DNA Demystified

Chaney C. Lin
chaneyl@princeton.edu

Liangsheng Zhang
liangshe@princeton.edu

Abstract

In this work, we performed various regression analyses on DNA methylation levels to predict unmeasured ones. Among all models tried, two seemingly different linear regression models provided the best results. We provide a detailed analysis of all the models, and offer an explanation for some phenomena observed.

1 Introduction

External factors affect how genes are expressed; one such factor is DNA methylation. Just as our understanding of genetics develops with better statistics and more data, so it is with DNA methylation. However, assaying methylation levels across the full genome remains prohibitively expensive; more affordable partial assays are available, but for these, the difficulty is predicting the methylation levels (MLs) of unmeasured sites, so-called imputing. This is what we aim to achieve in our project: a reliable model for imputing.

2 Data Description

The training data is a set of $N = 32$ whole genome bisulfite sequences, from Ziller et al [1], which are the reference sequences and will be referred as `training`. The test data is a partial assay, also from Ziller et al [1]. Two separate sets are included in the test data: one called `sample`, another called `test`; both refer to the exact same sequence, but the values are mostly unobserved in `sample`, while observed in `test`. Data for twenty chromosomes were provided; in this report, we work only with the first chromosome.

For each sequence i, we have the MLs $\{\beta_{i,d}\}$ at CpG sites, where d is the starting position of a given site, and where $\beta \in [0, 1]$ is the fraction of methylated reads at the site from this sequence i with starting position d. Starting and ending positions for each site were provided, but the difference between each pair is the same; so henceforth, by position, we refer only to the start position. The end position is redundant. The positions are the same for all sequences in `training` and `sample`. We denote the `sample` MLs as $\beta_{0,d}$, and the `training` MLs as $\beta_{i,d}$ where $1 \leq i \leq N$. Each sequence contains 379,551 sites. Other features are provided, e.g. strand, and each d uniquely corresponds to one set of features.

Models are fit to `training`, to predict the `sample` values $\hat{\beta}_{0,d}$, which are then compared against the observed `test` values $\beta_{0,d}$ to estimate the models' prediction error, according to Eqn. (3).

3 Methods

3.1 Data processing

The data set was downloaded on March 20, 2015 from Bianca Dumitrescu's COS424 directory on Princeton's Nobel cluster. Both `training` and `sample` contain sites with unmeasured MLs, labeled "NaN".

The only preprocessing is imputing on `training` to replace the NaN values. We have chosen to set a NaN value as the average of observed values with the same d, in the other reference sequences.

A subset of the `sample` MLs has been observed. We denote this subset by Ω.

3.2 Models

The models were applied using built-in functions of the scikit-learn Python library [2]. Default parametrizations were used, unless otherwise specified. Below, we define each model, explain our motivation for them, and summarize their performance.

Naive mean model (NMM). This was our benchmark model which was simple and fast. It predicts using the mean of the reference sequences

$$\hat{\beta}_{0,d} = \operatorname*{avg}_i \{\beta_{i,d}\}$$

Such a model would be optimal if for each d, the only information available were $\beta_{i,d}$.

Simple linear regression (SLR1). This model was essentially a simple linear regression. For each site $d \notin \Omega$, a simple linear regression is performed, using features $\{\beta_{i,x}\}_{x\in\Omega}$ to fit the value $\beta_{i,d}$. Because i runs from 1 to N, there are N data points (for each $d \notin \Omega$). The resulting parameters $\{a_x\}_{x\in\Omega}$ are used to predict $\beta_{0,d}$ by

$$\hat{\beta}_{0,d} = \sum_{x\in\Omega} a_x \beta_{0,x} \ . \tag{1}$$

In general, for different d', the parameters $\{a'_x\}$ will differ from $\{a_x\}$. The parameters $\{a_x\}$ for a given d thus implicitly incorporates all of the information given by d. It was significantly slow due to the huge number of linear regressions to be performed and its predictions need not lie in the range $[0, 1]$.

Generalized linear model (GLM). This model was intended to tackle the issue of prediction range. It was a generalized linear model based on the logit link function $f(\beta) : [0, 1] \to (-\infty, \infty)$ with the form

$$f(\beta) = \ln(\beta^{-1} - 1) \ .$$

Regularized linear regression (RegLR). These were our attempts to reduce the dimensionality of the feature space, by introducing regularizers in the linear regression. We looked at both LASSO regression and ridge regression. The regularization constant α was determined through cross validation.

