**Introduction**

Feature selection techniques are an important aspect of machine learning, especially in the field of genomics, where it is common to have hundreds, or even thousands, of predictor variables. Using feature selection reduces the complexity of the model, enables the machine learning algorithm to train faster, and reduces overfitting. There are various methods that one can choose when using feature selection techniques. The two types of methods were implemented here are referred to as *Wrapper Methods* and *Embedded Methods*.

In wrapper methods, the objective is to use a subset of features and train a model. Features are then added or removed based on inference of a particular performance metric. The wrapper methods used here were the forward selection and the backward elimination. Forward selection is an iterative method in which we start with having no features in the model. In each iteration, a new feature is added which most improves the model until adding a feature does not improve the model, at which point the algorithm terminates. In backward elimination, the algorithm works similar to forward selection except in the opposite direction. We first start with all features and then remove the least significant one which improves the performance of the model. This process is repeated until the removal of a feature does not result in an improved model.

Embedded Methods are implemented by algorithms that have their own built-in feature selection methods. The embedded method used is the *Boruta* function from the Boruta package in R. Where most wrapper methods tackle the problem known as “minimal-optimal,” the Boruta package attempts to identify all attributes which are in some circumstances relevant for classification, the so-called “all-relevant” problem.

**Methods**

Publicly available topologically associating domain data was obtained from GEO with accession GSE53525. The domain data was in the form of two-column genomic coordinates from the GM12878 cell line, with each coordinate representing a TAD boundary. The coordinates were concatenated into a long vector and were sorted. The genome was then binned into a series of 1kb intervals (500 bases on either side of the boundary point) and an indicator vector Y was created based on whether there was a TAD boundary in the 1kb interval (Y=1) or not (Y=0). There were 247632 bins that were created, of which 1629 were classified as TAD boundaries.

Various genomic annotations were obtained from the GM12878 cell line of the ENCODE Project. These annotations made up the feature space **X**, including p predictor variables ({X₁,...,X­p}). Two types of predictors were created for each annotation. First, a binary variable was created denoting whether or not the 1kb bin overlapped with the annotation or not. Second, a continuous variable was calculated for each annotation as the distance to the nearest coordinate from the 1kb bin.

Prior to implementing the variable selection methods, any variables with near zero variance were removed. Given how unbalanced the data is, the concern is that these predictors may become zero- or near zero-variance predictors when the data is split into cross-validation samples. This can cause the machine learning algorithms to be unstable.

For the implementation of the wrapper methods (forward and backward selection), the dataset was first reduced in order to created balanced classifications. Two methods were used: random sampling and SMOTE. For random sampling, the majority class was randomly sampled to match the number of minority classes. For SMOTE, both minority oversampling and majority undersampling are used to create a balanced dataset. Therefore, a total of 4 algorithms were implemented, two from random sampling and two from SMOTE (forward % backward). For each algorithm, 10-fold cross-validation was used.

The Boruta algorithm is a wrapper built around the random forest classification algorithm implemented in the R package randomForest. Firstly, it adds randomness to the given data set by creating shuffled copies of all features. Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important. At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z score than the maximum Z score of its shadow features) and constantly removes features which are deemed highly unimportant. Finally, the algorithm stops either when all features gets confirmed or rejected or it reaches a specified limit of random forest runs.

Once the variables were chosen from each variable selection technique, an Elastic Net model was implemented to assess the performance. The algorithms were compared based on AUC and the variables that were chosen were presented in a table. All analysis was performed in R version 3.4.2.

**Results**

The data consisted of 106 features, they are listed in Table 1 (a).Likewise, Table 1 (b) presents a list of the variables that were classified as having near zero variance, and thus were removed. The final data set used in the variable selection methods consisted of 65 variables. Table 2 presents the list of features that remained after each variable selection technique.

From Figure 1 (a) and (b), we see that the forward and backward selection methods on the randomly sampled data performed slightly better than the rest, with the forward selection specifically performing the best overall with an AUC of 0.8061. The numbers above the bars indicate how many features remained in the dataset after performing the specific variable selection. The forward and backward techniques yielded 21 and 22 features respectively. There was a significant overlap among the features chosen with a total of 18 features in common between the two methods.

