Computational Health Laboratory

Simone Baccile, Edoardo Federici, Federico Silvestri

June 2022

1 Introduction

This report describes our work on the assigned project 7 for the Computational Health Laboratory course (academic year 2021/2022).

Project Name Data Integration in metabolic diseases.
Project Description Study the differences between various metabolic diseases starting from multiple GEO datasets and combining biomarkers. Use classification and feature selection to extract biomarkers and classify patients in various disease classes.

2 Phase 1

The first phase of the project consisted of an exploratory analysis of metabolic diseases followed by a standard preprocessing pipeline to clean, standardize, and merge data from various sources.

2.1 Exploratory analysis

Our first approach with the project was to search various metabolic diseases and how much data were available for each one. This analysis can be resumed in the following steps:

1. Scraping of different sites to obtain a long list of both common and rare ones, which yielded about 500 diseases.
2. Querying the Gene Expression Omnibus (GEO) repository of the National Library of Medicine (NLM) searching for datasets with patients affected by those diseases. This resulted in 75 diseases found, with 150 different datasets.
3. Macro analysis of those datasets to get general information like how many patients from each dataset are afflicted by metabolic diseases, the platform used to measure gene expression levels, and the type of the experiment. After this analysis, we removed all diseases with too few patients, keeping a total of 43 diseases.
4. Since the experiment performed to obtain data was radically different among the various GEO datasets we found, we decided to keep only expression profiling experiments. In those datasets, the expression level of the genes was either measured by array, by high throughput sequencing or by RT-PCR. This left out all the datasets that employed methylation profiling, non-coding RNA, genome binding, and SNP genotyping. At the end of this process we were left with 16 metabolic diseases spread across 55 GEO datasets. We decided to use this experiment type because it was the one with the most useful patients.
5. Exploring in depth the remaining datasets with R to select a subset of diseases to be used in the classification. To do this, from each GEO dataset we:
    - Extracted only patients with metabolic diseases.

- Excluded all control/healthy patients.
- Selected the most salient tissue when the dataset had gene expression levels for multiple tissues.
- Excluded all treated patients with drugs or any other treatment used to control the efficiency of therapy.
- Selected only the first measurement for those patients where the gene expression levels were extracted in different hours, days or weeks.

6. Due to the different platforms used in different GEO datasets to extract the gene expression levels, the gene names were very different from each other. Since the extraction of biomarkers would have depended on the name of the genes, we translated all gene names from specific platforms to unique gene symbols. For a few datasets, the unique gene symbols were already saved within the dataset itself. For most datasets, we extracted the gene symbols using the R library *org.Hs.eg.db* which contains genome annotations for Human. In some cases, we had to download the annotation package of the single platform (*hugene10sttranscriptcluster.db, illuminaHumanv2.db, hgu133plus2.db, etc.*). As we have not been able to translate all datasets, we have excluded those without translation.

After the exploratory analysis, we found 11 diseases spread over 32 GEO datasets with at least 10 patients each, and unique gene symbols. However, we decided to continue the analysis only with diseases with more than 20 patients because we feared that with too few patients we would incur in generalization problems in the classification task (which we encountered, as we'll see later). So we selected 5 diseases.

2.2 Diseases

- **Diabete**: is a chronic, metabolic disease characterized by elevated levels of blood glucose, which over time leads to serious damage to the heart, blood vessels, eyes, kidneys, and nerves. We found 12 GEO datasets (GSE54350, GSE38642, GSE26887, GSE13760, GSE27949, GSE25462, GSE19790, GSE174475, GSE162622, GSE189849, GSE176230, GSE179568) with 176 total samples. The number of unique genes for each dataset ranges from approximately 13 000 to 42 000.
- **Gitelman syndrome (GS)**: is a rare genetic disorder in which there is a specific defect in kidney function. It impairs the kidney's ability to reabsorb salt and causes changes in various electrolyte concentrations as well as contraction of extracellular fluid volume causing dehydration. We found 1 GEO dataset (GSE117146) with 29 samples. The number of genes found for this disease is about 12 000.
- **Neonatal adrenoleukodystrophy (NALD)**: is a leukodystrophy that causes damage to the myelin sheath, an insulating membrane that surrounds nerve cells in the brain. It belongs to the Zellweger spectrum of peroxisome biogenesis disorders, itis considered a moderately severe form, and it is caused by a mutation of several PEX genes. We found 3 GEO datasets (GSE117647, GSE85804, GSE34308) with 38 total samples. The number of unique genes for each dataset ranges from approximately 20 000 to 42 000.
- **Multifactorial chylomicronemia (MCM)**: is the most frequent cause of severe hypertriglyceridemia and is associated with an increased risk of acute pancreatitis, cardiovascular disease, and non-alcoholic steatohepatitis. We found 1 GEO dataset (GSE149607) with 28 samples. The number of genes found for this disease is about 15 000.
- **Alpha-1 antitrypsin deficiency (A1A)**: is an inherited condition that raises your risk of lung and liver disease. Alpha-1 antitrypsin (AAT) is a protein that protects the lungs. The liver makes it. If the AAT proteins aren't the right shape, they get stuck in the liver cells and can't reach the lungs. We found 1 GEO dataset (GSE109516) with 128 samples. The number of genes found for this disease is about 23 000.

