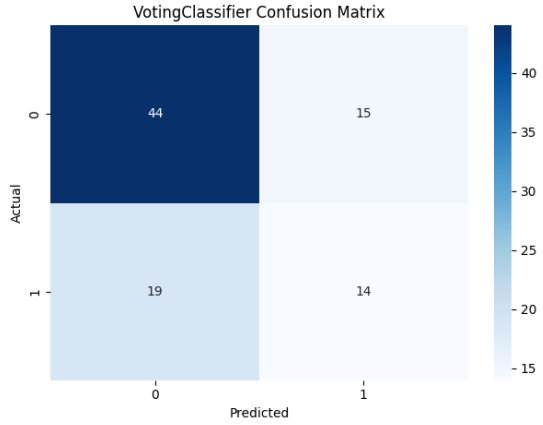

FIG 5.3 Voting Classifier Confusion matrix

A confusion matrix is a table that provides a detailed breakdown of a classification model's performance by comparing its predicted labels to the actual labels from the dataset. It is especially useful in binary classification tasks. The matrix is typically structured as a 2x2 table, where the rows represent the actual classes and the columns represent the predicted classes. The four cells in the matrix correspond to true positives (correctly predicted positive cases), true negatives (correctly predicted negative cases), false positives (incorrectly predicted positive cases, also known as Type I errors), and false negatives (incorrectly predicted negative cases, also known as Type II errors). By analyzing the distribution of values in these four categories, one can calculate important performance metrics such as accuracy, precision, recall (sensitivity), specificity, and the F1 score. Overall, the confusion matrix provides a more complete picture of a model's strengths and weaknesses than accuracy alone, helping identify

whether the model is biased toward certain classes or is making systematic types of errors.(As shown in Fig 5.1,5.2 & 5.3)

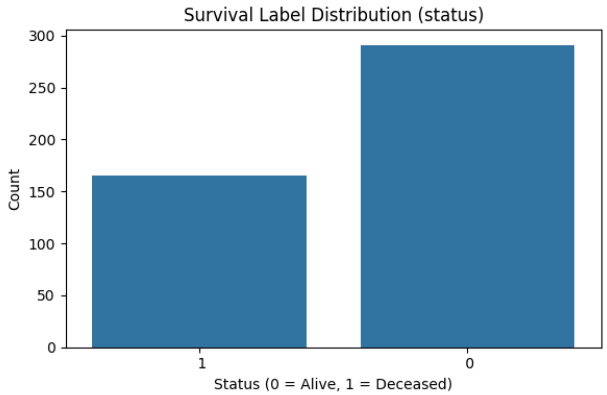

FIG 5.4: Survival Label Distribution

This bar chart illustrates the distribution of survival labels in the dataset, with the variable status representing whether an individual is alive or deceased. The x-axis indicates two categories: 0 (Alive) and 1 (Deceased), while the y-axis shows the corresponding count of individuals in each category. The chart reveals that a larger proportion of individuals are alive (approximately 290) compared to those who are deceased (around 170). This indicates an imbalanced distribution between the two classes, with more individuals surviving than not. (As shown in Fig 5.4)

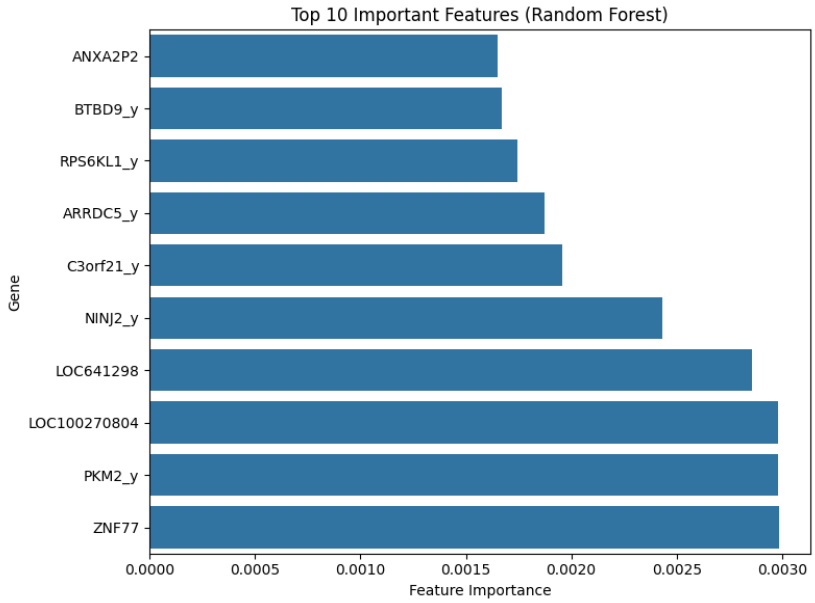

FIG 5.5 Top 10 Important Features

This bar plot highlights the top 10 gene features ranked by importance in a Random Forest model used to predict survival or disease status in LUAD (Lung Adenocarcinoma). ZNF77 emerges as the most influential gene, followed closely by PKM2_y, LOC100270804, and LOC641298. These genes exhibit the highest feature importance scores, suggesting they provide the most informative signals for the model's predictions. Other notable contributors include NINJ2_y, C3orf21_y, ARRDC5_y, and RPS6KL1_y, with slightly lower but still significant impact. BTBD9_y and ANXA2P2 round out the top 10. The inclusion of both well-characterized genes like PKM2_y, known for its role in cancer metabolism, and lesser-known or uncharacterized loci such as LOC100270804, underscores the model's ability to uncover both established and potentially novel biomarkers relevant to LUAD prognosis. (As shown in Fig 5.5)

