Methylation Imputation with Gaussian Mixtures

Vlad Feinberg, Siddhartha Jayanti, Jerry Liu, Zi Xiang Pan
Undergraduates
COS 424 Final Project, Group 38
Princeton University
{vyf,sjaryanti,jjliu,zpan}@princeton.edu

Abstract

This paper presents an original approach to impute methylation levels in DNA. We used a Gaussian mixture model with each cell line being a 380K-dimensional feature vector. This ends up being intractable for default GMM models, so we built a custom implementation of EM and marginal posterior computation using TensorFlow to handle the high-dimensional data in a numerically stable way. We achieved an R^2 value of 0.84707 and RMSE of 0.0661 which is comparable to regression techniques.

1 Introduction

Methylation patterns in DNA determine gene activation in various tissues in the human body. Methylation values represent the proportion of the sites at a certain sequence in a genome that contain a methyl group attached to a CpG site [15]. Because the methylation values manifest as activation patterns for protein production, they are important to study for many medical analyses. Different tissues and different levels of health will alter methylation values.

However, counting the proportion of sites that are methylated along an entire genome is expensive. We ask the question of whether it is possible to impute the majority of methylation values after constructing some model for their behaviors from trends learned on densely assayed tissue samples.

In a previous work, we imputed methylation values of sparsely assayed samples via regression on methylation values at known sites of other densely assayed samples.

Regression methods, however, are limited to considering *a single site* in various tissues to impute the value of the site in the target tissue. Thus, these methods are unable to use all the information in the data matrix when imputing. Thus, in this work we propose imputation by Gaussian Mixture Model (GMM) to cluster entire tissue samples by their similarities with each other, and using these clusters to impute the missing values in the partially complete target sample. Furthermore, this provides a probabilistic method for regression that allows us to have confidence intervals.

2 Related Work

In [14], the authors solve an identical problem of methylation imputation. As features, they rely on methylation levels of nearby sites, sequence-encoded information such as genetic context, and indicators for whether the given site is within the sequence for a cis-regulatory element (CRE) [14]. For regression, the authors apply random forests, from which they infer methylation-predictive CREs, earning a 0.41 R^2 without other tissue samples, only relying on the contextual metadata.

An additional finding from [14] is the predictive power of neighboring sites for methylation values. In particular, the randomly-sampled CpG sites were correlated with median absolute $r = 0.22$,

with a sharp correlative increase as one nears the original site. The article correctly notes that this observation is not useful for inferring the sparsely-sampled microarray methylation values because of the large distances in bp between known values.

In [2] the authors use many learning methods to predict methylation status across the whole genome, and discover that kernelized SVM outperforms k-means clustering, linear discriminant analysis, and logistic regression, by achieving 86% accuracy in predicting methylation propensity in CpG islands.

3 Dataset

Our dataset is a set of WGBS libraries comprising 30 distinct primary cell lines, tissues, in vitro derived cell types and cell lines from [15]. The methylation is performed on the Illumina HiSeq 2000 platform (ID GPL11154).

The primary dataset has samples from 379551 sites. There are 34 samples for each site, each taken from an expensive WGBS procedure. This method is able to measure about 91% of sites [6]. Our test sample only has about 2% of the sites available from a cheaper procedure: methylation microarrays [14].

We are to use the limited information to impute the missing values in the sample. Because we can only use WGBS as the true value, there are some sites that we don't know the correct value for, even in testing, due to invalid values. We therefore disregard these sites in our imputation procedure even though the methods constructed attempt to impute the entire genome.

The sites available at test time are consistent - the technology used to assay the methylation values samples sites consistently [4]. This opens up opportunities to take advantage of learning patterns particular to the sites consistently tested by the methylation microarrays - for instance, we can mask one of the well-sampled tissues enabling cross-validation on a reduced feature set.

