Classification of five types of cancer based on RNA-seq data using machine learning

21217588 [a]

[a] Faculty of Sciences, Aix-Marseille University, Marseille, France

Abstract

This report presents multiple machine learning approaches to classify cancer based on gene expression data obtained through RNA-seq. The gene expression data was used to train a machine learning model on a dataset consisting of various types of cancer. The results of this report suggest that machine learning can be an effective tool for classifying patients suffering from cancer by analyzing the gene expression matrix.

Keywords:

INTRODUCTION

In recent years, there has been increasing interest in using machine learning to classify cancer based on genomic data[1]. Gene expression data, obtained through techniques such as RNA-seq, provides a high-dimensional representation of the genes of each of these patients. This data can be used as input to machine learning algorithms to classify cancer. In this report, we present many machine learning approaches to classify cancer patients using an RNA-seq gene expression matrix. We describe the dataset and the machine learning model used, as well as the results of the classification process. Overall, our findings demonstrate the potential of using machine learning on gene expression data for the purpose of cancer classification.

MATERIALS AND METHODS

Dataset

The dataset is multivariate and based on real-life measurements. It is made from the expression of 20531 genes of 801 patients having one of five different types of cancer. There are two files, one of them is "data.csv" which contains the gene expression matrix of the patients, and the other one is "labels.csv" which contains the classification of each patient. The approach in our study will be "supervised" since the data is already annotated with acronyms (table 1) . The data is available here .

Acronym	Explanation
COAD	Colon Adenocarcinoma
KIRC	Kidney Renal Clear Cell Carcinoma
BRCA	Breast Cancer
LUAD	Lung Adenocarcinoma
PRAD	Prostate Adenocarcinoma

Table 1: types of cancer in the dataset

Feature selection

Feature selection is the process of identifying a subset of the most relevant features in a dataset for use in modeling. It can help to identify the most important features, reduce computation and make it easier to understand how the model is making predictions.

Dimensionality reduction

Since the data contains a lot of genes measured, it's necessary to reduce its dimensionality for different reasons: As the number of dimensions increases, the amount of data required to adequately sample the space increases exponentially. This can lead to situations where a model may be unable to learn anything useful from the data, a phenomenon known as the curse of dimensionality. In our first approach we will use PCA to reduce the dimensions, while on the second approach we will use sklearn's SelectPercentile based on a chi2 test to select a certain number of features rather than taking them all.

Choice of model

Using Scikit-learn[2], the main analysis of this report will be based on multi linear regression model. However, we also tested two other models which are SVM and Naïve Bayes. We will also test for different machine learning parameters including the number of features, epochs, and different metrics of evaluation…

Python and libraries

The bash scripts to set the environement are available inside the README.md. A conda environment has been set and it uses Python 3.9 and the following packages: matplotlib (3.3.4), – numpy (1.22.4), pandas (1.5.0), seaborn (0.11.1), sklearn(1.2.0) and umap (0.5.3). The results are available in a jupyter notebook file. There is also a python script but it's recommended to run jupyter notebook to understand what is happening step-by-step.

RESULTS

Data exploration

First of all, from the two mentioned files we will establish two dataframes. The first one containing the gene expression data, for each gene one column, and for each sample one row. The second dataframe contains the label of the samples. It would be interesting to generate some statistics around the distribution of the samples in the data, to check if some labels of the samples are over or under-presented (figure 1). In order to understand our data and comprehend how close the classes really are, we also made a UMAP[3] on the whole data during this phase and it visually showed that the classes are well separable from each other (Figure 2).

