‘These footprints are leading us nowhere!’

investigation of the usage of footprinting analysis for ATAC-seq data to find regulatory elements

Thijs Benschop 23-12-2022

Abstract

ATAC-seq is a technique that sequences unbound parts of the chromatin. Parts of the DNA that are not bound by histones, and are therefore easily accessible, are sequenced most often in ATAC-seq. Analysis of the sequencing of open chromatin is used to examine which parts of the DNA are active for specific cells. Figuring out which elements of the DNA make cells behave the way they do is relevant for understanding genetic diseases better. Extracting concise conclusions from ATAC-seq data about gene regulatory elements is still difficult. ATAC-seq data is noisy and therefore hard to interpret. One method to analyze ATAC-seq data is using footprint analysis. When certain parts of the sequenced chromatin have a lower read count than expected, while surrounded by a normal or higher read count, this area can be examined. This depletion of read counts and surrounding normal read counts is called a footprint. Most likely a protein was bound to the DNA here. These proteins can be regulatory elements that influence the behaviour of a cell. Knowing which regulatory elements or transcription factors are bound to the DNA in different cells can give insight into why certain cells are diseased.

This paper examines the benefits of footprinting analysis on ATAC-seq data. Footprinting analysis gives a score for each base pair in the sequence. This score can be interpreted as the chance of a protein being bound at that location while sequencing. The informative capacity when making predictions about transcription factor binding sites of the footprint scores is examined and compared to the information captured by the read counts from ATAC-seq. The predictions made by footprint scores are also compared to predictions of bound transcription factors by a neural network that predicts open chromatin and lastly to the current state of the art of predicting transcription factor binding sites. A logistic regression analysis is used to compare these methods. To extract extra information hidden in the spatial distribution of the different values, convolutional neural networks were trained to predict bound transcription factors. All methods are trained to predict where active regulatory elements are in a HEPG2 cell line. A comparison of the predictions of the different classifiers concludes footprint scores do not contain more information than read counts from ATAC-seq data. More work is needed to make concise predictions for regulatory elements in ATAC-seq data and the need for accurate predictions is large.

Layperson’s summary

ATAC-sequencing is a popular method to measure and sequence the accessible parts of the DNA in a cell. The accessible parts of the DNA are called open chromatin. The closed chromatin is the part of the DNA that is inaccessible. The closed parts of the DNA are bound to nucleosomes, the spool-like proteins that keep DNA organized and compact. These parts are difficult to be accessed by any proteins, and so they can not be reached by the proteins that transcribe DNA and make the cell function as it does. The parts that are not bound to anything, the open parts, play an important role in how the cell behaves. When smaller proteins than nucleosomes are bound to the DNA, this is visible in the ATAC-seq results. These small proteins play a huge part in the behaviour of a cell. A way of analyzing these small bound proteins in the ATAC-seq data is by doing a footprinting analysis. A footprinting analysis could give a lot of information about the behaviour of a cell. When this analysis

is done on a diseased cell, information could be obtained about which parts of the DNA and which bound proteins cause a cell to be diseased. This is valuable information for attempts to cure many genetic diseases. Analyzing ATAC-seq data to learn about cell behaviour is still difficult. This paper examines the use of the footprinting analysis on ATAC-seq data by comparing it to other methods used to investigate cell regulation.

Introduction

Every type of cell behaves differently, whether it is healthy or diseased, young or old. This is a result of differential gene expression per cell. Gene expression is highly regulated by DNA accessibility[48]. Open parts of the DNA are easily accessible to transcription factors and other DNA-binding proteins, leading to the genes on the open parts being expressed more easily. This also means that examining open parts of the DNA can give insight into which genes are being expressed in the cell and which regulatory elements play a role in the expression. Open chromatin profiling relies on this logic to study gene regulation from chromatin accessibility assays[52]. Good analysis of the data obtained by open chromatin sequencing is a topic still that requires work to improve predictions of gene regulation, but at the same time gives hope for a better understanding of what causes cells to behave the way they do. Understanding why cells turn diseased will be easier when the regulation of these cells is understood better. Open chromatin profiling could play an important role in understanding diseased cells.