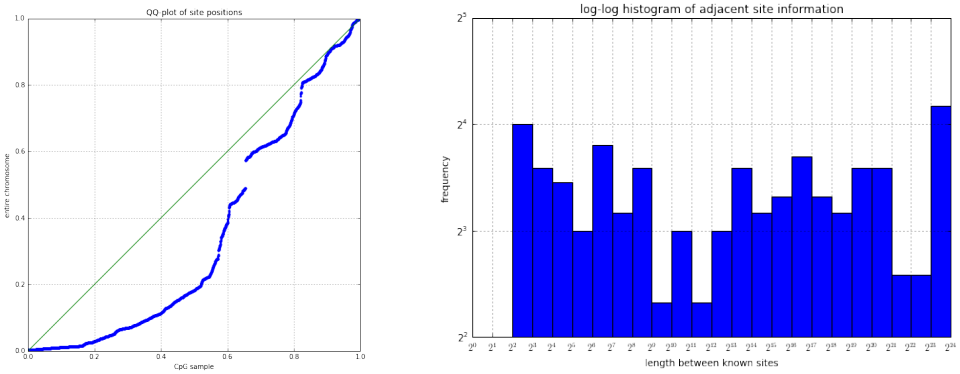
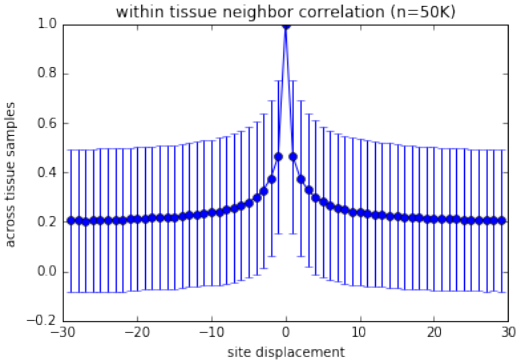


Figure 1: The above demonstrates the distribution of the known sites in the test sample. The mean distance in bp between sites is 33058, with a standard deviation of 262002. The QQ-plot demonstrates that the microarray samples throughout the genome.

The microarray may only provide about 2% of the chromosome's information, but its uniformity, displayed in Figure 1, coupled with the observations of correlation among neighboring sites from Figure 2, tells us that a whole-sequence approach may generalize well because sequence-wide patterns may be captured. This differs from a regression-based approach, which just postulates that a chromosome must be a linear combination of the others and that this relationship is independent of site location.

Figure 2: The graph demonstrates estimates of the correlation of methylation values with the neighbors a given distance away from a sample site on the same tissue sample. The "bottoming out" of correlation at about 0.25 as we distance ourselves from the site matches the observed background correlation from [14]. Error bars are $1\hat{\sigma}$.



Note that in both Figure 2 and the analysis below, we dropped tissue samples 14, 25, 26, and 33 due to the sparsity (or near-constant values) – see Figure 3 for the correlation matrix. This affected the model - not dropping the sample caused the GMM to fit one of the clusters to a near-zero mean Gaussian, which was not useful for prediction. Figure 2 also informs us that using linear interpolation to fill in the few `NaN` values in the WGBS assayed-samples is safe.

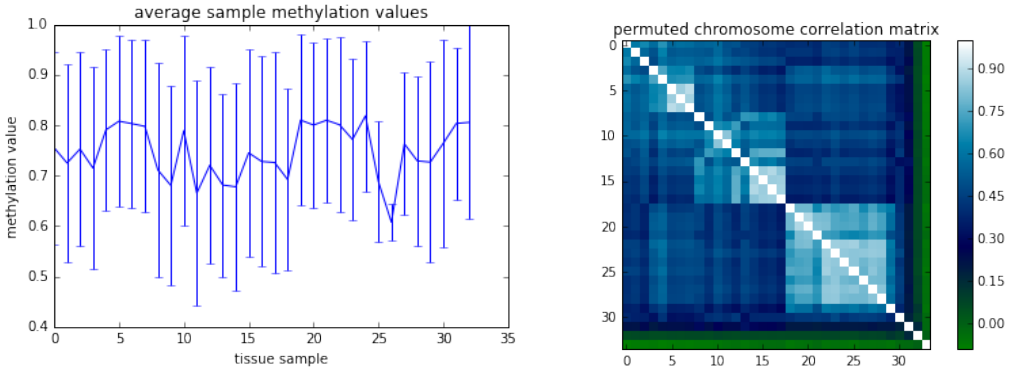


Figure 3: While the tissue sample means and standard deviations ($1\hat{\sigma}$ error bars) show no obvious characterizations, the correlation matrix does – note that after permuting the tissues we see three overarching clusters; mixed membership seems like a real phenomenon in these data.

