Bioinformatics Project Report

Active Regulatory Regions Prediction Using ML

Emir Erdogdu - Matriculation number: V08870 - emir.erdogdu@studenti.unimi.it

Report

1. Introduction

The complex process by which a cell converts the data in its DNA into molecules that carry out the necessary metabolic and structural functions is known as gene expression. Enhancers and promoters are examples of cis-regulatory non-coding DNA regions that control how genes are expressed. Promoters are groups of nucleotides that are close to a gene and serve as the transcription's first beginning point. Enhancers are DNA sequences that aid in triggering a gene's transcription but are not the gene itself (this makes more difficult the identification of these regions).

Two tasks were performed on the data:

- Active Enhancers versus Inactive Enhancers
- Active Promoters versus Inactive Promoters

After the data is generated, the task is to create learning models such as FFNN, CNN and MMNN to see if the performance increase can be obtained.

2. Models

2.1 Feed Forward Neural Network

The basic deep learning models are deep feedforward networks, commonly known as feedforward neural networks or multilayer perceptrons (MLPs). A feedforward network's objective is to simulate some function f*. For instance, y = f*(x) transfers an input x to a category y for a classifier. A feedforward network establishes the mapping y = f(x;θ) and discovers the value of that yields the best function approximation.

2.2 Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take in an input image, give various elements and objects in the image importance (learnable weights and biases), and be able to distinguish between them. Comparatively speaking, a CNN requires substantially less pre-processing than other classification techniques. CNN have the capacity to learn these filters and properties, whereas in primitive techniques filters are hand-engineered.

2.3 Multi-Modal Neural Network

A multi-modal neural network (MMNN) is essentially a neural network that incorporates information from several modalities, in this case DNA sequence and epigenomic data. It accomplishes the task by combining and truncating the output layers of two concurrent neural networks (in this case, an MLP and a CNN). Since this particular network makes use of both information sources, we can anticipate more trustworthy outcomes.

2.4 Hyper-Parameters

Multilayer perceptrons have multiple parameters that are not learned from the data thats being classified. These parameters are there to fine-tune the algorithm thats being used.

Some of the hyper-parameters are:

- Dropout Rate: Rate of discard to avoid overfitting
- Decay: Modifies learning rate over time
- Network depth: Number of layers

3. Experimental Setup

3.1 Datasets

In this project the data from ENCODE and FANTOM Project are used. Former dataset is developed by US National Human Genıme Research Institute for researching functional element identification in human genome , and the latter is led by RIKEN, focused on mammalian genomes.

Retrieving data for the project was done by using epigenomic_dataset package by Luca Cappelletti, to automate the downloading and importing process.

3.2 Data pre-proccessing

Label Binarization

Binarization of the labels are needed to start the classification tasks. FANTOM suggests to use 1 TPM for both promoters and enhancers. In the figure below, the distribution and class counts can be seen.

KNN

There are a number of methods to substitute NaN values. The algorithm chosen is nearest neighbors imputation, where each missing feature is imputed using values from the feature's k nearest neighbors. In this project, the hyperparameter K is set to 1, and the neighbors' features are averaged equally.

Z-Scoring

After accomplishing all the pre-proccesing steps, Robust Scaler is used to remove the medians and scale it to the according quantile range.

Feature Selection

In many machine learning pipelines, features selection is a critical step. In this project, we want to choose only the most pertinent features out of the entire collection. Boruta is used in this project.

The importance measure of an attribute is obtained as the loss of accuracy of classification caused by the random permutation of attribute values between objects. Then the average and standard deviation of the accuracy loss are computed. Alternatively, the Z score computed by dividing the average loss by its standard deviation can be used as the importance measure, unfortunately the Z score is not directly related to the statistical significance of the feature importance returned by the random forest algorithm.

