DNA Sequence Feature Selection for Intrinsic Nucleosome Positioning Signals Using AdaBoost

Yu Zhang, Xiuwen Liu, Justin Fincher
Department of Computer Science
Florida State University
Tallahassee, FL 32306, U.S.A.
{yzhang,liux,fincher}@cs.fsu.edu

Jonathan H. Dennis
Department of Biological Science
Florida State University
Tallahassee, FL 32306, U.S.A.
dennis@bio.fsu.edu

ABSTRACT

Recent genome wide experiments indicate that DNA sequences themselves strongly influence nucleosome positioning as an intrinsic cell regulatory mechanism. While some sequence features are known to be nucleosome forming or nucleosome inhibiting, there is no systematic study on identifying optimal sequence features for quantitatively modeling of DNA binding affinity. In this paper, we propose a computationally efficient method of identifying a (small) number of sequence features for intrinsic nucleosome positioning. By using a modified version of AdaBoost, the proposed method is able to identify features to be used with a strong classifier to categorize nucleosome forming and nucleosome inhibiting local DNA sequences. Experimental results on extensive datasets show that the resulting classifiers give typically better prediction performance than the existing discrimination models on all the tested datasets with a much smaller number of features.

Categories and Subject Descriptors

J.3 [**Biology and Genetics**]: Computational Biology; I.5.1 [**Models**]: Statistical Modeling; I.5.2 [**Design Methodology**]: Classifier—*AdaBoost*

General Terms

Nucleosome positioning

1. INTRODUCTION

DNA of eukaryotic cells is organized into repeating nucleosomes that are connected by linker DNA of variable length [6, 7]. As DNA located on a nucleosome tightly warps around the associated histone octamer, the accessibility of the coded sequence is significantly limited compared to the state when the DNA is open [6, 7, 11]. Therefore, the positions of nucleosomes play a fundamental role in regulation of genes and other functional activities in a cell [7]. While nucleosome positions are determined by a number of factors (including ATP dependent chromatin remodellers, site-specific DNA-binding proteins (see, e.g. [11])), recent genome wide studies [7, 4] have demonstrated that DNA sequences themselves also play an important role by enhancing or reducing their binding affinity to nucleosomes, thus encoding their intrinsic preferences of nucleosome positions; similar nucleosome position preferences have been observed *in vitro* [10] as well as *in vivo* [11].

These experiments have led to several computational models [5, 10, 9, 4, 13] that try to model the relationship between the nucleosome positions and the DNA sequences quantitatively based on the sequences and measured nucleosome positions. These models, based on either aligned DNA sequences, or just nucleosome forming and inhibiting local regions, differ in prediction accuracy (e.g. [13]), where support vector machine based on models appear to be most accurate. However none of them addresses explicitly local DNA sequence features for nucleosome forming and inhibiting. While there are known nucleosome forming and inhibiting sequences due to their physical properties (e.g., bendability), there is no systematic method to identify the sequence features using the available genome wide datasets of nucleosome forming and nucleosome inhibiting sequences.

In this paper, we propose a computational model that can be used to integrate different promoting and suppressing factors of nucleosome positioning in particular regions of DNA sequences. Computational experiments on several nucleosome positioning datasets show that the proposed model gives better prediction performance on all of them than a support vector machine based model [4]. In addition, our model suggests that a small number of chosen features often provide a better performance than using all the features.

The rest of the paper is organized as follows. In Section 2 we briefly introduce the nucleosome positioning model. Section 3 presents the AdaBoost algorithm for feature selection and classifier construction and a method for identifying common features. Section 4 shows the experimental results on several datasets and compares the results with other existing methods. Section 5 concludes the paper with a brief summary.

