RESEARCH

Project-3

Sujan Darai (sujad96@zedat.fu-berlin.de), Matanat Mammadli (matanam94@zedat.fu-berlin.de), Samra Hamidovic (samrah96@zedat.fu-berlin.de), Ibrahim Ibraimi (ibraimii@hu-berlin.de)

Full list of author information is available at the end of the article

Abstract

Goal of the project: Develop (train) a Random Forest classifier to classify healthy vs cancer samples. Comparison of performance of the Random Forest classifier with Support Vector Machine from the papers of Zhang and Best.

Main results of the project: Random Forest classifier model's accuracy was slightly higher than the SVM classifier (in Best paper [2]) but worse for noisy data. Extracting top features and comparing them with Zhang's paper [10] results do not match and realizing that we need to do features selection carefully for our future projects.

Personal key learning:

1 Sujan Darai: Learned to balance the data and noise addition
2 Samra Hamidovic: Learned to modify code and to handle errors better.
3 Matanat Mammadli: Learned to describe methods and plots better, and gained a clearer understanding of our code outputs and dataset insights.

Estimated working hours:

1 Sujan Darai: 8 hours
2 Samra Hamidovic: 8 hours
3 Matanat Mammadli: 8 hours

Project evaluation: 1

Number of words: 3074

Keywords: Precision medicine, Tumor-educated platelets (TEP), Oversampling technique, Confusion matrix

1 Introduction

Precision medicine has emerged as a cornerstone of modern healthcare, particularly in the field of oncology. This approach recognizes that each individual's genetic makeup, environmental exposures, lifestyle factors, and tumor biology are unique, influencing their response to disease and treatment. By harnessing advanced technologies and analytical tools, precision medicine aims to tailor diagnostic, therapeutic, and monitoring approaches to specific patient subgroups, optimizing therapeutic outcomes while minimizing adverse effects.

Cancer remains a significant global health challenge, with early detection and accurate diagnosis playing pivotal roles in improving patient outcomes. Traditional methods for cancer diagnosis often involve invasive procedures and can be limited in their ability to detect tumors at an early stage. However, recent advancements in liquid biopsy techniques offer a non-invasive and potentially more sensitive approach to cancer detection. One such innovative method involves the analysis of tumor-educated platelets (TEPs), which are platelets that have been altered by interactions with tumor cells. Myron G. Best explores in his research paper "RNA-Seq of Tumor-Educated Platelets for Blood-Based Cancer Diagnostics" [2] the potential of TEPs as a novel biomarker for pan-cancer diagnostics. By analyzing the RNA profiles of TEPs, researchers have identified a class-specific gene signature that shows promising correlations with curated gene sets.

This report delves into the implications of using TEP-based liquid biopsies for guiding clinical diagnostics and personalized therapy selection in cancer patients. With machine learning, particularly Random forest, the RNA profiles of tumor-educated platelets (TEPs) were analyzed. Through training the algorithm on a dataset of TEP samples, the research aims to accurately differentiate between non-metastasized and metastasized tumors. Using the unique RNA content of TEPs influenced by factors such as the transcriptional state of bone-marrow megakaryocytes and tumor-specific educational stimuli, to explore various stages of cancer, including in situ carcinomas and pre-malignant lesions.

2 Goal of the project

Our goal for this project was to develop (train) at least one ML classifier (not SVM) that can be used to classify healthy vs cancer samples. To avoid statistical errors, we maintained equal sample sizes across all groups. Furthermore, an evaluation of our model, using a confusion matrix including the accuracy and an ROC curve (including the AUC value) for training AND validation set, was mandatory.

We aimed to compare the performance of our model to the SVM model from Best's paper [2].

In the end, a feature importance analysis was performed, and a comparison of our 18 most important features to the top 18 features from Zhang's paper [10]. Additionally, we introduced noise to our input data and repeated these analyses.

