Prediction of Transcription Factor Binding Based on DNA Physical Properties

Nathan Clairmonte (260673075), Ella Kantor (260849009), Sonia Sharapova (260907516)
December 9, 2021

1. Introduction

The four nucleic acids that build the DNA double helix all have the same general structure with the exception of the nitrogenous base. These structural differences, however, have a significant effect on the physical interactions that the bases have with other molecules. Each unique sequence of nucleotides has physical properties of the shape of the DNA double helix which differ from other sequences (Oguey et al., 2010). These physical properties include things like major and minor groove width and degree of bending or twisting of the helix (King et al., 2006). In addition, bound proteins (such as a TATA box or TATA-binding proteins) and solvent conditions can cause deformation and change the bond angles in the helix (Rohs et al., 2009). Many types of proteins bind to the DNA double helix in order to carry out functions such as transcription and replication. These proteins must bind to a specific place in the DNA strand and their affinity to this location is determined by their own structure as well as the structure of the DNA.

There are two common forms of protein-DNA recognition, base-readout and shape-readout, also known as direct-readout and indirect-readout. Base-readout is the direct bonding of amino-acid side chains of the protein to the base, usually with hydrogen bonds, hydrophobic interactions or water-mediated interactions (Yesudhas et al., 2017). In this form, the exact sequence of nucleotides is the major determinant of binding. Shape-readout is when protein binding is specified by physical properties of the DNA shape. It is thought that the interplay of both base-readout and shape-readout is the cause for the exact specificity of protein binding (Slattery et al., 2014). For example, if two short sequences of DNA have the same base order, but one has a transcription factor bound but the other does not, it is likely that this is caused by shape-readout mechanisms recognizing a slight difference in the shape of the two sites.

Transcription factor proteins are one of these molecules that interact with DNA by binding to the sequence at a certain location and initiate or regulate the transcription of DNA to RNA. Analyzing these protein-DNA recognition mechanisms in the context of transcription factor binding site predictions, we can see that DNA shape data is important in being able to predict the binding sites. Being able to better predict transcription factor binding is relevant in expanding our relatively limited knowledge of gene regulation and expression. It has applications in medicine, environmental metagenomics, epigenetics, and many other current fields. Sequencing technology has been able to give us full genomes, but the nuances of specific

genes and how their transcription is controlled is the next step in identifying the causes and locations of diseases and irregularities in biological processes.

The most basic models for predicting transcription factor binding sites use the position weight matrix and make the assumption that the protein-DNA interaction is based on the independent sequence of nucleotides. More complex transcription factor flexible models use Hidden Markov Models and account for flexibility in the length of motifs and interdependence of nucleotide positions (Mathelier et al., 2013). In newer models, deep learning is utilized and neural networks are trained using an extensive database of existing ChIP-seq binding site data. They learn to recognize motifs based on the sequences where transcription factors bind, and since different families of transcription factors can have different binding preferences, they can be specific to certain types of transcription factors (Park et al., 2020).

In this project, we examined whether DNA structural properties are sufficient to predict transcription factor binding sites. To do this, we used existing data to create a set of sequences known to be bound by transcription factor UAK42 and look like they should be based on UAK42's position weight matrix (PWM), and a set of sequences that are not bound by UAK42 but still look like they should be based on UAK42's PWM. We then attempted to classify between the bound samples from the unbound samples with a machine learning model using only their physical properties. Multiple machine learning models were trained using these datasets to determine which was the best predictor for our problem. We hypothesize that it is possible for a machine learning classifier to make predictions with significant accuracy.

2. Methodology

Our methodology for this analysis is divided into three main stages as described in this section. Three datasets were used to create our samples:

1) A set of active regulatory regions from GM12878 cells derived from lymphoblasts
2) A set of genomic coordinates of transcription factor binding sites of multiple transcription factors
3) Position weight matrices for each transcription factor.

2.1 Identification of Bound and Unbound Sequences

This stage involved the identification of positive and negative samples for the transcription factor UAK42. A positive sample is one where the active regulatory region contains a motif of a transcription factor binding site and the transcription factor binds there. A negative sample is one where the active regulatory region contains a motif of a transcription factor binding site and the transcription factor does not bind. For this stage, positive and negative samples were found for all transcription factors in dataset (3), and UAK42 was chosen in the end because it produced the highest number of samples.