2. AN INTRINSIC MODEL

As nucleosome positions in eukaryotic cells are determined by a number of factors, Segal and Widom [11] propose an equilibrium model for nucleosome positioning that integrates different factors in a unified framework. Motivated by the study, we propose the following computational model for the intrinsic affinity of nucleosome positioning of a local DNA

ACM-BCB 2010, Niagara Falls, NY, USA.

sequence,

$$A_{int}(\vec{x}) = \sum_{i=1}^{I} \alpha_i \, h(f_i(\vec{x}), b_i, s_i), \quad (1)$$

where $\vec{x}$ is a local DNA sequence under consideration, $f_i(\vec{x})$ is a feature derived from sequence $\vec{x}$, α_i is the weight coefficient for feature f_i; when $\alpha_i > 0$, the presence of a_i enhances the affinity of nucleosome positioning while when $\alpha_i < 0$, the presence of a_i suppresses the affinity of nucleosome positioning, and I is the total number of features used. Here $h(x, b, s)$ is the hyperbolic tangent function, given by $h(x, b, s) = \frac{1-e^{-2\,s\,(x-b)}}{1+e^{-2\,s\,(x-b)}}$, where b specifies a center for the feature, and s controls the steepness of the function. In particular, if $s \to \infty$, it becomes a step function. Note that the actual positioning of nucleosome *in vivo* depends on $A_{int}(\vec{x})$ as well as other extrinsic factors. In this model, both promoters and suppressors are specified explicitly. In contrast, a commonly used support vector machine model uses $f(\vec{x}) = \sum_{i=1}^{I} \alpha_i \, g(\vec{x_i}, \vec{x})$, where $g(\vec{x_i}, \vec{x})$ is a kernel function that measures the similarity/distance between $\vec{x}$ and given training example $\vec{x_i}$; here α_i is positive if $\vec{x_i}$ is a positive training example and α_i is negative if $\vec{x_i}$ is a negative training example; the α_i's are given when a support vector machine is trained. Note that training samples that are non support vectors will have 0 as their α_i and thus are excluded from the model. It is clear that in a support vector machine model, the prediction is based on a linear combination of positive and negative examples and the sequence features are not identified explicitly.

3. OPTIMAL FEATURE IDENTIFICATION

The key question is how to identify optimal features and the associated parameters given a genome wide dataset. Note that the model in Equation (1) is continuous and in general can be used to fit continuously nucleosome positioning measurements. Here, following most other computational studies on nucleosome positioning, we convert the general nucleosome positioning problem into a binary classification one. That is, based on the measurements, a set of DNA positions with strongest affinity is identified as positive examples and another set of DNA positions with weakest affinity is identified as negative examples. The problem is to predict whether a particular DNA sequence is positive (nucleosome forming) or negative (nucleosome inhibiting).

In this classification problem, as the parameter s of the hyperbolic tangent function is not essential, and it becomes a step function as we set s to ∞. Under this setting, the nucleosome positioning problem can be formulated as follows: given a set of positive examples and a set of negative examples of local DNA sequences (such 50 base pairs), identify the most effective factors and estimate their coefficients among the given features in predicting DNA sequence labels.

The given problem can be solved by a forward stagewise additive model; if we choose an exponential loss function for optimization, this leads to the AdaBoost algorithm [3]; a similar problem has been studied for face detection [12]. Following the algorithm used in face detection, the key problem is how to find the optimal feature for i-th component, which, as shown in [12], can be solved efficiently through one pass of the training examples.

To proceed, let $f_1(\vec{x}), \ldots, f_K(\vec{x})$ be all the sequence features to be considered. As in [9, 4], we use all k-mer features with $1 \le k \le 6$ and $K = 2772$. Given a set of local DNA sequences with labels, $(s_1, y_1), \ldots, (s_n, y_n)$, where $y_i = +1$ if s_i is a nucleosome forming sequence and $y_i = -1$ if s_i is a nucleosome inhibiting sequence. The key advantage of AdaBoost feature selection is its computational efficiency of learning features [12]. This is achieved by using a greedy procedure to select features one by one. The influence of features that have chosen so far is encoded implicitly in the weights given to each training sample. When a training sample is misclassified, its weight becomes higher; on the other hand, when a training sample is classified correctly by the current classifier, its weight becomes smaller. Thus more difficult training samples will have more influence on the features to be chosen. The final prediction is given by Equation (1), where the features are given by the optimal features selected sequentially and the α_i's are also computed accordingly.

3.1 Common Sequence Features

Factors due to intrinsic nucleosome positioning of different cells of the same organism should be identical as their DNA sequences are the same. However, the computational models derived from different datasets are different typically. To overcome this inherent problem, we propose an additional common sequence feature identification procedure from the available cell types of a given organism, e.g. humans in our case.

