Predicting chromatin contact through histone modification data and evaluating different models

Qing Liu
Johns Hopkins University
3400 N. Charles Street
Baltimore, Maryland 21231, USA
qingliu@jhu.edu

Yuan He
Johns Hopkins University
3400 N. Charles Street
Baltimore, Maryland 21231, USA
yhe23@jhu.edu

Abstract

Genome-wide mapping of the dynamic 3D chromatin structure could provide important insights into gene regulation studies. While experimental procedures are currently costly, using the widely available histone modification data and computational methods to predict the contacting domain could be a much more economical way. In this project, we used public Hi-C data and ChIP-seq data of three histone modifications including H3K4ME3, H3K4ME1 and H3K27AC from lymphoblastoid cell lines (LCLs) to built several classifiers using six different machine learning algorithms: logistic regression, random forests, support vector machines (SVM), Naive Bayes, Bayesian additive regression trees (BART), and neural network. We then applied them on the task of predicting chromatin contact status. According to the results, we found that all models performed well using the test set from the same chromosome as the training set (AUC all above 0.80 except for Naive Bayes), while more complex models such as the sum-of-tree methods and neural networks held their good performance in the test set from different chromosome.

1 Introduction

The chromatin has been proved to form 3D structures through looping and folding. In recent years, many technologies have been developed to experimentally map the chromatin contact loci, including 3C, 4C and HiC. Among these methods, HiC provides the most comprehensive view of genome-wide chromatin interactions. However, most of these methods are currently difficult and costly.

Histone modifications, with the functional effect of controlling accessibility of chromatin, may provide useful information about chromatin interactions. At the same time, the technology using chromatin immuno-precipitation followed by high-throughput sequencing (ChIP-seq) to profile histone modifications is much mature. It has been shown in several studies that there exist association between HiC contacts and histone modification peaks. Here, we tested this hypothesis using six machine learning algorithms, including logistic regression, random forests, support vector machines (SVM), Naive Bayes, Bayesian additive regression trees (BART), and neural network. While digging into the association between HiC contacts and histone modification sequence reads, we also want to compare the performance of each algorithm in this task, and how the different feature sets we selected can lead to different results.

2 Method

For this project, we first extracted information from the published histone modification data and HiC data, defined positive and negative classes with 1/0 labels, and prepared one training set and two test sets. For the purpose of including local information regarding the distances between the histone modification peaks to the interested locus and evaluate how this information would help to improve the performance, we prepared three different feature sets. We then built six classifiers using different machine learning algorithms for each feature set, and tested

their performance by computing the Receiver Operating Characteristic (ROC) curve on corresponding test sets. For the model training and testing, we generally used packages from MatLab Statistics and Machine Learning Toolbox and R packages.

2.1 Data preparation

Data source and preprocess: The data used in this project was dowloaded from a published paper. They provided ChIP-seq data of 3 histone modifications (H3K27AC, H3K4ME3, H3K4ME1) in lymphoblastoid cell-lines (LCLs) gathered from 75 unrelated individuals of the Yoruba (YRI) population. Also, they provided a HiC interaction matrix generated in a reference LCL (GM12878). The raw reads count of histone marks was preprocessed through quantile normalization. The normalization and peak calling of HiC data have been done earlier in our lab.

Define contact hubs and negative loci from HiC data: To define contact hubs, we first filtered out entries that are not significant in the HiC interaction matrix using the correlation coefficient threshold of 0.4. We then picked out the top 800 chromatin anchors according to their interaction frequency on the specific chromosome, and labeled them as 1. To define negative loci, we chose the 800 chromatin anchors with fewest but non-zero interactions and labeled them as 0. The reason for excluding chromatin anchors associated with no detectable interactions was to remove the bias toward mappable genome and GC-rich sequences.

