Meta Analysis and MSS/MSI Classification on Colorectal Cancer Tumor Specimens

Ali Emre Nebiler

Department of Computer Engineering

Senior Project

Advisor: Res. Asst. Sultan Sevgi Turgut

Introduction

Cancer is a disease that is seen in millions of people. Early diagnosis is very important for such disease.

There are more than 200 types of cancer, the third most common type is colorectal cancer.

In this project, it was desired to develop a colorectal cancer classification software which is cheaper, based on machine learning and classification methods.

Steps

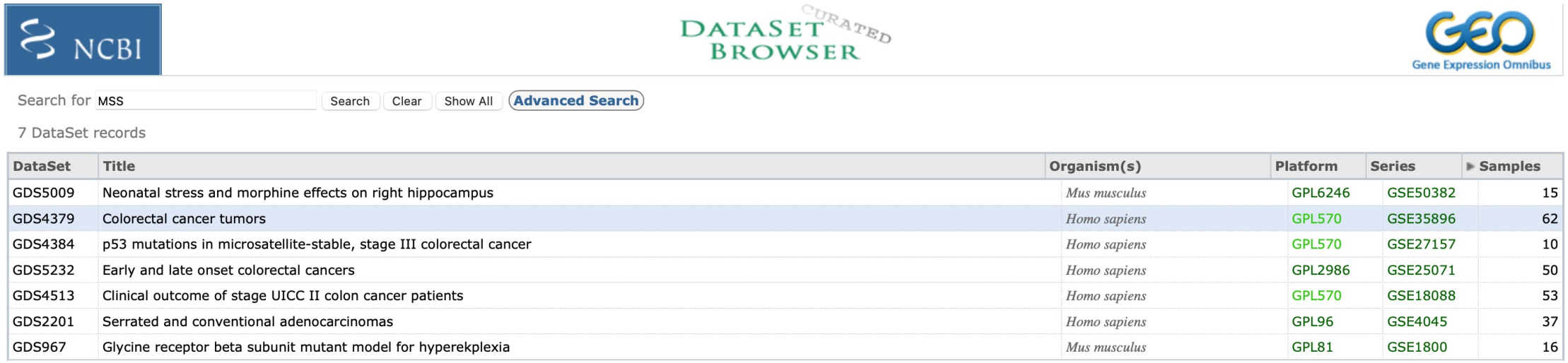
1. Research & Find Data
2. Develop the Code
    a. Prepocessing
    b. Training & Testing
    c. Survival Analysis
3. Gene-Disease Research