4 Methods

4.1 GMM

For a given K, a set of N observations $\mathbf{z}_i \in \mathbb{R}^D$, and priors in the K-simplex $\boldsymbol{\pi} \in \Delta_K$, we fit the following model, denoting the set of integers $[n] = \mathbb{Z} \cap [1, n]$:

$$C_i \sim \text{Categorical}\left(\boldsymbol{\pi}\right) \qquad i \in [N]$$

$$\mathbf{z}_i \sim \sum_{j=1}^{K} C_{ij} \mathcal{N}\left(\boldsymbol{\mu}_j, \Sigma_j\right) \text{ iid} \qquad i \in [N]$$

For the categorical variable C_i, $C_{ij} = 1_{C_i = j}$.

The GMM is visualized in plate notation in Figure 4

The above is fitted with Expectation-Maximization (EM). We use a uniform distribution for our cluster priors $\boldsymbol{\pi}$. For our means $\boldsymbol{\mu}_j \in \mathbb{R}^D$, random initialization is done by uniformly selecting K points from our N samples.

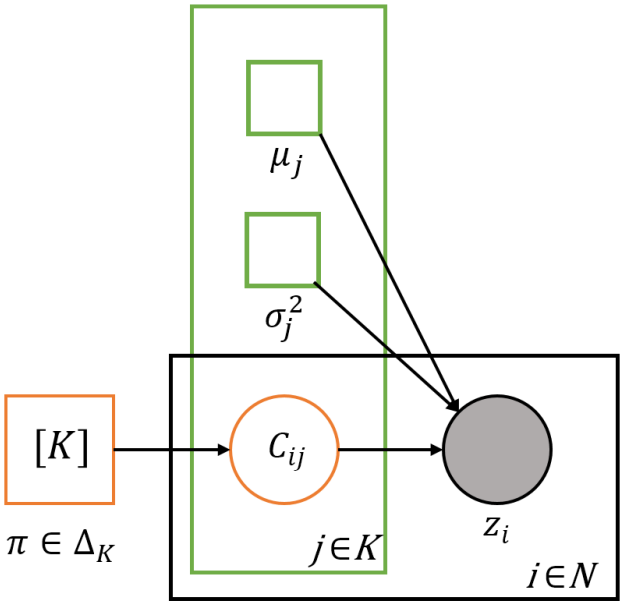


Figure 4: GMM in plate notation

The covariance Σ_j is initialized as the same for all Gaussians, set to the diagonal matrix of the variance of each dimension.

4.2 Marginal Posterior Computation

After fitting a GMM model, we are given some observations of a test value $\mathbf{y} \in \mathbb{R}^O$ where $O < D$. For a single multivariate normal (MVN), the marginal posterior is just another normal, whose values are determined by the Schur complement [13]:

$$\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_\mathbf{x}\\ \boldsymbol{\mu}_\mathbf{y}\end{bmatrix}, \begin{bmatrix}A & B\\ B^\top & C\end{bmatrix}\right) \implies \mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\boldsymbol{\mu}_\mathbf{x} - BC^{-1}(\mathbf{y} - \boldsymbol{\mu}_\mathbf{y}), A - BC^{-1}B^\top\right) \tag{1}$$

Taking the GMM model from Section 4.1, we show that we may use the result from above to compute the posterior as a combination of these Gaussians. Let the GMM consist of K Gaussians $\left\{\mathcal{N}_j = \mathcal{N}\left(\boldsymbol{\mu}_j, \Sigma_j\right)\right\}_{j=1}^K$. Let the event that a $\mathbf{z}$ is drawn from the j-th cluster be C_j. Note $\mathbf{z}|C_j = \mathcal{N}_j$ and $\mathbb{P}\left(C_j\right) = \pi_j$.