Summarize histone modification data for the loci: For each locus identified in the positive and negative set, we summarized the local pattern for each histone mark by averaging the sequence reads from all individuals at range a) 0 - 50 kb b) 50 - 100 kb c) 100 - 150 kb distance from the locus center, which formed our *feature set 1* consisted of 9 variables. Our *feature set 2* was the sum up version of the first set, where we averaged the total sequence reads for each histone mark over the full 300 kb window around the locus and this produced 3 variables. This histone modification pattern representing strategy was adopted from the *Genome Biology* paper, and we suspect its performance since it failed to include more specific information regarding the distance between the histone mark peaks and the locus center. In our *feature set 3*, we picked out the top three important variable from the *feature set 1* reported by the random forest algorithm, which were H3K4ME3 (bin1: 0 - 50 kb), H3K4ME3 (bin2: 50 - 100 kb), and H3K27AC (bin1: 0 - 50 kb). We were curious to see whether these three selected variables could achieve as good performance as the full set.

Generate training set and test sets: We first focused our study on chromosome 22. From the 800 positive loci and 800 negative loci, we randomly picked out 640 loci from each class and put them together to form our *training set* containing 1280 entries. The rest loci from both classes formed our *test set 1*, containing 320 entires. We then introduced a *test set 2* using all 1600 loci from chromosome 21. The purpose of this test set was to evaluate the generalization power of our classifiers when applied to different chromosome.

2.2 Logistic regression

Logistic regression is a special case of the *Generalized Linear Models* family. It uses *Bernoulli distribution* in binary classification tasks, and the predicted values are probabilities restricted to (0,1), which is generated by passing the linear combination of the input through a sigmoid function. We used the MatLab general linear model function `glmfit` and `glmval` to train the model and predict for test dataset, by setting the *distr* to `binomial` and *link* to `logit`. Besides the estimated coefficient for the model based on maximum likelihood estimation, the function also returns `stats` for the fitting, which contains the coefficient p-values, indicating the predictive power of the features included in the current model.

2.3 Random forests

Random forests is an ensemble learning method. It is based on decision trees learning and combines the *bagging* idea and the random selection of features. This method has been broadly used in chromatin structure prediction tasks. We used the MatLab `TreeBagger` class and relevant functions to train the model and predict for test dataset. During training, by setting the `OOBPredictorImportance` (or `OOBVarImp` in earlier version) to `on`, the model will store out-of-bag estimates of feature importance in the ensemble, by measuring the increase in prediction error

if the values of that variable are permuted across the out-of-bag observations. By looking into the `OOBPermutedVarDeltaError` property of the resulting TreeBagger object, we can find out the importance for each predictor variable.

2.4 SVM

An SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other class. The best hyperplane for an SVM means the one with the largest margin between the two classes. It works by solving the quadratic programming problem, and it is usually computationally simpler to solve the dual form. For nonlinearly separable data, SVM can still work by performing the *kernel trick*, which implicitly mapping the inputs into high-dimensional feature spaces. We used the MatLab `fitcsvm` function to construct and train the binary support vector machine classifier using the default `ISDA` (iterative single data algorithm) solver for optimization, and tested the performance of *linear kernel*, *polynomial kernel*, *sigmoid kernel* and *RBF kernel*.

2.5 Naive Bayes

Naive Bayes is a Bayesian approach that enumerate through the possible conditions for the outcome, then use the Bayes rule to calculate the probabilities. R package `e1071` provides a function called `naiveBayes` to compute the conditional posterior probabilities for dependent variables. Since we only have nine variables, we don't need sparse predictors, so we set the `laplace` parameter to be 0. `type` was set to be "raw" to obtain probabilities for each class. By using `predict`, we got the predicted probabilities for data in the test sets.

2.6 BART(Bayesian Addictive Regression Tree)

BART linearly combines the posterior distribution from single Bayesian tree. The difference between BART and sum-of-trees model is that it imposes a prior to weaken the individuals trees, and also fits the model by iteratively backfitting MCMC. We used the R package `BayesTree`, which provides the function `bart` for building the BART model. Since BART uses Bayesian methods, `bart` provides parameters to adjust for priors for the trees. Also, this used an MCMC approach that includes parameters for number of burn-ins and number of sampling afterwards. Since BART is a modification of sum-of-tree approach, the adjustable parameters also include number of trees, tree cutoffs etc.

2.7 Neural network

Neural network provides an approach to construct non-linear structure of the potential associations between features and the outcome. It uses hidden layers to support more than one layer of association, and thus makes it possible to build complex models. We used the R package `neuralnet` to build the neural network model. It provides the `neuralnet` function for model construction.

