Data Analysis and Exploration: a research project on Venous Thromboembolism

Andrea Ambrosi - 211590
University of Trento - Department of Mathematics
MSc Data Science, A.A. 2019-2020
andrea.ambrosi-1@studenti.unitn.it

Abstract— This project is based on the study "*Whole blood gene expression profiles distinguish clinical phenotypes of venous thromboembolism*" of *Deborah A. Lewis, Sunil Suchindran, Michele G. Beckman, W. Craig Hooper, Althea M. Grant et al.* published on *Thrombosis Research* in 2015 [1].
The goal of the project is to use the dataset provided by the aforementioned research and apply models of machine learning to understand if there is the possibility of predict conditions of *venous thromboembolism* based on gene expression.

Index Terms—Venous thromboembolism (VTE), Gene expression, Data analysis, Machine learning

I Introduction

Venous thromboembolism is a complex problem resulting from a *venous thrombosis* which then causes an *embolism*.

A Definitions

Thrombosis is the formation of a blood clot inside a blood vessel which hinder the blood flow through the circulatory system. It can happen that the thrombosis may occur in veins so it takes the name of venous thrombosis or in arteries and it becomes arterial thrombosis.

Venous thrombosis leads to congestion of the affected part of the body, while arterial thrombosis affects the blood supply. An arterial or a venous thrombus can break off as an embolus which may be a blood clot as in this case of thrombus, a fat globule (fat embolism), a bubble of air or other gas (gas embolism), or foreign materia.

Embolus can travel through the circulation system and lodge somewhere else as an embolism. The embolism that happens with a blood clot is known as a thromboembolism and if it occurs within a vein it is called venous thromboembolism or commonly VTE.

In this project, data from patients at risk of venous thromboembolism will be analyzed and the objective is to characterize gene expression profiles in order to distinguish between different levels of affections and the control group.

B The dataset

In the study[1] were enrolled patients with venous thromboembolism separated into 3 groups: the first one as 'low-risk' patients that had ≥ 1 provoked VTE; the second one labeled as 'moderate-risk' that were patients had no more than 1 unprovoked VTE; the last one were the one with patients with 'high-risk' patients had ≥ 2 unprovoked VTE. Other that these patients some individuals with no history of VTE were enrolled as 'healthy controls'.
After some elaborations, data that have been included in the dataset used in the study are from 132 patients divided as 40 at high-risk, 33 at moderate-risk, 34 at low-risk and then 25 healthy for controls. For each of the patients have been captured 47304 probes from the 'Illumina HT-12 V4 Beadchips'.
The dataset that results is structured as in the sample in table 1.

Table 1: Data example

	GSM1164706	...	**GSM1164837**
ILMN_1343291	15.132735	...	15.19889
ILMN_1343295	13.341013	...	13.00436
ILMN_1651199	7.360679	...	6.390681
ILMN_1651209	7.192222	...	8.282515
...	...	...	...
ILMN_3311190	7.623467	...	7.829984

Within the patients the behaviour is very similar and it can be seen by the fact that probes are equally and evenly distributed for each observation from a global minimum of 0.0144 to a maximum of 15.49.

C The problem

The aim of this project is to analyze data related to venous thromboembolism. It will be characterized by some of the state-of-the-art techniques that will be presented in the next sections.

After this overview about the problem and the dataset, the report will procede with the description of the used methods like PCA, random forest, SCUDO etc before going to the results of these algorithms that will provide some information about the feasibility of a prediction analysis to try to predict the condition of patients on the basis of their gene expression profile.

II Methods

In this section will be analyzed various techniques used in the field of machine learning to try to predict the class of some observations based on the available data.

The objective of this project is to try to use the probes related to patients and find the subset of optimal predictors to build models that, based on the observed values, correctly classify a patient in the 4 categories available: 'Healthy Control', 'Low Risk', 'Moderate Risk' and 'High Risk'.

The first set of techniques used is part of the unsupervised learning scope as the observation labels are not used in the prediction models. The algorithms used are PCA, hierarchical clustering and k-means.

In the second set, supervised learning techniques such as random forest and LDA will be used. In this second case the model knows which are classes of each patient and use this information to do the classification.

A PCA

Principal Component Analysis is often used when the objective is to try to reduce the dimensionality of a dataset both to be able to represent it in an easily understandable way such as on two or three dimensions and in case the aim is to understand which subset of variables can be used to represent the dataset without using it entirely.

In figure 1 is represented the variance explained for each principal component. Note that in this plot there are only the first 25 principal components but are calculated as many as the observations that are in the dataset: in this case 132 (one for each patient). What is represented in the figure 1 is what is called 'scree plot', a line plot of the eigenvalues of factors or principal components in an analysis [2]. This is used to decide how many components to use to represent the data based on the level of variance you want to explain.

