Exploratory Analysis of a Breast Cancer Gene Expression Dataset

Algorithm Programming in Science

Gonçalo Sousa

2023/2024

Dataset

The Breast_GSE45827 dataset, which can be found at www.kaggle.com, contains gene expression levels for 54676 genes (columns) from 151 samples (rows). This dataset contains 5 distinct forms of breast cancer (including healthy tissue) (column "type"). More information about this dataset, as well as other file formats such as TAB and ARFF, data visualisation, and classification and clustering benchmarks, is freely accessible at the official CuMiDa website, which can be found at http://sbcb.inf.ufrgs.br/cumida.

Aim

Creation of the project BCAnalizer, aimed at conducting an exploratory analysis through:

- General Dataset Information;
- Visualization of various graphs related to gene expression;
- Statistical calculation;
- Revealing the most relevant genes;
- Save the outputs in different files;
- Display a menu, containing the different functionalities that the program is able perform and to receive the input from the user.

BCAnalizer

The program is designed for comprehensive genomic data analysis, employing various Python modules including Pandas, Matplotlib, Seaborn, and Scikit-learn. Each section offers distinct functionalities enabling users to explore and analyse genomic datasets efficiently.

To initiate the program, the user must introduce the directory and name of the main file (breast_cancer_analysis.py). This leads to the presentation of the program and to the set of options of the interactive menu (Figure 1).

```
\breast-cancer-eda\src> python .\breast_cancer_analysis.py
----- Main Menu -----
1. Load data
2. Data information
3. Show classes and genes
4. Plot gene distribution by class
5. Compare gene expression across classes
6. Correlation matrix and Heatmap
7. Cluster visualization with PCA
8. Identify important genes
9. Save results
0. Exit
Choose an option (0-9):
```

Figure 1 – Interactive menu

The "Load Data" section enables users to load genomic data stored in CSV format. It utilises Pandas to read the CSV file, convert it into a Data Frame, and store it in the variable 'resultados'. This section is option 1 from the menu and is a mandatory step in the program (Figure 2).

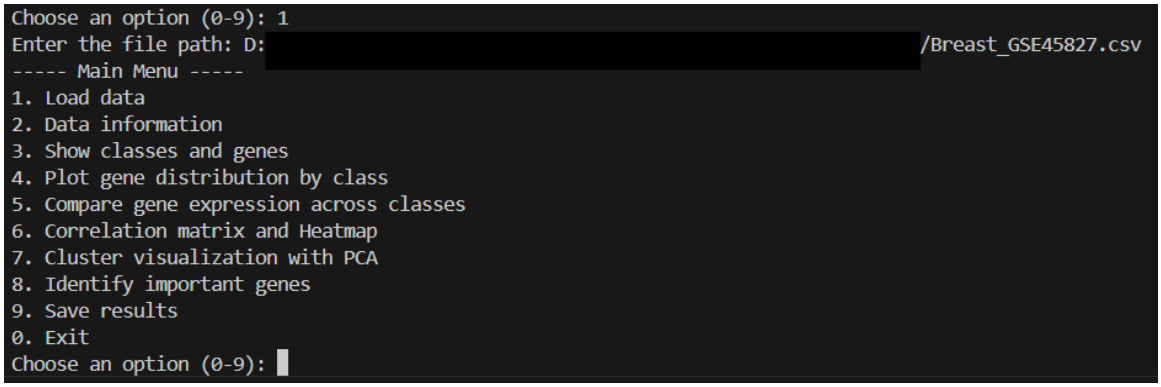

Figure 2 - Option 1 loads a file via its path file, and once loaded, the menu shows again, allowing you to pick a new option associated to the loaded file.

