Imputing Methylation Status

Curtis Belmonte
Princeton University
curtislb@princeton.edu

Mohamed El Tonbari
Princeton University
mtonbari@princeton.edu

Abstract

"Epigenetics refers to the study of non-genetic cellular processes that may be inherited, are stable through cell division, and may change in response to external and internal cellular stimuli". Epigenetic modification of DNA can be studied through DNA methylation which is a biochemical process where a methyl group is added to the DNA base cytosine near CpG sites. DNA methylation can be responsible of altering the expressions of genes during cell division and are important in better understanding cancer development and cell identity/memory [1]. In this project, we address the problem of predicting methylation levels of CpG sites not on the Illumina 450k array using a variety of regression models and feature sets. We first start by ignoring the missing values in the train data, compute the mean methylation level at each position to impute the missing methylation levels in the sample file. This method led to an RMSE of 0.1. Another method which led to better results consists of filling in the missing values in the train data, training on the observable methylation value in the sample data and predicting the missing ones. Different regression models lead to similar results with an RMSE of around 0.06. Finally, we also took into account the methylation values of neighboring positions which led to better results for chromosome 1.

1 Introduction

DNA methylation plays an important role in repressing gene activity, altering the expression of DNA and ensuring normal development. The study of DNA methylation is important in explaining certain aspects of cancer and aging to name a few [2]. The better and more accurate way of predicting methylation values is the assay of DNA methylation patterns using whole genome bisulfite sequencing (WGBS). However, WGBS is expensive and contains noisy or incomplete data because it is hard to collect. We thus need a quantitative method to predict methylation values which is the objective of this project.

In this project, we evaluate five different regression models and the Naive Imputation method to predict methylation values using the data set which consists of three files which are the train file, the sample file and the test file [3]. The train file contains 33 reference samples with whole genome bisulfite sequence data with the genomic locations of each CpG site, the sample file contains only one sample with many missing methylation values and a few observable values and finally, the test file which we use to evaluate our methods. All tests were performed on chromosome 1 files. We first test the performance of the Naive Imputation method which simply takes the average of the beta values at each position to predict the methylation values at the corresponding location of the sample. We then measure the performance of the regression models by training on the available values in the sample file and attempting to predict the missing ones, using the methylation values from the corresponding rows of the train file as features. Finally, we use the methylation values of neighboring position as additional features.

The code for our project can be found at the following GitHub repository: `https://github.com/curtislb/Methylation`

2 Methods

We evaluate six different methods with and without 5-fold cross validation. Although using the beta values at each position as features leads to interesting and positive results, it would be interesting to see the performance of the methods built when adding other features such as the methylation values of neighboring positions.

2.1 Data processing

For all methods and features used, we ignored the first four columns and the last column which include the chromosome, the start and end positions, the strand ("+" or "-") and whether the position defined is present on the illumina 450k chip (0 or 1). When testing the Naive Mean Imputation, we ignore the nan values in the train file and simply take the average of the beta values to predict the missing methylation values at a given location in the sample file. To test the linear models, we first fill in the missing values in the train file by computing the mean of the remaining values to have a complete feature set. Using the sample file, we then train on the $7,523$ available beta values and test on the $368,411$ positions which are not on the 450K chip and which do not contain a nan in the test file (file containing the true methylation values), using the 33 beta values at the corresponding position from the train file as features and adding 66 features when taking into account the previous and next location's beta values. The linear model then assigns weights accordingly.

2.2 Regression Models

We use five different generalized linear models from the SciKitLearn Python libraries [4] and the Naive Mean Imputation method. All parameterizations are the default unless specified.

1. *Naive Mean Imputation* (NMI): using the mean of the beta values at each position
2. *Ordinary Least Squares* (LS)
3. *Lasso* (L): using LassoCV to find optimal α
4. *Elastic Net* (EN): using ElasticNetCV to find optimal α
5. *Stochastic Gradient Descent* (SGD): learning rate α of 0.0001
6. *Passive Aggressive Regressor* (PAR): maximum step size of 1 and ϵ of 0.1.

