Final Report-Paula Vazquez

paulavazq@gmail.com

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 in cancer 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) were applied to predict cancer outcomes using gene expression tabular 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 to discover putative novel RNA biomarkers for prostate cancer. Gene Ontology analysis of these genes and the DE genes was also conducted to enhance our comprehension of the pathways disrupted or contributing to disease progression.

INTRODUCTION

Prostate cancer stands as the second leading cause of cancer-related death for men globally, second only to lung cancer (1,2). Early diagnosis plays a pivotal role in the prevention and successful treatment of patients with prostate tumors. Nowadays, the gold standard for prostate cancer diagnosis involves a prostate biopsy performed after a previous clinical suspicion based on prostate-specific antigen (PSA) levels and digital rectal examination (DRE) (3). However, concerns persist regarding the reduced sensitivity of DER and the low specificity of PSA. Therefore, the identification of novel biomarkers and targets linked to these diseases is crucial for enhancing diagnostic methods, developing personalized therapies, and predicting treatment outcomes.

Over the past two decades, an increasing amount of 'omics' data has become publicly available. The Cancer Genome Atlas (TCGA) database houses comprehensive information on the molecular attributes of over 20,000 primary cancer and matched normal samples, encompassing 33 types of cancer (4). This is a joint effort between NCI and the National Human Genome Research Institute that began in 2006, bringing together researchers from diverse disciplines and multiple institutions. The TCGA generated over 2.5 petabytes of genomic, epigenomic, transcriptomic, and proteomic data. The data is publicly available for anyone in the research community to use. One of the key components of this database is the transcriptome profiles, representing genes actively transcribed into RNA within the samples. Perturbations in RNA expression levels are frequently observed in various diseases, including cancer. Consequently, RNAs exhibiting differential expression in cancer cells compared to normal cells can serve as valuable diagnostic biomarkers or therapeutic targets.

In this project, machine learning (ML) and convolutional neural network (CNN) algorithms were employed to assess whether RNA expression levels can predict prostate cancer outcomes. We further aimed to identify cancer-type-specific biomarkers for prostate cancer by combining differential expression (DE) analysis and ML algorithms on transcriptome profile data of cancer and normal samples. Additionally, we explored the genes and molecular pathways activated or inactivated in prostate cancer by identifying the most differentially expressed genes (DE) in tumors and corresponding normal tissue samples, combined with an analysis of the genes deemed most important for model prediction. The data flow of the entire project is shown in Figure 1A.

DATA

Data retrieval

The RNA seq data used in this project is available on the Cancer Genome Atlas (TCGA) web page (4). For this project, RNA sequencing data from 306 prostate Primary tumor samples, along with 52 solid normal tissue control samples, was retrieved from the TCGA database.

The data retrieval script (5, see methods and corresponding **Jupyter notebook called: Tcga downloader**) categorized the data into distinct folders based on clinical outcome classifications. Each folder contains individual TSV files, representing data from each specific sample (an example of a single file is shown in Table 1).

Example of reads i sigle file: DataMiner-main/PRAD/Primary_Tumor/0a9f83f1-86f9-4153-b8cf-07e8bcb18efb.rna_seq.augmented_star_gene_counts.tsv

Out[69]:

	gene_id	gene_name	gene_type	unstranded	stranded_first	stranded_second	tpm_unstranded	fpkm_unstranded	fpkm_uq_unstranded
0	N_unmapped	NaN	NaN	4285467	4285467	4285467	NaN	NaN	NaN
1	N_multimapping	NaN	NaN	4257920	4257920	4257920	NaN	NaN	NaN
2	N_noFeature	NaN	NaN	2685016	24317042	24406998	NaN	NaN	NaN
3	N_ambiguous	NaN	NaN	4873766	1156082	1156051	NaN	NaN	NaN
4	ENSG00000000003.15	TSPAN6	protein_coding	2051	1054	997	36.7048	11.2893	11.1675
...	...	...	...	...	...	...	...	...	...
60659	ENSG00000288669.1	AC008763.4	protein_coding	0	0	0	0.0000	0.0000	0.0000
60660	ENSG00000288670.1	AL592295.6	lncRNA	225	119	113	10.1078	3.1088	3.0753
60661	ENSG00000288671.1	AC006486.3	protein_coding	0	0	0	0.0000	0.0000	0.0000
60662	ENSG00000288674.1	AL391628.1	protein_coding	4	2	2	0.0338	0.0104	0.0103
60663	ENSG00000288675.1	AP006621.6	protein_coding	45	46	50	2.1744	0.6688	0.6616

60664 rows × 9 columns

Table 1. 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.

The 4 initial rows of the data represent the total counts for mapped, multi-mapped, and unmapped sequencing reads in the sample with reference to the genome. These rows were not used in the analysis. Following this, the data provides details in the following format:

- **Gene_Id:** Ensembl ID gene identifier.
- **Gene_name**: Annotated name of the gene**.**
- **Gene_type**: Type of gene categorized based on from the Ensembl biotype annotation, accessible at: http://www.ensembl.org/info/genome/genebuild/biotypes.html.
- **Unstranded**: Total raw reads mapped to each gene using an Unstranted library preparation method. This approach does not retain information about the

originating RNA strand, making it valuable for quantifying overall gene expression levels.

- **Stranded_first**: Number of raw reads mapped to the first cDNA strand generated during library preparation using a Stranded RNA-seq method.
- **Stranded_second:** Number of raw reads mapped to the second cDNA strand generated in a Stranded RNA-seq method.
- **Tpm-unstranded** (Transcripts per million): Tpm is a measure of gene expression normalized by the total number of transcripts in the sample and scaled to a million.
- **Fpkm- unstranded** (Fragments per kilobase of transcript per million mapped reads): Fpkm is another metric used to quantify gene expression levels. It takes into account the length of the gene and the total number of fragments (reads) that map to that gene.
- **Fpkm-uq-unstranded (Upper quartile fpkm):** Fpkm-uq is a modified version of fpkm that is normalized by the upper quartile of the fpkm values across all genes in a given sample.

Metadata retrieval

In the context of RNA sequencing data or transcriptomics, metadata serve to describe the sample from which the RNA sequence was derived, including crucial details such as the organism, cell line, and the specific library-preparation method employed. For the dataset used within this project, a metadata TSV (Tab-Separated Values) file was downloaded simultaneously with the data, following the protocol outlined in the method section. This metadata file encompasses pertinent information related to each downloaded file. The most pertinent information includes:

- **Sample ID:** A unique identifier assigned to each specific sample/patient.
- **Disease Information:** Details regarding the disease associated with the patient from whom the sample was obtained for analysis.
- **Sample Type:** Designation specifying whether the sample corresponds to a tumor sample or solid normal tissue.

- **Data Category and Data Type:** Information outlining the specific data category and type corresponding to each sample.
- **File name:** the name of the CSV file for each sample, is included within the metadata. This particular detail can be used to extract sample-specific information from the metadata file, thereby facilitating comprehensive data analysis and interpretation.

DATA PREPARATION AND CLEANING

Creating input data frames for DE analysis using pyDeseq2

As a prerequisite for utilizing PyDESeq2, the count matrices and metadata matrices need to be prepared, aligning with the requirements outlined in the PyDESeq2 package documentation (6). The count matrix is essentially a data frame with genes in the columns and sample IDs in the rows. The metadata data frame is structured accordingly, with the sample IDs in the rows and the outcome/condition in the columns. The preparation of these data frames is explained below, and the code is available in the Jupyter notebook titled "Data preparation."

Create Count Matrix Data Frame as Input for PyDESeq2 by Merging Single Files into a Single Data Frame:

The use of FPKM (fragments per kilobase of transcript per million mapped reads) and TPM (transcripts per million) gene counts for sample-to-sample comparisons and differential gene expression analysis is discouraged, as they are only suitable for gene expression comparison within the same sample (7-8). To compare gene expression between samples, normalized raw counts are recommended (7-8). Therefore, the columns labeled ‘unstranded,’ representing the raw sequencing counts for each gene, were selected for further analysis.

Since the raw sequencing data for each sample exists in individual files, a merged data frame was created by appending the ‘unstranded’ raw counts column for each sample. A Python script (notebook: Data Preparation) was devised to iterate through the files within each folder, extracting the 'unstranded' column from each TSV file, using the file name (sample identifier) as the column name, and merging all the data into a single frame for each

condition. This procedure was replicated for folders containing both normal and tumor samples. The data frames generated for each condition were then consolidated into a single data frame.

This final consolidated data frame was transposed to have the sample identifiers in the rows and the genes in the columns, preparing the data frame as input for PyDESeq2 differential expression analysis.

Data cleaning:

Numerous undetected genes across different samples were revealed by the preliminary examination, with raw count values observed to be 0 or very low in all control and tumor samples. It is essential for these genes to be removed from the analysis, as the lack of expression or detection through the sequencing method precludes the possibility of identifying differential expression. As a first cutoff, all genes with fewer than 10 total counts (sum) across all the samples were removed. The number of genes/features to be analyzed was thus reduced from 60,660 to 51,956.

Further, the samples with gene count NaN values were removed, which filtered out 2 normal tissue samples. Thus, the final data set used for analysis consists of 52 normal tissue and 304 prostate cancer tissue samples (Figure 1A, B). A partial representation of the final count matrix data frame used for pyDeseq2 analysis is shown in table 2.

gene_id	ENSG00000000003.15	ENSG00000000005.6	ENSG00000000419.13	ENSG00000000457.14	ENSG00000000460.17
unstranded_2fa8c89c-893e-465a-aeb5-e3fb81a200c7.rna_seq.augmented_star_gene_counts.tsv	3245.0	19.0	1520.0	1014.0	180.0
unstranded_288d2dc8-9fb3-4164-8e6a-cee107d11404.rna_seq.augmented_star_gene_counts.tsv	2949.0	10.0	925.0	619.0	159.0
unstranded_7b97885a-f365-4f11-8f35-9bf17a7344b5.rna_seq.augmented_star_gene_counts.tsv	3617.0	2.0	1477.0	1204.0	155.0
unstranded_e32fc401-5ddd-4dfc-8be4-8d4072a080c3.rna_seq.augmented_star_gene_counts.tsv	3117.0	5.0	1218.0	1006.0	268.0
unstranded_16eca87b-206a-4c2d-8ab7-f8c2fdb51b0e.rna_seq.augmented_star_gene_counts.tsv	2873.0	3.0	2157.0	888.0	232.0
...	...	...	...	...	...
unstranded_422d5778-bfe0-4e2f-b254-2a753c777686.rna_seq.augmented_star_gene_counts.tsv	865.0	257.0	1037.0	527.0	103.0
unstranded_60028d9d-c8e5-4c4e-a75c-cc3df6f2eea1.rna_seq.augmented_star_gene_counts.tsv	7666.0	49.0	1923.0	1495.0	263.0
unstranded_2c022016-756e-4cbf-9baa-dab196b6e715.rna_seq.augmented_star_gene_counts.tsv	6343.0	24.0	2178.0	1625.0	287.0
unstranded_998581d3-8fdc-4f4a-92cd-d98fc0f910bc.rna_seq.augmented_star_gene_counts.tsv	1582.0	503.0	1451.0	745.0	150.0
unstranded_9975ae8f-09f2-4343-9281-cbe13c775dca.rna_seq.augmented_star_gene_counts.tsv	1267.0	240.0	1052.0	500.0	137.0

358 rows × 51956 columns

Table 2. A partial view of the count matrix data frame for pyDeseq2 input is presented. Sample IDs (358 samples) are in the rows, and genes labeled with Ensemble gene identifiers are in the columns. The numbers represent the raw gene counts.

Creating the Metadata data frame for pyDEseq2

The metadata frame was created by generating a data frame with the samples names in the rows and adding the corresponding conditions Primary_Tumor_PRAD or Normal_Tissue_PRAD to each sample. The final Metadata data frame used for pyDeseq2 analysis is depicted in Table 3.

	sample_ID	Condition
0	unstranded_2fa8c89c-893e-465a-aeb5-e3fb81a200c...	Primary_Tumor_PRAD
1	unstranded_288d2dc8-9fb3-4164-8e6a-cee107d1140...	Primary_Tumor_PRAD
2	unstranded_7b97885a-f365-4f11-8f35-9bf17a7344b...	Primary_Tumor_PRAD
3	unstranded_e32fc401-5ddd-4dfc-8be4-8d4072a080c...	Primary_Tumor_PRAD
4	unstranded_16eca87b-206a-4c2d-8ab7-f8c2fdb51b0...	Primary_Tumor_PRAD
...	...	...
47	unstranded_422d5778-bfe0-4e2f-b254-2a753c77768...	Normal_Tissue_PRAD
48	unstranded_60028d9d-c8e5-4c4e-a75c-cc3df6f2eea...	Normal_Tissue_PRAD
49	unstranded_2c022016-756e-4cbf-9baa-dab196b6e71...	Normal_Tissue_PRAD
50	unstranded_998581d3-8fdc-4f4a-92cd-d98fc0f910b...	Normal_Tissue_PRAD
51	unstranded_9975ae8f-09f2-4343-9281-cbe13c775dc...	Normal_Tissue_PRAD

358 rows × 2 columns

Table 3. A partial representation of the metadata data frame for pyDEseq2 input is shown. Sample IDs (358 samples) are in the rows, and the condition is in the first column.

RESULTS

Differential expression (DE) analysis

In order to compare gene expression levels between tumor and normal tissue samples, a differential gene expression (DE) analysis was conducted using the PyDESeq2 package (6). The resulting output yielded a comprehensive table comprising gene identifiers, expression level

ratios between cancer and control samples as log2-fold changes, and pertinent information concerning the statistical significance of these changes. This information includes p-values, adjusted p-values (p-adj), test statistics, and confidence intervals (Table 4).

	Gene_id	BaseMean	log2Fold Change	lfcSE	stat	pvalue	padj	gene_name	gene_type
0	ENSG00000000003.15	3335.706899	-0.123783	0.080297	-1.541568	1.231786e-01	1.899741e-01	TSPAN6	protein_coding
1	ENSG00000000005.6	30.197803	-3.221012	0.319472	-10.082304	6.615613e-24	3.540108e-22	TNMD	protein_coding
2	ENSG00000000419.13	1377.271973	-0.077316	0.044415	-1.740738	8.172947e-02	1.339952e-01	DPM1	protein_coding
3	ENSG00000000457.14	823.547162	-0.076069	0.062807	-1.211149	2.258384e-01	3.145029e-01	SCYL3	protein_coding
4	ENSG00000000460.17	177.066627	-0.049628	0.070122	-0.707744	4.791040e-01	5.736930e-01	C1orf112	protein_coding
...	...	...	...	...	...	...	...	...	...
51951	ENSG00000288667.1	0.326720	0.600740	0.956261	0.628217	5.298617e-01	6.191116e-01	NaN	NaN
51952	ENSG00000288669.1	0.081420	0.011927	1.316544	0.009059	9.927718e-01	NaN	NaN	NaN
51953	ENSG00000288670.1	293.707593	-0.257065	0.063193	-4.067948	4.742893e-05	1.640609e-04	NaN	NaN
51954	ENSG00000288674.1	6.028666	0.101024	0.141648	0.713205	4.757190e-01	5.707712e-01	NaN	NaN
51955	ENSG00000288675.1	33.699975	0.969440	0.129579	7.481431	7.351754e-14	9.559198e-13	NaN	NaN

51956 rows × 9 columns

Table 4. DE analysis between the 2 tested conditions. A partial representation of the pyDeseq2 output data frame is shown. The log2FoldChange and Wald test p-value depict the differential gene expression levels of prostate cancer primary tumors vs. normal tissue. A positive log2FoldChange indicates that the gene has higher expression (is up-regulated) in tumor samples compared to normal tissue. A negative log2FoldChange identifies genes that have lower expression levels (down-regulated) in tumors compared to normal tissue. BaseMean: mean of normalized counts. lfcSE: standard error. Stat: Wald statistic. Pvalue: Wald test p-value. Padj: BH adjusted p-value.

The most important columns in the pyDeseq2 results table are the log2FoldChange and the padj. The padj column represents the p-value adjusted for multiple testing. Typically, a threshold such as padj < 0.05 is used as a starting point for identifying significant genes. The default method for multiple test correction in DESeq2 is an implementation of the Benjamini-Hochberg false discovery rate (FDR). Another important column in the results table is the log2FoldChange. In the output table, genes exhibiting log2FoldChange values greater than 0 are categorized as upregulated genes (more expressed in tumor samples than normal samples), whereas those with values less than 0 are classified as downregulated genes (lower expression in tumor samples).

Differential expression (DE) data were then examined and visualized through the creation of volcano plots and interactive volcano plots, utilizing the Plotly (9) and Matplotlib (10) libraries (Figure 2A). For each gene, the log2FoldChange was plotted against the -log10 of the adjusted p-value, as is customary for the y-axis. With large significant gene lists, it can

be challenging to extract meaningful biological relevance. Therefore, genes meeting the criteria of a log2FoldChange value greater than or equal to 2 or less than or equal to -2, along with a padj value of less than 0.05, indicating that the difference in gene expression was significant, were selected as being upregulated or downregulated. The rest were classified as unchanged. With this cutoff, it was found that, from the total genes analyzed (51,956), 725 genes were upregulated and 655 genes were downregulated in prostate tumor samples compared to normal tissue samples. This represents a total of 1,380 genes that were shortlisted for further comprehensive analysis and data exploration. The gene type of the differentially expressed (DE) genes was also explored by plotting the log2FoldChange of the genes against the gene type and indicating whether they belong to the upregulated, downregulated, or unchanged lists (Figure 2B-C). Most of the DE genes, either upregulated or downregulated in tumors, represent protein-coding genes, long non-coding RNAs (lncRNAs), and pseudogenes. Interestingly, the small RNA categories, such as miRNAs and soRNAs, were only present on the upregulated gene lists.

Gene Ontology analysis of the differentially expressed genes was performed to investigate potential disruptions in biological pathways leading to disease progression. This analysis was conducted in R using the Cluster Profiler package (11). Figure 3A shows to which biological process (BP) the whole list of DE genes, the upregulated, and the downregulated ones are associated. The plots display the gene ratio (number of genes associated with a given pathway over the total number of genes in the list) associated with each biological process. Genes involved in differentiation, such as neurogenesis, cell fate-commitment, pattern specification, and lipid localization, were upregulated in tumor samples. This is in agreement with previous reports showing that axonogenesis, neurogenesis, and tumor cell neural/neuroendocrine differentiation are increased in prostate tumors. Neurogenesis is also correlated with aggressive and recurrent features in prostate cancer (12,13). Moreover, Prostate cancer (PCa) is well-recognized as a lipid-enriched tumor (14). On the other hand, inhibitors of enzymes involved in the breakdown of nutrients like hydrolases and endopeptidases were downregulated in tumors, correlating with the capability of tumor cells to grow and divide faster than normal cells.

The same analysis, but focused on the molecular function of the genes, shows that upregulated genes were enriched for transcriptional activators, also needed for the cell division and growth of the tumors (Figure 3B).

The analysis of the cellular component showed that upregulated genes are enriched for genes that form part of the collagen-containing extracellular matrix. In concordance with these results, there is a known correlation between the type of collagen and the content of collagen, including orientation, and prostate cancer aggressivity (15) (Figure 3C).

The GO analysis suggests that the performed DE analysis using prostate cancer RNAseq data from the TACG database is reliable for identifying genes responsible for tumor development. Thus, the selected genes can be further utilized as features for cancer prediction by machine learning and deep learning algorithms.

Applying machine learning algorithms to predict outcome (Prostate cancer vs Normal Tissue) with the gene expression data of the selected DE genes (features)

Creating input data frames for ML

For machine learning, the already generated data frame that combined all the samples (tumor and normal tissue) and the raw gene counts for each sample were used. To facilitate accurate comparisons of gene expression between samples, normalization of the raw counts is essential (7-8). To normalize the raw counts, the PyDESeq2 package (6), which employs the median of ratios method to normalize the raw counts of each gene in the samples, thereby accounting for variations in sequencing depth and RNA composition, was used. Once the normalization was done on the entire data set, only the shortlisted genes generated by DE analysis (1380) were kept, and the outcome was added to the data frame as binary values representing the conditions for each sample: 0 for normal tissue and 1 for prostate primary tumor. The gene IDs were replaced with the gene names for easier interpretation of the results (Table 5). A partial representation of the data frame is shown in table 5.