3.3 Data Correlation

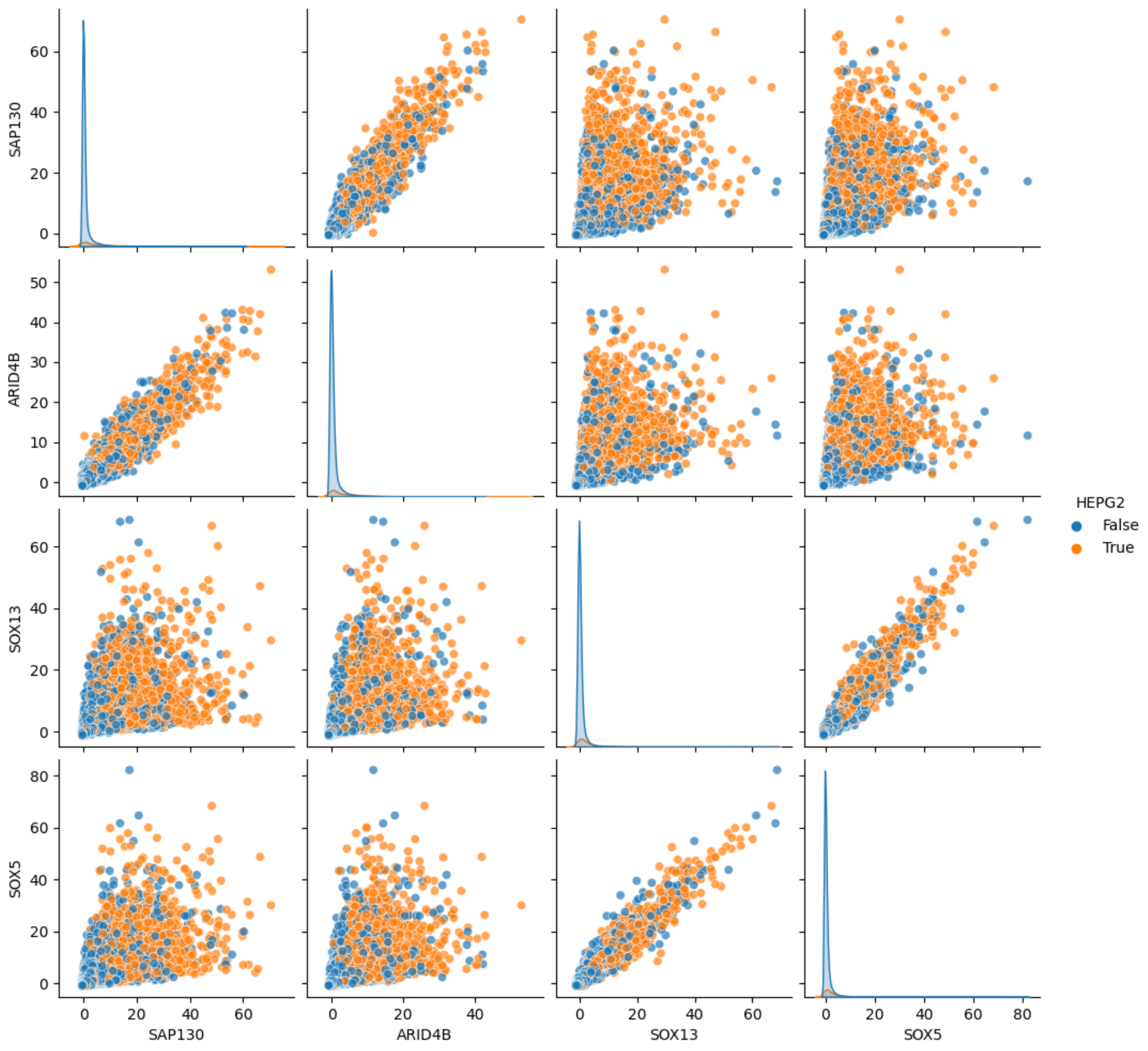
After processed data is obtained verification between the correlations of features and the outputs.

