MGT 6748 - Applied Analytics Practicum – Spring 2022
Final Report

Dog Diseases Classification and Prediction though Genetic Composition

Yi Feng Zhu; Shara Handigund; Adrian Pardo

Acknowledgement

We would like to thank you Mr. Calvin Leather from Embark, who continuously give us insightful information and guide us throughout the project. We also like to thank our TAs for their tremendous efforts in organizing this group project.

1. Introduction

Embark Veterinary is a provider of genetic testing services for dogs. It provides a dog owner with a test kit that contains a swab, which allows the owner to perform a swab in the dog's mouth to obtain salvia. The salvia contains a dog's DNA, which is extracted and applied to a microarray chip for analysis. The microarray is a silicon chip which consists of a number of microscopic chemical probes, and each probe detects a binary genetic state at a specific position in a dog's genome and fluorescing a certain color that depends on the identified state.

The color and intensity are captured by a camera and then digitized into a pair of floating-point numbers between 0 and 1. In this case, they are called b_allele_frequency and log_r_ratio in the raw dataset. With this pair of data, it can be classified into 3 genetic states; they are at risk, carrier or free from a disease. We study this raw dataset that contains around 4000 data points, build and compare different classification models that can classify each data point into a specific genetic state.

2. Problem

The DNA from the swab is applied to a genotyping array and stained with fluorescent tags; then, the array will be analyzed to determine whether a dog is at risk, or a carrier or free from a specific disease. The problem is the same DNA array can be assessed with multiple methods; some methods are more accurate but cost more, while some methods are less accurate but cost less.

3. Goal

We develop supervised featurization and classification models that translate the raw microarray data into health risk/phenotype status; then, we predict the health risk/phenotype status for new raw data based on the model. We also explore quantifying confidence to support active learning.

4. Outlines of Process Steps

First, we explore the data by using bar charts and scatterplots to see how the raw data is distributed, which helps us visualize whether the data has outliers or not, and determine if the class imbalance issue exists in the dataset. Based on this observation, we do data cleanup by

removing outliers, and then try different classification models. Finally, we use F1-macro score as our metric to choose the best modeling method for the raw data.

Fig. 1 Outlines of process steps

5. Exploratory Data Analysis

First, we explore a classified genotype dataset. This dataset contains about 4000 data points from 16 microscopic chemical probes; for each probe, each data point is a binary genetic state such as AA, GG, GA. The binary state is derived from two floating point numbers we mentioned above. A partial of the dataset is shown below.

chrX_86333309	chrX_98838845	chrX_104108957	short_id
G G	A A	A A	6965220932964176893
A A	A A	G A	3758760163069260076
A A	A A	A A	736952308690390811

Fig. 2 Binary genotype from each probe from each dog

We use a bar chart to visualize how each genotype is distributed for each probe.

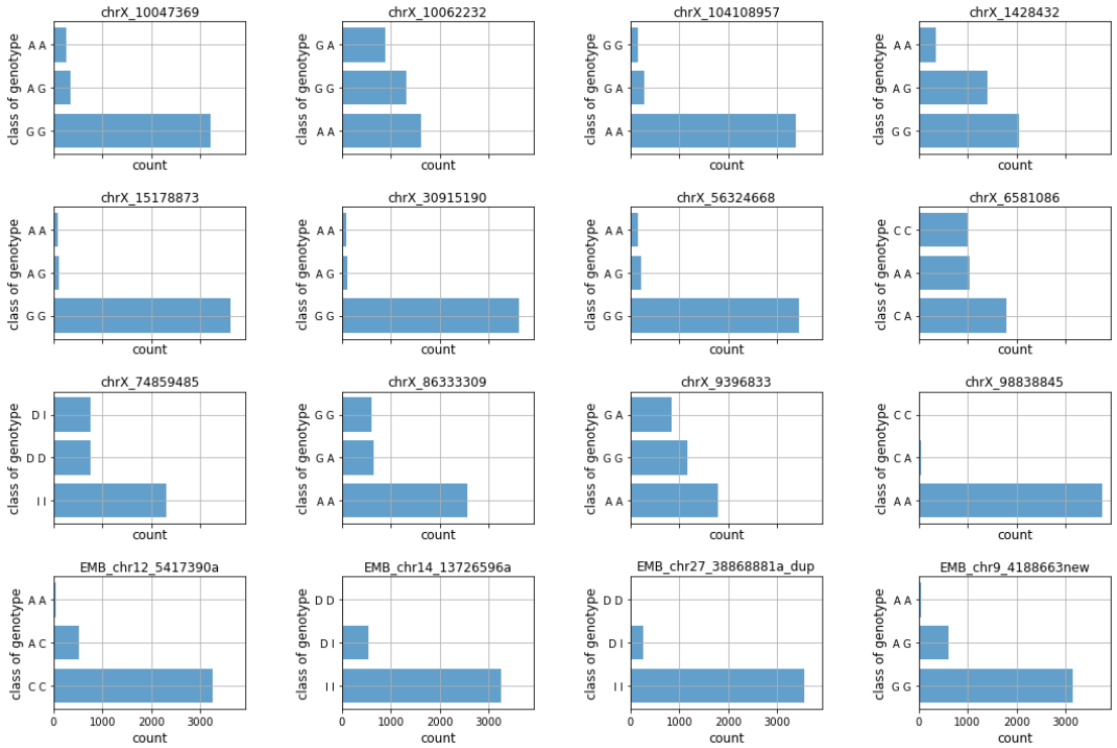


Fig. 3 Genotype distribution for each probe

The distribution of each genotype for each probe is not balanced. Since we are going to use classification models to classify and predict these genotype labels, the imbalanced data can pose a challenge for classification models, because these models are built based on the assumption that each class has an equal distribution of data samples. If one class has a significantly small number of data samples, the predictive performance for this class will be lower because this minority class is more important to the model and hence, sensitive to the classification error. At the end of this report, we will explore more for class imbalance and methods to overcome this issue.

In addition to the genotype dataset, we have another raw genotype dataset. This dataset consists of floating numbers ranging from 0 to 1; they are b_allele_frequency and log_r_ratio, which are the raw data we mentioned above. This dataset has more than 60000 entries; each data point is captured at a specific location on the microarray. There are 16 different genotypes, which are the ones we see in the classified genotype dataset above. A snippet of this dataset is shown below.

SNP Name	b_allele_frequency	log_r_ratio	short_id
chrX_10047369	1.0	-0.231731	5286875961930235904
chrX_10062232	1.0	0.012933	5286875961930235904
chrX_104108957	0.0	-0.137437	5286875961930235904

Fig. 4 Raw genotype data from each probe from each dog

Notice both the raw genotype dataset and the classified genotype dataset has a short_id column where each id represents a dog. If we merge these two datasets, we get the following.

SNP Name	b_allele_frequency	log_r_ratio	short_id	EMB_chr9_4188663new
EMB_chr9_4188663new	0.000000	0.390494	528435666444227822	A A
EMB_chr9_4188663new	0.005518	0.521873	5343089894682058657	A A
EMB_chr9_4188663new	0.013333	-1.688816	8791026527077684672	A A

Fig. 5 Merged dataset based on raw genotype dataset and classified genotype dataset

Then, we use a scatter plot to visualize how each pair of raw data point is distributed for each probe.

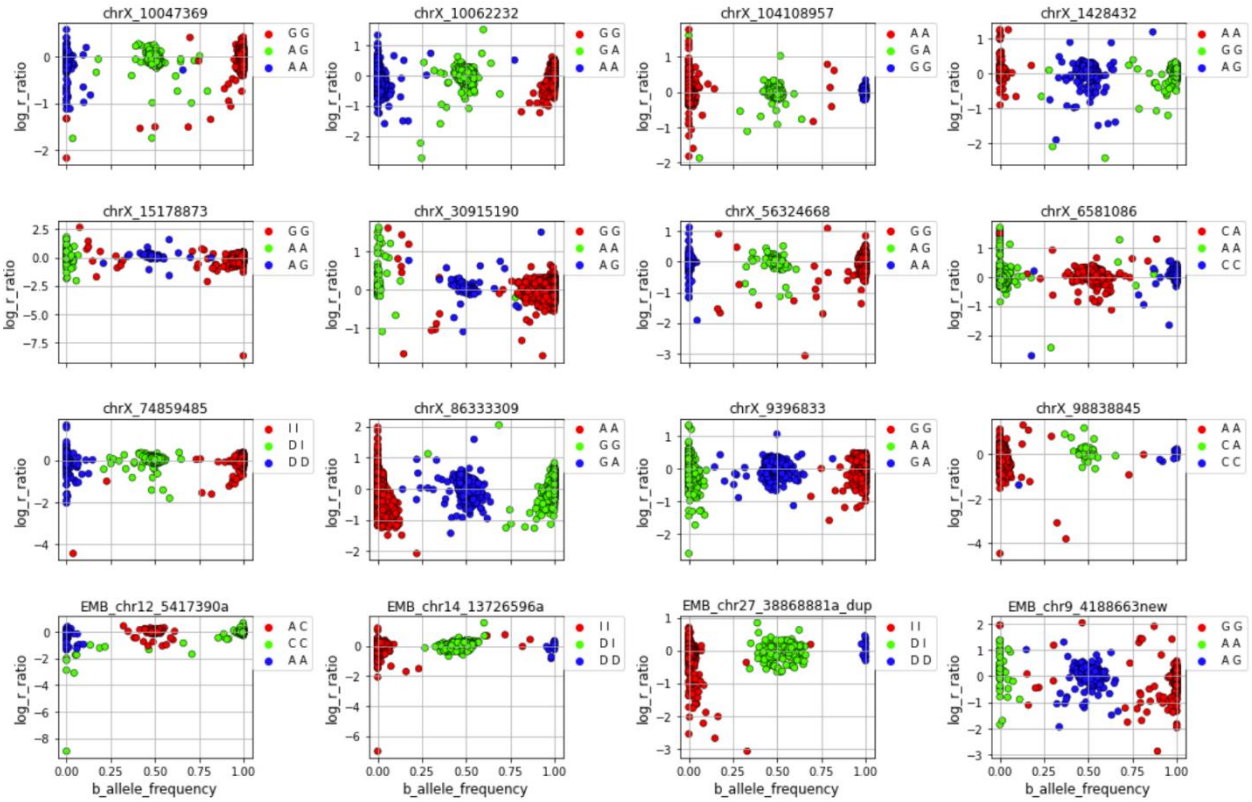


Fig. 6 Scatter plot for raw genotype distribution for each probe

We can see there are mainly three clusters in each plot, and each cluster is classified by a genetic state. Since the classification results are manually verified by a researcher. We can treat this merged dataset as a training and testing dataset, build classification models to predict a dog's genetic state based on the raw data from the probe directly.

However, like other data analytics projects, we need to do a cleanup for the dataset. Take a look at one scatter plot for one probe, although we can see tree types of classes, there are quite a few outliers in the dataset. For example, while most data points are classified as GG shown on the far right of the graph, there are some points classified as GG but are located at the far left where the class AA is located. Hence, in order to reduce the impacts from those outliers when we are doing classification modeling, we have to remove outliers for each class for each probe. We will investigate the impact of outliers in later sections; but for now, we build classification models using the raw data as is.

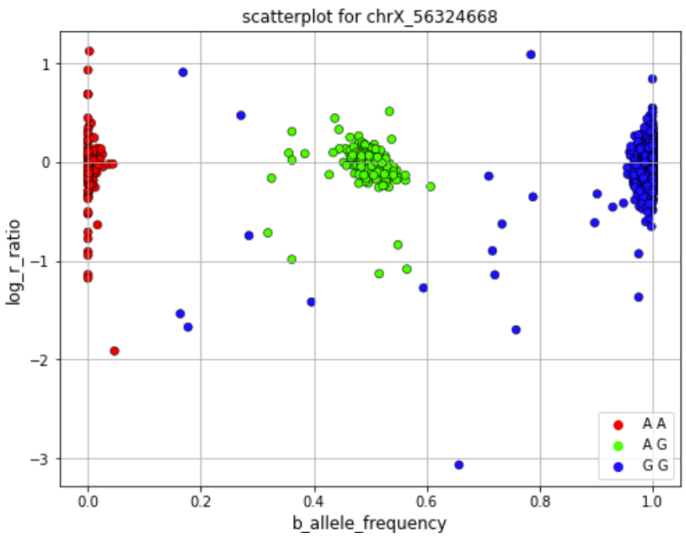


Fig. 7 Scatter plot showing some outliers

6. Data Preparation

We study the raw data from a probe called: chrX_56324668. The choice of this probe is entirely random. We like to demonstrate the workflow and our findings; then, apply the same methods to other probes. First, we filter down the data related to chrX_56324668 in the pre-classified genotype dataset, then, we merge it with the raw genotype dataset based on the short_id column. A snippet of this merged dataset is shown below.