2.8 Figure plotting

To calculate the false positive rate and true positive rate for the ROC curve, we used `perfcurve` in Matlab and `AUC` package in R. We first store the false positive rate and true positive rate at every cutoff, then use the `ggplot2` package in R to represent the figures.

3 Result

3.1 Data exploration

Before building the models, we first checked the distribution of data points in their feature space, as well as the correlation between features. As shown in Fig 1a, Fig 1b and Fig 1c, the green lines represent the group of fragments with high HiC contact, and the red lines represents the fragments with little evidence of HiC contact. Three different line types stand for the three bins for each histone modification peaks. The reads are after log2 transformation. H3K4ME3 shows the best ability to separate the two groups, while H3K4ME1 and H3K27AC show less clear separation. This indicate that there does exist association between HiC contact and histone modification peak reads. Fig 1d shows the correlation between nine features. It is reasonable to see here that the three features for the same histone modification show stronger correlation than features from different types. The correlation between features indicated that the assumption of independent variables may not be met for models such as logistic regression and Naive Bayes.

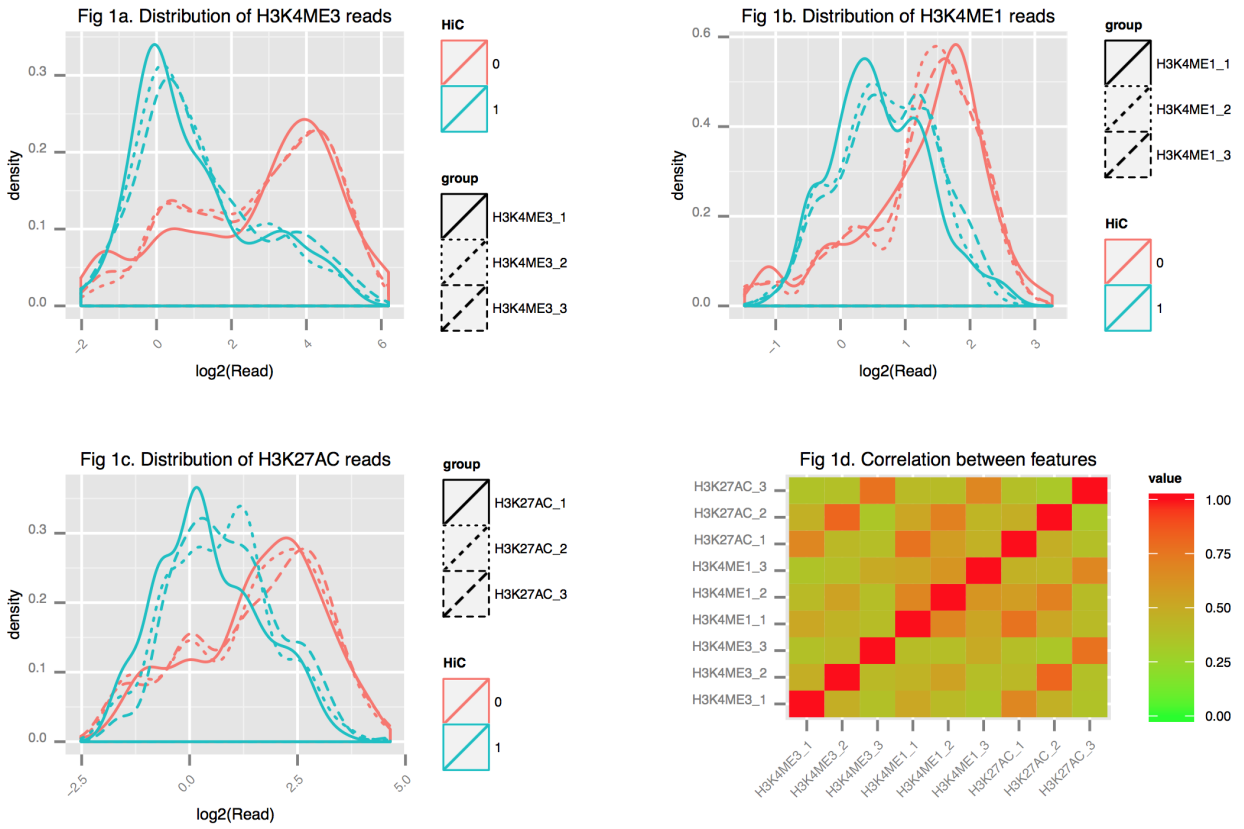

Figure 1: Data exploration

3.2 Logistic regression

From the `stats` returned by the `glmfit` function, we looked into the coefficient p-values which implies the importance of the features, and in *feature set 1*, all features are significant except H3K4ME1-2 and H3K27AC-2. This result is somehow controversial with our knowledge about the importance of H3K4ME1 and the result reported by random forests model.

