UTM UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTING
UTM Johor Bahru

SECB3203 PROGRAMMING FOR BIOINFORMATICS

Report

Semester 1 2025/2026

Section 01 Group 12

NAME	MATRIC NUMBER
NGU YU LING	A23CS0149

Lecturer: **DR. SEAH CHOON SEN**

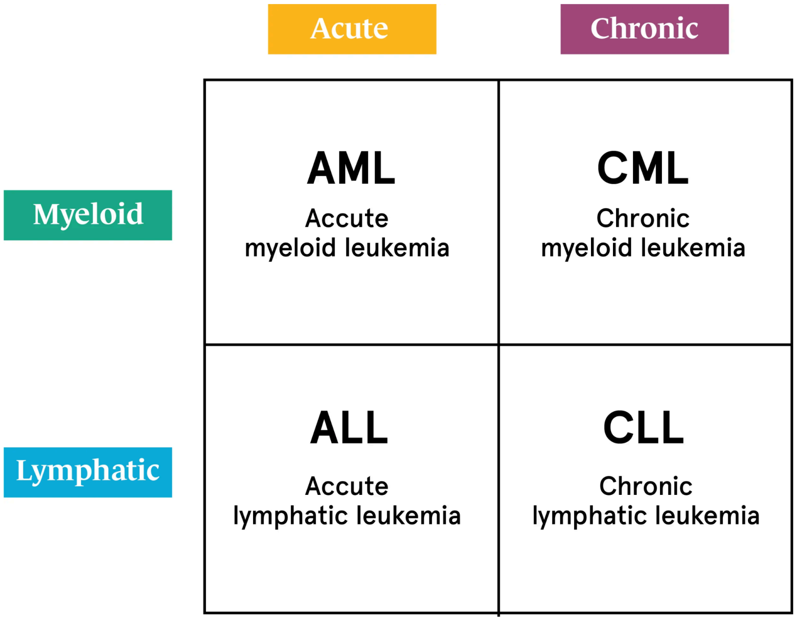
Date: 2026

TABLE OF CONTENT

1.0 INTRODUCTION

1.1 Problem Background

Leukemia, a group of blood cancers, is highly heterogeneous, comprising four major subtypes: Acute Lymphoblastic Leukemia (ALL), Acute Myeloid Leukemia (AML), Chronic Lymphocytic Leukemia (CLL), and Chronic Myeloid Leukemia (CML).



	Acute	Chronic
Myeloid	AML Accute myeloid leukemia	CML Chronic myeloid leukemia
Lymphatic	ALL Accute lymphatic leukemia	CLL Chronic lymphatic leukemia

Figure 1: Leukemia and Lymphoma: The Differences and Similarities (rocky-mountain-cancer-center)

Accurate and rapid diagnosis of the specific subtype is paramount, as treatment protocols and prognosis vary dramatically across these classifications. Misclassification can lead to suboptimal or potentially toxic therapies, directly impacting patient survival and quality of life. Traditional diagnostic methods, such as morphology assessment and basic flow cytometry, can be subjective, time-consuming, and sometimes lack the molecular resolution required for precise classification.

This project addresses the need for a faster, objective, and quantitative diagnostic tool based on molecular gene expression data. The goal is to leverage the vast amounts of genomic data available in public repositories like the Gene Expression Omnibus (GEO) to create a computational tool that assists in molecular diagnosis, moving towards personalized medicine.

1.2 Problem Statement

The central challenge in computational diagnostics is handling the high dimensionality and noise inherent in genomic data. The problem is to develop a reliable, machine learning classification model using public gene expression data that can accurately distinguish and predict the four major leukemia subtypes (ALL, AML, CLL, and CML) with a high degree of confidence (e.g., achieving a macro-averaged F1-score greater than 0.85) and demonstrate its generalizability using rigorous cross-validation. This requires successful data wrangling, aggressive feature selection to mitigate overfitting, and systematic model optimization.

