Evalution of binary classification pipelines and methods for 16S rRNA gene data

Running title: Machine learning methods in microbiome studies

Begüm D. Topçuoğluˆ1, Jenna Wiensˆ2, Patrick D. Schloss[1†]

† To whom correspondence should be addressed: pschloss@umich.edu

1. Department of Microbiology and Immunology, University of Michigan, Ann Arbor, MI 48109

2. Department of Computer Science and Engineering, University or Michigan, Ann Arbor, MI 49109

Abstract

Introduction

As gut microbiome field continues to grow, there will be an ever-increasing demand for reproducible machine learning methods to analyze microbiome sequence read count data and to determine association with a continuous or categorical phenotype of interest.

Colorectal cancer is one of the leading cause of death among cancers in the United States. Early diagnosis increases the chance of survival. However the current diagnostic methods are expensive and invasive. As a less invasive tool, numerous studies use relative abundances of the gut bacteria populations to predict disease progression. Most microbial communities are pretty patchy and the likelihood of a single feature that explains the differences in health is pretty small. It is likely that many biomarkers are needed to account for the patchiness as well as the context dependency of the features.

ML use in microbiome literature is a bit like the wild west with lack of clarity over methods, testing, validation, etc. There is a need for guidance on how to properly implement these different methods. We need to emphasize good machine learning practices and pipelines and discuss the reproducibility, robustness and actionability of models.

We established a non-leaky pipeline. We performed L1 and L2-regularized logistic regression, Linear SVM, Non-Linear SVM, Decision tree, Random forest, XGBoost and Feed Forward Neural Net classification models. We evaluated the classification performance of different machine learning methods. We also want to discuss the reproducibility, robustness, actionability, interpretibility and susceptibility to overfitting of each method.

Generalisation Perfomance of each model. Is there a maximum threshold of prediction with all these methods? Does an increase in model complexity improve predictibility? Synthesis statement regarding modeling 16S microbiome data

Results and Discussion

Conclusions

Materials and Methods

Table 1: Optimized hyper-parameters, pre-processing and cross-validation methods and software implementation of the classification algorithms.

Method	**Parameter**	**Cross Validation**	**Epoch**	**Scaler**	**Sklearn Function**
Logistic Regression	C	5-fold, 100-repeats	100	MinMax	LogisticRegression
L1 SVM Linear Kernel	C	5-fold, 100-repeats	100	Standard	LinearSVC
L2 SVM Linear Kernel	C	5-fold, 100-repeats	100	Standard	LinearSVC
SVM RBF Kernel	C, gamma	5-fold, 100-repeats	100	Standard	SVC
Decision Tree	max_depth, min_samples_split	5-fold, 100-repeats	100	MinMax	DecisionTreeClassifier
Random Forest	n_estimators, max_features	5-fold, 100-repeats	100	MinMax	RandomForestClassifier
XGBoost	n_estimators, colsample_bytree, learning_rate, subsample, max_depth, min_child_weight	5-fold, 100-repeats	100	MinMax	XGBClassifier

Figure 1. Generalization and classification performance of modeling methods AUC values of all cross validation and testing performances.