SNP Name	b_allele_frequency	log_r_ratio	short_id	chrX_56324668	chrX_56324668_coded
chrX_56324668	0.003158	-0.112676	208372981168597684	A A	0
chrX_56324668	0.000000	-0.776971	8789827500040913209	A A	0
chrX_56324668	0.000000	-0.909364	2612743550791185753	A A	0

Fig. 8 A snippet of a merged dataset for chrX_56324668

We also plot a bar chart to see how the three genotype classes are distributed. The GG class outnumbers the other classes. This is one issue we want to keep in mind when building classification models. Later in the report, we will explore other methods to tackle the issue of imbalanced data.

chrX_56324668	count
G G	3443
A G	230
A A	147

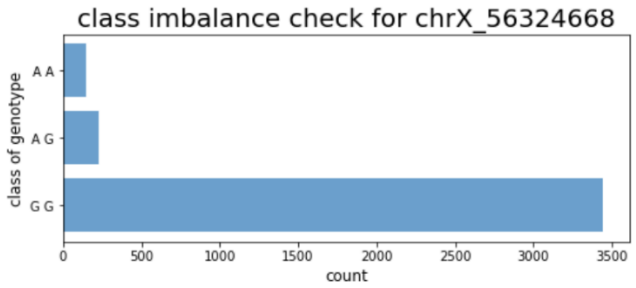


Fig. 9 Class imbalance check

Separation of Training and Testing Data
We allocate 70% of the data for training and 30% of the data for testing.

7. Classification Models

7.1 Model Overview
We explore the following classification methods: multi-nominal logistic regression, k-NN, SVM and Random Forest. Every classification model has its pros and cons. The multinomial logistic regression is used when we have a categorical dependent variable with two or more classes. In our case, we have three genotype classes. Multinomial logistic regression is similar to logistic regression, except that logistic regression can only classify binary classes such as 0 or 1, Yes or No, and etc. If the boundary cannot be linearly separated, we can use SVM along with non-linear RBF kernel method to do classification. If the training data is too noisy, meaning the dataset contains some outliers, we can use k-NN method because it is robust to noisy training data; however, we need to tune the parameter K, which is the number of nearest neighbors, and also choose the type of distance to be used for k-NN. Random Forest consists of a group of Decision Trees, where each tree consists of decision branches to separate data. Random Forest can handle categorical features very well as well as high dimensional data with a large amount of training data.

7.2 Metric used for Model Evaluation
One of the most important things during model evaluation is to choose an appropriate evaluation metric. Common evaluation metrics include but not limited accuracy, precision, recall, weighed average, and so on. When dealing with multi-classes classification project, the F1 score is commonly used. First, we demonstrate a simple example of how F1-micro and F1-macro score compares.

Suppose we have a confusion matrix with three classes; the values are the diagonal line are total number of classes predicted correctly, and all other values means incorrect predictions.

		Predicted		
Actual		AA	AG	GG
	AA	2	7	2
	AG	9	1	2
	GG	1	2	50

Then, we can calculate True Positive, False Positive and False Negative.

		True Positive TP	False Positive FP	False Negative FN
Actual	AA	2	10	9
	AG	1	9	11
	GG	50	3	3

To calculate a F1 score, we need to calculate precision and recall first. Precision calculates how many predicted positive cases are truly positive. The equation is:

$$Precisoin = \frac{TP}{TP + FP}$$

Recall calculates of all the actual positive cases, how many of them are predicted to be positive correctly. The equation is:

$$Recall = \frac{TP}{TP + FN}$$

Then, F1 score is calculated as:

$$F1\ score = 2\frac{Precision \times Recall}{Precision + Recall} = \frac{TP}{TP + 0.5(FP + FN)}$$

Hence, we have the following results:

		True Positive TP	False Positive FP	False Negative FN	Precision	Recall	F1 score
Actual	AA	2	10	9	0.167	0.181	0.173
	AG	1	9	11	0.1	0.083	0.091
	GG	50	3	3	0.943	0.943	0.943

With the F1 score, we can calculate F1-macro and F1-micro score. The macro-averaged F1 score is computed by taking the arithmetic mean (aka unweighted mean) of all the per-class F1 scores.

$$F1 - macro\ score = \frac{\sum F1}{n}$$

		True Positive TP	False Positive FP	False Negative FN	F1 score	F1-macro score
Actual	AA	2	10	9	0.173	$\frac{0.173 + 0.091 + 0.943}{3} = \mathbf{0.402}$
	AG	1	9	11	0.091	
	GG	50	3	3	0.943	

On the other hand, F1-micro score is derived by counting the sums of the True Positives (TP), False Negatives (FN), and False Positives (FP). We first sum the respective TP, FP, and FN values across all classes and then plug them into the F1 equation to get our F1-micro score:

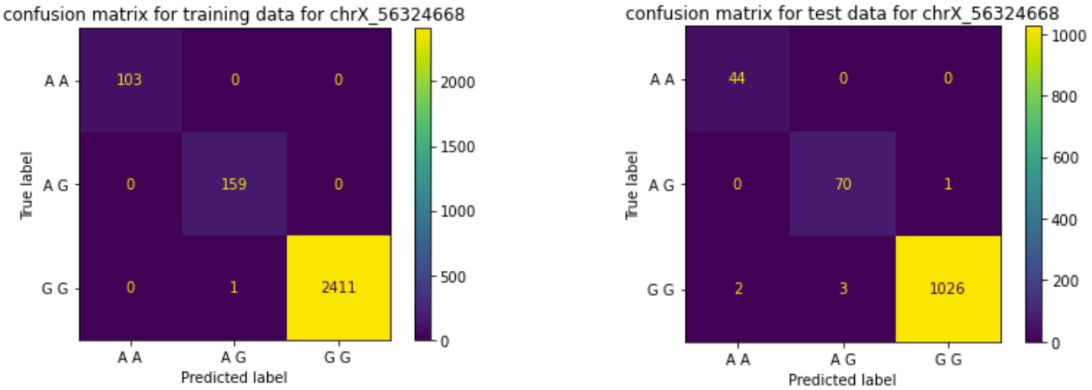
		True Positive TP	False Positive FP	False Negative FN	F1-micro score
Actual	AA	2	10	9	$\frac{53}{53 + 0.5(22 + 23)} = \mathbf{0.612}$
	AG	1	9	11	
	GG	50	3	3	
	Total	53	22	23	

As we can see, the F1-micro score is higher than the F1-macro score. The micro averaging computes the proportion of correctly classified cases out of all cases.

The purpose of doing the calculations is to show how different metrics can give different results. If not careful, the model evaluation can be misleading. In our case, we have a dataset with imbalanced classes, each class is equally important to be predicted correctly. Hence, we want to treat all classes equally; therefore, F1-macro score is preferred, and it is the metric we use for model evaluation.

7.3 Multi-nominal Logistic Regression

Logistic regression is a supervised classification algorithm. It is used to calculate the probability of a binary event, such as 1 or 0, Yes or No, etc. For example, in a binary class of 0 and 1, if the predicted probability of one event is greater than a common default value, 0.5, that event will be classified as 1. Multi-nonnominal Logistic Regression is an extension of Logistic Regression, it can handle more than two types of classes. In our case, we have three genotypes to classify for a specific probe. The package we use is LogisticRegression() from scikit-learn. Both the confusion matrix for training and testing data is shown below.



f1 macro score for training dataset with tuning is: 99.78%

f1 macro score for testing dataset with tuning is: 98.24%

Fig. 10 Confusion matrix for training and testing data using Multi-nominal Logistic Regression

In a confusion matrix, the values on the diagonal lines mean the number of data points are correctly classified by the algorithm. In this case, the Multi-nominal Logistic Regression achieve a F1-macro score of 99.78% on the training dataset and 98.24% on the testing dataset. We also plot a boundary plot for a better visualization.

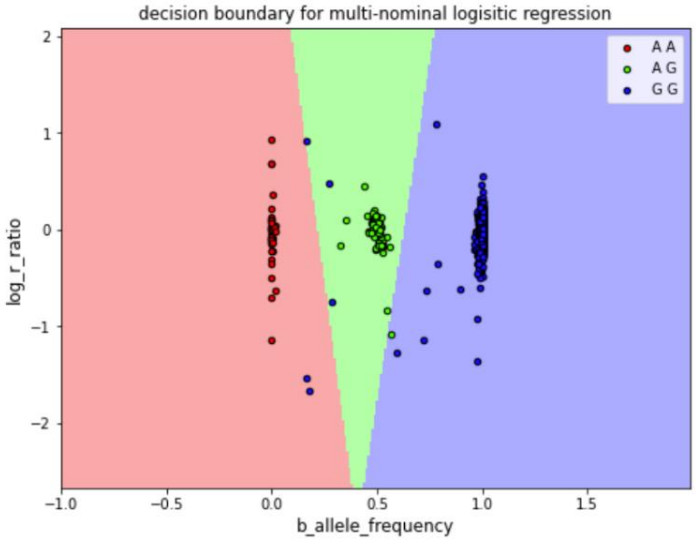


Fig. 11 Decision boundary for Multi-nominal Logistic Regression on testing data

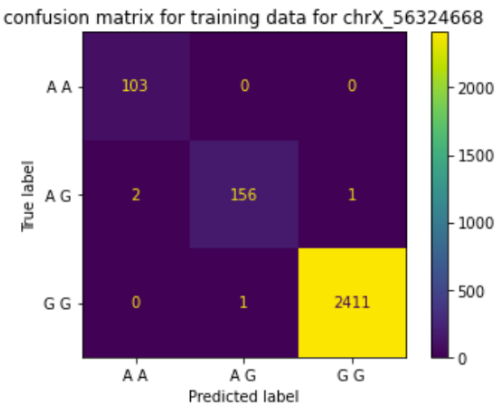
The boundary plot gives a better visualization which shows how each testing data point is classified into each boundary. Because Logistic Regression is a linear model, it cannot handle non-linearity in the dataset. If two clusters of data are not well separated, Logistic Regression may incorrectly classify data points at the boundaries. In the boundary plot above, we see that the boundaries are linear. Some data points labeled as GG are mis-classified. This is also due to outliers we mentioned before. Overall, most data points are correctly classified.

7.4 k-NN

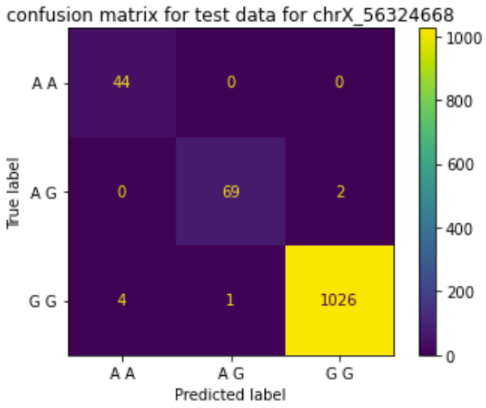
k-NN is a simple supervised learning algorithm. It makes no assumptions because it simply calculates the distance between two data points for all the data on a graph; then, it adds a paired of the calculated distance and the index of the data point to an ordered collection. Finally, sort the collection of distances and indices from smallest to largest. Depending on the number of k, each data point will be labeled with a k value, which becomes our class.

There are two main drawbacks from k-NN, which are relevant to the training data we have. k-NN is sensitive to outliers and imbalanced data. Since it's a distance-based algorithm, outliers in the dataset can be misclassified in most cases. In terms of an imbalanced dataset, if most data points belong to one class, then, the model will give a lot of preference to that class. As a result, the data points in the minority class will get misclassified.