To represent data in 2-dimensions are used the first 2 principal components that usually explain quite well the behaviour of data. In this case were there are a lot of variables it is a difficult approach to use and as it is possible to see using the first two principal component it is represented only $\sim 40\%$ of the entire variability but this is the price to pay in order to have a 2-dimension plot instead of a 130 more.

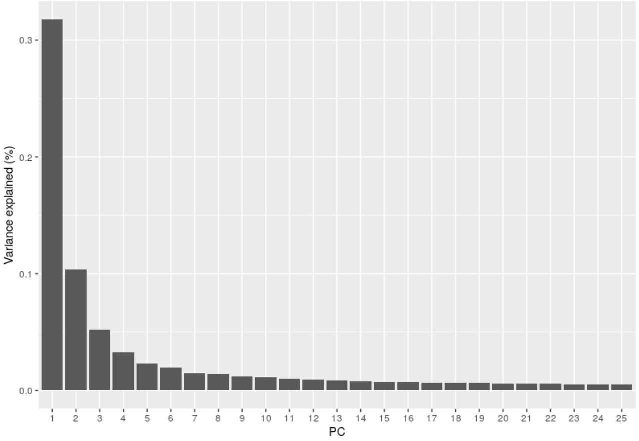

Fig. 1: PCA scree plot of the first 25 principal components

In figure 2 the observations are plotted according to the first two principal components. It is not very clear but already in this case it is possible to note clusters according to the labels: 'Healthy Control' patients are on the higher part of the plot divided in a sparse group on the left and a dense group on the right; 'High Risk' patients are on the right part of the graph more concentrated on the lower side; 'Moderate risk' patients are more clustered on the higher-right side with a lot of patients outside the cluster while the last group of 'Low Risk' patients are also in the higher-right side of the plot but are very sparse.

This first view on data is highly limited by the small amount of explained variance but as seen there is already a sort of division inside the representation.

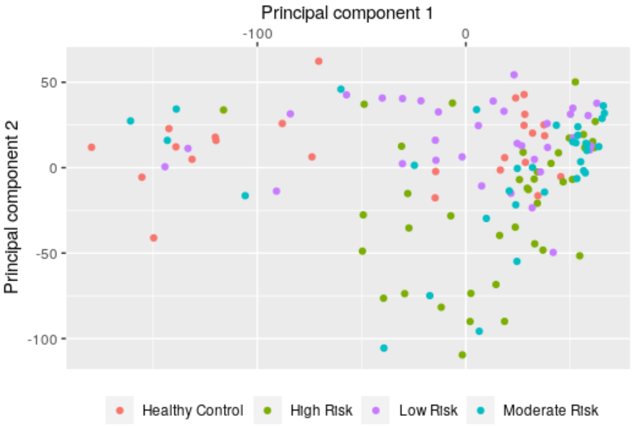

Fig. 2: First principal component vs. second principal component

B K-means

K-means is the first algorithm that will be used to predict the class of some observations. It works partitioning n observations into k clusters in which each observation belongs to the cluster with the nearest mean. The partitions are made in a way that minimizes within-cluster variances using the Euclidean distance. To do this it divides the 2-dimensional space into k partitions with lines that graphically divides the observation space.

Figure 3 shows the output for the observed dataset. Clusters made by k-means doesn't have a label because they are made on the basis of observations that have no label so it doesn't know to which subset an observation belongs.

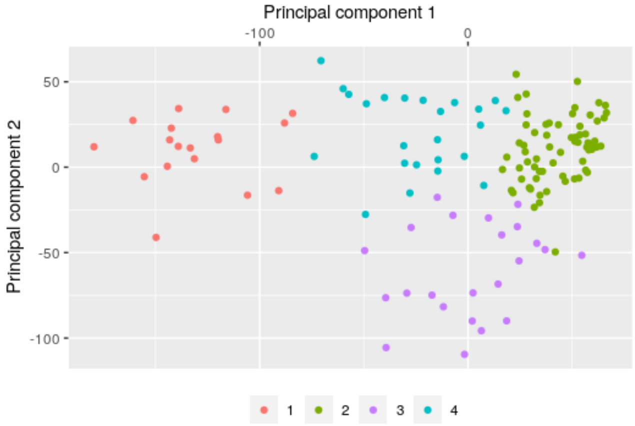

Fig. 3: K-means classification results

Results of clustering are plotted on the first 2 principal components in order to have a comparison between the real class divison in figure 2 and the one by k-means in figure 3.