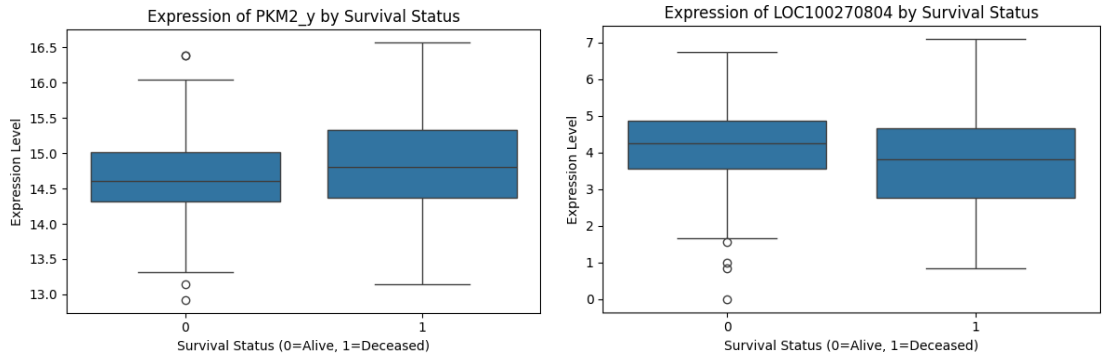

FIG 5.6: Expression of PKM2_y and LOC100270804 by Survival Status

The boxplot analysis reveals differential expression patterns of LOC100270804 and PKM2_y based on survival status in LUAD patients. LOC100270804 shows relatively similar median expression levels between the alive and deceased groups. However, the deceased group exhibits a slightly broader spread and higher upper range, which may indicate elevated expression in a subset of patients with poorer outcomes. The presence of multiple low-expression outliers in the alive group could reflect underlying biological variability or heterogeneity in gene regulation. PKM2_y, a gene well-known for its role in cancer metabolism and tumor progression, demonstrates slightly higher median expression in the deceased group compared to survivors. This suggests a potential negative prognostic role, with elevated PKM2_y expression being associated with worse survival. (As shown in Fig 5.6)

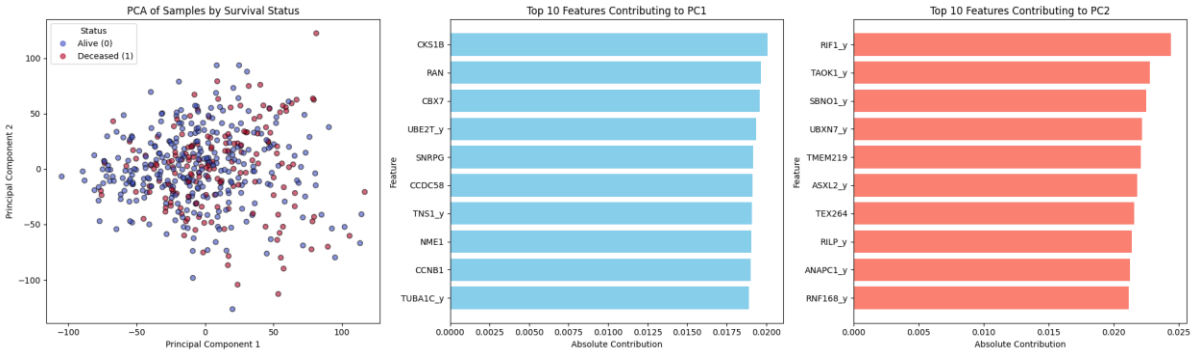

FIG: 5.7 PCA Scatter plot analysis

PCA Scatter Plot of Samples by Survival Status

The left plot shows a **2D scatter plot** where each sample is represented as a point. The samples have been transformed using Principal Component Analysis (PCA), and the plot displays them according to their values on the first two principal components (PC1 and PC2). Each point is colored by survival status — blue for Alive (0) and red for Deceased (1). The overlap between the two classes indicates that survival status is not clearly separable in this 2D PCA space, meaning that the top two components alone do not fully capture survival-related variation. Nonetheless, the spread of points suggests that the dataset contains meaningful variability that PCA has captured. The middle bar chart displays the **top 10 features that contribute most to PC1**, which is the horizontal axis in the scatter plot. PC1 represents the direction in the data that explains the greatest amount of variation across all samples. In other words, the position of a sample along the PC1 axis depends strongly on these top-contributing features. Genes such as **CKS1B**, **RAN**, and **CBX7** are driving the differences along PC1 — meaning that differences in how these genes behave across samples account for much of the variation seen horizontally in the PCA plot. These features likely reflect underlying biological processes that vary the most across the dataset. The right bar chart shows the **top 10 features contributing to PC2**, which corresponds to the vertical axis in the scatter plot. PC2 captures the second most important source of variability in the dataset, independent from PC1. The vertical position of a sample in the scatter plot is largely influenced by these features. Genes such as **RIF1_y**, **TAOK1_y**, and **SBNO1_y** are the strongest drivers of variation along PC2. This means that these features differentiate samples along the vertical dimension, helping to uncover additional patterns or biological differences in the data that are not captured by PC1. (As shown in Fig 5.7)