	Condition	TNMD	AOC1	PDK4	ZMYND10	MYH13	SLC13A2	MATK	TFAP2B	TENM1
unstranded_2fa8c89c-893e-465a-aeb5-e3fb81a200c7.rna_seq.augmented_star_gene_counts.tsv	1	15.966137	865.532686	3723.471197	31.091951	0.840323	0.840323	874.776239	0.000000	715.955193
unstranded_288d2dc8-9fb3-4164-8e6a-cee107d11404.rna_seq.augmented_star_gene_counts.tsv	1	11.725003	34.002509	1224.090334	69.177519	0.000000	1.172500	315.402586	68.005019	1279.197849
unstranded_7b97885a-f365-4f11-8f35-9bf17a7344b5.rna_seq.augmented_star_gene_counts.tsv	1	1.597117	251.545950	1652.217685	46.316397	8.784144	3.194234	528.645775	1.597117	177.280003
unstranded_e32fc401-5ddd-4dfc-8be4-8d4072a080c3.rna_seq.augmented_star_gene_counts.tsv	1	4.584119	51.342133	150.359103	257.627488	2.750471	8.251414	110.935680	8.251414	16160.853136
unstranded_16eca87b-206a-4c2d-8ab7-f8c2fdb51b0e.rna_seq.augmented_star_gene_counts.tsv	1	2.606933	196.388934	13978.373392	48.662745	0.868978	0.000000	146.857211	2.606933	78.207982
...	...	...	...	...	...	...	...	...	...	...
unstranded_422d5778-bfe0-4e2f-b254-2a753c777686.rna_seq.augmented_star_gene_counts.tsv	0	362.026157	85.928388	359838.505069	23.947256	7.043310	8271.663796	30.990566	153.544168	209.890652
unstranded_60028d9d-c8e5-4c4e-a75c-cc3df6f2eea1.rna_seq.augmented_star_gene_counts.tsv	0	37.214004	5730.956592	877.946703	26.581431	0.000000	0.759469	30.378779	0.759469	841.492169
unstranded_2c022016-756e-4cbf-9baa-dab196b6e715.rna_seq.augmented_star_gene_counts.tsv	0	15.963598	27217.934445	12411.697379	17.959048	0.000000	9.312099	67.845291	1.330300	1197.934994
unstranded_998581d3-8fdc-4f4a-92cd-d98fc0f910bc.rna_seq.augmented_star_gene_counts.tsv	0	501.478180	7.975796	33270.036453	41.872930	0.000000	6171.272232	24.924363	121.630891	56.827547
unstranded_9975ae8f-09f2-4343-9281-cbe13c775dca.rna_seq.augmented_star_gene_counts.tsv	0	312.325806	361.777392	175230.393498	117.122177	0.000000	5366.798434	24.725793	144.450685	769.102297

356 rows × 1381 columns

Table 5. Data Frame generated for Machine learning input.S ample 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, and column values represent normalized counts of RNA expression.

Splitting the data in training, validation and testing sets

The data is unbalanced, with significantly fewer samples corresponding to normal tissue (5.8 times less). Due to the low number of normal tissue samples, a decision was made to split the data into two sets: a training set (80%) and a validation set (20%) to test performance of different models with as much data as possible. Since unseen data was not available to make a final validation of the model, another approach was taken to train, validate, tune and finally test the models performances. In this approach, Ten percent (10 %) of the data was set aside as unseen for the final testing of the model. The remaining data set was then split into training (80%) and validation (20%) sets. The splitting was carried out with stratification in every case to ensure that the class ratios (normal tissue vs primary tumor samples) were kept similar to those in the original data set.

When the data was split into three sets, the training and validation sets were used for searching and determining the best hyperparameters to train the model. Subsequently, the model was retrained on the entire data set (train plus validation sets) using the best hyperparameters found, and tested on the unseen data subset.

Applying Logistic Regression for prostate cancer prediction

The data that was split either into 2 (Figure 4A) or 3 data sets (Figure 4B), as explained before was loaded into a logistic regression model (16). In the logistic regression performed with the data set split into 2 sets, the model converged after 35 iterations (Figure 4A, left panels). There was no improvement in validation accuracy after 35 iterations. The model reached an accuracy of 0.931 in the validation set. The confusion matrices and classification report show that while the model performs very well in the training set, the recall for the normal category was 0.82, meaning that 18% of the normal samples were classified incorrectly as Primary Tumor samples (Figure 4A, right panels). On the other hand, the recall on the Tumor samples was 0.95, indicating that 5% of the tumor samples are incorrectly labeled as normal samples (Figure 4A, right panels).

When the data was split into 3 sets, the model took a bit longer to converge, converging at 52 iterations (Figure 4B, left panels). Surprisingly, the accuracy in the validation set was slightly higher, at 0.95 as shown in the classification report for the validation set (Figure 4B, right panel). Since we kept stratification in the split, this increase in accuracy could be due to several reasons. For example, the removed 10% might have included noisy or unrepresentative samples that negatively affected the model's performance. Additionally, randomly splitting the data can cause variation in the results, and with this small data set, the reduced dataset might have fewer opportunities for overfitting, leading to better generalization on the validation set. However, looking at the other measures of model performance, we can see that while the recall on the Tumor samples was slightly better (0.98), the recall for the normal samples, the underrepresented category, was lower (0.78) for the validation set (Figure 4B, right panel).

After training the logistic regression model with the optimal number of iterations on the entire dataset (including both training and validation sets), we evaluated its performance on the separate kept unseen test set (Figure 4C). The classification reports and confusion matrixes for the entire data set and the kept unseen test set are depicted (Figure 4C). The model achieved a high overall accuracy of 0.94 in the test set. While the recall for Tumor samples was excellent, indicating that the model correctly identified most Tumor cases, the recall for Normal samples was significantly lower at 0.6, just at the limit of random choice.

Figure 4C, right panel). This indicates that 40% of Normal samples are misclassified as Tumor samples. This disparity suggests that the model is biased towards learning the overrepresented class of Primary Prostate Tumor samples. Consequently, the model struggles to generalize well for the underrepresented Normal samples. Using the class_weight parameter in LogisticRegression to give more importance to the underrepresented class (Normal samples) did not improve the model (data not shown). Thus, other techniques like oversampling by creating synthetic data to increase the number of normal samples in the training set (see discussion) or undersampling to reduce the number of Tumor samples to balance the class distribution should be tested.

Applying Random Forest Classifier for cancer outcome prediction

Since random forest is often considered a good choice for addressing class imbalance and handling non-linear relationships in the data, the Random Forest model was then tested on the same data sets (17).

Optimizing Random Forest hyperparameters with a Single Random Split

As previously described, the data was split into 2 or 3 sets. In each scenario, we explored the number of trees (n_estimators) and maximum depth (max_depth) for optimal model performance (Figure 5A-B, 5D-E, respectively). The corresponding classification reports and confusion matrices for the validation set, obtained after training the model with the selected best hyperparameters, are illustrated in Figure 5C and Figure 5F. Impressively, the model trained on the dataset split into 2 sets achieved a remarkable accuracy of 0.99 and exhibited perfect recall in the Tumor class, along with a high recall of 0.91 for the Normal Tissue category (Figure 5C). Notably, this outperformed the logistic regression model. Even when the data was divided into 3 sets, resulting in reduced training data, the impact on model performance was minimal, with an accuracy of 0.98 (Figure 5F). The recall in the validation set for the Tumor sample category remained perfect, while for Normal samples, it was slightly lower at 89%, yet still within an acceptable range and notably better than the logistic regression scores (Figure 5C). Upon retraining the model with the entire dataset (training plus

validation sets) and testing it on the kept unseen data, the accuracy on this test dataset decreased to 0.94 (Figure 5G, right panel). The lower recall for the Normal samples (0.8) compared to the Tumor categoy (0.97) in this test set further suggests that the random forest classifier still exhibits bias, demonstrating better learning of Tumor category patterns but struggling to generalize to unseen data for the normal samples (Figure 5G, right panel). This bias likely stems from the unbalanced nature of the dataset. However, only 20% of the normal samples will be wrongly predicted as tumor samples, which is notably better than the logistic regression scores on kept unseen data.

Optimizing Random Forest Hyperparameters with Cross-Validation (CV)

To gain a better understanding of the model's ability to generalize to unseen data, cross-validation (CV) was performed using StratifiedKFold splitting on the whole dataset, as well as on a subset where 10% of the data was kept unseen for final testing (Figure 6). The results indicate that when cross-validation was employed, the mean accuracies across folds for the best-selected hyperparameters (tested in Figure 6A) were similar to the accuracies obtained in the single random-split experiment, indicating the reproducibility of the model performance (Figure 6B). Similar recall rates were also observed in the validation set, as depicted in the median confusion matrices calculated for the folds of CV (Figure 6C). Moreover, when 10% of the data was kept unseen, the mean accuracy in the validation set across folds of CV with the best hyperparameters selected (Figure 6D) was also similar to the single random-split experiment (Figure 6E). The median recall across folds of cross-validation for the tumor samples was perfect, but the recall for normal samples was 0.8, as observed in the median confusion matrices across folds of CV (Figure 6F). When the model was retrained with these best hyperparameters, selected by cross-validation on the entire dataset (train plus validation datasets), and finally tested on the unseen data, 20% of the normal samples were again mislabeled as tumor samples (recall = 0.8) in this test set (Figure 6G). This indicates that the model training is biased towards correctly classifying the overrepresented tumor sample category.

This cross-validation study indicates that the Random Forest Classifier model is likely to generalize well to unseen data and outperformed Logistic Regression in terms of

classifying the underrepresented normal samples. (See also Table 6 for model metrics comparisons). However, applying techniques to overcome the problem of unbalanced data and bias in training would be advisable to increase model performance (see the Discussion and Outlook section for solutions on this topic).

Random Forest with the principal components generated by PCA analysis.

PCA Analysis

To attempt to reduce the dimensionality/features of the dataset, a PCA analysis was conducted (Figure 7) (18). The explained variance ratio revealed that 77 components explain 80% of the variance. Additionally, to explain 90% of the variance, 134 components are required (Figure 7A). The first two principal components together account for 27.8% of the total variance, while the first 3 components explain 31.8% (Figure 7B). The visualization plot with the first 3 PCs demonstrates that the samples can be separated quite well using only these components (Figure 7C).

Random Forest with the most important principal components

Ideally, a set of principal components (PCs) that explain around 80-90% of the variance in the data should be selected as new features for input into machine learning models, resulting in approximately 77-137 features. However, a very high feature cut-off was initially tested by applying a random forest classifier using only the 2 most important components (PC1 and PC2) as input (Figures 8 and 9). The analysis was conducted using both a single random split approach (Figure 8) and a cross-validation approach (Figure 9). As before, the models were trained on datasets split into either 2 or 3 subsets. Surprisingly, even though only 27.8% of the variance was explained by the 2 PCs, they yielded the similar performance in the validations sets as the random forest classifier using all 1380 original features (genes) as input when trained with the best identified hyperparameters (Figure 8C, 9B-C and Table 6). Similarly, equivalent model performance was observed when using the first 3, 4, or 8 PCs (not shown, but tested in the notebook). Importantly, in the dataset split into 3 subsets,

predictions on unseen data were very consistent using a either the single random split or CV approaches, achieving a recall of 0.8 in the normal sample category and 0.97 for tumor samples, with an overall accuracy of 0.94 (Figure 8G, 9G).

The equivalence in performance can be explained by the fact that the first few principal components capture the most significant structure in the data. Since the expression of genes from the same biological pathways tends to have similar expression patterns, many correlated features can be expected. These correlations mean that a few principal components can capture the majority of the informative variance present in the data. Therefore, using the top principal components can provide a more efficient and equally effective representation of the data for the classifier, leading to similar performance as using all original features.

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

Convolutional neural networks (CNNs) have been successfully utilized in data domains such as speech and imaging, where essential information is embedded in the feature order. However, most tabular data do not inherently assume a spatial relationship between features, rendering them unsuitable for CNN modeling. To address this challenge, several algorithms have been developed to convert tabular data into images for neural network input, including DeepInsight (19), REFINED (20), and OmicsMapNet (21). Recently, a novel algorithm called the Image Generator for Tabular Data (IGTD) was developed by Zhu et al. (22)

This algorithm assigns features to pixel positions in a manner that ensures similar features are proximate in the resulting image. The algorithm searches for an optimized assignment by minimizing the difference between the ranking of distances between features and the ranking of distances between their assigned pixels in the image. The authors claimed that IGTD, compared to other methods, does not require domain knowledge and yields compact image representations while preserving the neighborhood structure of features. They also applied the IGTD algorithm to transform the gene expression profiles of cancer cell lines (CCLs), showcasing its utility for RNAseq data (22). Thus, the IGTD algorithm was used to convert the RNAseq data from prostate cancer and normal samples into images. Examples of the generated images are depicted in Figure 10A. Upon exploration of the data, distinct

patterns emerge in the images. Control samples generally exhibit stronger pixels in the upper-left corner, while tumor samples display more distributed pixels, with enrichment in the lower part and the upper-right corner of the images.

The images were then inputted into various Convolutional Neural Networks (CNNs) structures implemented using the PyTorch deep learning framework (23). One CNN structure, consisting of four convolutional neural networks with ReLU activation, followed by max pooling and flattening to feed into a linear layer with ReLU activation, then followed by another linear layer with sigmoid activation and one output neuron, exhibited good performance. The sigmoid activation provided probabilities for each class directly (Figure 10B-C). An example of a single run of the CNN with a set of data loaders and a split into 80% training and 20% validation sets is shown in Figure 10B.

Different learning rates were tested over various epochs, as depicted in Figure 10B. The best hyperparameters were then selected using early stopping, and the Confusion Matrix and Classification report for the validation set are displayed in Figure 10C. Table 6 presents the mean values obtained after running the CNN training and validation five times. The accuracy achieved similar levels as the predictions made using tabular data with random forest classifiers (0.967 +/- 0.018). However, the recall for the normal samples reached better maximal levels (0.945 +/- 0.071), while the recall for the tumor samples was similar to that observed in the random forest model (0.9717 +/- 0.07).

The addition of Tanh activation to the four CNN structures worsened the predictions, causing the data to predominantly learn the overrepresented tumor samples, as evidenced by the very low recall for this category (data not shown, but tested in the corresponding notebook).

Further analysis was conducted on the wrongly predicted samples for each run. It was observed that the misclassified samples in the validation sets exhibited patterns resembling the opposite category (as shown in the example in Figure 10D). For instance, misclassified tumor samples displayed strong pixels in the upper-left corner of the image, reminding normal sample previously observed patterns (Figure 10A). On the other side, misclassified normal samples lacked this pattern and showed patterns like those seen for tumor samples (as depicted in Figure 10A). This pattern was consistently observed across different runs.

Having a CNN operating on the entire dataset, the data was further split into three sets, with 10% of the data reserved as unseen, 80 % for training and the other 10% for

evaluation. The model underwent training with the training set and evaluation with the validation set to search for the best hyperparameters (Figure 11A-B, showing an example of a single run with the generated train and validation loaders). Once the optimal hyperparameters were determined, the CNN was re-trained on the entire dataset (combining validation and training loaders), and inference was performed using the kept unseen test set (Figure 11C). The confusion matrix and classification report for the kept unseen data is shown (Figure 11C).

Although direct comparison is challenging, the mean accuracy observed for the CNN was slightly better than that observed for the random forest model (Table 6). Additionally, the recall was slightly improved for the normal samples and remained nearly unchanged for the tumor samples. By utilizing CNN with images, the number of false positives (normal samples wrongly predicted as cancer samples) was reduced to 11.4% (mean, 5 runs), compared to 20% with the random forest classifier in the test unseen data. Conversely, the number of false negatives (tumor samples detected as normal) was quite similar (3% for random forest and 2.6% for the CNN model).

Once again, upon examining the wrongly predicted samples, they consistently exhibited a pattern visually resembling the pattern identified as corresponding to the opposite category. During CNN training, loaders were not split in a stratified manner. It was observed that stratification significantly improved the performance of the random forest classifier (data not shown). Therefore, enhancing this step could potentially further improve the performance of the CNN. Even without stratification, the CNN appeared to predict the underrepresented normal tissue category better than the random forest classifier (at least it can reach better performances). Perform cross-validation and loaders stratified splitting to accurately compare model performances should also be done. This would provide a more comprehensive evaluation of the models' capabilities.

Search for Biomarkers and Targets by Combining Differential Expression Analysis and Identification of Important Features for Outcome Prediction in Random Forest Classification

Calculating Feature importance's for model prediction

A common way to identify RNA biomarkers or potential therapeutic targets is to focus on the differentially expressed (DE) genes between cancer and normal samples. However, setting cutoff thresholds for gene expression is not straightforward. Some genes may not show strong differences in expression but can still have a significant impact on cancer progression. I set out to test if identifying feature importance can be used as an alternative method to determine which genes or groups of genes should be further tested as biomarkers or therapeutic targets. Additionally, I aimed to determine if the models rely on the most differentially expressed genes for outcome prediction. As a first approach, we identified important features in the random forest model trained with a single random split, as shown in Figure 5A-C. Several algorithms have been developed to identify important features for model prediction.

Gini importance, also known as mean decrease in impurity (MDI), is a way of measuring feature importance in a Random Forest model (17). The Gini importance of a feature is computed as the total decrease in Gini impurity brought by that feature, weighted by the probability of reaching that node (approximated by the proportion of samples reaching that node). This measure indicates how often a feature is used to split a node and how much this feature reduces Gini impurity. Gini importance is a measure internal to the Random Forest model and is calculated during the training process on the training set. The first 35 important genes/features and their Gini importances are shown in Figure 12A. In total, 67 genes had importances higher than 0 and were selected for further analysis.

Permutation importance is another method to measure the importance of features in a model. Unlike Gini importance, which relies on the internal structure of the model, permutation importance is model-agnostic. It works by shuffling the values of each feature and measuring how much the shuffling decreases the model's performance. The idea is that if a feature is important, permuting its values will result in a significant decrease in the model's

performance. The the feature_importances_ attribute of the RandomForestClassifier class in the scikit-learn library was used for calculating feature Importance's (24). Permutation importances for the training and validation sets are shown in Figure 12B. The number of trees (n-estimators = 7) and the depth of the trees (max-depth = 8) used for training and validation were chosen based on the results in Figure 5A-C. These hyperparameters were selected because they provided the best accuracy and recall for both types of samples. However, at this lower number of estimators, the model is not very stable, and the validation error starts to converge at higher estimators. When calculating feature importances with a model using a higher number of estimators where the validation accuracy curve stabilizes (e.g., 100), the feature importances on the training set yield 0 values for all features (not shown). This is likely due to overfitting in the training set. Ideally, other regularization hyperparameters (min_samples_split, min_samples_leaf, max_features) should be adjusted so that the model with a large number of estimators converges to a stable value that is also optimal. Using this intermediate model with a lower number of estimators, we were able to detect important genes for model prediction (Figure 12B). A total of 22 genes had feature absolute importance values higher than 0 in both the training and validation sets, indicating that the model relies primarily on the counts of these genes for outcome prediction. These features were selected for further analysis.

SHAP (SHapley Additive exPlanations) values provide a unified measure of feature importance based on cooperative game theory. They explain the output of any machine learning model by assigning each feature an importance value for a particular prediction. SHAP values are consistent and additively explain how each feature contributes to the model's output (25,26). The SHAP values for the 35 most important features in the training and test sets are shown in Figure 12C. A total of 67 genes had mean absolute SHAP values higher than 0 and were selected for further analysis.

Identifying common genes deemed important by all the algorithms.