Why do open chromatin profiling? What can be found?

Open chromatin profiling gives accessibility data of the DNA while sequencing it. This data can be used to examine the active cis-regulatory elements in a cell. A cis-regulatory element is a part of the DNA that acts on the regulation of a gene that is present on the same DNA-strand[34]. Four main types of cis-regulatory elements are currently known (Fig. 1). Promotors, enhancers, repressors and insulators. Promotors are the basic binding site for the transcriptional machinery allowing for the transcription of the gene (fig 1A). Enhancers promote the transcription activity of a promotor by helping form a chromatin loop where transcription factors bound to the enhancer are brought closer to the promotor, increasing promotor activity[33] (fig. 1C). Repressors decrease the transcription activity of a promotor (fig 1B). Insulators serve two distinct purposes. They can block the activation of a promotor by an enhancer that is actually activating a different promotor, thus causing the enhancer to find its promotor more robustly (fig 1D). Insulators can also prevent the condensation of chromatin, causing the promotor to stay accessible for enhancers and transcription factors. These four types of cis-regulatory elements all have a large influence on the regulation of gene transcription and are thus the main target for chromatin accessibility assays.

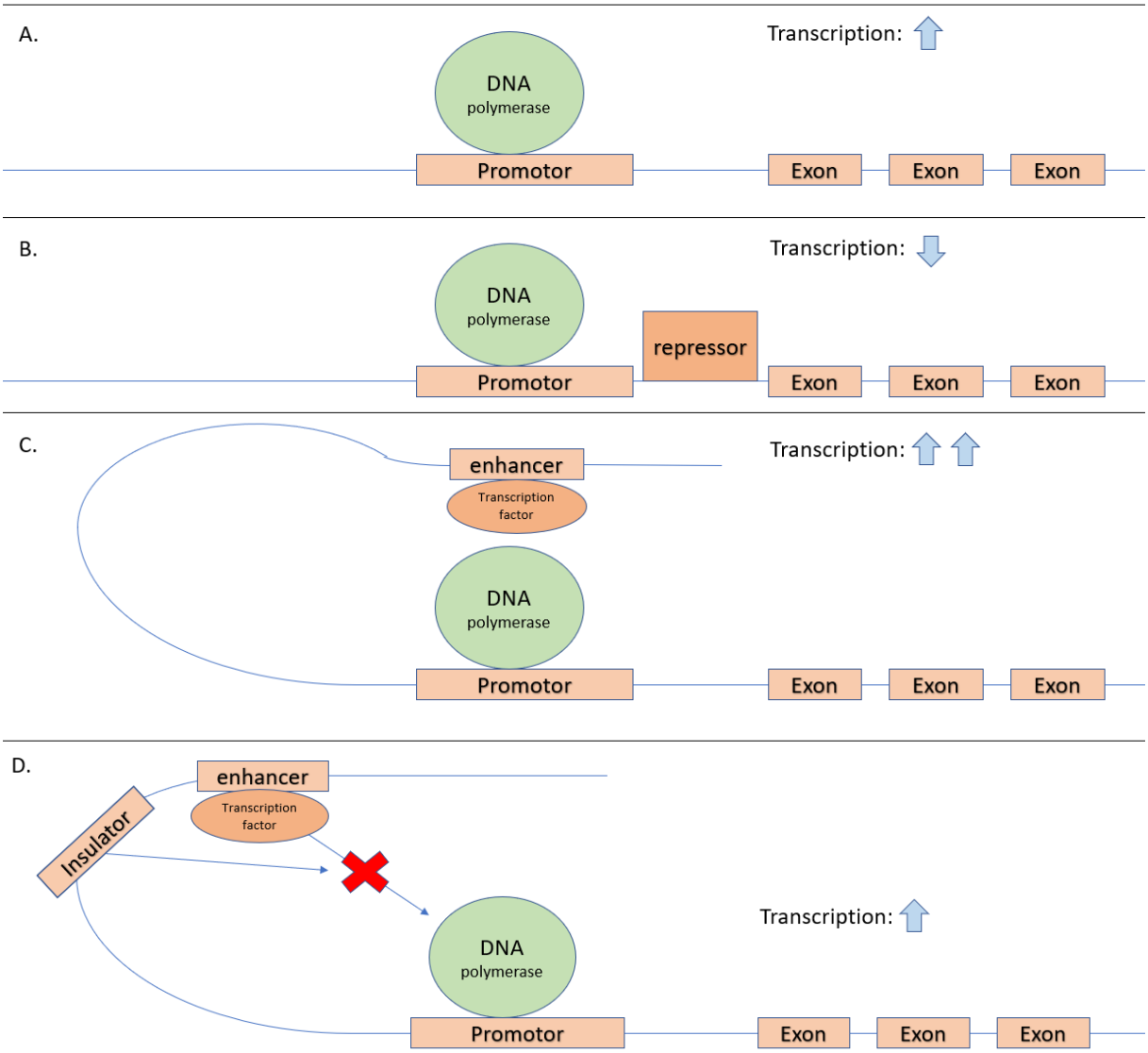

Figure 1. Schematic overview of different cis-regulatory elements. A. Promotor binds DNA polymerase causing transcription of the gene. **B** repressor binds the DNA causing a decrease of transcription of the gene. **C**. enhancer binds transcription factors that help DNA polymerase increase transcription **D**. Insulator causes enhancers to not help in the transcription of a gene.

ATAC-sequencing

One of the most booming methods for open chromatin profiling at the moment is ATAC-seq (Assay for Transposase Accessible Chromatin)(list of acronyms at the end). ATAC-seq exploits the properties of a hyperactive variant of transposase Tn5 to cleave open chromatin[15]. Tn5 can cut open DNA and add adaptors for sequencing at the same time[39]. The Tn5 is treated with the adaptors beforehand to be able to do the cutting and tagging simultaneously. After the Tn5 is added to the DNA, the paired-end sequencing of the tagged parts can be done. The fact that ATAC-seq only consists of these two major steps is one of the reasons why it is such a popular and promising technique and makes it so easy to use as a high-throughput technique or as a single-cell technique.