1. Research & Find Data

Data Source

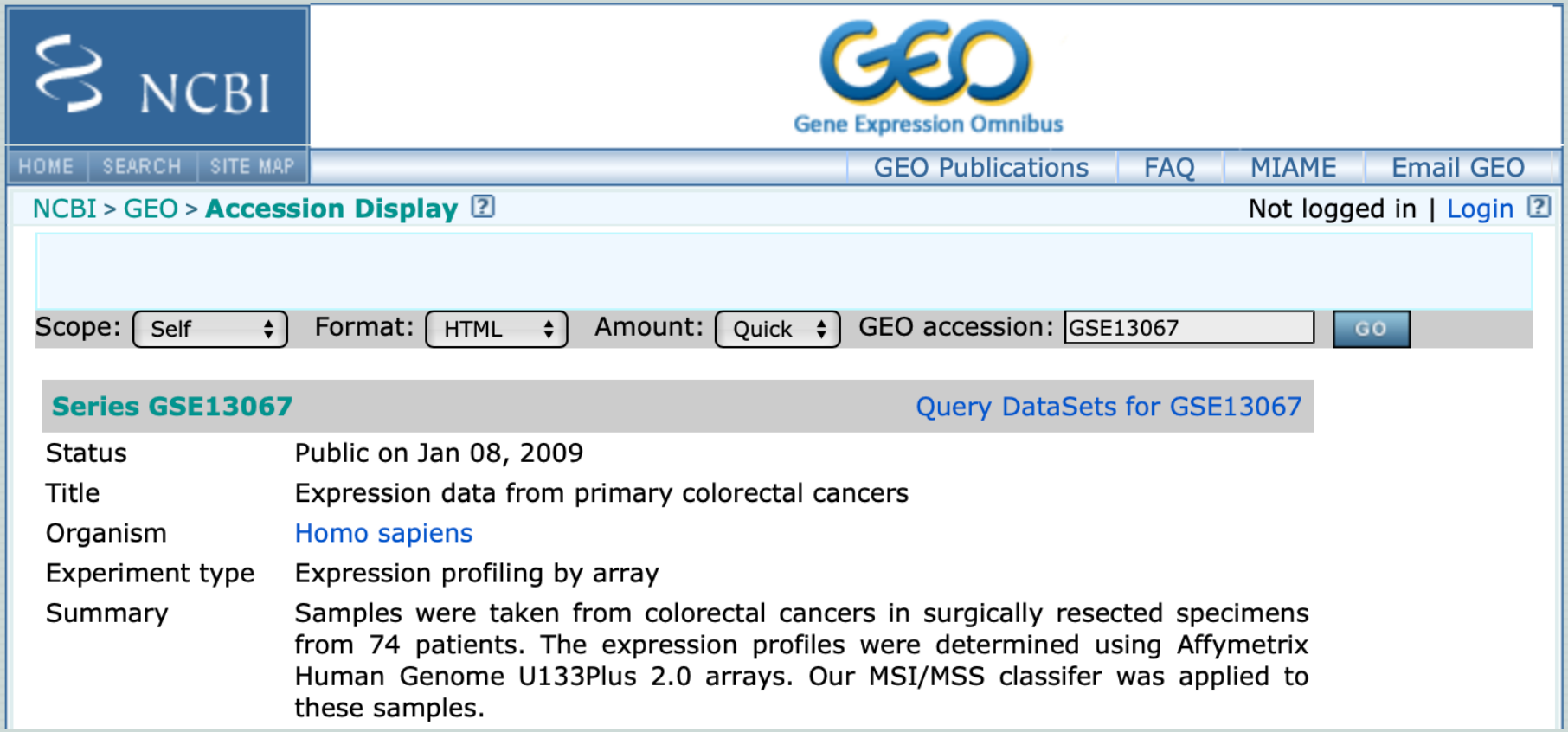
NCBI GEO Dataset Browser

https://www.ncbi.nlm.nih.gov/sites/GDSbrowser



DataSet	Title	Organism(s)	Platform	Series	Samples
GDS5009	Neonatal stress and morphine effects on right hippocampus	*Mus musculus*	GPL6246	GSE50382	15
GDS4379	Colorectal cancer tumors	*Homo sapiens*	GPL570	GSE35896	62
GDS4384	p53 mutations in microsatellite-stable, stage III colorectal cancer	*Homo sapiens*	GPL570	GSE27157	10
GDS5232	Early and late onset colorectal cancers	*Homo sapiens*	GPL2986	GSE25071	50
GDS4513	Clinical outcome of stage UICC II colon cancer patients	*Homo sapiens*	GPL570	GSE18088	53
GDS2201	Serrated and conventional adenocarcinomas	*Homo sapiens*	GPL96	GSE4045	37
GDS967	Glycine receptor beta subunit mutant model for hyperekplexia	*Mus musculus*	GPL81	GSE1800	16

Datasets


NCBI
Gene Expression Omnibus
HOME
SEARCH
SITE MAP
GEO Publications
FAQ
MIAME
Email GEO
NCBI > GEO > Accession Display
Not logged in | Login
Scope: Self
Format: HTML
Amount: Quick
GEO accession: GSE13067
GO
Series GSE13067
Query DataSets for GSE13067
Status
Public on Jan 08, 2009
Title
Expression data from primary colorectal cancers
Organism
Homo sapiens
Experiment type
Expression profiling by array
Summary
Samples were taken from colorectal cancers in surgically resected specimens from 74 patients. The expression profiles were determined using Affymetrix Human Genome U133Plus 2.0 arrays. Our MSI/MSS classifer was applied to these samples.

Datasets

Platforms (1)	GPL570 [HG-U133_Plus_2] Affymetrix Human Genome U133 Plus 2.0 Array
Samples (74)	GSM327282 00-TB-002: MSI
More...	GSM327283 00-TB-005: MSS
	GSM327284 01-TB-006: MSS

Data table header descriptions

ID_REF

VALUE quantile normalized and log-2 transformed

Data table

ID_REF	VALUE
1007_s_at	8.860321274
1053_at	7.242220904
117_at	4.843266364
121_at	7.373590426

Gene Expression:

The process by which the information encoded in a gene is turned into a function [1].

This mostly occurs via the transcription of RNA molecules that code for proteins or non-coding RNA molecules that serve other functions.

[1] ”Gene Expression”, National Human Genome Research Institute, https://www.genome.gov/genetics-glossary/Gene-Expression

MSS/MSI:

Microsatellite Stable / Instable

Shows the stability of the tumor.

MSS: Stable,

MSI-L: Instable, the level is low (accepted as stable for this project)

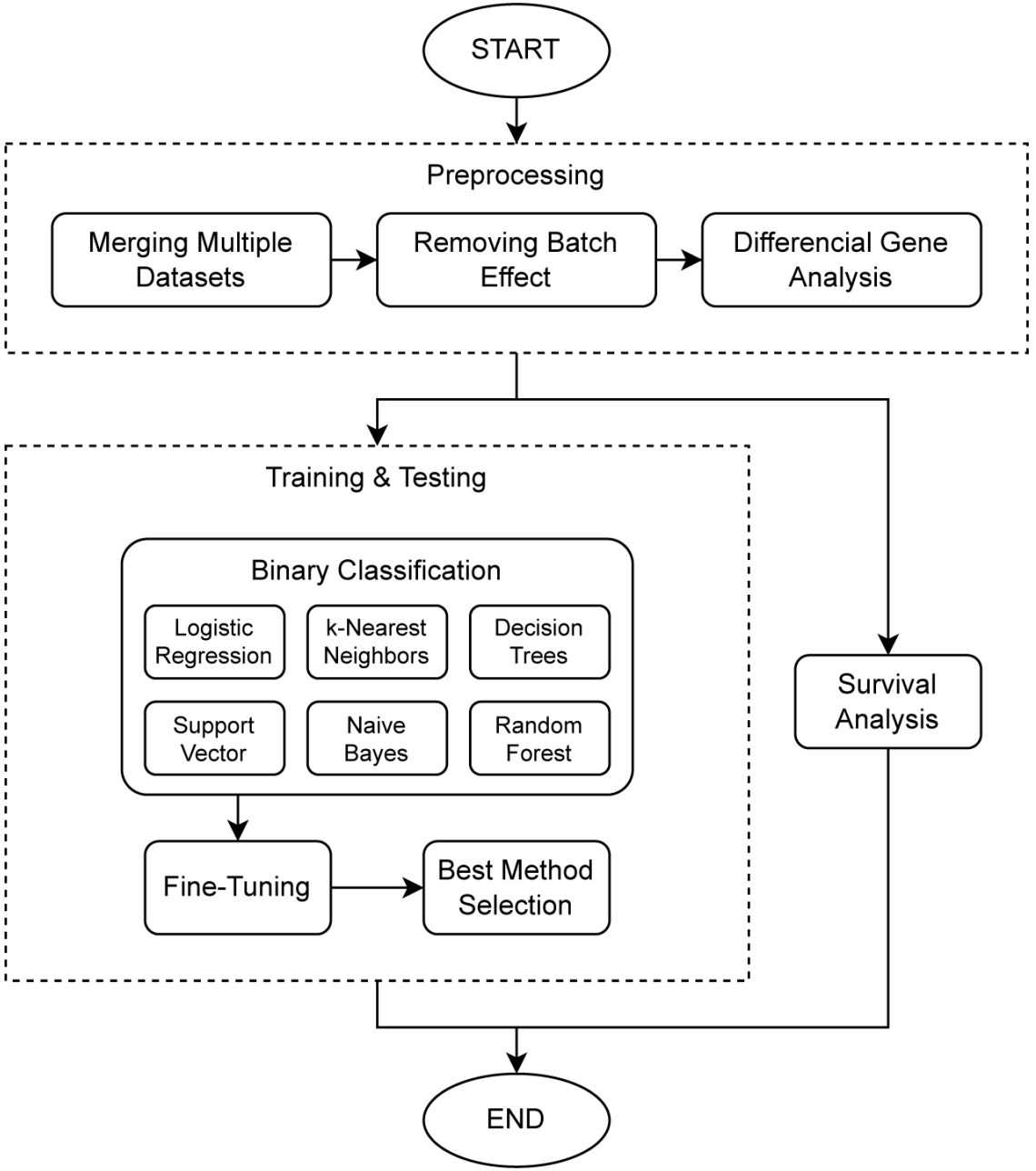
MSI-H: Instable, the level is high

pMMR: Proficient mismatch repair (MSS)

dMMR: Deficient mismatch repair (MSI-H)