The "Data information" section (option 2) provides essential details about the loaded dataset, including column information, data types, and the identification of null values. Users are offered choices to handle null values if detected. (Figure 3)

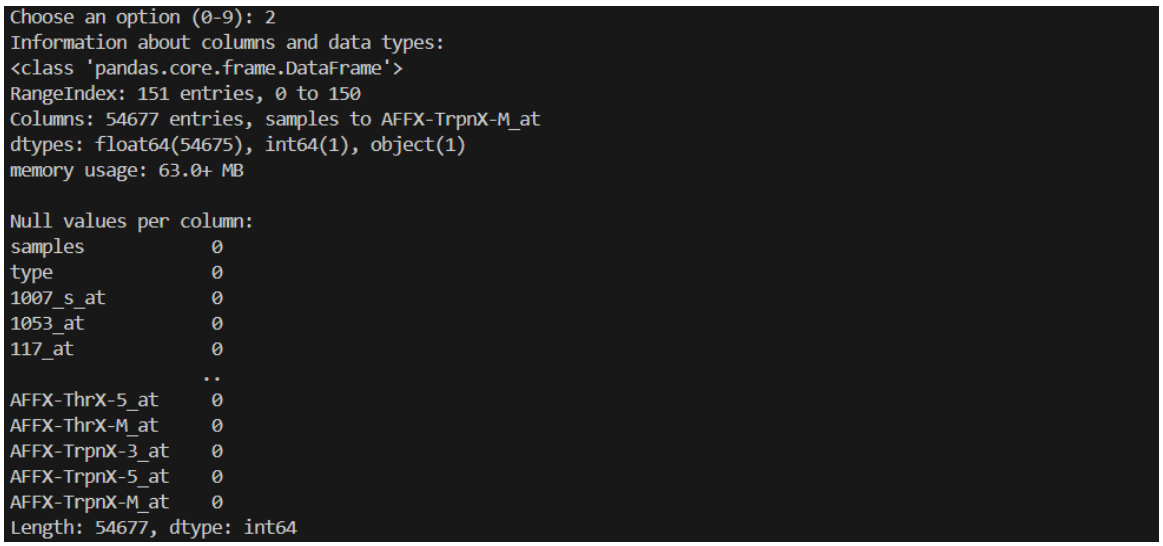

Figure 3 - Data information and Null values.

"Show classes and genes" (option 3) displays available cancer classes in the dataset and allows users to visualise matching genes for a certain class, providing a more in-depth understanding of the genetic composition of distinct cancer types. The user can select a class of interest to learn about the genes expressed in that class. Example in Figure 4.

```
Choose an option (0-9): 3
Cancer classes in the dataset:
['basal' 'HER' 'cell_line' 'normal' 'luminal_A' 'luminal_B']
Choose a cancer class to view the corresponding genes:
```

Figure 4 - Option 3 shows all classes in the variable 'type' of the dataset. Next, the user can choose the class of cancer of interest and observe the expressed genes contained in the class.

"Plot gene distribution by class" (option 4) and "Compare gene expression across classes" (option 5) utilize Matplotlib and Seaborn to plot histograms and scatter plots, respectively, facilitating a visual understanding of gene expression patterns across different classes (Figure 5 and 6).