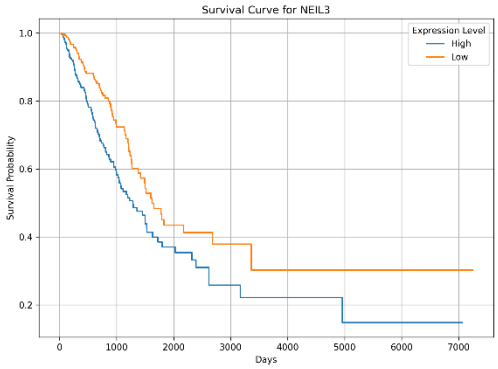

FIG 5.8: Survival curve for NEIL3

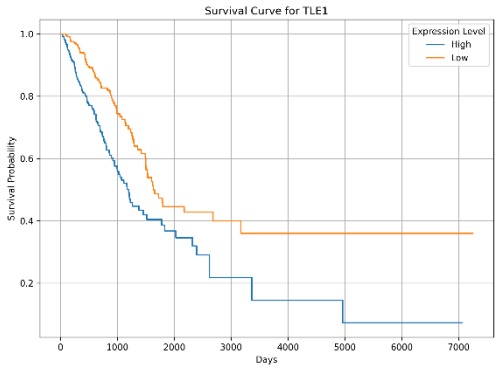

FIG 5.9: Survival curve for TLE1

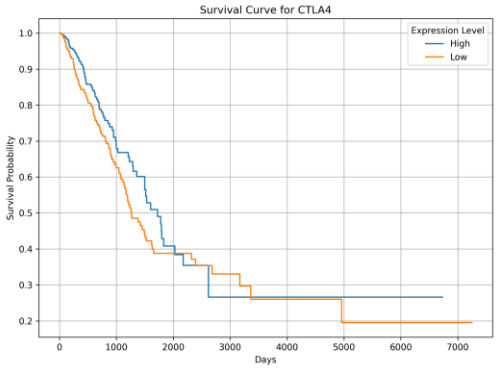

FIG 5.10: Survival curve for CTLA4

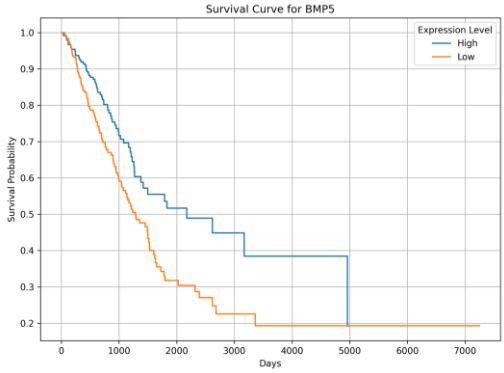

FIG 5.11: Survival curve for BMP5

These four Kaplan-Meier survival curves illustrate the relationship between gene expression levels and patient survival probability over time for the genes **BMP5**, **CTLA4**, **NEIL3** and **TLE1**. In each plot, patients are divided into two groups based on whether their gene expression is high (blue line) or low (orange line). Across the first two genes, a consistent trend emerges: **high expression** of the gene is generally associated with lower survival outcomes. For **TLE1** and **NEIIL3**, patients with high expression show a rapid decline in survival probability. Across the bottom two graphs, a consistent trend of high expression showcasing higher expression of gene is associated with better survival outcome. For **BMP5**, patients with high expression show a slower decline in survival probability compared to those with low expression, suggesting a protective effect. Similarly, the **CTLA4** curve shows that while the difference between high and low expression groups is somewhat less pronounced, the high expression group tends to maintain higher survival probabilities over time. Together, these curves suggest that elevated expression of these genes may be positively correlated with patient prognosis, highlighting their potential relevance as biomarkers for survival outcomes. (As shown in Fig 5.8, 5.9,5.10 & 5.11)

FIG 5.12: Heatmap of top 100 genes

This clustered heatmap visualizes the expression patterns of the **top 100 most informative gene features** across all samples in the dataset. Each row represents a sample, and each column represents a gene, with color indicating the level of expression — red for high expression, blue for low expression, and white for intermediate values. Hierarchical clustering has been applied to both the genes (columns) and the samples (rows), with the resulting dendrograms shown along the top and left sides of the heatmap. This clustering groups together genes with similar expression profiles and samples that exhibit similar patterns across these genes. The presence of distinct blocks of red and blue suggests that certain genes are consistently upregulated or downregulated in specific groups of samples. The clustering of samples may reflect underlying biological differences, such as survival status or disease subtypes, while the gene clusters could point to co-regulated pathways or functional gene sets. Overall, this heatmap highlights structured variation in the dataset and provides insights into the relationships between genes and samples. (As shown in Fig 5.11)

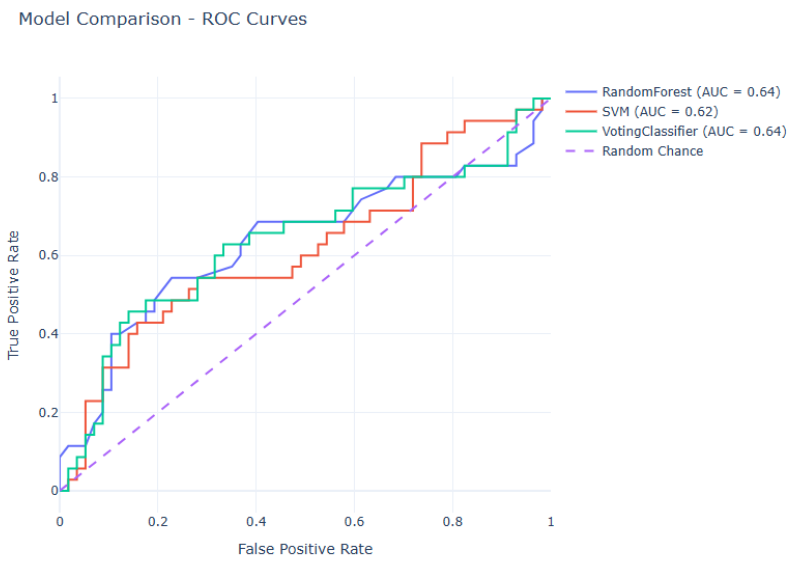

FIG 5.13: Model Comparison - ROC Curves

The ROC curve compares the performance of three models—Random Forest, SVM, and Voting Classifier—for predicting survival in LUAD patients. All models performed better than random chance, with Random Forest and Voting Classifier achieving the highest Area Under the Curve (AUC) of 0.64, followed closely by SVM with an AUC of 0.62. Although these AUC values indicate moderate predictive ability, they suggest that the models can distinguish between survival outcomes to some extent. However, further improvements, such as better feature selection or model tuning, may enhance predictive accuracy.