2. Develop the Code

Work-Flow


START
Preprocessing
Merging Multiple Datasets
Removing Batch Effect
Differencial Gene Analysis
Training & Testing
Binary Classification
Logistic Regression
k-Nearest Neighbors
Decision Trees
Support Vector
Naive Bayes
Random Forest
Fine-Tuning
Best Method Selection
Survival Analysis
END

2.a. Preprocessing

2.a. Configurations

```r
# Set GEO data set names
gset_names <- list(
  "GSE13067",
  "GSE13294",
  "GSE18088",
  "GSE26682",
  "GSE35896",
  "GSE39084",
  "GSE39582",
  "GSE75316",
  "GSE35566" # This will be the test dataset
)

# Set column names which can include MSS values
mss_colnames <- list(
  "characteristics_ch1",
  "characteristics_ch1",
  "microsatellite status:ch1",
  "microsatellite instability (msi) status:ch1",
  "microsatellite.status:ch1",
  "group:ch1",
  "mmr.status:ch1",
  "microsatellite status:ch1",
  "mss/msi status:ch1"
)
```

2.a. Contents of Datasets

	Dataset Name	Sample Amount	Gene Amount
1	GSE13067	74	54675
2	GSE13294	155	54675
3	GSE18088	53	54675
4	GSE26682	160	19473
5	GSE35896	61	54675
6	GSE39084	70	54675
7	GSE39582	536	54675
8	GSE75316	59	54675
9	GSE35566	19	54675

Total Samples: 1187
Total Genes: 12809

2.a. Merging Datasets Processes

```r
# Delete sample names which has unknown value
for (sample_name in gset_sample_names) {
  if (!grepl("MSI|MSS|dMMR|pMMR", gset_mdata[sample_name, mss_colname], ignore.case = TRUE)) {
    gset_sample_names <- gset_sample_names[gset_sample_names != sample_name]
  }
}
```

```r
# Set MSS status
if (grepl("MSI-L", gset_mdata[sample_name, mss_colname], ignore.case = TRUE)) {
  gset_merged_mdata[sample_name, "Status"] <- "MSI-L"
} else if (grepl("MSI|dMMR", gset_mdata[sample_name, mss_colname], ignore.case = TRUE)) {
  gset_merged_mdata[sample_name, "Status"] <- "MSI-H"
} else if (grepl("MSS|pMMR", gset_mdata[sample_name, mss_colname], ignore.case = TRUE)) {
  gset_merged_mdata[sample_name, "Status"] <- "MSS"
} else {
  stop('MSS status of sample "', sample_name, '" is unknown.')
}
```

2.a. Merging Datasets Outputs

```
...
Getting the metadata values of sample #1185 out of 1187: GSM870737
Getting the metadata values of sample #1186 out of 1187: GSM870738
Getting the metadata values of sample #1187 out of 1187: GSM870739
Got all metadata values.

...
Getting the expression values of sample #1185 out of 1187: GSM870737
Getting the expression values of sample #1186 out of 1187: GSM870738
Getting the expression values of sample #1187 out of 1187: GSM870739
Got all expression values.
```

2.a. Merged Dataset

	Dataset	Status
GSM327282	GSE13067	MSI-H
GSM327283	GSE13067	MSS
GSM327284	GSE13067	MSS
GSM327285	GSE13067	MSS
GSM327286	GSE13067	MSS
GSM327287	GSE13067	MSS
GSM327288	GSE13067	MSI-H
GSM327289	GSE13067	MSS

	INTS3	SBNO2	SLC27A3	TRAPPC12	DLG
GSM327282	6.949213533	2.6309975105	6.477683317	6.084686157	5.00
GSM327283	6.805719912	4.662356874	6.341688152	6.736915504	5.65
GSM327284	6.920400173	2.8357554815	6.372942428	5.21333475	6.28
GSM327285	7.392657445	3.719267132	5.6295189065	5.955208112	5.81
GSM327286	6.592754643	4.834758062	6.579106106	5.978106201	5.26
GSM327287	7.32644986	5.5553274205	6.7972747255	5.730106395	5.89
GSM327288	6.509280485	4.7726442415	6.711350831	5.797783632	4.98
GSM327289	6.609537997	4.822522375	6.2396119175	5.635043605	5.87

2.a. Removing Batch-Effect

```
message('Removing batch effect...')

# Reverse column and row names, create matrix
exprs_table_matrix <- as.matrix.data.frame(t(exprs_table))
mode(exprs_table_matrix)='numeric'

# Remove batch effect
exprs_table_corrected <- removeBatchEffect(exprs_table_matrix, mdata_table[,"Dataset"])

# Turn matrix into data frame
exprs_table_corrected <- as.data.frame(t(exprs_table_corrected))
colnames(exprs_table_corrected) <- colnames(exprs_table)
rownames(exprs_table_corrected) <- rownames(exprs_table)
```

2.a. Removing Batch-Effect

2.a. Feature Selection

Differential Gene Expression (DGE) Analysis:

Finds upregulated and downregulated genes.

(Classified upregulated if increases the risk of MSI, else downregulated.)

LogFC (Fold Change):

A measure describing how much a quantity changes between an original and a subsequent measurement.