To identify the common sequence features, we rank all the features based on their average relative ranking. For each cell type with a particular training set and validation set division, we apply the above AdaBoost algorithm and it selects features sequentially one by one. The importance of a feature is given by its rank in the selected feature list. As a feature may be selected more than once due to the nonlinear nature of the model, the rank of its first appearance is used as its rank. Also some of the features may not be chosen and we give all these features the same and the largest rank. The rankings of all the training-validation divisions from all the available cell types are then averaged for each feature and then all the features are sorted. The feature with the least rank is the first common feature to be selected and more features are selected based on the their average rankings.

After a given number of features are chosen, we then learn strong AdaBoost classifiers using only the chosen features for different cell types using 10-fold cross validation or other procedures; if preferred, other classifiers such as a support vector machine can also be used but with the chosen small number of features. As shown by the experimental results (see Section 4.3), for the given datasets of human cells, as few as five common features give good prediction for all the ten cell types. This suggests that intrinsic nucleosome positioning can be realized with only a small number of sequence features, making it more biologically plausible.

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup

We have applied the AdaBoost feature selection and the AdaBoost learning algorithm on all the ten datasets used in [4]. The Dennis dataset [1] consists of three microarrays; each microarray contains about 120,000 probes, which cover 25 kb regions upstream of 42 genes, using 50-mer probes tiled every 20 bases. In the dataset, there are six measurements

for each 50-mer, three times of the sequence itself and three times of its reverse complement. Nine other datasets are from [8], as they are used in [4]; among the nine datasets, it appears that Ozsolak A375 has a different characteristic than the other eight datasets (see the caption of Figure 3 of the supplementary data for [8]).

In each of the datasets, the binding affinity of 50-mer probes is measured and one thousand 50-mer sequences with strongest nucleosome forming are selected and one thousand 50-mer sequences with weakest affinity are selected to be nucleosome inhibiting. Following [9, 4], each 50-mer is represented by a 2772 vector. For each dataset, we use ten-fold cross validation. That is, first the 1000 positive examples and 1000 negative examples are divided into ten folds randomly, each fold with 100 positive examples and 100 negative examples and we label the folds as $\mathcal{V}_1$, $\mathcal{V}_2$, ..., $\mathcal{V}_{10}$. Then each fold is used as a validation set to evaluate a classifier learned using the other nine folds and this procedure is repeated until all the folds are used once as a validation set. The cross validation is the most common way to evaluate the generalization performance (i.e., the performance on unknown test samples) of a classifier. Note this is essential as there are procedures that may give high accuracy on the training set itself but perform poorly on new test samples; this is known as overfitting [2]. In the following experiments, we use the ROC score for each ROC curve; ROC score is defined simply as the area under the ROC curve. For ten fold cross validation experiments, we have ten ROC scores and use the median ROC score to measure the overall effectiveness of the features and the classifiers, as in [4].

4.2 Results on Using All Features

As discussed earlier, we use the AdaBoost procedure to first select k-mer features sequentially and then construct a strong classifier. We apply the AdaBoost procedure on each dataset using 10 fold cross validation. For each fold, we continue to choose features as long as the ROC score increases and we stop when the score does not increase significantly or the number of the features reaches two hundred. Figure 1 shows the results. Clearly the cross validation performance varies significantly from dataset to dataset, reflecting the differences in data collection conditions and also the nature of the datasets. Note that more features do not always correspond to better validation performance, such as the Ozsolak MCF7 dataset; this is because the error that is minimized is the error on the training set while the error shown here is the error on the validation set. Clearly the observation made in [9] that none of the single k-mer gives a better performance than using all the 2772 features does not justify the use of all the features.

Table 1 compares the results given by the support vector machines as in [4] to our results. Our method gives better performance in all the datasets and in some of them significantly. For example, on the Ozsolak MCF7 dataset, the ROC score of the prediction accuracy is improved from 0.706 to 0.767, a significant improvement in accuracy. The ROC scores on Ozsolak PM, Ozsolak Raw, and Ozsolak T47D also improve significantly. In addition, the number of k-mer features needed to achieve the performance is much smaller, compared to the 2772 features. In particular, it only requires 39 features on the Ozsolak MCF7 dataset to obtain the maximum improvement; fewer than one hundred features are needed on Ozsolak A375, Ozsolak IMR90, Ozsolak MALME, Ozsolak PM, and Ozsolak Raw datasets to achieve the respective optimal performance. While there are common features among the datasets, with A/T and C/G being the most prominent in most of the cases, different optimal features are chosen for different datasets.