The selected features were examined using Venn diagrams to pinpoint common genes identified as significant by each algorithm (Figure 12D). For this analysis, emphasis should be placed on the relevant features, which include those calculated on the validation set through

permutation and the SHAP explainer, and Gini importances calculated on the training set (Figure 12D, most left Venn diagram). There was complete overlap among the 67 features considered important, with values greater than 0, derived from the SHAP explainer in the validation set and Gini importance in the training set. Moreover, the 22 features identified as important by the permutation importance algorithms in the validation set fall within this complete overlap. We called these features: Int_all_Split. Additionally, there is complete overlap between the features calculated using the SHAP explainer in both the test and training sets.

Interestingly, the features calculated by permutation importances in the train and validation sets only had a 50% overlap. This discrepancy could be explained by differences in the data distribution or characteristics between the training and validation sets. It's possible that certain features were more influential or informative in one set compared to the other due to variations in sample composition, noise, or other factors. Additionally, the inherent randomness in the permutation importance calculation process could have contributed to differences between the two sets. However, all these features lie inside the calculated importance's using the Gini algorithms and the SHAP explainer. Suggesting that all these genes are important for model performance.

The fact that permutation importances only capture a subset of features identified by the SHAP and Gini algorithms could be attributed to the nature of the data and the presence of highly correlated features in the dataset. This is not surprising for this type of gene expression data, where genes regulating other genes in various ways can activate or deactivate cascades of gene expressions.

Thus, to corroborate this hypothesis, the correlation of the important features calculated by the SHAP explainer in the validation set was studied by performing hierarchical clustering on the gene expression data of the 67 most important features. The clusters were then plotted either together with the SHAP values (Figure 13A) or as dendrograms (Figure 13B). The graphs demonstrate that the most important features are highly correlated, and no clear clusters are observed. This suggests that during the calculation of feature importance using permutation importance algorithms, certain features may not be identified. This could occur because other features might compensate for the model's performance when the specific feature under scrutiny is removed for testing its importance in model prediction

Genes identified by feature importance algorithms for outcome prediction are not among the top DE genes.

At first glance, one might anticipate that the model would prioritize genes that exhibit high differential expression. The Top 100 differentially expressed (DE) genes, selected based on either their highest absolute log2FoldChange or lowest p-adj value, were compared to two lists: the 22 common features identified across all sets (Int_all_Split, Figure 12D, left Venn Diagram) and the 67 features identified using the SHAP explainer and Gini importances (Figure 12D, middle Venn diagram). However, genes identified as important for model prediction do not appear among the top DE genes as shown in Figure 14A. There was very little overlap between the genes identified by feature importance algorithms and the Top DE genes. The same observation was made when comparing the genes selected by feature importance to the top 100 upregulated and 100 downregulated genes based on log2Fold changes (Figure 14B). This indicates that the model is not solely relying on the top DE genes for its predictions. Biological systems are highly complex, and gene regulation involves intricate networks of interactions. ML may capture patterns beyond simple differential expression, such as gene-gene interactions or regulatory mechanisms, leading to different sets of important features compared to traditional DE analysis.

The different lists of genes were analyzed to determine if they were enriched for genes involved in specific biological pathways through Gene Ontology (GO) analysis (Figure 14C). Consistent with Figure 3, the top 100 DE genes were found to be enriched for genes involved in similar pathways as those identified for the entire list of DE genes. Surprisingly, the list of 67 top genes identified by the SHAP and Gini algorithms did not exhibit significant enrichment in any specific pathway, despite displaying the fact that these features show a high level of correlation as revealed by cluster analysis (Figure .13) Interestingly, however, the list of genes observed in all sets (Int_all_Split) showed enrichment for genes involved in leukocyte proliferation and activation, processes known to be implicated in cancer (27) (Figure 13C, right graph). Since genes with well-established roles in relevant biological pathways are considered more promising biomarkers, these genes merit further experimental validation.

As a means of assessing whether the top DE genes and those selected by feature importance are enriched for genes associated with prostate cancer or cancer in general, a search was conducted in PubMed. This search aimed to retrieve the number of publications

where the gene name and various synonyms for cancer were mentioned, with or without the term 'prostate' appearing in the abstract and/or title of the publication (Figure 14D). Findings reveal that, in comparison to a list of random genes, the genes utilized as input for machine learning, which were already differentially expressed and selected based on a cutoff of DE log2 fold < 2 or > -2, retrieved more genes with publications related to cancer than the list of random genes.

Interestingly, selecting either the top 100 DE genes or those identified by model feature importance calculations did not further enrich for genes with publications specifically related to prostate cancer or cancer in general. However, it's worth noting that genes already known to be involved in prostate cancer progression or utilized as biomarkers for prostate cancer were identified. Figure 14E show the number of publications found for each gene (to up of 10 publications). This also points that our study is successful in identifying genes related to the disease.

Additionally, many of these feature-selected genes did not exhibit publications associated with cancer, indicating their potential as candidates for further experimental validation (Figure 14E). The notebook also retrieves the tittle and abstracts of the interested genes to further evaluate easily and select the best candidates for experimental validation.

On the other hand, this study of feature importance is limited, as it was conducted with the model tuning and training using a single random split. Moreover, as explained earlier, this random forest model requires adjustments to ensure that the model with a large number of estimators converges to a stable and optimal value, achieved by experimenting with other regularization hyperparameters such as min_samples_split, min_samples_leaf, and max_features. Additionally, calculating the mean feature importances during model tuning by cross-validation may yield more accurate and generalized results for unknown data.

CONCLUSIONS AND OUTLOOK

In this study, we used different machine learning (ML) algorithms to predict prostate cancer outcomes using RNA sequencing data from the TCGA database. The performance of the different models tested, along with the types of training, evaluation, and testing used, is summarized in Table 6. While analyzing the original tabular data, the random forest classifier outperformed logistic regression. The use of PCA to reduce features was beneficial, as the

random forest model maintained similar performance while reducing the 1,380 features to just the 2 principal components. This reduction in features would be advantageous for saving computational power and time. A convolutional neural network (CNN) was also tested on the converted tabular data transformed into images. The CNN slightly improved model performance, especially in terms of achieving better recall for the underrepresented category. Additionally, visualizing the samples was insightful, as distinguishing differences in tabular data with 1,380 features is challenging. Preliminary analysis of these visualizations indicated that samples exhibited distinctive patterns depending on whether they belonged to cancer or normal groups. These patterns were inconsistent in the misclassified samples, suggesting that these misclassifications could be due to inherent differences in these samples, such as being originally mislabeled or representing early stages of cancer. Another explanation could come with the nature of the samples, since in the context of the Cancer Genome Atlas (TCGA) project database, specimens of histologically normal tissue adjacent to surgically removed tumors are considered normal. However, these tissues may not be completely normal due to the numerous effects tumors may have on the neighboring cells, including biased growth factors and cytokine balances, pathological inflammation, and altered vascularization. Consequently, the control samples derived from the TCGA database might not be optimal and may exhibit some gene expression patterns related to tumorigenesis. Alternative databases containing normal sample data, such as the Oncobox Atlas of Normal Tissue Expression (ANTE) platform (28) or the GTEx Portal (29), could be used as additional sources for normal tissue data. However, this hypothesis requires further evaluation and in-depth analysis of the specific wrongly classified samples.

The main problem with the performed analysis was that the data is unbalanced, with fewer samples belonging to the control, normal tissue class (5.8 times less) than the tumor samples. Although all models achieved high accuracy, they struggled more with classifying the underrepresented samples and performed very well on the tumor samples. Using stratified splitting improved the performance of the random forest model in predicting the underrepresented samples (data on the unstratified samples is not shown in this report but was tested in the notebook). Since we did not use stratified splitting for loading the CNN input on the image data, it is likely that implementing stratification could further enhance the CNN model's performance. Therefore, improving the CNN code to apply stratification and cross-validation would be the next steps. Additionally, optimizing the Random Forest model by

adjusting other regularization hyperparameters (min_samples_split, min_samples_leaf, max_features) should also be undertaken to improve model stability, mitigate overfitting, and enhance overall model performance.

To overcome the problem of unbalanced samples, techniques such as undersampling to reduce the number of tumor samples, thereby balancing the class distribution, or oversampling by increasing the number of normal samples in the training set through the generation of synthetic data could be tested. Synthetic data generation using generative adversarial networks (GANs) for bulk RNA-seq data has already been reported and could be implemented to augment the dataset for training and validation (30-31). Since the model works well with the principal components (PCs), generating synthetic data using only these newly generated features would be more convenient. Data augmentation for the generated images used as input for the CNN was preliminarily tested (results not shown). However, when rotations and flipping were applied, the model's performance worsened. This is likely due to the fact that, as previously mentioned, the pixel values are spatially ordered in the images, and these augmentation techniques destroy this spatial information. Therefore, it could be beneficial to experiment with other types of image augmentation techniques, such as brightness adjustment, contrast adjustment, and adding noise, which might preserve the spatial relationships.

Another significant problem was the lack of an easily accessible dataset for performing a final, independent test of the model. Consequently, even though the dataset was limited, primarily due to the insufficient number of normal samples, we conducted a test by keeping 10% of the samples unseen for final evaluation. This test showed that the random forest model and the CNN performed quite well in predicting the unseen data, although the recall for the normal category was lower. However, since these samples came from the same dataset, this test has the limitation that the data is not completely independent.

To address the problem of unbalanced samples and the lack of independent data for final model performance testing, it is advisable to increase the dataset or obtain independent data. Synthetic sample production is one possibility, as mentioned earlier. Additionally, other datasets available for RNA-seq of normal tissues, such as those referenced previously (28-29), can be utilized. Independent RNA-seq data on prostate tumor samples, published and available in the Gene Expression Omnibus database (32,33), could also be used to augment the dataset. However, this approach is not straightforward, as sequencing reads have been

mapped differently, resulting in a varying number of identified genes due to the use of different genome versions during gene mapping. To overcome this, one could remap all the samples in a consistent manner to compile more samples for either training and validation or for final independent testing of the trained models. However, remapping raw sequence reads is complex and requires significant time and expertise, which were not available for this report but can be addressed in future work.

Analyzing key features to identify genes that could serve as biomarkers, therapeutic targets, or explain the cancer phenotype is highly challenging. Interestingly, the most important features identified by the random forest model do not appear to be related to the level of differential expression (DE). This suggests that the model is not solely relying on the top DE genes for its predictions. Gene regulation involves intricate networks of interactions, and machine learning (ML) may capture patterns beyond simple differential expression, such as gene-gene interactions or regulatory mechanisms, leading to different sets of important features compared to traditional DE analysis. ML algorithms may be more sensitive to subtle yet informative patterns in the data or capable of capturing nonlinear relationships that influence feature selection but are not detected by traditional DEG analysis methods. The complexity of the ML model and its ability to uncover these intricate patterns allow it to select features that are not among the top DEGs but still contribute significantly to predictive performance. Overall, while the discrepancy between top DEGs and features selected by machine learning algorithms might initially seem surprising, it underscores the complementary nature of different analytical approaches and highlights the complexity of biological data analysis. Whether these features represent promising markers or therapeutic targets will still require experimental validation, validation with independent data sets, and testing in different cohorts.

When plotting the level of expression for the genes selected by feature importance in all models (Int_all_split list, Figure 14), in tumor and normal samples, the expression of most of the selected genes was significantly different between prostate cancer and normal tissue samples (Figure 15). Notably, genes already known to be involved at some extend in prostate cancer, or cancer, and highlighted in orange in the figure, had higher total gene counts. This is probably because they were identified by experimental methods with a certain detection limit. Several candidates, marked in blue, also had relatively decent total counts and could be the first candidates to test experimentally. These include the upregulated genes AC005064.1

and ENSG00000286989.1, and the downregulated genes JPH4, AC005180.1 and CCD27. Moreover, JPH4, AC005180.1, ENSG00000286989.1, and CCD27 were also found to be important in PCA analysis based on the PCA loadings (Figure 16). Several of the other selected genes had low total counts, making them less promising for experimental validation by techniques such as qPCR. However, these genes with lower expression might still be promising candidates as therapeutic targets and should not be discarded. A more in-depth analysis of the top selected individual genes, including all 67 genes identified with the SHAP explainer and Gini importance methods, should also be conducted to determine which genes are most valuable for experimental testing. Adjusting the threshold for the total count of gene occurrences across samples, currently set at 10 during data cleaning, could potentially enhance our findings and offer more insightful genes for experimental validation. This adjustment is particularly relevant given that certain genes identified as influential for the Random Forest model's performance exhibit minimal normalized counts for both experimental conditions, often hovering around zero. The model's reliance on these very low expressed genes for outcome prediction might be attributed to their correlation with other features that hold significance for the outcome. In such scenarios, including these features can be advantageous as they collectively capture valuable information. This observation aligns with the presence of correlated features as evidenced by hierarchical cluster analysis (Figure 13). Varying the log2FoldChange cutoff, currently set to an absolute value of 2, should also be tested to determine the most appropriate cutoff for identifying relevant genes or genes with higher level of detection by other experimental methods. It is also observed that some samples exhibit outlier expression levels for specific genes, with abnormal counts compared to other samples. It is important to investigate whether these outliers correspond to the same samples, and to assess whether these samples were incorrectly predicted or did not align with the overall data clustering patterns for example with the PCA analysis. Furthermore, it would be valuable to examine if these specific samples show substantial deviations in total read counts or in the percentage of unmapped reads compared to others, which could suggest potential technical issues during sequencing, sample degradation or contamination issues, warranting a careful reconsideration of their inclusion in the analysis. Overall, a more stringent data cleaning approach may have enhanced the analysis, highlighting the importance of thorough quality control measures in RNAseq data analysis.

The TCGA includes Age and ethnicity data. Given that the age of the patients can significantly influence gene expression patterns, it could be also advisable to analyze the age distribution and ethnicity of the samples and if this is correlated to the wrongly predicted samples or samples that do not cluster well. Additionally, the ethnicity of the patients from whom the samples were obtained introduces another potential bias that must be carefully considered, particularly when generalizing the results for application in the healthcare sector across other populations with different ethnic group compositions.

In conclusion, although the tested ML models require further regularization and optimization, they have shown promise in predicting prostate cancer outcomes from RNAseq data. Random forest models are well-documented for their effectiveness with RNAseq data. The use of CNN for cancer prediction, employing tabular data conversion to images, is a novel approach that has shown slightly better performance than random forest in our study. Additionally, the benefit of this approach lies in the ability to visually analyze images to better understand poorly predicted samples. It would be interesting to use all the features for image generation to test if the model performance improves with the images generated from more features.

Furthermore, feature importance algorithms have demonstrated their ability to identify genes /RNAs potentially involved in cancer progression, which may not be the most differentially expressed ones typically tested in experiments. These RNAs hold promise as potential targets or biomarkers and warrant further experimental validation. The final aim will be to use these RNA markers for early and non-invasive diagnosis of cancer. Notably, circulating cell-free or extracellular vesicle-packaged tumor-related nucleic acids, encompassing mRNAs, microRNAs, and lncRNAs in bodily fluids such as peripheral blood, nipple aspirate fluid, sweat, urine, and tears, have emerged as potential non-invasive diagnostic biomarkers, supplementing existing clinical methods for early cancer detection (3, 34). Following the identification of biomarker combinations capable of predicting prostate cancer, subsequent wet lab experiments are imperative to ascertain the detectability of these RNA biomarkers in fluids using techniques like quantitative PCR (qPCR). The goal is to achieve a prediction of cancer outcomes using a combination of these novel selected biomarkers with enhanced or equivalent accuracy, sensitivity, and specificity compared to conventional and diagnostic methods currently in use or under development and that are easily detected through a non -invasive technique.

METHODS

Infrastructure and packages

Local Python installation (35,36) within Anaconda (37) and Google Collab served as the project's primary Infrastructure. The project used the following main modules and packages**.** OS Package (38) and TCGA Downloader Python Script (5) were used to facilitate the download of transcriptome profile raw count data from the TCGA database (4). Initially, a 'manifest' file was acquired directly from the web interface of the database. The PyDESeq2 package (6) was used to normalize the row count data and to perform the differential expression (DE) gene analysis. Pandas (39) and NumPy (40) libraries were widely used for creation of the data frames, data filtering and preparation of the data for subsequent analysis and visualization. The Data Visualization Libraries, Plotly (9), Matplotlib (10), matplotlib_venn (41) and Seaborn (42) were be used for diverse data visualization and exploratory data analysis purposes. The IGDT script described was used to convert tabular data to images (Ref). Machine Learning Libraries**:** Scikit-learn (24) was employed for implementing logistic regression and random forest ML methods. pyTorch (23) was used for implementing the Convolutional neuronal network (CNN) on RNAseq data converted to images. SHAP were calculated using the SHAP libraries (25,26). Biopython package (43) was used to retrieved information from the PuMed. The GO analysis was done in R using the ClusterProfiler package (11). In this study, the IGTD algorithm (22) was utilized to transform RNA-seq tabular data into images for subsequent loading into a CNN aimed at cancer prediction. The required environment was set up and the required packages were installed following the authors specifications (22, and GitHub repossitory). Key functions from the IGTD package, including min_max_transform, table_to_image, and select_features_by_variation, were applied with specific parameters. Euclidean distance was employed for computing pairwise feature and pixel distances, while the absolute function was utilized to assess the discrepancy between the feature distance ranking matrix and the pixel distance ranking matrix, following the methodology outlined by the authors. Additionally, hyperparameters such as num_row = 30 and num_col = 30 were set to define the dimensions of the image representation in terms of pixel rows and columns, respectively. The parameter save_image_size = 3 dictated the size of images saved during the

IGTD algorithm's execution, while max_step = 30000 determined the maximum number of iterations for algorithm convergence. A validation step of val_step = 300 was also included to ascertain algorithm convergence within the specified iterations as a default specified by the authors. To enhance the clarity and quality of the manuscript, OpenAI's ChatGPT was used for assistance in refining the text and improving the code.

Data retrieval

Transcriptome profiles (RNAseq data) of prostate cancer samples and their corresponding normal tissue samples were downloaded from The Cancer Genome Atlas (TCGA) data set, accessible via the National Cancer Institute's (NCI's) Genomic Data Commons (GDC) platform (4). The retrieval process involved the generation of a manifest file directly on the GDC Data portal, following the specified portal instructions. The following steps were undertaken to produce the manifest files. Under the 'Repository' section, the criteria for downloading transcriptome data were Data category: Transcriptome Profiling, Data type: Gene Expression Quantification, Experimental Strategy: RNA-Seq and, Workflow Type: Star-Counts. Within the 'Cases' section, the data selected was the TCGA-PRAD for prostate gland cancer project samples. These selected samples were added to the browser's 'shopping cart', leading to the download the manifest file as a text file. Subsequently, the tcga_downloader.py Python script, developed by Vincent Appiah (5), was employed to facilitate the download of data utilizing the generated manifest file.

ACKNOWLEDGEMENTS

I would like to express my gratitude to all the teachers and Dr. Sigve Haug, for their exceptional guidance and support throughout this CAS. Special thanks are due to Aris Marcolongo for generously dedicating his time to provide invaluable advice for this final project. I am also thankful to my colleagues, especially my partners in the working modules, Matthias and Mahajan, for engaging discussions and collaborative learning experiences.

REFERENCES

1. Hyuna Sung *et.al*. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. CA: A *Cancer Journal for Clinicians*, 71(3), 209-249.
2. Le Wang *et al*. (2022). Prostate Cancer Incidence and Mortality: Global Status and Temporal Trends in 89 Countries From 2000 to 2019. *Front Public Health,*10:811044.
3. McGrath S. *et. al*. (2016). Prostate cancer biomarkers: Are we hitting the mark? *Prostate International,* 4 (4):130-135.
4. TCGA Research Network Data: https://www.cancer.gov/tcga.
5. Vappiah V. (2021). tcga downloader python script. Github repository: https://github.com/vappiah/DataMiner. Tutorial: https://www.youtube.com/watch?v=YJxcsm4aJXI&t=150s.
6. Muzellec *et.al.* (2022). Pydeseq2: a python package for bulk rna-seq differential expression analysis. *bioRxiv*. doi:10.1101/2022.12.14.520412.
7. Yingdong Z *et al.* (2021). TPM, FPKM, or Normalized Counts? A Comparative Study of Quantification Measures for the Analysis of RNA-seq Data from the NCI Patient-Derived Models Repository. *Journal of Translational Medicine,* 19:269.
8. Harvard Chan Bioinformatics Core (HBC) team. Introduction to DGE-Archived. Available at: https://hbctraining.github.io/DGE_workshop/lessons/02_DGE_count_normalization.html.
9. Plotly Technologies Inc. (2015). Collaborative data science Publisher: Plotly Technologies Inc. Montréal, QC. Available at: https://plot.ly
10. Hunter J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering. IEEE COMPUTER SOC*. 9 (3), 90-95.
11. Wu, T. *et al.* (2021). clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *Innovation*, 2(3), 100141. doi: 10.1016/j.xinn.2021.100141.
12. Ayala G.E. *et al.* (2008). Cancer-Related Axonogenesis and Neurogenesis in Prostate Cancer. *Clin Cancer Res*, 14 (23): 7593–7603.