2.a. Differential Gene Expression

```r
# Create expression set
exprs_set <- ExpressionSet(assayData = exprs_mat, phenoData = AnnotatedDataFrame(pdata))
table(pData(exprs_set)[, "Label"])

# Create the design matrix
design <- model.matrix(~0 + Label, data = pData(exprs_set))
head(design, 4)
colSums(design)

# Create the contrast matrix
cm <- makeContrasts(
  MSIvMSS = LabelMSI - LabelMSS,
  levels = design
)
```

2.a. Differential Gene Expression

```r
# Fit coefficients
fit <- lmFit(exprs_set, design)

# Fit contrasts
fit2 <- contrasts.fit(fit, contrasts = cm)

# Calculate t-statistics
fit2 <- eBayes(fit2)

# Summarize results
results <- decideTests(fit2) # Default: p.value = 0.05, lfc = 0
summary(results)
```

2.a. DGE Outputs

```
Reading CSV files...
Sample names are matching.
DGE analysis in progress...
Saving as CSV...
Total Genes (Before): 12809
Total Genes (After): 8402 | Up: 5154 | Down: 3248
Completed.
```

2.a. Up/Down Gene Matrix

logFC	AveExpr	t	P.Value	adj.P.Val	B	gene	type
2.06897474915583	6.1357218316355	16.7304747403055	1.37070731333458e-56	1.64293616114568e-54	117.879982596608	HOXC6	up
1.79613099155455	5.80029897120898	13.8229626235527	2.05807340763715e-40	7.16710158308235e-39	80.906113880017	CXCL13	up
1.62764177845087	10.1330206237236	16.6783091358639	2.78339629007652e-56	3.18791655090098e-54	117.176435701108	EIF5AP4	up
1.61372619474536	7.77468106371293	7.33547956952629	4.07685222541001e-13	1.3140773214361e-12	18.7962852647281	REG1A	up
1.48380514526371	5.35954265273518	18.5635939786813	9.36642069856071e-68	1.78794564001414e-65	143.421925777744	GNLY	up
-2.61303377235687	7.06588511464512	-16.2887072830032	5.28804474251775e-54	5.20471797687808e-52	111.965419655336	KRT23	down
-2.34938620808741	7.15493216142209	-16.5176177293072	2.44614622762855e-55	3.05580113359135e-53	115.017807165041	LY6G6F.LY6G6D	down
-2.34938620808741	7.15493216142209	-16.5176177293072	2.44614622762855e-55	3.05580113359135e-53	115.017807165041	LY6G6D	down
-2.10682677553756	8.65624899598278	-15.6577111543299	2.19465107778461e-50	1.51664397886051e-48	103.693116142105	CXCL14	down
-2.02679443468854	8.51550304580614	-17.8561117882579	2.29028279334585e-63	4.37578736046313e-61	133.381774571378	TDGF1	down

2.b. Training & Testing

2.b. Configurations

```
Classifier(
    name="Naive Bayes",
    classifier=GaussianNB(),
    param_grid={
        "var_smoothing": np.logspace(0, -4, 100),
    },
),
Classifier(
    name="Support Vector",
    classifier=SVC(probability=True),
    param_grid={
        "C": [0.1, 1, 10],
    },
),
Classifier(
    name="Decision Tree",
    classifier=DecisionTreeClassifier(),
    param_grid={
        "min_samples_split": [2, 3, 4],
    },
),
```

```
Classifier(
    name="Random Forest",
    classifier=RandomForestClassifier(),
    param_grid={
        "n_estimators": [115, 130],
        "min_samples_split": [9, 10],
    },
),
Classifier(
    name="k-Nearest Neighbor",
    classifier=KNeighborsClassifier(),
    param_grid={
        "n_neighbors": list(range(1, 31)),
        "weights": ["uniform", "distance"],
    },
),
Classifier(
    name="Logistic Regression",
    classifier=LogisticRegression(),
    param_grid={
        "C": [0.01, 0.1, 1, 10, 100],
        "max_iter": [1000],
        "penalty": ["l2"],
    },
),
```

2.b. Grid Search

```
def find_best_params_with_grid_search(clf, param_grid, scoring, cv, features, labels):
    # Set grid search classifier
    gs_classifier = GridSearchCV(clf, param_grid=param_grid, scoring=scoring, cv=cv)

    # Train the model
    gs_classifier.fit(features, labels)

    # Get best parameters
    return gs_classifier.best_params_
```