Figure 2 shows the bar graph of feature importance created from the Boruta function. The yellow bars indicate tentative features that the algorithm was unable to distinguish from the shadow features that were created by inducing randomness in the data. The green bars indicate significantly important features. The tentative features were kept for downstream analysis resulting in a total of 20 features. The Boruta data slightly underperformed compared to the rest with an AUC of 0.804.

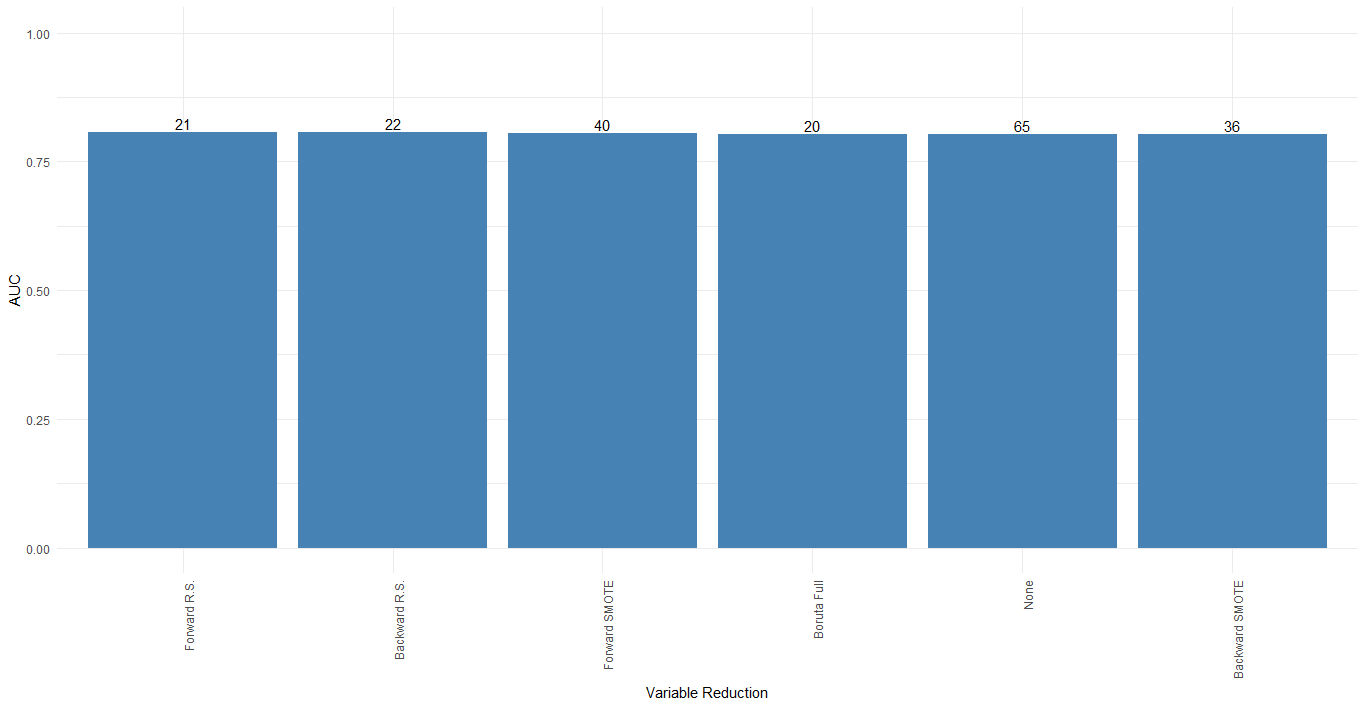
After, finalizing each dataset using the various variable selection techniques, an Elastic Net model was performed. Figure 3 shows the variable importance of each dataset used. The top right graph in red represents the data from the forward selection. We see that the most important features recognized by the elastic net aligned with previous results regarding topologically associating domain data. That is, the insulator and CTCF variables are among the top 3 most important variables. The importance plots from the other datasets show conflicting results.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variable Name** | | |
| **Binary** | A | inversion | DNA |
|  | B | mobile\_element\_insertion | line |
|  | complex | novel\_sequence\_insertion | low\_complexity |
|  | deletion | sequence\_alteration | LTR |
|  | duplication | tandem\_duplication | other |
|  | gain\_loss | Gerp | RC |
|  | insertion | rRNA | RNA |
|  | rRNA | tRNA | TxnElongation |
|  | satellite | unknown | WeakTxn |
|  | scRNA | UCNE | Repressed |
|  | simple\_repeat | VMR | Heterochromlo |
|  | SINE | WeakPromoter | RepetitiveCNV14 |
|  | snRNA | PoisedPromoter | RepetitiveCNV15 |
|  | srpRNA | StrongEnhancer4 | ActivePromoter |
|  | StrongEnhancer5 | TxnTransition | R |
|  | WeakEnhancer6 | CTCF | T |
|  | WeakEnhancer7 | E | TSS |
|  | Insulator | PF | WE |
| **Continuous** | A | other | ActivePromoter |
|  | B | RC | WeakPromoter |
|  | complex | satellite | PoisedPromoter |
|  | deletion | simple\_repeat | StrongEnhancer4 |
|  | duplication | SINE | StrongEnhancer5 |
|  | gain\_loss | srpRNA | WeakEnhancer6 |
|  | insertion | unknown | WeakEnhancer7 |
|  | inversion | se\_GM12878 | Insulator |
|  | mobile\_element\_insertion | se\_GM12878 | TxnTransition |
|  | novel\_sequence\_insertion | UCNE | CTCF |
|  | sequence\_alteration | UCNE\_score | E |
|  | tandem\_duplication | VMR | PF |
|  | gerp | TxnElongation | R |
|  | gerp\_score | WeakTxn | T |
|  | DNA | Repressed | TSS |
|  | line | Heterochromlo | WE |
|  | low\_complexity | RepetitiveCNV14 |  |
|  | LTR | RepetitiveCNV15 |  |