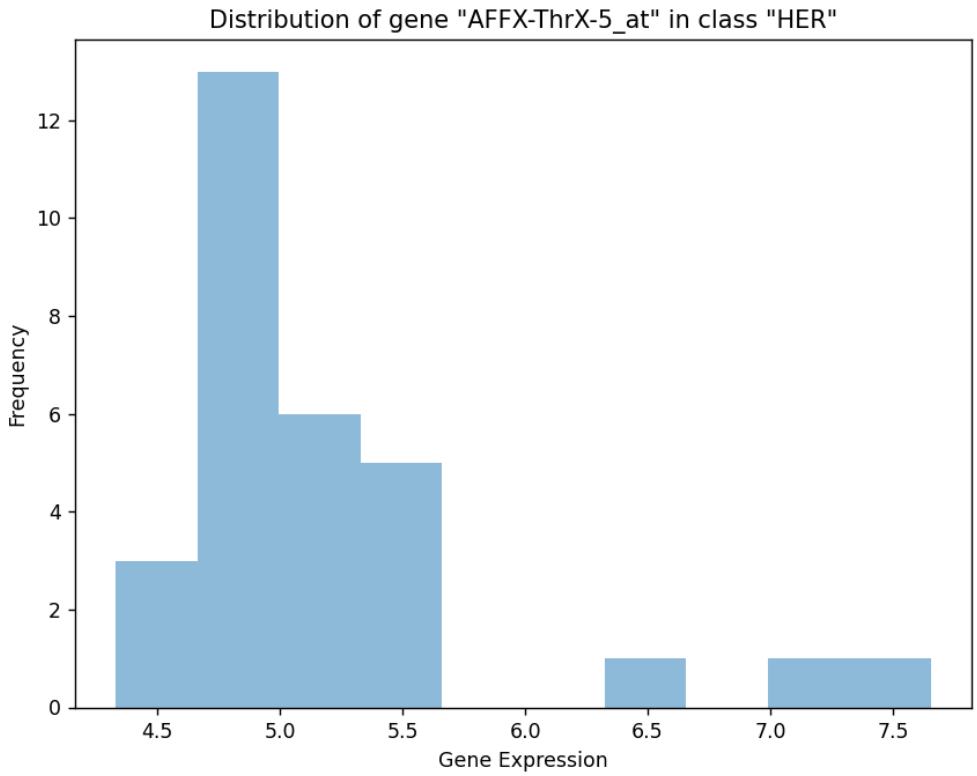

Figure 5 - Option 4: The user can choose any gene from the class of interest and see the distribution of the expression in the samples of the class. In this example, it shows the distribution of the gene AFFX-ThrX-5_at in the samples of class HER. The histogram has an option at the bottom in the form of a save icon to save the figure.