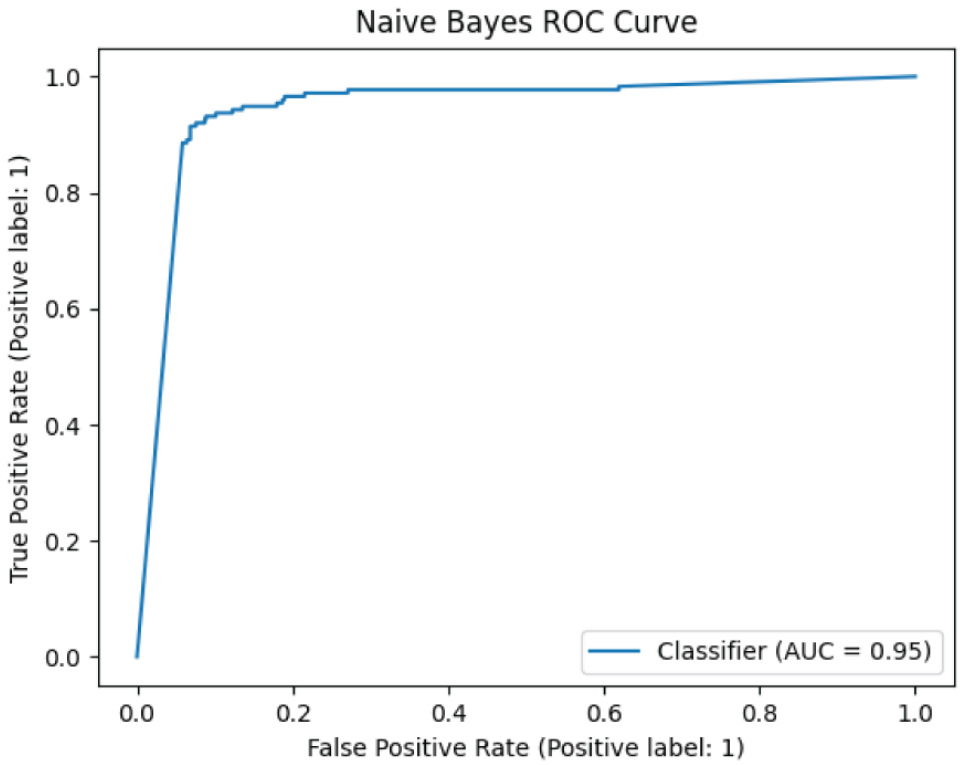
2.b. Train & Test

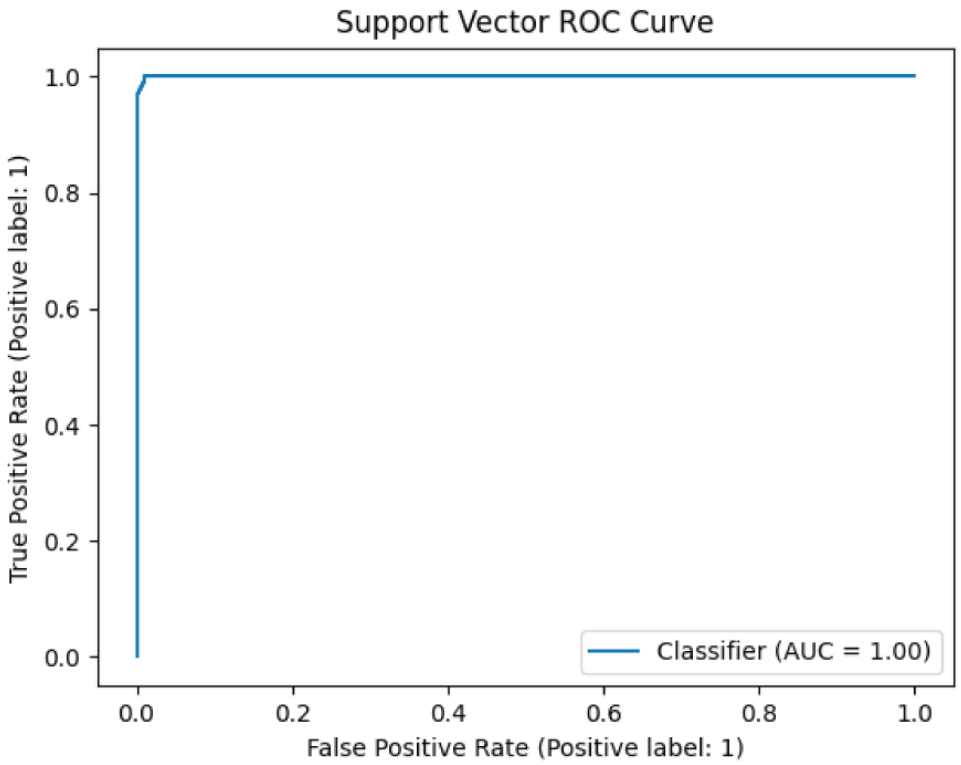
```
# Train the model
CLASSIFIERS[i].model = train_with_classifier(
    clf=CLASSIFIERS[i].classifier,
    params=CLASSIFIERS[i].best_params,
    features=features_internal_train,
    labels=labels_internal_train,
)

# Test with internal data
CLASSIFIERS[i].int_score = test_the_model(
    model=CLASSIFIERS[i].model,
    features=features_internal_test,
    labels=labels_internal_test,
)

# Test with external data
CLASSIFIERS[i].ext_score = test_the_model(
    model=CLASSIFIERS[i].model,
    features=features_external_test,
    labels=labels_external_test,
)
```

2.b. Confusion Matrix & ROC Curve

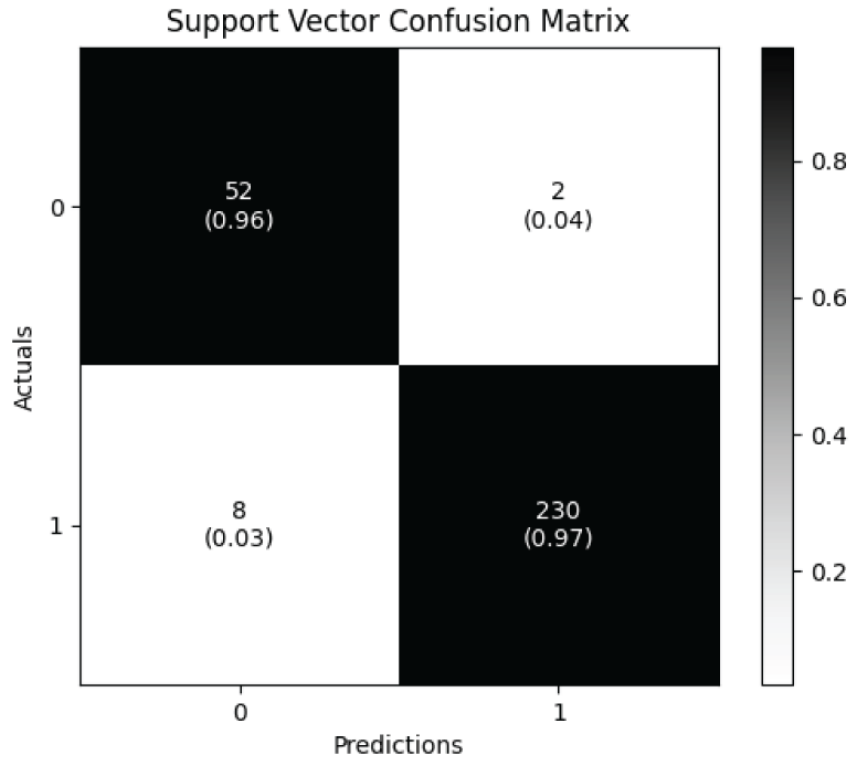
```
# Save confusion matrix as PNG
conf_mat_file_path = os.path.join(
    CONF_MAT_FOLDER_PATH,
    "conf_mat.png",
)
save_confusion_matrix_plot(
    clf_class=CLASSIFIERS[i],
    saved_file_path=conf_mat_file_path,
)

# Save ROC curve as PNG
roc_curve_file_path = os.path.join(
    ROC_CURVE_FOLDER_PATH,
    "roc_curve.png",
)
CLASSIFIERS[i].roc_curve_auc = save_roc_curve_plot(
    clf_class=CLASSIFIERS[i],
    features=features_internal_train,
    labels=labels_internal_train,
    saved_file_path=roc_curve_file_path,
)
```

