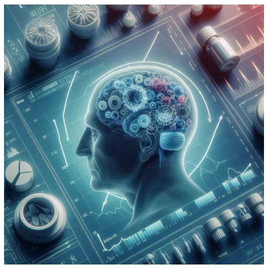
Jul 23, 2024

In-Silico Validation of Biomarkers using ROC and AUC Curve Analysis in R: A Comprehensive Protocol



DOI
dx.doi.org/10.17504/protocols.io.q26g713qqgwz/v1

Adarsh V[1], Shreya Bhat[2], Vidya Niranjan[2]

[1]RV College of engineering; [2]R V College of Engineering

Centre of Excellence in ...

Vidya Niranjan
R V College of Engineering

DOI: **https://dx.doi.org/10.17504/protocols.io.q26g713qqgwz/v1**

Protocol Citation: Adarsh V, Shreya Bhat, Vidya Niranjan 2024. In-Silico Validation of Biomarkers using ROC and AUC Curve Analysis in R: A Comprehensive Protocol. **protocols.io** **https://dx.doi.org/10.17504/protocols.io.q26g713qqgwz/v1**

Protocol status: Working
We use this protocol and it's working

Created: July 19, 2024

Last Modified: July 23, 2024

Protocol Integer ID: 103679

Keywords: Biomarker validation, in-Silico validation, ROC Analysis, R programming, AUC analsis, statistical analysis , silico validation of biomarker, identification of novel biomarker, potential biomarker, performance of biomarker, significance of potential biomarker, predicting alzheimer, using roc, biomarker, novel biomarker, protocol biomarker, employing roc, silico validation, unique gene identifier, ensuring unique gene identifier, roc, statistical validation, alzheimer, auc curve analysis, gene expression, complex conditions like alzheimer, differential gene expression, validation, data preprocessing, comprehensive protocol biomarker

Disclaimer

Abstract

Biomarkers are essential for the early detection, diagnosis, and management of diseases, particularly in complex conditions like Alzheimer's disease. This paper presents a comprehensive protocol for the in-silico validation of biomarkers using Receiver Operating Characteristic (ROC) and Area Under the Curve (AUC) analysis in R. The protocol emphasizes the importance of rigorous data preprocessing and statistical validation, utilizing a Universal dataset, GSE36980, which comprises expression data from post-mortem Alzheimer's disease brains.The dataset was subjected to differential gene expression (DGE) analysis, and the significance of potential biomarkers was evaluated using statistical t-tests. The protocol outlines detailed steps for data preprocessing, including handling missing values, ensuring unique gene identifiers, and creating a binary classification variable based on log fold change cutoffs.By employing ROC and AUC curve analysis, this protocol aims to provide researchers and clinicians with a robust framework for assessing the performance of biomarkers in predicting Alzheimer's disease. The findings from this in-silico validation can facilitate the identification of novel biomarkers and enhance decision-making in clinical practice. This comprehensive approach not only streamlines the validation process but also contributes to the growing body of knowledge in biomarker research

Materials

GPU Server
R Studio
Differential gene expression analysis file
Graph Pad Prism Software

Troubleshooting

Before start

Ensure that R and RStudio are updated to the latest versions.

Introduction

1 Biomarkers play a crucial role in the early detection, diagnosis, and management of various diseases. However, the development and validation of reliable biomarkers is a complex and challenging process that requires rigorous evaluation of their analytical and clinical performance. Biomarker validation involves assessing the accuracy, precision, sensitivity, specificity, and reproducibility of the biomarker in a laboratory setting (analytical validation), as well as evaluating its ability to accurately detect or predict the clinical condition of interest in a target population (clinical validation)[1]. In-silico validation, which refers to the computational evaluation of biomarkers using mathematical models, simulations, and data analysis techniques, has become an increasingly important aspect of the biomarker validation process. In-silico validation offers several advantages, including cost-effectiveness, rapid screening of potential biomarkers, hypothesis generation, optimization of assays, and risk assessment. By leveraging computational power and advanced statistical methods, in-silico validation can help identify novel biomarkers, guide the design of subsequent experimental studies, and optimize the performance of biomarker assays. One of the most widely used methods for evaluating the performance of binary classification models, such as those used to predict the presence or absence of a disease based on biomarker levels, is Receiver Operating Characteristic (ROC) curve analysis.[2] ROC curves plot the true positive rate (sensitivity) against the false positive rate (1 - specificity) for different decision thresholds, while the Area Under the ROC Curve (AUC) serves as a summary statistic that represents the overall accuracy of the classification model . ROC and AUC curve analysis provide valuable insights into the performance of biomarkers and help in the selection of optimal cutoff values for clinical decision-making.[3]The objective of this protocol is to provide a comprehensive guide for the in-silico validation of biomarkers using ROC and AUC curve analysis in R, a widely used programming language for statistical computing and graphics. The protocol will cover the necessary steps for data preprocessing, exploratory data analysis, ROC and AUC curve generation, and performance evaluation of biomarkers. By following this protocol, researchers and clinicians can effectively assess the potential of biomarkers for their intended applications and make informed decisions about their clinical utility.