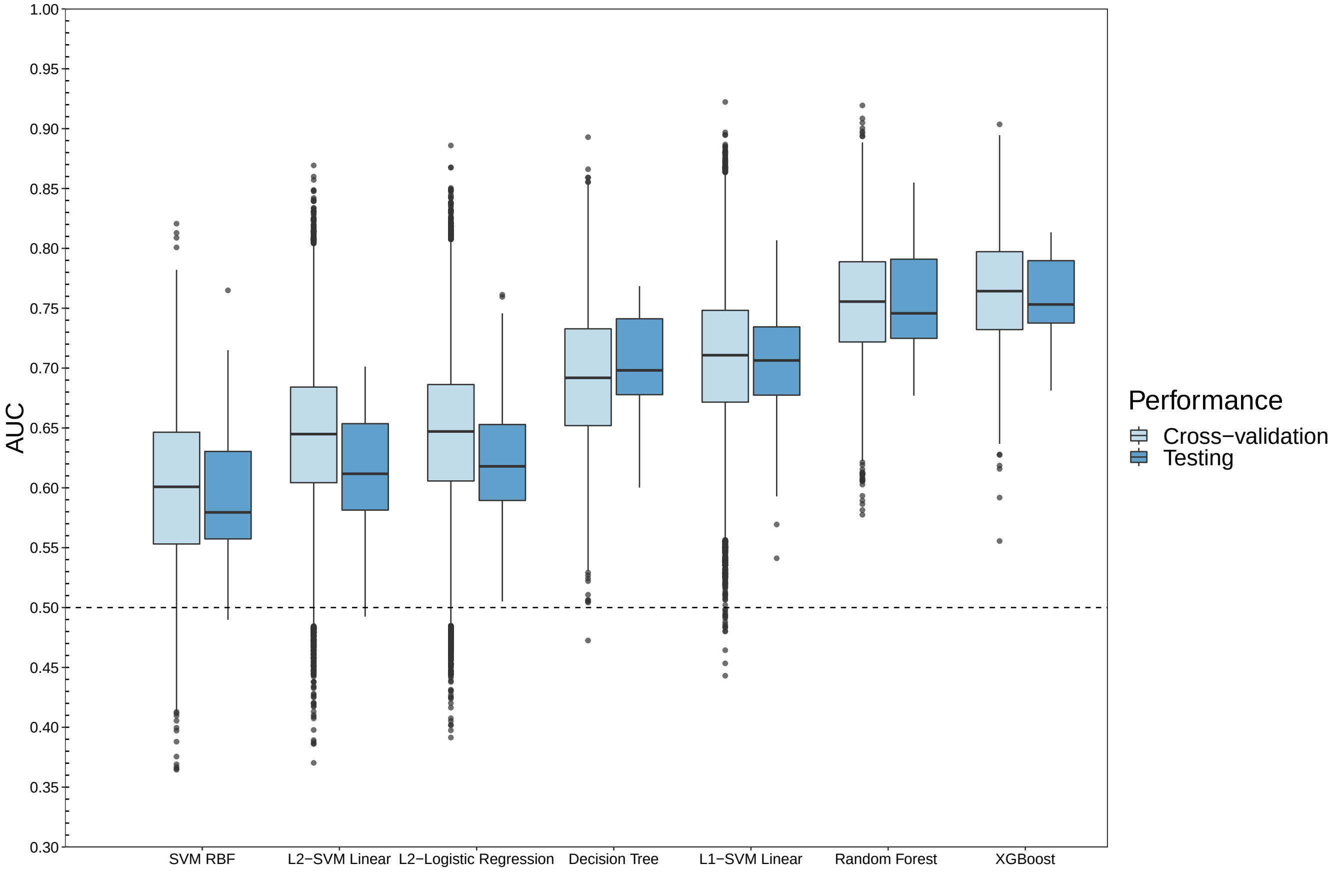

Table 2: The range of optimized hyper-parameters for logistic regression and support vector machines.

Parameter	L2 Logistic				L1 SVM Linear				L2 SVM Linear			SVM RBF				
C	0.01	0.1	1	10	0.001	0.01	0.1	1	0.01	0.1	1	1e-06	1e-05	1e-04	0.001	0.01
gamma	-	-	-	-	-	-	-	-	-	-	-	1e-09	1e-08	1e-07	-	-

Table 3: The range of optimized hyper-parameters for tree based classification algorithms.

Parameter	Random Forest					Decision Tree				XGBoost		
learning_rate	-	-	-	-	-	-	-	-	-	0.01	0.1	1
max_depth	-	-	-	-	-	6	8	10	50	6	7	8
max_features	10	80	500	1000	1500	-	-	-	-	-	-	-
min_child_weight	-	-	-	-	-	-	-	-	-	1	2	3
min_samples_split	-	-	-	-	-	10	25	50	-	-	-	-
n_estimators	1000	-	-	-	-	-	-	-	-	100	-	-
subsample	-	-	-	-	-	-	-	-	-	0.7	0.8	0.9