2.3 Dataset Cleaning

Before proceeding with the biomarker identification, as the data came from different experiments and platforms, we decided to clean up datasets of the selected diseases individually. First, we removed

outliers from each dataset using a Robust Scaler (fitting it ad hoc for each dataset) and removing genes outside the interquartile range. Then we scaled each dataset using a MinMax Scaler.

2.4 Dataset Merging

As a final step, we have merged the datasets relating to the 5 metabolic diseases. To do this, we have found the common genes between all different datasets and we just kept these genes for each dataset. In the end, we simply concatenated the datasets. Using this approach we reduced a lot the number of genes for the classification task. In fact, after the intersection of the genes, we obtained a total of 4 601 genes, and 399 patients.

3 Phase 2

The goal of the second phase was to build a classifier capable to distinguish diseases from each other by analyzing the expression levels of patients. As a first approach, we tried to identify the significant biomarkers using differential expression analysis. As we will see, this approach has not led to great results. The second step was to build a classifier, testing different models and selecting the best one for our task.

3.1 Statistical Tests

To identify molecular components that show a statistically significant difference between samples, we applied the pairwise statistical testing technique. It consists in calculating hypothesis testing between all pairs of diseases, selecting genes statistically significant for every pair, and then computing the intersection of genes between all pairs [1]. We tested different hypotheses testing comparing their results and we applied the Bonferroni correction to the significance level as we were doing multiple comparisons. We have chosen as significance level the standard value of 0.05 which with Bonferroni correction has become 0.005. This approach didn't work as we hoped: for each test, the intersections of the genes between the disease pairs were empty. So, instead of doing the intersection of genes between pairs, we decided to do the union of them. We followed this approach since in any case, the next step would have been the classification, and a multiclass classifier would have been involved in the diagnosis of the patients. The following table summarizes the results obtained from statistical testing.

Testing	# intersected genes	# joined genes
T-test ($\alpha = 0.005$)	0	3 960
Wilcoxon test ($\alpha = 0.005$)	0	3 945
Kolmogorov-Smirnov test ($\alpha = 0.005$)	0	**4 420**
Volcano ($\alpha = 0.005$, LF threshold=0.5)	0	3 184

Table 1: Results obtained from statistical testing

In the end we chose to continue with the biomarkers selected by the Kolmogorov-Smirnov test, reducing the total number of genes from 4 601 to 4 420.

3.2 Statistical Learning

Thanks to the previous work, we arrived at this stage with a well defined dataset, both in terms of rows (patients) and columns (the genes common to all the selected diseases). This means that we could try any model we wanted, but we had to be wary of the following inherent problems: the number of features was much higher than the number of samples and the classes were heavily unbalanced. We easily solved the second problem with a stratified splitting, to build the training set (75% of the

[1] Zhang, L., Thapa, I., Haas, C. et al. Multiplatform biomarker identification using a data-driven approach enables single-sample classification. BMC Bioinformatics 20, 601 (2019). https://doi.org/10.1186/s12859-019-3140-7

dataset) and the test set (25%). We trained different models using the training set, which was split in turn into training (75%) and validation set (25%). The validation set was used internally to chose the best hyperparameters of each model, and also to select the best model of all.

The following table summarizes the results obtained for each model tested, more details on the uses of the models can be found in the relative subsections.

Model	Train accuracy	Val accuracy	Val precision	Val recall	Val F-1 score
Decision tree	0.85	0.73	0.69	0.73	0.71
KNN	1.0	0.96	0.96	0.96	0.96
Neural Network	1.0	0.92	0.92	0.92	0.92
RAC	0.91	0.84	0.88	0.84	0.84
SVM	**1.0**	**0.96**	**0.96**	**0.96**	**0.96**

Table 2: Results obtained from classification (referring to the weighted average)

Disclaimer: in the following section we will often refer to the *accuracy* metric. We are aware of the fact that it isn't really suited for multiclass classification (because it doesn't take into account class imbalance), but for brevity's sake we will refer to it when in reality we also gave heavy consideration to the single precision and accuracy scores of each class.