g: Profiler Results

Survival Genes

1.BMP5

BMP5 (Bone Morphogenetic Protein 5) is part of the TGF-β superfamily, involved in: Bone and cartilage development, cell growth and differentiation, tissue architecture. BMP5 is often studied in the context of cancer, inflammation, and developmental biology. The functional enrichment analysis of BMP5 reveals its involvement in key molecular functions and signaling pathways relevant to lung adenocarcinoma (LUAD) progression and patient survival. Notably, BMP5 is associated with BMP receptor binding and receptor signaling regulation, indicating its role in modulating the TGF-β signaling pathway, an essential regulator of cell proliferation, differentiation, and apoptosis in cancer. Biological process terms such as the negative regulation of insulin-like growth factor receptor signaling and steroid hormone biosynthesis suggest that BMP5 may suppress tumor-promoting pathways,

potentially contributing to improved clinical outcomes. Additionally, its association with the Hippo signaling and cytokine-cytokine receptor interaction pathways links BMP5 to mechanisms controlling tumor growth, immune response, and microenvironment interactions. While the adjusted p-values in the enrichment results are not statistically significant, the biological context supports BMP5 as a gene of interest, with possible prognostic relevance in LUAD survival through its influence on tumor behavior and systemic signaling networks. (As shown in Fig 5.14)

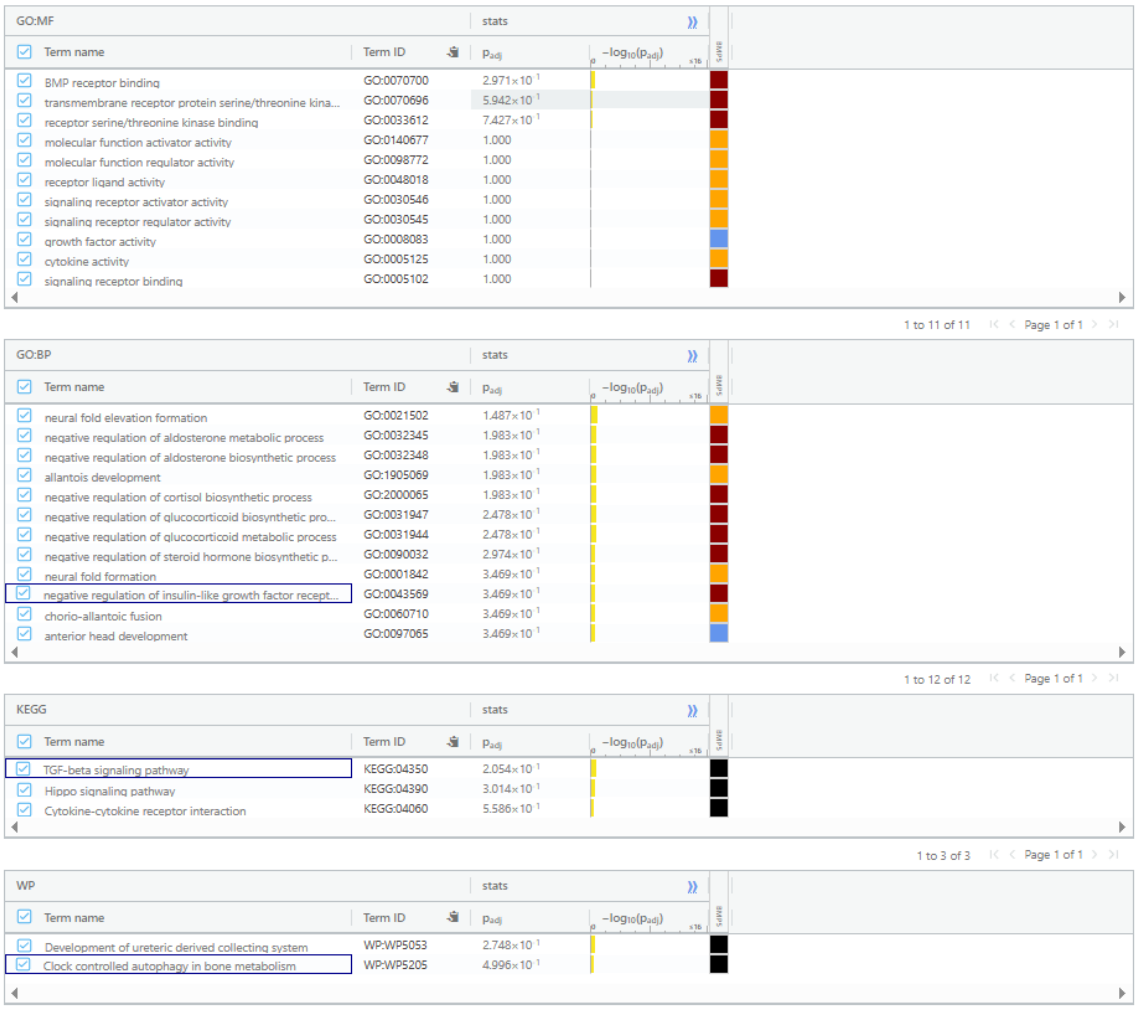

Fig 5.14: Detailed overview of BMP5 with log values and Term ID.