1.3 Objectives

The primary objectives of this project are designed to follow a complete bioinformatics data science pipeline, from data acquisition to model evaluation:

- To identify, collect, and preprocess a suitable multi-class gene expression dataset (from GEO) and perform necessary steps like normalization, probe-to-gene mapping, and cleaning to prepare the data for machine learning.
- To perform comprehensive Exploratory Data Analysis (EDA), including statistical analysis (ANOVA) and visualization (box plots), to identify initial gene candidates that show statistically significant expression differences between the four leukemia subtypes.
- To apply effective feature selection methodologies (variance filtering, SelectKBest, or Random Forest feature importance) to reduce the dataset dimensionality.
- To implement and train at least two distinct machine learning classification models (Logistic Regression and Random Forest) capable of multi-class prediction.

- To evaluate all trained models rigorously using stratified K-fold cross-validation and key multi-class metrics (Precision, Recall, F1-score, and a Confusion Matrix) to select the optimal, most generalized classifier.

1.4 Scopes

This project is defined by the following inclusions and exclusions:

	Inclusion (What the project WILL do)	**Exclusion (What the project WILL NOT do)**
Data Source	Utilize secondary, publicly available high-throughput gene expression data from databases like GEO.	Generate or process primary sequencing data; rely on private or proprietary datasets.
Classification	Focus on classifying the four major leukemia subtypes: ALL, AML, CLL, and CML.	Attempt to classify rare subtypes or predict patient survival/prognosis.
Model	Implement supervised machine learning algorithms for multi-class classification.	Use unsupervised learning methods (clustering) as the primary analysis goal.
Biomarker	Identify a panel of key features (genes) that contribute to the model's predictive power.	Conduct deep biological validation or mechanistic studies of the identified biomarkers.
Tools	Develop the analysis pipeline entirely in Python using standard bioinformatics libraries.	Use specialized statistical software or build a full-scale web application/API.

Table 1: Inclusions and exclusions of this project in different aspects

1.5 Conclusion

This project provides a comprehensive demonstration of a bioinformatics data science pipeline, from data wrangling to model optimization. The successful classification model and the resulting biomarker panel will serve as a powerful proof-of-concept, illustrating how machine learning can be leveraged to deliver objective, molecular insights for the critical task of leukemia subtype diagnosis.

2.0 DATA COLLECTION AND PRE-PROCESSING

This phase details the identification, acquisition, and preparation of the raw gene expression data required for the multi-class leukemia classification task. The primary goal is to transform the high-dimensional, raw microarray data into a clean, labeled matrix suitable for machine learning training and evaluation.

2.1 Importing Dataset

The dataset selected for this project is the **Microarray Innovations in LEukemia (MILE) study Stage 2**, publicly available under the Gene Expression Omnibus (GEO) accession **GSE13164**. (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE13164)

Characteristic	Detail	Rationale for Selection
Accession ID	GSE13164 (MILE Study Stage 2)	Selected for its large sample size while maintaining reasonable download and processing feasibility on local hardware.
Sample Size (*N*)	*N* = 1,152 samples	Provides a robust cohort size to minimize overfitting and maximize statistical power for model evaluation.
Data Type	Gene expression profiling by array (Roche AmpliChip Leukemia Custom Microarray)	Standardized microarray format, ideal for the planned Python analysis pipeline.
Leukemia Subtypes	Includes ALL, AML, CLL, and CML	Direct support for the project's multi-class classification problem statement.

Table 2: Dataset details and rationales of selection

Data acquisition was performed using the `GEOparse` Python library, which handles the secure download and initial parsing of the raw data files, including the normalized expression matrix and the corresponding sample metadata (GSM files).

2.2 Data Wrangling

Data wrangling is a critical step to standardize the raw gene expression matrix and clean the clinical labels. This process consists of three main stages: label refinement, probe-to-gene mapping, and data aggregation.

2.2.1 **Label refinement and filtering** the raw metadata from the GEO series contains detailed subtype information. To match the project's scope, the following steps were taken:

- Standardization: All detailed subtypes were mapped to the four target classes: ALL, AML, CLL, and CML.
- Filtering: Any non-target samples (e.g., normal controls, MDS, or unclassified samples) were excluded to focus purely on the four classification types.
- Encoding: The final categorical labels (ALL, AML, CLL, CML) were converted into numerical codes (e.g., 0, 1, 2, 3) using Scikit-learn's LabelEncoder for machine learning input.