Feature Selection As a feature selection technique, we decided to combine the information extracted from Random Forest (RF) classifier about feature importances, with a Recursive Feature Elimination (RFE) strategy. First, we trained a simply RF to extract the features that were most relevant in the RF. This led to a gene reduction of the initial genes by a quarter. Then we used a RFE on those genes, reducing the number of genes at each step by 5. The RFE has selected a total of genes equal to 283. These techniques were used for all the models after the grid search, and we noticed that the results obtained from models with fewer features were all better.

3.2.1 Decision Tree

We tested *sklearn*'s Decision Tree classifier. First, we selected the hyperparameters with a grid search getting an accuracy of 61% on validation. The model didn't classify diseases very well. For example, MCM disease was completely misclassified. RFE led to an increase in accuracy to 73%, improving accuracy in recognizing diabetes and MCM, but not recognizing NALD.

3.2.2 K-Nearest Neighbours

We tested *sklearn*'s KNN classifier. First, we selected the hyperparameters with a grid search getting an accuracy of 92% on validation. The model recognizes very well all diseases, with some misclassification for diabetes. The RFE strategy led to an increase in accuracy from 92% to 96%, but also in recall which also goes from 92% to 96%. The problem with this model was that the best selected hyperparameter k was equal to 1. This made us think that the model in doing so is overfitting the training, and could have generalization issues.

3.2.3 Neural Network

We tested a custom implementation of a feed-forward neural network, similar to a Multi-Layer Perceptron Classifier, written with *tensorflow*. We started by trying different layer depths (from 1 to 4) and we focused on 3, which yielded the best preliminary results (accuracy around 60%). Since the dataset is heavily unbalanced, we added three dropout layers (one after each hidden layer) to try to mask the most prevalent classes. We then proceeded with a k-folded grid search on many hyper-parameters; it further enhanced the classification power and we reached a weighted average peak of all metrics of 0.91%. The RFE technique led us to another 0.1% increase in all metrics. Overall the NN managed to recognize pretty well both the A1A and the NALD diseases and consistently misclassified some GS and MCM samples as Diabetes.

3.2.4 Rank Aggregation Classifier

We tested the Rank Aggregation Classifier which is a classifier that uses rank aggregation to classify objects [2] . We got an initial accuracy of 77% on validation, recognizing very well A1A, GS, and NALD. Then we improved it using RFE leading to an increase in accuracy to 84%. It works very well with all diseases, but it seems to have trouble ranking in the largest diabetes class.

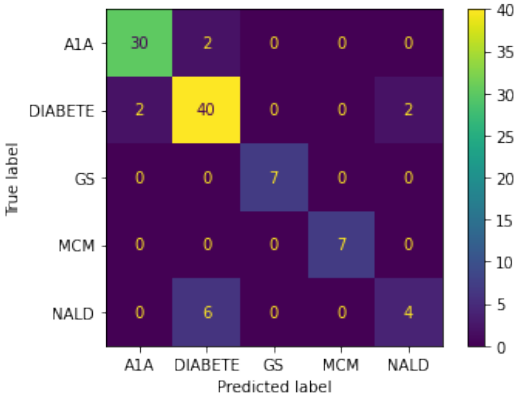
3.2.5 Support Vector Machine

We tested *sklearn*'s SVM classifier. First, we selected the hyperparameters with a grid search getting an accuracy of 95% on validation. The model recognizes very well all diseases except MCM, probably due to the low number of samples. RFE led to an increase in accuracy from 95% to 96%, without however solving the "problem" with MCM disease.

4 Conclusion

To conclude, we selected the SVM with RFE as the best model because it had the best performance on the validation test considering all the metrics. The SVM obtained had a polynomial kernel with degree=2 and a regularization parameter C=4.3, selected by grid search. The results obtained from the model on the test set were the following:

Accuracy	Weighted precision	Weighted recall	Weighted F-1 score
0.88	0.87	0.88	0.87



The model recognized very well A1A, diabetes, GS, and MCM, but badly classified NALD patients. In general, due to some random initialization inside the models, the results may vary by ±4%. All tested models seem to offset each other in the classification of these 5 diseases. Probably an ensemble model, built from these models, could be the right solution in this case.

5 Code

The code of the project can be found at the following link: `https://github.com/federicosilvestri/chl-project`

- *dataset*: folder with the preprocessed dataset downloaded from GEO.
- *src/preprocessing*: folder with R notebooks and scripts used to preprocess the datasets and translate the gene symbols.
- *src/notebooks*: folder with notebooks used to clean and standardize the datasets (*Pipeline.ipynb*), statistical testing (*Biomarker Identification.ipynb*), and classification (*Classification.ipynb, DecisionTree.ipynb, nn grid and strKfold.ipynb*). The final model was tested in *Best Model Test.ipynb*.
- *src/analysis*: folder with python scripts used in every step of this project.

[2]. `https://github.com/alsri/RAC`