2.CTLA4

In this study, CTLA4 was found to be enriched in patients who survived lung adenocarcinoma (LUAD), suggesting a potential role in favorable clinical outcomes. CTLA4 is an immune checkpoint molecule known to suppress T cell activation and is often associated with immune evasion by tumors. However, its presence in survivors may indicate an immune microenvironment primed for response to immunotherapy. Specifically, high CTLA4 expression may reflect an active but regulated immune landscape where checkpoint inhibition

(e.g., anti-CTLA4 therapies) could be particularly effective. Furthermore, enrichment of CTLA4-related pathways, including "T cell receptor signaling," "co-inhibition by CTLA4," and "cancer immunotherapy by CTLA4 blockade," supports its relevance as both a prognostic marker and therapeutic target. These findings align with growing evidence that immune checkpoint activity—while potentially suppressive—can signal a **therapy-responsive state**, especially in patients receiving or eligible for checkpoint inhibitors. Thus, CTLA4 may serve as a valuable biomarker in stratifying LUAD patients for immunotherapy and predicting long-term survival outcomes. (As shown Fig 5.15)

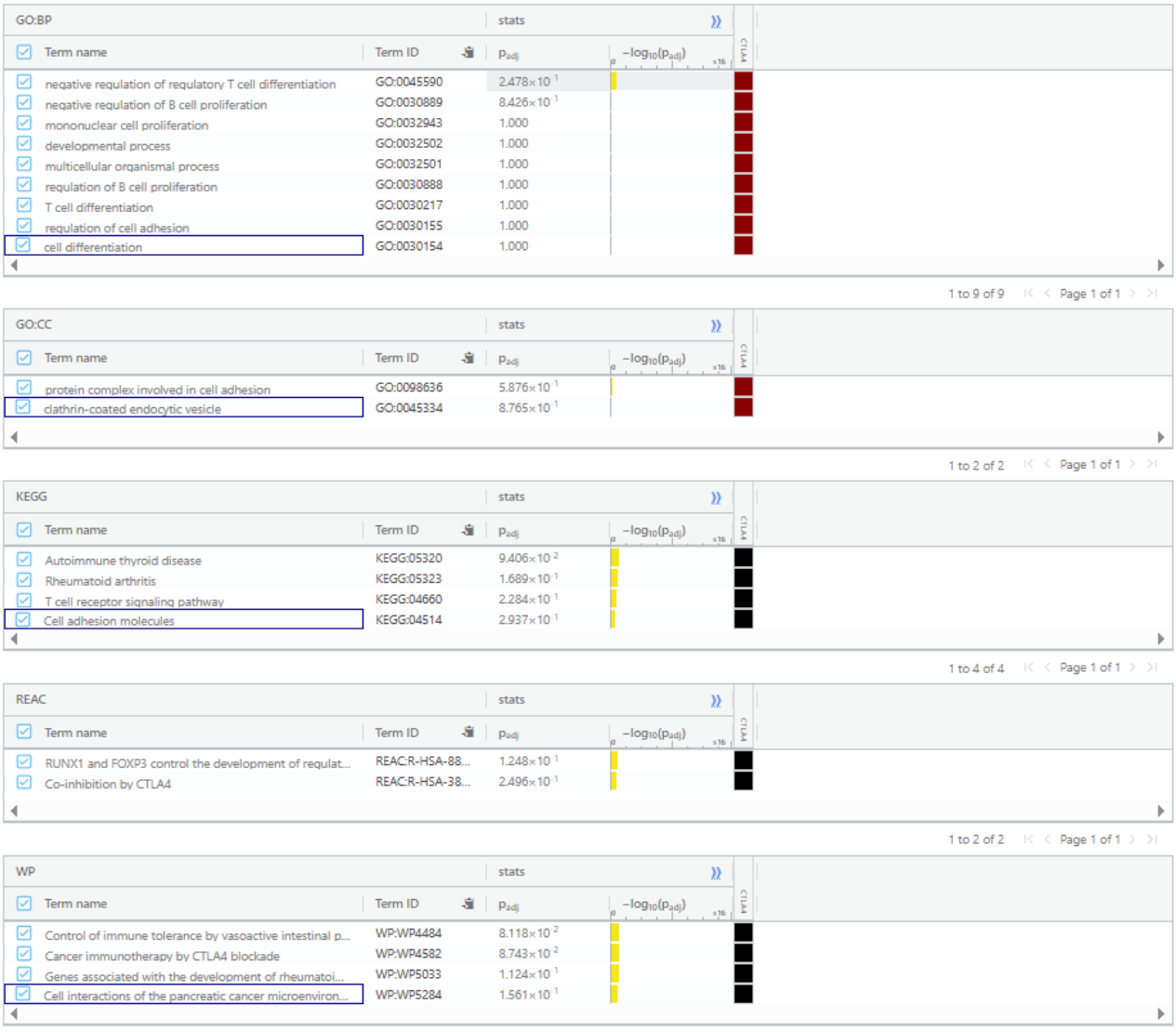

FIG 5.15: Detailed overview of CTLA4 with log values and Term ID.