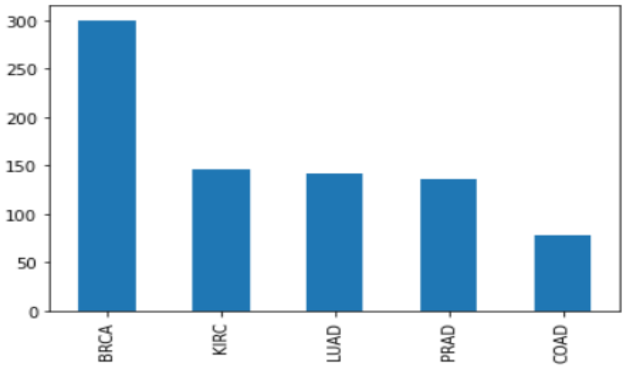


Figure 1: Distribution of the samples in the data

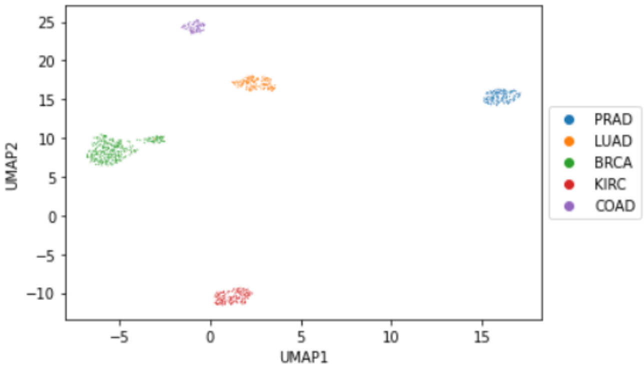


Figure 2: UMAP dimension reduction on the data

Because of the imbalance of the samples, we should verify that in our train/test split the proportion of samples are respected. The data will be split in 70% train (+ cross-validation) and 30% test.

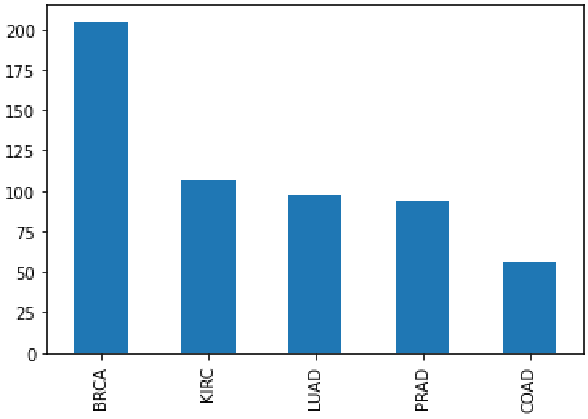


Figure 3: Distribution of the labels of the train samples

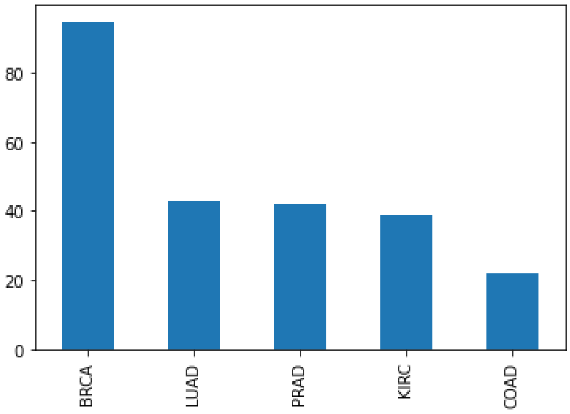
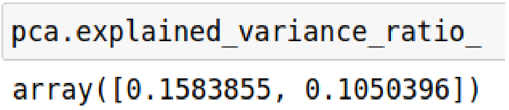


Figure 4: Distribution of the labels of the test samples

Based on figure 3 and 4, the distribution of the labels is preserved after the data split by using stratify=y.

PCA for dimensionality reduction

In our first approach, we will describe the application of PCA[4] over the dataset. This step should reduce the dimensions to only two (n=2) and it understands the directions of the spread of our data using Eigenvectors. By applying PCA, two newly created dimensions will try to explain the variances of the data. The obtained results (figure 5) show that that these two dimensions will explain a total of ~26% of the variance, which is a low number.



```
pca.explained_variance_ratio_

array([0.1583855, 0.1050396])
```

Figure 5: PCA explained variance

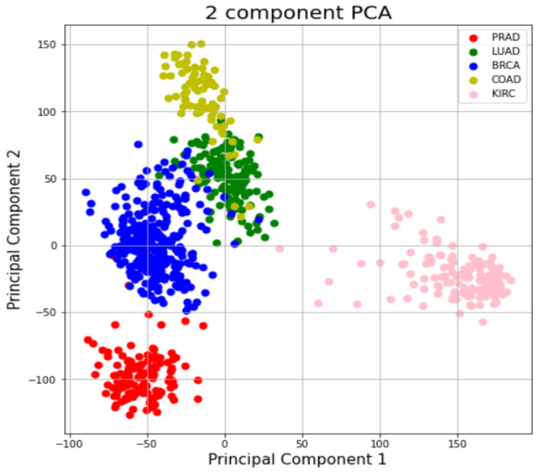
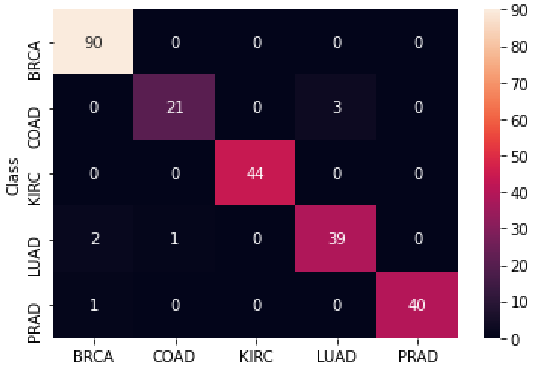