Using the Bio.motifs package from the Python library BioPython and the position weight matrices for each transcription factor from dataset (3), the positive strand of the active regulatory regions in dataset (1) were searched for motifs of each transcription factor using a threshold of 3.0. These found motifs were then compared with the list of actual transcription factor binding sites in dataset (2) to form our collections of positive and negative samples for each transcription factor. The found samples were written to text files in FASTA format to facilitate easier retrieval of their physical properties, as discussed subsequently. Finally, as mentioned before, the transcription factor with the most samples found (UAK42) was chosen for our subsequent machine learning analysis.

2.2 Retrieval of Physical Properties

The physical properties for each of the sequences identified in the first stage were obtained using GBshape, a genome browser database for DNA shape annotations. For each organism, the database provides annotations for physical properties such as minor groove width, propeller twist, roll and helix twist. The sample FASTA files were uploaded to GBshape and the corresponding physical properties for our positive and negative samples were retrieved. These properties were used to build features for the classification process in the third stage.

2.3 Classification of Bound and Unbound Sites

This stage involved using a machine learning classifier to classify bound sites from unbound sites based on their physical properties. This was a binary classification task for which the targets were either bound or unbound. As mentioned, the features used were built using the physical properties obtained from GBshape for each of the samples identified.

3. Machine Learning Methodology

We compared the prediction accuracy of various machine learning models in order to distinguish between bound and unbound sequences for given transcription factor binding sites based on DNA structure.

3.1 Data Preprocessing and Helper Functions

Since the raw data was in the form of a text file, preprocessing and data augmentation was necessary in order to present the data in an available format for the models. This would allow the input files to be recognized as features and labels by the algorithm. In order to achieve

this, we implemented various methods and helper functions. Through these functions we were able to load the data as a list of strings corresponding to the lines in the file, remove line breaks, and load the physical properties downloaded from DNAshape into a Pandas dataframe.

When forming the final feature matrix, it was observed that the array of values obtained for each physical property for each sample contained values that were relatively similar throughout. Therefore, each array was averaged into one value to reduce the dimensionality of the data. Finally, these individual feature arrays were concatenated to form the final feature matrix, which consisted of 4 features (1 averaged value for each of the 4 physical properties; minor groove width, roll, propeller twist, and helix twist) for each sample. For the labels, a 1 was attributed to positive samples while a 0 was attributed to negative samples.

3.2 Data Splitting

The data was split into Features and Targets containing the concatenation of the positive and negative features of the properties. This data was used as training and testing data in the models, with 80% of the data being used as training data (features), and the remaining 20% testing and thereby validating the accuracy of the models' predictions (targets/labels). Leaving a certain amount of data out of the training process in order to validate the models' performance is necessary to avoid overfitting, a modeling error resulting in a model producing an analysis too close to a particular set of data.

3.3 Machine Learning Models

Using the Scikit-learn package (Sklearn), we were able to effectively test a variety of predictive models. Sklearn is a python library which provides a selection of tools for machine learning which we used to run predictions on our data. A number of different models were implemented to test for the best prediction accuracy, shown in Table 1 alongside their prediction accuracy. Each model was trained on the features and labels of the training data and validated through the testing data. For every model, we plotted a confusion matrix, which allowed for a visualization of the model's categorizations, and predicted the Accuracy, F1 Score, Precision, and Recall values. Accuracy is the measurement for identifying relationships and patterns between the expected and predicted labels, Precision is the ratio between the True Positives and all the positives in the samples, Recall is the measure of the model correctly identifying true positives, and the F1 score is the weighted average of Precision and Recall, taking both false positives and false negatives into account.

3.4 Hyperparameters

In order to improve the accuracy of the model, we tested different hyperparameters in order to see their effects on the models training. Hyperparameters control the learning process, and thus the behaviour of a machine learning model. We decided to investigate the effects of hyperparameters on the top three models.

4. Results

In all cases, the recall was higher than the precision, implying that positive samples are in the minority class and the negative examples could become false positives. The table below shows the training models with their respective baseline accuracies, before any hyperparameter tuning has been applied.

