Università degli Studi di Torino - M.Sc. in Stochastic and Data Science - A.Y. 2021/2022

Final project of Statistical Machine Learning (MAT0043)

Gene selection for cancer type classification

The purpose of our project is to work on a high-dimensional genomics data and find a relatively small number of genes to predict the cancer type of a given tumorous cells. This is known as Genes Selection for Cancer Classification and it is in line with many up-to-date problems of applied medicine.
Our dataset contained 1.032 cancerous cells and their knock-out probabilities, i.e. the probabilities of stopping the growth of tumour by inhibiting one of the ~ 17.000 genes. Each cell line was characterised by one of 10 possible cancer labels: Eye, Gastrointestinal, Gynecologic, Musculoskeletal, Neurological, Breast, Head-Neck, Hematologic, Genitourinary and, finally, Lung. We explored in details three Features Selection algorithms: Random Forests (RF) combined with Feature Importance, Lasso-SVM and Neural Networks (NN) combined with Permutation Importance. We studied two binary classifications (Blood-cancer vs. All, Lung-cancer vs. All) and a multiclass classification. Besides lung models, we achieved satisfying classification accuracies and we were able to select about 100-200 genes from the initial ~ 17.000 ones. Models fitted on such genes obtained classification accuracies ranging from 67% to 98%.
Therefore, it seems that classifying cancer type from an extremely small set of genes depends on the cancer type itself. These methodologies worked incredibly well on Blood cancer, as we reached almost 100% accuracy with the reduced classifier, whereas failed miserably on Lung cancer.

1 Introduction

Cancer is a complex disease characterized by the uncontrolled growth of abnormal cells anywhere in the body. These abnormal cells are extremely invasive and we usually identify them with the name of their original tissue (for instance, breast cancer, lung cancer, brain cancer, etc.). In recent years, medicine has made a great step forward in finding new and efficient therapies for different diseases and cancer is one of them. In particular, thanks to numerous advances in technology, collecting huge amount of data is no longer an issue, so that one can exploit such information to define personalized treatments for patients. In this regards, the DepMap project[1] and, in particular, the Achilles project[2] aim to use genome-wide screens to collect data regarding mutations of cancerous cells, identify essential genes and report vulnerabilities across hundreds of human cancers.

Many researches are currently using DepMap datasets to identify relatively small sets of genes which are responsible of cancers growth. This procedure is often driven by medical knowledge, which we do not possess, together with some rough measures of importance. Being Maths student, we instead base our research on statistical models and on the hypothesis that, if a given classifier is able to distinguish different types of cancer, then the most relevant genes are the most important features for that given classifier (the meaning of "important features" will be clarified later). Of course, selecting few truly significant genes has outstanding implications in the medical field: building faster diagnosis tools and synthesising less toxic drugs are only two examples.

[1] DepMap Portal: https://depmap.org/portal/
[2] Achilles Project: https://depmap.org/portal/achilles/

2 Dataset

We used two public datasets from the DepMap Portal website[3] :

D1 *CRISPR_gene_dependency.csv*, which contains 1.032 cancer cell lines characterised by 17.393 gene scoring results

D2 *sample_info.csv*, which contains cell lines information, such as primary disease and sample collection site

Data were collected from real patients and successively processed, so that element (i, j) of this (1.032×17.393)-data frame is the probability that knocking out gene j has a real depletion effect on the i-th cell. Before proceeding with our analysis, we removed missing values, which affected only 10 rows coming from different tumours, and we looked for weird observation. In particular, we found 2 "Non-Cancerous" and 6 "Engineered" cells. The first can be reasonably discarded, whereas the latter requires a little care. Engineered cells are synthetically modified samples in lab and, here, they are manly associated to the Eye sample collection site. We decided to keep them and associate them to the cancer corresponding to their site.
Side note: in general, looking also at the sample collection site do not ease tumour classification. Indeed, metastasis of the original compromised tissue can be found all over the body.