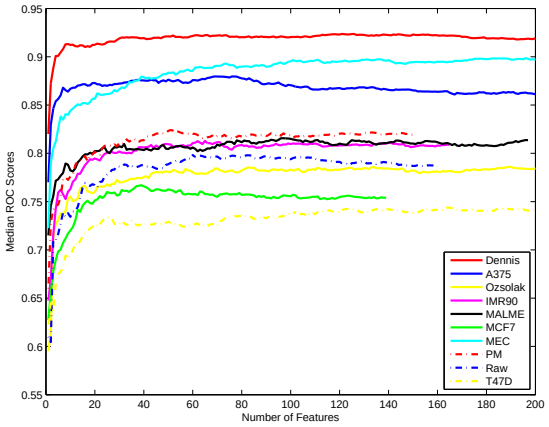

Figure 1: Median ROC score vs. different number of features on different datasets.

Table 1: Comparison of prediction accuracy (median ROC score) of AdaBoost and SVM [4]

Dataset	SVM	AdaBoost	
	ROC score	# of Features	ROC score
Dennis	0.921	121	0.924
A375	0.878	69	0.880
Ozsolak	0.737	190	0.787
IMR90	0.799	65	0.813
MALME	0.811	97	0.813
MCF7	0.706	39	0.767
MEC	0.880	196	0.899
PM	0.783	51	0.824
Raw	0.739	60	0.798
T47D	0.706	164	0.744

4.3 Common Features for Different Cells

As the most effective features identified on different datasets are different, an interesting question is whether there is a small number of features that will be effective on all the available datasets. As discussed earlier, for intrinsic nucleosome positioning, it would be biologically more relevant if only a small number of features are needed for all the cell types of the same organism (e.g., humans in this case).

To address this question, we identify common features from the ten datasets. As described in Section 3.1, we first apply the AdaBoost feature selection on each training-validation division to select a number of features (200 in this case). For each feature that is chosen, it is assigned a rank to be the order that it is first chosen, where the first chosen feature has rank 1, the second has rank 2, and so on; for the features that are not chosen, we assign a rank of 201. After we run the ten fold cross validation tests on all the ten datasets, for each feature, we compute the average rank using the 10×10 rankings from different cross validations of all the datasets. Then all the features are ranked accord-

ing to their average ranking from low to high and we select features from low to high accordingly.

With the identified common features, we apply AdaBoost algorithm with only a specified number of common features to test whether the small number of common features are sufficient to predict accurately on all the datasets. We have done experiments using 5, 10, 20, 50, and 100 common features, and Fig. 2 shows the median ROC scores for 20 features.

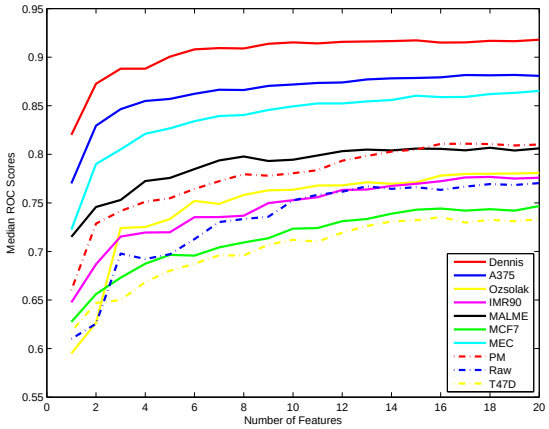

Figure 2: Median ROC scores when twenty sequence features are used for all cell types.

Figure 3 compares the median ROC scores using all 2772 features, 5, 10, 20, 50, and 100 features on the ten datasets using 10 fold cross validation. It shows clearly that on Dennis and Ozsolak A375 accurate prediction can be achieved using as few as five features and all the choices give good performance. On the other datasets, it appears that five features are not sufficient and they require ten features (or even more as on Ozsolak IMR90) to achieve performance close to the one using all features. This appears to be directly related some of the systematic biases in these datasets, as noticed in Figure 3 of supplementary data for [8]. In some cases, using a small number of features gives even slightly better performance; this is due to the greedy nature of the AdaBoost feature selection and the randomness in dividing the ten folds.