The results of k-means broadly retrace the idea presented above on the class division in the principal components plot. Obviously there are a lot of imperfections due to the fact that the division in the reality is not so clean as k-means would like to show. Therefore this method is far from being a good way to model these data.

In table 2 are reported the results of the algorithm. Based on the considerations made looking at the principal components result, it would be possible to say that cluster 1 is likely to be 'Healty Control' patients, cluster 2 'Moderate Risk' patients, cluster 3 'High Risk' patients and cluster 4 'Low Risk' patients. With a division such this one, it can be easily note that the error rate is very high considering those that are miss-classified in this setting the error is about 56%.

TABLE 2: K-MEANS CLUSTERING RESULTS

	Healthy Control	Low Risk	Moderate Risk	High Risk
1	9	4	4	1
2	12	17	20	18
3	1	0	6	16
4	3	13	3	5

C Hierarchical clustering

Hierarchical clustering is a method of cluster analysis that can build a hierarchy of clusters by starting from the entire dataset and dividing it at each step until the desired performance is reached or it can go reversely, starting with each observation alone and step by step cluster them.

'R' provides few algorithms to perform hierarchical clustering and in table 3 are reported the sizes of clusters with each method.

TABLE 3: HIERARCHICAL CLUSTERING RESULTS

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Ward D	15	26	77	14
Ward D2	15	27	76	14
Single	129	1	1	1
Complete	19	81	19	13
Average	14	116	1	1
Mcquitty	24	30	77	1
Median	129	1	1	1
Centroid	129	1	1	1

As seen, the structure of these data doesn't allow to use every algorithm. In fact, algorithms like 'Single', 'Average', 'Median' and 'Centroid' doesn't work at all. Let's now graphically compare the results of the 'Ward D2', 'Complete' and 'Mcquitty' methods that are the only ones that perform at least a little better than the others.

They are plotted again on the first two principal components using the same representation of the first plot. As can be seen, they override more or less the behaviour of k-means. This is because they work with the same idea of dividing observations based on distances. However in this case, doing the same pairing of true-classes and predicted-classes based on the considerations of the first PCA plot, results are quite better but again far from being precise.

Considering only these last 3 algorithms, the multi-class area under the curve is 0.5802 for the 'Ward D2', 0.5979 for the 'complete' and 0.5842 for the 'mcquitty' algorithm meaning that more or less they have a 40% of area that remains higher than the line.