3 Data and preprocessings

3.1 Data source

In the project, GSE68086 data was taken from the National Center for Biotechnology Information (NCBI), which consists of RNA-sequencing information of 285 blood platelet samples, where 231 tumor-educated platelets (TEP) tests were collected from patients with six different tumors such as non-small cell lung cancer, colorectal cancer, pancreatic cancer, glioblastoma, breast cancer, and hepatobiliary carcinomas. The remaining 54 blood platelet samples were collected from the healthy person. This data reflects the ability of TEP RNA-based 'liquid biopsies' in patients with cancer, including the ability for pan-cancer, multiclass cancer, and companion diagnostics. The data contains 57736 rows and 285 columns in which gene IDs are specified in rows and specific cancer types are mentioned in the columns.

3.2 Data preprocessing

The dataset GSE68086 was loaded in Jupyter Notebook for the preprocessing. Since the dataset has 57736 rows and 285 columns, it was then transposed to make it easy for the data analysis. The specific cancer types (i.e. columns) after being transposed into rows were converted to binary class (i.e. index) outcomes 0 and 1 using the regex expression for classification purposes. The class (i.e. index) outcome 0 indicates the healthy person and 1 indicates the cancer patient. It has been found that the data has an imbalance output. The imbalance dataset was balanced using the oversampling technique and finally data has 231 samples from each class. Afterward, we split the 70 % data into training and 30 % into validation datasets with a random state of 1 for the data reproducibility. Furthermore, the noise was added to the training and validation dataset using the Gaussian distribution function with a mean value of 0 and a standard deviation of 20. The reason for the addition of noise was to give the machine learning model to generalize on noisy data and learn the underlying patterns in the data instead of fitting every data point. This will prevent overfitting and accelerate the model's performance.

4 Methods

4.1 Random forest classifier

A random forest is a meta-estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Random Forest classifier is commonly used for classification tasks where the goal is to predict categorical outcomes or class labels (in our case, to classify and predict healthy versus cancer-diagnosed patients).

First, the `RandomForestClassifier` class from the `sklearn.ensemble` module of the `scikit-learn` library was imported. This is a class specifically designed for classification tasks using the Random Forest algorithm.

Afterwards a Random Forest classifier object (`rf`) with a specified random state (`random_state=1`) for reproducibility was created.

4.1.1 Hyperparameter optimization

Hyperparameter optimization refers to the process of tuning the hyperparameters of a machine learning algorithm to improve its performance on unseen data.

In our data, hyperparameter optimization was performed using `GridSearchCV`. `GridSearchCV` is a technique for tuning hyperparameters by exhaustively searching through a specified grid of hyperparameters for the best combination. The `param_dist` dictionary that we used for our data contained the parameter distributions that would be explored during the grid search for finding the best hyperparameters for the Random Forest classifier.

Then a `GridSearchCV` object named `grid_search` was created. This object takes the `RandomForestClassifier` model (`rf`), the hyperparameter grid (`param_dist`), and other parameters such as `cv` (number of folds for cross-validation) and `refit` (whether to refit the best estimator on the whole dataset). We set the parameters of the `param_dist` dictionary to specific values, `max_features` was set to sqrt, meaning it would consider the square root of the total number of features, we set `max_depth` to [5, 10, 15, 20], this parameter controls the maximum depth of the tree. We set `criterion` to gini, which refers to the Gini impurity criterion. This parameter specifies the function to measure the quality of a split. We set `n_estimators` to 100, this parameter defines the number of trees in the random forest.

The fit method was called on the `grid_search` object with the training data (`X_train` and `y_train`). This step performs an exhaustive search over the hyperparameter `grid`, training and evaluating the model with each combination of hyperparameters using cross-validation.