5. CONCLUSION

In this paper we propose an AdaBoost based method for both feature selection and classifier learning for intrinsic nucleosome positioning. A distinctive advantage of the proposed method is that it identifies explicitly a small number of features from a large set of candidates efficiently, in addition to that the resulting classifier generally gives more accurate prediction. Using the ten datasets, we are able to also identify a small number of features that give accurate prediction on all of them. While the results are convincing, additional evaluations need to be done, especially genome wide prediction with known biological features.

Acknowledgments

This work was partially funded by NSF grant DMS-0713012 and the National Institutes of Health through the NIH Road-map for Medical Research, Grant U54 RR021813. Information on the

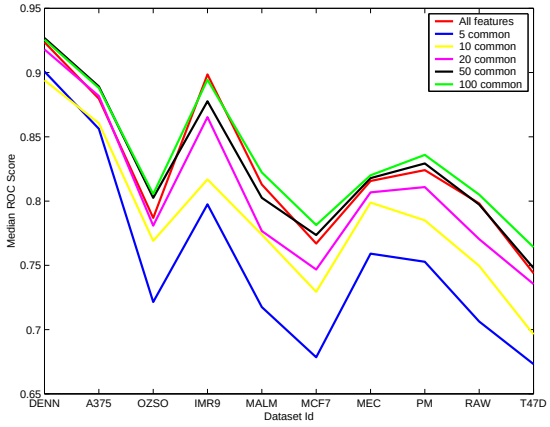

Figure 3: Comparison of median ROC scores of all 2772 features, 5, 10, 20, 50, and 100 common features on the ten datasets.

National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih .gov/bioinformatics.

6. REFERENCES

[1] J. H. Dennis, H.-Y. Fan, S. M. Reynolds, G. Yuan, J. C. Meldrim, D. J. Richter, D. G. Peterson, O. J. Rando, W. S. Noble, and R. E. Kingston. Independent and complementary methods for large-scale structural analysis of mammalian chromatin. *Genome Research*, 17:928–939, 2007.

[2] R. O. Duda, P. H. Hart, and D. G. Stork. *Pattern Classification*. John Willey and Sons, 2001.

[3] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

[4] S. Gupta, J. Dennis, R. E. Thurman, R. Kingston, J. A. Stamatoyannopoulos, and W. S. Noble. Predicting human nucleosome occupancy from primary sequence. *PLoS Computational Biology*, 4(8):1–11, August 2008.

[5] I. P. Ioshikhes, I. Albert, S. J. Zanton, and B. F. Pugh. Nucleosome positions predicted through comparative genomics. *Nature Genetics*, 38:1210–1215, 2006.

[6] R. Kornberg. The location of nucleosomes in chromatin: specific or statistical. *Nature*, 292:579–580, 1981.

[7] R. D. Kornberg and Y. Lorch. Twenty-five years of the nucleosome, review fundamental particle of the eukaryote chromosome. *Cell*, 98:285–294, 1999.

[8] F. Ozsolak, J. S. Song, X. S. Liu, and D. E. Fisher. High-throughput mapping of the chromatin structure of human promoters. *Nature Biotechnology*, 25:244–248, 2007.

[9] H. E. Peckham, R. E. Thurman, Y. Fu, J. A. Stamatoyannopoulos, W. S. Noble, K. Struhl, and Z. Weng. Nucleosome positioning signals in genomic dna. *Genome Research*, 17(8), 2007.

[10] E. Segal, Y. Fondufe-Mittendorf, L. Chen, A. Thåström, Y. Field, I. K. Moore, J.-P. Z. Wang, and J. Widom. A genomic code for nucleosome positioning. *Nature*, 442:772–778, 2006.

[11] E. Segal and J. Widom. What controls nucleosome positions? *Trends in Genetics*, 25(8):335–343, 2009.

[12] P. Viola and M. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[13] G.-C. Yuan and J. S. Liu. Genomic sequence is highly predictive of local nucleosome depletion. *PLoS Computational Biology*, 4(1):164–174, 2008.