13. Sigorski D. *et. al.* (2021). Investigation of Neural Microenvironment in Prostate Cancer in Context of Neural Density, Perineural Invasion, and Neuroendocrine Profile of Tumors. *Front Oncol*, 11: 710899.
14. Dang,Q, Chen Y A. and Hsieh J. T. (2019) The dysfunctional lipids in prostate cancer. Am. *J. Clin. Exp. Urol.,* 7(4): 273-280.
15. Ageeli W. *et. al*. (2021). Characterisation of Collagen Re-Modelling in Localised Prostate Cancer Using Second-Generation Harmonic Imaging and Transrectal Ultrasound Shear Wave Elastography. *Cancers* (Basel), 13:21
16. Hosmer D. W., Lemeshow S. and Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley.
17. Breiman L. (2001). Random Forests. *Machine Learning*, 45(1), 5-32.
18. Jolliffe I. T. (2002). *Principal Component Analysis* (2nd ed.). Springer.
19. Sharma A. *et al.* (2019).DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Sci. Rep.* 9, 11399.
20. Bazgir O. *et al.* (2020). Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks. *Nat. Commun.* 11, 4391.
21. Ma, S. and Zhang, Z. (2018).OmicsMapNet: Transforming omics data to take advantage of deep convolutional neural network for discovery. arXiv:1804.05283
22. Zhu Y. *et al.* (2021). Converting tabular data into images for deep learning with convolutional neural networks. *Scientific Reports* 11: 11325. https://github.com/zhuyitan/IGTD.
23. Paszke A. *et al.* (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems, 32:* 8024-8035.
24. Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12, 2825–2830.
25. Lundberg S.M. *et al*. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* (2), 56–67
26. Lee S.I. and Lundberg S.M. (2017). 31th Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. https://github.com/shap/shap.

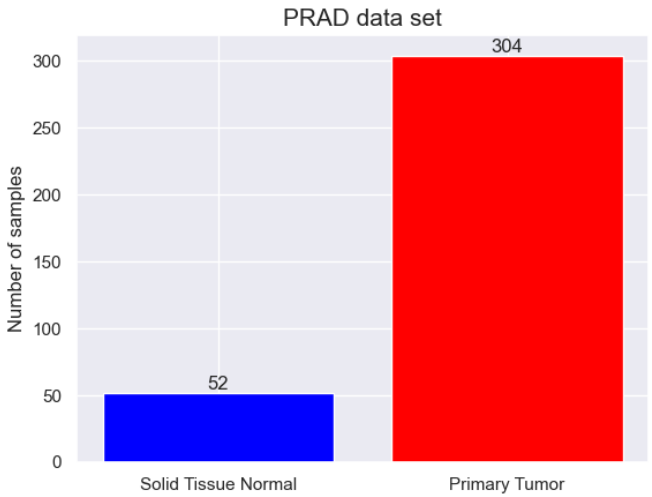
27. Labateya N. *et al* .(1986). Leukocyte activation in advanced cancer as an explanation for absent leukocyte adherence inhibition to cancer extracts and chemoattractant. *Eur. J. Cancer Clin Oncol,* 22(1): 33-43.
28. Suntsova M. *et al.* (2019). Atlas of RNA sequencing profiles for normal human tissues. *Scientific Data,* 6: 36.
29. GTExPortal: https://gtexportal.org/home/aboutAdultGtex.
30. Lacan A., Sebag M. and Hanczar B. (2023). GAN-based data augmentation for transcriptomics: survey and comparative assessment. *Bioinformatics*, 39(1). i111–i120.
31. Wang, Y. et al . (2024). Generating bulk RNA-Seq gene expression data based on generative deep learning models and utilizing it for data augmentation. *Computers in Biology and Medicine*, 169.
32. Stelloo S *et al.* (2018). Integrative epigenetic taxonomy of primary prostate cancer. *Nat Commun*, 9(1):4900.
33. Gene Expression Omnibus data base: https://www.ncbi.nlm.nih.gov/geo/
34. Li J. *et al.* (2020). Non-Invasive Biomarkers for Early Detection of Breast Cancer. *Cancers(Basel),* 12(10): 2767.
35. Python Software Foundation. Python Language Reference, version 2.7. Available at:http://www.python.org
36. G. van Rossum. (1995). Python tutorial, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam.
37. Anaconda Software Distribution. (2020). Anaconda Inc. Available at: https://docs.anaconda.com/.
38. Van Rosuum. (2022). os-Miscellaneous operating system interfaces. Available at: https://docs.python.org/3/library/os.html
39. The Pandas Development Team. (2020). pandas-dev/pandas: Pandas. v 2.1.2. Zenodo. Available at: https://doi.org/10.5281/zenodo.3509134
40. Harris Charles R. et al. (2020). Array programming with Numpy. *Nature*, 585:7825, 357-362.
41. Konstantin Tretyakov. (2012). Matplotlib-venn 0.11.9. Available at: https://pypi.org/project/matplotlib-venn/
42. Waskom, M. L., (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021, https://doi.org/10.21105/joss.03021.

43. Cock P.A. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25, 1422-1423.

A

B

Figure 1



Data acquisition
retrieve RNA seq from the TCGA data base

RNA-seq of prostate cancer patients from TCGA (Normal=52 , Tumour=306)
Number of genes (features) = 60660
Format = single files in 2 folders labeled with the condition

Data processing
Create input data frames for DE analysis

Merged single files into a single data frame , extracting the total counts 'unstrandred' for the different genes and for each sample ID. Remove genes with 0 or very low counts in all the samples (cut off 10 counts), features =51956
Remove samples with Nan values: (Normal=52 , Tumour=304).

Prepare counts and metadata data frames required as inputs for DE expression

Differential gene expression (DE) analysis

Performed Differential expression analysis between tumour and normal samples using pyDeseq2

DE Data visualization, exploration: Volcano plots, biological pathways

Detect the most DE expressed genes

Prediction of disease outcome and identification of biomarkers / putative therapeutic targets

Prepare a data frames as input for ML models: Data frames with normalized gene counts and outcome.
Dimensionality/Feature reduction:
a) **Based on DE :** Select genes with log 2-fold> 2 or log 2 -old <-2 and padj <0.05 as a start point :1380 genes.

Supervised machine learning with tabular data:
- random forest
-Evaluate model performance

- Analysis of Feature (genes) importance: Permutation, Gini, SHAP

Identify common features /genes as biomarkers/ putative targets

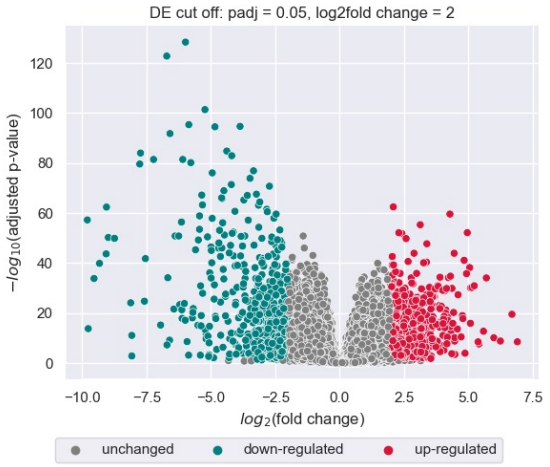
PCA analysis
-dimensionality reduction
-Random forest with PCs
-Evaluate model performance

Covert tabular data to images using IGDV conversion

Supervised machine learning with images:
-CNN
-Evaluate model performances

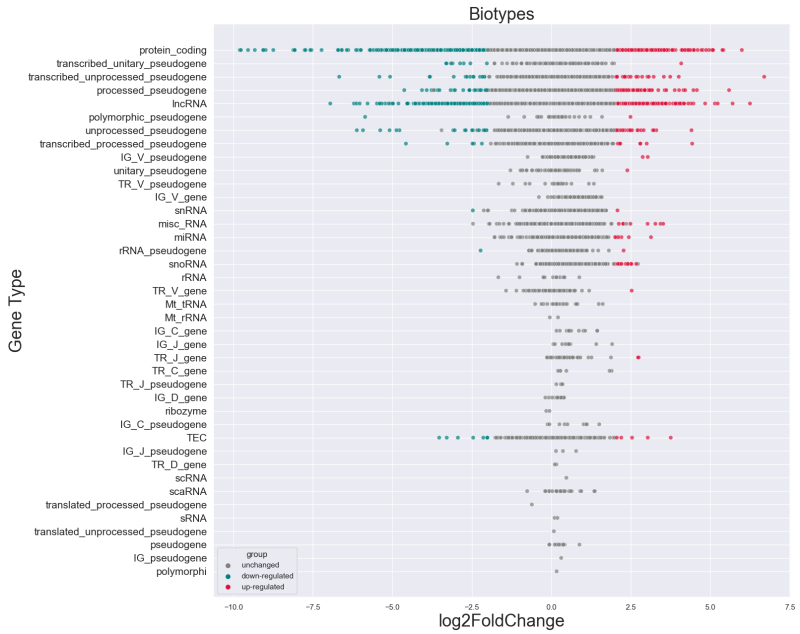
Figure 1. (A) Flow chart of the research process**. (B)** Plot showing the number of RNAseq samples retrieved for each category representing, Solid Normal Tissue and prostate Primary Tumor tissues.

A



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

B



C

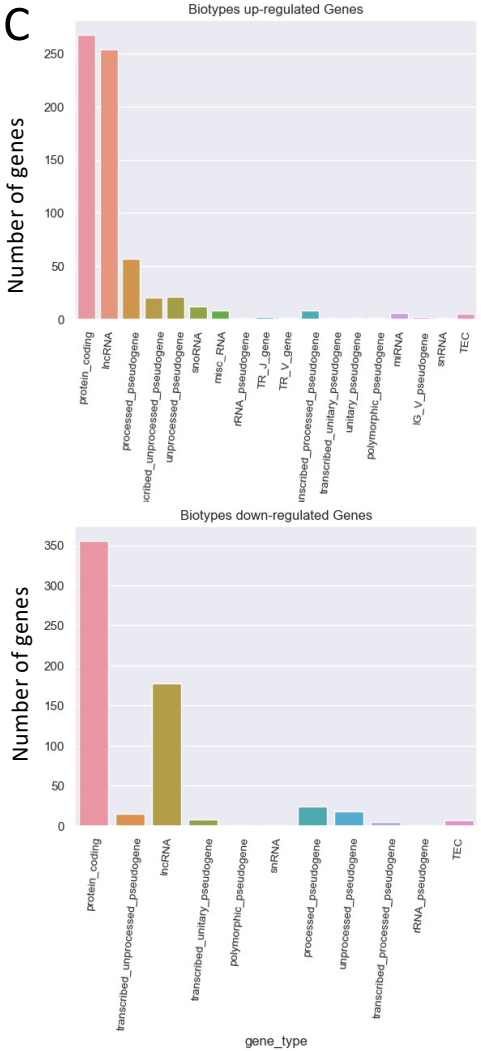


Figure 2. Differential Expression (DE) analysis visualization. **(A)** Volcano plot displaying the log2 FoldChange value versus the -log10 of the padj value for all genes with reads. A cutoff of an absolute log2 FoldChange value of 2 and a padj value < 0.05 was selected. Genes expressed at higher levels in prostate cancer samples (upregulated genes) are shown in red. Genes expressed at lower levels in prostate cancer compared to normal tissue samples are marked in green (negative log2 FoldChange values). Genes that are not differentially expressed are labeled in gray. In the notebook, the interactive plot can display the gene names for each dot. **(B)** Biotypes of the upregulated, downregulated, and unchanged genes selected by the same cutoff. **(C)** Counts of genes observed from each gene biotype in the upregulated and downregulated sets.

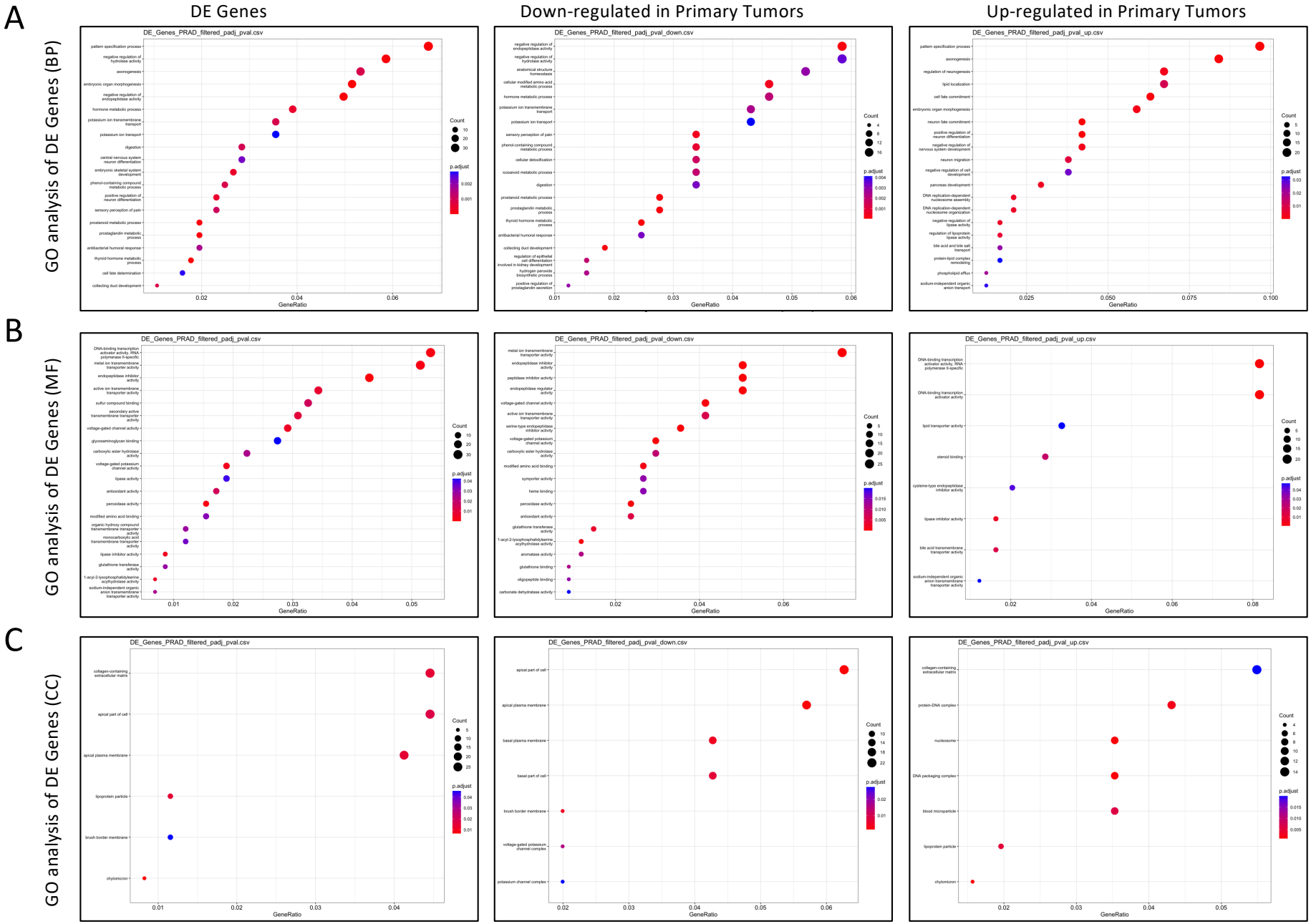


Figure 3. Gene Ontology Analysis of Differentially Expressed Genes. Gene Ontology (GO) analysis of differentially expressed genes using ClusterProfiler, which groups genes based on their involvement in common biological processes (BP) (A), molecular functions (MF) (B), and cellular components (CC) (C).

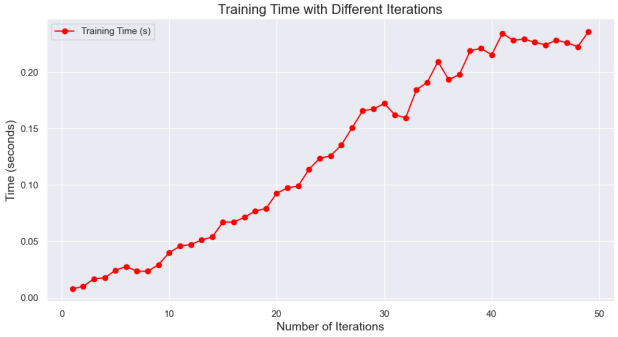
Figure 4

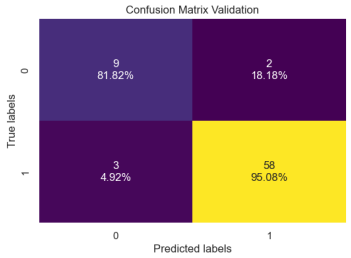
A

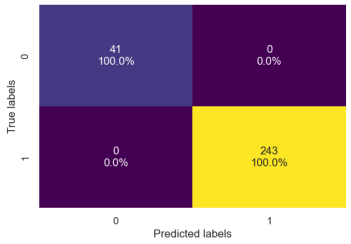
Logistic regression
(Split: Train (80%)-Validation (20%))

Classification report
(Validation Set, 35 itinerations)

	precision	recall	f1-score	support
0.0	0.75	0.82	0.78	11
1.0	0.97	0.95	0.96	61
accuracy			0.93	72
macro avg	0.86	0.88	0.87	72
weighted avg	0.93	0.93	0.93	72





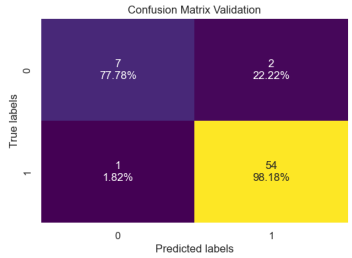


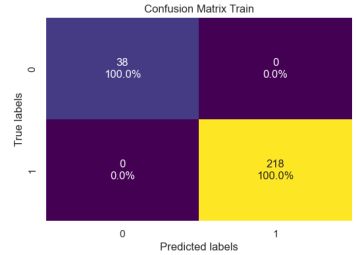
B

Logistic regression
(Split: Test (kept unseen data (10%),
Rest: Train (80%)- Validation (20%)

Classification report
(Validation Set, 52 itinerations)

	precision	recall	f1-score	support
0.0	0.88	0.78	0.82	9
1.0	0.96	0.98	0.97	55
accuracy			0.95	64
macro avg	0.92	0.88	0.90	64
weighted avg	0.95	0.95	0.95	64





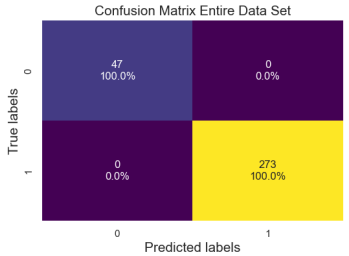
C

Classification Report (Entire Dataset):

	precision	recall	f1-score	support
0.0	1.00	1.00	1.00	47
1.0	1.00	1.00	1.00	273
accuracy			1.00	320
macro avg	1.00	1.00	1.00	320
weighted avg	1.00	1.00	1.00	320

Classification Report Test(Kept unseen data):

	precision	recall	f1-score	support
0.0	1.00	0.60	0.75	5
1.0	0.94	1.00	0.97	31
accuracy			0.94	36
macro avg	0.97	0.80	0.86	36
weighted avg	0.95	0.94	0.94	36



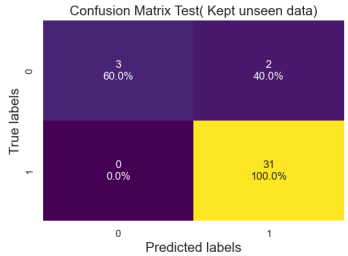
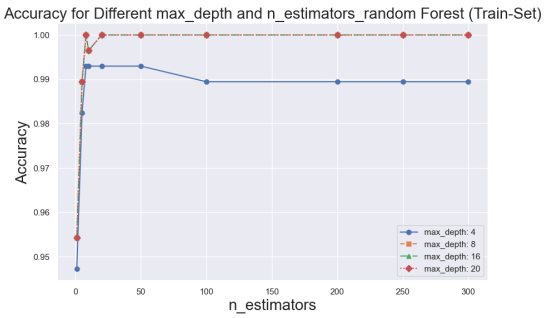


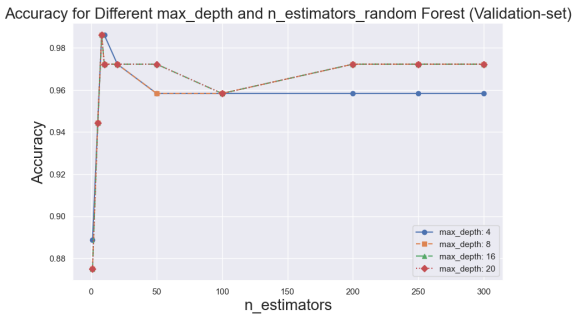
Figure 4. Predicting Prostate Cancer Outcomes Using Logistic Regression on RNA-seq Data. (A) The dataset was split into training (80%) and validation (20%) sets. The accuracy for both validation and training sets, as well as the training time, were plotted for an increasing number of iterations. The model converged after 35 iterations. The classification report and confusion matrices for the validation and training sets are shown on the right. The model achieved a final accuracy of 93% on the validation set, correctly predicting 95% of tumor tissue samples and 82% of normal samples, indicating that 18% of normal samples were misclassified as tumor samples.**(B)** The same model was tested on a separate set where 10% of the data was kept initially as unseen. The model was trained and validated on the remaining data. With the reduced set, the model reached convergence at 52 iterations, achieving an accuracy of 95% on the validation set, but with a higher recall for tumor samples (78%). The classification reports and corresponding confusion matrices are depicted. **(C)** The model was retrained with 52 iterations on the entire dataset (train plus validation) and tested on the unseen test data. The classification reports and confusion matrices for the whole dataset and the unseen test set are shown. While the model demonstrated good overall accuracy on the unseen test data by perfectly predicting tumor samples, it exhibited a lower recall for normal samples, correctly predicting only 60% of them, resulting in 40% false positive cancer predictions. Labels are 0 for solid tissue normal samples and 1 for prostate primary tumor samples.

