# Integrative Analysis of Sarcoma Biomarkers Based on Age Classification through TCGAbiolinks and Machine Learning Approaches

## Project Objective

In this project, a gene expression dataset for sarcoma from The Cancer Genome Atlas (TCGA) was pre-processed and analyzed downstream to identify potential cancer biomarkers based on age classification using differential expression and machine learning models. The biomarker team performed differential expression analysis and functional enrichment analysis, while the machine learning team conducted feature selection and random forest classification. The project involved data preprocessing, visualization, and interpretation of the identified biomarkers and model performance.

## Biomarker Discovery

**Requirements**

The following R libraries were used

1. TCGAbiolinks
2. SummarizedExperiment
3. data.table
4. dplyr
5. ggplot2
6. gplots
7. biomaRt

**1.** **Data collection and preprocessing**

* The sarcoma dataset ("**TCGA-SARC**" project) was obtained from The Cancer Genome Atlas (TCGA) using the **TCGAbiolinks** package in R.
* A query was prepared to retrieve "**Gene Expression Quantification**" data from the "**Transcriptome Profiling**" data category of the "**TCGA-SARC**" project, focusing on the sample types **"Metastatic"**, **"Primary Tumor"**, and **"Recurrent Tumor"**.
* Data was obtained for the "**age\_at\_diagnosis**" sub-group (metadata).
* Samples with missing or inaccurate age data were excluded.
* The values of the "**age\_at\_diagnosis**" subgroup were converted from days to years, and the samples were divided into two age groups based on the threshold of 40 years (14,610 days): **18-40 years** and **≥40 years.**
* The unstranded dataset was selected andtwenty samples were selected at random from each age group for analysis.
* The **TCGAanalyze\_Normalization()** function was used to normalize the gene expression data by gene length, and quantile filtering was used to eliminate low-expression genes. The cutoff was set at the first quantile (0.25).
* The final filtered data was used for downstream differential expression analysis and ML analysis.

**2.** **Differential expression analysis**

* The **TCGAanalyze\_DEA()** function was used to compare gene expression levels between the age groups ≥40 and 18-40 using the **edgeR** pipeline. Genes were categorized as upregulated, downregulated, or not significant based on a log fold change threshold of >4 or <-4 and a false discovery rate (FDR) < 0.005.
* Avolcano plotwas generated using the **ggplot()** function andaheatmapwas generated using the **heatmap.2()** function in R to visualize the differentially expressed genes.

**3.** **Functional enrichment analysis**

* The **biomaRt package** was used to convert Ensembl IDs to HGNC symbolsfor genes with significant upregulation or downregulation.
* The **TCGAanalyze\_EAcomplete()** function was used to conduct functional enrichment analysis for the genes with significant upregulation or downregulation.
* The results of enrichment analysis were presented as bar plots using the **TCGAvisualize\_EAbarplot()** functionto highlight the most enriched terms for biological processes, cellular components, molecular functions and pathways.

## Machine Learning

**Requirements**

The following R libraries were used

1. caret
2. DALEX
3. pROC
4. randomForest
5. dplyr
6. ggplot2
7. reshape2

These can be installed by running:

install.packages(c(“caret”,”DALEX”,”pROC”, “randomForest”, “dplyr”, “ggplot2”, “reshape2”))

**4.** **Data preparation for ML**

The expression dataset was transposed to get a dataframe with samples as rows and genes as columns. The expression dataset and metadata were merged based on the sample barcodes using the **merge()** function.

The expression values in the merged data were numericized using **mutate()** and **as.numeric()** functions. The age\_group column in the data was changed as a factor using the **as.factor()** function. The **preProcess()** function was used to scale and center the data.

**5.** **Feature selection**

The feature selection was done by selecting the top 1000 variable features (genes) out of 30,000+ features. SD values of the gene expression values were found, and they were sorted in a decreasing manner using the **sort()** function from which the top 1000 genes/features were filtered out.

**6.** **Random forest classification**

The data was split into train-set and test-set with 70% and 30% data respectively using **createDataPartition()**. Using **randomForest()** function, a RF model was built based on the train set and predictions were made. A confusion matrix for the test data was built using **confusionMatrix()** function. Metrics like precision, recall and F1 score were computed to determine the classification accuracy. Functions like **posPredValue()** and **sensitivity()** were used to compute precision and recall, while F1 score was determined by dividing the product of 2\* precision and recall by sum of precision and recall.

The important features from the RF model were filtered out using **randomForest::importance(rf\_model)**. These features were then ordered using **order()** in a decreasing manner based on the **MeanDecreaseGini** values. The top 20 were selected from that list and plotted for their importance using the **ggplot()** function.

**ggplot()** was also used to plot a violin overlay box plot to visualize the expression of the top 20 genes across the two age groups. For better visualization, log of expression values were used in the y-axis by using **log()** function.

The function **roc()** was used to return an ROC object which tells about the model performance, and this was plotted for visualization using **plot func()**.