K-means clustering (Kmeans). This model, based on K-means clustering, can be considered as an approximation to our simple linear regression. For $d \in \Omega$, $B_{d,k}$ refers to a vector $(\beta_{1,d}, \ldots, \beta_{N,d})$ that has been classified into the kth cluster using K-means clustering. We assign the same classification to the corresponding site in `sample`, which we denote by $\beta_{0,d,k}$. A simple linear regression is performed, using as features the centroids $C_k = \operatorname{avg}\{B_{d,z} : d \in \Omega, z = k\}$ of the K clusters to predict B_d for $d \notin \Omega$. The resulting parameters a_k are then used to predict $\beta_{0,d}$ by

$$\hat{\beta}_{0,d} = \sum_{1\le k\le K} a_k C_{0,k}$$

where $C_{0,k} = \operatorname{avg}\{\beta_{0,d,z} : d \in \Omega, z = k\}$ are the centroids of the K clusters in `sample`. It significantly reduces the dimensionality of the feature space when K is small. We investigated $K = 2^n$ for $3 \le n \le 9$.

Simple linear regression 2 (SLR2). This was another simple linear regression model which investigated the data from a different perspective, using as features $\{\beta_{i,d}\}_{1\le i\le N}$ to fit the value $\beta_{0,d}$. Because every position $d \in \Omega$ provides one data point, the total number of data points is the size of Ω. The resulting parameters $\{a_i\}_{1\le i\le N}$ are used to predict $\beta_{0,d}$ by

$$\hat{\beta}_{0,d} = \sum_{1\le i\le N} a_i \beta_{i,d} \ . \tag{2}$$

3.3 Evaluation

The metric used to evaluate the predictions $\left\{\hat{\beta}_{0,d}\right\}$ for a given model is

$$\epsilon^2 = \frac{1}{|\Gamma| S^2} \sum_{d\in\Gamma} \left(\hat{\beta}_{0,d} - \beta_{0,d}\right)^2 \tag{3}$$

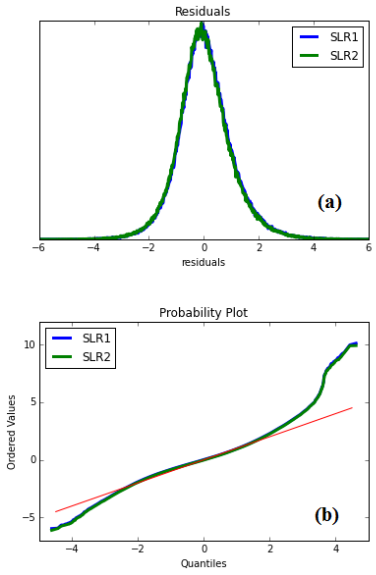

Figure 1: (a) shows the distribution of all prediction errors from SLR1 and SLR2. (b) demonstrates the quantile-quantile (qq) plot of the standardized distribution against a normal distribution. Details are provided in the discussion section.

Model	t_F	t_C	ϵ^2
SLR2	00:00:23	-	0.12617176
RidgeLR	01:47:21	-	0.12680164
SLR1	07:00:54	-	0.12684927
GLM	07:00:50	-	0.12804621
256means	00:12:28	00:02:21	0.13797524
512means	00:16:53	00:04:08	0.14206517
128means	00:09:58	00:01:06	0.14661424
64means	00:09:09	00:00:31	0.15221729
LassoLR	21:35:56	-	0.15377764
16means	00:05:22	00:00:20	0.17631774
8means	00:04:36	00:00:14	0.18287992
NMM	00:00:08	-	0.50788960
32means	00:07:44	00:00:26	0.81595437

Table 1: For each model, we report: (i) the fitting and prediction time t_F; (ii) the clustering time t_C for the K-means models; and (iii) the prediction error ϵ^2, defined in (3). Models are sorted (in decreasing order) by ϵ^2. All times are in the format hh:mm:ss

where Γ is the set of sites with observed values (i.e. not NaN) in `test` that are not observed in `sample`. The $\{\beta_{0,d}\}$ are the known values in `test`. S^2 is the variance of $\{\beta_{0,d}\}_{d\in\Gamma}$. This metric is non-negative, and smaller values correspond to better predictions. If $\hat{\beta}$ were fitted by a linear regression on β, then $\epsilon^2 = 1 - R^2$, where R^2 is the coefficient of determination. Alternatively, it can be interpreted as how good the prediction is, relative to just predicting with one number, the sample mean; this can be seen from the definition of S^2, and observing that without the S^2, the right hand side is simply the average error squared.

4 Results and Discussion

The data referred to in the text is contained in Table 1.

4.1 Computational speed

The fastest model to predict was NMM, given that it does not have to do any fitting. Next fastest was SLR2, which performed only one fitting. All other models performed multiple iterations of linear regression, so were significantly slower. Of these, the K-means models were the fastest, as they reduce the dimensionality of the feature space in the clustering step. The remaining regression models were much slower than K-means, with LASSO regularized regression performing the slowest of all. The disparity in the times between the regularized models and SLR1 suggests that the scikit-learn package handles the calculations differently. Among the K-means models, one sees that the clustering time t_C and the fitting and predicting time t_F increases with the number of clusters, as expected: more clusters means more centroids and more features.