Figure 5

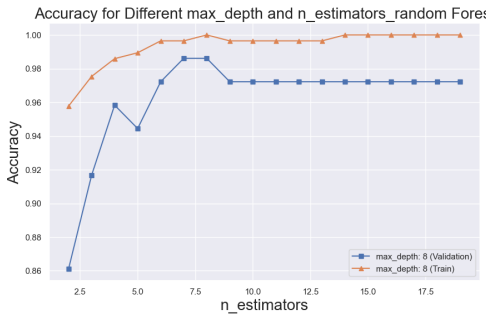
Random Forest Classifier with Single Random Split (Split: Train (80%)-Validation (20%))

A



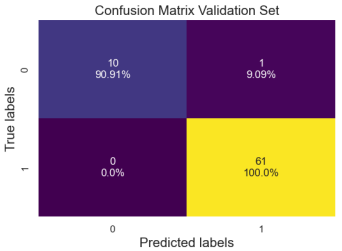


B



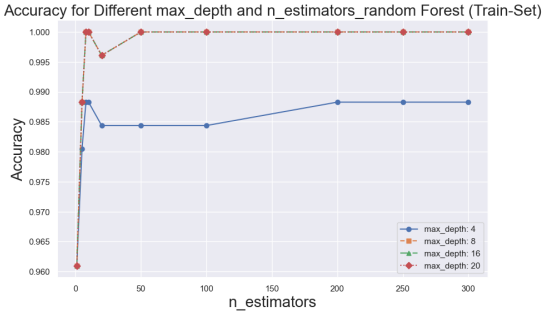
C

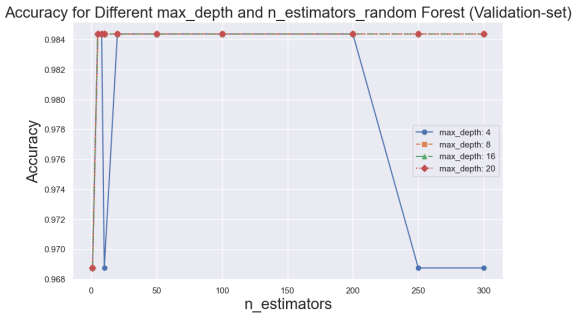
```
estimator: 7 depth: 8 val score:0.9861111111111112
 estimator: 7 depth: 8 train score:0.9964788732394366
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.91      0.95        11
         1.0       0.98      1.00      0.99        61

    accuracy                           0.99        72
   macro avg       0.99      0.95      0.97        72
weighted avg       0.99      0.99      0.99        72
```



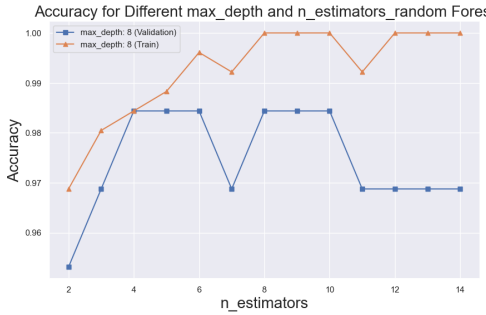
Random Forest Classifier with Single Random Split
(10% kept as unseen (test-set), rest split 80%-20% training-validation sets)

D



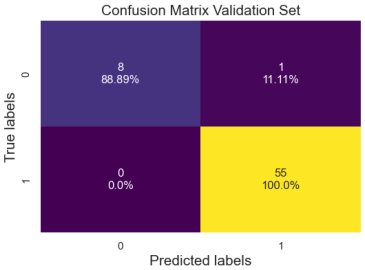


E

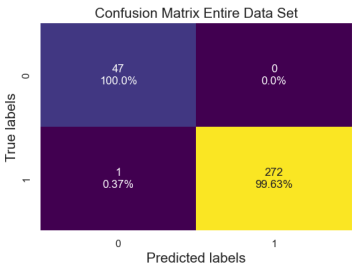


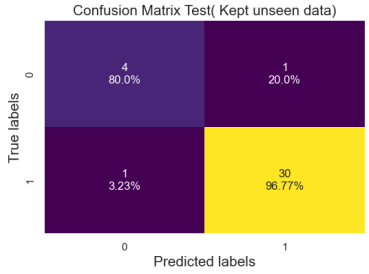
F

```
estimator: 4 depth: 8 val score:0.984375
 estimator: 4 depth: 8 train score:0.984375
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.89      0.94         9
         1.0       0.98      1.00      0.99        55

    accuracy                           0.98        64
   macro avg       0.99      0.94      0.97        64
weighted avg       0.98      0.98      0.98        64
```



G






```
Classification Report (Entire Dataset):
              precision    recall  f1-score   support

         0.0       0.98      1.00      0.99        47
         1.0       1.00      1.00      1.00       273

    accuracy                           1.00       320
   macro avg       0.99      1.00      0.99       320
weighted avg       1.00      1.00      1.00       320
```

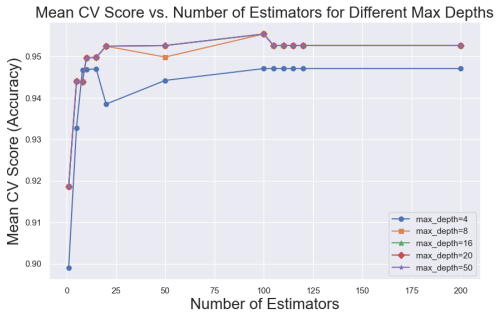
```
Classification Report Test( Kept unseen data):
              precision    recall  f1-score   support

         0.0       0.80      0.80      0.80         5
         1.0       0.97      0.97      0.97        31

    accuracy                           0.94        36
   macro avg       0.88      0.88      0.88        36
weighted avg       0.94      0.94      0.94        36
```

Figure 5. Predicting Prostate Cancer Outcome by Applying Random Forest on the Prostate Cancer RNA seq Data: Optimizing Random Forest with Single Random Split model tunning (A-C) The dataset was split into training (80%) and validation (20%) sets. **(A-B)** Accuracy in the training and validation sets was plotted for different numbers of estimators (n_estimators) and maximum depths (max_depth) to identify the best hyperparameters for model prediction. **(C)** The highest accuracy in the validation set was observed with 7 estimators (7 trees) and a max_depth of 8, achieving an accuracy of 99% in the validation set. The classification report and confusion matrix for the validation set under these setting is shown. **(D-F)** The random forest model was also tested on a separate set where 10% of the data was initially kept unseen. The model was trained and validated on the remaining data, split 80% for training and 20% for validation. **(D-E)** Model tuning involved testing different n_estimators (n_est) and maximun depth (max_depth) values. **(F)** The classification report and confusion matrix for the validation set with the best parameters (n_estimators=4, max_depth=8) showed that the model achieved a high accuracy of 98% in the validation set and improved recall for normal samples (89%), outperforming the logistic regression model. **(G)** The model was retrained with these parameters on the entire dataset (training plus validation) and then tested on the kept unseen test data. The model performed better than logistic regression, achieving an accuracy of 94% and significantly improving the recall for the underrepresented normal tissue samples (80%). Normal tissue samples are labeled as 0, and prostate primary tumor samples are labeled as 1.

Figure 6

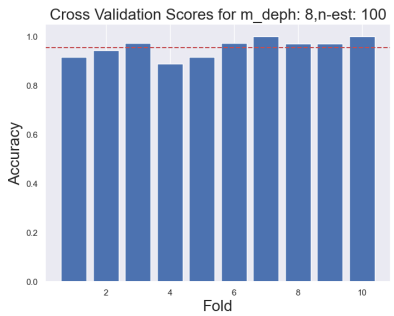
Random Forest Classifier with Crossvalidation (CV) (90% training-10% validation in each fold of CV)

A



Best Parameters: {**'n_estimators': 100, 'max_depth': 8}**
Best Mean CV Score: 0.9553

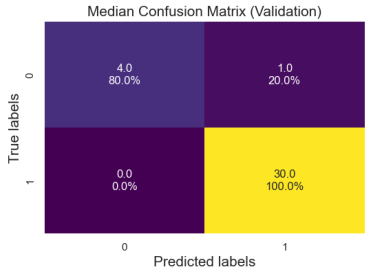
B

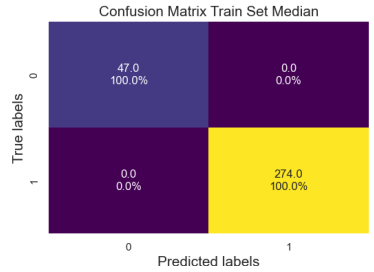


Mean CV Score: 0.9553
Median CV Score; 0.971

C

Median-confusion matrixes from cross validation folds
('n_estimators': 100, 'max_depth': 8)





Median Clasification Report (Validation):

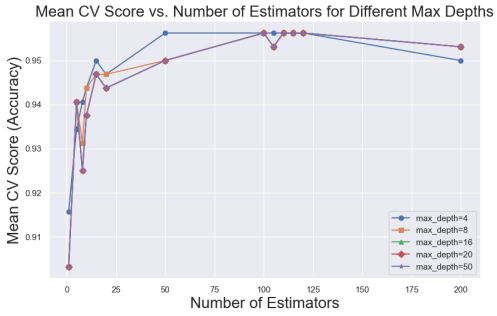
Class	Precision	Recall	F1-score	Support
0.0	1.00	0.80	0.89	5.0
1.0	0.97	1.00	0.98	30.0

Median Clasification Report (Train):

Class	Precision	Recall	F1-score	Support
0.0	1.00	1.00	1.00	47.0
1.0	1.00	1.00	1.00	274.0

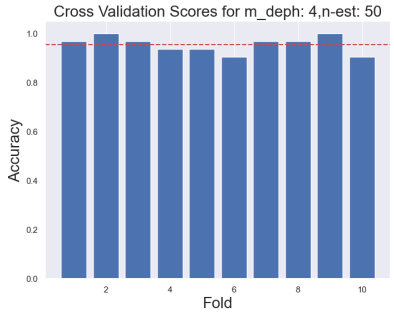
Random Forest Classifier with Crossvalidation (CV) (90% training-10% validation in each fold of CV)-10% kept unseen as test set

D



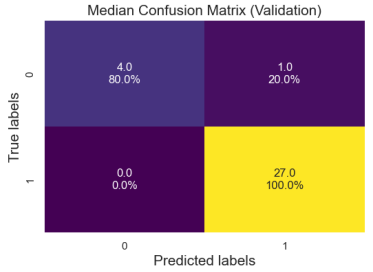
Best Parameters: {'n_estimators': 50, 'max_depth': 4}
Best Mean CV Score: 0.95625

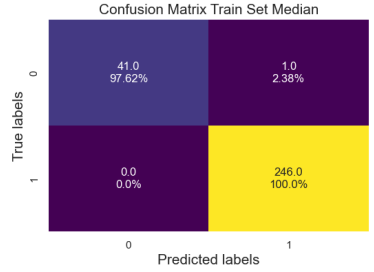
E



Mean CV Score: 0.95625
Median CV Score: 0.96875

F





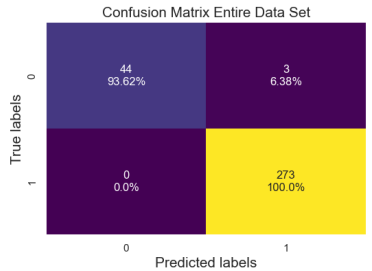
Median Clasification Report (Validation):

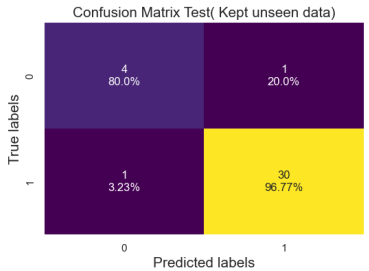
Class	Precision	Recall	F1-score	Support
0.0	1.00	0.80	0.89	5.0
1.0	0.96	1.00	0.98	27.0

Median Clasification Report (Train):

Class	Precision	Recall	F1-score	Support
0.0	1.00	0.98	0.99	42.0
1.0	1.00	1.00	1.00	246.0

G





Classification Report (Entire Dataset):

	precision	recall	f1-score	support
0.0	1.00	0.94	0.97	47
1.0	0.99	1.00	0.99	273
accuracy			0.99	320
macro avg	0.99	0.97	0.98	320
weighted avg	0.99	0.99	0.99	320

Classification Report Test(Kept unseen data):

	precision	recall	f1-score	support
0.0	0.80	0.80	0.80	5
1.0	0.97	0.97	0.97	31
accuracy			0.94	36
macro avg	0.88	0.88	0.88	36
weighted avg	0.94	0.94	0.94	36

Figure 6. Predicting Prostate Cancer Outcomes Using Random Forest on RNA-seq Data: Optimizing Random Forest Parameters with Cross-Validation (CV). **(A-C)** StratifiedKFold splitting was applied during cross-validation. During each fold, the data was split into 90% training and 10% validation. The cross-validation (CV) score was set to calculate accuracy. **(A)** The mean CV score (accuracy) across the 10 folds of cross-validation was plotted for different max_depth and n_estimators**. (B)** CV scores (accuracy) for each fold using the best combination of tested parameters (n_estimators=100, max_depth=8) are shown. With these parameters, the model achieved a median accuracy of 97.1%. **(C)** The median classification report and the median confusion matrix across folds of cross-validation on the validation and training sets are shown for the best parameters. **(D-E)** 10% of the data was kept as unseen for final validation. The remaining data was trained as in A**. (D)** Shows the mean CV score (accuracy) across the folds. **(E)** Depicts the accuracy across folds for the best parameters found in D (n_estimators=50, max_depth=4). **(F)** Shows the median classification report and the median confusion matrix across folds of cross-validation on the validation and training sets for the best parameters. **(G)** The model was retrained with the best parameters found in D-E on the entire dataset (training plus validation samples). After training the model was tested on the kept unseen data (test set). The confusion matrix and classification reports for the entire dataset and the test set are shown. The model showed an accuracy of 94% on the unseen data, with a recall for the underrepresented normal tissue category of 80%, indicating a false positive rate for cancer prediction of 20%. The false negative rate in detecting cancer samples was 3%. The model outperformed logistic regression principally, in predicting non-cancer samples and reducing the rate of false positives. Normal tissue samples are labeled as 0, and prostate primary tumor samples are labeled as 1.

Figure 7

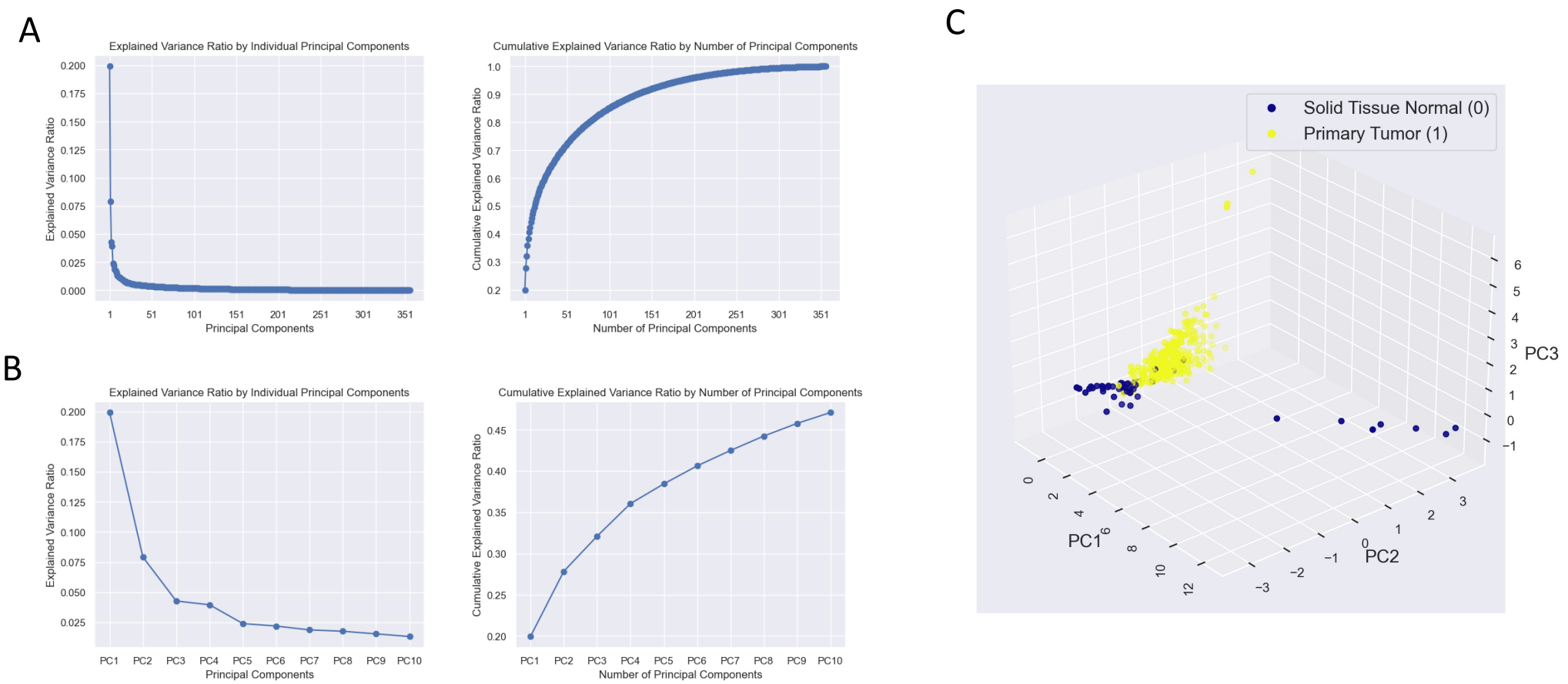
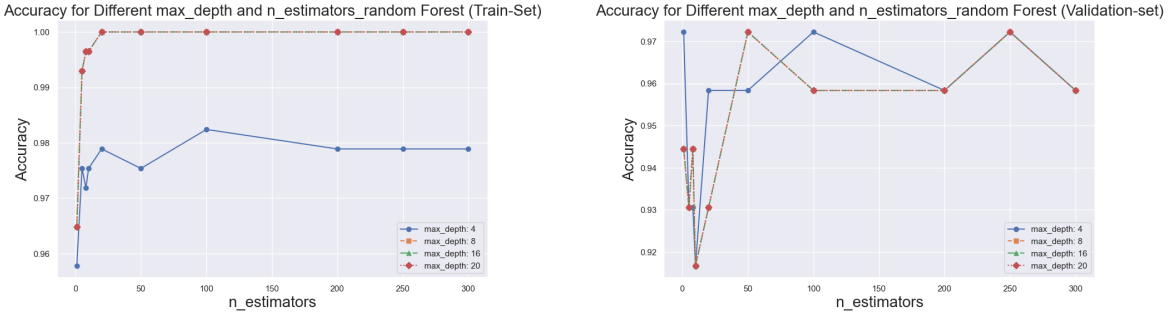


Figure 7. Principal Components Analysis (PCA). (A) Explained variance ratio and cumulative explained variance ratio for all the detected principal components (PCs). **(B)** Same as in (A) but showing only the first 10 principal components. **(C)** 3D plot showing the PC1, PC2, and PC3 values for all the samples. The solid normal tissue samples (category 0) are depicted in yellow, and prostate primary tumor samples (category 1) are shown in blue. These three PCs can separate the different categories, with a few normal samples still overlapping the primary tumor samples and viceversa.