Tn5 is more likely to cut open the DNA in places where the chromatin is unwound. This leads to the sequencing of open parts of the DNA in that specific cell, giving hints as to what parts of the DNA are open, and consequently most likely active in that cell. To be able to analyze the sequenced DNA, a

couple of steps need to be taken first. As is always important when sequencing, the quality of the reads needs to be examined, and the adaptors need to be removed from the sequence. After this, the alignment of the sequences to a known genome should be made. Then, the quality of alignment should be checked. If all is okay, the real analysis can start, beginning with peak calling.

Peak calling is one of the important steps in the analysis of ATAC-seq data. Peak calling decides which reads are present because of an accessible chromatin region, and which reads are present because of background noise or because of the sequencing or preparation techniques. For example, Tn5 can also bind in the small open spaces in between nucleosomes. This leads to larger reads spanning over the amount of nucleosomes present there. These reads are way longer than the open chromatin reads and need to be filtered out by not calling these peaks. Smaller reads spanning over a transcription factor binding site need to be not filtered out by the peak calling.

Three main ways of peak calling are currently used. Count-based calling, shape-based calling and calling using hidden Markov models (HMM's). Only shape-based callers are not currently used in ATAC-seq analysis. Count-based callers use statistics to determine whether a peak is significant or not. Comparing the reads found to the background reads is the main way of analyzing the peaks. Shape-based callers analyze the shape of the peaks found next to just using the number of reads. This method is mostly used in other open-chromatin profiling methods such as Chip-seq[52]. This method has been proven to improve results from Chip-seq analyses[40]. Lastly, hidden Markov models can be used to identify peaks in the reads[37]. Hidden Markov models have been shown to improve the peak calling in ChiP-seq longer ago, and a HMM peak caller for ATAC-seq also already exists[46].