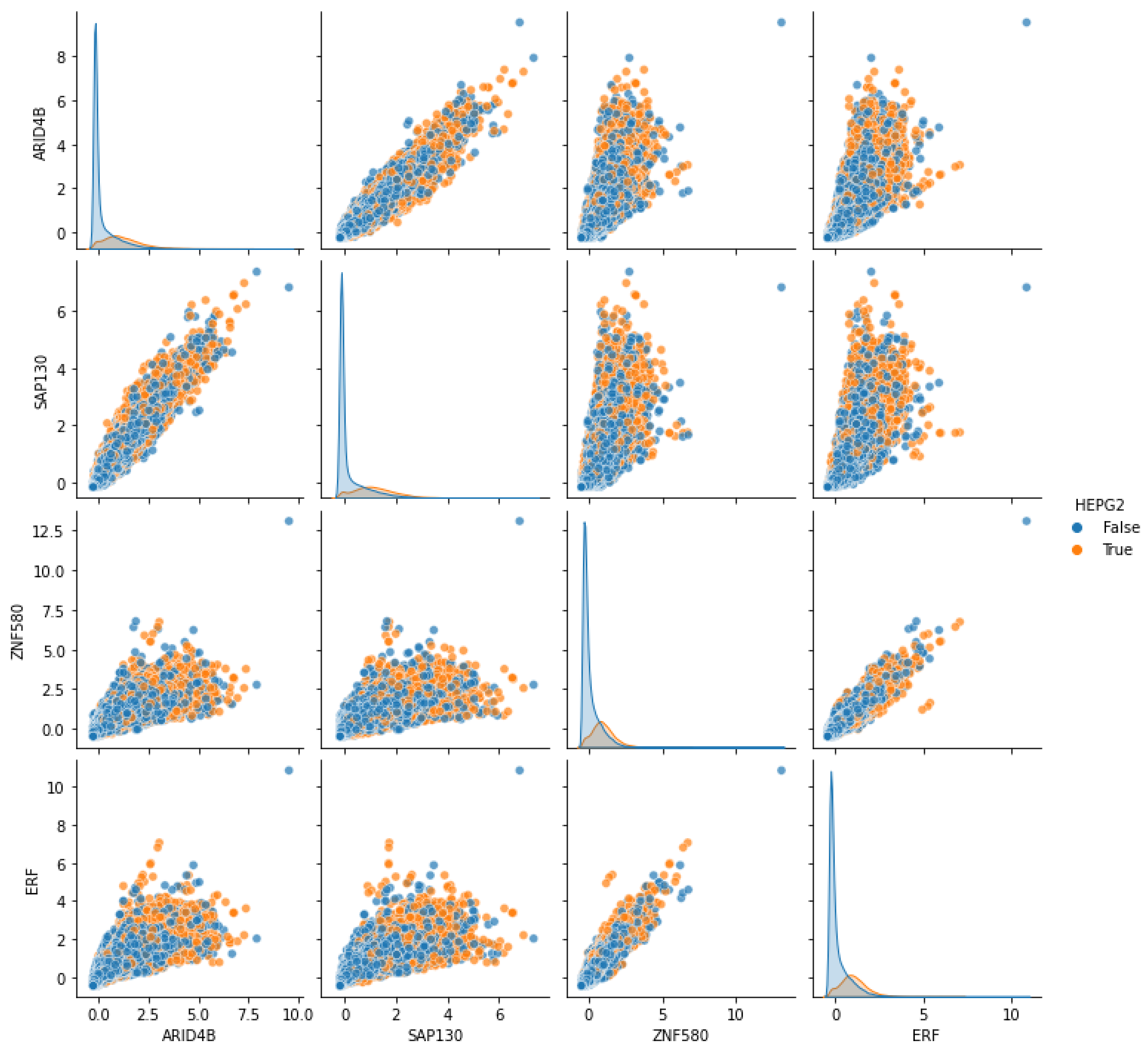
A statistical measure known as correlation explains the relationship between two variables by describing how one variable responds to changes in the other. Pearson and Spearman correlation coefficients were the two that's utilized in this project.

- • Pearson coefficient measures the linear correlation between to variables (it has a value between +1 and -1, with +1 total positive linear correlation, 0 no linear correlation and -1 negative linear correlation)
- • Spearman correlation occurs when each of the variables is a perfect monotone function of the other

The features ARNT, NBN and ZNF737 are deemed irrelevant to the output for promoters for Pearson, while the features SNRNP70 and ZNF382 are deemed uncorrelated for Spearman. Both Pearson's test and Spearman's test indicate 2 unrelated features with the output. I then went on to remove the features that weren't connected to the output.

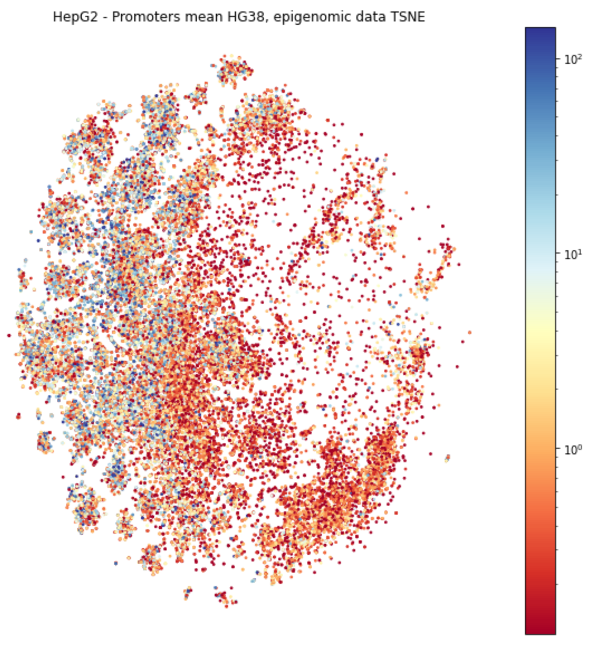
In the figures below, the visualization of most correlated features for promoters and enhancers are given.