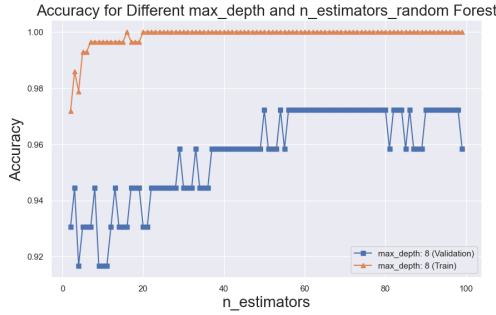
Figure 8

Random Forest Classifier with Single Random Split and the 2 most important Principal Components (PCs) (Set Split: Train (80%)-Validation (20%))

A

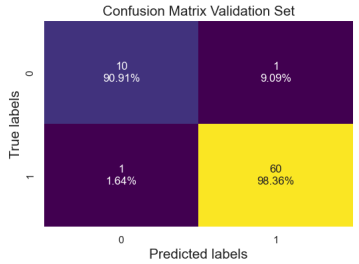


B



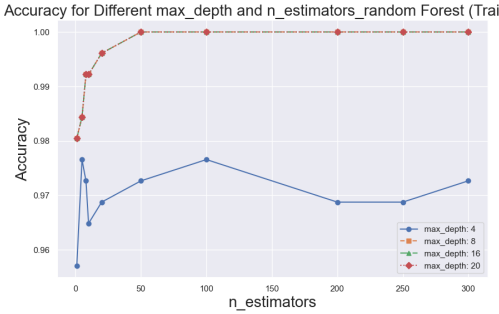
C

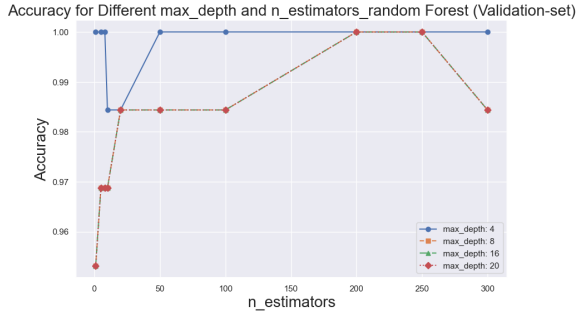
```
estimator: 60 depth: 8 val score:0.9722222222222222
 estimator: 60 depth: 8 train score:1.0
Classification Report:
              precision    recall  f1-score   support

         0.0       0.91      0.91      0.91        11
         1.0       0.98      0.98      0.98        61

    accuracy                           0.97        72
   macro avg       0.95      0.95      0.95        72
weighted avg       0.97      0.97      0.97        72
```



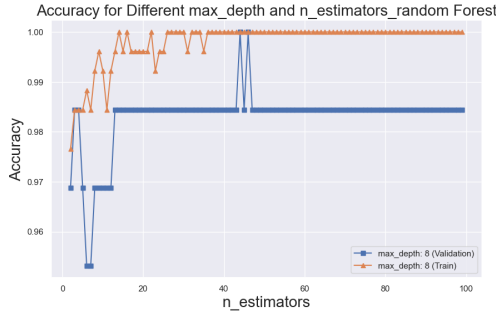
Random Forest Classifier with Single Random Split and the 2 most important PCs (10% keep unseen (test-set), rest split 80%-20% training-validation)

D



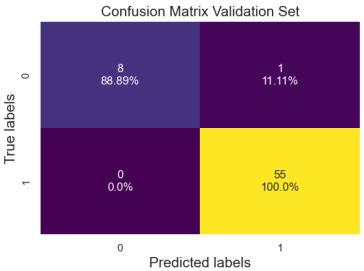


E

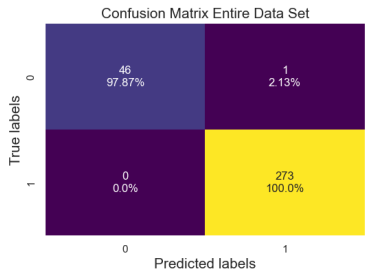


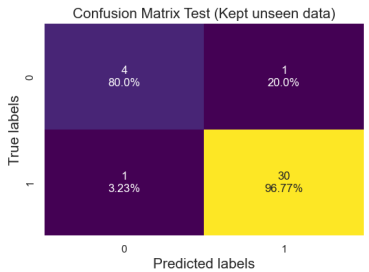
F

```
estimator: 15 depth: 8 val score:0.984375
 estimator: 15 depth: 8 train score:0.99609375
Classification Report:
              precision    recall  f1-score   support

         0.0       1.00      0.89      0.94         9
         1.0       0.98      1.00      0.99        55

    accuracy                           0.98        64
   macro avg       0.99      0.94      0.97        64
weighted avg       0.98      0.98      0.98        64
```



G






```
Classification Report (Entire Dataset):
              precision    recall  f1-score   support

         0.0       1.00      0.98      0.99        47
         1.0       1.00      1.00      1.00       273

    accuracy                           1.00       320
   macro avg       1.00      0.99      0.99       320
weighted avg       1.00      1.00      1.00       320
```

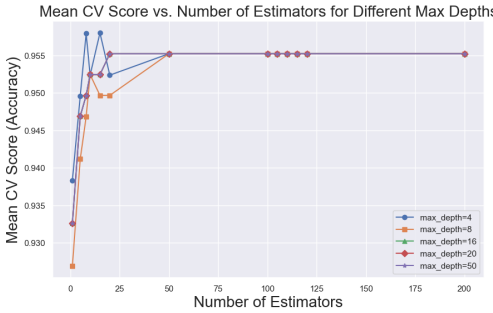
```
Classification Report Test(Kept unseen data):
              precision    recall  f1-score   support

         0.0       0.80      0.80      0.80         5
         1.0       0.97      0.97      0.97        31

    accuracy                           0.94        36
   macro avg       0.88      0.88      0.88        36
weighted avg       0.94      0.94      0.94        36
```

Figure 8. Random Forest Classifier Applied to Principal Components (PCs). Optimization of Random Forest parameters using a Single Random Split model tuning, with the first two principal components (PC1 and PC2). (A-C) The dataset was split into training (80%) and validation (20%) sets. **(A-B)** Accuracy in the training and validation sets was plotted for different numbers of estimators (n_estimators) and maximum depths (max_depth) to identify the best parameters for model prediction using these two PCs. **(C)** The highest accuracy in the validation set was achieved with 60 estimators and a max_depth of 8, resulting in an accuracy of 97%. The recall for the underrepresented normal sample category (0) was 91% and 98 for tumor samples, indicating a false posite rate of 9% and a false negative rate of 3%. The classification report and confusion matrix for the validation set under these settings are shown. **(D-G)** The Random Forest model was further tested on a separate set where 10% of the data was initially kept unseen. The remaining data was split 80% for training and 20% for validation. **(D-E)** Model tuning involved testing various n_estimators (n_est) and max_depth values. **(F)** The classification report and confusion matrix for the validation set with the best parameters (n_estimators=15, max_depth=8) showed an accuracy of 98% and an improved recall for normal samples (89%), outperforming the logistic regression model and achieving similar performance to the model using all 1380 features. **(G)** The model was retrained with these optimal parameters on the entire dataset (training plus validation) and then tested on the previously unseen test data. The model achieved an accuracy of 94% on the unseen data, with a recall of 80% for the underrepresented normal tissue category (20% false positives) and a false negative rate of 3% in detecting cancer samples. The model performed similarly to when all features were used for classification (Figure 5, Table 6). Labels are 0 for solid tissue normal samples and 1 for prostate primary tumor samples.

Figure 9

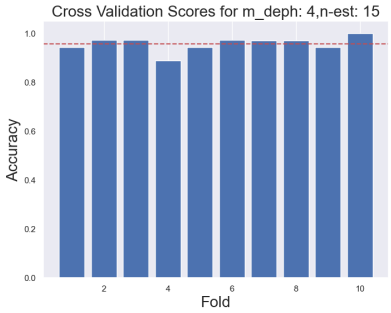
Random Forest Classifier on PC1 and PC2 with Crossvalidation (CV) (90% training-10% validation in each fold of CV)

A



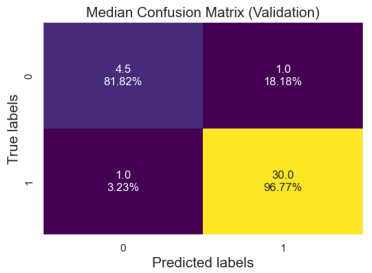
Best Parameters: {'n_estimators': 15, 'max_depth': 4}

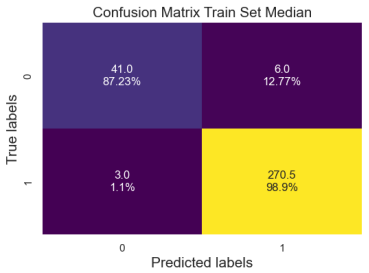
B



Mean CV Score: 0.958
Median CV Score; 0.971

C



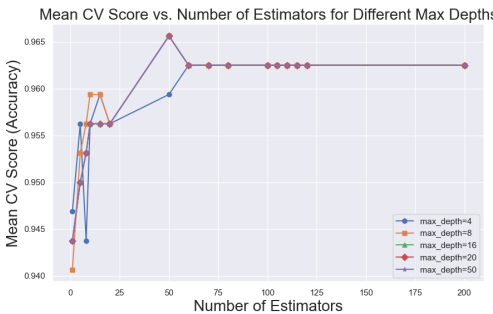



```
training with parameters:m_deph: 4,n_est: 15

Median Clasification Report (Validation):
Class        Precision  Recall     F1-score   Support
0.0          0.83       0.82       0.90       5.0
1.0          0.97       0.97       0.98       30.0
```

```
Median Clasification Report (Train):
Class        Precision  Recall     F1-score   Support
0.0          0.93       0.87       0.89       47.0
1.0          0.98       0.99       0.98       274.0
```

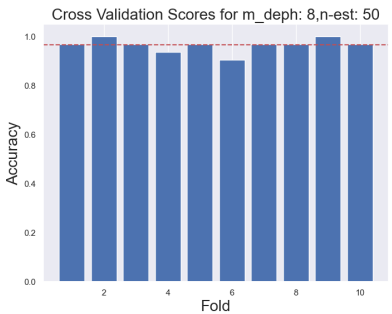
Random Forest Classifier PC1 and PC2 with Crossvalidation (CV) (90% training-10% validation in each fold of CV)-10% kept unseen as test set

D



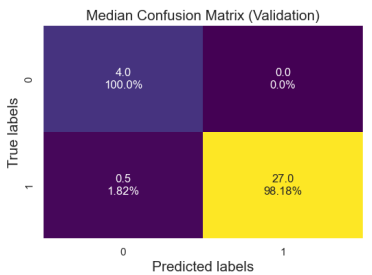
Best Parameters: {'n_estimators': 50, 'max_depth': 8}

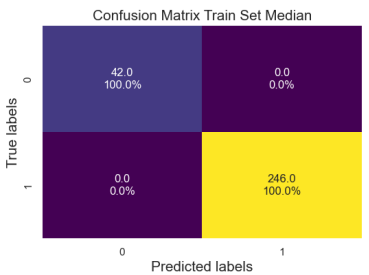
E



Mean CV Score: 0.966
Median CV Score; 0.969

F

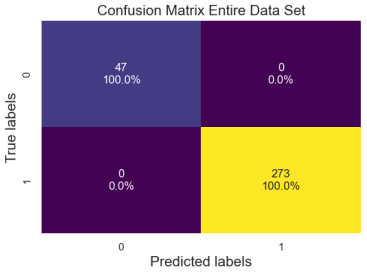


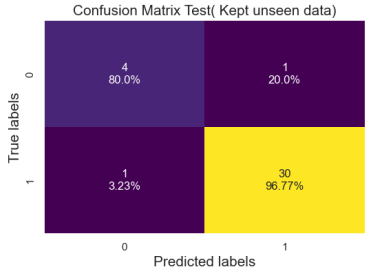



```
training with parameters:m_deph: 8,n_est: 50

Median Clasification Report (Validation):
Class        Precision  Recall     F1-score   Support
0.0          0.92       1.00       0.89       5.0
1.0          1.00       0.98       0.98       27.0
```

```
Median Clasification Report (Train):
Class        Precision  Recall     F1-score   Support
0.0          1.00       1.00       1.00       42.0
1.0          1.00       1.00       1.00       246.0
```

G





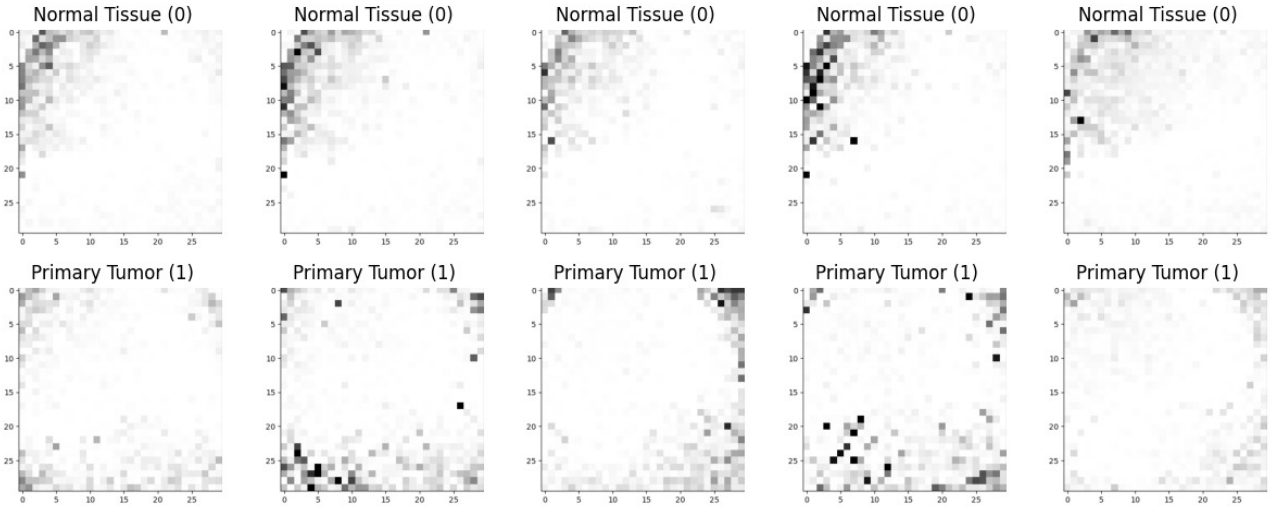

```
training with parameters:m_deph: 8,n_est: 50

Classification Report (Entire Dataset):
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00        47
         1.0       1.00      1.00      1.00       273

    accuracy                           1.00       320
   macro avg       1.00      1.00      1.00       320
weighted avg       1.00      1.00      1.00       320
```

```
Classification Report Test( Kept unseen data):
              precision    recall  f1-score   support

         0.0       0.80      0.80      0.80         5
         1.0       0.97      0.97      0.97        31

    accuracy                           0.94        36
   macro avg       0.88      0.88      0.88        36
weighted avg       0.94      0.94      0.94        36
```

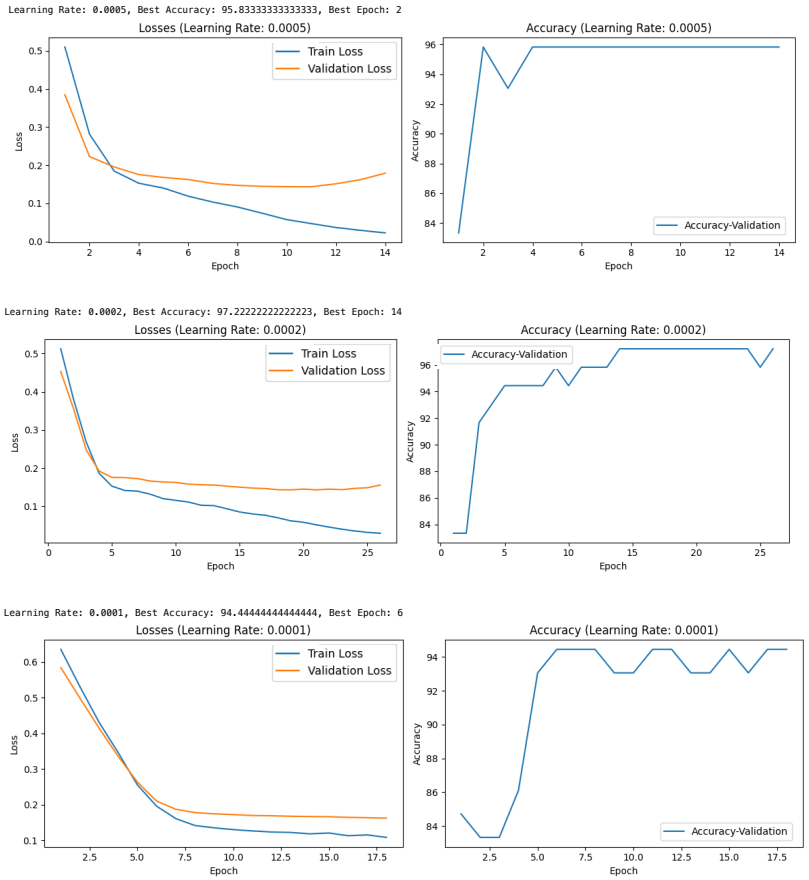
Figure 9. Random Forest Classifier Applied to Principal Components (PCs). Optimization of Random Forest parameters using cross-validation with the first two principal components (PC1 and PC2) for classification. StratifiedKFold splitting was applied during cross-validation, with accuracy used as the cross-validation (CV) score. During each fold, the data was split into 90% training and 10% validation. **(A)** The mean CV score (accuracy) across the 10 folds of cross-validation plotted for different max_depth and n_estimators values. **(B)** CV scores (accuracy) for each fold using the best combination of tested parameters (n_estimators=15, max_depth=4). The model achieved a median accuracy of 97.1%. **(C)** The median classification report and the median confusion matrix across folds of cross-validation on the validation and training sets for the best parameters are shown. **(D-G)** 10% of the data was kept unseen for final validation. The remaining data was trained as in (A).**(D)** Mean CV score (accuracy) across the folds. **(E)** Accuracy across folds for the best parameters found in (D) (n_estimators=50, max_depth=8). **(F)** The median classification report and the median confusion matrix across folds of cross-validation on the validation and training sets for the best parameters. **(G)** The model was retrained with the best parameters found in (D-E) on the entire dataset (training plus validation samples). After training, the model was tested on the kept unseen test set. The confusion matrix and classification reports for the entire dataset and the test set are shown. The model achieved an accuracy of 94% on the unseen data, with a recall of 80% for the underrepresented normal tissue category (20% false positives) and a false negative rate of 3% in detecting cancer samples. The model's performance using the reduced features was comparable to using all 1380 features (Table 6). Normal tissue samples are labeled as 0, and prostate primary tumor samples are labeled as 1.

