Predictive modeling using genomic annotations

Pitfalls and Recommendations

Spiro Stilianoudakis

August 9, 2018

Contents

INTRODUCTION

The advent of various genome-wide technologies, such as high-throughput conformation capture (Hi-C), have revealed how the spatial organization of the human genome may affect several epigenetic functions (Aiden et al.). Analyses have shown that the genome is tightly compacted into distinct compartments. There exist regions within these compartments that are highly conserved and self-interacting, and termed topologically associating domains (TADs) (Dixon et al.). Evidence suggests that regulatory elements and genes tend to interact more frequently within the same TAD (Symmons et al.). This suggests that the boundaries of TADs may play a role in restricting the function of elements such as enhancers, thereby impacting the transcription of genes.

More recently, it has been discovered that insulators have a primary role in orchestrating the topological arrangement of higher-order chromatin architecture (Phillips-Cremins et al). Insulators are multi-faceted

regulatory sequences that moderate a variety of genomic processes including activation, repression, and enhancer blocking. Specifically, the insulator binding protein CTCF has been found to be enriched at the boundary sequences of topologically associating domains in human cells and may therefore act as a mediator of long range chromatin contacts (Zuin et al.). Likewise, it was found that DNase I-hypersensitive sites were enriched at the boundaries of domains and correlated with the CTCF signals (Hong et al) [??? They identify more]. On the other hand, it is unclear how other types of regulatory elements such as histone modifications, which are associated with transcriptional initiation and open chromatin structure, are related to TAD boundaries.

The distinct patterns of some of these different proteins and functional elements point toward the opportunity of computational approaches in predicting the development of TAD boundaries. However, due to the size of Hi-C data and the abundance of available genomic features, few methods have been developed to study the role of specific sets of these features on the folding of chromosomes. Furthermore, many widely used methods ignore key characteristics of the data that may hinder the performance of certain parametric models [??? Like what? It may be best to devote a paragraph clearly describing them. Then, describe the Mourad and other approaches]. One such group have proposed a multiple logistic (MLR) model used to identify the most influential proteins with regards to TAD boundaries (Mourad et al.). They also provide an MLR model with LASSO estimated coefficients that they believe is better suited for predictors that may be correlated. However, key aspects of the data were ignored with each of these models, such as the sparsity of domain borders throughout the genome. Likewise, data pre-processing techniques such as normalization, the elimination of low variance predictors, and variable selection techniques were not considered. Furthermore, due to the large number of genomic features that can be considered, accounting for the relationship among features with interaction effects becomes computationally infeasible.

Data pre-processing techniques such as normalization and the elimination of low variance predictors are an integral part of model fitting with regards to genomic data [??? Clarity. Define 1) the problem, and 2) ways to solve it, which is modeling. Then, for modeling, describe problems connecting genomics+machine learning terms]. Continuous predictors tend to be highly shewed. Therefore, a common practice is to perform a log base 2 transformation. It is also sometimes necessary to standardize continuous predictors, especially if they are on different scales. It is unclear, however, which combination of these two concepts improves prediction. Additionally, binary predictors may only be concordant with the outcome of interest in very rare cases. This contributes to predictors with near-zero variances. Eliminating these predictors prior to model fitting can reduce both the noise and the computational speed of the model. We use the nearZeroVar function provided in the caret package to determine which predictors have near-zero variance [??? This sentence is Methods].

TADs can be up to a million base pairs in length and therefore the boundaries are sparse throughout the genome. Sparse data can create heavily imbalanced classes and may affect prediction performance of classification algorithms. There are many techniques that can be used to more evenly balance such data. These include oversampling the minority class, under sampling the majority class, and some combination of both. One such technique is referred to as SMOTE, which stands for Synthetic Minority Over-sampling TEchnique. SMOTE is a function in the DmWR package in R and incorporates both under-sampling and over-sampling. It has been shown that SMOTE out performs simple under-sampling of the majority class for some machine learning algorithms (Nitesh et al). We instead propose a method of taking multiple bootstrap under-samples from the majority class and then aggregating performance metrics by taking the average across all iterations. We then compare this method to SMOTE.

Modeling genomic data often involves a large number of predictors as well. As a result, different machine learning models can benefit from variable selection techniques to reduce the feature space, and thereby improve computational speed. There are several known selection techniques in the field of machine learning including forward, backward, and stepwise selection. These are known as wrapper methods because they measure the usefulness of a subset of features by actually training a model on it. Additionally, we incorporate recursive feature elimination as a method of variable reduction and compare it to the rest.

Furthermore, the model proposed by Mourad et al. falls short of addressing issues such as cross-validation and variable importance, which can be handled more efficiently through ensemble models like random forests and gradient boosting machines. Therefore, in addition to proposing a novel pipeline that addresses irregularities in the data mentioned above, we also apply a random forest model in order to find the key molecular drivers most associated with TAD boundaries. A random forest was chosen because of its ability to handle potentially correlated variables as well as its inability to overfit. We then compare the performance of the random forest model to the models proposed by Mourad et al.