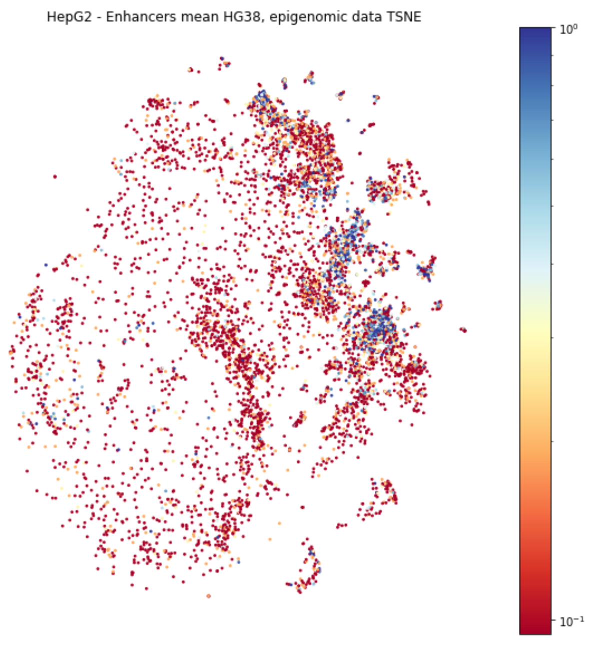

SAP130
ARID4B
SOX13
SOX5
HEPG2
False
True

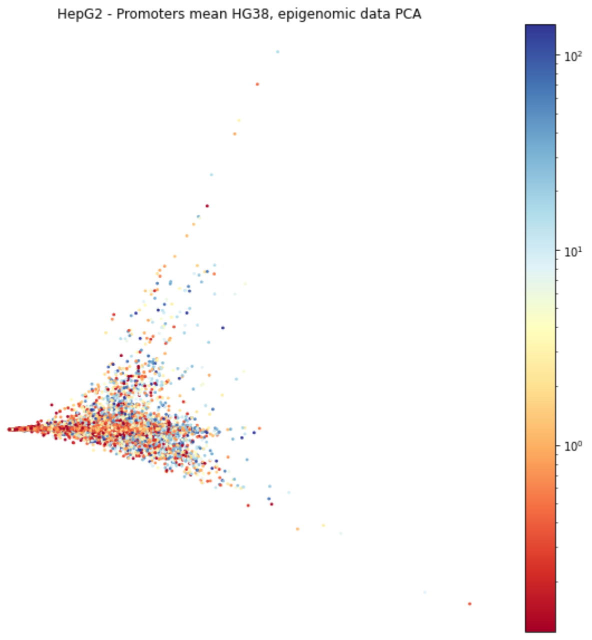


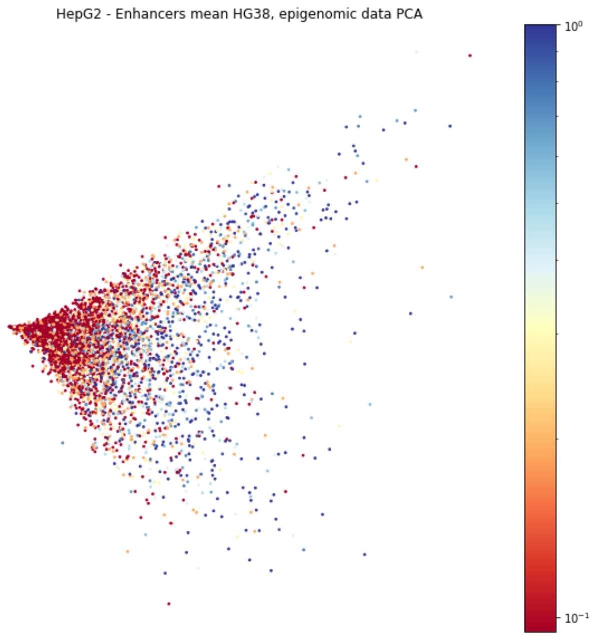
3.4 Data Visualization

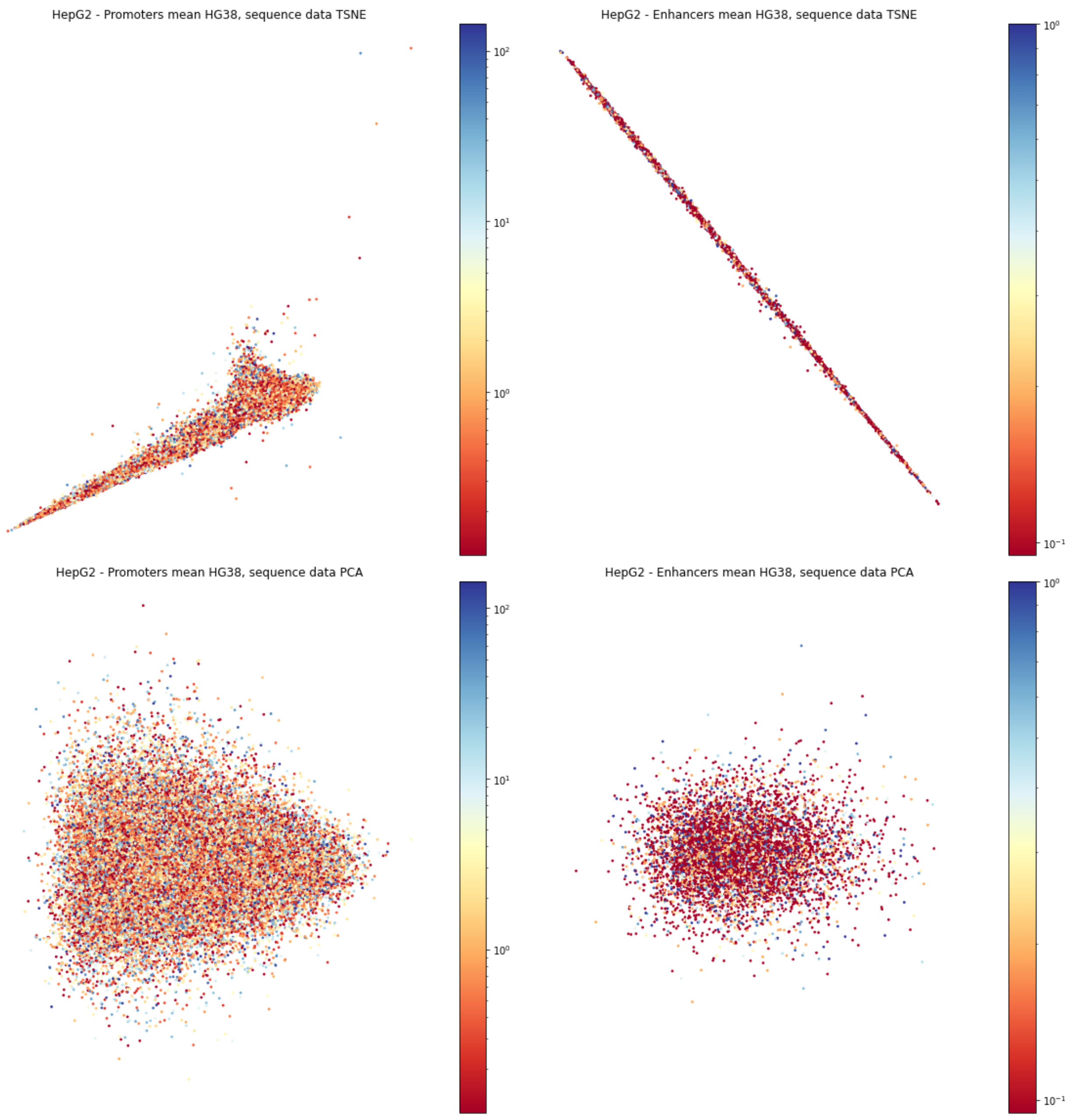
To find and understand the distribution of the data, PCA is used. PCA decomposition is a technique to identify a smaller number of uncorrelated variables known as principal components of a larger dataset.