Figure 10

A

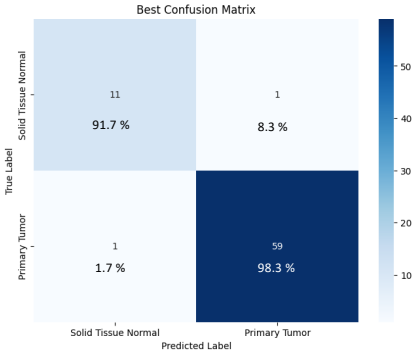


B



BEST SELECTED HYPERPARAMETERS:
Best Parameters: {'learning rate': 0.0002, 'best epoch': 14},
Best Accuracy: 97.22222222222223

C




```
Classification Report-Test:
                precision    recall  f1-score   support

           0       0.92      0.92      0.92        12
           1       0.98      0.98      0.98        60

    accuracy                           0.97        72
   macro avg       0.95      0.95      0.95        72
weighted avg       0.97      0.97      0.97        72
```

D

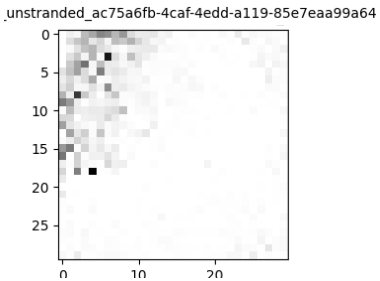


Image is (0=Normal, 1=Tumor): 1
Image was predicted as (0=Normal, 1=Tumor): 0.0

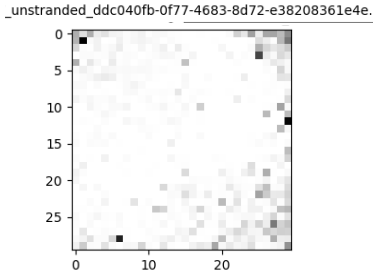


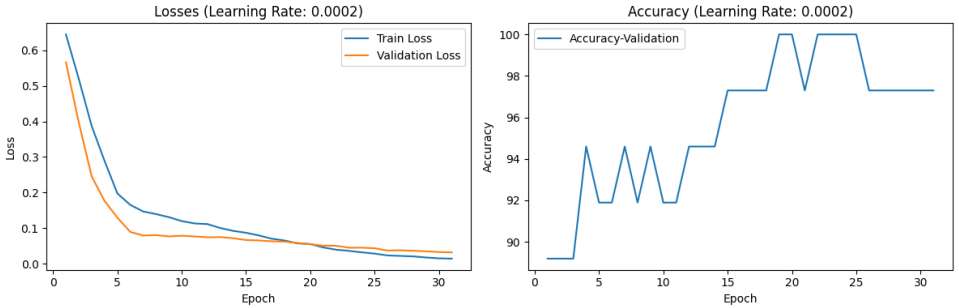
Image is (0=Normal, 1=Tumor): 0
Image was predicted as (0=Normal, 1=Tumor): 1.0

Figure 10. Convolutional Neural Networks (CNN) Applied to Prostate Cancer Tabular RNAseq Data Converted to Images for Cancer Prediction. (A) Tabular RNA-seq data (derived from the preselected 1380 features) was transformed into images using the IGTD algorithm (See Methods). Random examples of the generated images for Normal Tissue (category 0) and Primary Prostate Tumor samples (category 1) are displayed. **(B)** A CNN was then 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 data loaders were divided into 80% training and 20% validation sets. An example of a single run of the CNN is shown, with loss for the training and validation sets, as well as accuracy in the validation set, depicted with different learning rates (lr) over various epochs. The best hyperparameters were selected using early stopping (patience = 12). **(C)** The Confusion Matrix and Classification report for the validation set using the best parameters selected for this run (lr = 0.0002, epoch = 14) are displayed. The model achieved an accuracy of 97% on the validation data set, with a recall of 92% for the underrepresented normal tissue category (7 % false positives) and a false negative rate of 2% in detecting cancer samples. **(D)** Images and file names of the incorrectly predicted samples for this specific run are shown.

Figure 11

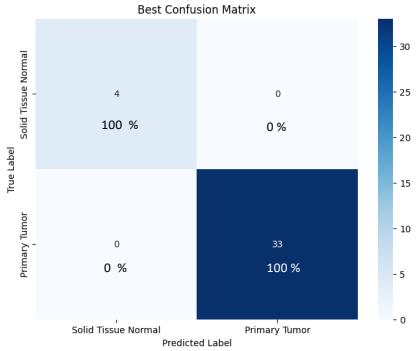
A

CNN (Split : 10% keep unseen (test-set), 80% train , 10% validation)



BEST SELECTED HYPERPARAMETERS:
Best Parameters: {'learning_rate': 0.0002, 'best_epoch': 19}
Best Accuracy: 100.0

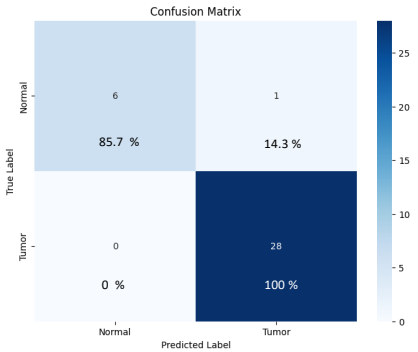
B



Best Classification Report-Test:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	4
1	1.00	1.00	1.00	33
accuracy			1.00	37
macro avg	1.00	1.00	1.00	37
weighted avg	1.00	1.00	1.00	37…

C



…Classification Report Test (Unseen data):

	precision	recall	f1-score	support
0	1.00	0.86	0.92	7
1	0.97	1.00	0.98	28
accuracy			0.97	35
macro avg	0.98	0.93	0.95	35
weighted avg	0.97	0.97	0.97	35

…

D

_unstranded_e3c5a10a-81cf-459b-82f5-0448d8f1665d.

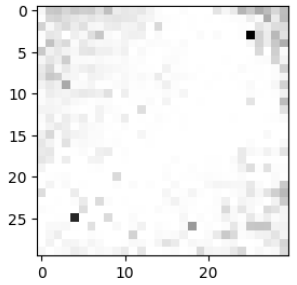


Image is (0=Normal, 1=Tumor): 0
Image was predicted as (0=Normal, 1=Tumor): 1.0

Figure 11. Convolutional Neural Networks (CNN) Applied to Prostate Cancer Tabular RNAseq Data Converted to Images for Cancer Prediction: Keeping 10% Data Unseen for Final Evaluation. (A) The same CNN architecture as in Figure 11 was utilized, but 10% of the data was reserved as unseen before training. 80% used for training and 10% as validation set during training for hyperparameter tunning. 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. In this particular run, the accuracy and recall for both types of samples were perfect. **(C)** The CNN was retrained with the entire dataset, consisting of the validation and train sets, using the best hyperparameters set in (A). 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 show an accuracy of 97%, similar to the random forest model. However, the model could improve the recall in the normal samples on unseen data to 86% (14% false positives). 100% recall was observed for tumor samples.

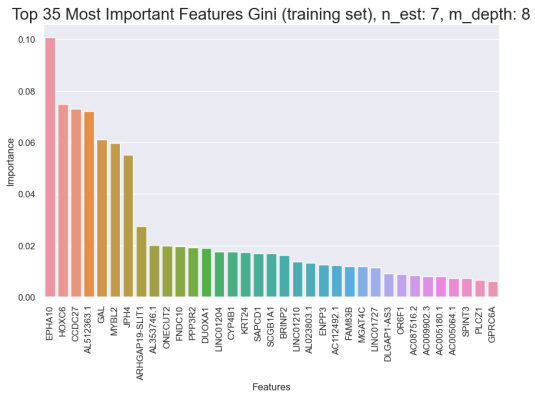
Data / Features	Model	Traning and evaluation	Accuracy Validation set	Recall Validaton set	Accuracy Test set (unseen data)	Recall Test set (unseen data)
-Tabular data -Features-reduction from DE analysis: log 2-fold> 2 or log 2 -old <-2 and padj <0.05 : 1380 genes	Logistic Regression	35 itinerations	0.931	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)

Table 6. Summary of the Performance of the Different Models Tested, along with the Types of Training, Validation, and Testing used. Accuracy and recall for the normal tissue category (0) and the prostate primary tumor category (1) in the validation and test sets, when applicable, are shown. For the CNN, the model was run the indicated numer of times, and the mean values of accuracy and recall are depicted. The random forest and logistic regression models did not vary in performance across different runs since we used a random seed of 42 to ensure consistent comparison of model performances. The CNN showed slightly better performance, achieving higher mean accuracies and recall on the unseen data, especially for the underrepresented normal tissue category. In yellow are mark the training and evaluation test, were 10 % of the data was kept unseen for final testing of the model performance.

Figure 12

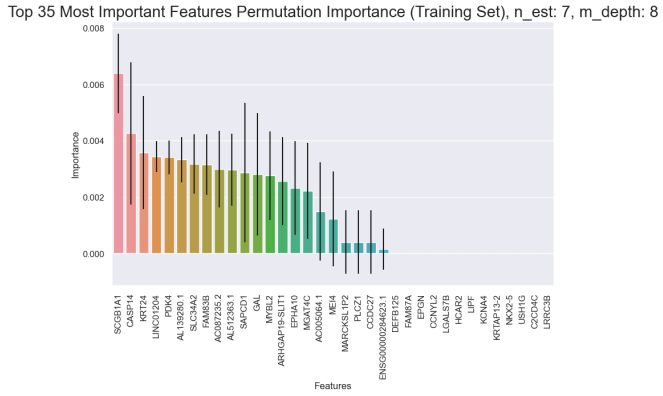
A

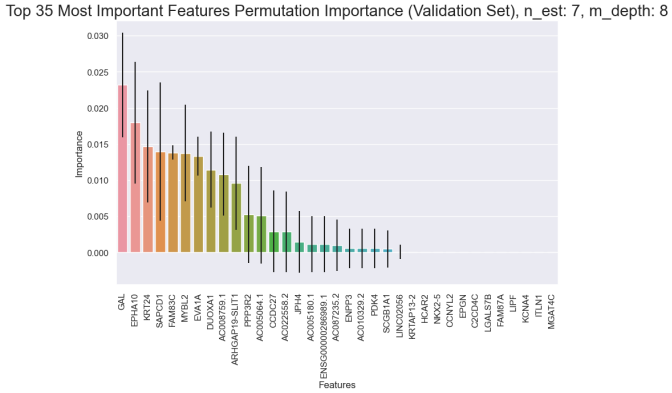
Gini importance



B

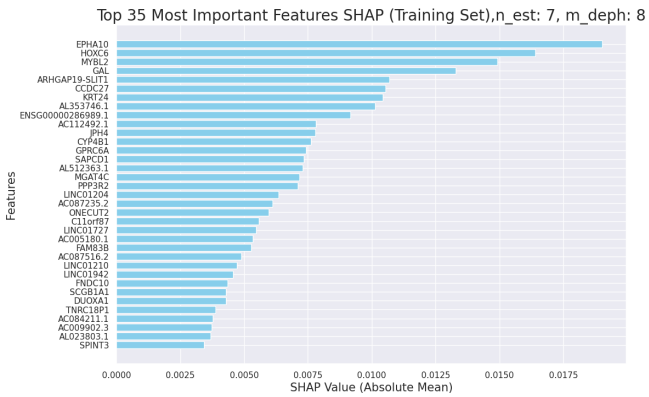
Permutation Importance

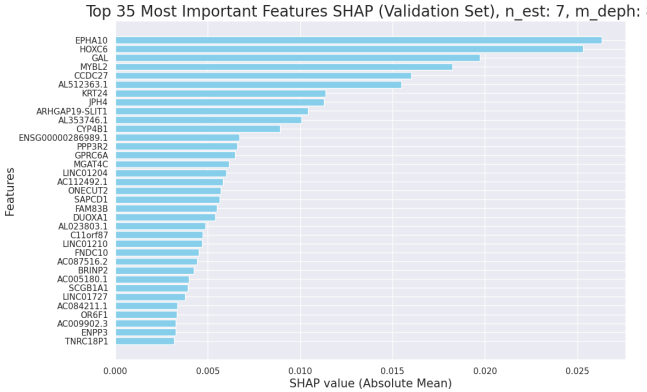




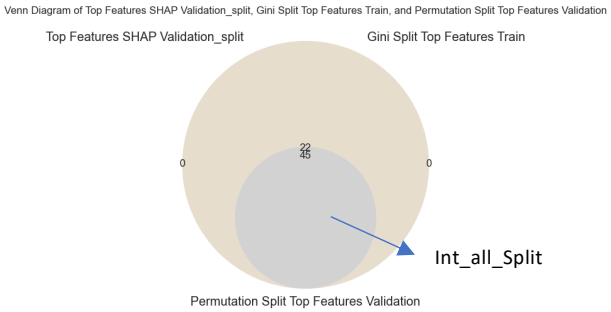
C

SHAP explainer

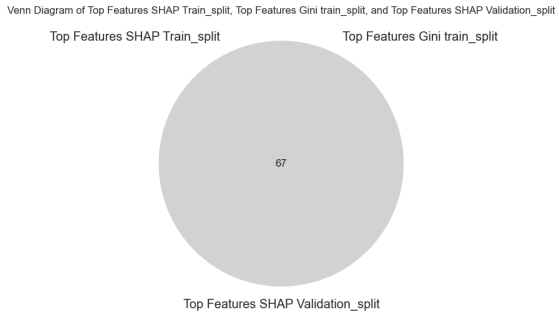




D



Int_all_split = Common genes in all sets: 'FAM83C', 'ARHGAP19-SLIT1', 'GAL', 'SAPCD1', 'EPHA10', 'KRT24', 'EVA1A', 'PPP3R2', 'DUOXA1', 'ENPP3', 'AC005064.1', 'AC022558.2', 'PDK4', 'MYBL2', 'AC008759.1', 'JPH4', 'SCGB1A1', 'AC010329.2', 'CCDC27', 'AC087235.2', 'ENSG00000286989.1', 'AC005180.1'



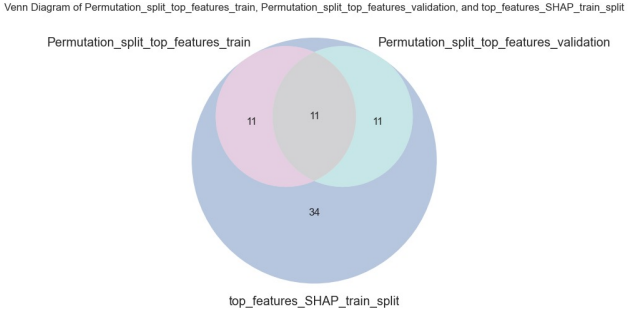
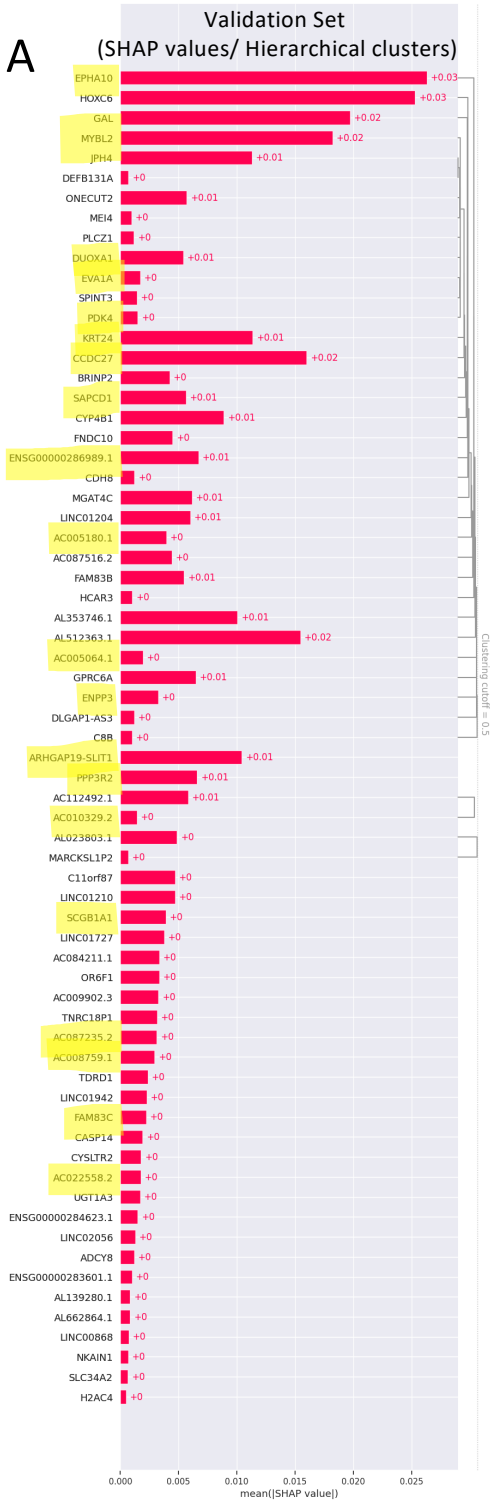


Figure 12. Importance Features Analysis to Identify Genes Important for Cancer Prediction as Putative Biomarkers or Therapeutic Targets. The feature importance analysis was conducted to identify genes crucial for cancer prediction using the Random Forest Classifier optimized by a single random split. The data was split into 80% for training and 20% for validation, and the model was trained with the best hyperparameters selected in Figure 5C. **(A)** The top 35 important features with the highest absolute importance values calculated using Gini importance values, which indicate the importance of each feature calculated internally within the Random Forest classifier, are shown for the training set. **(B)** The top 35 important features with the highest absolute importance values calculated using permutation importance in both the training and validation sets are displayed. **(C)** The top 35 important features with the highest SHAP (SHapley Additive exPlanations) values in both the training and validation sets are presented. **(D)** Features/genes with absolute importance/SHAP values greater than 0 were selected for further analysis. For Gini importances and SHAP importances, 67 genes with absolute importance values greater than 0 were identified in each set. Permutation importances identified only 22 genes in the training and validation sets with absolute importances greater than 0. Venn diagrams illustrate the intersection between the genes found important for model prediction through the different tests conducted in (A-C).