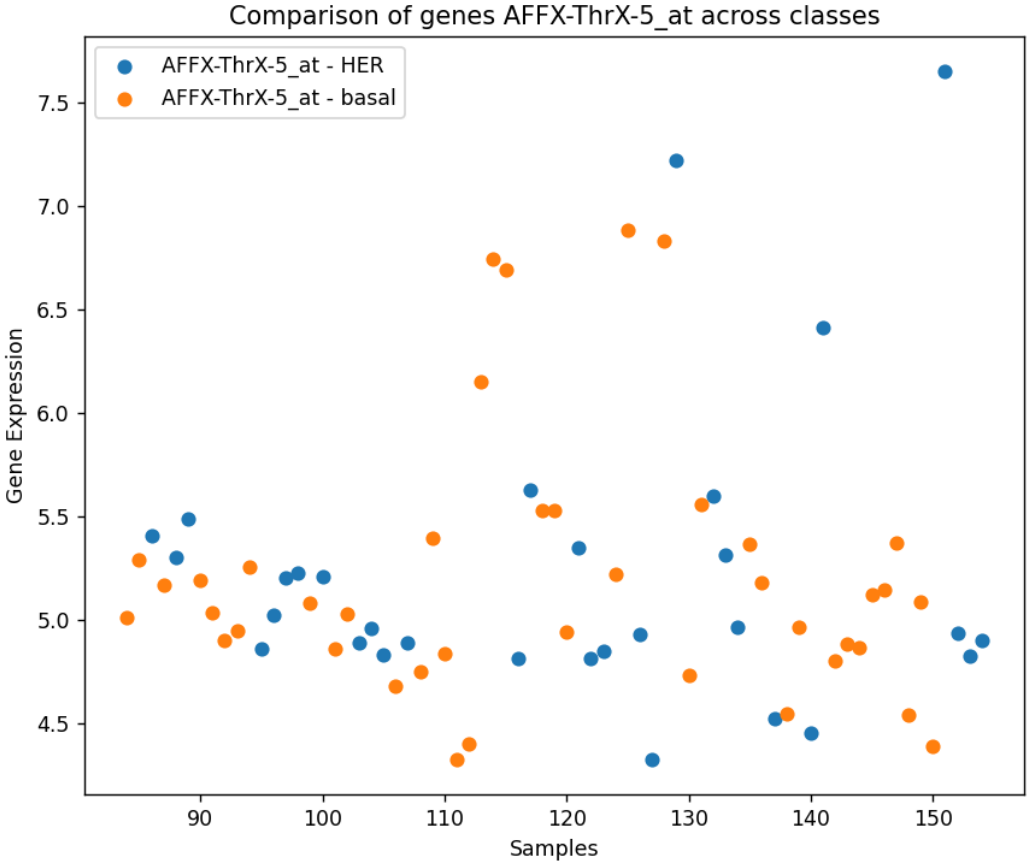

Figure 6 - Option 5: The user can choose one or more genes to understand if the expression of the genes is related or not in the same or another class. In this example, I chose gene AFFX-ThrX-5_at from two different classes: HER and basal. The distribution figure has a save icon at the bottom.

The " Correlation matrix and Heatmap" section (option 6) generates a correlation matrix among genes and visualizes it as a heatmap using Seaborn, aiding in the identification of correlations between gene expressions (Figure 7).

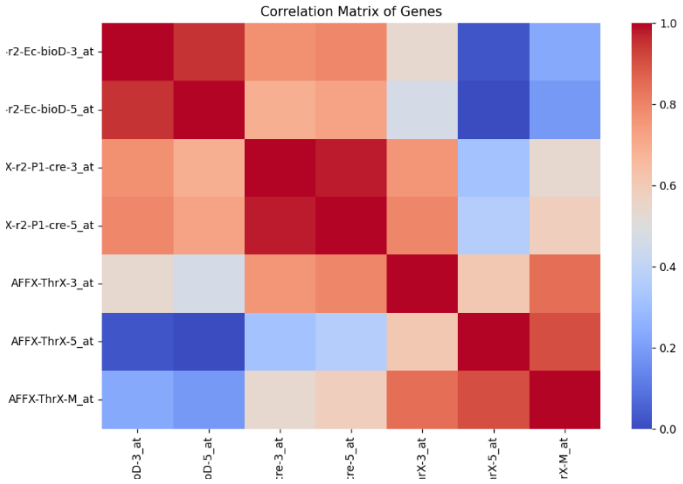

Figure 7 - Option 6: The user can choose all the genes or choose a set of classes to perform a matrix correlation of their expressions. Red is a positive correlation, blue is a negative correlation, and near-grey is considered a non-correlation. In this example, we chose this set of genes: AFFX-r2-Ec-bioD-3_at, AFFX-r2-Ec-bioD-5_at, AFFX-r2-P1-cre-3_at, AFFX-r2-P1-cre-5_at, AFFX-ThrX-3_at, AFFX-ThrX-5_at, and AFFX-ThrX-M_at. This figure has a save icon in the bottom.

" Cluster visualization with PCA" (option 7) performs dimensionality reduction using PCA and K-Means to visualize clusters of samples, providing insights into the distribution and grouping of data points (Figure 8).

