Self-supervised Epigenomic Imputation using Autoencoders

Runjia (Luke) Li

Introduction

Large-scale next-generation sequencing projects [1,2] have generated rich collections of epigenomic data across different samples and epigenetic marks in the form of coordinate-based data tracks. However, due to cost constraints, it remains unrealistic to exhaustively obtain data tracks for all sample-mark combinations. To overcome this problem, many high-performance methods have been proposed to impute missing data tracks by leveraging information from available tracks. ChromImpute [3] uses an ensemble of regression trees to predict the data track of a specific sample-mark combination, taking information from (a) other marks of the target sample and (b) the target mark of samples similar to the target sample. Despite its great accuracy, it is only able to impute different tracks independently from different training sets. PREDICTD [4] and its successor AVOCADO [5] decompose the epigenomic data tensor (with the axes corresponding to samples, marks and genomic positions) into low-ranked matrices; through an approximate reconstruction of the data tensor, they can jointly impute all missing tracks but with high computational resource expenditure. We propose a novel and resource-efficient approach that enables joint imputation of multiple epigenetic data tracks through denoising autoencoders [6], neural networks that allow prediction of missing data by learning their relations with observed data. Our approach, while having varying performance across different epigenetic marks, is somewhat able to capture sample-specific information that agrees with imputations from ChromImpute.

Methods

Data description

We obtained 25-bp resolution -log10 p-value tracks for 8 primary marks (H3K4me1, H3K4me3, H3K36me3, H3K27me3, H3K9me3, H3K27ac, H3K9ac and DNAse) on chromosome 21 for all NIH Roadmap [2] samples from the ChromImpute reference website (http://www.biolchem.ucla.edu/labs/ernst/ChromImpute/) [3]. There are 21 samples with all 8 marks profiled, and we used 19 of them as the training set (E004, E005, E006, E007, E008, E017, E114, E116, E117, E118, E119, E120, E121, E122, E123, E124, E125, E126, E127) and the rest 2 (E003, E128) as the testing test. We then attempted to impute the following missing tracks: DNase track for E020 (iPS), H3K9ac and DNase tracks for E071 (brain hippocampus middle), and H3K9ac track for E092 (Fetal stomach).

Model architecture

Our model is a 3-layer fully-connected neural network, with the 328 neurons in the input layer, n (a number to be selected) neurons and ReLU activation [7] in the second layer, and 328 neurons in the output layer.

Model training

The model takes as input a 8 by 41 matrix consisting of -log10 p-values of the 8 primary marks on 41 consecutive 25-bp genomic bins, which is for convenience flattened in row-major order into an equivalent 328-element feature vector. During training time, we concatenated the tracks for all 19 samples in the training set and randomly sampled 100,000 starting positions, resulting in 100,000 matrices. Then for each matrix, we randomly zero-masked a number of rows to simulate missing marks in a sample before feeding the matrix into the model. The probability that each row was masked followed a Bernoulli distribution with parameter 0.3 (that is, on average each matrix will have 2.4 missing marks). We used mini-batch gradient descent to train the model and ADAM [8] as the optimizer, with batch size 9000 and initial learning rate 0.01. The model is deemed self-supervised because its expected output should be the unmasked input: for a batch of input matrices, the total cost of the model was defined as the total root-mean-square error between all pairs of input and output (the 328 neurons in the output layer was reshaped back to a 8 by 41 matrix) matrices, plus the l2 regularization term; thus, we hypothesized that through minimization of the cost, the model should be able to not only reconstruct the non-missing rows in the input matrix but also fill the missing rows, thereby imputing tracks. We performed 5-fold grid search cross-validation and obtained optimal hyperparameters: 2000 neurons in the second layer and l2 regularization coefficient of 0.001.

Model testing

For each sample in the testing set, we performed the following procedure: (1) we zero-masked the entire track for one mark, (2) we obtained all windows of 41 25-bp bins and fed the matrix corresponding to each window into the trained model, (3) for each 25-bp bin of the masked track, its imputed value was computed as the average predicted value of this bin across all windows that contain this bin, (4) For each of the 8 marks we repeated steps 1-3, and for each of the missing tracks (as in Data Description), we repeated steps 2, 3.

Results

Training performance

On average, the model was able to converge under 20 minutes and used less than 15G memory. For the training set, the Pearson correlation between all masked values and their corresponding imputed values is 0.706. Despite this, as shown in figure 1(left), our model often underestimated the true values. In addition, the Pearson correlation is significantly different across marks (table 1).

Testing performance

We observed even more significantly varying performance of our model across different marks, in both samples of the testing set, as presented in figure 1(middle and right) and table 2. For some marks (e.g. H3K27me3 and H3K9me3) our model was showing minimal predictive performance, while some other marks (e.g. H3K4me3 and H3K9ac) seemed to be very easily imputed by our model.

Imputations

We compared imputed tracks of our model to those of ChromImpute. Visually, while only a handful of sparsely-distributed peaks were predicted by our model, they show great agreement with the predicted peaks of ChromImpute (figure 2, top). In addition, the imputations of our

model had a certain extent of sample-specificity. Shown in figure 2 (bottom), some peak topologies distinct to E071 were predicted by both our model and ChromImpute.