2.3 Evaluation

We evaluate each method's performance and each feature set used by computing the root mean squared error (RMSE) and the coefficient of determination r^2 which is a value between 0 and 1. A low RMSE and an r^2 close to 1 indicate a predictive model and vice versa. Moreover, in order to see how the classifiers generalize to unseen data, we perform 5-fold cross validation where we take the average of the RMSE and the r^2 score across the different folds.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)}{n}},$$

where $\hat{Y}_i$ are the predictions of n positions and Y_i are the true values.

$$SS_{res} = \sum_{i=1}(Y_i - \hat{Y}_i)^2, \quad SS_{tot} = \sum_{i=1}(Y_i - \bar{Y}_i)^2$$

$$r^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where Y_i are the true values, $\hat{Y}_i$ the predicted values, $\bar{Y}_i$ the mean of the true values, SS_{res} the sum of squares of residuals and SS_{tot} the total sum of squares.

3 Results

3.1 Evaluation results

The following table summarizes all of our results. The table is divided into two sections: the training and testing where we train on the available beta values and predict the missing values in the sample file, and the 5-fold cross validation conducted on the test file which contains the true values. For each section, we compute the RMSE and r^2 values when using the beta values from the corresponding rows of the train file as features and when adding the beta values of the previous and next positions to the feature set. The RMSE and r^2 values in the cross validation correspond to the averages across the five folds.

	Training and Testing (Based on Sample)				Cross Validation (Based on Test)			
	Current Row		Neighboring		Current Row		Neighboring	
Method	r^2	RMSE	r^2	RMSE	r^2	RMSE	r^2	RMSE
NMI	0.5173	0.1175	N/A	N/A	0.5126	0.1184	N/A	N/A
LS	0.8752	0.05975	0.8773	0.05922	0.8781	0.05922	0.8810	0.0585
L	0.8756	0.05964	0.8784	0.05896	0.8781	0.05923	0.8809	0.0585
EN	0.8757	0.05961	0.8786	0.05892	0.8781	0.05923	0.8809	0.0585
SGD	0.8625	0.06269	0.8644	0.06227	0.8769	0.05952	0.8801	0.0587
PAR	0.8078	0.07413	0.8160	0.07253	0.8673	0.06180	0.8708	0.0609

We find very similar performance across the methods, with slight deviation. The Naive Mean Imputation performed the worst in all cases, with and without cross validation. Without cross validation, the regression models all have very similar performance with the exception of the Passive Aggressive Regressor performing slightly worse than the rest. The RMSE and r^2 values of the regression models (excluding Passive Aggressive Regressor) differ by less than 0.02 and less than 0.01 respectively. We find even less variability when conducting 5-fold cross validation. In addition, adding the neighboring beta values slightly improves performance. Additionally adding the distances to the neighboring locations decreased performance. We do not include the characteristic values for the Naive Mean Imputation method when adding the neighboring beta values to the feature set because imputing the mean for the current row across multiple rows is inappropriate and predictably yields worse performance. Finally, we apply the same methods to chromosome 2 and get very similar performance which indicates that our methods perform well when using different data sets. However, it is interesting to note that although for chromosome 1 having the neighboring beta values improved performance of all of our models, for chromosome 2 it improved the performance of Least Squares Regression but hurt the performance of Passive Aggressive Regressor.

3.2 Residual Plots

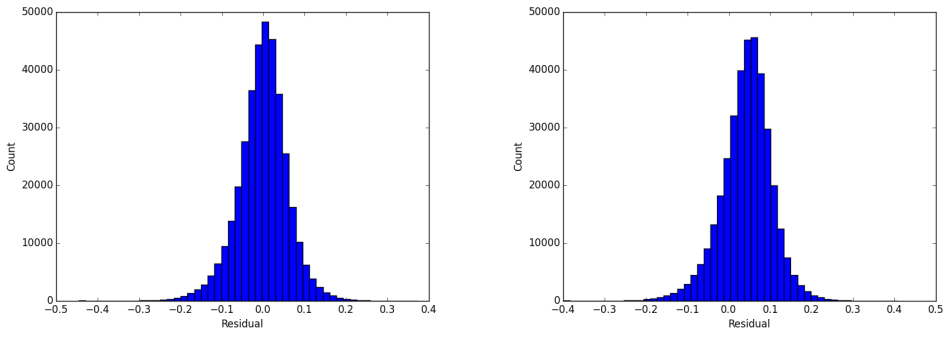

Figure 1: Least Squares (left) and Passive Aggressive Regressor (right) Residual Plots without neighboring values