Model	Accuracy (%)
Logistic regression	69.966
Decision tree	66.769
Random forest	71.589
Gradient boosted tree	70.442
Bernoulli naive bayes	50.470
Gaussian Naive bayes	67.482
Multi-layer perceptron	70.003
Linear Discriminant Analysis	69.871

Table 1. Machine learning models used and their respective predicting accuracy.

Out of the baseline models tested, bernoulli naive bayes showed the worst prediction accuracy of 50%, and the random forest, gradient boosted trees, and multi-layer perceptron models performed best, with their accuracies being above 70%. Figure 1 shows the comparisons between their respective confusion matrices. Random Forests performed with the best baseline prediction, so this was chosen as the predictive model.

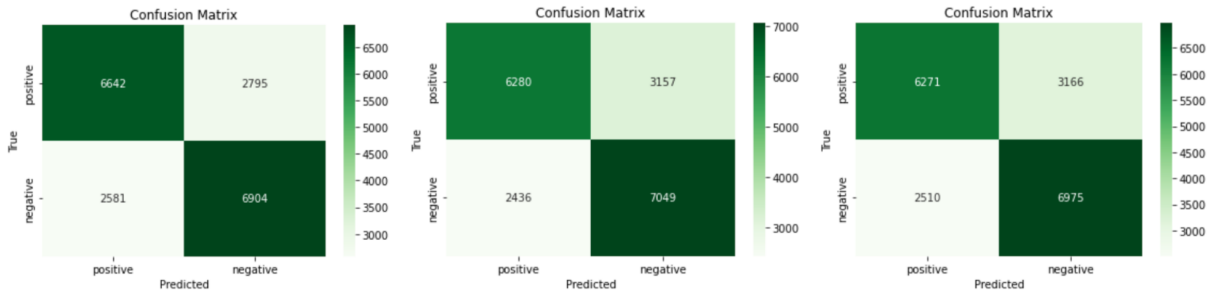

Figure 1. Confusion matrices for random forest, gradient boosted tree, multi-layer perceptron.

The random forest model is a classification algorithm which uses a combination of many decision trees and using bagging and feature randomness. In general, this modeling strategy adds randomness to the model which results in a wider diversity. They do this by searching for the best feature among a random subset of features. This creates an uncorrelated forest of trees whose prediction is more accurate than that of any individual tree.

Random Forests showing the best accuracy could be attributed to the physical properties not being strongly correlated with each other, so drawing from an ensemble of relatively uncorrelated decision trees that each split on the features in different ways could lead to a more robust prediction.

4.1 Hyperparameter performance

We tested the effect of hyperparameter modification of four different models: logistic regression, random forest, gradient boosted tree, and bernoulli naive bayes. The first three models were selected due to their high baseline prediction accuracies with the goal of obtaining even better predictions. The worst performing model, bernoulli naive bayes, was chosen in order to see if hyperparameters could significantly improve its performance.

