Molecular Biology Data Analysis with Streamlit and Docker

Alexandros Sinekidis, inf2022186
Nikoletta Hadjiangeli, inf2022241
Thomas Nakos, inf2022141
Department of Informatics
Academic Year: 2024-2025

May 2025

Figure 1: Ionian University logo.

Abstract

This project presents the development of an interactive application in **Streamlit** for molecular biology data analysis. The application includes functionalities for *PCA*, *Clustering*, *Volcano Plot*, and processing of the COVID-19 dataset. Additionally, execution via **Docker** is provided, ensuring portability and independence from local settings. The objective is to simplify exploratory data analysis workflows in molecular biology through intuitive interfaces and interactive visualizations.

1 Introduction

The rapid increase of biological data due to high-throughput technologies like RNA-seq, microarrays, and single-cell sequencing has created a need for intuitive and extensible analysis tools. Researchers without strong programming backgrounds often find traditional scripting-based tools (like R or Python) to be a barrier to efficient data interpretation.

Our solution leverages the Streamlit framework to create an interactive web-based platform for molecular data analysis, integrating key machine learning (ML) and statistical tools in a user-friendly manner. Containerization with Docker ensures that the environment remains consistent across different systems and eliminates the "it works on my machine" problem.

The main goals of this project are:

- To facilitate gene expression data analysis through an accessible GUI.
- To visualize complex relationships (via PCA and Clustering).
- To identify and interpret differentially expressed genes (via Volcano Plot).
- To ensure reproducibility and ease of deployment via Docker.

2 Implementation Design

The application is modular and split into four primary functionalities:

1. **PCA Analysis**: Dimensionality reduction and visualization of expression data to identify major patterns.
2. **Clustering**: Grouping samples or genes based on similarity, often used for subtype discovery.
3. **Volcano Plot**: Visualization of gene expression changes (log2 fold change vs statistical significance).
4. **COVID-19 Dataset Viewer**: Exploration of curated COVID-19 gene expression data.

Technological Stack

- **Frontend**: Implemented in Streamlit, allowing rapid UI prototyping.
- **Backend**: Uses pandas for data handling, scikit-learn for ML, seaborn/matplotlib for plotting.
- **Data Input**: Supports multiple file types including ‘.csv‘, ‘.tsv‘, ‘.xlsx‘, ‘.txt‘.

Streamlit widgets allow users to:

- Upload data files.
- Choose PCA dimensions (e.g., PC1 vs PC2).
- Set cluster number for k-means.
- Define thresholds for log2FC and p-value cutoffs.

3 UML Diagrams

3.1 Use Case Diagram

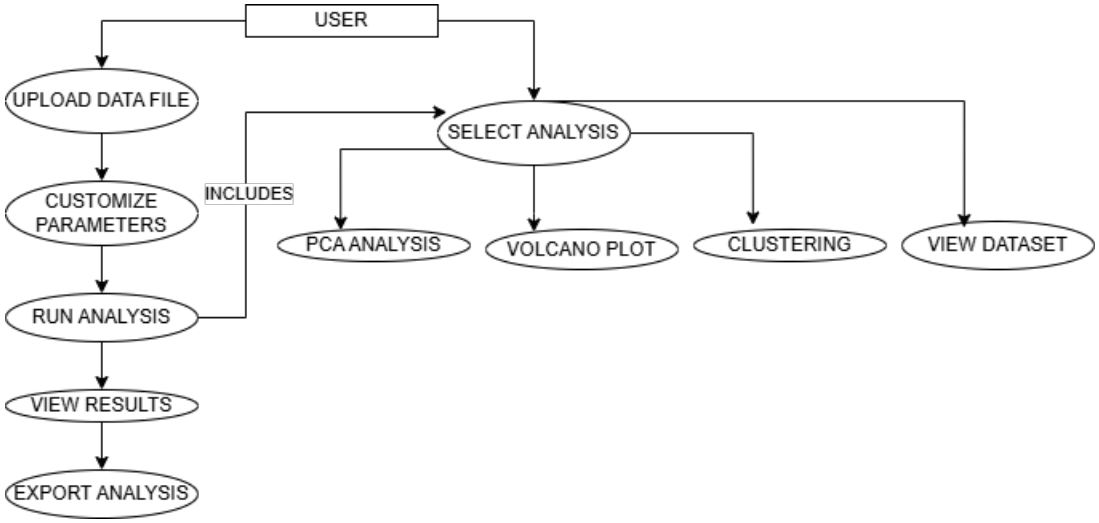

Figure 2: Use Case diagram of the application

The use case diagram emphasizes the role of the user in initiating and configuring the analysis pipeline. Customization before execution allows for flexibility in exploration.

3.2 Component Diagram

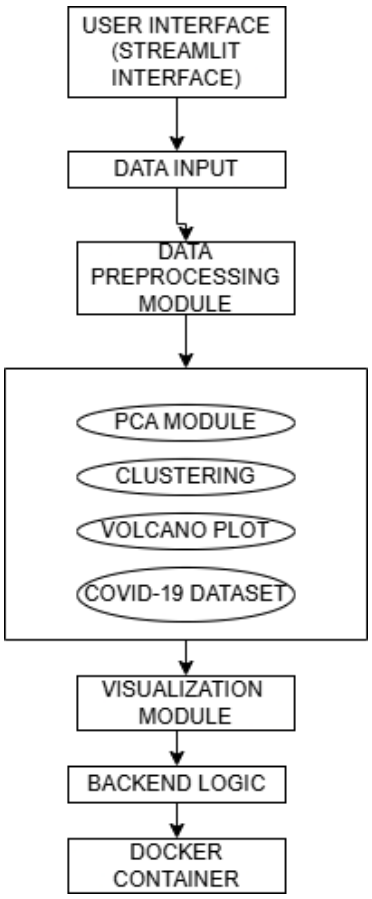

Figure 3: Component diagram of the application's architecture

Data Flow Summary:

- Input → Preprocessing → Analysis → Visualization
- The results can be exported or stored in memory for chaining analyses.