According to the test result shown in Fig 2, logistic regression's performance is not very good in both test sets and all feature sets, which is expected because the feature space should not be able to be easily linearly split for our classification task. Besides, in *Data exploration* section, we also saw there are strong correlation between features, which may lead to unreliability of feature ranking and solutions.

3.3 Random forests

In this model, we first tested the optimal number of decision trees by plotting the *Out-of-Bag Classification Error* over the number of grown classification trees (data not shown here), and found that the error became stable after 50 trees were grown.

We then looked into the reported feature importance in corresponding feature set (Table 1). The result in *feature set 1* (IMP1 column) indicated that for each histone mark type, the sequence reads in a closer window to the locus are generally more predictive, and the overall power of H3K4ME1 is slightly higher than H3K4ME3, both better than H3K27AC. According to the result in *feature set 2*, when the detailed distance information was excluded, among the three histone modification types, H3K4ME1 is most predictive, the second one is H3K4ME3, and H3K27AC is the least, which is in good agreement with the *Genome Biology* paper. However, when looked into *feature set 3*, we saw that H3K4ME3-1 and H3K4ME3-2 are both more important than H3K4ME1-1, which implies that the distance information could affect histone marks' predictive power.

Table 1: Feature Importance

feature	IMP1	IMP2	IMP3
H3K4ME3-1	2.59		6.56
H3K4ME3-2	3.18	4.45	6.35
H3K4ME3-3	1.62		NA
H3K4ME1-1	2.86		4.50
H3K4ME1-2	2.57	4.98	NA
H3K4ME1-3	2.24		NA
H3K27AC-1	2.14		NA
H3K27AC-2	1.97	2.42	NA
H3K27AC-3	1.83		NA

As shown in Fig 2, random forests is the best model when applied to *test set 1* in all three feature set groups, but in *test set 2*, its performance dropped dramatically, especially in *feature set 2* and *feature set 3*.

3.4 SVM

For this model, among all kernels we tested, *linear kernel* gave relatively poor prediction accuracy, which is expected since we already knew from the *logistic regression* model that the feature space cannot be linearly split for our classification task. And as many people suggested, *RBF kernel* gave the best prediction accuracy, so we used this kernel to present the model's test results.

In Fig 2, we saw that *SVM with RBF kernel* could generally achieve good performance in *test set 1*, but not so good in *test set 2*.

3.5 Naive Bayes

According to the ROC result, Naive Bayes performance is the worst among almost all tests, which might be caused by the strong correlation between features, or the poor match between the distribution model (Gaussian) with the real distribution.

3.6 BART

In trees' priors, we tuned the `power` parameter which controls the power for tree prior and found little difference in the resulting AUC value. We also adjusted the `base` parameter and found the default value of 0.95 could already achieve optimal. Other parameters were set as the default, such that first 100 steps were treated as burn-in and thus discarded. After this, 1000 draws were obtained for sampling. In test data, this model produced a matrix of 1000 rows and *number of data points* columns, where each row represented a draw from the model. The ij^{th} cell in the result represents the i^{th} draw for the j^{th} data point. So we averaged through the 1000 rows to obtain the probability of achieving 1 for each data point in the test set.

The resulting ROC in Fig 2 shows that at least in the first two feature sets, BART can have pretty good performance, which is in good agreement with the *Genome Biology* paper.

3.7 Neural network

Among the parameters tunable in the `neuralnet` function, we consider three important ones: number of hidden layers, stopping criteria and algorithm. While adjusting the number of hidden layers, we observed that as the number of layers increase, the time for building the model increases significantly. We also found that models with 5 layers has an increase of 15.6% in the AUC value compared with the models with 1 hidden layer. Among the three algorithms we tried, *back propagation*, *resilient back propagation*, and *modified globally convergent*, the first two didn't show much difference in resulting AUC value, the third one failed in our dataset, probably because of insufficient data points. Thus to present the model's final performance, we used back-propagation with 5 layers to build the neural network model.