4.1.1 Logistic Regression

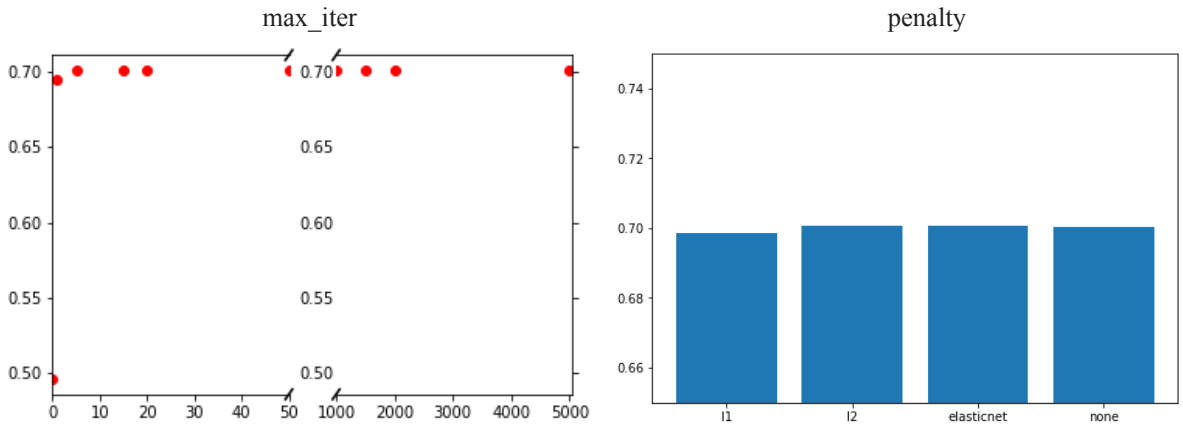

Figure 2. Hyperparameter modifications for the logistic regression model. The parameters max_iter and penalty were tested at various input values.Optimal performance was observed at max_iters >= 15 and penalty=l2, with accuracies of 70.06% and 70.08% respectively.

Max_iter: Maximum number of iterations taken for the solvers to converge.
Penalty: none, l2, l1, elasticnet

The maximum iterations rose with increased max and plateaued at an accuracy of 70.06%, and the best penalty was l2 at 70.08%. This is not a great increase from the baseline prediction of 69.97%, so the hyperparameters did not have a big effect on the model's prediction accuracy.

4.1.2 Random Forest

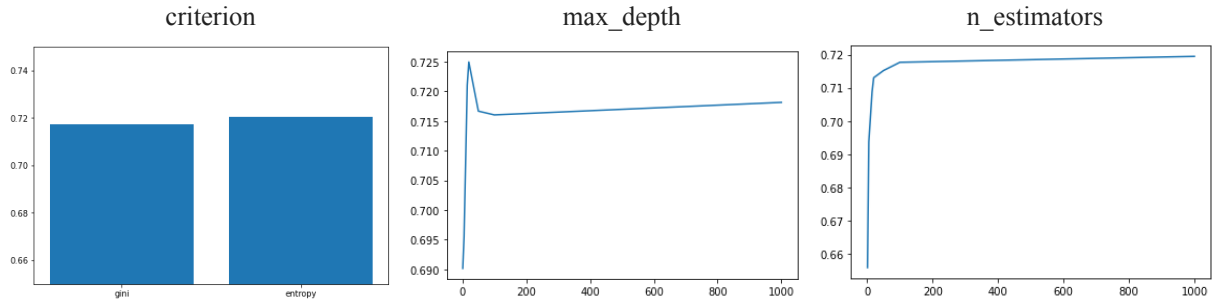

Figure 3. Random Forest hyperparameter modifications. The parameters criterion, max_depth, and n_estimators were tested at various input values. Optimal performance was observed with criterion='entropy' at 72.05%, max_depth=20 at 72.5%, and n_estimators=1000 at 71.95%.

Criterion: The function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “entropy” for the information gain.
Max_depth: The maximum depth of the tree.
N_estimators: The number of trees in the forest.

As shown in Figure 3, the hyperparameters with the best predictions were entropy criterion, a max depth of 20, and high n_estimators value (>1000, as the model shows an increase with higher number of trees). Max_depth showed the greatest increase from the baseline with an increase of ~1%.

4.1.3 Gradient Boosted Tree

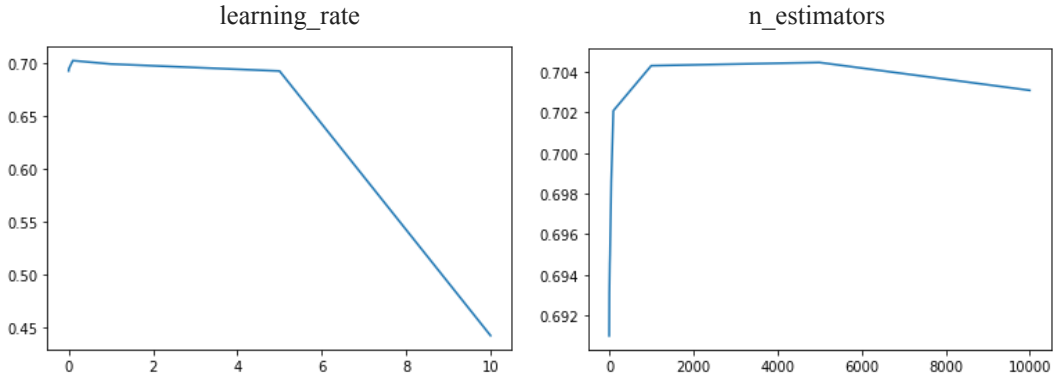

Figure 4. Gradient Boosted Tree hyperparameter modifications. The learning_rate and n_estimators were tested at various input values. Optimal performance was observed with learning_rate=0.1 at 70.21% and n_estimators=5000 at 70.45%.