Since $\{C_j\}$ partitions the outcome space, we have that the posterior density p of $\mathbf{x}$ where $\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}$ given $\mathbf{y}$ is a dimensionally reduced GMM as well:

$$p(\mathbf{x}|\mathbf{y}) = \sum_{j=1}^{K} p(\mathbf{x}|\mathbf{y}, C_j)p(C_j|\mathbf{y}) \tag{2}$$

The first term $p(\mathbf{x}|\mathbf{y}, C_j)$ is given by Equation 1 since $p(\mathbf{z}|C_j) = \mathcal{N}_j$ and the second term can be found by Bayes' Rule, $p(C_j|\mathbf{y}) \propto p(\mathbf{y}|C_j)\pi_j$.

4.3 Restriction to Diagonal

For $D \approx 380$K, computing the likelihood of a single normal with its D^2-size covariance Σ is intractable. As observed in Figure 2, some variance was observed in adjacent sites. As such, a GMM model with nonzero $< k$-off-diagonal bands (corresponding to the covariance in the k most adjacent sites) is more appropriate for the data. Unfortunately, the likelihood computation is intractable. The

likelihood computation involves computing Σ^{-1}, which is a dense matrix for $k > 0$. The simplest case of computing the inverse of a tridiagonal matrix (with two off-diagonal bands) takes $O(D^2)$ in both time and memory with dynamic programming techniques and serial computation [12]. Fully parallelizing the algorithm on infinite processors would bring the runtime down to $O(D)$; however the space usage would remain intractable, as it would be impossible to compute an inner product using a dense matrix in $O(D)$ space.

Furthermore, one may notice from Equation 1 that sites more than k bp away from observed values are unaffected by the observed methylation values. Since we observe only 2% of the test sample, there is not much gained in the direct posterior computation by using the intractable diagonal matrix.

Instead, we opt for a purely diagonal representation of Σ, assuming independence between sites. Then Σ^{-1} is diagonal as well, which allows for linear-time likelihood processing, posterior calculation, and inversion.

4.4 Implementation

Because the dimensionality of the dataset was too large for `scikit-learn` (memory errors were incurred from a quadratic implementation), we were forced to implement a numerically stable EM algorithm using Google's TensorFlow [1]. The package allowed us to express the high-level linear algebra operations in a manner that could be automatically parallelized by an efficient, vectorized native backend.

Various techniques for numerical stability were utilized, such as working in the logarithmic domain and performing a safe log-sum-exp composition without underflow.

There are several hyperparmeters which affected our models that we needed to tune besides K:

1. The posterior likelihood $p(\mathbf{x}|\mathbf{y})$ from Equation 2 induces several guesses - we can choose $\operatorname{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})$ or $\operatorname{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, C_{j^*})$, where $j^* = \operatorname{argmax}_j \mathbb{P}(C_j|\mathbf{y})$. The latter may be more robust if mixed membership is not realizable, and it is also much easier to compute; we simply take the mean of the cluster with the highest posterior given the observed values.

2. We can restart the initial means some number of times, and choose the initialization with the highest data likelihood.

3. The implementation enforces a minimum variance, which can be enlarged to account for and be resistant to measurement noise inherent to the assaying procedure. The minimum variance value is added in to the covariance matrix diagonal after every EM iteration.

We also tried initializing the initial means as the empirical means of a random partition of the N data points, but this did not have a significant effect on the likelihood.

The implementation of the mode of the posterior likelihood, $\operatorname{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})$, is not trivial - because the posterior of a GMM is not a normal distribution, but another GMM, we need to use gradient descent. This was also implemented using TensorFlow [1].

Since the GMM is a convex combination of single-maxima smooth functions, its global max must also be a convex combination of the maxima of its constituent normals. For this reason, it suffices to run gradient descent starting from every mean on the posterior $p(\mathbf{x}|\mathbf{y})$ and choosing the most likely outcome after the gradient descent converged.

For every run of the gradient descent starting from some posterior mean $\boldsymbol{\mu}_i^*$, we needed to make sure that the step size was small enough that we would not leave the convex hull of $\{\boldsymbol{\mu}_j^*\}_{j\in[K]}$. This was ensured by setting the initial step size to be $\min_{j\neq i}\|\boldsymbol{\mu}_i^* - \boldsymbol{\mu}_j^*\|\alpha_{ij}$, where $\alpha_{ij} \in [0, 1]$ is a ratio selected to speed up descent; this quantifies approximately how close $\boldsymbol{\mu}_i^*$ is to a local maxima induced by a convex combination of $\boldsymbol{\mu}_i^*$ and $\boldsymbol{\mu}_j^*$. We set $\alpha_{ij} = \frac{L_j}{L_i+L_j}$, where $L_j = p(\boldsymbol{\mu}_j^*|\mathbf{y})$.