Data Pre-processing

2 The input file for the in-silico validation protocol should be in CSV (Comma-Separated Values) format, obtained from differential gene expression (DGE) analysis. The file must compulsorily contain two columns: "gene_id" and "logFC". The "gene_id" column should contain the unique identifiers for each gene, while the "logFC" column should provide the log fold change values for each gene, indicating the relative expression between two conditions (e.g., disease vs. control).

Installing Library

3

install.packages("readr"): This command installs the readr package, which provides functions to read
rectangular data like CSV files efficiently.

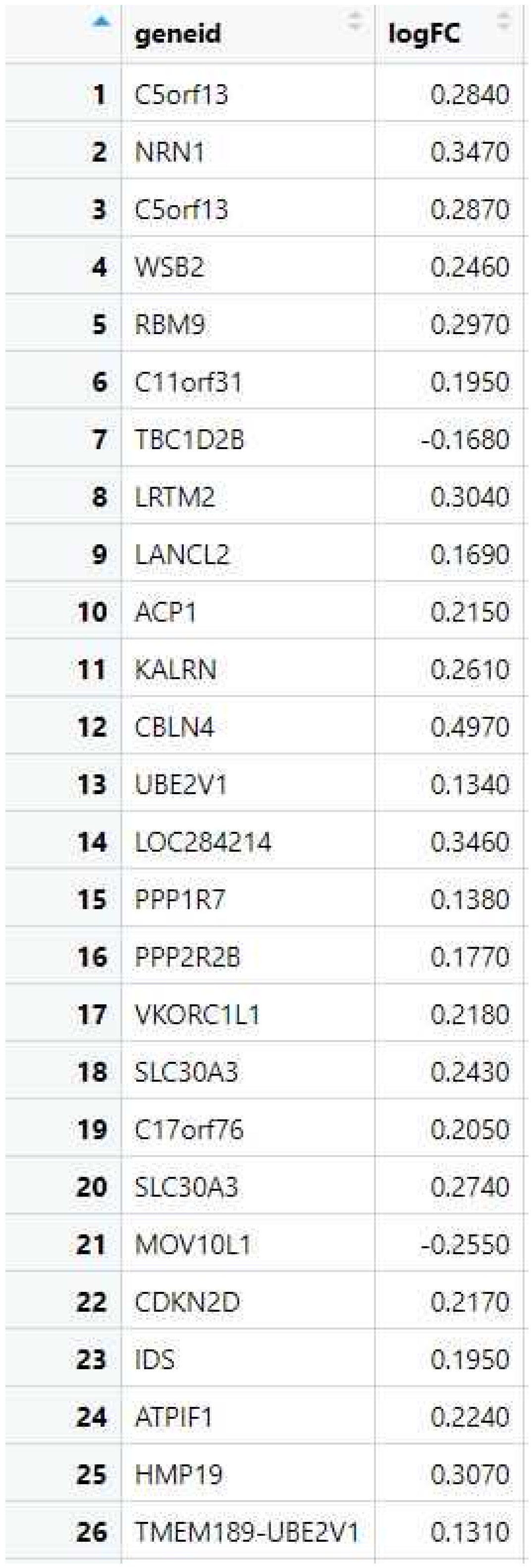
install.packages("pROC"): This command installs the pROC package, which is used for computing and visualizing Receiver Operating Characteristic (ROC) curves and the Area Under the Curve (AUC).

install.packages("dplyr"): This command installs the dplyr package, which offers a set of tools for data manipulation and transformation.

```
> library(readr)
> library(pROC)
> library(dplyr)
```