METHODS

THE DATA

Publicly available topologically associating domain data was obtained from GEO with accession GSE53525 for the GM12878 and K562 cell lines. The domain data was constructed from in situ Hi-C contact matrices at a 5kb and 10 kb resolution for the GM12878 and k562 cell lines respectively. The Arrowhead algorithm was used to identify TAD boundaries for a given contact matrix (Rao et. al). Identified TAD boundaries

were represented by their genomic coordinates (hg19 human genome assembly), including chromosome, start and end coordinates. The start and end coordinates were concatenated, sorted, and unique coordinates representing unique TAD boundaries were obtained [??? Check for accuracy]. Next, the genome was binned into a series of 1kb intervals (flanked by 500 bases on either side of the boundary point) and an indicator vector Y was created based on whether there was a TAD boundary located in the 1kb interval (Y=1) or not (Y=0) [??? Should it be simplified - just consider boundaries to be flanked?].

GENOMIC ANNOTATIONS

Annotation data was obtained from the Encyclopedia of DNA Elements (ENCODE) Consortium [??? We need a table. Check Hong 2017 supplemental, there may be something else that I don't have in GenomeRunner database. Also, check iGD data http://big.databio.org/igd/]. These annotations consisted of genomic coordinates and were used to build the feature space, X = {X1, . . ., Xp}, of the subsequent models. The data consisted of genomic coordinates and chromosomal information. For each genomic feature, an indicator variable (1 for yes; 0 for no) was created based on whether the coordinates of the annotation overlapped with the coordinates representing a flanked TAD boundary. Likewise, a continuous variable representing the distance to the center of the nearest flanked TAD boundary was included [??? Measured in?]. Genomic features that were located inside of a flanked TAD boundary were given a distance of 0. Figure X presents an illustration for how the feature space was constructed.

DATA PIPELINE

We developed a pipeline that systematically evaluates the problems associated with genomic data in predictive modeling settings. At each stage in the pipeline, Elastic-Net models were performed and performance metrics such as ROC curves and subsequent AUCs were evaluated. The Elastic-Net model was used here because of its ability to be tuned and its computational speed relative to other machine learning algorithms. For each elastic-net model the mixing parameter, alpha, was tuned on a grid of five values (0.1, 0.325, 0.550, 0.775, and 1.0). A sequence of values for the regularization parameter, lambda, was automatically provided by the CARET package in R. The algorithm then chose the optimal combination of alpha and lambda, and the model was evaluated. [??? Unclear, is it cross-validation tuning or what?]

Prior to initiating the pipeline, predictors with near-zero variance were removed. To identify said predictors, the nearZeroVar function in the CARET package was used. The function considers two metrics. First, the ratio of the most frequent value over the second most frequent value was calculated. Ideally, this ratio would

be close to one for well-behaved predictors. Next, the number of unique values divided by the total number of samples multiplied by 100 was calculated. Here, if the frequency ratio was greater than 0.99 and the percentage of unique values was less than 0.001, the predictor was flagged as having near-zero variance and removed.

CLASS IMBALANCE

To assess the class imbalance problem due to the sparcity of domains throughout the genome, two methods were used. The first consisted of incorporating the SMOTE command from the DmWR package in R. SMOTE stands for Synthetic Minority Over-sampling Technique. The SMOTE algorithm incorporates both under-sampling (percent under) of the majority class and over-sampling (percent over) of the minority class. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors (here, k=5). Then, the majority class is under sampled by taking a random sample that matches some pre-specified percentage of the updated minority class. For example, consider a data set with 700 observations, where there are 35 minority classes and 665 majority classes. Now, when using SMOTE, consider a 200 percent over sample of the minority class and a 100 percent under sample of the majority class. This means that two synthetic observations will be created for every minority class (35+70=105) and 1 majority class will be randomly chosen for every new minority class created (70), resulting in a classification of 70/105 (majority/minority). Now consider, a 200/200 combination. The new minority class is still 105 (70 new synthetic observations created), but now twice as many majority classes are randomly chosen (70*2=140), resulting in a 140/105 classification. It is important to note that using a combination of 100/200 will always result in perfectly balanced classes. A total of eight different combinations of percent over and undersampling were used which included: 100/200, 200/200, 300/200, 400/200, 100/300, 200/300, 300/300, 400/300. Here, the 100/200 combination creates a completely balanced dataset as mentioned before, while the others create an unbalanced dataset with the "No" class as the majority to reflect the original data. The second method involved iteratively taking 100 bootstrap samples of the majority class to match the number in the minority class. For each sample, an elastic-net model was tuned and evaluated. Then row wise means of the sensitivities, specificities, and AUCs of each of the 100 bootstrap samples were taken to evaluate the models as a whole. The SMOTE combination with the best performance was then compared to the bootstrap method. ROC curves were plotted and corresponding AUCs were presented in a table.