We will refer to the gradient descent method to find $\operatorname{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y})$, the mode of the posterior likelihood, as simply the mode, and the per-cluster method $\operatorname{argmax}_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, C_{j^*})$ as argmax. For a graphical illustration of argmax vs mode, please refer to the Appendix (Section 8).

4.5 Evaluation

To evaluate our model, we performed LOOCV on the densely-assayed tissues. As the covariance matrix from Figure 3 demonstrates, some of the tissues are outliers and thus would not contribute to an informative clustering. They were removed. Then, leaving one sample out, a GMM model is fitted to the remaining training data. Pretending that only the micro-array-assayed sites are visible in the left-out sample, the rest of the sample was imputed, and the out-of-sample R^2 was calculated.

This 30-fold cross-validation process enabled us to have a better glimpse at the generalization error of potential models (where models differed by K and the parameters enumerated in the previous section). The best model by median CV R^2 was then retrained on the whole dataset (under the assumption that an additional sample would not necessitate another cluster), and evaluated on the holdout data.

5 Results

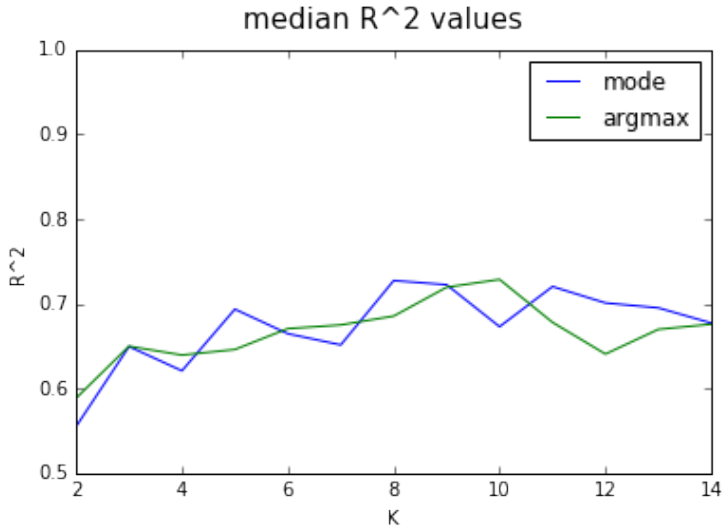


Figure 5: Median R^2 values over different cluster counts during cross-validation for each posterior prediction method. Used optimal hyperparameters for minimum covariance.

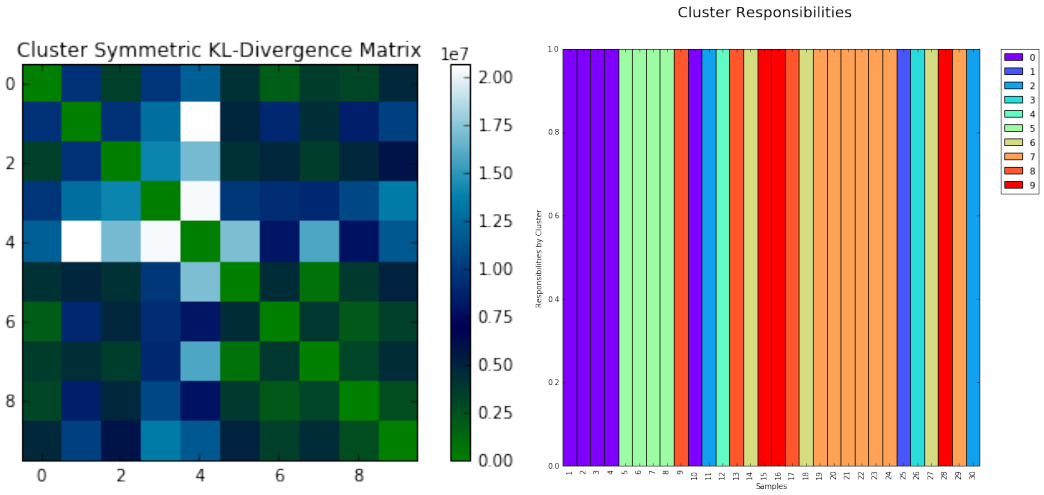
Figure 5 demonstrates the best hyperparameter configurations for both posterior prediction methods through cross-validation. We tested cluster counts $K \in 2, ..., 14$ for both the mode and argmax methods, as well as for the minimum variance (during EM fitting) set to 0.001 and machine epsilon.