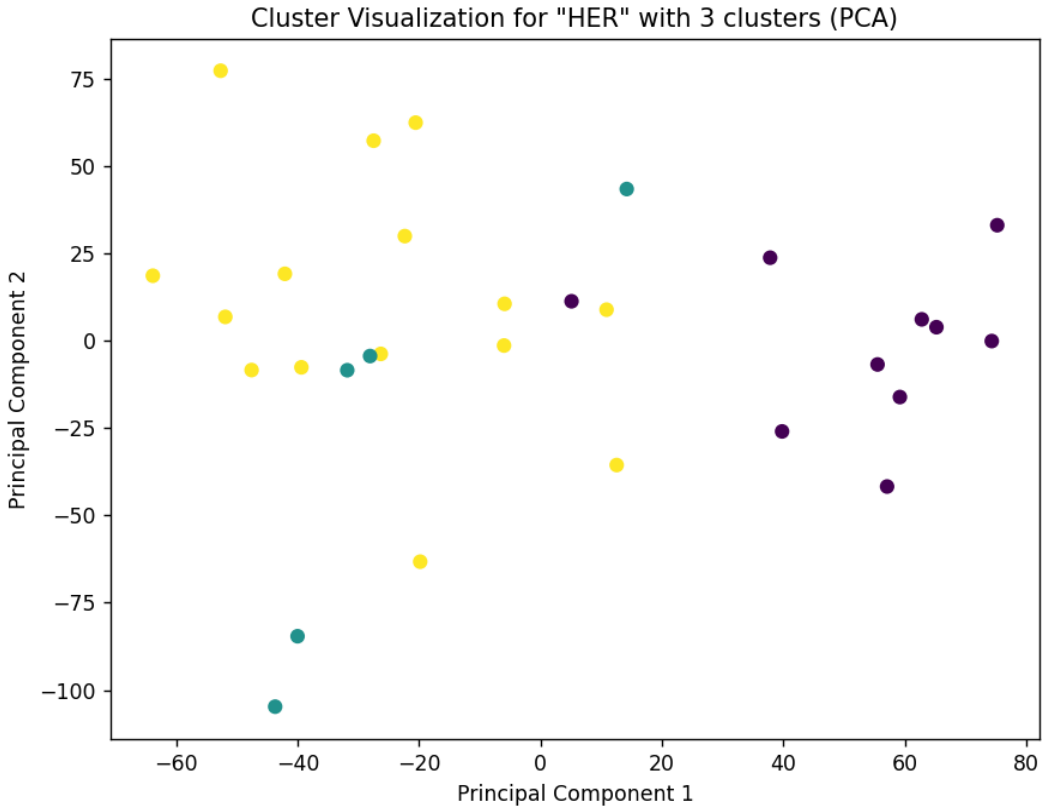

Figure 8 - Option 7: The user chooses the class of cancer and the number of clusters to perform a dimensionality reduction. In this example, I chose the class HER and three clusters. The figure has a save icon in the bottom.

" Identify important genes" (option 8) uses Scikit-learn's SelectKBest method to identify the most relevant genes for classifying specific cancer types, a crucial step in understanding the genetic factors contributing to different cancers (Figure 9).

```
Choose an option (0-9): 8
Available cancer classes:
['basal', 'HER', 'cell_line', 'normal', 'luminal_A', 'luminal_B']
Choose a cancer class for analysis: HER
Enter number of top genes to select: 10
C:\Users\gonza\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\feature_selection\_univariate_sel
ection.py:106: RuntimeWarning: invalid value encountered in divide
  msb = ssbn / float(dfbn)
Top 10 genes relevant for HER:
Index(['AFFX-r2-Ec-bioD-3_at', 'AFFX-r2-Ec-bioD-5_at', 'AFFX-r2-P1-cre-3_at',
       'AFFX-r2-P1-cre-5_at', 'AFFX-ThrX-3_at', 'AFFX-ThrX-5_at',
       'AFFX-ThrX-M_at', 'AFFX-TrpnX-3_at', 'AFFX-TrpnX-5_at',
       'AFFX-TrpnX-M_at'],
      dtype='object')
```

Figure 9 - Option 8: The user can choose the most expressed genes in the cancer class. The errors are linked to the genes that are not expressed. In this example, we want to know 10 genes more expressed in class HER.

Finally, the "Save Results" section (option 9) enables users to store obtained results, including dataframes, textual information, and graphs, in various file formats for future reference or further analysis. The program is closed when choose option 0.

Final Remarks

In conclusion, this program offers a comprehensive suite of functionalities for genomic data analysis. While its usability is contingent upon accurate data input and user interaction, its potential scientific contribution is significant, aiding researchers in exploring gene expressions across various cancer types and identifying essential genetic markers.