2.2.2 **Probe-to-gene mapping and aggregation** because the microarray data uses Probe IDs to measure expression, not official Gene Symbols. To create an interpretable feature set, probes must be mapped to genes:

- Mapping: The Platform Data (GPL file) was used to associate each Probe ID with its corresponding Gene Symbol.
- Handling Duplicates: Since multiple probes can target the same gene, the expression values for these duplicate probes were aggregated by calculating the mean intensity, resulting in a single expression value per unique gene symbol.
- Final Matrix Construction: This process transforms the initial high-dimensional matrix into the final, cleaned dataset (approx. 20,000 Gene Symbols x N_{filtered} samples), with the numerical target code attached as the final column.

2.3 Software And Hardware Requirements

The data wrangling process was revised to construct the feature matrix by aggregating data from individual Sample (GSM) files, executed by the `data_wrangling.py`.

2.3.1 Software Requirements

The primary software requirement is a Python 3.8+ environment. The following core libraries will be utilized:

Category	Library	Purpose
Data Acquisition	`GEOparse`	Downloading and parsing raw gene expression data files from the Gene Expression Omnibus (GEO).
Data Processing	`pandas, numpy`	Essential for data wrangling, cleaning, normalization, merging expression data with clinical metadata, and linear algebra operations.
Statistics/ EDA	`scipy.stats`	Conducting statistical tests (e.g., ANOVA or t-tests) to assess differential gene expression (Objective O2).
Machine Learning	`scikit-learn`	Implementing classification models (Logistic Regression, Random Forest), feature selection (e.g., `SelectKBest`), cross-validation (`KFold`), and hyperparameter tuning (`GridSearchCV`).
Visualization	`matplotlib, seaborn`	Generating descriptive visualizations for EDA (histograms, heatmaps) and model evaluation (ROC curves, Confusion Matrices).
Reporting	Python environment	Documentation of the pipeline and results.

Table 3: Libraries needed and their purpose in different phase of the peoject

2.3.2 Hardware Requirements

- Given that gene expression datasets can contain thousands of features, moderate hardware is required to handle the data size and computational demands of model training and cross-validation:
- Processor (CPU): Multi-core processor for efficient processing of data frames and model training.
- Memory (RAM): Minimum of 8 GB RAM to handle large gene expression matrices.
- Storage: Sufficient local storage to store the downloaded GEO files and the analysis environment.
- Platform: A Python environment installed locally (e.g., Anaconda) or cloud resources (e.g., Azure Notebooks or Google Colab) to ensure reproducibility.

2.3.3 Data Matrix Assembly

Since the full expression matrix was not consistently available, the data was assembled iteratively. The custom `data_wrangling.py` first filtered the 973 individual Sample (GSM) files to ensure only the four target leukemia subtypes were included. It then extracted the Probe ID and expression value from each GSM file. These individual sample columns were merged using the Probe ID as the key, which resulted in a raw expression matrix containing 1,480 Probes and 973 Samples.

2.3.4 Feature Consolidation (Probe-to-Gene Mapping)

The GenBank Accession number `GB_ACC` was used as the unique gene identifier. All Probes mapping to the same `GB_ACC` were grouped together, and their expression values across all samples were averaged (mean) to represent the consolidated gene expression level. This aggregation step successfully reduced the feature space from 1,480 Probes to 1,420 unique Gene Accessions.

2.3.5 Final Output Structure

The data wrangling process completed by generating two clean, aligned files that are ready for downstream machine learning tasks:

	Role in Analysis	**Final Shape (Samples x Features)**
GSE13164_cleaned_features.csv	Input Features (X)	(973, 1420)
GSE13164_cleaned_labels.csv	Target Labels (Y)	(973, 3)

Table 4: Files generated and their usage in the next phase

The alignment of 973 sample rows across both the feature and label files was verified.

3.0 FLOWCHART OF THE PROPOSED APPROACH

3.1 Exploratory Data Analysis

3.2 Model Development

3.3 Model Evaluation

4.0 TESTING AND VALIDATION

5.0 CONCLUSIONS