The median R^2 curve in Figure 5 has a rough parabolic shape, where we would have more certainty in the K value if the number of samples increased - at 30 samples for 10 clusters, we need more data.

The overall best CV performance was by argmax with $K = 10$ and minimum variance at 0.001, this was the model used on the whole training set and tested on the holdout. We did 100 random restarts on the whole-dataset GMM, choosing the one with the highest data likelihood.

Figure 6a shows the pairwise KL divergence between the K fitted clusters in order to measure their differences. As expected, each cluster has no divergence with itself, and each cluster demonstrates a generally large degree of divergence with other clusters. Some of the clusters; namely 3 and 7, do exhibit some similarity - perhaps indicating K could be smaller.

Figure 6b is a stack plot which represents the normalized responsibility of each cluster for a given sample. However, we note that the "stack" for each sample is not visible, because each sample is dominated by the responsibility of a cluster. This makes sense because our data is so high-dimensional, so that if a sample is closer to one cluster than another, the log likelihood of belonging to the other cluster is extremely low.



(a) KL divergence matrix of clusters

(b) Cluster responsibilities

Figure 6

While some clusters have high responsibilities for multiple samples, others have high responsibilities only for very few samples. This may indicate overfitting of the number of clusters and may be resolved by more data. The clustering of the samples may reveal the underlying biological relationships between the samples - perhaps representing the organ that each sample came from. This is because each organ produces different proteins, resulting in different epigenetic markers within the chromosome.

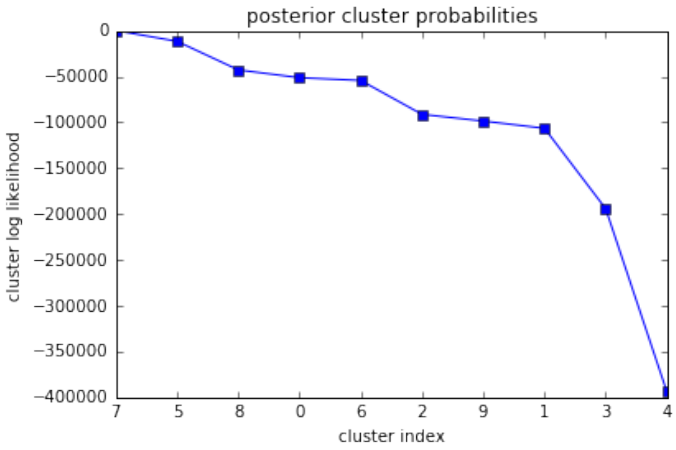


Figure 7: Test sample posteriors. Notice that the posterior for the test chromosome is overwhelmingly dominated by one of the clusters – the second most likely cluster has a log likelihood of -11757.

Because we had a full distribution over the test posterior (which, due to the observation in Figure 7, is approximately normal), we were able to compute a 95% confidence interval around each site's mode. Approximately 0.9596 of the test data was within its 95% CI.

The RMSE is about 0.0661 and the R^2 is about 0.84707.

Lasso regression gives an R^2 of 0.873934 and RMSE of 0.060040, which is slightly better than our GMM model. Regression techniques from the previous class homework are more successful working with a smaller number of samples, since they consider each site value independently, resulting in over 7K examples to train on. Another reason regression performs better on the test set is because the test tissue has methylation extremely close to one of the training tissues – a coincidence the regression fit catches and then uses to have a high test R^2. Hence we anticipate that our GMM and regression may have more similar performance on other training/test sets.