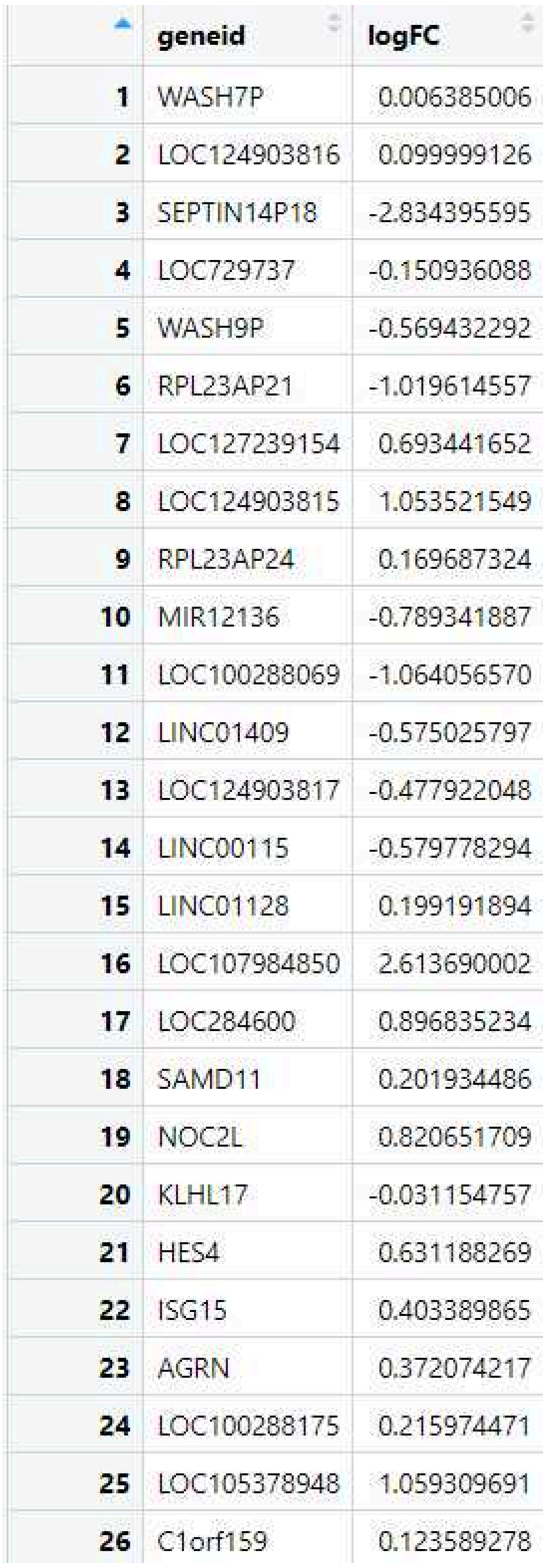
Dataset Preparation

4 Start with two datasets, one being the test data set and one being a universal dataset. Each must contain only **geneid** and **logFC** columns. The first picture depicts Test dataset and second picture shows the universal dataset respectively.



	geneid	logFC
1	C5orf13	0.2840
2	NRN1	0.3470
3	C5orf13	0.2870
4	WSB2	0.2460
5	RBM9	0.2970
6	C11orf31	0.1950
7	TBC1D2B	-0.1680
8	LRTM2	0.3040
9	LANCL2	0.1690
10	ACP1	0.2150
11	KALRN	0.2610
12	CBLN4	0.4970
13	UBE2V1	0.1340
14	LOC284214	0.3460
15	PPP1R7	0.1380
16	PPP2R2B	0.1770
17	VKORC1L1	0.2180
18	SLC30A3	0.2430
19	C17orf76	0.2050
20	SLC30A3	0.2740
21	MOV10L1	-0.2550
22	CDKN2D	0.2170
23	IDS	0.1950
24	ATPIF1	0.2240
25	HMP19	0.3070
26	TMEM189-UBE2V1	0.1310

Dataset of test case



	geneid	logFC
1	WASH7P	0.006385006
2	LOC124903816	0.099999126
3	SEPTIN14P18	-2.834395595
4	LOC729737	-0.150936088
5	WASH9P	-0.569432292
6	RPL23AP21	-1.019614557
7	LOC127239154	0.693441652
8	LOC124903815	1.053521549
9	RPL23AP24	0.169687324
10	MIR12136	-0.789341887
11	LOC100288069	-1.064056570
12	LINC01409	-0.575025797
13	LOC124903817	-0.477922048
14	LINC00115	-0.579778294
15	LINC01128	0.199191894
16	LOC107984850	2.613690002
17	LOC284600	0.896835234
18	SAMD11	0.201934486
19	NOC2L	0.820651709
20	KLHL17	-0.031154757
21	HES4	0.631188269
22	ISG15	0.403389865
23	AGRN	0.372074217
24	LOC100288175	0.215974471
25	LOC105378948	1.059309691
26	C1orf159	0.123589278

Dataset of Universal data

4.1 **Importing dataset ：**

read.csv("dge_ALZ.csv"): Reads the dge_ALZ.csv file into a data frame called dge_ALZ which refers to the test dataset.

read.csv("udata1_ALZ.csv"): Reads the udata1_ALZ.csv file into a data frame called udata1_ALZ which is the universal dataset.

```
> dge_ALZ <- read.csv("dge_ALZ.csv")
> udata1_ALZ <- read.csv("udata1_ALZ.csv")
```

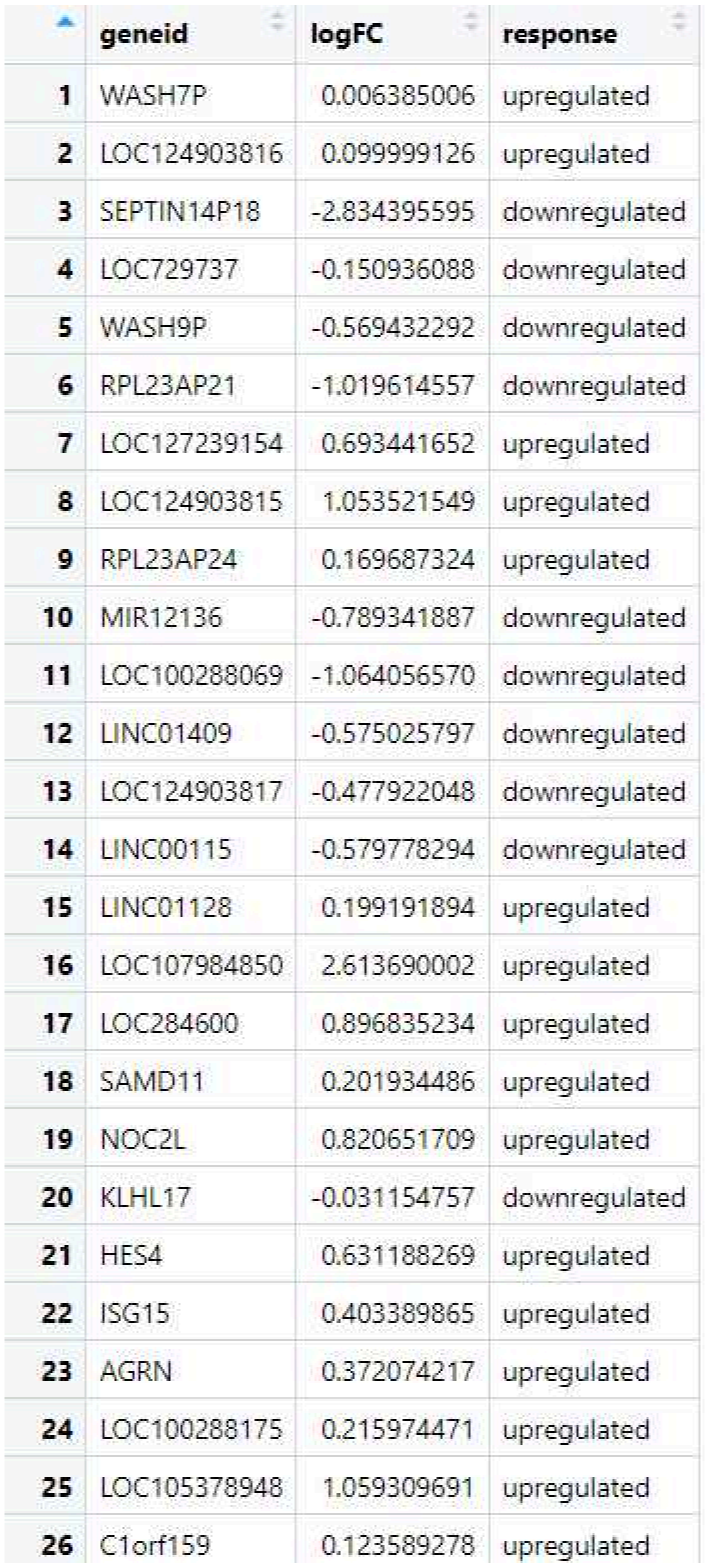
5 **Convert Gene IDs to Character Type:** After reading the datasets, it is important to ensure that the gene id columns are treated as character type. This is because gene ID values, which represent gene identifiers, are often alphanumeric and need to be recognized as such to facilitate accurate data merging and manipulation. By converting the gene ID columns in both dge_ALZ and udata1_ALZ data frames to character type, we prevent any potential issues that might arise from treating these identifiers as numerical values. This step is critical for maintaining the integrity of the data and ensuring that subsequent merging operations work correctly.

```
> dge_ALZ$geneid <- as.character(dge_ALZ$geneid)
> udata1_ALZ$geneid <-as.character(udata1_ALZ$geneid)
```

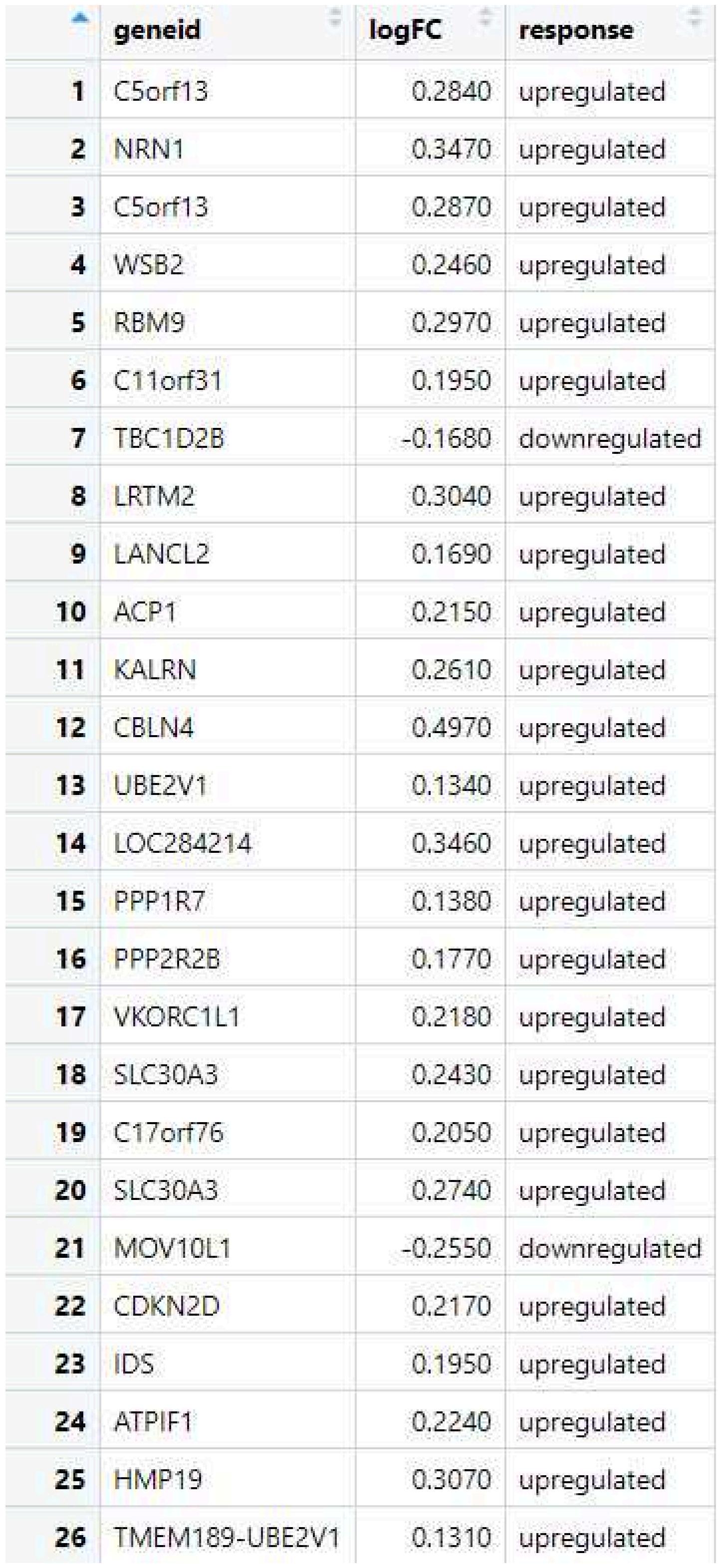
6 **Create a Response Variable Based on logFC :** Next, we create a new column that categorizes each gene based on its logFC value, labeling them as either "upregulated" or "downregulated." This categorization transforms the continuous logFC values into a binary format, which is necessary for defining the response variable required for ROC analysis. By creating this response variable, we set the stage for evaluating the classification performance between the two datasets.

```
  >   dge_ALZ$response <- ifelse(dge_ALZ$logFC> 0, "upregulated",
"downregulated")
  >   udata1_ALZ$response <-ifelse(udata1_ALZ$logFC > 0,
"upregulated","downregulated")
```

6.1 **Creation of response variable based log FC :** The first picture depicts Test dataset and second picture shows the universal dataset respectively.



	geneid	logFC	response
1	WASH7P	0.006385006	upregulated
2	LOC124903816	0.099999126	upregulated
3	SEPTIN14P18	-2.834395595	downregulated
4	LOC729737	-0.150936088	downregulated
5	WASH9P	-0.569432292	downregulated
6	RPL23AP21	-1.019614557	downregulated
7	LOC127239154	0.693441652	upregulated
8	LOC124903815	1.053521549	upregulated
9	RPL23AP24	0.169687324	upregulated
10	MIR12136	-0.789341887	downregulated
11	LOC100288069	-1.064056570	downregulated
12	LINC01409	-0.575025797	downregulated
13	LOC124903817	-0.477922048	downregulated
14	LINC00115	-0.579778294	downregulated
15	LINC01128	0.199191894	upregulated
16	LOC107984850	2.613690002	upregulated
17	LOC284600	0.896835234	upregulated
18	SAMD11	0.201934486	upregulated
19	NOC2L	0.820651709	upregulated
20	KLHL17	-0.031154757	downregulated
21	HES4	0.631188269	upregulated
22	ISG15	0.403389865	upregulated
23	AGRN	0.372074217	upregulated
24	LOC100288175	0.215974471	upregulated
25	LOC105378948	1.059309691	upregulated
26	C1orf159	0.123589278	upregulated

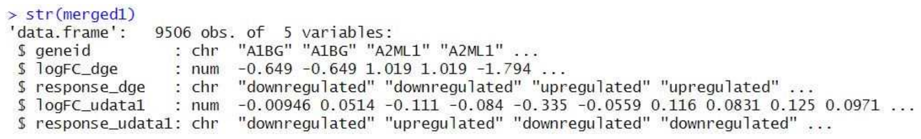


	geneid	logFC	response
1	C5orf13	0.2840	upregulated
2	NRN1	0.3470	upregulated
3	C5orf13	0.2870	upregulated
4	WSB2	0.2460	upregulated
5	RBM9	0.2970	upregulated
6	C11orf31	0.1950	upregulated
7	TBC1D2B	-0.1680	downregulated
8	LRTM2	0.3040	upregulated
9	LANCL2	0.1690	upregulated
10	ACP1	0.2150	upregulated
11	KALRN	0.2610	upregulated
12	CBLN4	0.4970	upregulated
13	UBE2V1	0.1340	upregulated
14	LOC284214	0.3460	upregulated
15	PPP1R7	0.1380	upregulated
16	PPP2R2B	0.1770	upregulated
17	VKORC1L1	0.2180	upregulated
18	SLC30A3	0.2430	upregulated
19	C17orf76	0.2050	upregulated
20	SLC30A3	0.2740	upregulated
21	MOV10L1	-0.2550	downregulated
22	CDKN2D	0.2170	upregulated
23	IDS	0.1950	upregulated
24	ATPIF1	0.2240	upregulated
25	HMP19	0.3070	upregulated
26	TMEM189-UBE2V1	0.1310	upregulated

7 **Merge the Data Frames on geneid :** The next step involves merging the two datasets based on the geneid column. By performing an inner join on geneid, we combine the datasets into a single data frame, ensuring that each row corresponds to a unique gene present in both datasets. We also add suffixes to the column names to distinguish between the logFC values from dge_ALZ and udata1_ALZ. This merged dataset allows us to directly compare the logFC values and response variables across the two sources, providing the necessary structure for subsequent ROC curve analysis.

```
> merged1 <- merge(dge_ALZ, udata1_ALZ, by = "geneid", suffixes =
c("_dge", "_udata1"))
```

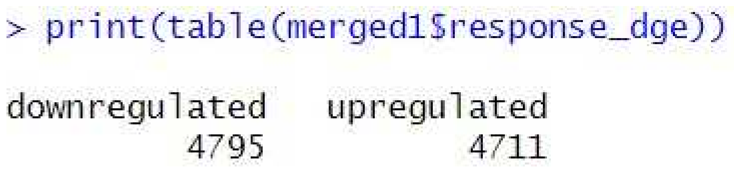
8 **Check the Structure of the Merged Data Frame :** The str(merged1) function in R is used to display the internal structure of the merged1 data frame. This command is particularly useful for understanding the composition and attributes of the data frame after merging two datasets.



```
> str(merged1)
'data.frame':   9506 obs. of  5 variables:
 $ geneid         : chr  "A1BG" "A1BG" "A2ML1" "A2ML1" ...
 $ logFC_dge      : num  -0.649 -0.649 1.019 1.019 -1.794 ...
 $ response_dge   : chr  "downregulated" "downregulated" "upregulated" "upregulated" ...
 $ logFC_udata1   : num  -0.00946 0.0514 -0.111 -0.084 -0.335 -0.0559 0.116 0.0831 0.125 0.0971 ...
 $ response_udata1: chr  "downregulated" "upregulated" "downregulated" "downregulated" ...
```

```
> str(merged1)
```

9 **Print the Frequency Table of response_dge :** This step is performed to get a better understanding of the number of downregulated and upregulated genes respectively.



```
> print(table(merged1$response_dge))

downregulated   upregulated 
         4795          4711 
```

```
print(table(merged1$response_dge))
```

10 **Rename logFC Columns:** Renaming columns in a data frame is an important step in data preparation to ensure clarity and consistency, especially when dealing with merged datasets with overlapping column names. Here, the columns logFC_dge and logFC_udata1 in the merged1 data frame are renamed to Logfc_dge and Logfc_udata1, respectively. This involves identifying the indices of the columns to be renamed using the which function, and then assigning new, standardized names. This step enhances readability and avoids confusion in later analysis stages, ensuring the data frame is easy to understand and work with, thus facilitating accurate data manipulation and interpretation.

```
> colnames(merged1)[which(colnames(merged1)== "logFC_dge")] <-
"Logfc_dge"
> colnames(merged1)[which(colnames(merged1)== "logFC_udata1")] <-
"Logfc_udata1"
```

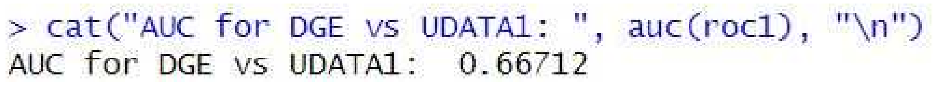
11 **Check for Required Columns :** This conditional statement checks if the columns Logfc_dge and Logfc_udata1 exist in the merged1 data frame. If either column is missing, the stop() function terminates the execution with an error message. This step ensures data integrity before proceeding with the analysis.

```
> if(!("Logfc_dge" %in%colnames(merged1)) | !("Logfc_udata1" %in%
colnames(merged1))) {
+    stop("Logfc columns not found in merged1")
+ }
```

Computing ROC and AUC

12 The roc() function from the pROC package creates a ROCcurve object roc1 using the response_dge as the true class labels and Logfc_udata1 as the predictor values. The levels parameter specifies the order of the response categories, and direction indicates the direction of comparison. The auc() function calculates the Area Under the Curve (AUC) for the ROC curve, which is printed using cat().

```
  >  roc1 <- roc(merged1$response_dge,merged1$Logfc_udata1, levels
= c("downregulated",
"upregulated"), direction = "<")
 >  cat("AUC for DGE vs UDATA1: ", auc(roc1), "\n")
```



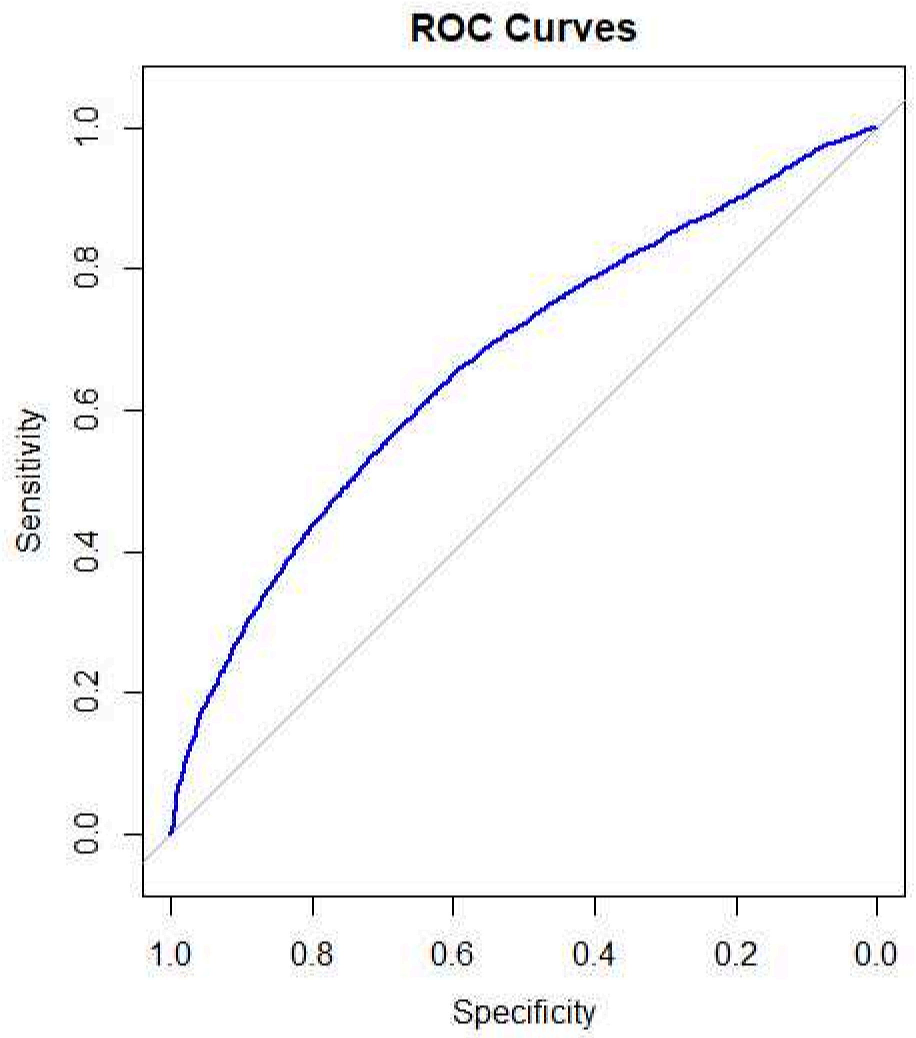
```
> cat("AUC for DGE vs UDATA1: ", auc(roc1), "\n")
AUC for DGE vs UDATA1:  0.66712
```

> AUC Value

Plot the Curve

13 Finally, the plot() function visualizes the ROC curve, with customization for color and line width. The blue
line represents DGE vs udata.

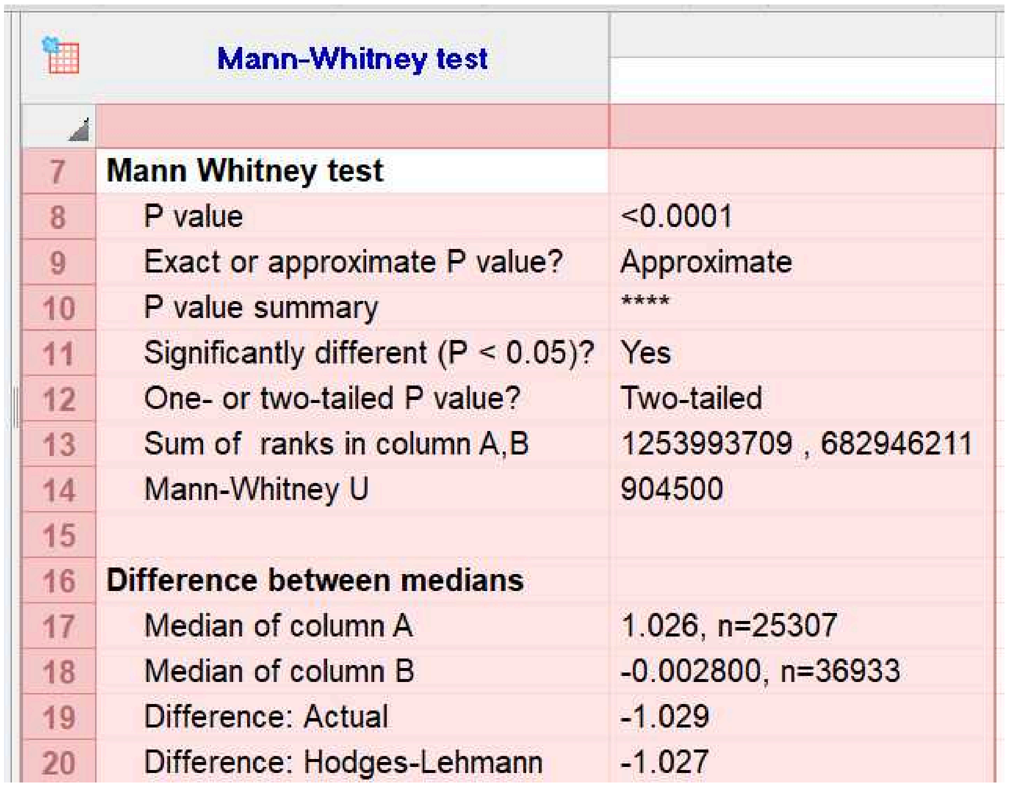
```
>    plot(roc1, col = "blue", main ="ROC Curves", lwd = 2)
```



Result Obtained from ROC analysis

Statistical Analysis

14 To perform the Mann-Whitney test in GraphPad Prism, organize your data into two columns for each group and input it into the data table. Click "Analyze," select "Nonparametric tests," and choose the "Man nWhitney test." Ensure the option to compare medians is selected, then click "OK" to run the analysis. Review the output for the U statistic and p-value; a p-value < 0.05 indicates a significant difference. Finally, report the U statistic, p-value, and median differences.[4,5]



Mann-Whitney test

7	**Mann Whitney test**	
8	P value	<0.0001
9	Exact or approximate P value?	Approximate
10	P value summary	****
11	Significantly different (P < 0.05)?	Yes
12	One- or two-tailed P value?	Two-tailed
13	Sum of ranks in column A,B	1253993709 , 682946211
14	Mann-Whitney U	904500
15		
16	**Difference between medians**	
17	Median of column A	1.026, n=25307
18	Median of column B	-0.002800, n=36933
19	Difference: Actual	-1.029
20	Difference: Hodges-Lehmann	-1.027

The results of the Mann-Whitney test indicate a highly significant difference in biomarker levels between the test samples and the universal dataset. The median biomarker level in the test samples (Column A) is significantly higher than that in the universal dataset (Column B), with a difference of approximately 1.029 units. This finding suggests that the biomarker may be a useful indicator for distinguishing between the two groups, potentially supporting its role in diagnostic or prognostic applications related to the condition being studied. These results provide strong evidence for the validity of the biomarker in the context of your research, indicating that it may be effective in differentiating between the populations represented by the two groups.

Protocol references

1. Rai, A. J. (Ed.). (2019). *Biomarker validation: Principles and practice*. Springer.
2. Tarca, A. L., Romero, R., & Draghici, S. (2006). Analysis of microarray experiments of gene expression profiling. *American Journal of Obstetrics and Gynecology, 195*(2), 373-388. https://doi.org/10.1016/j.ajog.2006.03.021
3. Hajian-Tilaki, K. (2013). Receiver operating characteristic (ROC) curve analysis for medical diagnostic test evaluation. *Caspian Journal of Internal Medicine, 4*(2), 627-635.
4. Google Developers. (2022). Classification: ROC curve and AUC. Retrieved from https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
5. GraphPad Software. (n.d.). *GraphPad Prism 10 statistics guide - Choosing a test*. Retrieved from https://www.graphpad.com/guides/prism/latest/statistics/stat_qa_choosing_a_test_to_compare_.htm