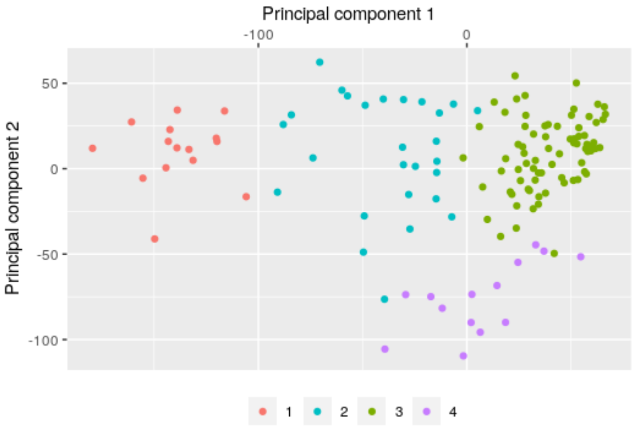

Fig. 4: Hierarchical clustering: Ward D2

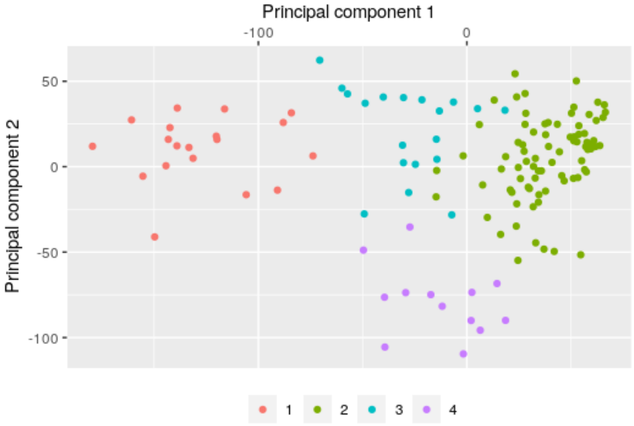

Fig. 5: Hierarchical clustering: complete

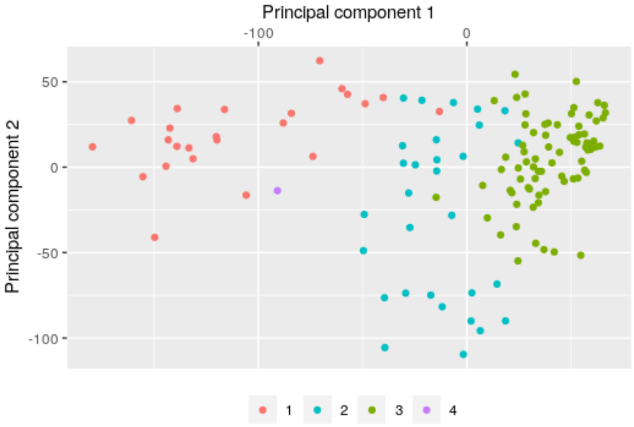

Fig. 6: Hierarchical clustering: Mcquitty

The ROC curve is created by plotting the sensitivity (or true-positive rate) against the false-positive rate (1-specificity). The sensitivity measures the proportion of positives that are correctly identified while specificity measures the proportion of negatives that are correctly identified. As can be imagined, the ideal area under the ROC curve is equals to 1 meanings that the specificity and sensitivity are both 1 providing the perfect classifier.

Passing to the error rate for these 3 algorithms, it results that it is quite high: 62.12% for the ‘Ward D2’, 60.60% for the complete’ and 74.24% for the ‘Mcquitty’. Remind that these were the top 3 algorithms so they provide a very poor performance.

To conclude this first part on the unsupervised learning it is possible to note that none of these algorithms actually fits well the data in this dataset.

Now is the turn of the supervised learning techniques that will hopefully provide better performances.

D *Random forest*

Let’s move then on the supervised learning techniques. The first analyzed is random forest. It operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees[3].

A first classification step was performed using the random forest with all the available predictors and with a forest having $n = 1000$ trees. The random forest returns a list of importance attributed to each predictor which is used to decide at each step which predictor to use to distinguish between the observations and let them fall into one branch or the other of the tree. Given the large number of predictors, it can be misleading in some cases to work with so many variables, thus causing the background noise to overwhelm the variables that are more important at the level of prediction. Once known the predictive importance of each variable, it is possible to decide to use a smaller subset of predictors to train the model. In order to understand the optimal number of predictors to be used in the model some tests have been made varying the number of predictors taken in order of importance from the result of the previous run. In figure 7 it can be noted that, using only the most important predictors ($n = 2$) the error is very high. By adding predictors step by step it is possible to reach the minimum around 400 ± 300 with a minimum error of 33%. Going on with the addition of predictors the error rate grows again reaching a maximum of 56% when all predictors are used.

This result confirm that, as seen before, not all predictors have the same importance and it is therefore better to select a small set of predictors to use in the prediction model rather

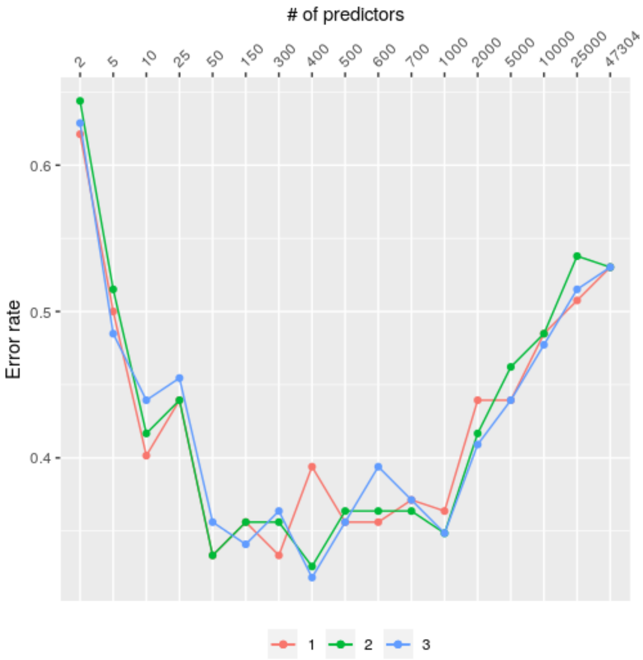

Fig. 7: Error rate comparison for 3 different seed

than insert them all. The large number of predictors will therefore act as a distractor leading the model to pick the wrong class as the number of predictors increase.

Since there is still some variability in the results for the way the sample is created, this procedure should be tried with a high number of seeds to understand what is the minimum error rate that can be achieved. In this case 3 tests have been done using 3 different seed and the results for the best one are plotted in the table 4. This table is computed using the prediction of the function *randomForest*(...) without the train/test usual strategy. The minimum is reached using the *seed* = 3 which brings the global error rate to 0.3182 with the use of the first 400 predictors (in order of importance).