Learning_rate: Learning rate shrinks the contribution of each tree by learning_rate. There is a trade-off between learning_rate and n_estimators.
N_estimators: The number of boosting stages to perform. Gradient boosting is fairly robust to overfitting so a large number usually results in better performance.

A learning rate of 0.1 and n_estimators of 5000 showed optimal accuracy results. These hyperparameters did not perform better than the default with validation accuracy 70.44%

4.1.4 Bernoulli Naive Bayes

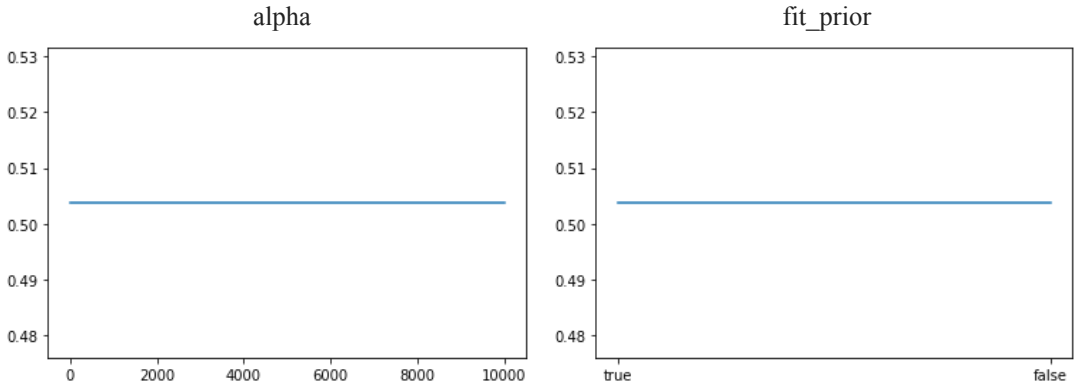

Figure 5. Bernoulli Naive Bayes hyperparameter modifications. The hyperparameters alpha and fit_prior were tested, with no optimal performance as the hyperparameters did not have an effect on the validation accuracy of the model. It remained at the score of 50.57%

Alpha: Additive (Laplace/Lidstone) smoothing parameter (0 for no smoothing).
Fit_prior: Whether to learn class prior probabilities or not. If false, a uniform prior will be used.

The hyperparameters did not affect the model's performance, and it remained at the worst score of 50.57%.

Overall, hyperparameters did not have a great impact on the performance of our models, with the accuracy never exceeding 72.5%. Since most models plateaued at around 70%, this could be attributed to the data used rather than the models themselves. There is also a risk of the models having "memorized" the data through consistent exposure as they were trained multiple times, however this would result in higher than expected accuracies which we did not observe.

5. Discussion and Future Work

In this report, we have demonstrated that a machine learning model trained with DNA shape data can have a significant amount of accuracy when predicting transcription factor binding sites. Though it is not comparable with the accuracy of deep learning models like neural networks, it shows that DNA physical shape attributes are an important data point that should be considered in all models of predicting binding sites.

Regarding the performances of the various models we analysed, we found that the random forest classifier outperformed other models. While it was not immediately clear why this

was the case, we theorized that it may have something to do with the fact that individual physical properties such as minor groove width and propeller twist do not necessarily have a strong correlation with each other. Therefore, drawing from an ensemble of relatively uncorrelated decision trees that each split on the features in different ways, as is done with a random forest classifier, may lead to more robust and accurate predictions. Unfortunately, we were unable to find any existing literature which either supported or refuted this theory.