The package we use is KNeighborsClassifier() from scikit-learn. Both the confusion matrix for training and testing data is shown below.



f1 macro score for training dataset with tuning is: 99.50%



f1 macro score for testing dataset with tuning is: 97.73%

Fig. 12 Confusion matrix for training and testing data using k-NN

In this case, k-NN also achieves a higher F1-macro score for both training and testing data points. The decision boundary plot also confirms this.

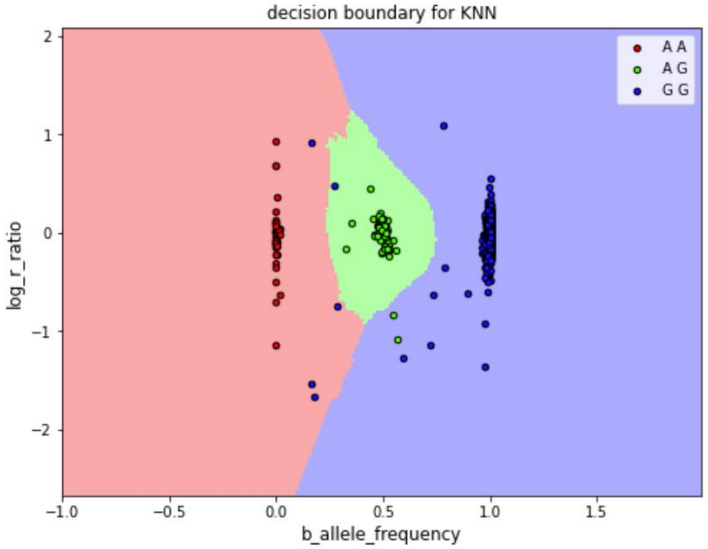


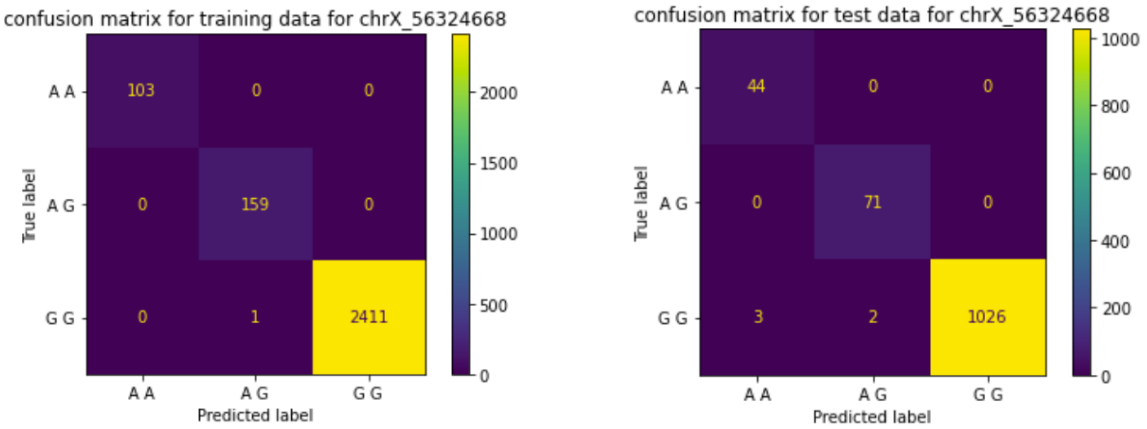
Fig. 13 Decision boundary for k-NN on testing data

Since k-NN calculates the distance between two points, the formula used for calculating the distances can be nonlinear; hence, compared to Multi-nominal Logistic Regression, the decision boundaries at some outliers are nonlinear, as k-NN tries to best fit those points to their individual class. Overall, k-NN correctly classifies most data points.

7.5 SVM

The idea behind SVM is to create a separation line or a hyperplane that separates two class of data points. SVM identifies points that are close to a separation line, the points are called Support Vectors; SVM will then compute the distance between the line and each support vector. This distance is called margin. The goal is to find a separation line or a hyperplane that can achieve a maximum margin possible.

The package we use is SVC() from scikit-learn. Both the confusion matrix for training and testing data is shown below.



f1 macro score for training dataset with tuning is: 99.29%

f1 macro score for testing dataset with tuning is: 98.36%

Fig. 14 Confusion matrix for training and testing data using SVM

SVM by far has the highest F1-macro score; but the gain over the previous two models is negligible. One of the advantages of SVM is it can fit high dimensional training data with a clear margin of separation. In our case, though we don't have a high dimensional data, the separation of each class in the training data is clear, making it easy for SVM to classify. However, one of the drawbacks of SVM is that it is sensitive to outliers, and with its nonlinear kernel method, SVM will try to fit those outliers in the training dataset, causing misclassification or overfitting. The boundary plot for the testing data is shown below.

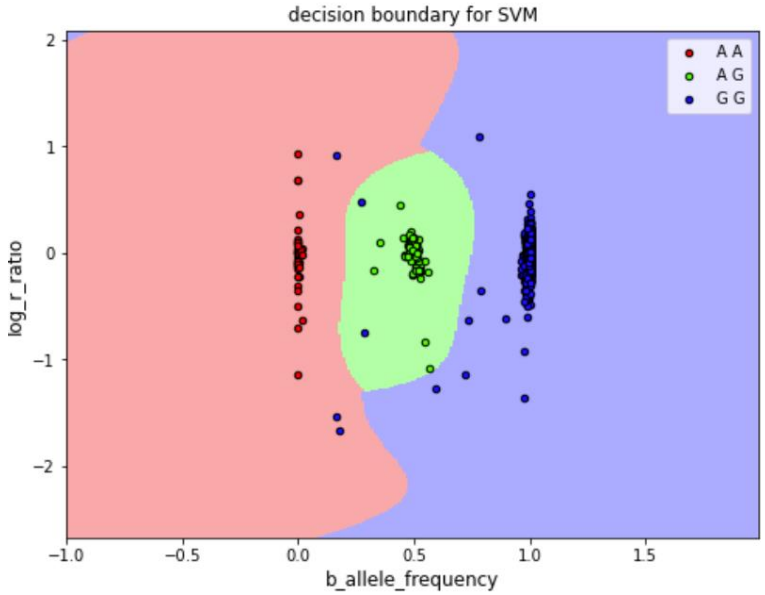


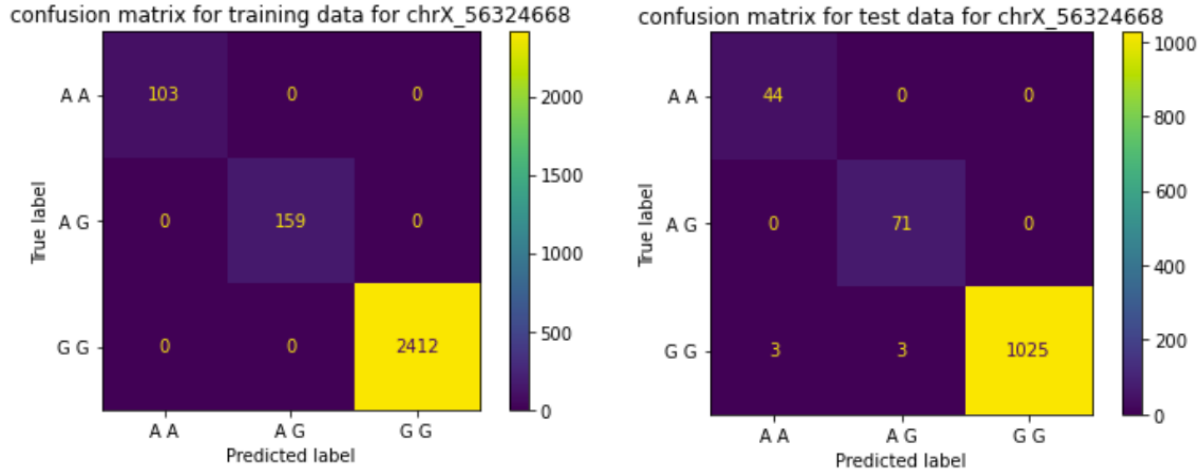
Fig. 15 Decision boundary for SVM on testing data

The decision boundaries from SVM exhibit a more pronounced nonlinearity at some outliers; in other words, it tries to fit those outliers that are far away from the majority of the data, in this case, the GG class. That is the reason why we have to remove outliers for SVM. In the later section, we will explore and see if this issue can be improved.

7.6 Random Forest

Random Forest is a tree-based machine learning algorithm that leverages the power of multiple decision trees for making decisions. Random Forest consists of numerous decision trees, and each decision tree consists of decision nodes, which can be a continuous or categorical variable used to separate the data points. This node creates two branches, and those branches become leaf nodes, which further be used to create additional branches based on additional criteria. The risk is obvious as the tree can grow too deep, causing overfitting.

The package we use is RandomForestClassifier() from scikit-learn. Both the confusion matrix for training and testing data is shown below.



f1 macro score for training dataset with tuning is: 99.89%

f1 macro score for testing dataset with tuning is: 98.11%

Fig. 16 Confusion matrix for training and testing data using Random Forest

Random Forest can achieve a very high F1-macro score, and the mis-classified data points comes from outliers. The decision boundary plot demonstrates as well.

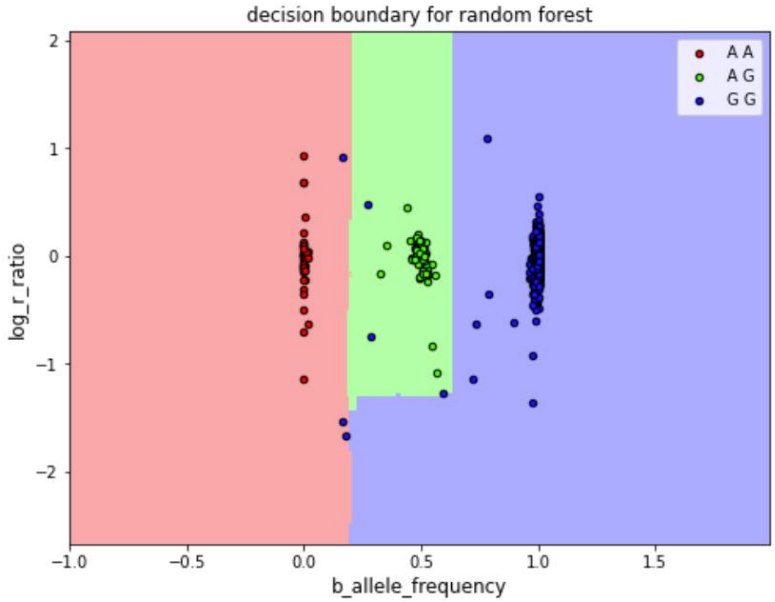


Fig. 17 Decision boundary for Random Forest on testing data

7.7 Hybrid Classification

The above classification algorithms use a single model from training data. Hybrid classification employs basic classification algorithms for model induction and for data preprocessing. Misclassification instances are usually considered to be noise, yet those still may carry useful information for identifying the class values of some other instances. We will be using hybrid classification and use training set to build 3 different models and testing set is classified by one of the classification models

7.8 Summary so far

We apply four classification modeling method to classify genotypes using original data without the removals of any outliers. All models can achieve a very high F1-macro score. SVM has the highest score.

Table 1: F1-macro scores for four models

Model Name	F1 macro score
Multi-nominal Logistic Regression	98.236152
k-NN	97.728181
SVM	98.357112
Random Forest	98.114167

Below is a grand summary of F1-macro scores for all four models and 16 probes.