TABLE 4: RANDOM FOREST CONFUSION MATRIX RESULTS

	Healthy Control	Low Risk	Moderate Risk	High Risk	Class Error
Healthy Control	17	6	1	1	32.0%
Low Risk	4	24	1	5	29.4%
Moder. Risk	4	2	18	9	45.5%
High Risk	2	6	1	31	22.5%

In general, random forest performs better than the unsupervised learning techniques seen above. However, with these percentages of error it cannot be considered satisfying result considering a prediction that would be incorrect, as in the case of the 'Moderate Risk' class, with a still very high percentage.

In the last two tables for the random forest algorithm are reported the summary of the test error rate in table 5 and the summary of the test AUC in table 6 that are the values of area resulted from the multiclass-ROC computation at each iteration. They have been both computed using the best 400 predictors according to the previous considerations and running the analysis with 1000 different train/test sets. Remember that the AUC value is always between 0 and 1 and is equal to the probability that a classifier will rank a randomly chosen *positive* instance higher than a randomly chosen *negative* one (assuming *positive* ranks higher than *negative*)[4].

As aforementioned, the error rate still remain too high even if the results of the AUC are quite good, considering that in the original study [1] the best performance is achieved considering separately the groups and with an AUC that vary between 0.69 and 0.84 by situation to situation.

TABLE 5: SUMMARY OF THE RANDOM FOREST ERROR RATE

Min.	1st Q.	Median	Mean	3rd Q.	Max.
0.125	0.281	0.344	0.337	0.375	0.594

TABLE 6: SUMMARY OF THE AUC FOR THE RANDOM FOREST COMPUTATION

Min.	1st Q.	Median	Mean	3rd Q.	Max.
0.448	0.715	0.773	0.764	0.816	0.924

Considering the distribution of the values of area provided by the ROC curve, an average value of 0.764 can be considered quite satisfactory noting also that the first quartile reach a value of 0.715 within 1000 test.

E LDA

Linear Discriminant Analysis is used to find a linear combination of features that separates two or more classes of observations and can be applied for dimensionality reduction and for classification problems. The idea of linear discriminant analysis is to fin the most discriminant projection by maximizing between-class distance and minimizing within-class distance. With this algorithm comes immediately an important decision to be taken: the use of all the predictors in this case causes 'R' to crash because it would require an enormous space in memory to be able to execute the $lda(...)$ call. To decide which predictors to keep into the analysis using LDA, the previous result of the 'random forest' method has been taken into consideration and the 400 best predictors were then used. When the call is made, a 'warning' is shown in the result saying that there is collinearity between some variables. Predictive power can decrease with an increased correlation between predictor variables [5]. This is not to be considered an error but it must be taken into consideration that in the dataset there are variables with a very similar behavior that cause the performance of the algorithm to drop. With this in mind, some tests were made with a sort of cross-validation and 100 different seeds to understand the goodness of the predictions. In table 7 is reported the summary of the error rate that results after running 100 different LDA.

TABLE 7: SUMMARY OF THE LDA ERROR RATE

Min.	1st Q.	Median	Mean	3rd Q.	Max.
0.219	0.344	0.406	0.398	0.469	0.594

The result does not bode well as only on 2 occasions out of 100 the error is less than 25%.

Considering instead the AUC value, the table 8 shows that there is a peak of 0.905 that is far better than the original study [1]. Considering that in other 7 cases the AUC value for the LDA algorithm is over 0.84 (the best AUC value of the original study) this can be considered a huge goal in this analysis. The best value of AUC also corresponds to the best error rate which is the minimum shown in the table 7.

TABLE 8: SUMMARY OF THE AUC FOR LDA COMPUTATION

Min.	1st Q.	Median	Mean	3rd Q.	Max.
0.556	0.681	0.734	0.730	0.781	0.905

F SCUDO

'Signature-based ClUstering for DiagnOstic purposes', is an online tool for the analysis of gene expression profiles for diagnostic and classification purposes [6]. This algorithm works with sample of genes that acts as signatures of each observation. The analysis is therefore done using signature-to-signature similarity measure. The objective of this algorithm is to classify observations on the basis of gene-expression with the advantage that it has less impact on overfitting and batch effect that are sometimes common with other methods.

Signature-to-signature similarity works defining a signature for each profile. This is done using the n most expressed probes and the m less expressed probes. The parameters n and m are usually between 200 and 300 genes considering an average dataset that have 30/40 thousand probes. Then, a similarity matrix is drawn to compute the similarity measure between signatures. The resulting information will be used to draw a graph that will show the observations connected by an edge (usually only the best 30% are plotted to reduce noise and this can be tuned with the parameter N) with a length inversely proportional to the similarity value between the observations so similar observations will be plotted close to each other.