According to Fig 2, though the model's output is not very good in *test set 1*, to our surprise, its AUC value in *test set 2* is relatively high, which is hard to explain due to the model's *black box* property.

4 Discussion

Our result shows that there does exist strong association between local histone modification read counts (H3K4ME3, H3K4ME1, H3K27AC) and HiC contact frequency. However, since there could be other factors influencing HiC contacts, for example, the chromosomal activity, sequence GC content etc., the model trained on one chromosome achieved poorer performance in another chromosome. This was reflected by the overall lower AUC value in *test set 2*.

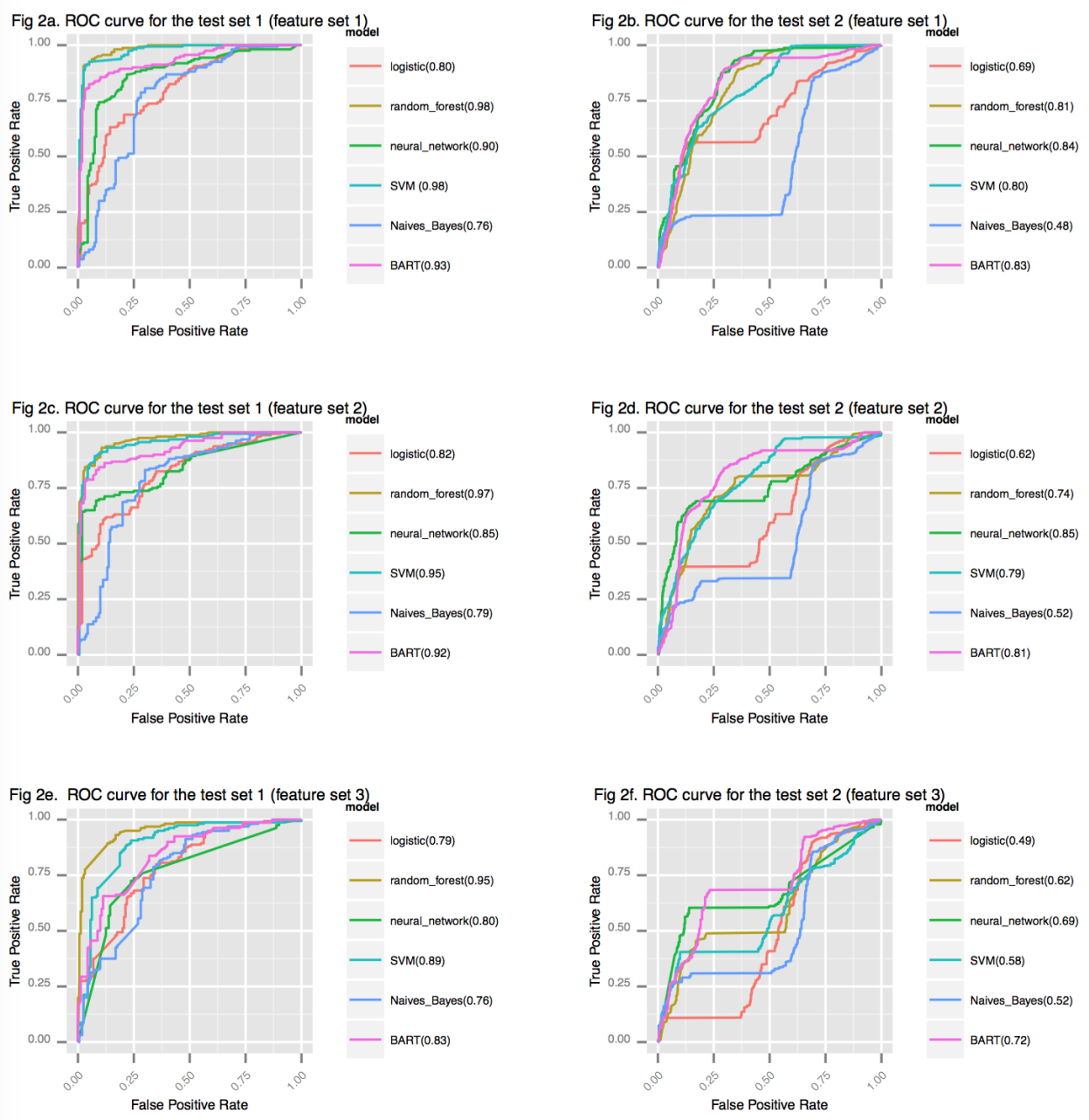

Figure 2: Model performances using ROC curve

In order to obtain more general applicable models for prediction, we suggest to leverage data from all chromosomes and add in more biologically meaningful features.

Among the six models we studied, Naive Bayes and logistic regression's results are the worst. This is reasonable considering that there does exist strong correlation between the features we used and the feature spaces are highly possible to be non-linearly separatable. In addition, the poor match between the distribution model (Gaussian) with the real distribution in the Naive Bayes model could also be a non-trivial factor that affected its performance. The comparison between models could lead to the conclusion that those complex non-linear models can perform better in this task.

Overall, *feature set 1* could achieve the best performance. The *feature set 2* which was the sum

up version of *feature set 1* produced slightly worse output. This indicates that the distance in between could also have impact on the association between the histone mark and the HiC contact, but the impact is not as significant as the sequence reads. The *feature set 3* which included partial information from *feature set 1* showed relatively poor performance, especially in *test set 2*. We think this is reasonable since this set does miss certain amount of information, and the three features were selected based on the random forests model trained on chromosome 22 data, which could possibly report different importance ranking when applied to another chromosome.

Comparison to proposal