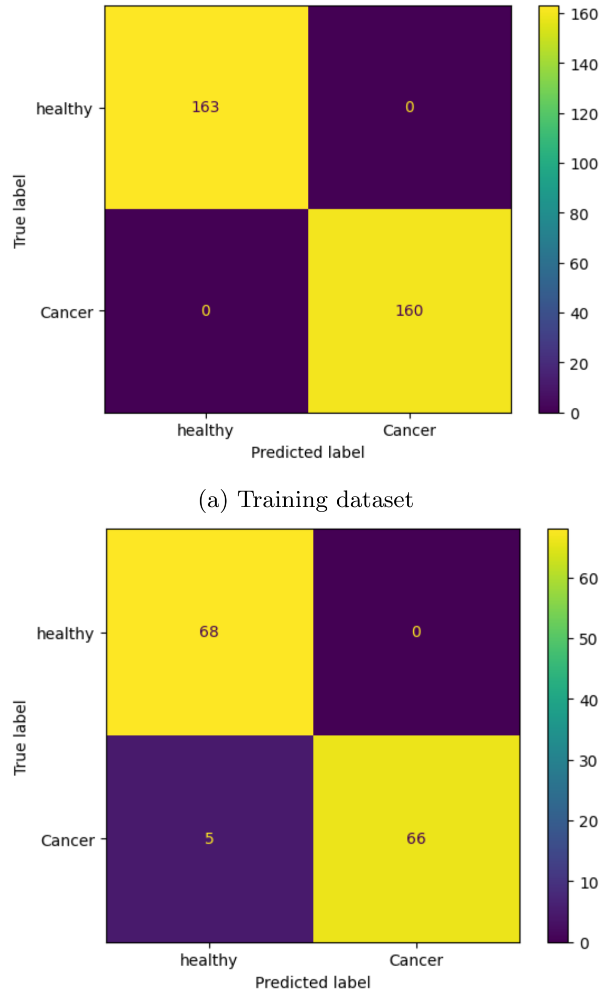
The `grid_search.best_estimator_` attribute was used to retrieve the best-performing model found during the grid search. This model is then stored in the variable `best_rf`.
The `grid_search.best_params_` attribute containing the hyperparameters that resulted in the best performance during the grid search were printed later on.

4.1.2 Confusion matrix

The confusion matrix is a performance measurement tool used in machine learning for classification problems. It is a table that allows visualization of the performance of an algorithm by displaying the counts of true positive, true negative, false positive, and false negative predictions made by the model.
For our data, the `confusion_matrix` and `ConfusionMatrixDisplay` classes from the `sklearn.metrics` module were imported. These classes are used for generating and displaying confusion matrices.
Then predictions (`y_pred`) were generated using the best model (`best_rf`) on the training dataset (`X_train`) and afterwards on test dataset (`X_test`).
The confusion matrix (`cm`) was calculated by comparing the actual target values (`y_train` and `y_test`) with the predicted values (`y_pred`). The confusion matrix was visually represented using the `ConfusionMatrixDisplay` class.
Finally, predictions (`grid_predict`) were generated using the best model (`grid_search`) on the test dataset (`X_test`) and the classification report was printed by calling the `classification_report` function with the actual target values (`y_test`) and the predicted values (`grid_predict`). The classification report includes metrics such as precision, recall, F1-score, and support for each class, as well as the overall accuracy of the model on the test dataset.
We also calculated and printed the accuracy score by comparing the true target values (`y_test`) with the predicted values (`y_pred`) obtained from the model.

4.2 Bootstrap evaluation

Bootstrap evaluation is a resampling method used to estimate the uncertainty of a statistical estimator or to assess the accuracy of a predictive model. It involves repeatedly sampling the data with replacement to create multiple datasets (bootstrap samples) that are similar to the original dataset. These bootstrap samples are then used to compute the statistic or evaluate the model multiple times.
For this part, a bootstrap model was build using the `cb.bootstrap.Per()` function from the `cimcb` package, specifying the original model (`model`) and the number of bootstrap samples (`bootnum=100`) and running the bootstrap samples with `run()`.
In the next step, the bootstrap models were evaluated on our train and test dataset (`EvalTrain` and `EvalTest`). Various evaluation metrics to assess the performance of the bootstrap models were computed. Lastly, for the important feature analysis, we performed a grid search using cross-validation (`cv=5`) to find the best hyperparameters for a Random Forest classifier (`RandomForestClassifier`), and the best model was stored in the `best_rf` variable. The indices of the top 18 important features based on their feature importances from the best Random Forest model (`best_rf`) were printed followed by a dataframe with the associated gene-IDs in it.