An important consideration when peak calling is the cleavage bias that is inherent to Tn5. Tn5 is more likely to cleave certain pieces of sequence depending on the CG-ratio of bases in that part[27]. Pipelines that investigate this bias do exist and should be used during analysis[36]. These analyses rely mainly on the sequencing depth, so it is difficult to correct when fewer samples are present. ATAC-seq data is quite noisy on its own. This makes peak calling an important step in order to analyze the data correctly. The next step in the analysis is also negatively influenced by the inherent noisiness of ATAC-seq data.

Peak annotation reveals the function of the sequenced peaks in ATAC-seq data

After preparation and initial analysis of the sequences, the results are ready to be interpreted further. An example of this is functional peak annotation. Peak annotation can be done In different ways. The most simple way is by looking at which gene is closest to the peak. However, this still does not indicate the exact function of the regulatory element found, only the gene that it might affect. More information is needed to obtain interpretable results. The main two methods for interpreting peaks are looking at motifs found on transcription factor binding sites on the open chromatin reads and looking at footprinting by transcription factors.

Databases with regulatory-element-binding motifs can be used to annotate peaks better. When a known motif is found in a peak, a lot of information can be obtained about the mechanics behind the regulatory elements in this part of the DNA. Perfect annotation would give a large insight into how the different transcription factors and other regulatory elements work together. If certain TF's are present more often than expected, the function of these TF's could be examined in this cell specifically. A problem with motif-based peak annotation is that not all transcription factor motifs are known yet. However, the main problem is that a small piece of sequence can match many different motifs, leading to no clear decision on which motif is most likely to be active in the peak.

Footprinting of ATAC-seq data uncovers the location of bound transcription factors

The other main analysis of ATAC-seq data is examining footprinting of transcription factors in the reads. When a transcription factor binds to the DNA, the Tn5 transposase can not cleave the DNA in that spot. The Tn5 can cleave the parts before and after the TF, and thus can only attach sequencing adaptors around the TF, meaning that there will be only reads starting before and after the binding site, but no reads starting in the binding site. The TF binding site is sequenced, but it will have fewer total reads than the parts that are accessible to Tn5. This is similar to how nucleosomes are recognized by ATAC-seq, a lack of read depth gives a hint that a nucleosome may be present. The difference is that TF-binding sites are way shorter than the parts of DNA bound to a nucleosome. A TF-binding site with a bound TF will look like a peak with a lower part in the middle in the reads.

Bias in ATAC-seq data needs to be corrected for proper analysis

Transposase Tn5 has a bias when cleaving DNA. An article by Lu *et al.* (2017) shows that ATAC-seq on DNA without any proteins bound still leads to small footprints being found during the analysis[22]. These footprints are a result of the Tn5 cleavage preferences and not of a transcription factor being present. It is important to realise that not all footprints found in ATAC-seq are present because of regulatory proteins. Protocols correcting for Tn5 binding preference do already exist, correcting sequences that Tn5 binds to most easily[25].

The GC-content of a sequence can also influence the cleavage by Tn5. This is a factor that can also be corrected for when analysing and footprinting ATAC-seq data[27]. However, peak callers and footprint software that ignore the GC-content have also been shown to work well.

Machine learning shows great use for the analysis of the large amount of data that ATAC-seq produces

Next-generation sequencing techniques, like ATAC-seq, produce large datasets in no time. Datasets from these sequencing techniques are too large and complex, studying these large datasets without any prior analysis is impossible and finding conclusions would be difficult. Luckily, machine learning proves to be a great solution to this problem. Especially deep learning with the use of neural networks proves to be a promising topic in genomics. Deep learning is a form of machine learning that can handle the great complexity of the input thanks to the flexibility of the model itself. For example, a paper from 2018 shows a neural network that can predict the lab in which DNA was modified by looking at only the DNA sequence [30]. The reason that neural networks can be so complex is that the many hidden layers of a neural network can each have many nodes, and all of these nodes can be given different parameters to scale the network more precisely to a certain feature. After creating a suitable model for a certain feature, the model can be reused to research many datasets in very little time. To make machine learning an even more time-efficient way of analysis, end-to-end models can be created. These models eliminate much of the need for manual processing of data and return meaningful results, causing fewer processing steps to be necessary and saving large amounts of time.