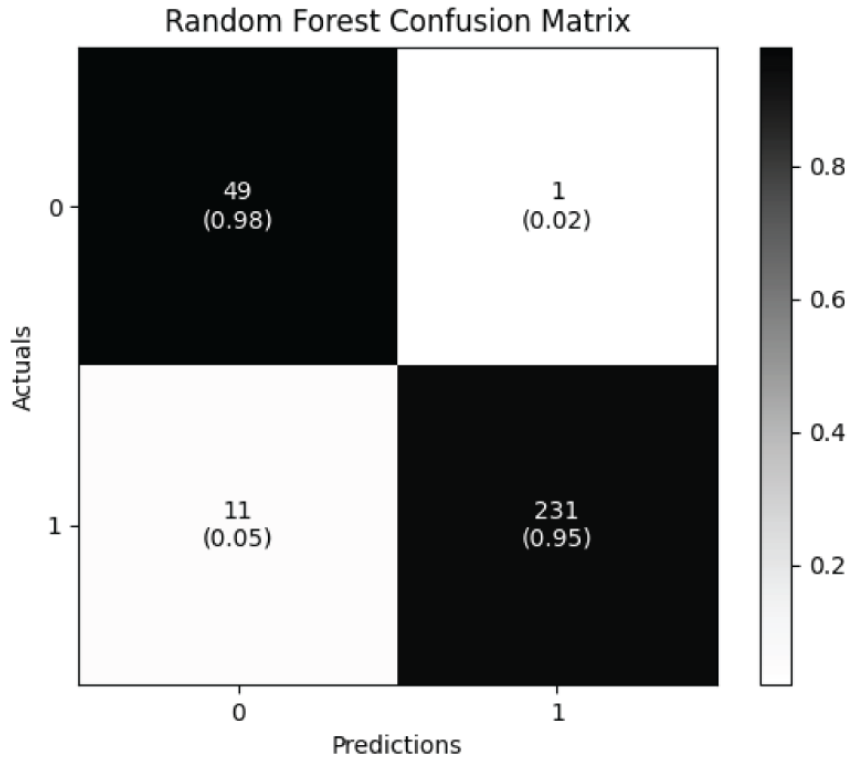
2.b. ROC Curves (Worst)





2.b. Confusion Matrices (Best)





2.b. Calculation of Performance Metrics

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 Score = (2 * Precision * Recall) / (Precision + Recall)

Accuracy = (TP + TN) / (TP + TN + FP + FN)

2.b. Model Testing Results

Classifier	Int. Score	Ext. Score	Best Parameters
Naive Bayes	92.465753%	73.684211%	{'var_smoothing': 0.0422924287438949 66}
Support Vector	96.575342%	78.947368%	{'C': 10}
Decision Tree	92.123288%	84.210526%	{'min_samples_split': 4}
Random Forest	95.890411%	52.631579%	{'min_samples_split': 9, 'n_estimators': 130}
k-Nearest Neighbor	91.438356%	78.947368%	{'n_neighbors': 9, 'weights': 'distance'}
Logistic Regression	94.863014%	68.421053%	{'C': 100, 'max_iter': 1000, 'penalty': 'l2'}

Classifier	Accuracy	Precision	Recall	F1 Score	AUC
Naive Bayes	0.92465753	0.77941176	0.88333333	0.828125	0.94530263
Support Vector	0.96575342	0.96296296	0.86666667	0.9122807	0.99975545
Decision Tree	0.92123288	0.7761194	0.86666667	0.81889764	0.99999185
Random Forest	0.95890411	0.98	0.81666667	0.89090909	1.0
k-Nearest Neighbor	0.91438356	0.79661017	0.78333333	0.78991597	1.0
Logistic Regression	0.94863014	0.9245283	0.81666667	0.86725664	1.0

2.c. Survival Analysis

2.c. Selected Up/Down Genes

	logFC	type	MedianExpr
HOXC6	2.06897474915583	up	5.27100434250863
CXCL13	1.79613099155455	up	5.38591767300377
EIF5AP4	1.62764177845087	up	10.50209095633
REG1A	1.61372619474536	up	7.36807289195732
GNLY	1.48380514526371	up	5.06452151931355
PLA2G2A	1.45901924548829	up	9.14399291013303
DUSP4	1.43564247677064	up	6.62873504382264
EIF5AL1	1.42828431399249	up	10.0111825078244
ADGRG6	1.40976295095422	up	6.74385251417528
CXCL9	1.40037680761574	up	7.27083935734323

	logFC	type	MedianExpr
KRT23	-2.61303377235687	down	7.4252092972981
LY6G6F.LY6G6D	-2.34938620808741	down	7.34920734885119
LY6G6D	-2.34938620808741	down	7.34920734885119
CXCL14	-2.10682677553756	down	9.13186027401262
TDGF1	-2.02679443468854	down	8.96346971629868
TDGF1P3	-2.02679443468854	down	8.96346971629868
FABP1	-1.82610385860403	down	10.3357136687554
RUBCNL	-1.80648290782754	down	8.82105917275978
SLC26A3	-1.782865158537	down	8.36896387878145
QPRT	-1.7823666789576	down	7.76292514178407

2.c. Gather Overall Survival (OS) Data

```r
# Get time and event (censor) data
time <- as.numeric(survival_mdata_table[,"OS.Time"])
censor <- as.numeric(survival_mdata_table[,"OS.Event"])

# Run survival analysis for every gene
for(gene_name in selected_up_and_down_genes) {
  message('Survival Analysis for "', gene_name, '" gene...')

  gene_exprs_values <- survival_exprs_table[,gene_name]
  surv_data <- cbind.data.frame(time, censor, gene_exprs_values)
```