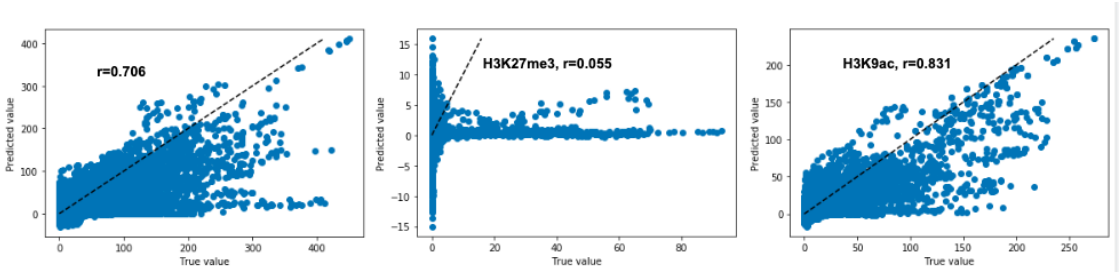

Figure 1, Predicted values vs true values (dashed lines showing unit slope) (left) for the entire training set, (middle) for H3K27me3 of sample E003, and (right) for H3K9ac of sample E003

	H3K27me3	H3K36me3	H3K4me1	H3K4me3	H3K9ac	H3K9me3	DNase	H3K27ac
Training set	0.0555	0.0442	0.4841	0.7691	0.8308	0.0168	0.4631	0.7987
E003	0.0938	0.1296	0.3492	0.7625	0.7493	0.06413	0.3818	0.5630
E128	0.0396	0.1238	0.4756	0.7429	0.8057	0.0183	0.3665	0.7750

Table 1, Pearson correlations between predicted and true values in the training and testing sets

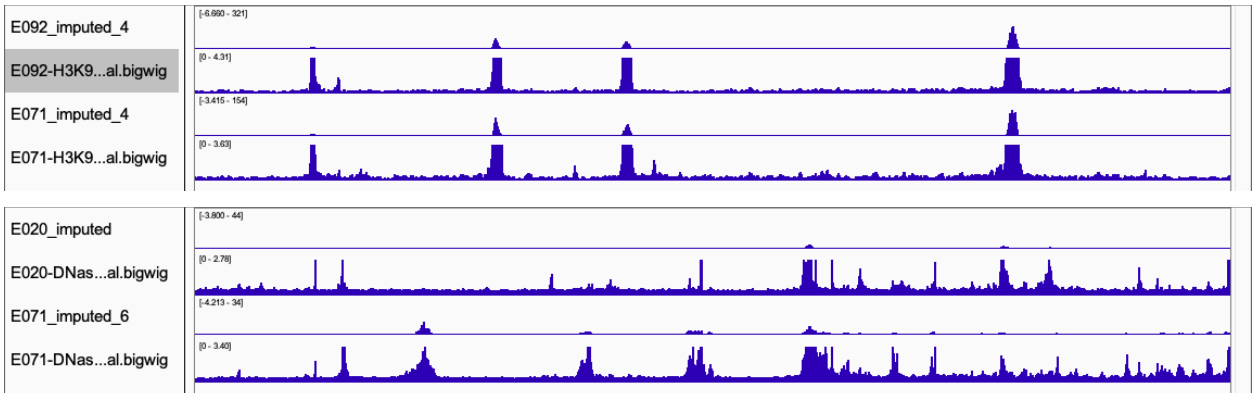

Figure 2, Our model's imputed tracks (odd-numbered rows) and corresponding ChromImpute tracks (even-numbered rows). Top: H3K9me3 for E092, E071. Bottom: DNase for E020, E071

Discussion

Our imputation model demonstrates that while neural networks are powerful in general, greater efforts are required to understand their performances. As of now, it is unsure what is the exact cause of our model's varying performance across different marks, despite the fact that the marks were masked in the training set with equal probabilities. In addition, the model would benefit from the following improvements: (a) using a more complex architecture (for example, more stacked layers), (b) tuning of more hyperparameters besides number of neurons and regularization coefficient, and (c) incorporating more samples into the training set, including those with missing tracks. We envision that given the model's current performance under an extremely simple architecture, the novel approach of using autoencoders for imputation tasks in bioinformatics will receive increasing attention.

References

[1] ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature* 489, 57-74 (2012).

[2] Kundaje, A., et al. Integrative analysis of 111 reference human epigenomes. *Nature* 518, 317-330 (2015).

[3] Ernst, J., & Kellis, M. Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. *Nat Biotechnol* 33, 364-376 (2015).

[4] Durham, T.J., Libbrecht, M.W., Howbert, J.J. et al. PREDICTD PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition. *Nat Commun* 9, 1402 (2018).

[5] Schreiber, J. et al. Multi-scale deep tensor factorization learns a latent representation of the human epigenome. bioRxiv (2019).

[6] Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proc. 25th International Conference on Machine Learning* 1096–1103 (2008).

[7] Glorot, X., Bordes, A. & Bengio. Y. Deep sparse rectifier neural networks. In *Proc. 14th International Conference on Artificial Intelligence and Statistics* 315–323 (2011).

[8] Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representations* 1–15 (2015).