We hence grouped the various cancer types in 10 classes according to common medical knowledge[4] and we obtained classes as reported in Figure 1. "Eye" is the smallest one as there are only 16 observations, 5 of which labelled as "Engineereed". On the other hand, "Gastrointestinal" is the largest group and it comprehends 7 types of cancer, making this group quite heterogeneous. We decide to investigate two binary classification problems, Blood vs Rest and Lung vs Rest, and the multiclass problem. We chose Lung because of the nature of such class: it is the most numerous group composed only by Lung cancer samples. The choice of Blood was instead driven by some underlying biological knowledge. In fact, Blood cancer is quite different from other tumours because

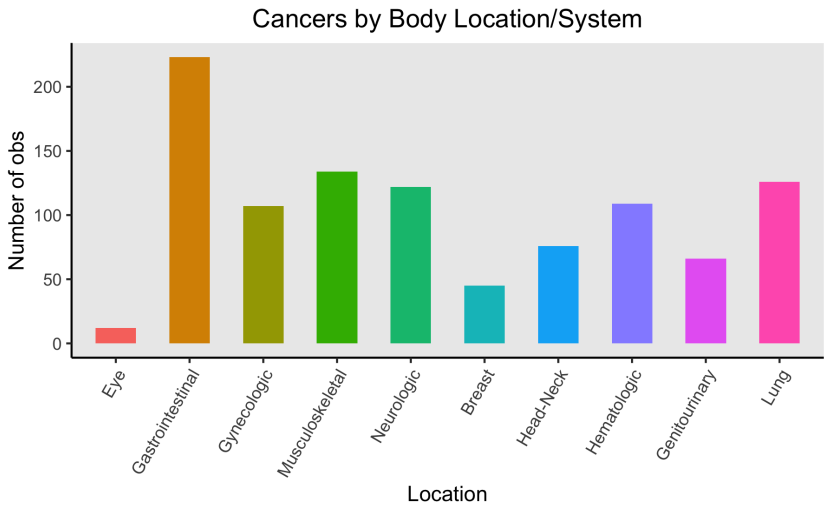

Figure 1: Cancer classes

- blood is in the whole body
- Leukemia, Lymphom and Myeloma are the main kinds of cancer but they all affect white blood cells
- not all blood cancers require a treatment, just periodical monitoring

[3]Download dataset from DepMap: https://depmap.org/portal/download/:
[4]Cancer types grouped by body location: https://www.cancer.gov/types/by-body-location

3 Methods

3.1 Random Forest

Whenever we are only interested in model performance and not in interpretability, Random Forest (RF) is a valid starting point, often used as "baseline" for benchmarking better methods. Being an ensemble of decision trees, a RF works by learning hierarchical if/else questions. It frequently performs well on imbalanced data and it is a good compromise when working with correlated high-dimensional data. Indeed, each decision tree classifiers of the RF is fitted only on a subset of the p features, generally $\sqrt{p}$ of them, and averaging is used to improve predictive accuracy.
Moreover, each tree in the RF learns from a random sample which is drawn with replacement and it can be shown that on average not all observation are used to fit the tree. These remaining observations are known as out-of-bag observation and are used to quickly estimate the generalization accuracy.
Finally, RFs are a simple tool to detect the most important features. Prediction accuracy are first measured on the out-of-bag samples. Then, the values of the variable in the out-of-bag-sample are randomly shuffled, keeping all other variables the same. Finally, the decrease in prediction accuracy on the shuffled data is measured and the mean decrease in accuracy across all trees is reported. Intuitively, the random shuffling means that, on average, the shuffled variable has no predictive power. This importance is a measure of by how much removing a variable decreases accuracy, and vice versa — by how much including a variable increases accuracy.
We hence selected the most important feature in two different ways:

- *Cross-validation*. We perfomed a 5-fold cross-validation on the model, selected the top most important features of each model and finally averaged.
- *Boruta algorithm*: Boruta repeatedly measures feature importance then carries out statistical tests to screen out the features which are irrelevant. Here the steps:
    1. Create copies of all the original features present in your dataset.
    2. Add randomness by shuffling these new feature. This is done to ensure that these new features show no correlation with the response variable.
    3. Run random forest classifier on the extended dataset and generate feature importance score based on mean accuracy decrease estimate for all the shadow variables.
    4. Compare Z-score of original variable with the maximum Z-score of shadow variables. Real features that have low score compared to the best of shadow features are deemed unimportant.
    5. Remove shadow features and repeat the process until an importance score is generated for all the variables.

3.2 SVM-Lasso

3.3 Neural Networks

Nowadays, Neural Networks are a very attractive approach to obtain excellent performances.

4 Results

4.1 Binary classifications

4.2 Multiclass classification

5 Conclusion and future works