An important step is the tuning of the parameters n and m because from them depends the length and the complexity of the signature. In order to select the best couple of parameters that provide the highest level of accuracy and specificity, a cross validation step have been used.

Before of this phase, a step of features selection has been done keeping only those features under a p-value of 0.01. Then the last parameter N have been set to 0.25 to have a low-dense matrix. With this setting have been tried every combination of n and m within this set of 10 values: 50, 100, 150, 200, 250, 300, 350, 400, 450 and 500. At the end of this step, using the package 'caret' it is possible to have the best couple of n and m provided considering a lot of metrics other than accuracy, specificity, sensitivity etc. In this case, the best couple with the provided setting is $n = 300$, $m = 500$ so it gives more importance to the down-regulated probes instead of the up-regulated probes.

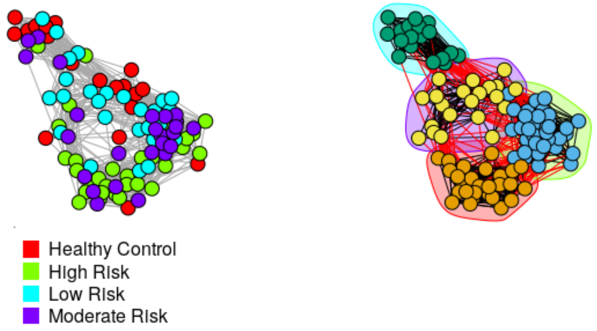

Fig. 8: Train network result

In figure 8 it is possible to see the resulting SCUDO plot after the training step on the left and the relative cluster-view on the right. The complexity of this dataset doesn't allow to provide a good visualization of the results where it is possible to note some hint of similarity in few section of the plot but as for the previous methods it will be difficult to have a good performance in the test step.

After this part were have been done the tuning of the parameters and then the training of the model, it is possible to use it

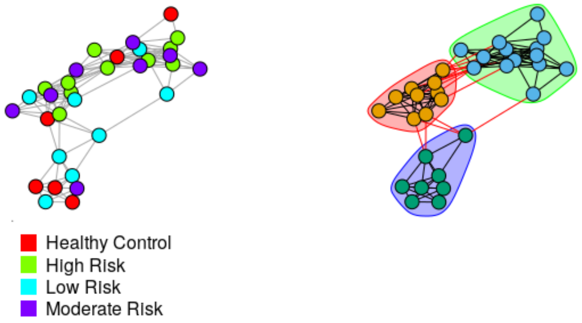

Fig. 9: Test network result

to predict the behaviour of new observations. Resulting network of the test is provided in figure 9 where due to the low size of the sample it difficultly recognize 4 different clusters (maintaining the same parameter setting as before).
Then in table 9 is reported the confusion matrix provided by the *scudoClassify*(...) function where it computes a SCUDO model based on the train function used before.
Results are a bit less encouraging if compared to random forest and LDA but with few running more, with different parameters, would be a chance to observe different signatures and then different results. As mentioned above, this result is given using the same p-value of 0.01 as threshold for probes to use in the signature and with an N of 0.25 for keeping the matrix sparse. Increasing the number of connections between observations could be interesting given that in this context, as can be seen from the graphs, there are few collections of very similar observations and this could increase the predictive power of the model.

TABLE 9: CONFUSION MATRIX OF SCUDO CLASSIFY FUNCTION

	Healthy Control	**Low Risk**	**Moderate Risk**	**High Risk**
Sensitivity	0.333	0.500	0.250	0.700
Specificity	0.846	0.792	0.875	0.773
Pos Pred Value	0.333	0.444	0.400	0.583
Neg Pred Value	0.846	0.826	0.778	0.850
Prevalence	0.188	0.250	0.250	0.313
Detection Rate	0.063	0.125	0.063	0.219
Detection Prevalence	0.188	0.281	0.156	0.375
Balanced Accuracy	0.590	0.646	0.563	0.736

To conclude with SCUDO and to have a comparison parameter between different forecast models, with the best parameter setting, SCUDO reaches an AUC of 0.6174.

III TOOLS

To improve the analysis and to try to increase the value of the best probes used to discriminate between the different VTE risk groups were used two tools: DAVID and Enrichnet.