Table 2: F1-macro scores for all models and probes

Model Name	Multi-nominal Logistic Regression (%)	k-NN (%)	SVM (%)	Random Forest (%)
chrX_10047369	98.02	98.49	98.51	98.32
chrX_10062232	99.83	99.73	99.73	99.80
chrX_104108957	100.00	99.78	99.34	99.79
chrX_1428432	99.74	98.97	99.61	99.74
chrX_15178873	96.10	94.45	95.43	97.72
chrX_30915190	95.64	94.32	93.42	95.49
chrX_56324668	98.24	97.73	98.36	98.11
chrX_6581086	99.59	99.53	99.53	99.63
chrX_74859485	99.81	99.39	99.81	99.81
chrX_86333309	99.82	99.78	99.38	99.82
chrX_9396833	100.00	99.91	99.91	100.00
chrX_98838845	96.74	93.26	92.31	94.71
EMB_chr12_5417390a	99.63	99.76	99.88	100.00
EMB_chr14_13726596a	99.88	99.76	99.64	99.76
EMB_chr27_38868881a_dup	99.76	100.00	100.00	100.00
EMB_chr9_4188663new	97.03	97.90	96.31	97.14

Models with the highest F1-macro scores are highlighted. Multi-nominal Logistic Regression and Random Forest can marginally outperform other models, but we notice that Multi-nominal Logistic Regression is much faster to run for this dataset. Hence, we would recommend Multi-nominal Logistic Regression.

7.9 Outlier Removals

Our initial runs are based on raw data with outliers. As we have mentioned several times that outliers have impacts on classification models. Our next steps are to remove outliers and re-fit all models. The widely used formulas for calculating outliers are as followed:

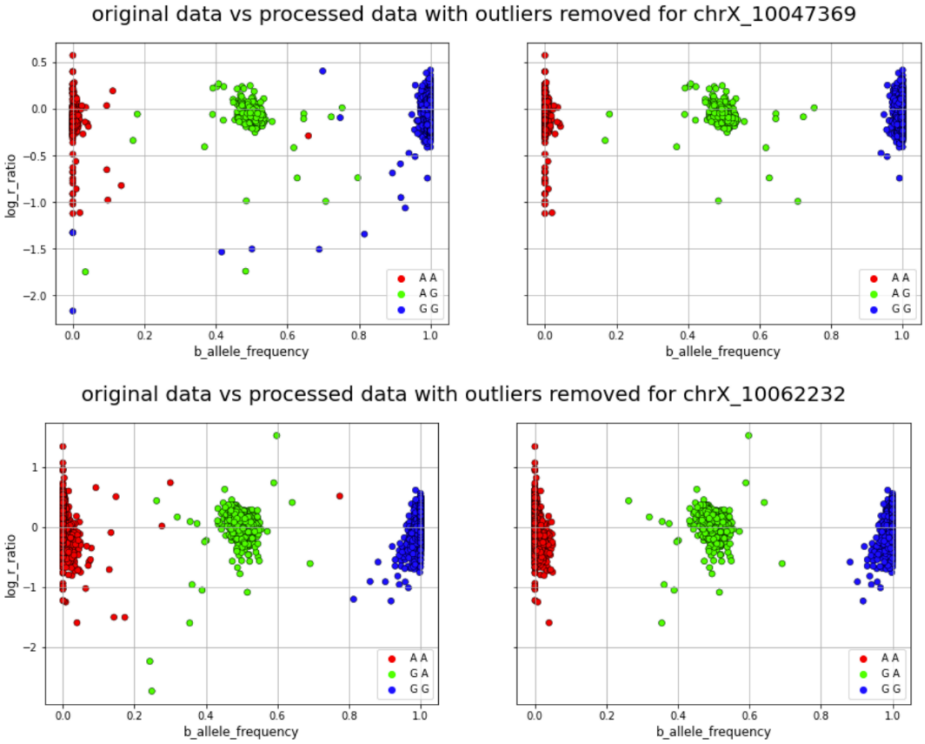
$$Low = Q_1 - 1.5 \times IQR$$
$$High = Q_3 + 1.5 \times IQR$$

where Q_1 and Q_3 is calculated as:

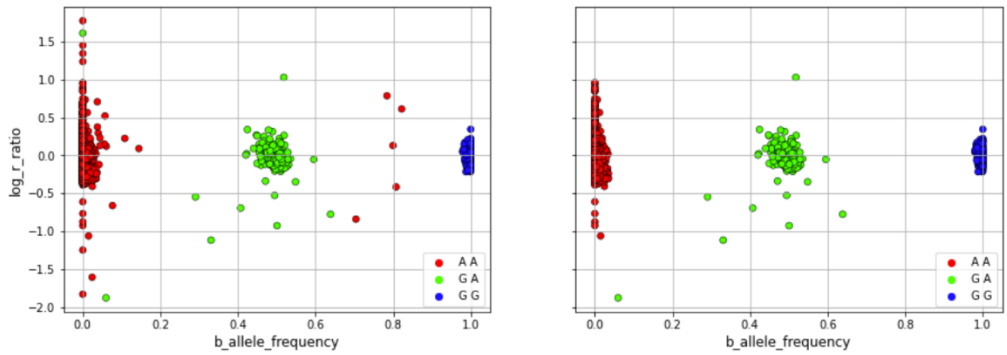
$$Q_1 = q_1(n+1)$$
$$Q_3 = q_3(n+1)$$

where q_1 and q_3 are the quantiles and n is the total number of data points. Before calculating Q_1 and Q_3, the value of each data point is ranked from the smallest to the largest. In our case, we pick $q_1 = 0.01$ and $q_3 = 0.95$ because we are conservative about removing outliers as we don't

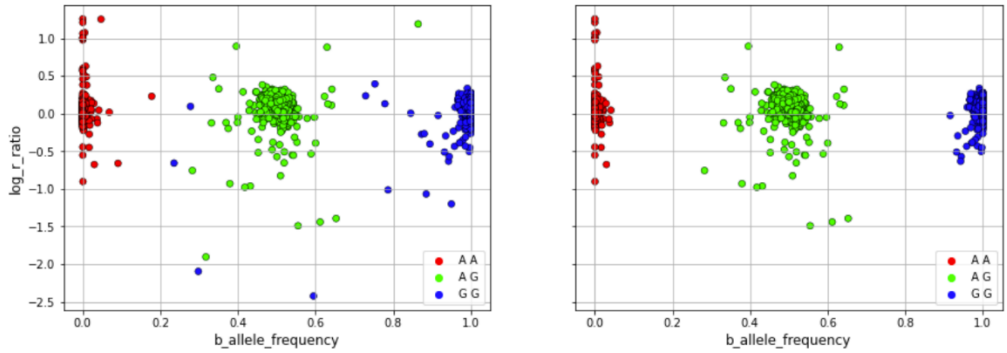
want to lose too much data points. When $q_1 = 0.01$, it means 1% of the data is below the position $0.01(n+1)$ and 1% of the data is above the position $0.01(n+1)$. IQR represents interquartile range, it is the difference between Q_3 and Q_1. Any data outside of the low and the high range will be purged. A comparison of before/after outlier removals is shown below for all probes.



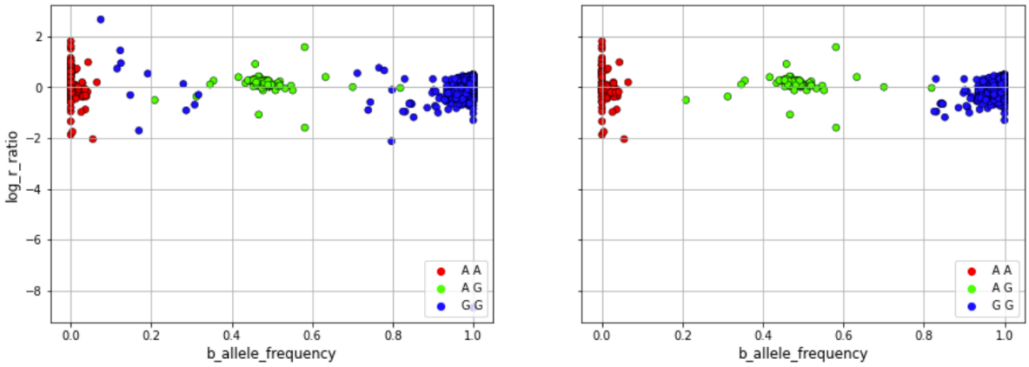
original data vs processed data with outliers removed for chrX_104108957



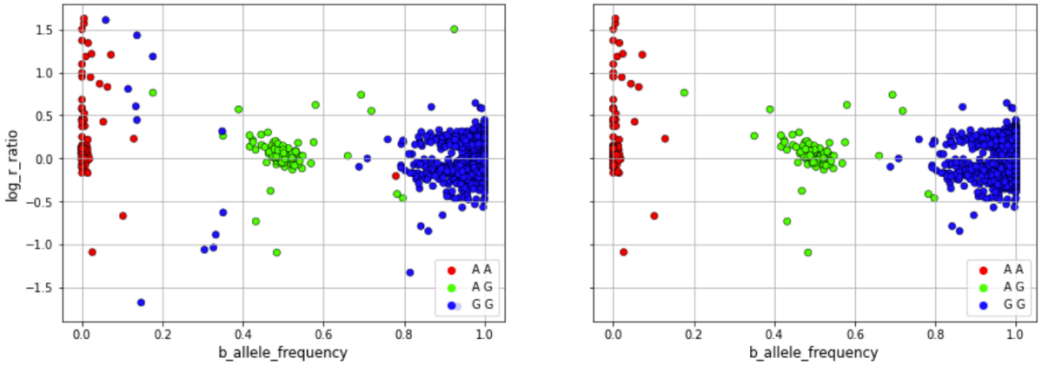
original data vs processed data with outliers removed for chrX_1428432



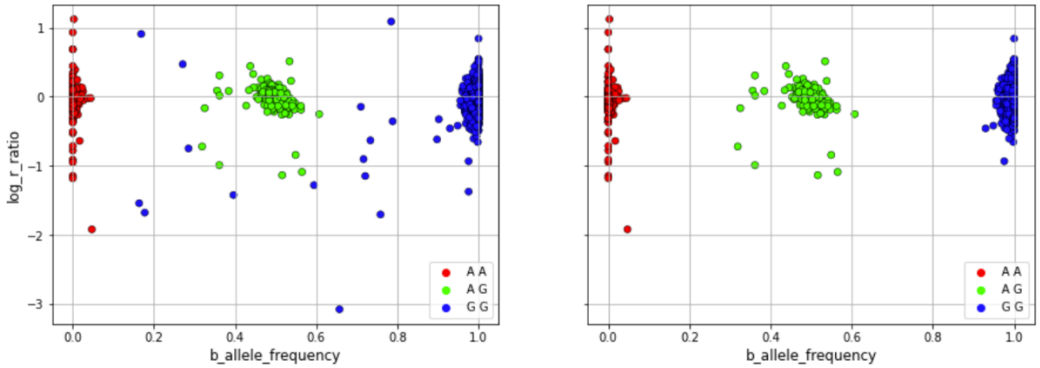
original data vs processed data with outliers removed for chrX_15178873



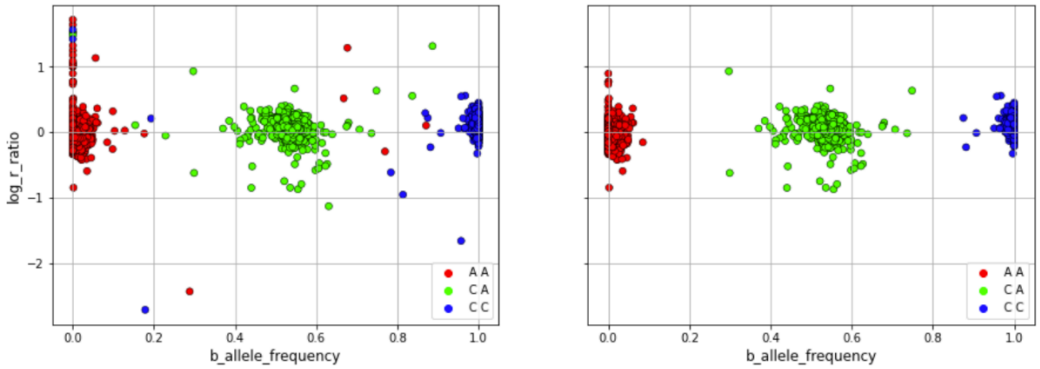
original data vs processed data with outliers removed for chrX_30915190