The modular nature makes it easy to add future features (e.g., Heatmaps, t-SNE, Gene Ontology).

4 Implementation Analysis

The implementation is handled via a single 'app.py' script organized using tabs and sidebars. Key modules and their responsibilities are:

Preprocessing

- Detects numerical columns.
- Handles missing values via imputation.
- Standardizes the data using `StandardScaler`.

PCA Module

- Computes principal components.
- Allows 2D/3D visualization.
- Supports explained variance ratio plots.

Clustering Module

- Uses k-means clustering.
- Projects data via PCA before clustering.
- Supports silhouette score output (optional).

Volcano Plot

- Visualizes up- and down-regulated genes.
- User-defined log2FC and p-value thresholds.
- Highlights significant genes in color.

COVID-19 Dataset

- Pre-loaded dataset from public COVID-19 research.
- Filter by condition (infected vs control).
- Display summary statistics and sample metadata.

5 Result Visualizations

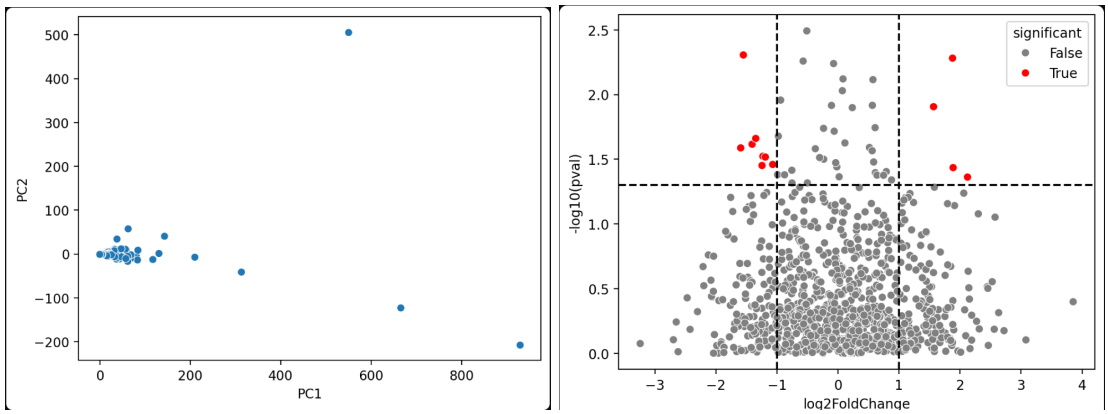

Figure 4: Left: PCA plot. Right: Volcano plot with log2FC and p-values

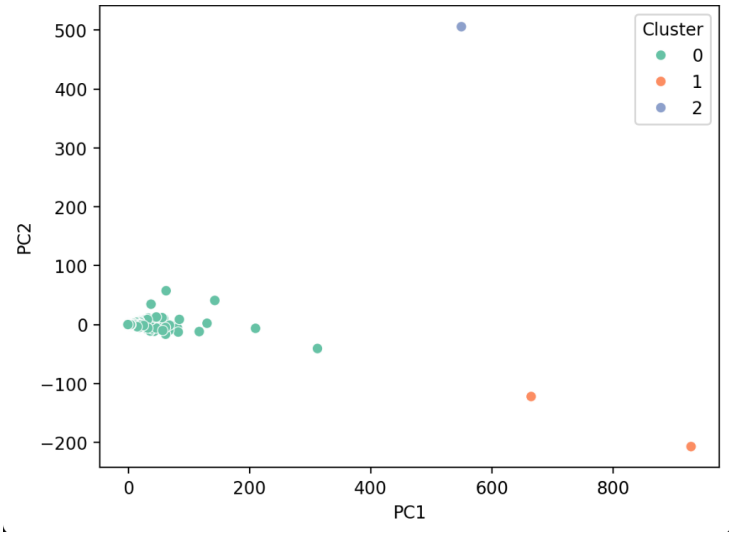

Figure 5: Clustering (k-means) projection in 2D space via PCA

The visual output allows researchers to:

- Detect sample outliers.
- Identify gene expression trends.
- Assess clustering quality.
- Focus on biologically relevant gene subsets.

6 Dockerization

Docker provides a reproducible runtime environment. All dependencies are encapsulated within a container, preventing version mismatches.

Included Files

- `Dockerfile` – Defines the build process.
- `requirements.txt` – Streamlit, numpy, pandas, seaborn, etc.
- `.dockerignore` – Ignores unnecessary files during build.

Usage Instructions

```
docker build -t gene-app .
docker run -p 8501:8501 gene-app
```

7 GitHub Repository

The full codebase, example data, screenshots, and environment configuration are available on GitHub:

`https://github.com/banana-babap/Gene-Expression-Analysis-App-on-Python-Streamlit`

The repository includes:

- Annotated source code.

- Sample datasets.
- Instructions for both Docker and manual installation.

8 Conclusion

This project demonstrates how a modern data science stack can streamline complex bioinformatics workflows. Through Streamlit, we have enabled non-programmers to explore molecular data interactively, while Docker ensures reproducibility and seamless deployment. Ultimately, this application bridges the gap between computational biology and accessibility, offering a scalable, open-source tool for the research community.