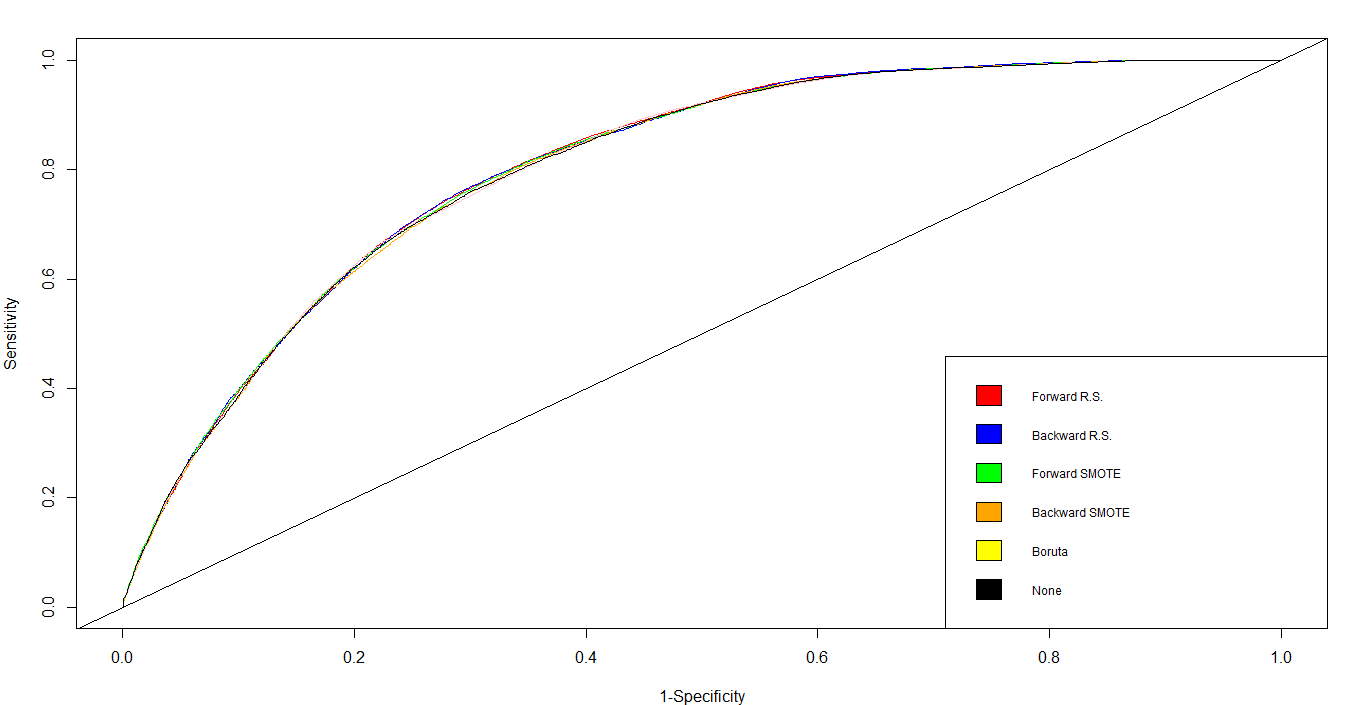
**Table 1 (a)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variable Name** | | |
| **Binary** | complex | inversion | DNA |
|  | rRNA | mobile\_element\_insertion | WeakPromoter |
|  | satellite | novel\_sequence\_insertion | low\_complexity |
|  | scRNA | sequence\_alteration | PoisedPromoter |
|  | simple\_repeat | tandem\_duplication | other |
|  | snRNA | rRNA | RC |
|  | srpRNA | tRNA | RNA |
|  | WeakEnhancer6 | unknown | TxnTransition |
|  | E | UCNE | CTCF |
|  | TSS | PF | WE |
|  | Insulator |  |  |
| **Continuous** | A | gerp\_score | UCNE\_score |
|  | B |  |  |