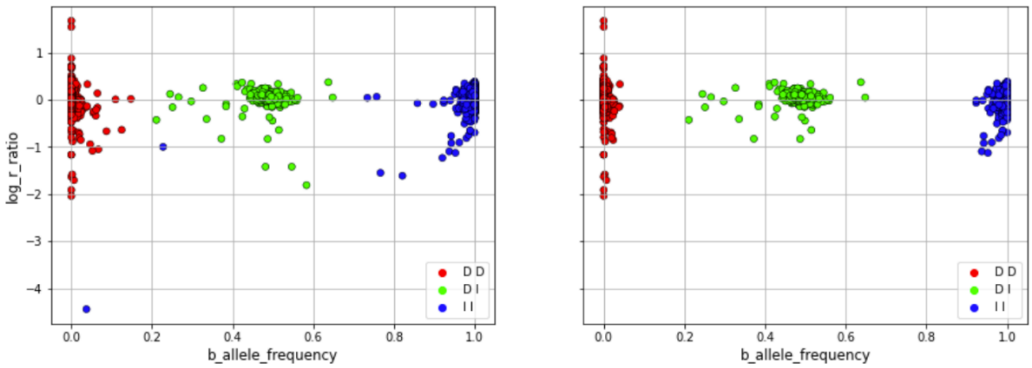
original data vs processed data with outliers removed for chrX_56324668



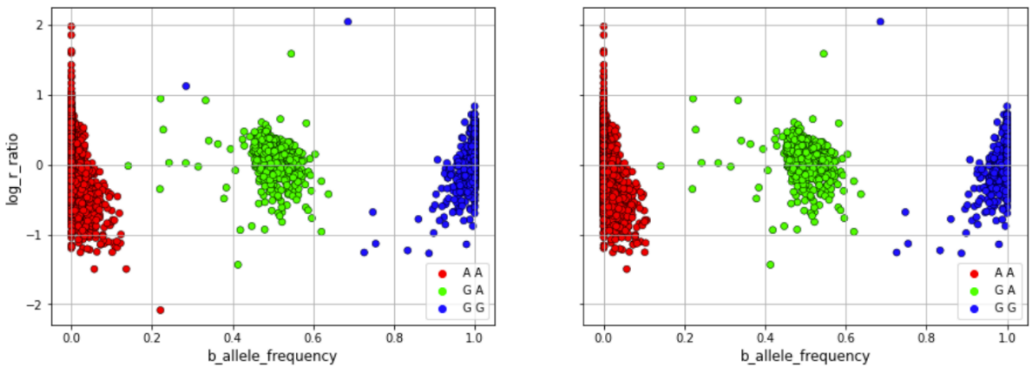
original data vs processed data with outliers removed for chrX_6581086



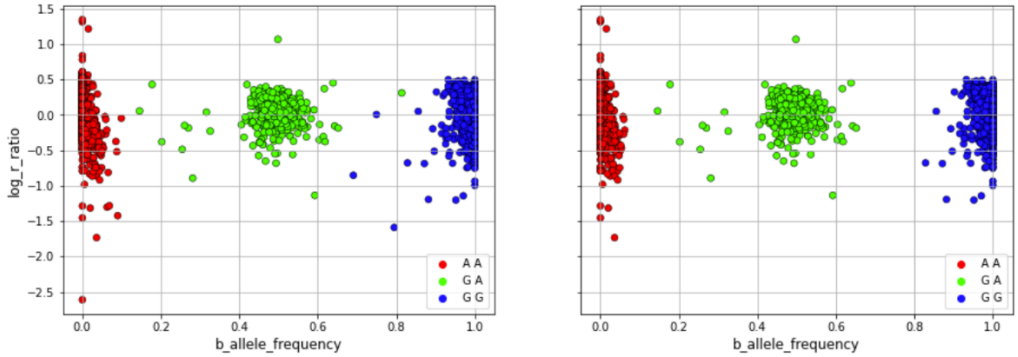
original data vs processed data with outliers removed for chrX_74859485



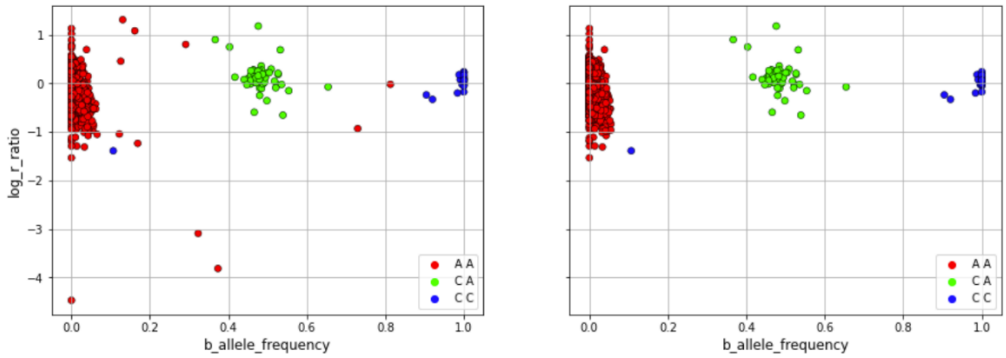
original data vs processed data with outliers removed for chrX_86333309



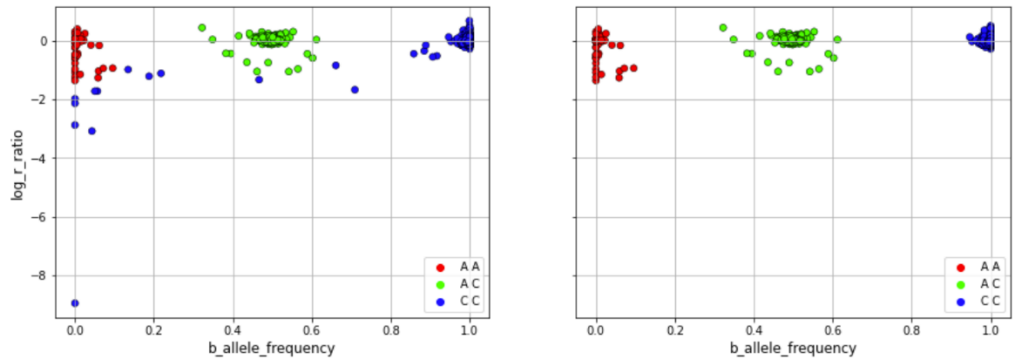
original data vs processed data with outliers removed for chrX_9396833



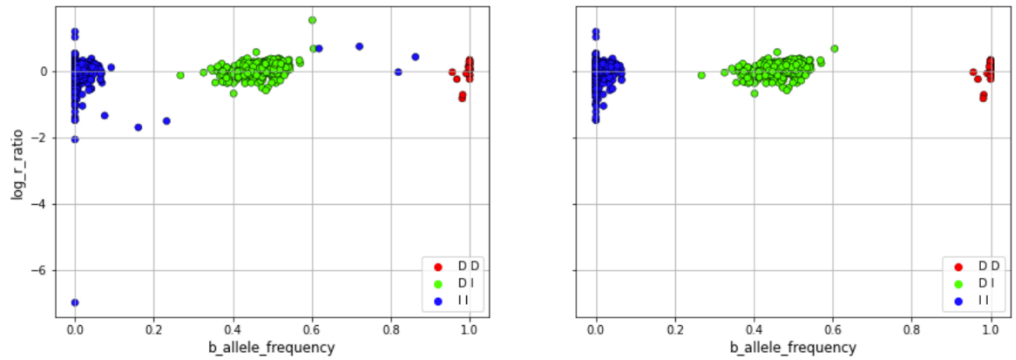
original data vs processed data with outliers removed for chrX_98838845



original data vs processed data with outliers removed for EMB_chr12_5417390a



original data vs processed data with outliers removed for EMB_chr14_13726596a



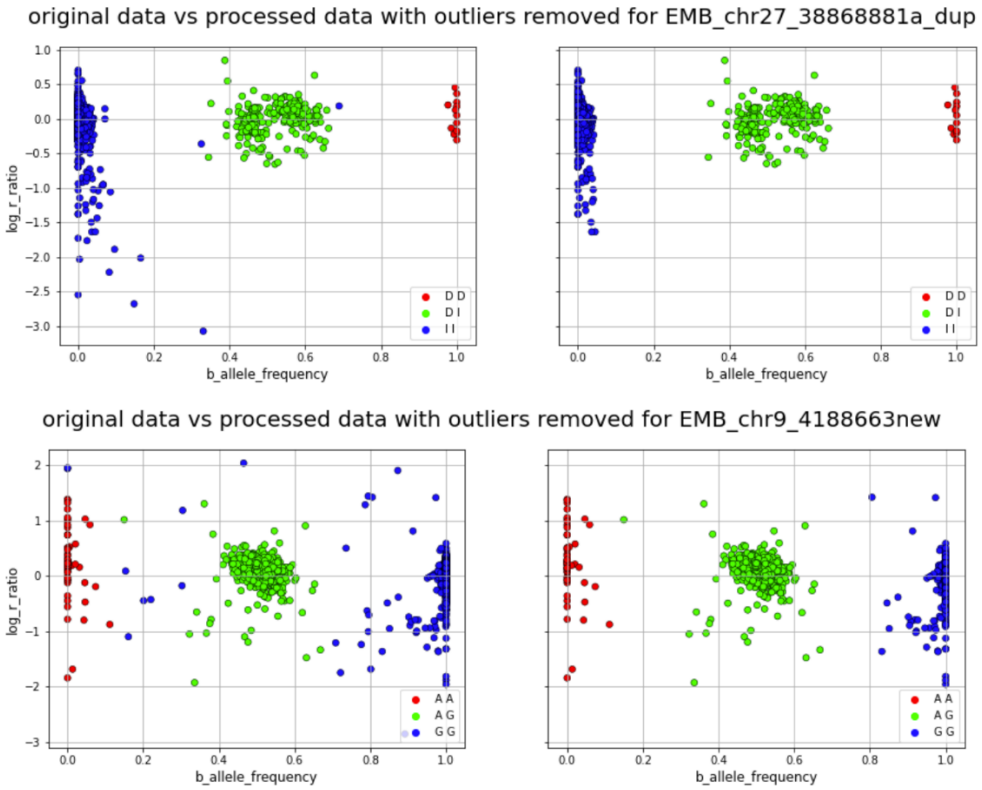


Fig. 18 A comparison of before/after outlier removals

From all the scatter plots above, the data becomes much cleaner after outliers are removed. Our next step is to re-run all four models based on the cleaned dataset.

7.10 Model Re-runs

After we re-run the models, below is a grand summary of F1-macro scores for all models for all probes.

Table 3: F1-macro scores for all models and probes using cleaned dataset

Model Name	Multi-nominal Logistic Regression (%)	k-NN (%)	SVM (%)	Random Forest (%)
chrX_10047369	100.00	100.00	100.00	100.00
chrX_10062232	100.00	100.00	100.00	100.00
chrX_104108957	100.00	100.00	100.00	100.00
chrX_1428432	100.00	100.00	100.00	100.00
chrX_15178873	99.42	98.29	98.29	98.29
chrX_30915190	99.04	99.53	99.52	99.04
chrX_56324668	100.00	100.00	100.00	100.00
chrX_6581086	100.00	100.00	100.00	100.00
chrX_74859485	100.00	100.00	100.00	100.00

chrX_86333309	100.00	100.00	100.00	100.00
chrX_9396833	100.00	100.00	100.00	100.00
chrX_98838845	100.00	100.00	100.00	100.00
EMB_chr12_5417390a	100.00	100.00	100.00	100.00
EMB_chr14_13726596a	100.00	100.00	100.00	100.00
EMB_chr27_38868881a_dup	100.00	100.00	100.00	100.00
EMB_chr9_4188663new	100.00	100.00	100.00	100.00

The effect of outlier removal is quite evident. All models achieve a 100% F1-macro scores except for test data from two probes. This confirms that outliers do have influences on our modeling, and we highly recommend to remove those outliers before attempting classification models.

7.11 Class Imbalance and Methods to Overcome

Class imbalance exists when the number of samples from one class is larger than other classes. The class that has a larger number of samples is called the majority class, while the class that has fewer samples are called the minority class. Take the probe chrx_56324668 for example; the GG class outnumbers both AA an AG. The context of GG is risk free, and the contexts for AA and AG are at risk and potential carrier of a disease, respectively. In the medical field, most cases are returned as negative (not detected), while only a few cases are detected as positive. Hence, this genotype classification problem, to some extent, reflects the real-world scenario.