The predictive models we were able to build did not show very high accuracies between the expected and predicted labels with the highest accuracy being predicted at around 72.5%. The hyperparameters we tested did not have a great impact on the performance of the models which could either be attributed to the baseline models being representative of the data provided, or the wrong hyperparameters being modified. The greatest increase from the baseline of 1%, as seen in the max_depth plot of Figure 3. Given that all models converged at about 70%, we believe that this is the highest validation accuracy that can be achieved given the information from the datasets that we used.

More robust models could be built to provide higher distinction between bound and unbound sequences for a given transcription factor. This could be done through testing new hyperparameters in the models which performed with the best baseline accuracy and observing which selections are optimal.

Our model relies on the shape-readout mechanism of protein-DNA recognition. Though this mechanism is an integral part of protein-binding site specificity, it is thought that the interplay of both shape-readout and base-readout mechanisms is a closer approximation of what happens in the cell (Slattery et al., 2014). Therefore, our model could likely be improved upon by including both of these mechanisms in the training of the machine learning model.

There are many different types of transcription factors in the human body alone. They can be broken down into families with different functions in cellular activities. These different families have different dominant recognition mechanisms to each other which adds another layer to prediction models. One prediction model can have varying accuracy among each transcription factor it is tested on. This could mean that our model would be more accurate on certain transcription factors that depend more on shape-readout mechanisms but less accurate on transcription factors that depend more on base-readout mechanisms.

In the future, our model can be used to identify transcription factors with a high sensitivity to DNA physical shape properties. Further uses could even be to analyze the DNA shape data and see which physical motifs are most associated with positive sites for specific transcription factors.

6. Bibliography

King, R. C., Stansfield, W. D., & Mulligan, P. K. (2006). *A dictionary of genetics*. Oxford University Press.

Mathelier, A., & Wasserman, W. W. (2013). The Next Generation of Transcription Factor Binding Site Prediction. *PLOS Computational Biology*, *9*(9), e1003214. https://doi.org/10.1371/journal.pcbi.1003214

Oguey, C., Foloppe, N., & Hartmann, B. (2010). Understanding the Sequence-Dependence of DNA Groove Dimensions: Implications for DNA Interactions. *PLOS ONE*, *5*(12), e15931. https://doi.org/10.1371/journal.pone.0015931

Park, S., Koh, Y., Jeon, H., Kim, H., Yeo, Y., & Kang, J. (2020). Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Scientific Reports*, *10*(1), 13413. https://doi.org/10.1038/s41598-020-70218-4

Rohs, R., West, S. M., Sosinsky, A., Liu, P., Mann, R. S., & Honig, B. (2009). The role of DNA shape in protein–DNA recognition. *Nature*, *461*(7268), 1248–1253. https://doi.org/10.1038/nature08473

Slattery, M., Zhou, T., Yang, L., Dantas Machado, A. C., Gordân, R., & Rohs, R. (2014). Absence of a simple code: How transcription factors read the genome. *Trends in Biochemical Sciences*, *39*(9), 381–399. https://doi.org/10.1016/j.tibs.2014.07.002

Stella, S., Cascio, D., & Johnson, R. C. (2010). The shape of the DNA minor groove directs binding by the DNA-bending protein Fis. *Genes & Development*, *24*(8), 814–826. https://doi.org/10.1101/gad.1900610

The BioPython Contributors. (n.d.). *Bio.motifs package—Biopython 1.75 documentation*. Retrieved December 5, 2021, from https://biopython.org/docs/1.75/api/Bio.motifs.html?fbclid=IwAR0YOfwiQDqOMXgbF21ATldqMGMiZIhLFs37HKxl2ycWtVqm-ROpcnpB2Sw

T.P. Chiu, L. Yang, T. Zhou, B.J. Main, S.C. Parker, S.V. Nuzhdin, T.D. Tullius, R. Rohs: GBshape: a genome browser database for DNA shape annotations. *Nucleic Acids Res.* 43, D103-D109 (2015)

Yesudhas, D., Batool, M., Anwar, M. A., Panneerselvam, S., & Choi, S. (2017). Proteins Recognizing DNA: Structural Uniqueness and Versatility of DNA-Binding Domains in Stem Cell Transcription Factors. *Genes*, *8*(8), 192. https://doi.org/10.3390/genes8080192