(a) Training dataset

(b) Testing dataset

Figure 1: Confusion matrix of training and testing dataset for RF classifier without noise

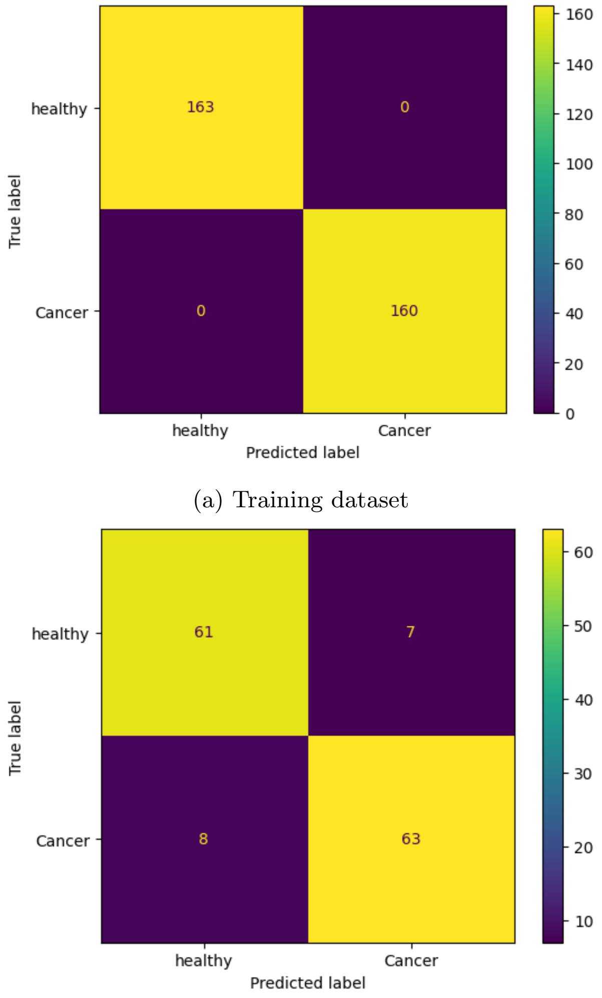
5 Results and discussion

A random forest classifier was used, accompanied by a feature importance analysis.

5.1 Evaluation of the model

Figure 1 displays confusion matrices of train and test datasets. The numbers in these confusion matrices represent counts. Specifically, they indicate the frequency of occurrences of each combination of true and predicted class labels in the dataset used for evaluation.

The confusion matrix of the training dataset (Figure 1a) shows 163 instances where the true class is positive and the model correctly predicts it as positive (top left box). It has zero instances of False Positive (The number of instances where the true class is negative, but the model incorrectly predicts it as positive (Type I error)) and False Negative cases (The number of instances where the true class is positive,



(a) Training dataset

(b) Testing datset

Figure 2: Confusion matrix of training and testing datasets for RF classifier with Gaussian noise

but the model incorrectly predicts it as negative (Type II error)). This confusion matrix displays 160 instances of True Negative cases (bottom right box), where the model correctly predicted the negative class.

Figure 1b illustrates the confusion matrix of the test dataset. Here we can observe 68 True Positive cases. There are zero instances of False Positive cases. On the bottom left, we see 5 instances of False Negative cases. At last, there are 66 True Negative instances.

The accuracy of our classification model for the test dataset is 0.964. The accuracy score represents the proportion of correctly classified samples out of the total number of samples in the test dataset. Figure 2 represents confusion matrices of the train and test datasets after adding Gaussian noise.