2.c. Necessary Survival Data

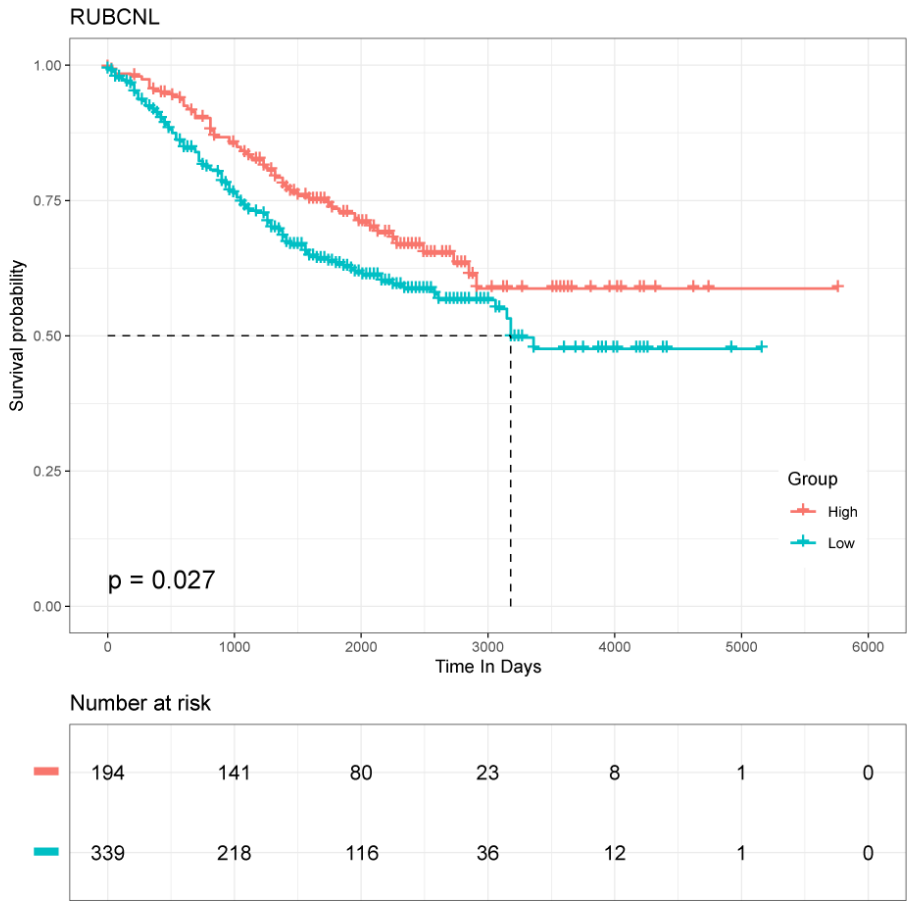
	HOXC6	CXCL13	EIF5AP4	REG1A	GNLY
GSM971957	Low	Low	Low	Low	High
GSM971958	High	High	High	Low	High
GSM971959	High	Low	High	Low	High
GSM971960	Low	High	High	Low	Low
GSM971961	High	Low	High	Low	Low

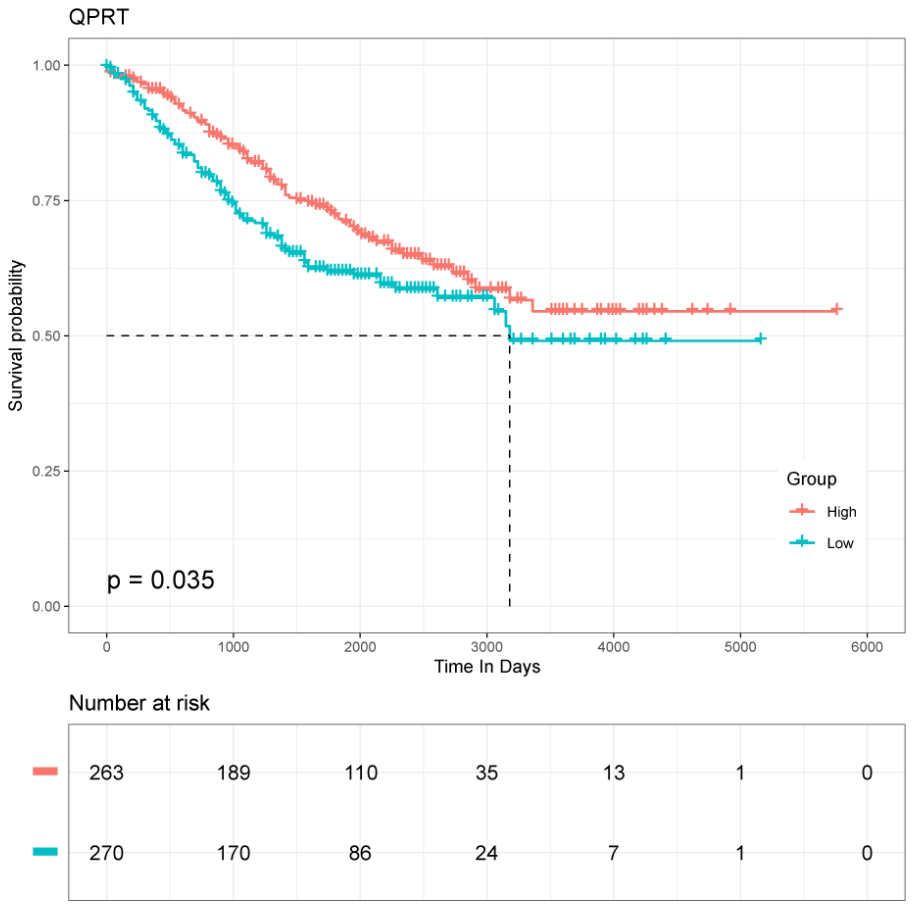
	Dataset	Status	OS.Time	OS.Event
GSM971957	GSE39582	MSS	300	1
GSM971958	GSE39582	MSS	270	1
GSM971959	GSE39582	MSS	1560	0
GSM971960	GSE39582	MSS	2220	0
GSM971961	GSE39582	MSS	960	1

2.c. Survival Fit & Save P-Value

```r
# Survival Fit
fit <- survfit(
  Surv(
    as.numeric(time),
    as.numeric(factor(censor))
  ) ~ gene_exprs_values,
  data = surv_data
)

# Save p-value and bio marker potential
p_values[gene_name, "P.Value"] <- surv_pvalue(fit)$pval
if (surv_pvalue(fit)$pval < 0.05) {
  p_values[gene_name, "Potential.Biomarker"] <- "Yes"
} else {
  p_values[gene_name, "Potential.Biomarker"] <- "No"
}
```

2.c. Survival Analysis Plots





2.c. Survival Analysis Results

	P.Value	Potential.Biomarker
HOXC6	0.0125353553606412	Yes
EIF5AL1	4.26532654160505e-05	Yes
RUBCNL	0.0273241031024493	Yes
QPRT	0.0349456910871061	Yes
CXCL13	0.0551716176716995	No
EIF5AP4	0.324088803513041	No
REG1A	0.536839668368961	No
GNLY	0.860139322325364	No
PLA2G2A	0.283214550835236	No
DUSP4	0.577644880134269	No