Must achieve: We derived biologically reasonable and mathematically informative training/test data, including features that describe peak reads and distances in between, as well as output labels from the HiC data. We constructed totally six classifiers for each set of features, and did fine tuning of the parameters to achieve the best performance of each model. For each model, we compared their test result, and generally explained their pros and cons combining the mathematics behind the algorithm. Therefore, we have fully completed this part.

Expected to achieve: We prepared three different feature sets to compare the predictive power of each individual histone marker and different combination of them. We have pulled the regulatory annotation for chromatin loci used in this project, but don't have time to generate a new feature set using this information. We will probably complete this second part of our *Expected to achieve* for our lab research project in near future.

Would like to achieve: We did apply our models learned from chromosome 22 to chromosome 21 dataset and compared their generalization performance. We will search for other publicly available histone modification data and HiC data to do further tests.

Acknowledgments

We would like to thank Dr. Ilya Shpitser for his teaching and guidance for the Introduction to Machine Learning course, and his suggestions for our project. Also, the two teaching assistants, Tuo Zhao and Hainan Xu, also provided a lot of help. Finally, we want to thank Dr. Alexis Battle for her advices and supports for our work.

References

Jon-Matthew Belton, Rachel Patton McCord, Johan Harmen Gibcus, Natalia Naumova, Ye Zhan, and Job Dekker. 2012. Hi–c: a comprehensive technique to capture the conformation of genomes. *Methods*, 58(3):268–276.

Hugh A Chipman, Edward I George, and Robert E McCulloch. 2010. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, pages 266–298.

Jesse R Dixon, Inkyung Jung, Siddarth Selvaraj, Yin Shen, Jessica E Antosiewicz-Bourget, Ah Young Lee, Zhen Ye, Audrey Kim, Nisha Rajagopal, Wei Xie, et al. 2015. Chromatin architecture reorganization during stem cell differentiation. *Nature*, 518(7539):331–336.

Fabian Grubert, Judith B Zaugg, Maya Kasowski, Oana Ursu, Damek V Spacek, Alicia R Martin, Peyton Greenside, Rohith Srivas, Doug H Phanstiel, Aleksandra Pekowska, et al. 2015. Genetic control of chromatin states in humans involves local and distal chromosomal interactions. *Cell*, 162(5):1051–1065.

Jialiang Huang, Eugenio Marco, Luca Pinello, and Guo-Cheng Yuan. 2015. Predicting chromatin organization using histone marks. *Genome biology*, 16(1):1–11.

Laura Toloşi and Thomas Lengauer. 2011. Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, 27(14):1986–1994.