The confusion matrix of the training dataset (Figure 2a) displays 163 instances of True Positive cases, zero instances of False Positive and False Negative cases, and

160 instances of True Negative cases. It has exactly similar results to the confusion matrix of the train dataset without the noise.

The confusion matrix of the test dataset (Figure 2b) shows 61 instances of True Positive cases, and 7 instances of False positive cases - which is worse than the results in the test dataset without the noise, 8 instances of False Negative cases - also worse result than in the test dataset without added Gaussian noise. Here we also had 63 instances of True Negative cases. The accuracy score of our classification model for the test dataset with added Gaussian noise is 0.892. Overall we can see that after adding Gaussian noise to our data, our model accuracy and performance was lower.

When we compare our RF classifier accuracy results with the Best paper summary results [6], where they used the SVM classifier, our accuracy for the RF classifier without noise (0.964) was higher than the SVM classifier (95% or 0.95), but our accuracy for RF classifier with noise (0.892) was less than the accuracy they achieved in the paper using SVM method. For our data, two random forest classifiers were used, one from `sklearn.ensemble` library and one from `cimcb` package.

The RandomForestClassifier from the `sklearn.ensemble` library is an implementation of the Random Forest algorithm for classification tasks. It is an ensemble learning method that fits multiple decision tree classifiers on various sub-samples of the dataset and uses averaging to improve predictive accuracy and control overfitting. We performed hyperparameter optimization for a Random Forest classifier using Grid Search Cross-Validation on our data.

The goal was to find the combination of hyperparameters that yields the best performance on the provided training data during the grid search Which in our case were, 'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt' and 'n_estimators': 100. In conclusion, Gini impurity criterion was used to measure the quality of a split, each decision tree in the forest could have a maximum depth of 10 levels, the maximum number of features to consider when looking for the best split was the square root of the total number of features and 100 trees were used.

Figure 3 shows ROC Curves for Random Forest classifiers, without added noise (3a) and with noise (3b). These graphs evaluate the performances of our RF binary classification models.

In Figure 3a, the ROC curve of the Random Forest (RF) classifier without noise exhibits a larger area under the curve (AUC) and closely approaches the top-left corner of the plot. This means that the model achieves high true positive rates (sensitivity) while keeping false positive rates (1 - specificity) low across various threshold values.

Conversely, in Figure 3b, after the addition of Gaussian noise, the AUC of the ROC curve diminishes, and the curve of the test data deviates from the top-left corner. This deviation suggests a decline in model performance compared to the noise-free scenario, indicating a deterioration in the model's ability to effectively discriminate between classes.

5.2 Feature importance

To identify the most relevant features or variables that contribute the most to the predictive performance of a model, we did a feature importance analysis. We ex-

tracted the top 18 most important features in our random forest model, based on their feature importance scores.
We also compared our most important features with the results they achieved in the Zhang paper [10].
However, our genes do not match the genes in the Zhang paper [10]. In this paper, the mRMR (maximum relevance minimum redundancy) method was used for feature selection to identify important genes for distinguishing different cancer subtypes and healthy controls 3. They used gene expression profiles from 285 samples, each represented by 13,445 features indicating the expression level of a gene in sample 10. To filter genes, they discarded genes whose expression level in more than 90% of samples was zero, leaving 13,445 genes for analysis 10. They set a threshold of 0.360 to select important features based on mutual information (MI) values, where features with MI values larger than 0.360 were considered significant [10].
However, we used feature importance from a Random Forest classifier to identify important features. Random Forest classifiers are capable of calculating feature importance scores based on how much each feature contributes to decreasing impurity (in our case, Gini impurity) in the decision trees within the forest. It's not strictly a feature selection method but rather a technique to identify the relative importance of features in a predictive model.
We didn't strictly use feature selection methods, rather than Random Forest Feature Importance, which ranks of features based on their importance in the classification task. Therefore our results are not the same as in Zhang's paper [10].