A DAVID

DAVID[7][8] is an online tool that helps in doing the functional annotation of genes. It works by uploading a list of probes that it translate in GOterms using gene ontologies pathways etc.
The list of probes to upload on DAVID have to be retrieved by previous steps of analysis. One way is to select those probes in random forest, LDA or SCUDO that provided the best performances in classifying results.
In table 10 are provided those terms that DAVID translated according to the uploaded list of 400 probes that in this attempt was taken by the execution of the best classification with the random forest algorithm. Other than those shown in the table, DAVID recognized other 308 terms that have not been reported. In table are present only those terms that have a Benjamini-corrected p-value under 0.1. This is not a very statistically-significant result but are the most important resulting terms with the given set of probes.

TABLE 10: DAVID RESULT WITH RANDOM FOREST PROBES

Term	**Count**	**%**	**P-Value**	**Benj.**
organelle lumen	42	13.9	9.1E-4	6.2E-2
membrane-enclosed lumen	43	14.2	7.1E-4	6.4E-2
intracellular organelle lumen	42	13.9	5.8E-4	7.7E-2
nuclear lumen	37	12.3	3.4E-4	9.0E-2

Another attempt have been done using the probes of the SCUDO algorithm. Scudo provides for each groups the set of probes that are used for the up-regulate and the down-regulate signature.
The first idea was to try the first 100 probes in the up-side and in the down-side for the ‘High Risk’ category but this provided very low statistically-significant results with the best Benjamini-corrected p-value of 0.6.
Then a second idea that have been tried was using all the probes in the signature of the ‘High Risk’ patients. This time have been produced few results that are someway more significant (but again, not statistically, because above the threshold of 0.05), reported in table 11.

TABLE 11: DAVID RESULT WITH SCUDO PROBES FOR 'HIGH RISK' GROUP

Term	Count	%	P-Value	Benj.
mitochondrion	65	10.3	3.1E-6	1.2E-3
REACT_1505: Integration of energy metabolism	21	3.3	4.6E-4	2.5E-2
ribosomal subunit	14	2.2	3.0E-4	3.0E-2
mitochondrial part	38	6.0	1.5E-4	3.0E-2
cytosol	68	10.8	2.6E-4	3.4E-2

Instead of using the entire signature in table 12 are reported the terms discovered using only the up-regulated probes for the 'High Risk' category. This setting provide more statistically-significant results.

TABLE 12: DAVID RESULT WITH UP-REGULATED PROBES OF SCUDO FOR 'HIGH RISK' GROUP

Term	Count	%	P-Value	Benj.
protein localization	32	12.9	5.0E-7	7.5E-4
membrane-bounded vesicle	23	9.3	5.2E-6	1.4E-3
vesicle	24	9.7	2.2E-5	2.0E-3
endosome	16	6.5	1.7E-5	2.3E-3
cytoplasmic membrane-bounded vesicle	21	8.5	3.6E-5	2.4E-3
establishment of protein localization	27	10.9	9.5E-6	4.7E-3
cytoplasmic vesicle	22	8.9	1.0E-4	5.5E-3
protein transport	27	10.9	8.0E-6	6.0E-3

To conclude, in table 13 are reported those terms retrieved by applying DAVID on the down-regulated probes with a statistically-significant Benjamini-corrected p-value.

TABLE 13: DAVID RESULT WITH DOWN-REGULATED PROBES OF SCUDO FOR 'HIGH RISK' GROUP

Term	Count	%	P-Value	Benj.
mitochondrion	50	13.0	3.3E-8	9.6E-6
ribosomal subunit	14	3.6	1.5E-6	2.1E-4
mitochondrial part	30	7.8	6.7E-6	4.9E-4
ribosome	17	4.4	5.8E-6	5.6E-4
organellar ribosome	8	2.1	4.0E-5	2.3E-3
mitochondrial ribosome	8	2.1	4.0E-5	2.3E-3
structural constituent of ribosome	15	3.9	6.5E-6	3.3E-3

B Enrichnet

The last tool used in the analysis is Enrichnet[9]. This online tool provide an uploading window where to load a list of genes and then it will be possible to use that list to make a network of genes enriching it with pathways, 'near' genes etc. To use this tool have been uploaded the probes list given by the best classification in random forest for a first attempt and the up-regulated probes for the 'High Risk' group in a second try. Since Enrichnet wants a list of genes, the probe list has been translated (using DAVID) to "Entrez gene" since the version of Illumina with which this data was captured was not present in Enrichnet. Then Enrichnet provides a set of base network that can be used in the enrichment like KEGG, GO, Reactome etc. The figure 10 show the result of Enrichnet using 2 different list of genes and using the same base network: GO. On the left the one computed with the random forest probes and on the right the one computed with the SCUDO up-regulated probes for the 'High Risk' group. It is possible to note that different list have provided different networks for the same enzyme. When Enrichnet provide the result of the computation, other that showing the pathways it finds, it gives also some metrics to evaluate the network like the significance of network distance distribution (XD-Score) or significance of overlap (Fisher-test, q-value) etc. Both the results in the figure 10 have not a very significant p-value but since they were provided using 2 different set of probes, it could be interesting to go deeper with this analysis.