Deceased Genes

1.TLE1

TLE1, identified as enriched in deceased LUAD patients, appears to play a critical role in transcriptional repression and oncogenic pathway regulation. Enrichment analysis reveals its involvement in DNA-binding transcription factor regulation, suppression of anoikis, and modulation of Wnt and Notch signaling pathways—core mechanisms known to drive tumor survival, metastasis, and therapy resistance. Specifically, TLE1's association with the β-catenin–TCF complex and deactivation of β-catenin transactivation points to a nuanced role in shaping the Wnt signaling landscape, a pathway frequently altered in lung cancers. The repression of anoikis highlights its potential to promote metastatic dissemination, while the convergence of multiple signaling cascades underscores its contribution to aggressive tumor behavior. These findings suggest that elevated TLE1 activity may serve as a marker of poor prognosis and a potential target in high-risk LUAD patients. (As shown in Fig 5.16)

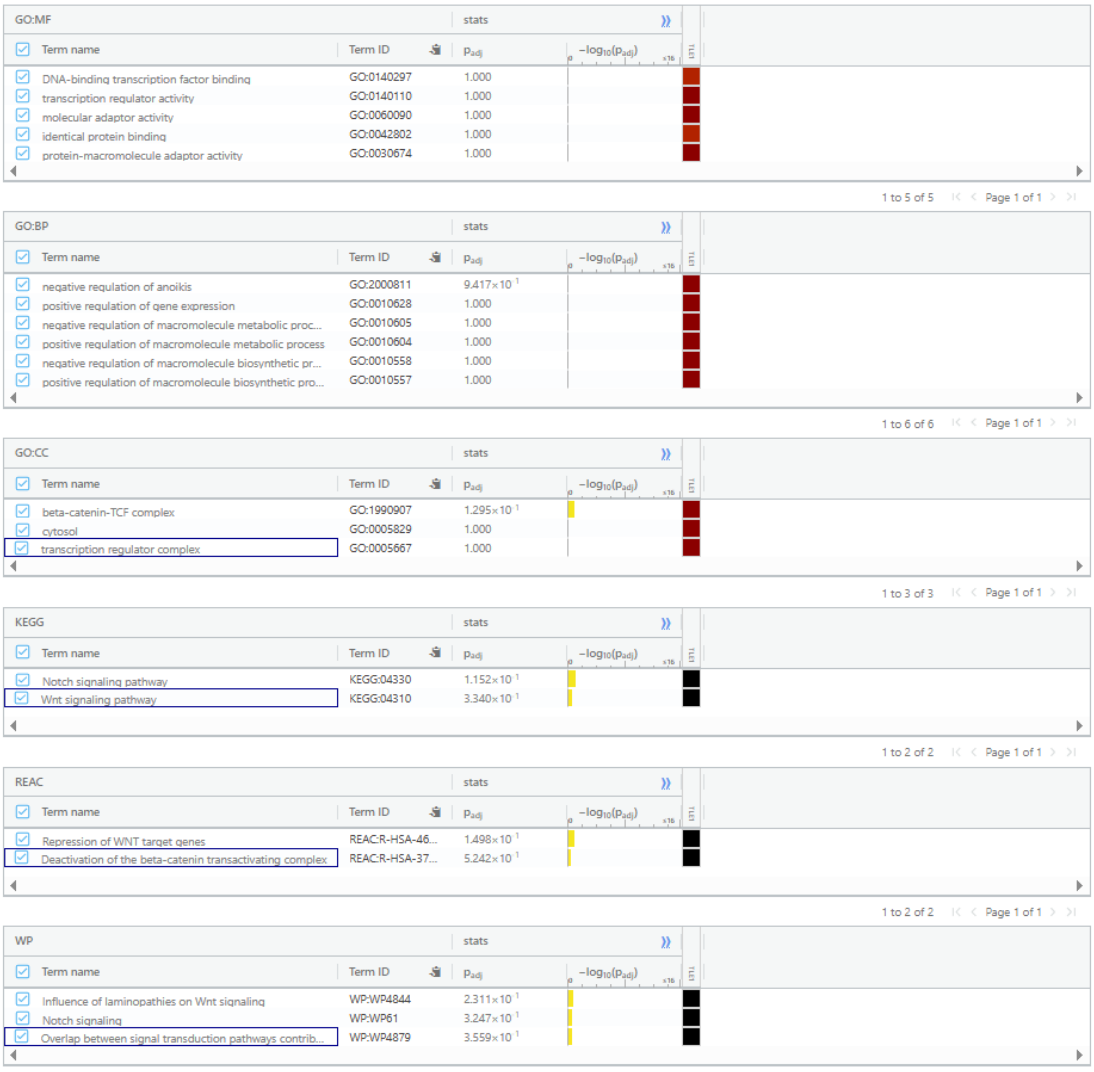

FIG 5.16: Detailed overview of TLE1 with log values and Term ID.

2.NEIL3

NEIL3, identified as enriched in deceased LUAD patients, plays a pivotal role in the base excision repair (BER) pathway, responsible for correcting oxidative DNA damage. Functional enrichment reveals NEIL3's strong association with DNA glycosylase activity, MCM complex binding, and repair processes such as AP site formation and depurination. These functions suggest that NEIL3 enables tumor cells to maintain genomic integrity under high replication stress or oxidative conditions, potentially promoting survival of malignant cells. Furthermore, its involvement in Reactome pathways such as "Diseases of Base Excision Repair" and "Defective BER associated with NEIL3" indicates that its dysregulation may lead to aberrant repair, increased mutational burden, and tumor aggressiveness. Importantly, enhanced DNA repair capability is often linked to resistance against DNA-damaging therapies such as chemotherapy and radiation, which could explain its association with poor patient survival. Thus, NEIL3 may serve as a marker of therapeutic resistance and a target for sensitizing LUAD tumors to genotoxic treatments. (As shown in Fig 5.17)