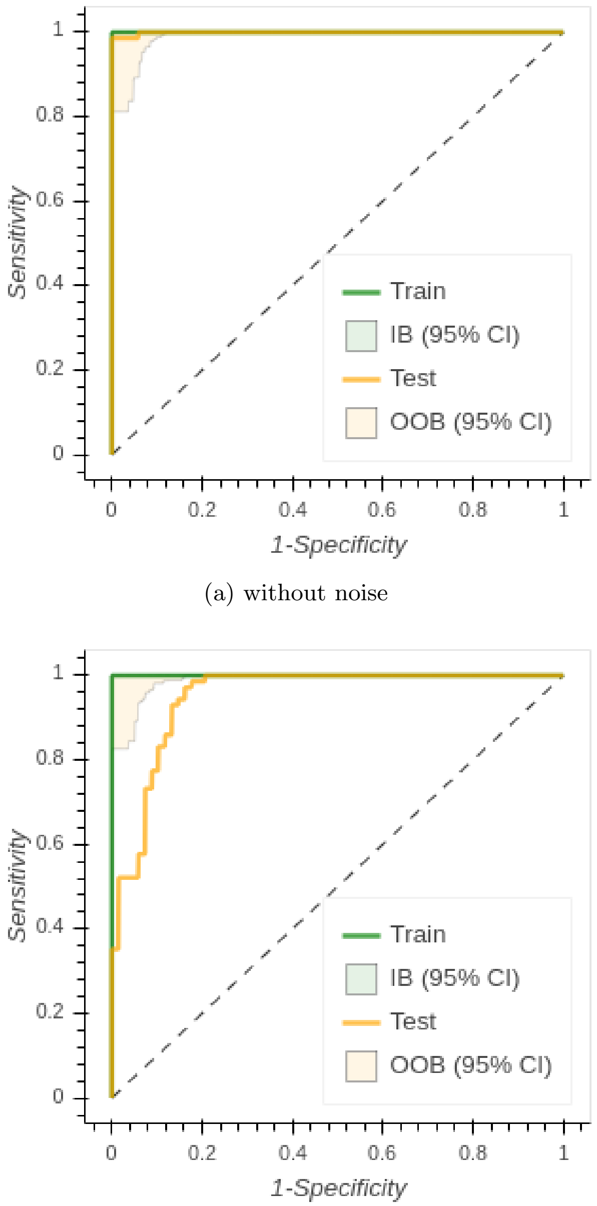
When we look at our top features from table 1 (RF classifier with Gaussian noise) and analyze their biological roles and functions, especially their roles in cancer biology, this is what we came across: Low blood levels of coenzyme Q10 have been detected in patients with some types of cancer. SLMAP has low cancer specificity. Multiple members of the SLC16 family could be used as prognostic indicators for many tumors, and were associated with immune invasion and tumor stem cells (pan cancer analysis results) [5]. IFNB1 expression in human breast carcinomas was found to inversely correlate with recurrence free survival (RFS) [9]. Carboxylesterase 1 (CES1) is expressed at various levels in Hepatocellular carcinoma (HCC) [4]. ILF3 is involved in the deterioration of GC by promoting proliferation of GC cells, and ILF3 protein detection may assist in the prediction of the prognosis of patients with gastric cancer (GC) [8].
And here are the top features from 2 (RF classifier without Gaussian noise) and their roles in cancer biology: The therapeutic intervention of the TAPBPL inhibitory pathway may represent a new strategy to modulate T cell-mediated immunity for the treatment of cancer, infections, autoimmune diseases, and transplant rejection [7]. NPAT has low cancer specificity TRBJ2-7 (T Cell Receptor Beta Joining 2-7): This gene is involved in T-cell receptor rearrangement, which is essential for immune responses against cancer cells. PLD4 promotes M1 macrophages to perform antitumor effects in colon cancer cells, furthermore, its expression was associated with clinical staging of colon cancer [3].