4.2 Model Performance

As expected, apart from 32means, all models perform significantly better than NMM, confirming one's expectation that incorporating information from other sites improves our prediction power. It is also interesting to observe that the two simple linear models are among the best. Detailed analysis of these two models will be given later in this section.

With GLM, we expected that enforcing the constraint that predictions lie in $[0, 1]$ would improve the results, but they were actually slightly worse than with SLR1. This suggests that the constraint is already handled well by SLR1.

K-means models, with one exception, had comparable performance to SLR1, while being significantly faster. This supports using clustering vectors B_d to reduce the feature space. The one exception was 32means, which performed markedly worse. We ran the K-means models multiple times, and continued to observe this performance anomaly in 32means. We do not yet have a good explanation for this.

For the regularized models, the regularization constant α, after cross-validation, was extremely small, indicating that these models were unable to significantly reduce the feature space. In future work, other regularization methods can be attempted. It is amusing to see RidgeLR performs slightly better than SLR1 while LassoLR performs significantly worse, given that theoretically they should not deviate much from SLR1. This possibly reflects differences in the implementation details of the algorithms.

It is surprising to see that our two simple linear regression models gave similar fitting accuracy, considering their dramatically different assumptions. Particularly, in SLR1, it is assumed that each site is statistically different and treated as a feature, whereas different samples for a given site are assumed to be drawn independently from a common underlying distribution. On the contrary, SLR2 assumes each sample to be statistically different while each site to be statistically the same. Note however, in either model, the sample/site information is still incorporated into prediction through explanatory variables. Their similarity thus suggests that in both models, sample and site information are considered on equal footing. Actually, if intercept were not fitted, then by ignoring invertibility issues of certain matrices, we can prove that the two models give the same results. Specifically, the result from SLR1 is

$$\hat{\beta}_{0,d} = \beta_{0,D}^T \left(X^T X\right)^{-1} X^T \beta_d$$

for each $d \in \Gamma$, where $\beta_{0,D} = \left(\beta_{0,d_1}, \beta_{0,d_2}, \cdots, \beta_{0,d_p}\right)^T$ with $p = |\omega|$ and $d_i \in \omega$ for $i = 1, 2, \cdots, p$ and

$$X = \left(\beta_{d_1}, \beta_{d_2}, \cdots, \beta_{d_p}\right)$$

with each β_{d_i} written as a column vector of dimension N for each site d_i. The result from SLR2 is

$$\hat{\beta'}_{0,d} = \beta_{0,d}^T \left(X'^T X'\right)^{-1} X'^T \beta_{0,D}$$

where $X' = X^T$. Denoting $C = \left(X^T X\right)^{-1} X^T$ and $C' = \left(X'^T X'\right)^{-1} X'^T$, then, since the prediction is a scalar, we have $\hat{\beta}_{0,d} = \beta_{0,D} C \beta_d$ and $\hat{\beta'}_{0,d} = \beta_{0,D} C'^T \beta_d$. Using the identity $X^T \left(X'^T X'\right) = X^T X X^T = \left(X^T X\right) X'$, we can show that $C = C'^T$, indicating the two predictions are the same. Reality is more complicated: we have intercepts, and $X^T X$ is unlikely to be invertible, as we have more features than data points. However, the spirit of this proof may still carry over, resulting in similar results.

To further understand prediction results from linear regression models, one may look at the distribution of prediction errors: $\hat{\beta}_{0,d} - \beta_{0,d}$ for all $d \in \Gamma$. To pool all errors together usually requires statistical independence. This is only implied by the assumption underlying SLR2, but it turns out that the distribution of errors from SLR1 follows roughly the same shape as the one observed for SLR2 (see Fig. 1(a)). This distribution has a Bell-like shape with a small sample variance; however, QQ plots (see Fig. 1(b)) suggest that the standardized errors have heavier tails (kurtosis about 5.1 for both) than the standard normal distribution. Moreover, the distributions are slightly skewed (skewness about 0.37 for SLR1 and about 0.35 for SLR2), suggesting a systematic overestimation. More quantitatively, the non-normality is reflected by both distributions failing the Jarque-Bera test.

5 Conclusion

In this work we implemented various models based on simple linear regression to predict unobserved methylation levels based on observed values. Among all methods implemented, two simple linear regression models provided the best results, even though their prediction errors do not follow a normal distribution. Though based on quite different assumptions, they gave strikingly similar results. A further understanding of the similarity may shed more insight into their underlying structure, and possibly even hidden data patterns. One may also seek to improve the performance of these two models by accounting for the observed heavy tails of prediction errors.

References

[1] Ziller MJ, Gu H, Müller F, Donaghey J, Tsai LTY, et al. (2013) Charting a dynamic dna methylation landscape of the human genome. Nature 500: 477–481.

[2] 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.