Urinary Biomarkers for Predicting Pancreatic Cancer

Lisa J.B. Hu (414264)
Bioinformatics - BFV3
Institute of Life Sciences & Technology
Hanze University of Applied Sciences
Dave Langers
Bart Barnard
November 14, 2022

Urinary Biomarkers for Pancreatic Cancer

Lisa Hu
414264
Bio-Informatics
Hanzehogeschool Groningen, ILST
Dave Langers (LADR) & Bart Barnard (BABA)
November 14, 2022

Preface

Abstract

List of Abbreviations

- **EDA**: exploratory data analysis
- **FN**: false negatives
- **FP**: false positives
- **FPR**: false positive rate
- **PDAC**: pancreatic ductal adenocarcinoma
- **ROC**: receiver operating characteristic
- **SN**: sensitivity
- **SP**: specificity
- **TN**: true negatives
- **TNR**: true negative rate
- **TP**: true positives
- **TPR**: true positive rate

Contents

1 Introduction

Pancreatic cancer is one of the deadliest variations of cancer with 5-year survival rates of below 20%. [1] Pancreatic ductal adenocarcinoma is mostly diagnosed when in its later stages as there are almost no useful biomarkers for detection in the earlier stages. Serum CA19.9 has been shown to be promising in the detection and is the only biomarker in clinical practice, but is not sensitive or specific enough to use for screening. [2] Other methods of cancer research, such as a biopsy, are rather invasive and usually done when it is already in the later stages.

While traditionally blood is the main source for biomarker analysis, urine presents alternatives. This method enables non-invasive sampling, easier repeated collections and a higher volume of samples. Not only is it beneficial for the patient: urine contains a higher concentration of the biomarkers due to the continuous ultrafiltration of the blood. [3]

To prevent patients getting diagnosed with PDAC in the later stages: Could a Machine Learning model be build to predict the patient its diagnose based on the urine samples' biomarkers?

2 Materials

To analyse the different biomarkers, an exploratory data analysis (EDA) was build in R. For this analysis, various packages were used:

Table 1: Software and packages

Software	**Package**	**Version**
R [4]		4.2.1
	tidyr [5]	1.2.1
	dplyr [6]	1.0.10
	pander [7]	0.6.5
	readr [8]	2.1.3
	FSA [9]	0.9.3
	ggplot2 [10]	3.3.5
	ggpubr [11]	0.4.0
	ggnewscale [12]	1.1.1
	RWeka [13]	0.4-44
Weka [14]		3.8.6
	thresholdSelector [15]	1.0.3

3 Methods

3.1 Quality metrics

To create a fitting model, different quality metrics need to be taken in account to optimize the output. The most evident metric is the accuracy. Because accuracy is not insightful enough to distinguish the performance of the different algorithms, other quality metrics need to be analysed as well.

One of those quality metrics is a confusion matrix. This table layout allows visualization on the performance of an algorithm. Table 2 shows an example of a confusion matrix.

Table 2: Example of a confusion matrix: TP = true positives; FN = false negatives; FP = false positives; TN = true negatives

a	b	<- classified as
TP	FN	a = Healthy
FP	TN	b = Malignant

In this confusion matrix, the so-called "hits" are the true positives (TP) whereas the "correct rejections" are the true negatives (TN). The falsely predicted instances are either false negatives (FN) or false positives (FP). In this case, FN imply the "Healthy" instances being predicted as "Malignant" and the FP imply the "Malignant" instances being predicted as "Healthy". For the creation of this model, it is important the FP is as low as possible while getting the highest possible outcomes for TN.

This leads to the next quality metrics: sensitivity - or true positive rate (TPR) -, the false positive rate (FPR) and the true negative rate (TNR). The TPR is calculated as $\frac{TP}{TP+FN}$ and the FPR as $\frac{FP}{FP+TN}$. The TNR can be derived from the FPR: $1 - FPR$. These different rates describe the probability the algorithm will classify it as. This model seeks a high TPR and low FPR, thus a high TNR.

Another noting quality metric is the receiver operating characteristic (ROC) curve. By plotting the TPR against the FPR, a curve is created and can tell the performance of an algorithm. The further the curvature is pushed to the top-left corner, the better the classifier model.

4 Results

4.1 Demographics

The patients were divided by diagnosis and the PDAC patients are further seperated by the stage of the disease. The number of samples per diagnosis and stage are shown in Table 3.

Table 3: Demographics of the samples. All values are the respective amounts.

Sample type	**Control**		**Benign**		**PDAC**		
	Sample	Sex	Sample	Sex	Sample	Sex	Cancer stage
Urine (n=590)	183	F = 115 M = 68	208	F = 101 M = 107	209	F = 83 M = 116	I-IIA = 27 II-IIB = 75 III = 76 IV = 21
Plasma (n=350)	92	F = 58 M = 34	108	F = 57 M = 51	150	F = 86 M = 64	I-IIA = 20 II-IIB = 60 III = 65 IV = 5

4.2 REG1B outperforms REG1A in detecting early stage PDAC

Though the performance of REG1A and REG1B are very similar, REG1B outperformed REG1A when control and benign samples were compared to stage I-IIA PDAC samples (Kruskal-Wallis test; $p < 0.001$ & $p < 0.0002$). [Table 4] Therefor, all experiments following were performed using REG1B as part of the biomarker panel.

Table 4: Adjusted p-values of Kruskal-Wallis test, Dunn's multiple comparisons; ns - not significant. The header shows the groups that were compared. (Table continues below)

	Control - I-II	Control - I-IIA	Control - III-IV
REG1A	1.928479e-05	ns	4.837915e-07
REG1B	3.864924e-15	0.0002123534	5.789369e-17

	Benign - I-II	Benign - I-IIA	Benign - III-IV
REG1A	0.000768779	ns	4.494778e-05
REG1B	1.200471e-12	0.001777207	3.927231e-14

4.3 Correlation in urine biomarkers for different diagnosis groups

The biomarker panel was tested in a total of 590 urine samples (183 control, 208 benign, and 199 PDAC). According to the PCA, the LYVE1 and REG1B biomarkers are close related to each other. Aside from that, the different diagnosis groups create clusters, meaning there is significant difference between the samples.

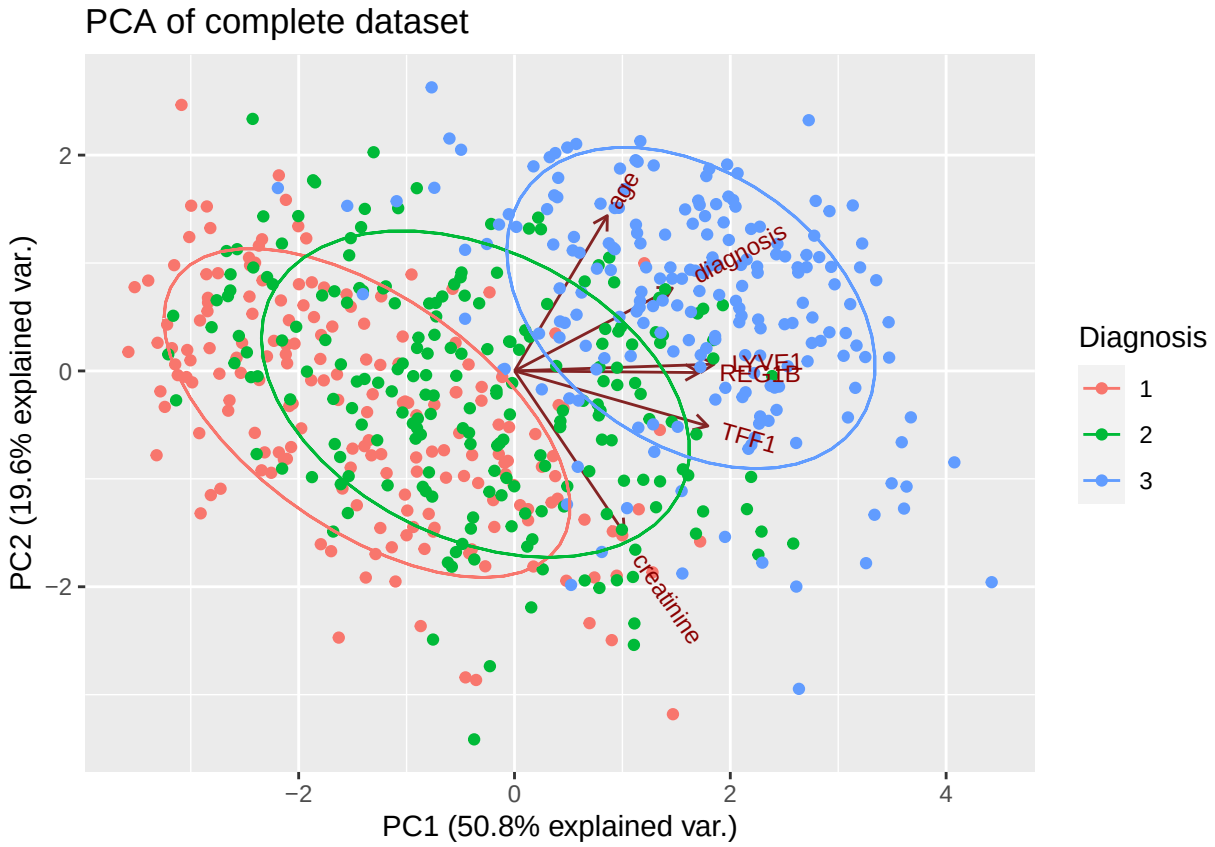


Figure 1: PCA plot showing the correlations between the biomarkers. Data is log transformed.

4.4 Performance of the different possible algorithms

To create a fitting model, multiple algorithms are run over the samples. Table 6 shows the important results of the different classifiers.

Table 6: Results of the different algorithms from Weka

Algorithm	Accuracy	Sensitivity	FPR	AUROC
ZeroR	35.25	0	0	0.5
OneR	49.51	0.4612	0.2047	0.6283
NaiveBayes	60.07	0.5645	0.1982	0.8165
Logistic	65.2	0.528	0.1551	0.8173
SimpleLogistic	64.31	0.5296	0.1629	0.814
SMO	62.81	0.4258	0.1165	0.7721
IBk	52.37	0.3905	0.1085	0.641
J48	59.08	0.5536	0.1935	0.7674
RandomForest	65.61	0.6581	0.1601	0.8502

These results show a relative low accuracy and sensitivity. Some algorithms also have a low ROC value, putting the cutoff at 0.8: OneR, SMO, IBk and J48 will not be used. As for the remaining three: NaiveBayes has by far the lowest accuracy of them and is therefor also dropped. Leaving the options Logistic and RandomForest. Since earlier shown there is a linear correlation between the different variables, the Logistic algorithm would be more fitting for this type of data.

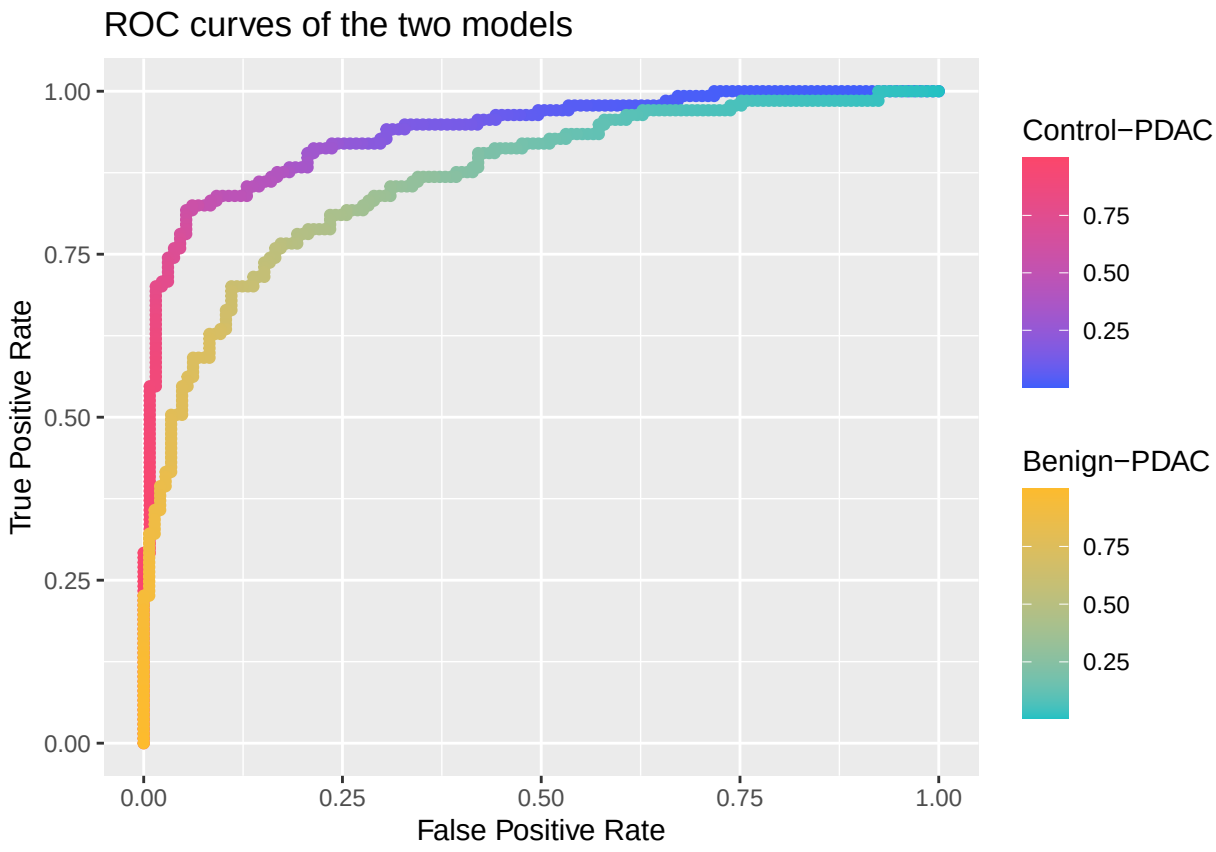
4.5 Algorithm optimization

Table 7: Results of the ThresholdSelector with different thresholds (Control vs. PDAC).

Algorithm (threshold)	Accuracy	TPR	FPR
Logistic (-)	85.23	0.8454	0.1412
ThresholdSelector (0.550)	86.39	0.8844	0.1558
ThresholdSelector (0.575)	87.1	0.9027	0.1594
ThresholdSelector (0.600)	87.33	0.9181	0.1695
ThresholdSelector (0.625)	87.43	0.9303	0.1792
ThresholdSelector (0.650)	87.18	0.941	0.1944
ThresholdSelector (0.675)	86.62	0.9464	0.2103

Table 8: Results of the ThresholdSelector with different thresholds (Benign vs. PDAC).

Algorithm (threshold)	Accuracy	TPR	FPR
Logistic (-)	79.23	0.794	0.2108
ThresholdSelector (0.550)	80.07	0.8264	0.2276
ThresholdSelector (0.575)	80.22	0.8443	0.2436
ThresholdSelector (0.600)	80.04	0.8581	0.2618
ThresholdSelector (0.625)	79.51	0.8636	0.2786
ThresholdSelector (0.650)	79.27	0.8775	0.2981
ThresholdSelector (0.675)	78.8	0.8893	0.3201



5 Conclusion

- REG1A shows no significant difference when control samples are compared to PDAC stage I-IIA samples, though REG1B does show significance: REG1B outperforms REG1A in detecting PDAC in early stages.
- LYVE1 and REG1B are closely related biomarkers. A good focus for the prediction following.
-

6 Discussion

Though REG1A is a viable biomarker, the data delivered contained a lot of missing values for this biomarker. Data imputation was not an option since that would cause an imbalance. Therefor it was left out for the creation of the model.

Plasma CA19.9 is a blood biomarker and not a urine biomarker. This model did include the data from the plasma CA19.9 results, but it can still be used without a CA19.9 sample.

Keep in mind when using the model, all data is log transformed.

7 References

1. Public Health England (2021). Cancer survival in ENgland between 2014 and 2018. https://www.gov.uk/government/statistics/cancer-survival-in-england-for-patients-diagnosed-between-2014-and-2018-and-followed-up-until-2019/cancer-survival-in-england-for-patients-diagnosed-between-2014-and-2018-and-followed-up-to-2019
2. Ballehaninna, U., & Chamberlain, R. (2011). The clinical utility of serum CA 19-9 in the diagnosis, prognosis and management of pancreatic adenocarcinoma: An evidence based appraisal. *Journal Of Gastrointestinal Oncology, 3*(2), 105-119. doi:10.3978/j.issn.2078-6891.2011.021
3. Thongboonkerd, V. (2007), Recent progress in urinary proteomics. Prot. Clin. Appl., 1: 780-791. https://doi.org/10.1002/prca.200700035
4. R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
5. Wickham H, Girlich M (2022). *tidyr: Tidy Messy Data*. R package version 1.2.1, https://CRAN.R-project.org/package=tidyr
6. Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.10, https://CRAN.R-project.org/package=dplyr
7. Daróczi G, Tsegelskyi R (2022). *pander: An R 'Pandoc' Writer*. R package version 0.6.5,https://CRAN.R-project.org/package=pander
8. Wickham H, Hester J, Bryan J (2022). *readr: Read Rectangular Text Data*. R package version 2.1.3,https://CRAN.R-project.org/package=readr
9. Ogle, D.H., J.C. Doll, P. Wheeler, and A. Dinno. 2022. FSA: Fisheries Stock Analysis. R package version 0.9.3, https://github.com/fishR-Core-Team/FSA
10. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016. https://ggplot2.tidyverse.org/
11. Kassambara A (2020). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0,https://CRAN.R-project.org/package=ggpubr
12. Campitelli E (2022). *ggnewscale: Multiple Fill and Colour Scales in 'ggplot2'*. R package version 0.4.8, https://CRAN.R-project.org/package=ggnewscale
13. Hornik K, Buchta C, Zeileis A (2009). "Open-Source Machine Learning: R Meets Weka." *Computational Statistics*, *24*(2), 225-232. doi:10.1007/s00180-008-0119-7 https://doi.org/10.1007/s00180-008-0119-7
14. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.
15. Eibe Frank (2013) thresholdSelector metaclassifier. https://weka.sourceforge.io/doc.packages/thresholdSelector/weka/classifiers/meta/ThresholdSelector.html