Order	**Feature name**	**Gene name**	**Description**	**Indices**
1	ENSG00000115520	COQ10B	Coenzyme Q-Binding Protein COQ10 Homolog B, Mitochondrial	**4474**
2	ENSG00000163681	SLMAP	Sarcolemma Associated Protein	**11204**
3	ENSG00000160633	SAFB	Scaffold Attachment Factor B	**10604**
4	ENSG00000168679	SLC16A4	Solute Carrier Family 16 Member 4	**12540**
5	ENSG00000122203	KIAA1191	KIAA1191	**5281**
6	ENSG00000229163	NAP1L1P2	Nucleosome Assembly Protein 1 Like 1 Pseudogene 2	**30314**
7	ENSG00000108296	no results	no results	**3536**
8	ENSG00000163710	PCOLCE2	Procollagen C-Endopeptidase Enhancer 2	**11218**
9	ENSG00000152595	MEPE	Matrix Extracellular Phosphoglycoprotein	**9642**
10	ENSG00000171855	IFNB1	Interferon Beta 1	**13301**
11	ENSG00000198848	CES1	Carboxylesterase 1	**18281**
12	ENSG00000089006	SNX5	Sorting Nexin 5	**1821**
13	ENSG00000149308	NPAT	Nuclear Protein, Coactivator Of Histone Transcription	**9270**
14	ENSG00000133597	ADCK2	AarF Domain Containing Kinase 2	**6821**
15	ENSG00000142920	AZIN2	Antizyme Inhibitor 2	**8360**
16	ENSG00000129351	ILF3	Interleukin Enhancer Binding Factor 3	**6178**
17	ENSG00000127311	HELB	DNA Helicase B	**5941**
18	ENSG00000155640	no results	no results	**9976**

Table 1: Top 18 Important features from bootstrap evaluation of training and testing datasets for the best RF with Gaussian noise [1]

Order	**Feature name**	**Gene name**	**Description**	**Indices**
1	ENSG00000181101	ENTR1P2	ENTR1 Pseudogene 2	**15194**
2	ENSG00000229163	NAP1L1P2	Nucleosome Assembly Protein 1 Like 1 Pseudogene 2	**30314**
3	ENSG00000139192	TAPBPL	TAP Binding Protein Like	**7847**
4	ENSG00000142539	-	Novel Protein	**8312**
5	ENSG00000186583	SPATC1	Spermatogenesis And Centriole Associated 1	**16557**
6	ENSG00000228650	-	Novel Transcript	**29961**
7	ENSG00000149308	NPAT	Nuclear Protein, Coactivator Of Histone Transcription	**9270**
8	ENSG00000211771	TRBJ2-7	T Cell Receptor Beta Joining 2-7	**22294**
9	ENSG00000166428	PLD4	Phospholipase D Family Member 4	**11966**
10	ENSG00000110245	APOC3	Apolipoprotein C3	**3794**
11	ENSG00000155640	no results	no results	**9976**
12	ENSG00000132481	TRIM47	Tripartite Motif Containing 47	**6651**
13	ENSG00000171161	ZNF672	Zinc Finger Protein 672	**13132**
14	ENSG00000196196	HRCT1	Histidine Rich Carboxyl Terminus 1	**17324**
15	ENSG00000186260	MRTFB	Myocardin Related Transcription Factor B	**16470**
16	ENSG00000112984	KIF20A	Kinesin Family Member 20A	**4162**
17	ENSG00000163491	NEK10	NIMA Related Kinase 10	**11129**
18	ENSG00000171855	IFNB1	Interferon Beta 1	**13301**

Table 2: Top 18 Important features from bootstrap evaluation of training and testing datasets for the best RF classifier without Gaussian noise [1]



(a) without noise

(b) with noise

Figure 3: Receiver Operator Characteristic (ROC) Curve using Random Forest Classifier a) without noise b) with Gaussian noise. The green color indicates the in-bag (IB) 95 % confidence interval and the yellow color indicates the out-of-bag (OOB) 95 % confidence interval