HepG2 - Promoters mean HG38, epigenomic data TSNE
10^2
10^1
10^0


HepG2 - Enhancers mean HG38, epigenomic data TSNE
10^0
10^{-1}


HepG2 - Promoters mean HG38, epigenomic data PCA
10^2
10^1
10^0


HepG2 - Enhancers mean HG38, epigenomic data PCA
10^0
10^{-1}



By PCA decomposition, we can assume that the sequential data that is collected is not separable. With the help of Figure 8, we can assume the classification of the data is not easy, which in conclusion, makes the performance

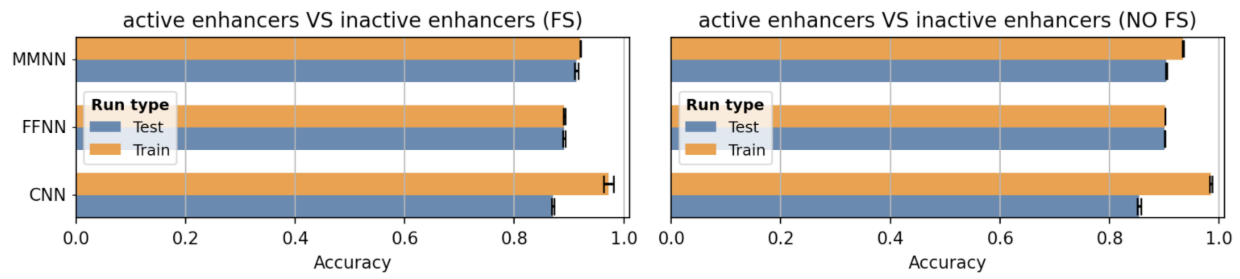
3.5 Holdouts

All the tasks that are performed by the model are executed using 20 stratified holdouts. Every holdout uses 80% of the samples for the training and the 20% of the samples for the test part.

4. Results

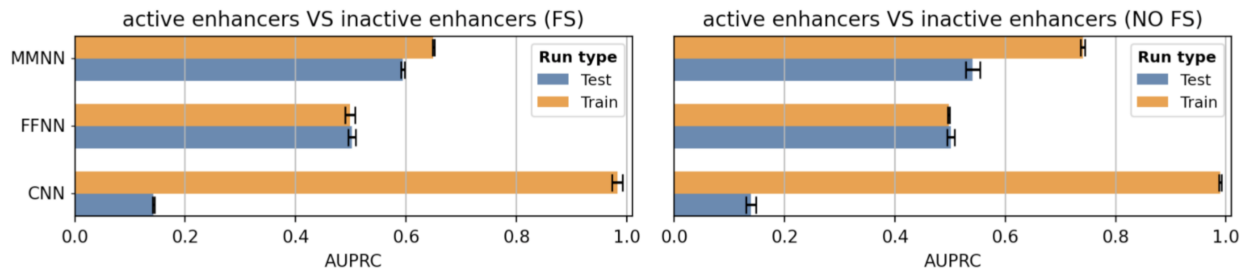
Three different metrics were considered to understand the performance of the trained models. Respectively, AUROC, AUPRC and accuracy were used.

By looking at the accuracy values below, it can be said that all of the models used have high accuracies. But the accuracy values does not give the full story all the time.

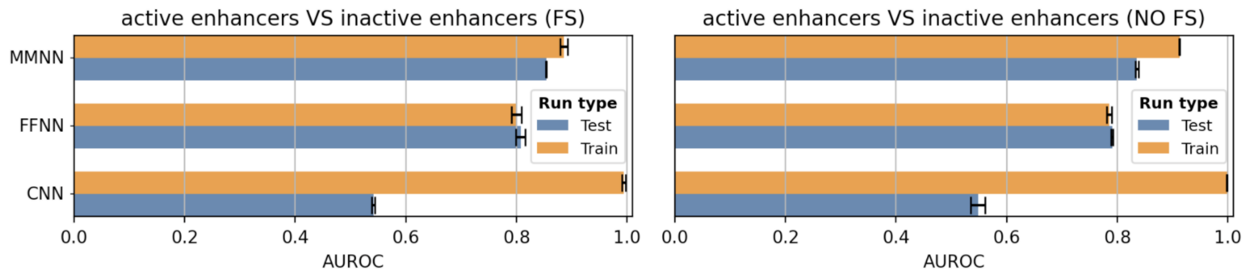


The AUROC and AUPRC metrics tell a different story, unlike the accuracies that we observed.

In AUPRC we can definitely see that CNN results have very low accuracy results.



And lastly in AUROC, we can see, all of our models except CNN performed well enough.



In general, we can see that the MMNN and FFNN models performs better than CNN.

Resources

- https://github.com/LucaCappelletti94/bioinformatics_practice
- https://github.com/AnacletoLAB/epigenomic_dataset

- https://github.com/LucaCappelletti94/ucsc_genomes_downloader
- https://github.com/scikit-learn-contrib/boruta_py