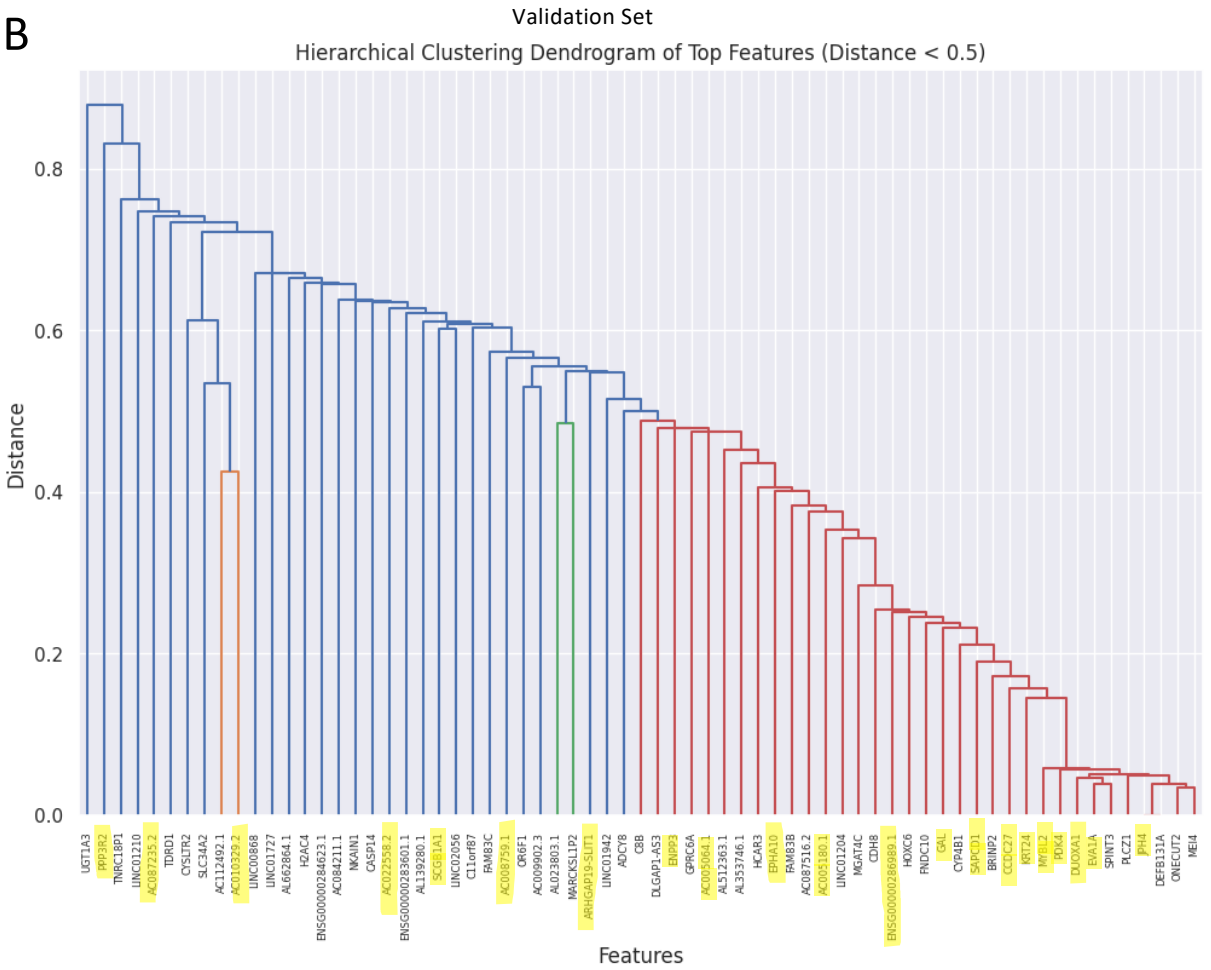
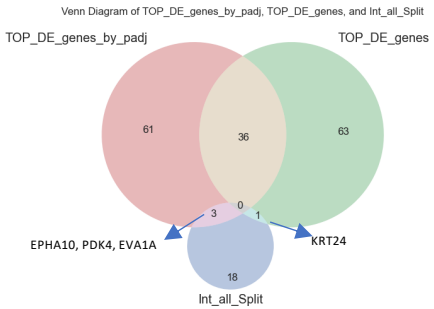


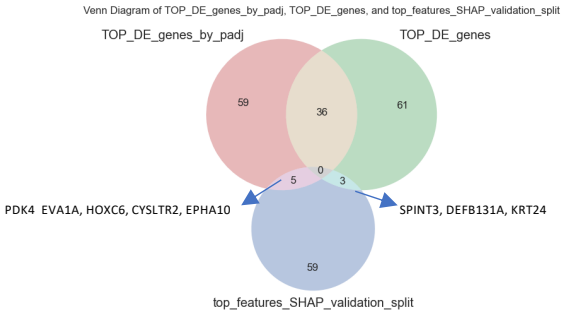
Figure 13

Figure 13. Hierarchical Clustering of the Top 67 Genes Identified by Feature Importance Using SHAP Values show genes are correlated. Hierarchical clustering was performed using the internal hierarchical clustering function of the SHAP (SHapley Additive exPlanations) utils package. The distance metric was set to default to xgboost_distances_r2, and the linkage method was single by default. **(A)** Displays the SHAP values alongside the clustering of the features. **(B)** Illustrates the clusters in the form of a dendrogram. Clusters with a distance less than 0.05 are colored. Genes also identified as important by permutation importances in the validation set are marked in yellow in A and B.

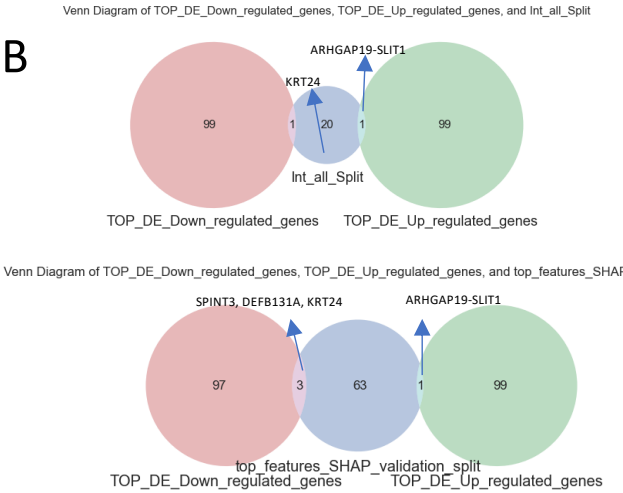
Figure 14

A

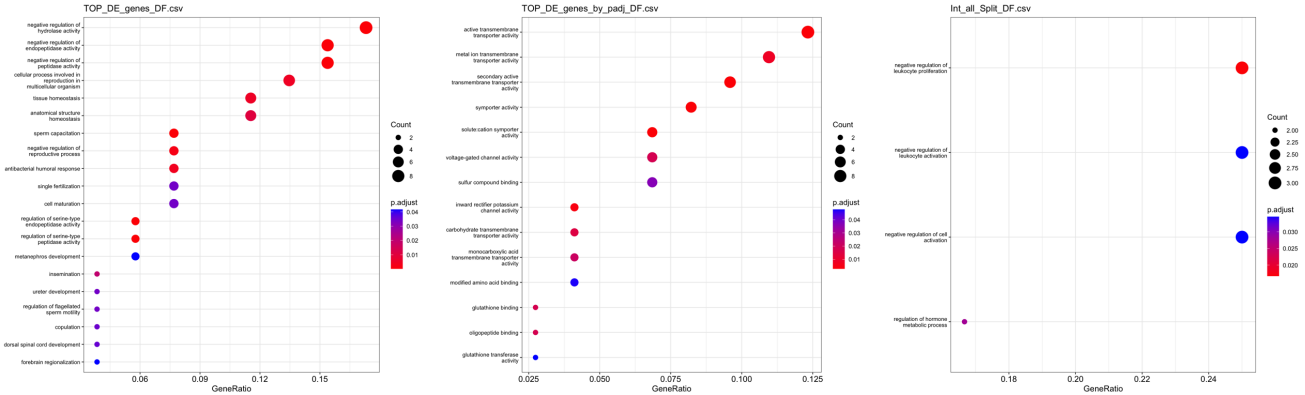




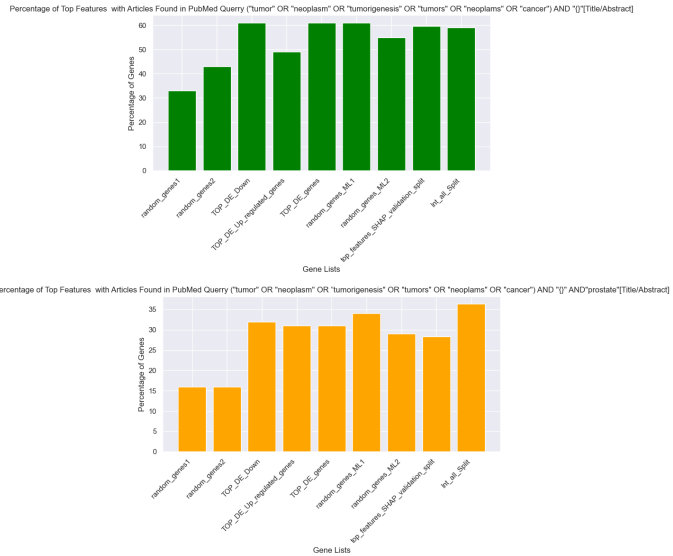
B



C



D



E

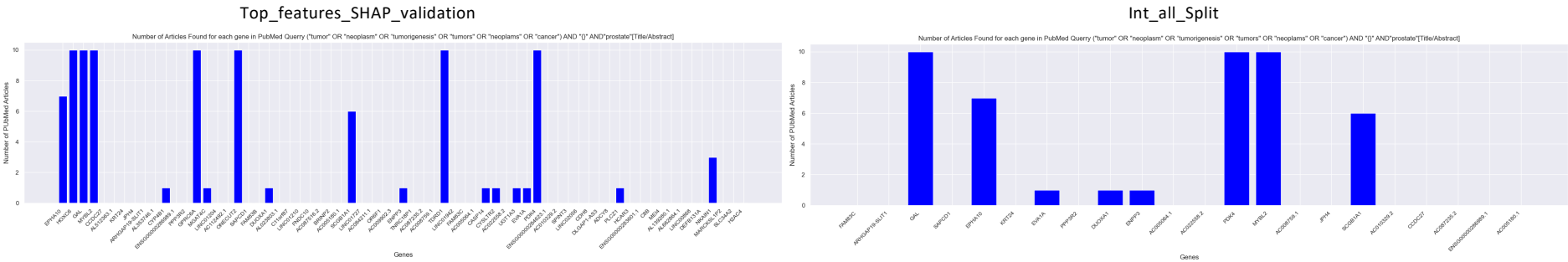


Figure 14. Identification of Genes Important for Predicting Prostate Cancer Outcome and Comparison with Differentially Expressed (DE) Genes. (A) Venn diagrams depict the intersection of the 100 top DE expressed genes selected based on either higher log2FoldChange (Top_DE_genes) or lowest padj value (Top_DE_genes_by_padj) with the genes identified in Figure 12. The intersections include either the 22 genes found importance by the 3 algorithms (Int_all_split) and the 67 genes found important with the SHAP and Gini importance algorithms (Top_features_validation_SHAP_split). **(B)** Additional Venn diagrams show the overlap between the genes selected by feature importance and the 100 top upregulated and downregulated DE genes chosen based on higher absolute log2FoldChange values. **(C)** Gene ontology analysis reveals biological pathway enrichment for the 100 top DE genes and those deemed important for model prediction. Notably, pathways related to leukocyte activation are enriched in the 22 genes identified by all feature importance algorithms (Int_all_split set), while the 67 genes identified with SHAP and Gini importance algorithms (Top_features_validation_SHAP_split) show no significant enrichment. **(D)** PubMed searches were conducted to retrieve the number of genes with at least one publication containing the gene name and cancer-related terms in the abstract or title of the publication. The percentage of genes retrieving at least one publication with these terms is shown in the top panel (green bars). If the abstract or title additionally included the term "prostate," the plot of the percentage of genes that retrieved any publication with this additional term is shown in the lower panel (orange bars). **(E)** The number of publications (up to 10) found for each gene in the list of genes identified as important for predicting the outcome in the Random Forest model is presented. Among the lists (Int_all_Split and, Top_features_validation_SHAP_split) there are several genes already known to be involved in prostate cancer or used as markers in prostate cancer. Additionally, the script is capable of retrieving the title, abstract, and citation for these publications. Arrows in A and B depicted the genes in the intersections.

Figure 15

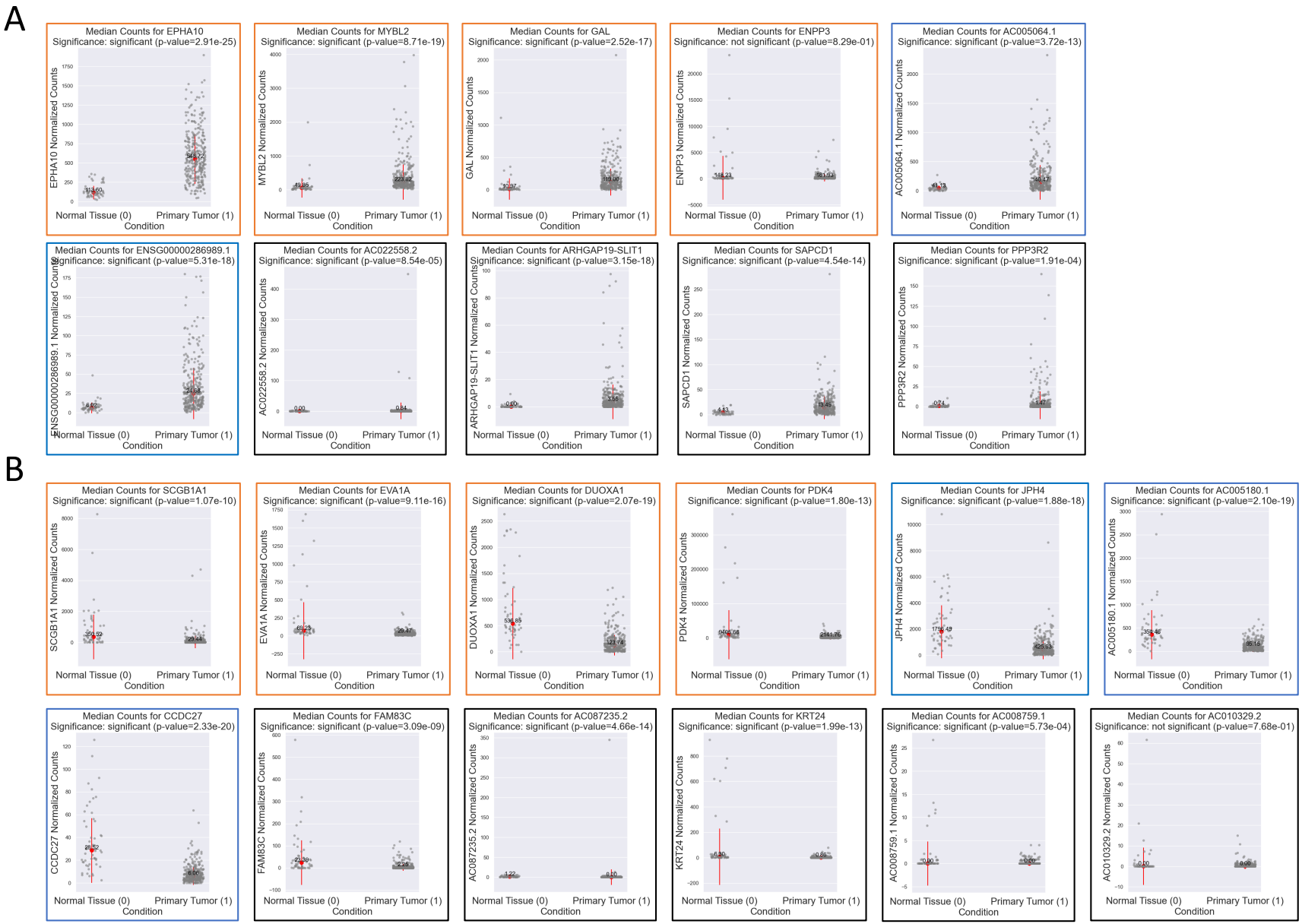
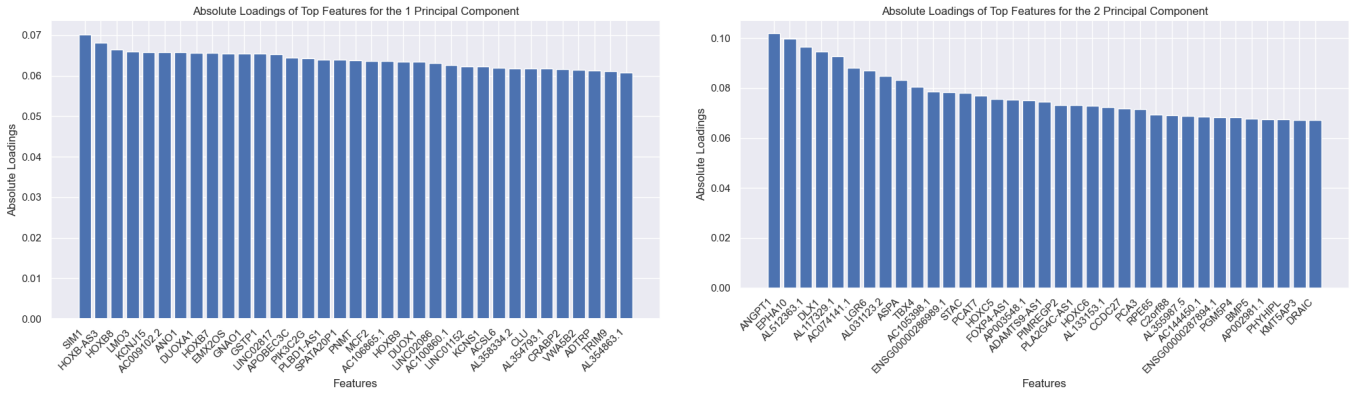


Figure 15. Expression of the Genes Identified to be Important for Outcome Prediction. Normalized raw counts, used as input for the ML models, are depicted for the genes identified as important for model performance by all feature importance algorithms (Int_all_Split). The significant difference in normalized counts between normal and cancer tissue samples was assessed using the Mann-Whitney U test. Among these genes, several that are published to be related to prostate cancer development (Figure 14E and surrounded by an orange marker) were identified and tend to exhibit higher normalized counts across samples. **(A)** Shows the genes that were detected as upregulated by pyDeseq2 and **(B)** shows the downregulated genes.

Figure 16

A



B

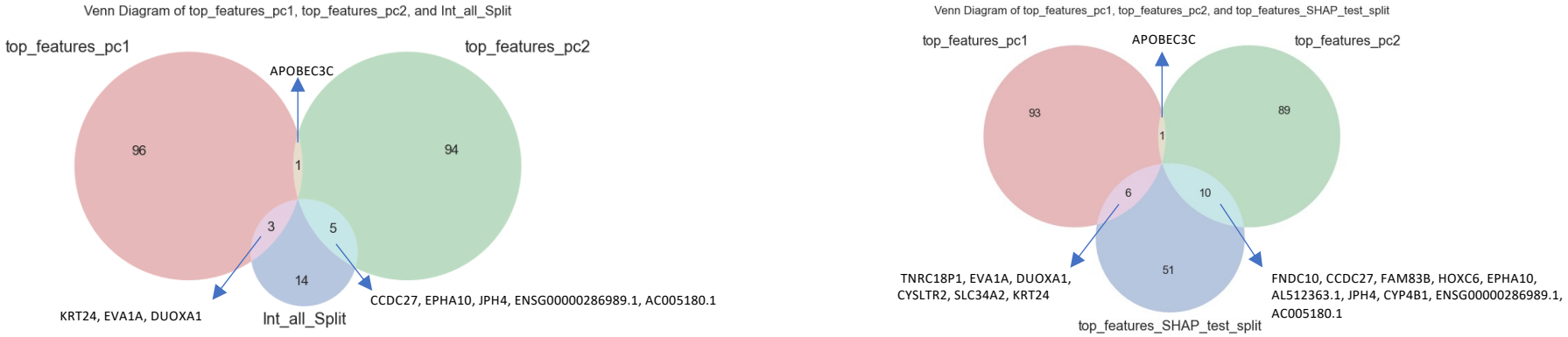


Figure 16. Important Features Identified by PCA Loadings. (A) The top 35 features with the highest absolute loadings for PC1 and PC2 are shown. **(B)** Venn diagrams showing the overlap between the top 100 features identified by PCA loadings and the sets identified by feature importance in the random forest classification (sets Int_all_Split and Top_features_SHAP_split). Common genes found in the intersections of the Venn diagrams are indicated (Arrows).