Indeed, the nearest-neighbor method (which uses the nearest neighbor from the training set as determined by the few observed values of the test tissue as the basis for the rest of the imputation) scores an RMSE of 0.839305 and R^2 of 0.067787. The 30-fold CV scores also reveal a more appropriate story for the generalization error: the lasso CV was 0.7813 and the nearest-neighbor CV was 0.6500. While the lasso CV R^2 is still higher than that of our GMM model, we believe that the generalization error of the GMM will improve with more tissue data.

6 Conclusion

In this paper, we achieved a modest R^2 with a new model, a GMM in the entire chromosome-1 space of about 380K dimensions. This approach allows us to analyze an entire chromosome holistically when making a posterior prediction. Our model has many parameters; it can likely become more stable and improve in performance if additional training data is added.

While our GMM model performs worse in terms of R^2 compared to regression methods, it is more advantageous for several reasons. The first is the biological significance: as mentioned above, the clusters we obtain in the model can give us insights into how methylation patterns are dependent on biological origin and other factors for each tissue. The second is that we hypothesize that the GMM will generalize well to datasets with significantly more methylation values. Finally, our technical exploration to implement a custom version of GMM in order to fit very high-dimensional data is invaluable in itself and can be utilized and expanded in other applications.

Additional approaches could use Bayesian consensus clustering to have more informed clustering across multiple datasets [7]. This would share clustering information for methylation values for samples across chromosomes, enabling us to more accurately depict the fluctuations that occur throughout the datasets; in particular, one latent variable this could expose is two samples coming from the same organ.

Besides the multiple dataset integration approach, iterative regularized SVD can be used to further inform sequence-wide trends - the problem of imputing methylation values has an interesting analogue to movie rating imputation [10].

7 Acknowledgments

The authors would like to thank Prof. Engelhardt and Brian Jo for their advice on this project and provision of data.

In addition, the authors leveraged the Princeton CS Department's `cycles` systems for many of the parallelizable grid-search and preprocessing tasks.

Finally, the authors heavily relied on `iPython` [11] notebooks, `sklearn` [9] for algorithm implementations, `scipy` [5] and `pandas` [8] for scientific computing functions, and `matplotlib` [3] for graphing.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Das, R., Dimitrova, N., Xuan, Z., Rollins, R. A., Haghighi, F., Edwards,

J. R., JU, J., BESTOR, T. H., AND ZHANG, M. Q. Computational prediction of methylation status in human genomic sequences. *Proceedings of the National Academy of Sciences 103*, 28 (July 2006), 10713–10716.

[3] HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering 9*, 3 (2007), 90–95.

[4] ILLUMINA. Ifinium HumanMethylation450 BeadChip. Data Sheet, March 2012.

[5] JONES, E., OLIPHANT, T., PETERSON, P., ET AL. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2016-02-23].

[6] LAIRD, P. W. Principles and challenges of genome-wide dna methylation analysis. *Nature Reviews Genetics 11*, 3 (2010), 191–203.

[7] LOCK, E. F., AND DUNSON, D. B. Bayesian consensus clustering. *Bioinformatics 29*, 20 (2013), 2610–2616.

[8] MCKINNEY, W. pandas: a foundational python library for data analysis and statistics.

[9] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG, V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[10] PERCY, M. Collaborative filtering for netflix, Sept. 2009.

[11] PÉREZ, F., AND GRANGER, B. E. IPython: a system for interactive scientific computing. *Computing in Science and Engineering 9*, 3 (May 2007), 21–29.

[12] RAN, R.-S., AND HUANG, T.-Z. An inversion algorithm for a banded matrix. *Computers & Mathematics with Applications 58*, 9 (2009), 1699 – 1710.

[13] WIKIPEDIA. Schur complement — Wikipedia, the free encyclopedia, 2016. [Online; accessed 7-May-2016].

[14] ZHANG, W., SPECTOR, T. D., DELOUKAS, P., BELL, J. T., AND ENGELHARDT, B. Predicting genome-wide dna methylation using methylation marks, genomic position, and dna regulatory elements.

[15] ZILLER, M. J., GU, H., MÜLLER, F., DONAGHEY, J., TSAI, L. T.-Y., KOHLBACHER, O., DE JAGER, P. L., ROSEN, E. D., BENNETT, D. A., BERNSTEIN, B. E., ET AL. Charting a dynamic dna methylation landscape of the human genome. *Nature 500*, 7463 (2013), 477–481.

8 Appendix

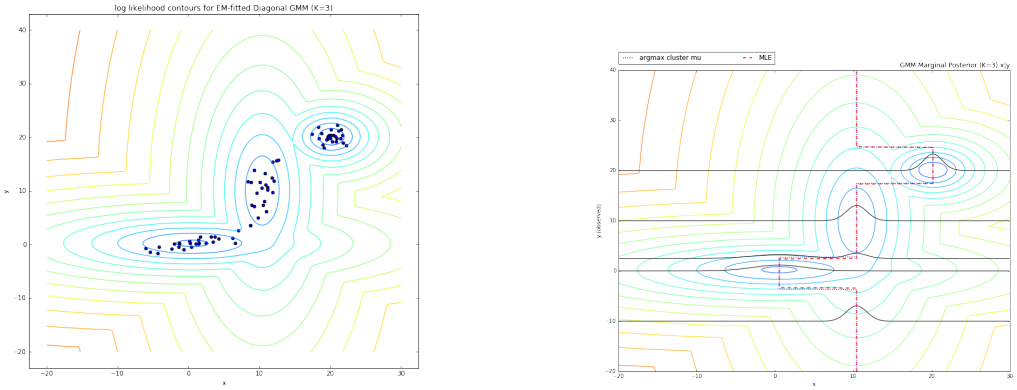


Figure 8: GMM model for $D = 2$ and $K = 3$ fitted to a dataset of three normals (some non-diagonal). In the plot on the right, the marginal posterior mode and argmax estimates for every given value of y are given. The black lines are superimposed marginal posterior likelihood plots for select y values.

Notice that the mode and argmax algorithms produce the same result for both $K = 2, 3$ in Figures 8 and 9. In Figure 9 in particular, we notice that the unstable situation of two equivalent normals still gives rise to near-identical mode and argmax estimates. This is because of the extremely light tails of the normal distribution. If one of the normal distributions k is more likely than some other j given some observation ($\pi_i p(\mathbf{y}|C_i) > \pi_j \mathbb{P}(\mathbf{y}|C_j)$ - note this proportional to the inequality $p(C_i|\mathbf{y}) > p(C_j|\mathbf{y})$), it is likely that it is significantly greater. In other words, most of the time, one of the normals dominates.

Note that both the mode and argmax estimates differ from the expectation of the GMM, which is the mode for a weighted sum of normals (but not their mixture). In Figure 9, this expectation runs along the center of the diagram and is stable, where as the mode and argmax estimates for the GMM are unstable relative to the fit.

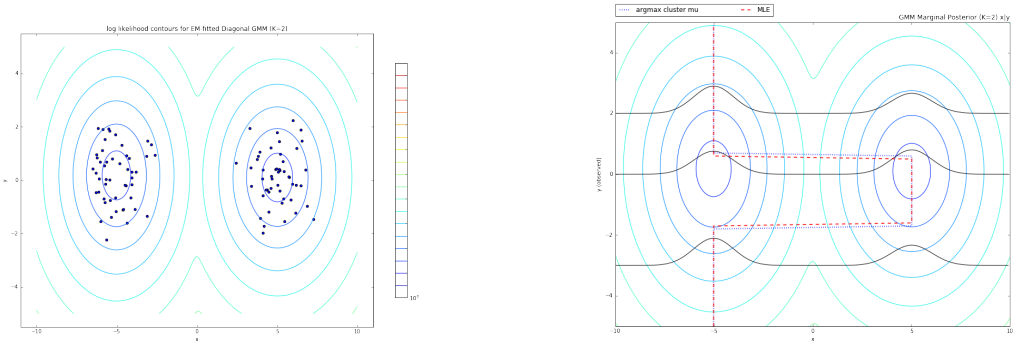


Figure 9: GMM model for $D = 2$ and $K = 2$ fitted to a dataset of two normals (both spherical). The plot here follows the same format as that of Figure 8, but here the likelihoods are nearly symmetric.