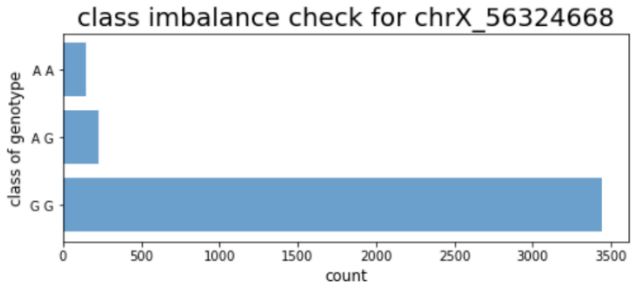
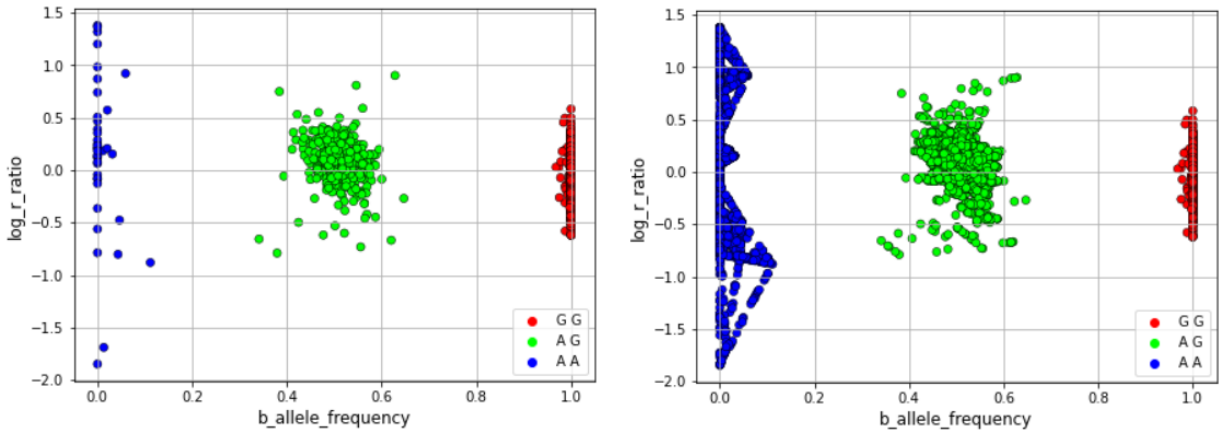


Fig. 19 Class imbalance in data from one probe

The class imbalance issue also keeps us cautious about choosing an appropriate model evaluation metric we have discussed above. In our case, we primarily avoid using Accuracy. For instance, if we have a total of 100 cases, where 99 cases are classified to be negative but one case is incorrectly classified as negative; the accuracy score will be 99%. This result is misleading because it's so costly that we fail to classify the one particular case correctly.

One commonly used method to tackle imbalanced data is to resample data. There are two resampling methods: Over-sampling and Under-sampling. Over-sampling creates synthetic samples for the minority class while under-sampling removes or merge samples in the majority class. In this study, we are going to explore one commonly used over-sampling method called SMOTE (Synthetic Minority Oversampling Technique), and one hybrid method that combines over-sampling and under-sampling at the same time.

SMOTE works by selecting random samples that are close in the feature space, drawing a line between the samples in the feature space and making a new sample as a point along that line. The package we use is SMOTE() from imblearn. A scatter plot for before/after over-sampling is shown below.



before SMOTE ({'G G': 2179, 'A G': 402, 'A A': 39}) after SMOTE ({'G G': 2179, 'A G': 2179, 'A A': 2179})

Fig. 20 Original data vs over-sampled data (SMOTE)

The over-sampled data is based on the training data without outliers. The scatter plot shows that the boundary among each genotype is clean; as we mentioned above, SMOTE works by interpolating a new point between two sample points. The effect is obvious for the AA class, where new data points are generated, and they fill in the 'gap'. We apply SMOTE to data for all probes, re-run all models and below is the F1-macro score matrix.

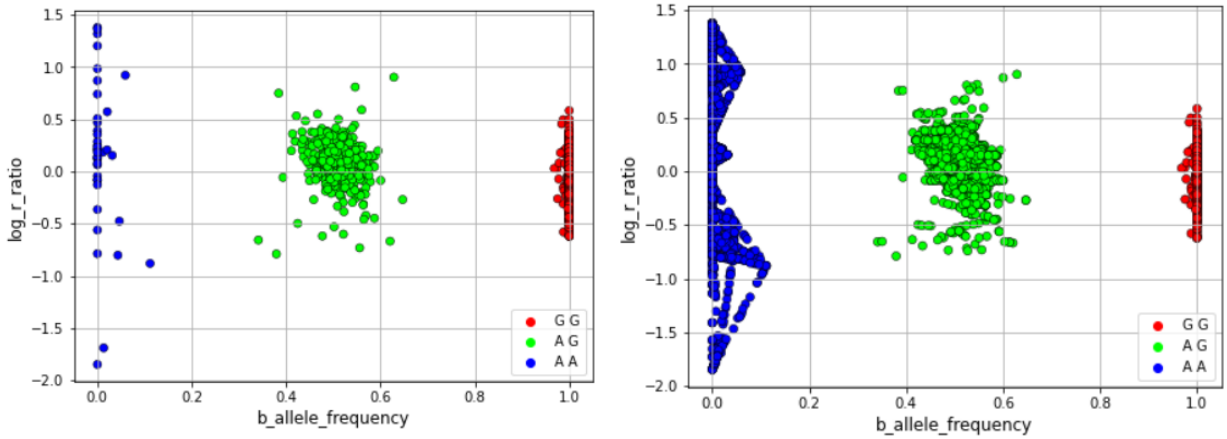
Table 4: F1-macro scores for all models and probes using balanced dataset with SMOTE method

Model Name	Multi-nominal Logistic Regression (%)	k-NN (%)	SVM (%)	Random Forest (%)
chrX_10047369	100.00	100.00	100.00	100.00
chrX_10062232	100.00	100.00	100.00	100.00
chrX_104108957	100.00	100.00	100.00	100.00
chrX_1428432	100.00	100.00	100.00	100.00
chrX_15178873	99.42	98.29	98.29	99.42
chrX_30915190	99.04	99.53	98.62	99.04
chrX_56324668	100.00	100.00	100.00	100.00
chrX_6581086	100.00	100.00	100.00	100.00
chrX_74859485	100.00	100.00	100.00	100.00
chrX_86333309	100.00	100.00	100.00	100.00
chrX_9396833	100.00	100.00	100.00	100.00
chrX_98838845	100.00	100.00	100.00	100.00
EMB_chr12_5417390a	100.00	100.00	100.00	100.00

EMB_chr14_13726596a	100.00	100.00	100.00	100.00
EMB_chr27_38868881a_dup	100.00	100.00	100.00	100.00
EMB_chr9_4188663new	100.00	100.00	100.00	100.00

Based on the results above, we don't see any improvement for F1-macro score, which is the same as what we get without SMOTE. We suspect each genotype class is already well separated in the original raw data. However, we still want to explore the hybrid method, which combines both over-sampling and under-sampling algorithms together.

The hybrid method is called SMOTE + Tomek Links. The goal of this method is to cleanup overlapping data points from different classes. First, SMOTE is applied to oversample the minority class, this may create synthetic points that are invading the spaces of other classes. Hence, what Tomek Links will do next is to identify nearest pairs of data points that are in different classes and remove one or both of points in the pairs that helps increase class separation near the decision boundaries. Below is a scatter plot comparing before/after SMOTETomek method, and the F-macro score matrix.



before SMOTETomek ({'G G': 2179, 'A G': 402, 'A A': 39}) after SMOTETomek ({'G G': 2179, 'A G': 2179, 'A A': 2179})

Fig. 21 Original data vs over-sampled data (SMOTETomek)

Table 5: F1-macro scores for all models and probes using balanced dataset with SMOTE method

Model Name	Multi-nominal Logistic Regression (%)	k-NN (%)	SVM (%)	Random Forest (%)
chrX_10047369	100.00	100.00	100.00	100.00
chrX_10062232	100.00	100.00	100.00	100.00
chrX_104108957	100.00	100.00	100.00	100.00
chrX_1428432	100.00	100.00	100.00	100.00
chrX_15178873	99.42	98.29	98.29	99.42
chrX_30915190	99.04	99.53	98.62	99.04
chrX_56324668	100.00	100.00	100.00	100.00
chrX_6581086	100.00	100.00	100.00	100.00

chrX_74859485	100.00	100.00	100.00	100.00
chrX_86333309	100.00	100.00	100.00	100.00
chrX_9396833	100.00	100.00	100.00	100.00
chrX_98838845	100.00	100.00	100.00	100.00
EMB_chr12_5417390a	100.00	100.00	100.00	100.00
EMB_chr14_13726596a	100.00	100.00	100.00	100.00
EMB_chr27_38868881a_dup	100.00	100.00	100.00	100.00
EMB_chr9_4188663new	100.00	100.00	100.00	100.00

We still don't see any difference between SMOTE and SMOTETomek. We still suspect that the original dataset with outliers removed has clear boundaries between each class, and data resampling may not show its power for this scenario.

8 Classification and Prediction for Health State based on Data Related to Chimerism

8.1 Introduction

So far, we explored 4 types of classification models to classify three genotypes for 16 probes. Now, we want to see if we apply the same methods to predict the health states based on the data related to chimerism.

8.2 Data Preparation

First, we filter data related to chimerism only in the raw genotype dataset. The 'SNP name' column should have string values starting with 'chrX'.

SNP Name	b_allele_frequency	log_r_ratio	short_id
chrX_9396833	1.000000	-0.135639	1544946901181039
chrX_10047369	1.000000	-0.167039	1544946901181039
chrX_10062232	0.000000	-0.101803	1544946901181039
chrX_104108957	0.001059	-0.225807	1544946901181039

Fig. 22 Filter data related to chimerism in raw genotype dataset

Next, in the health state dataset, we want to filter the 'health_id' column whose values are equal to 999999.

health_id	health_state	health_state_full_text	health_state_simplified	short_id
999999	20.0	chimeric	at risk	49387287904546034
999999	2.0	clear	clear	213225478192960038
999999	2.0	clear	clear	254441560615002658
999999	2.0	clear	clear	337507332821811584

Fig. 23 Filter data related to chimerism in health state dataset

Finally, we combine these two filtered datasets based on 'short_id' column.

SNP Name	b_allele_frequency	log_r_ratio	short_id	Unnamed: 0	health_id	health_state	health_state_full_text	health_state_simplified
chrX_10047369	0.996027	0.024893	49387287904546034	65	999999	20.0	chimeric	at risk
chrX_104108957	0.001052	-0.175677	5023911345393777239	90	999999	20.0	chimeric	at risk
chrX_10062232	0.000000	-0.068397	5023911345393777239	90	999999	20.0	chimeric	at risk
chrX_9396833	0.518181	0.157364	5004169003734631502	44	999999	20.0	chimeric	at risk

Fig. 24 Combined dataset

8.3 Exploratory Data Analysis

Below is a bar chart which shows that the class imbalance exists for this filtered dataset. Besides, a scatterplot shows how each class is distributed based on the b_allele_frequency and log_r_ratio.