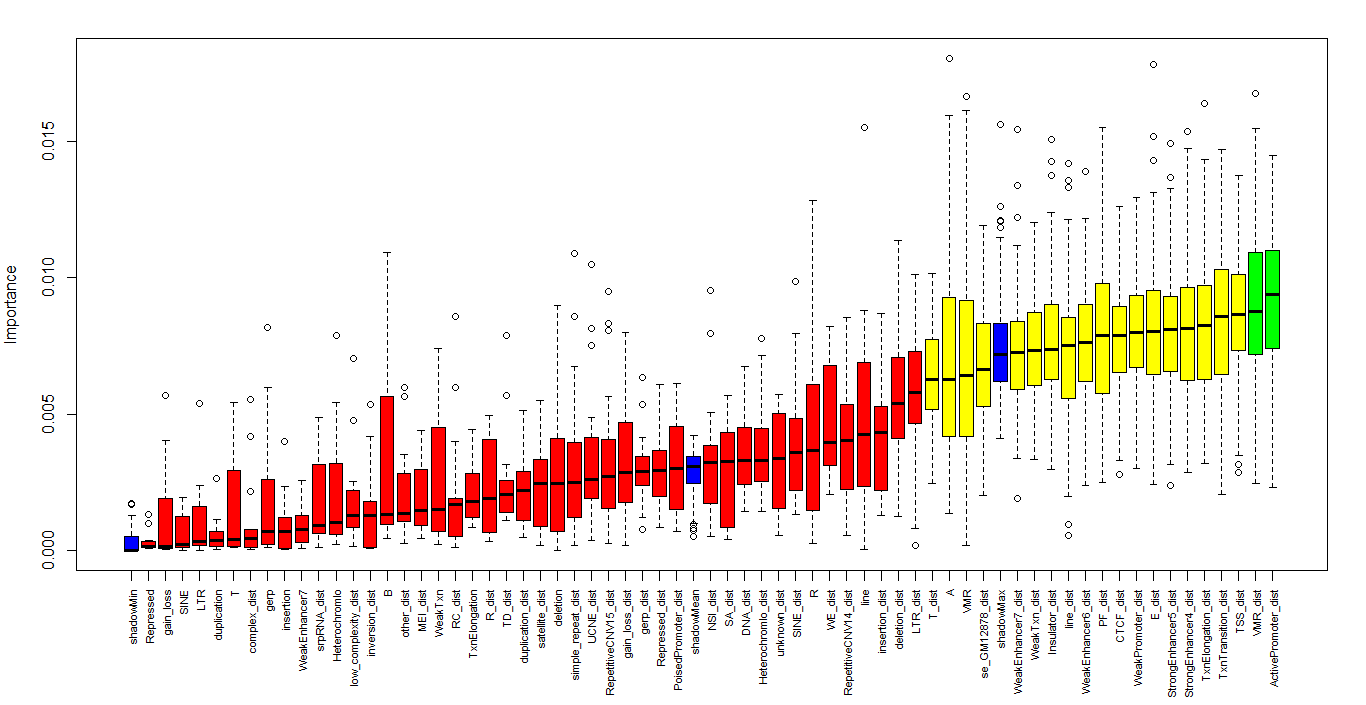
**Table 1 (b)**



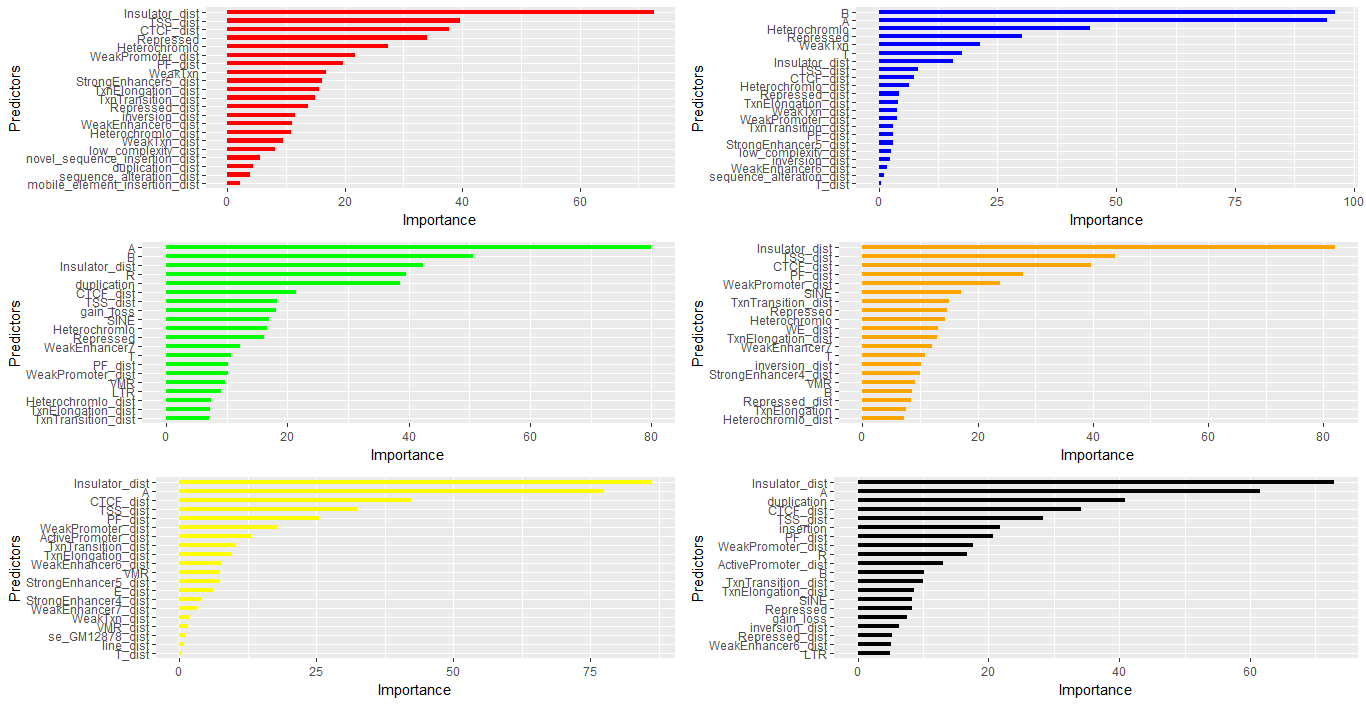
**Figure 1 (a).**



**Figure 1 (b)**



**Figure 2.**



**Figure 3.**

**Discussion**

Variable selection can be a useful part of data analysis in the field of genomics. It can allow for better interpretability and quicker results. Here we found that using forward selection prior to performing an Elastic Net showed better results than other techniques as well as not performing any variable selection.