Figure 6: Plot of the PCA dimensions of the samples

The figure 6 plot shows that the PCA segregates well KIRC and PRAD from the rest, but there is a certain mixture between 3 classes BRCA, COAD and LUAD. This means that there is a certain confusion between the classes

This model has an accuracy of 0.97 and a recall score of 0.92. We see that the false predictions are between the three classes of LUAD, COAD and BRCA (figure 7). That aligns well with the PCA plot we saw in figure 6. Therefore, it's possible to assume that the first two principal components don't explain the variances between these three classes which might yield some errors in the classification. Nevertheless, the accuracy is still very high.

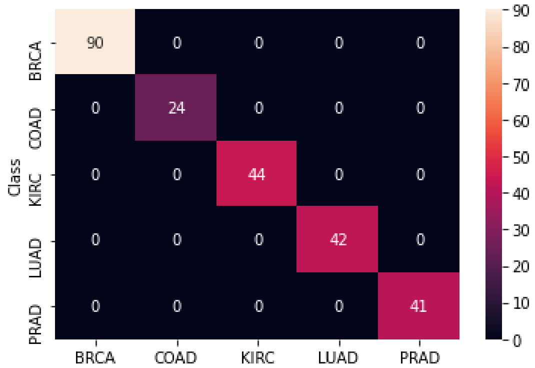


Class	BRCA	COAD	KIRC	LUAD	PRAD
BRCA	90	0	0	0	0
COAD	0	21	0	3	0
KIRC	0	0	44	0	0
LUAD	2	1	0	39	0
PRAD	1	0	0	0	40

Figure 7: Cross tab matrix between predicted and true labels

Second approach

Also using multinomial linear regression, but this time the original gene expression data matrix will be reduced differently. We will use scikit's SelectPercentile method employing the chi2 test to fetch out the top 5% of genes whose scores are the highest in the chi-squared stats of non-negative features for classification tasks. This way we will be left with 1027 genes instead of the initial 20000+. We will set a regulator C=1e40 to avoid overfitting.



Class	BRCA	COAD	KIRC	LUAD	PRAD
BRCA	90	0	0	0	0
COAD	0	24	0	0	0
KIRC	0	0	44	0	0
LUAD	0	0	0	42	0
PRAD	0	0	0	0	41

Figure 8: Cross tab matrix after selecting only top 5% genes

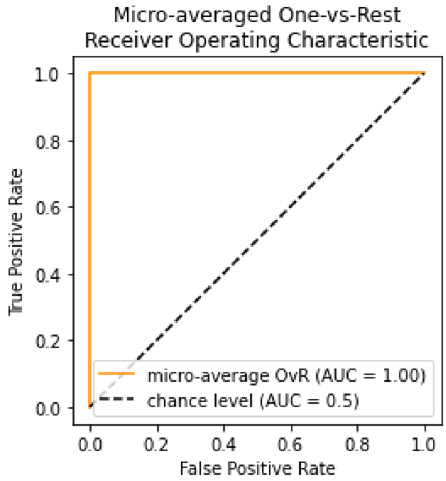


Figure 9: ROC Curve and AUC

This model has an accuracy, recall and F1-score all equal to 1. The ROC tested (figure 9) considers one against all approach and then calculates a micro average The area under the curve is also equal to 1. The same model has tested with cross-validation (the training set has been split into 10 chunks), and the results are that the accuracy is overall very high for all the sets. What is really important is that the accuracy of each chunk of cross-validation is as high as the training and testing which helps us avoid problems of overfitting.