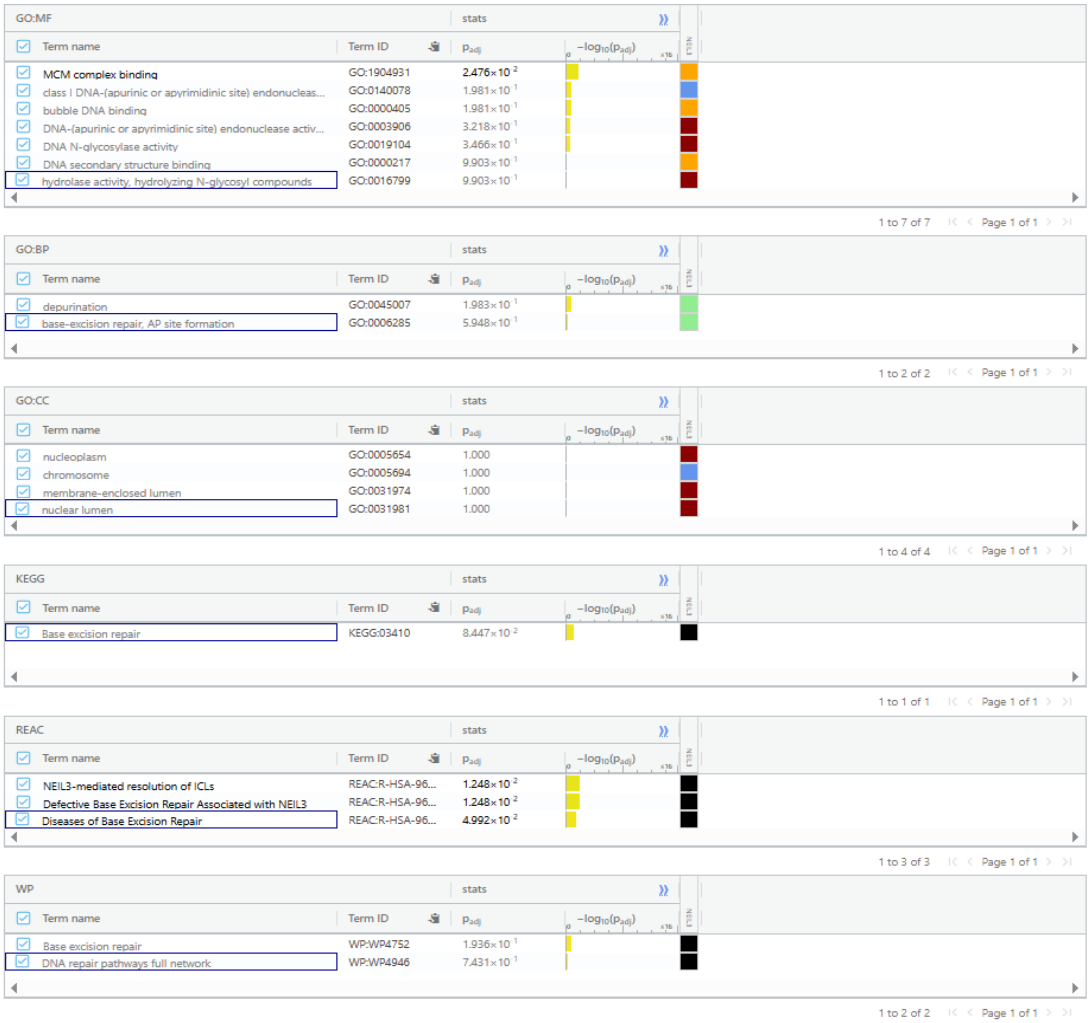

FIG 5.17: Detailed Overview of NEIL3 with log values and Term ID.

Model Comparison Report

Table 1: Random Forest Accuracy

Random Forest Accuracy: 0.6522

	precision	recall	f1_score	support
0	0.70	0.80	0.75	59
1	0.52	0.39	0.45	33
accuracy			0.65	92
macro_avg	0.61	0.60	0.60	92
weighted_avg	0.64	0.65	0.64	92

Table 2: SVM Accuracy

SVM Accuracy: 0.6304

	precision	recall	f1_score	support
0	0.69	0.76	0.73	59
1	0.48	0.39	0.43	33
accuracy			0.63	92
macro_avg	0.59	0.58	0.58	92
weighted_avg	0.62	0.63	0.62	92

Table 3: Voting Classifier Accuracy

Voting Classifier Accuracy: 0.6304

	precision	recall	f1_score	support
0	0.70	0.75	0.72	59
1	0.48	0.42	0.45	33
accuracy			0.63	92
macro_avg	0.59	0.59	0.59	92
weighted_avg	0.62	0.63	0.62	92

6.Summary And Conclusion

This study addresses the urgent need for better survival prediction and biomarker discovery in Lung Adenocarcinoma (LUAD), a leading cause of cancer-related deaths. By integrating multi-omics data—RNA-seq expression profiles, mutation data, and clinical metadata—from The Cancer Genome Atlas (TCGA), the project develops and evaluates machine learning models to classify patient survival outcomes. Three classifiers—Random Forest (RF), Support Vector Machine (SVM), and a Voting Classifier—were trained, with the Random Forest model achieving the highest accuracy (65.2%) and AUC (0.64). Feature importance from RF helped identify key genes driving model predictions.

A set of top-ranked genes was divided into survival-associated and death-associated groups. Functional enrichment analysis using g:Profiler revealed that survival-associated genes (e.g., BMP5, CTLA4) were linked to immune signaling and apoptosis regulation, while death-associated genes (e.g., TLE1, NEIL3) were involved in oncogenic pathways, DNA repair, and metastasis-enabling mechanisms like anoikis resistance. Kaplan-Meier analysis confirmed significant survival differences based on expression of these genes, suggesting their prognostic value. The pipeline not only demonstrated the predictive capability of machine learning on multi-omics data but also uncovered biologically relevant markers for clinical stratification.