	P.Value	Potential.Biomarker
ADGRG6	0.328860532091985	No
CXCL9	0.927920542627139	No
KRT23	0.176344771555069	No
LY6G6F.LY6G6D	0.509200370263944	No
LY6G6D	0.509200370263944	No
CXCL14	0.872249443280306	No
TDGF1	0.666246050309238	No
TDGF1P3	0.666246050309238	No
FABP1	0.230384402752102	No
SLC26A3	0.700590707901597	No

3. Gene-Disease Research

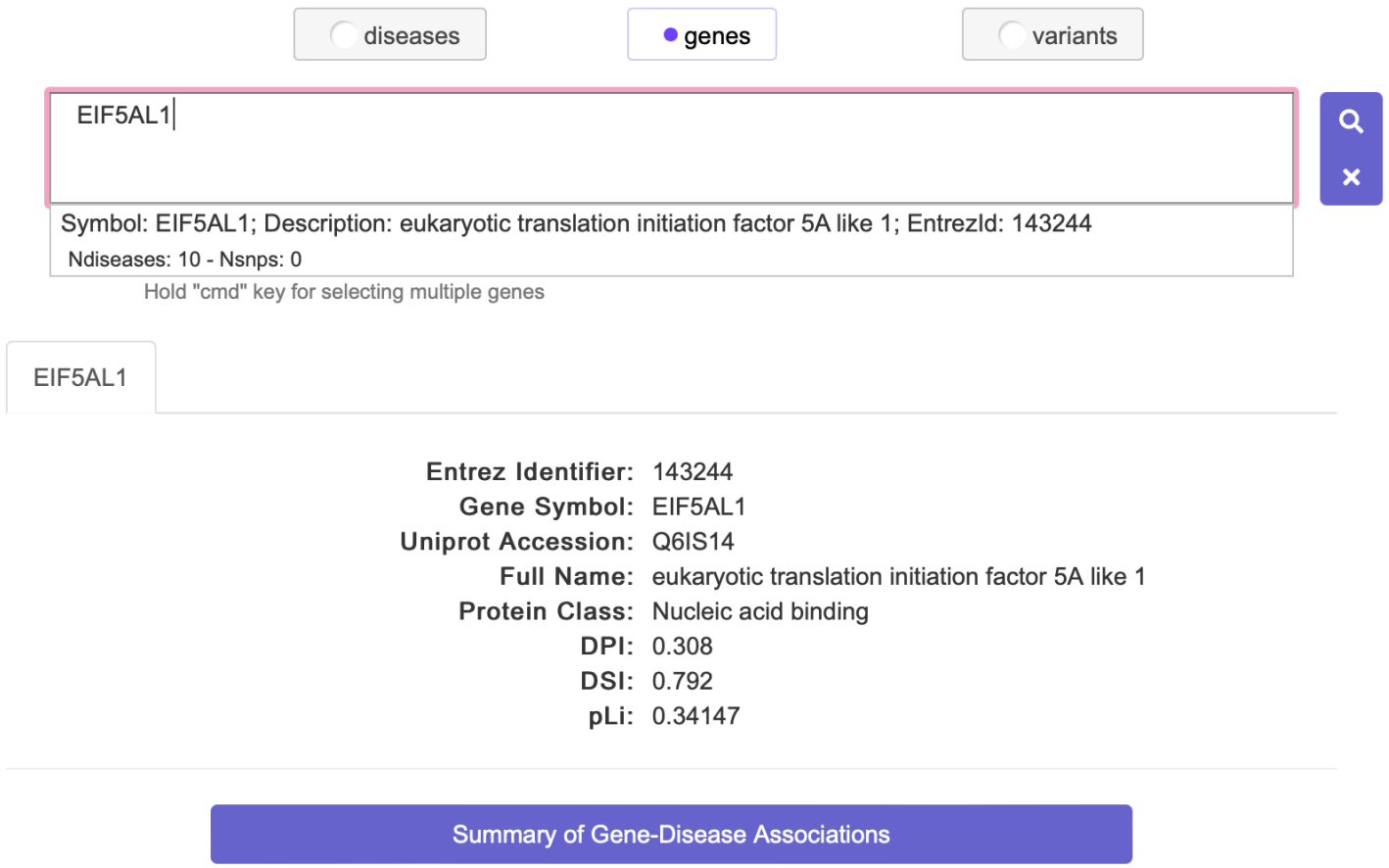
Search Gene-Disease Associations

DisGeNET:

A discovery platform containing one of the largest publicly available collections of genes and variants associated to human diseases.

https://www.disgenet.org/search

Research On DisGeNET



Research On DisGeNET

EIF5AL1, eukaryotic translation initiation factor 5A like 1, 143244

N. diseases: 10; N. variants: 0

Source: ALL

1 - 10 of 10 results

Add/Remove filter | Download | Share

Results per page 25

Filter within current results:

Disease	Type	Disease Class	Semantic Type	N. genes d	N. SNPs d	Score gda	EL gda	EI gda	N. PMIDs	N. SNPs gda	First Ref.	Last Ref.
Colorectal Carcinoma	disease	Digestive System Disea…	Neoplastic Process	5473	1962	0.010	None	1.000	1		2018	2018
Carcinoma, Ovarian …	disease	Neoplasms; Female Ur…	Neoplastic Process	2841	327	0.010	None	1.000	1		2001	2001
Dysferlinopathy	disease	Congenital, Hereditary, …	Disease or Syndrome	30	59	0.010	None	1.000	1		2020	2020
Malignant neoplasm …	disease	Neoplasms; Female Ur…	Neoplastic Process	2563	315	0.010	None	1.000	1		2001	2001

New Potential Survival Markers

- HOXC6: Prostate Carcinoma and Cervix Carcinoma (Shown to cause CRC)
- EIF5AL1: Ovarian Carcinoma, Colorectal Carcinoma and Lung Neoplasm
- RUBCNL: Cervix Cancer/Carcinoma and High Grade Cervical Intraepithelial Neoplasia (Shown to cause CRC)
- QPRT: Huntington Disease (an autosomal dominant progressive neurodegenerative disorder)

Thanks for Listening