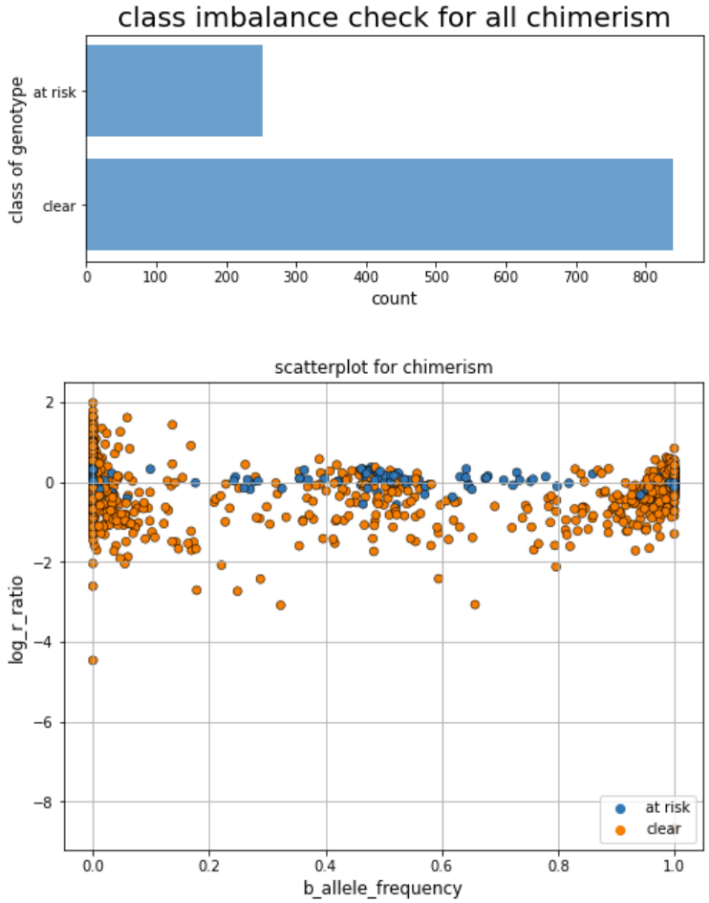
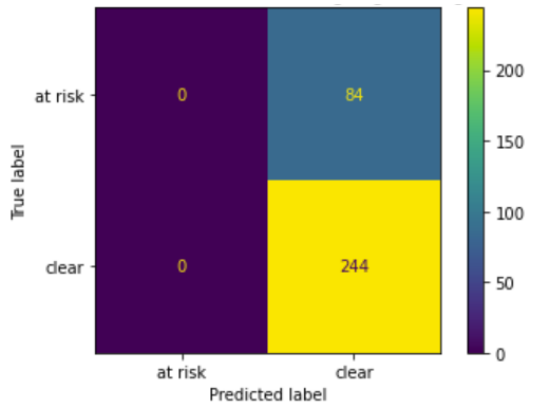


Fig. 25 Bar chart and scatterplot for raw data related to chimerism

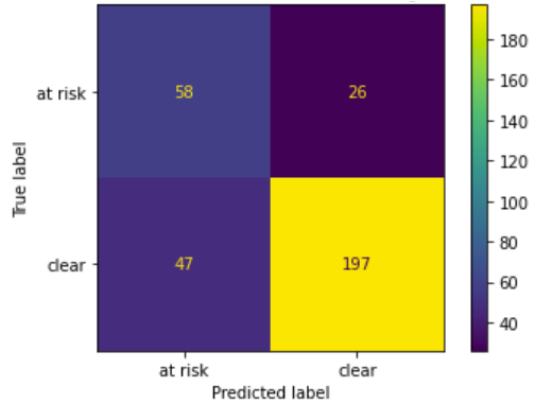
By looking at the scatterplot, we foresee it can pose a challenge for classification models. The reason is the data from two classes are mingled together, which can be hard to draw a decision boundary between them.

8.4 Model Evaluation

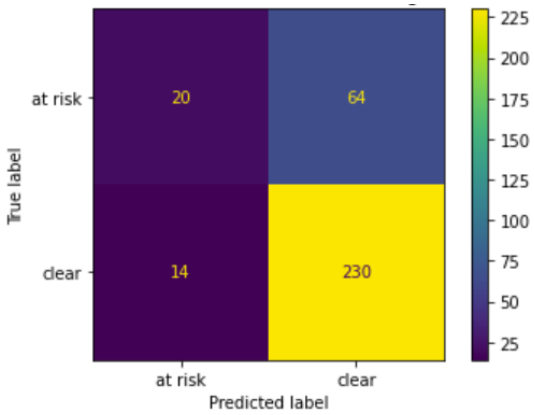
We apply four classifications methods we used for predicting the genotypes above, and below are the confusion matrices and boundary plots for all four models.



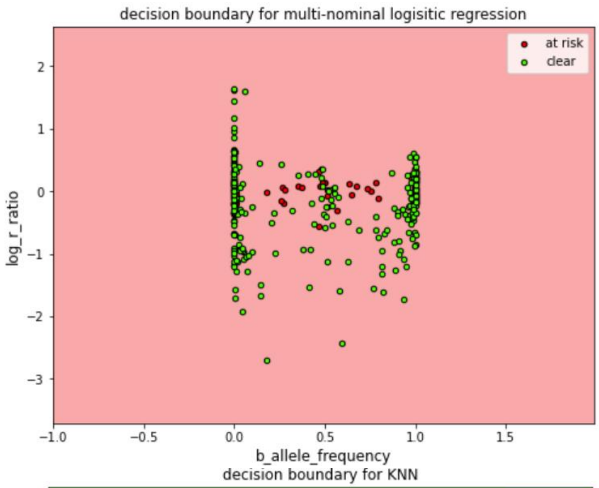
f1 macro score for testing dataset with tuning is: 42.66%

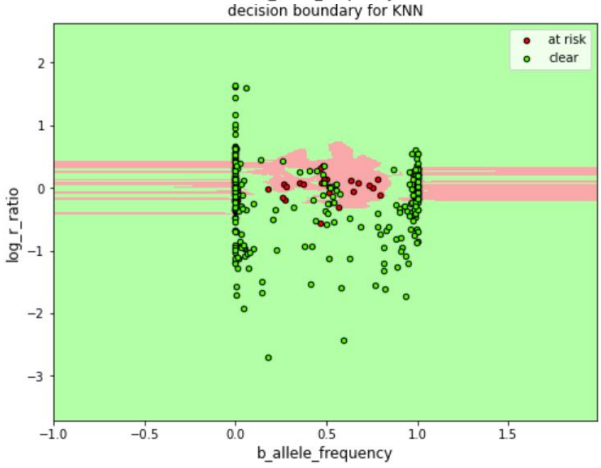


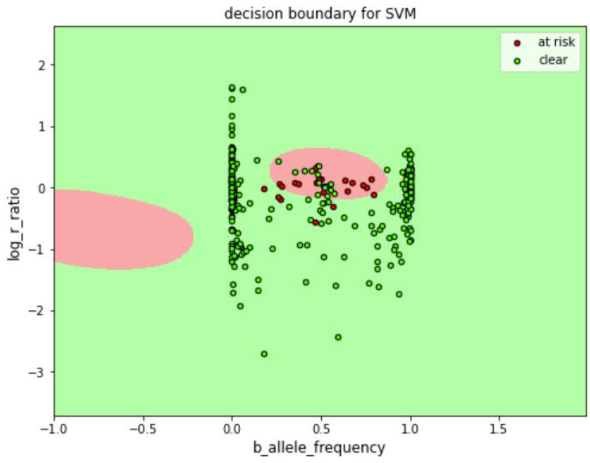
f1 macro score for testing dataset with tuning is: 72.87%

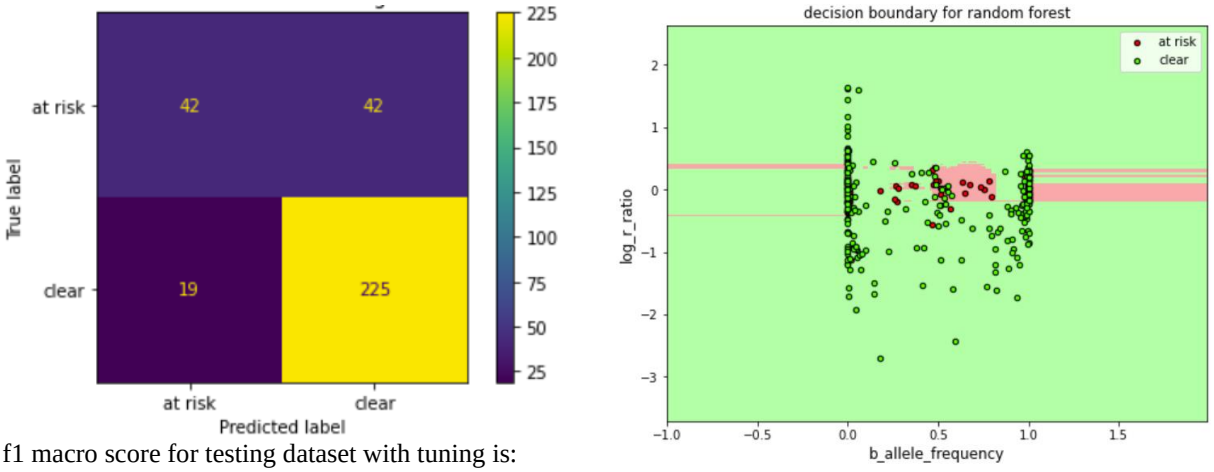


f1 macro score for testing dataset with tuning is: 59.70%









f1 macro score for testing dataset with tuning is: 73.00%

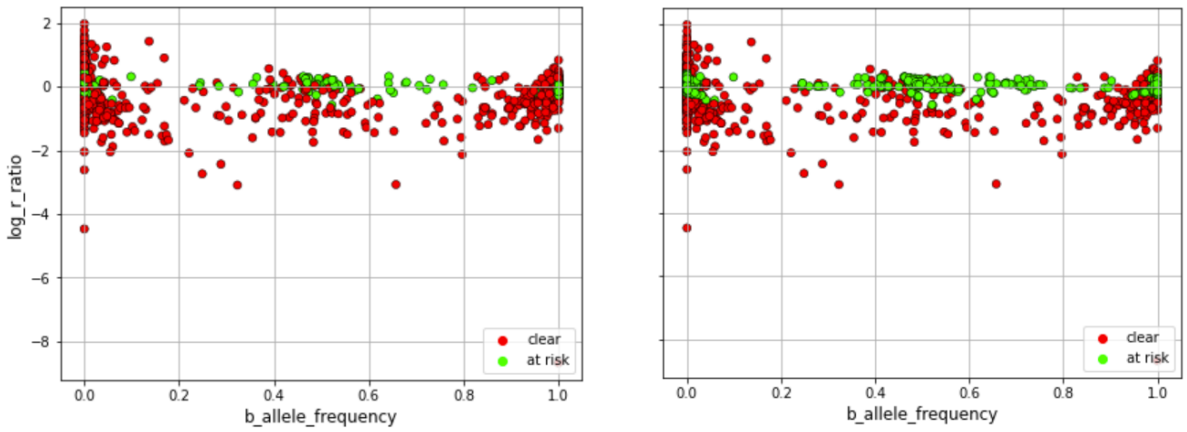
Fig. 26 Model evaluation results for four classification models

Table 6: F1-macro scores for four models			
Multi-nominal Logistic Regression (%)	**k-NN (%)**	**SVM (%)**	**Random Forest (%)**
42.66	72.87	59.70	73.00

Compare the above results with the ones we obtained from genotype prediction, the F1-macro scores are significantly lower. Also, we don't see a clear separation between two classes in all boundary plots; particularly with the multi-nominal logistic regression being the worst. Hence, while Random Forest has the highest F1-macro score among all models, we still need to be cautious to use it to predict a new value.

8.5 Class Balancing and Model Re-run

In the previous section, we discussed about class imbalance issue in the raw genotype dataset; the same issue still exists for raw data related to chimerism. So, we decide to apply class balance method and re-run the models to see if there are any differences or improvements. Below is a scatter plot comparing before and after class balance using SMOTETomek method.