NORMALIZATION

In order to correct for the skewness of the distances [??? Should be defined in the introduction], elastic-net models were evaluated based on a combination of log transformed and/or standardized predictors. A total of four elastic net models were performed including: with log transformation and standardization, without log transformation and with standardization, with log transformation and without standardization, and without log transformation and without standardization. One hundred bootstrap samples were taken from the majority class to create a balanced classification data. ROC curves were plotted and corresponding AUCs were presented in a table. Likewise, we looked at the variable importance quality of the models.

VARIABLE SELECTION

Due to the large number of genomic features included in the model, different variable selection techniques were evaluated in order to prevent overfitting, improve computational speed, and eliminate uninformative variables. There were three different wrapper methods that were used here, forward, backward, and stepwise selection. A null model consisting of just an intercept term and a full model, including all features, were specified for each selection technique. For all three, a logistic regression model was performed, and the AIC of the fitted model was used to establish which variables to include. That is, only a reduction in the model's AIC contributed to the inclusion of a feature in the model. Additionally, a recursive feature elimination (RFE) algorithm was performed. In RFE, a full model was created, a measure of variable importance was then computed that ranked the predictors from the most important to the least. Here, the importance of the model was based on the random forest importance criterion [??? Unclear how is it different from backward selection, compare and contrast]. Then, at each stage of the search, the least important predictors were iteratively eliminated prior to rebuilding the model. The process was continued for a pre-specified subset of predictors.

FINAL MODEL

Once the pipeline produced the appropriate dataset, a random forest algorithm was used to assess which genomic features that were most predictive of the development of TAD boundaries. The algorithm was run iteratively over 100 bootstrap samples, each time producing sensitivities, specificities, AUCs, and variable importances. For each iteration, the model was tuned on the best number of predictors to subset as well as the optimal number of trees to use. At the end of the 100 iterations, row wise means were performed to

aggregate the model performances. We then compared the random forest model to both the MLR model and the MLR model with LASSO estimated coefficients that was proposed by Mourad et al.

RESULTS

There was a total of 247632 genomic bins that made up the response vector Y, 1629 (0.7%) of which contained TAD boundaries (Y=1). A total of 68 genomic features were included in the analysis. Two predictors were included for every feature (one binary, one continuous), making for a total of 136 predictors. The list of predictors is provided in Table X. The feature space was reduced to 90 after all near-zero variance predictors were removed. The predictors that were removed are listed in Table X [??? Mark them in supplementary table of all features].

Figure X presents the model performance metrics of the eight different SMOTE combinations. The SMOTE combination utilizing percent over of 100 and percent under of 200 yielded the largest AUC (0.794). This was to be expected since this combination created perfectly balanced classes. However, looking at Figure X, the method using 100 bootstrap sample performed better than then model using SMOTE with an AUC of 0.809. Thus, it was concluded that using bootstrap samples would be more efficient moving forward.

The best normalization technique was found to be the model that included a log base 2 transform with no standardized predictors. The AUC was found to be 0.809 and presented in Figure X. We see that this value is only marginally greater than the model using a log base 2 transform and standardized predictors. However, from Figure X, we see that the variable importance plot associated with a log base 2 transform with no standardized predictors (top right) yielded more accurate results than all other combinations. Notably, the insulator, CTCF, and DNaseI features are located in the top 3 most important features for predicting TAD boundaries, as expected. This is in contrast to the other three combinations, which presented conflicting results. Thus, it was concluded that only performing a log base 2 transformation was necessary.

The best performing variable selection technique was found to be forward selection (AUC: 0.810) as shown in Figure X. All of the stepwise procedures performed better than the recursive feature elimination method, which not only performed worse, but also chose the most predictors. It should be noted that both forward and backward selection had the same AUC. However, the forward selection chose fewer predictors, 28 compared to 34 respectively. Thus, forward selection was chose as the most appropriate variable selection technique. Figure X presents the relationship between each variable selection technique. We see that there was significant agreement between forward and backward selections, with 26 predictors being shared by both. Likewise, there

was a total of 12 predictors in common between all four techniques. Table X lists the common predictors between each comparison.

After the data was ran through our pipeline, the final dataset consisted of 28 log transformed predictors. Lastly, 100 bootstrap sampled random forest models were performed on said data. Figure X presents the results compared to the multiple logistic regression models proposed by Mourad et al. We see that the random forest model, using the pipeline processed data, preforms better compared to both models proposed by Mourad et al with an AUC of 0.805. Moreover, the variable importance plot in Figure X reiterates the strong predictive performance of the random forest model by indicating that the DNaseI, insulator, and CTCF genomic features are among the top 3 variables most predictive TAD boundaries. The MLR with LASSO estimation only marginally out performs the regular MLR model with AUCs of 0.750 and 0.747 respectively. Likewise, from Table X we can see that there is more similarity in the ranking of important features between the random forest model and the MLR with LASSO model. However, there is still a large disparity, with 12 variables recognized as important by the random forest that are missing from the MLR with LASSO model. Thus, we conclude that a random forest algorithm applied to a dataset ran through the proposed pipeline performs uniformly better than a multiple logistic regression model, regardless if LASSO estimation is used.

DISCUSSION