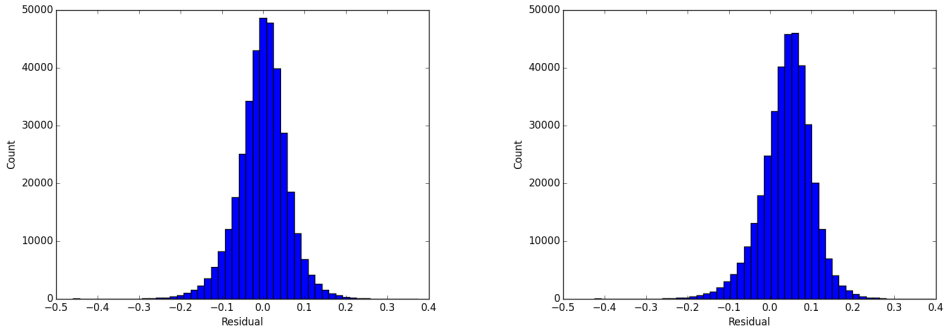

Figure 2: Least Squares (left) and Passive Aggressive Regressor (right) Residual Plots with neighboring values

As expected, although slightly skewed, the residual plots are normally distributed. Figure 1 and 2 show that the residual plots of Least Squares and Passive Aggressive Regressor are skewed to the right which means that our methods tend to overpredict the methylation values. A full list of residual plots of all of our methods, with and without cross validation, can all be found in our GitHub repository in the report/img subfolder.

3.3 Most Predictive Samples

Using all 33 reference samples from the train file is indeed beneficial to build a predictive model. Using SelectKBest method from SciKitLearn, we are able to find which sample in the train file is most predictive of the methylation values in the sample file. We find that sample 20 is the most predictive with a significantly higher score then the rest. This indicates that the sample in the sample file is most similar to the twentieth sample in the train file which is valuable information. The table below lists the seven most predictive samples from the train file along with their corresponding scores. The scores correspond to the F-score of the ANOVA and are obtained using the f_classif function.

Sample	F-Score
20	62.95
22	26.67
21	25.55
23	25.49
32	21.40
25	19.74
24	19.65

4 Discussion and Conclusion

In this project, we have compared the performance of six different methods including the Naive Mean Imputation and five different generalized regression models.The first feature set used to predict the beta value at a position are the beta values of the 33 reference samples at the same position in the train file. The second feature set added the beta values of the previous and next positions. We find that the Naive Mean Imputation has the worst performance while the five regression models perform well and have very similar RMSE and r^2 values.Adding the beta values of the neighboring positions improved performance across all methods. However, adding the distances to these positions yielded worse results.

Since adding the neighboring beta values to our feature set improved the results, an interesting extension to this project includes exploring more complex feature sets. Our models can be improved by adding a binary variable indicating whether the CpG site is in a CGI intronic or exonic region

and whether or not the given location is on a CpG island as extra features. Previous studies show that CpG island attributes, DNA structure patterns, whether the DNA is located on a CpG island and DNA sequence composition patterns are predictive features [5]. Moreover, another way to explore these features would be to run feature selection to have a better idea of which feature is more predictive. Finally, another addition to our feature set would be to include the beta values of the other locations by weighing these beta values according to how far they are from the location we are trying to predict. The further away the location, the less weight we assign to the respective beta values and vice versa.

References

[1] Courtney A Miller JDS (2007) Covalent modification of dna regulates memory formation. Neuron 59.

[2] Gonzalo S (2010) Epigenetic alterations in aging. Journal of Applied Physiology 109.

[3] Ziller MJ MFDJea Gu H (2013) Charting a dynamic dna methylation landscape of the human genome. Nature .

[4] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12: 2825–2830.

[5] Hao Zheng JL Hongwei Wu, Jiang SW (2013) Cpgimethpred: computational model for predicting methylation status of cpg islands in human genome. BMC Medical Genomics .