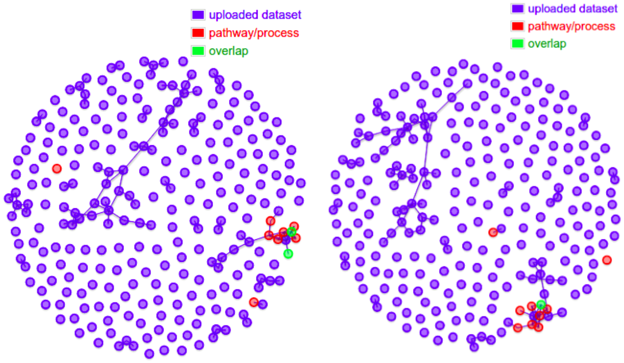

Fig. 10: GO:0008253 5'nucleotidase activity

IV FINAL CONSIDERATIONS

To summarize what has been done in this analysis, it is possible to start from the goal of this study that was to understand if through the dataset of the study [1] it was possible to make predictions on the group they belong to and therefore distinguish the different patients by their gene expression in

order to predict their risk of VTE.
The dataset was initially analyzed with unsupervised learning techniques: first K-means and then Hierarchical clustering. Both proved to be unsuitable for the study of the data in analysis as the average error remained very high, making preferable a random prediction.
The dataset was therefore analyzed using supervised learning techniques which gave more encouraging results. Between the three algorithms used, SCUDO produced the least number of correct results, with an AUC of 0.62 when using the best setting. Random Forest and LDA instead produced results that match those of the original study, overwhelming them on some occasions. In fact, Random forest achieves an AUC of 0.94 in its best score even if on average it still has an error rate of 34%. LDA also reaches an AUC of 0.91 in its best computation but considering the average error in this case it remains at 40%.
Comparing the results with the original study using the AUC value, it is possible to see how the results are very close, also considering that in the other case different tests were performed which directly compared the VTE risk groups without using all the data in the same test as in this analysis.
DAVID was then used to try to enrich the probes that discriminate between risk classes but given the nature of the dataset, it did not produce the desired results, giving statistically significant terms on a few occasions.
To conclude, it was interesting to see how in a complex dataset where making predictions was difficult considering the nature of the data, there were some algorithms such as random forest and LDA which in some cases have obtained excellent results. Considering the obtained results, there would be space for some improvements, especially after an enrichment phase which if deeply analyzed could introduce new genes or pathways not considered and which could significantly increase the level of accuracy in predicting risk levels of VTE.

V APPENDIX A: CODE

The data analysis code was written using the R language and is available at "https://github.com/aambrosi/VTE-data-analysis".

REFERENCES

[1] Deborah A. Lewis, Sunil Suchindran, Michele G. Beckman, W. Craig Hooper, Althea M. Grant, and et al. Whole blood gene expression profiles distinguish clinical phenotypes of venous thromboembolism. *Thrombosis Research*, Vol. 135(4):659–665, 2015.

[2] George Thomas Lewith, Wayne B. Jonas, and Harald Walac. Clinical research in complementary therapies: Principles, problems and solutions. *Elsevier Health Sciences*, page 354, 2010.

[3] Ho Tin Kam. Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pages 278–282, 1995.

[4] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, Vol. 27:861–874, 2006.

[5] Ö BÖKEOĞLU ÇOKLUK and Ş. BÜYÜKÖZTÜRK. Discriminant function analysis: Concept and application. *Eğitim araştırmaları dergisi*, Vol. 33:73–92, 2008.

[6] Mario Lauria, Petros Moyseos, and Corrado Priami. Scudo: a tool for signature-based clustering of expression profiles. *Nucleic Acids Research*, Vol. 43:188–192, 2015.

[7] Huang DW, Sherman BT, and Lempicki RA. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nature Protoc*, Vol. 4(1):44–57, 2009.

[8] Huang DW, Sherman BT, and Lempicki RA. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res*, Vol. 37(1):1–13, 2009.

[9] E. Glaab, A. Baudot, N. Krasnogor, R. Schneider, and A. Valencia. Enrichnet: network-based gene set enrichment analysis. *Bioinformatics*, 28(18):i451, 2012.