Conclusion

The integration of machine learning with multi-omics datasets presents a promising framework for survival prediction and biomarker discovery in LUAD. This project successfully demonstrates how computational models can stratify patients based on gene expression and mutational profiles, enabling insights into disease biology and prognosis. Genes such as BMP5 and CTLA4 were associated with better survival outcomes, likely due to their roles in immune regulation, while TLE1 and NEIL3 were linked to poor prognosis via involvement in Wnt signaling and DNA repair pathways, respectively.Though current model performance is moderate, the findings are biologically meaningful and provide a strong foundation for future improvements. This study contributes to precision oncology by identifying potential prognostic biomarkers and offering a reproducible machine learning pipeline. Future work incorporating more omics layers (e.g., proteomics), larger cohorts, and deep learning models may further enhance predictive power and clinical relevance.

7. Future Prospects and Application

The integration of machine learning with multi-omics data for survival prediction in lung adenocarcinoma (LUAD) marks a significant advancement in both computational biology and clinical oncology. This approach not only enhances the accuracy of prognostic models but also opens up new avenues for the discovery of biomarkers and therapeutic targets. As the field evolves, several exciting prospects and practical applications are anticipated.

Looking ahead, expanding the scope of multi-omics integration to include additional data types such as proteomics, metabolomics, and single-cell sequencing will provide a more comprehensive molecular portrait of LUAD. This broader perspective can facilitate the identification of novel disease mechanisms and potential intervention points that are not apparent from transcriptome or genomic data alone. The increasing availability of large, well-annotated datasets will further enable the development of robust and generalizable machine learning models, ensuring that findings are applicable across diverse patient populations.

Advancements in machine learning methodologies are also expected to play a pivotal role. The adoption of deep learning models and advanced ensemble techniques can capture complex, non-linear relationships within high-dimensional data, potentially leading to improved predictive performance. Additionally, the integration of explainable artificial intelligence tools will help make model predictions more transparent and interpretable, fostering greater trust and adoption among clinicians and researchers.

The practical applications of these innovations are far-reaching. In clinical settings, machine learning-based multi-omics models can be used to stratify patients according to risk, enabling more personalized treatment plans. High-risk patients can be identified early and prioritized for aggressive therapies or clinical trials, while low-risk patients might benefit from less intensive treatment regimens. Furthermore, the biomarkers identified through these approaches may serve as candidates for targeted therapies or as predictors of response to immunotherapy, aiding in the selection of the most effective treatment strategies for individual patients.

Beyond lung adenocarcinoma, the methodologies developed in this project are highly transferable to other cancer types and complex diseases. The pipeline of multi-omics integration, machine learning modeling, and biological interpretation can be adapted to address similar challenges in breast, colorectal, or prostate cancer, as well as in non-cancer diseases where multifactorial etiologies are involved.

In conclusion, the future of machine learning-based multi-omics integration in LUAD and other diseases is promising. Continued advancements in computational methods, data availability,

and interdisciplinary collaboration will accelerate the translation of these approaches from research to clinical practice, ultimately contributing to more precise and effective healthcare.

8. References

1. Cervantes J., Garcia-Lamont F., Rodriguez-Mazahua L., Lopez A. (2020). Support vector machine classification for large datasets. *Expert Systems with Applications*, 153, 113374.
2. Evgeniou T., Pontil M. (2001). Support vector machines: Theory and applications. *Machine Learning*, 46(1–3), 1–47.
3. Haoyu Yang, Zheng An, et al. (2018). Machine learning in bioinformatics: Methods, tools, and applications. *Briefings in Bioinformatics*, 19(6), 1234–1248.
4. Svetnik V., Liaw A., Tong C., Culberson J.C., Sheridan R.P., Feuston B.P. (2003). Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6), 1947–1958.
5. Wang T., Zhu X., Wang K., Li J., Hu X., Lin P., Zhang J. (2023). Transcriptional factor MAZ promotes cisplatin-induced DNA damage repair in lung adenocarcinoma by regulating NEIL3. *Pulmonary Pharmacology & Therapeutics*, 80, 102217.
6. Wangyang M., Han X., Zhao R., Li D., Li K., Meng Y., Chen J., Wang Y., Liao Y. (2021). The prognostic value of bone morphogenetic proteins and their receptors in lung adenocarcinoma. *Frontiers in Oncology*, 11, Article 608239.
7. Xu Z., Chen C. (2021). Abnormal expression and prognostic significance of bone morphogenetic proteins and their receptors in lung adenocarcinoma. *BioMed Research International*, 2021, Article 6663990.
8. Yao X., Kale Ireland S., Pham T., Temple B., Chen R., Raj M.H.G., Biliran H. (2014). TLE1 promotes EMT in A549 lung cancer cells through suppression of E-cadherin. *Biochemical and Biophysical Research Communications*, 455(3–4), 277–284.
9. Yuan D., Yang X., Yuan Z., Zhao Y., Guo J. (2017). TLE1 function and therapeutic potential in cancer. *Oncotarget*, 8(9), 15971–15976.
10. Zhang H., Dai Z., Wu W. et al. (2021). Regulatory mechanisms of immune checkpoints PD-L1 and CTLA-4 in cancer. *J Exp Clin Cancer Res*, 40, 184.
11. Zhang W., Zhao L., Zheng T., et al. (2024). Advances in multi-omics integration for cancer prognosis. *Nature Reviews Cancer*, 24(2), 123–137.

12. Zhao C., Liu J., Zhou H. et al. (2021). NEIL3 may act as a potential prognostic biomarker for lung adenocarcinoma. *Cancer Cell Int*, 21, 228.
13. Zheng Q., Min S., Zhou Q. (2021). Identification of potential diagnostic and prognostic biomarkers for LUAD based on TCGA and GEO databases. *Bioscience Reports*, 41(6), BSR20204370.
14. Zhou Z.-H. (2009). Ensemble learning. In: *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC.