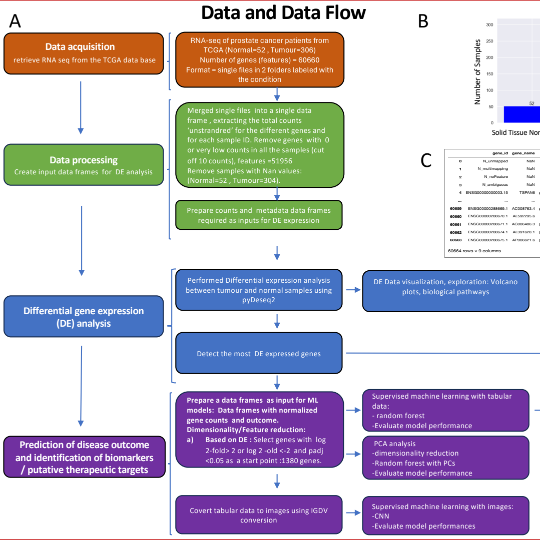
# Prostate Cancer Prediction and Biomarker Identification Using Machine Learning and Deep Learning Algorithms on Transcriptome Data from The Cancer Genome Atlas (TCGA) Database

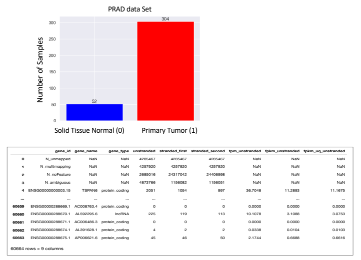
## ABSTRACT


The search for novel RNA biomarkers and innovative methods to identify cancerous tissues can significantly advance the development of RNA-based diagnostic and therapeutic strategies, leading to more effective and personalized approaches for cancer treatment and management. In this project, we investigated the feasibility of predicting or diagnosing prostate cancer, which ranks among the most prevalent cancers in the male population, by applying machine learning (ML) and convolutional neural network (CNN) algorithms to gene expression data of normal and primary tumor prostate gland samples. Genes/features used as input for ML were reduced by preselecting the most differentially expressed (DE) genes between cancer and normal samples. Machine learning algorithms (logistic regression, random forest, random forest on the most important principal components (PCs)) were applied to predict cancer outcomes using RNA expression data on the selected genes. A CNN was also tested on the same tabular data converted to images. Moreover, through an examination of the disturbed gene expression patterns in prostate cancer samples and the genes important for predicting cancer versus normal tissue outcomes by machine learning, we also set up up to discover putative novel RNA biomarkers for prostate cancer.


## RESULTS

### Data and Data Flow

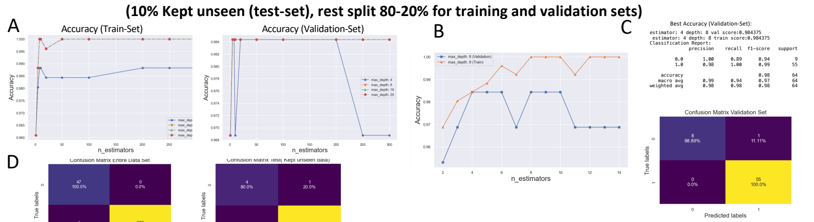


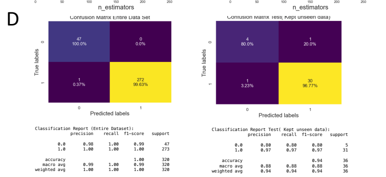




(A) Flow chart of the research process. (B) Plot showing the number of RNAseq samples retrieved for each category, Solid Normal Tissue and prostate Primary Tumor tissues. (C) Example of a single sequencing file data from a Prostate Primary tumor sample. The first rows 4 rows depict the total expression counts. The expression profile of each of the 60.660 genes detected in the RNA seq analysis is depicted in the rest of the rows

### Random Forest Classifier with Single Random Split

#### (10% Kept unseen (test-set), rest split 80-20% for training and validation sets)

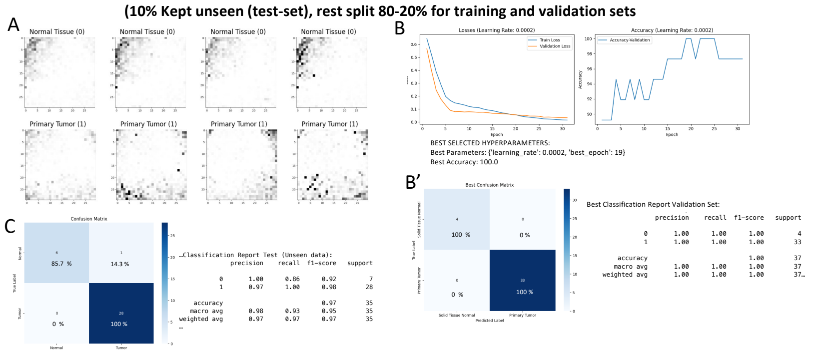


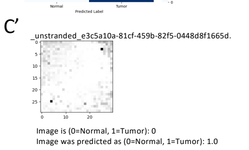




(A-B) Model tuning involved varying the number of estimators (n_estimators) and maximum tree depth (max_depth) during training and validation. (C) The classification report and confusion matrix for the validation set using the best hyperparameters (n_estimators=4, max_depth=8) is shown. (D) The model was retrained with these hyperparameters on the entire dataset (training plus validation) and then tested on the kept unseen test data. Confusion matrixes and classificaiton reports are depicted. Normal tissue samples are labeled as 0, and prostate primary tumor samples are labeled as 1.

### Convolutional Neuronal network (CNN) with Tabular data converted to Images

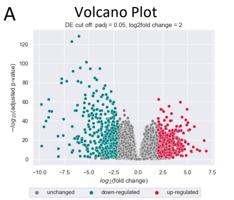
#### (10% Kept unseen (test-set), rest split 80-20% for training and validation sets



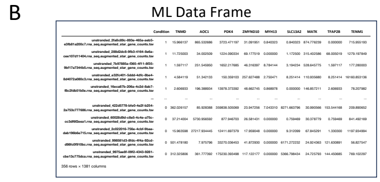




(A) Tabular RNA-seq data (derived from the preselected 1380 features) was transformed into images using the IGTD algorithm (Zhu *et al.* 2021). Random examples of the generated images for Normal Tissue (category 0) and Primary Prostate Tumor samples (category 1) are displayed. (B) A CNN was applied to the generated images from (A). The CNN architecture comprised four convolutional neural networks with ReLU activation, followed by one max-pooling layer and further flattening. The flattened data were fed into a linear layer with ReLU activation and then into a final linear layer with sigmoid activation and one output neuron. The sigmoid activation provided probabilities for each class directly. The losses in the training and validation sets and the accuracy of the validation set over the number of epochs for the best hyperparameters (lr: 0.0002, best_epoch: 19) are displayed after testing different learning rates for a single run. (B') The classification report and the confusion matrix for this run under these best hyperparameters are presented. (C) The CNN was retrained with the entire dataset (validation and train sets), using the best hyperparameters. This model was then tested on the 10% of data kept unseen. The confusion matrix and classification report for this test set for this specific run are shown. The model could improve the recall in the normal samples on unseen data to 86% (14% false positives). (C') Image and file name of the incorrectly predicted sample for this specific run is shown.

### Differential Expression (DE) Analysis and Machine learning Data preparation




Total genes (features): 51956
Number of DE genes: 1380
Number of up-regulated genes in Tumors: 725
Number of down-regulated genes in Tumors: 655
Number of unchanged genes: 50576

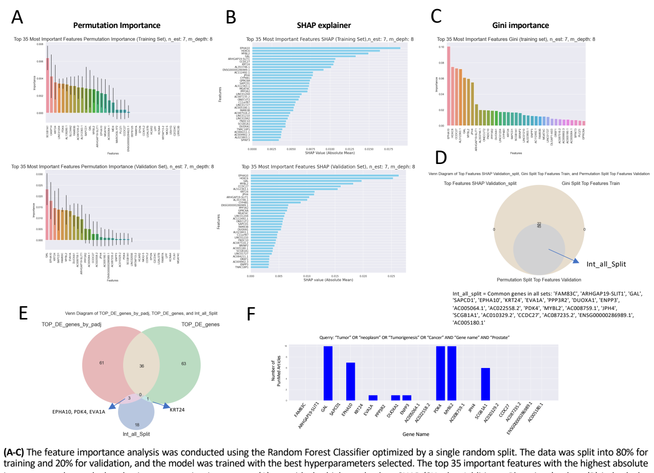



(A) Volcano plot displaying the log2 FoldChange value versus the -log10 of the padj value for all genes with reads. A cutoff of an absolute log2 FoldChange value of 2 and a padj value < 0.05 was selected. (B) Data Frame generated for Machine learning input. Sample IDs are in the rows. The column labeled "Condition" depicts the condition of each sample, represented with zeros (0) for normal tissue samples and ones (1) for primary prostate tumor samples. Gene names are depicted in the columns. Column values represent normalized counts of RNA expression.

### Performance of the Different Models Tested

| Data / Features | Model | Traning and evaluation | Accuracy Validation set | Recall Validaton set | Accuracy Test set ( unseen data) | Recall Test set (unseen data) |
|---|---|---|---|---|---|---|
| -Tabular data -Features-reduction from DE analysis: log 2-fold> 2 or log 2 -old <-2 and padj <0.05 : 1380 genes | Logistic Regression | 35 itinerations | 0.931 | 0.82(0), 0.95 (1) | | |
| | | 50 itinerations (keep 10% unseen) | 0.953 | 0.78(0),0.98(1) | 0.94 | 0.6 (0), 1(1) |
| | Random Forest | Grid-Search-single-random-split | 0.986 | 0.91 (0), 1 (1) | | |
| | | Grid-Search- single-random-split (keep 10% unseen) | 0.984 | 0.89 (0), 1 (1) | 0.94 | 0.8 (0), 0.97 (1) |
| | | Crossvalidation | 0.955 (mean CV score) 0.971 (median CV score) | 0.8 (0), 1(1) (median) | | |
| | | Crossvalidation (keep 10% unseen) | 0.956 (mean CV score) 0.969 (median CV score) | 0.8 (0), 1(1) (median) | 0.94 | 0.8 (0), 0.97 (1) |
| -Tabular data -Features reduction by DE follow by PCA analyis - Most important PCs used for ML. -PCA1,2 | Random Forest | Grid-Search-single random-split | 0.972 | 0.91(0), 0.98(1) | | |
| | | Grid-Search- single-random-split (keep 10% unseen) | 0.984 | 0.89(0),1(1) | 0.94 | 0.8 (0), 0.97 (1) |
| | | Crossvalidation | 0.958 (mean CV score) 0.971 (median CV score) | 0.82(0), 0.97(1) (median) | | |
| | | Crossvalidation (keep 10% unseen) | 0.966 (Mean CV score) 0.969 (Median CV score) | 1(0), 0.98 (1) (median) | 0.94 | 0.8 (0), 0.97 (1) |
| - Images (Tabular data(1380 features)-image conversion. | CNN | Grid-Search 4CNNs-Relu activation | 0.967 +/- 0.018 | 0.945 +/- 0.071 (0), 0.9717 +/- 0.07(1) (mean 6 runs) | | |
| | | Grid-Search 4CNNs-Than activation | 0.874 +/- 0.054 (mean 6 runs) | 0.147 +/- 0.3 (0), 0.997 +/- 0.008 (1) (mean 6 runs) | | |
| | | Grid-Search 4CNNs-Relu activation Keep 10% unseen | 0.978 +/- 0.021 (mean 6 runs) | 0.908 +/- 0.089 (0), 0.978 +/- 0.02(1) (mean 5 runs) | **0.952 +/- 0.016** (mean 5 runs) | **0.886 +/- 0.072 (0), 0.974 +/- 0.026 (1)** (mean 5 runs) |

### Feature Importance Analysis to identify Biomarkers or Therapeutic Targets




(A-C) The feature importance analysis was conducted using the Random Forest Classifier optimized by a single random split. The data was split into 80% for training and 20% for validation, and the model was trained with the best hyperparameters selected. The top 35 important features with the highest absolute importance values calculated using permutation importance (A) or with the highest absolute SHAP (SHapley Additive exPlanations) values (B) in both the training and validation sets are displayed. (C) The top 35 features with the highest absolute importance values calculated using Gini importance values are shown for the training set. (D) Features/genes with absolute importanc values greater than 0 were selected for further analysis. Venn diagrams illustrate the intersection between the genes found important for model prediction through the different tests conducted in (A-C). (E) Venn diagrams depict the intersection of the 100 top DE expressed genes selected based on either higher log2FoldChange (Top_DE_genes) or lowest padj value (Top_DE_genes_by_padj) with the 22 genes found importance by the 3 algorithms (Int_all_split). (F) The number of publications (up to 10) found for each gene in PubMed containing the gene name and cancer-related terms in the abstract or title of the publication. Among the lists (Int_all_Split and, Top_features_validation_SHAP_split) there are several genes already known to be involved in prostate cancer. Additionally, the script is capable of retrieving the title, abstract, and citation for these publications.

## CONCLUSIONS AND OUTLOOK

- Machine learning applied to RNAseq data has successfully predicted prostate cancer outcomes.
- Random forest outperformed logistic regression, enhancing recall for under-represented normal tissue.
- PCA feature reduction was effective; 2 PCs matched RF performance with 1,380 features.
- Transforming tabular data into images for a CNN improved model performance, particularly recall for the underrepresented category; visualization provided insights not easily discernible from 1,380 tabular features.
- Main issues: unbalanced, limited data and no accessible independent dataset for final validation. While models showed high accuracy, they struggled with underrepresented normal samples but excelled in classifying tumor samples.
- Stratified splitting improved Random Forest performance on underrepresented samples. Further enhancement of CNN could be achieved with stratification and cross-validation.
- Generating synthetic RNA-seq data and utilizing independent datasets is recommended.
- Optimizing Random Forest by adjusting hyperparameters (min_samples_split, min_samples_leaf, max_features) is advised to boost stability, reduce overfitting, and enhance performance..