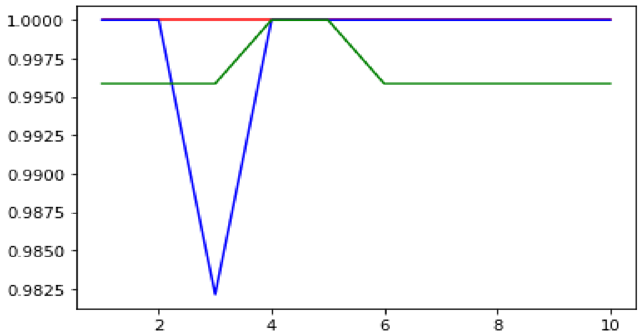


Figure 10: Accuracy plots (blue cross-validation, red training, and green testing)

Factors that improve the model

In this section we will go through some of the elements that affect the accuracy of the model with the purpose of testing what has been taught as a general fact about machine learning algorithm. First of all, we will being with the number of epochs (figure 11). As we can see, on extremely low number of epochs, the model fails to retain any important feature of the dataset, hence the accuracy is stuck around 0.5. It needs at least 5 epochs in order for it to begin hitting an accuracy close to 1.
Second, it's the number of features that are used to train the model. When the features are reduced to a very small number, the reduced dimension might fail to capture the specific genes or features that mostly describe the differences between the samples. Therefore, the model will be trained on unrepresentative data, therefore the predictions will be inaccurate. Such case is observed when only the first 10 genes are taken into account (figure 12), as the number of features decreases, so does the accuracy. The accuracy of this model is 0.53.
Also, other choices of models were tested. For instance, **SVM** (accuracy 0.99) and **Naïve Bayes** (accuracy 0.84). More infromation can be found inside the jupyter notebook file.

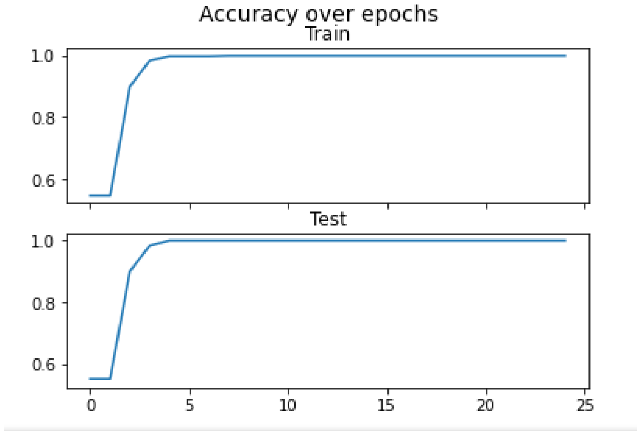


Figure 11: The evolution of accuracy for Train and Test with the number of epochs

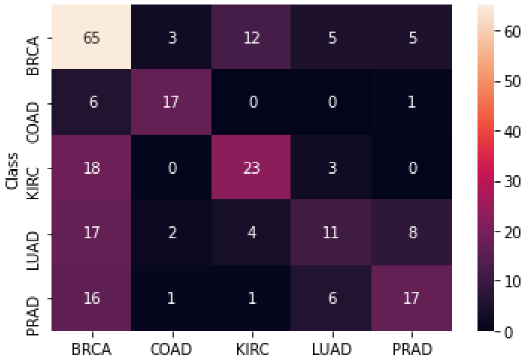


Figure 12: Cross tab of a model trained only on 10 genes

DISCUSSION AND CONCLUSION

In this report we went through many aspects of Machine Learning in order to make a categorical classification of different types of cancer. The data we treated differentiates well between the different types of samples, and that's why we see accuracy scores close to 1 in each run. This made our job easier in selecting the types of our models. We saw two different dimensionality reduction techniques being applied (UMAP, PCA). We also showcased the usage of cross-validation, number of epochs and number of features.
Finally, this dataset is "exceptional" in its quality and clarity of separating between individuals, usually for more complex problems like image and multi-omics issues it's not as easy to find a suitable model that functions well.

REFERENCES

[1] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and Structural Biotechnology Journal*, vol. 13, pp. 8–17, Jan. 2015, doi: 10.1016/j.csbj.2014.11.005.

[2] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.

[3] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction." arXiv, Sep. 17, 2020. doi: 10.48550/arXiv.1802.03426.

[4] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, Mar. 1993, doi: 10.1016/0098-3004(93)90090-R.