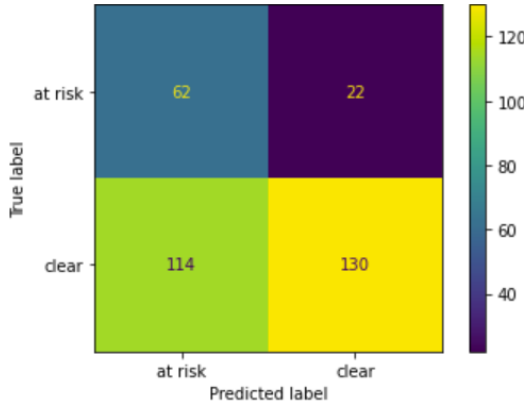


before SMOTETomek ({'clear': 596, 'at risk': 168}) after SMOTETomek ({'clear': 541, 'at risk': 541})

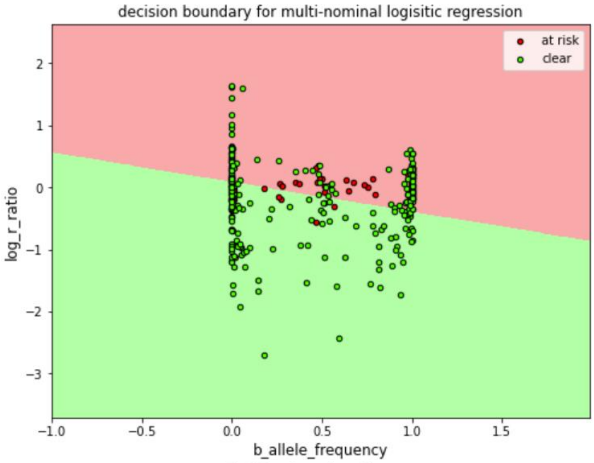
Fig. 27 Before and after class balance

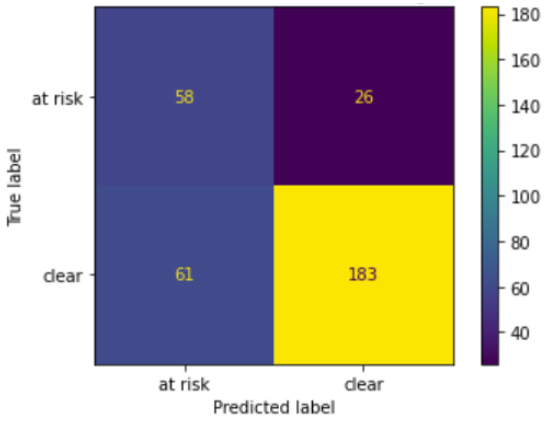
Because the two classes are mixed together, SMOTETomek shows its power as it tires to reduce the samples from the majority class while artificially increasing the samples in the minority class. We can see the number of samples in class labeled with ‘clear’ are reduced while the number of samples in class labeled as ‘at risk’ are increased.

Below are the confusion matrices and boundary plots for all four models based on balanced data.

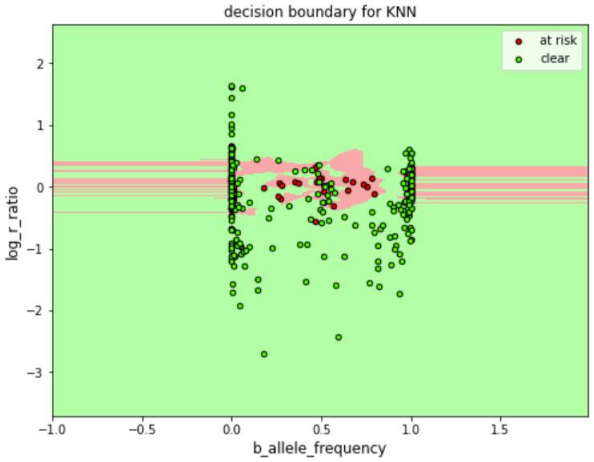


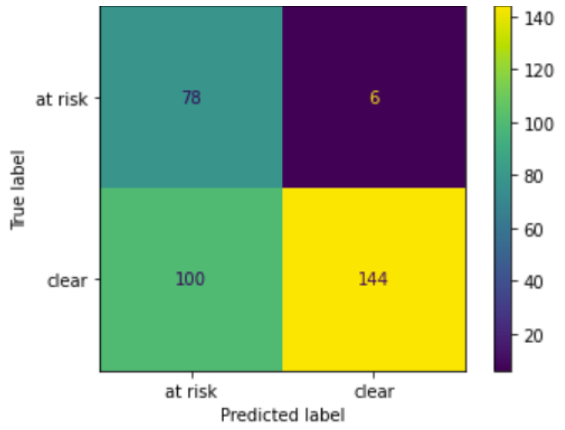
f1 macro score for testing dataset with tuning is: 56.67%



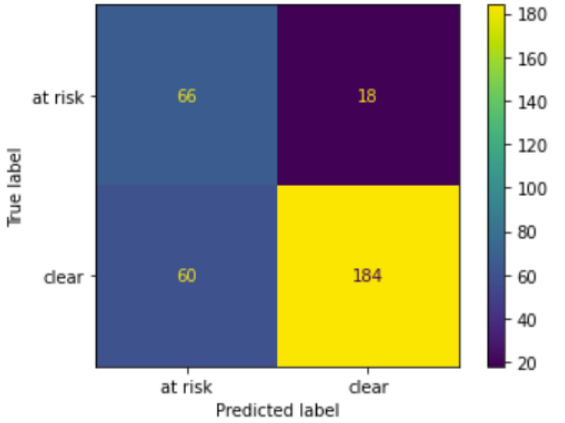


f1 macro score for testing dataset with tuning is: 68.97%

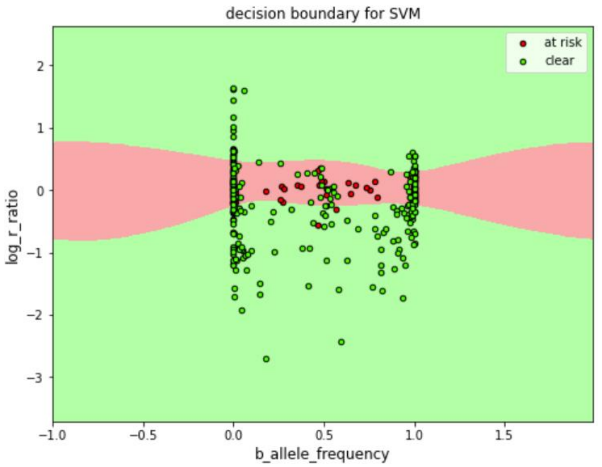




f1 macro score for testing dataset with tuning is: 66.32%



1 macro score for testing dataset with tuning is: 72.68%



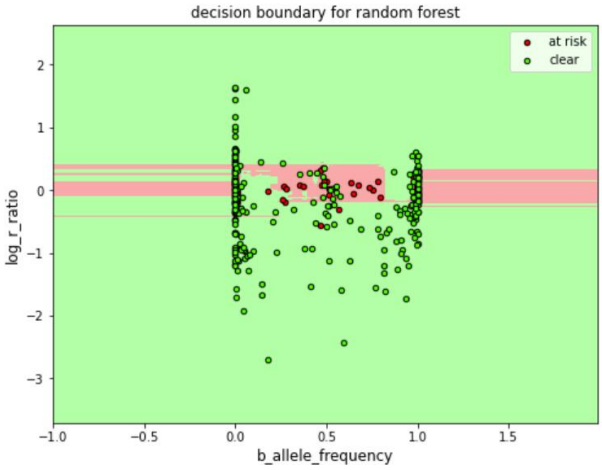


Fig. 28 Model evaluation results for four classification models with balanced data

Table 7: F1-macro scores for four models using balanced data

Data Input	Multi-nominal Logistic Regression (%)	k-NN (%)	SVM (%)	Random Forest (%)
Imbalanced data	42.66	72.87	59.70	73.00
Balanced data	57.20	68.97	66.32	72.68

After the data balancing method is applied, the boundaries between two classes are more pronounced, and f1-macro scores also show some improvement as well, with Random Forest still outperforms other models and Multi-nominal Logistic Regression is still the worst. However, considering data samples from two classes mix so closely, we would be cautious about using Random Forest to predict a class for a new pair of raw data.

8.6 Possible Reasons for Poor Model Performance

The study we did above is certainly a challenge; in contrast to predicting genotypes for individual probe, we tried to predict a health state based on all data combined that is related to chimerism. We used a scatterplot to demonstrate how the two class, ‘at risk’ and ‘risk free’, are

mixed. Let's make additional scatterplots that separate the two classes. (For a complete list of scatterplots, please refer to Jupyter notebook)

Fig. 29 Scatterplot for dogs identified at risk

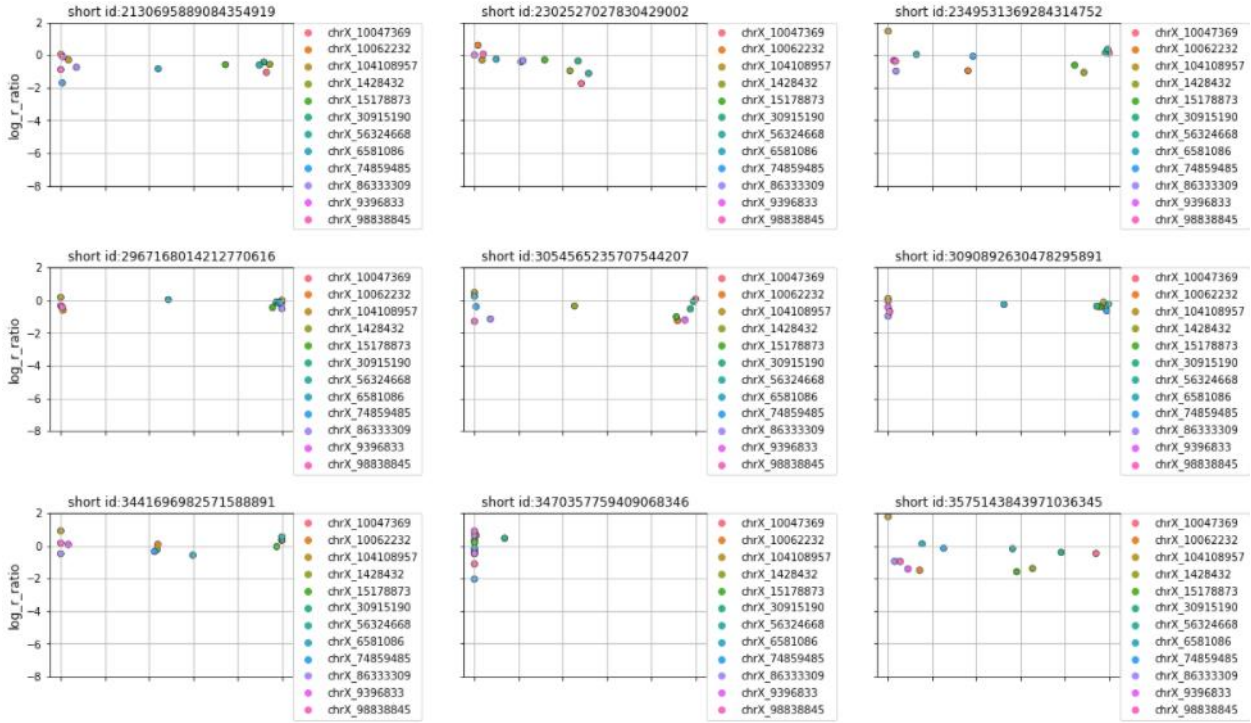


Fig. 30 Scatterplot for dogs identified as risk free

The first scatterplot shows dogs identified as ‘at risk’ based on a group of probes related to chimerism, and the second scatterplot shows dogs identified as ‘risk free’ based on the same group of probes. Now we can see why the previous classification models didn’t perform well on those raw training data. The raw data for these two classes are overlapping with each other, meaning the raw values of b_allele_frequency and log_r_ratio for ‘at risk’ and ‘risk free’ are close to each other. There is no distinct discrepancy between these two classes. That’s why we still give a warning about the Random Forest model even though it has above 70% F1-macro score, because the raw data itself is not too good for building classification models.

9 Conclusion

With different model analysis, we were able to classify the data to different classes “at risk” and “risk free” based on the probes data in relation to chimerism. We also did notice some overlap on the classes while examining the data.

A priori knowledge to draw some inferences based of on the raw values of b_allele_frequency was helpful and can be categorized by the in-depth study. Application of the more advanced classification and different hybrid combinations model approach to a larger dataset could be the next steps for advance study and future work. With hybrid model complexity, the problem like overfitting will be expected to emerge after application.

References

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-223