6 Contributions

Sujan Darai: Running and implementing code, writing abstract, data & preprocessing
Matanat Mammadli: Running code, writing Introduction, Goal of the project, Methods, Results & discussion
Samra Hamidovic: Running code, writing Abstract, Introduction, Results
Ibrahim Ibraimi:

7 Appendix

References

1. GeneCards - Human Genes | Gene Database | Gene Search.
2. Myron G. Best, Nik Sol, Irsan Kooi, Jihane Tannous, Bart A. Westerman, François Rustenburg, Pepijn Schellen, Heleen Verschueren, Edward Post, Jan Koster, Bauke Ylstra, Najim Ameziane, Josephine Dorsman, Egbert F. Smit, Henk M. Verheul, David P. Noske, Jaap C. Reijneveld, R. Jonas A. Nilsson, Bakhos A. Tannous, Pieter Wesseling, and Thomas Wurdinger. RNA-Seq of Tumor-Educated Platelets Enables Blood-Based Pan-Cancer, Multiclass, and Molecular Pathway Cancer Diagnostics. *Cancer Cell*, 28(5):666–676, November 2015. Publisher: Elsevier.
3. Long Gao, Yan Zhou, Shu-Xian Zhou, Xian-Jing Yu, Jin-Mei Xu, Luo Zuo, Yong-Hui Luo, and Xiao-An Li. PLD4 promotes M1 macrophages to perform antitumor effects in colon cancer cells. *Oncology Reports*, 37(1):408–416, January 2017.
4. Gang Li, Xin Li, Iqbal Mahmud, Jazmin Ysaguirre, Baharan Fekry, Shuyue Wang, Bo Wei, Kristin L. Eckel-Mahan, Philip L. Lorenzi, Richard Lehner, and Kai Sun. Interfering with lipid metabolism through targeting CES1 sensitizes hepatocellular carcinoma for chemotherapy. *JCI Insight*, 8(2):e163624.
5. Jun Li, Jiaheng Xie, Dan Wu, Liang Chen, Zetian Gong, Rui Wu, Yiming Hu, Jiangning Zhao, and Yetao Xu. A pan-cancer analysis revealed the role of the SLC16 family in cancer. *Channels*, 15(1):528–540.
6. Liesbet Lieben. Cancer genetics: RNA-seq for blood-based pan-cancer diagnostics. *Nature Reviews. Genetics*, 16(12):688, December 2015.
7. Yujun Lin, Cheng Cui, Min Su, Lawrence K Silbart, Haiyan Liu, Jin Zhao, Lang He, Yuanmao Huang, Dexin Xu, Xiaodan Wei, Qian Du, and Laijun Lai. Identification of TAPBPL as a novel negative regulator of T-cell function. *EMBO Molecular Medicine*, 13(5):e13404, May 2021.
8. Yü Liu, Yong Li, Yijie Zhao, Yang Liu, Liqiao Fan, Nan Jia, and Qun Zhao. ILF3 promotes gastric cancer proliferation and may be used as a prognostic marker. *Molecular Medicine Reports*, 20(1):125–134, July 2019.
9. Ana Maia, Zuguang Gu, André Koch, Mireia Berdiel-Acer, Rainer Will, Matthias Schlesner, and Stefan Wiemann. IFN1 secreted by breast cancer cells undergoing chemotherapy reprograms stromal fibroblasts to support tumour growth after treatment. *Molecular Oncology*, 15(5):1308–1329, May 2021.
10. Yu-Hang Zhang, Tao Huang, Lei Chen, YaoChen Xu, Yu Hu, Lan-Dian Hu, Yudong Cai, and Xiangyin Kong. Identifying and analyzing different cancer subtypes using RNA-seq data of blood platelets. *Oncotarget*, 8(50):87494–87511, September 2017. Publisher: Impact Journals.