Neural networks use multiple hidden layers that each calculate values for the next layer to go from an input to a prediction [14]. The networks used in functional genomics mostly have 3 different types of hidden layers: Fully connected, convolutional and recurrent layers. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are most often used for DNA applications. In an RNN, nodes are calculated one after another, using the output of the previous node in their memory to calculate the next node. This can be used to predict outcomes on a timescale. CNN's are used to scan inputs, they can look for certain patterns in an input, irrelevant of where in the input the motif is. The

convolutional layer can scan for many filters at the same time. The 3 types of layers are illustrated in figure 2.

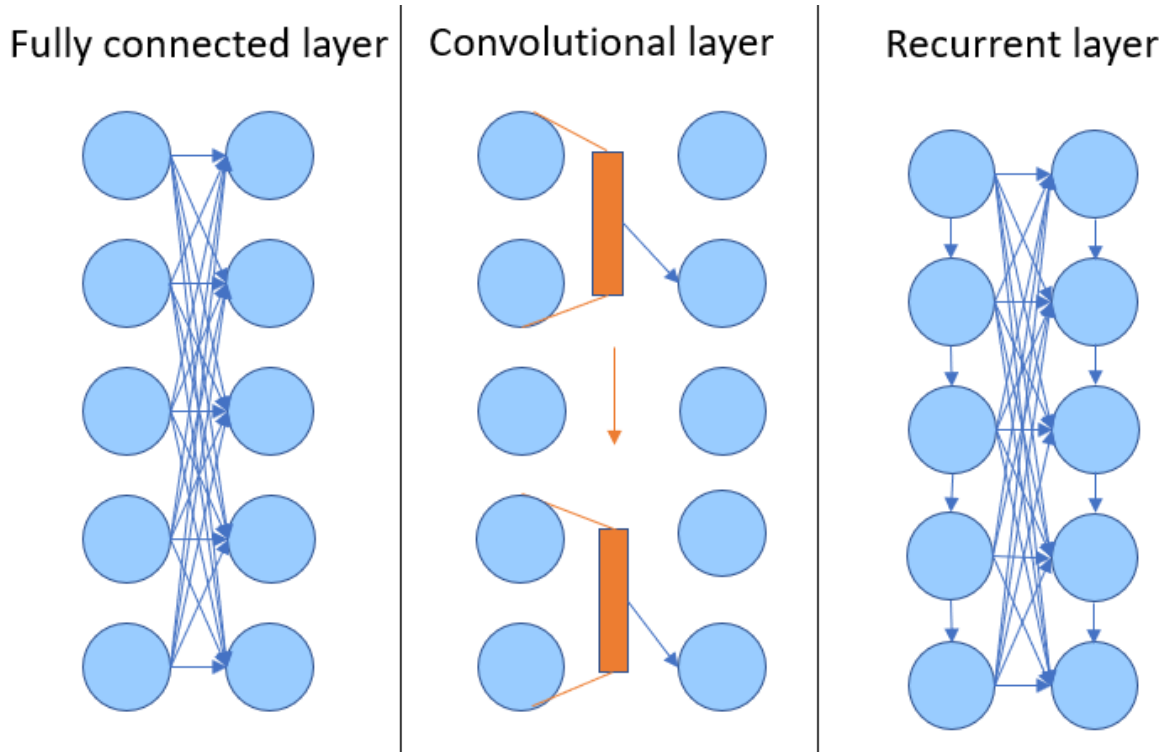

Figure 2. Schematic view of three layers used in neural networks. A fully connected layer has all nodes from one layer to the next layer connected. A convolutional layer uses a filter that scans over the input shown in orange colours. A recurrent layer has all nodes from one layer to the next connected, but also has the nodes within a layer connected to each other or to itself.

A relevant example of using CNNs on DNA is scanning for binding motifs over a sequence [3]. A motif is a set combination of a few bases. The convolutional layer scans for different motifs in a wide input layer. The output of this layer shows where the motifs are found in the input. A pooling layer then decreases the output of this layer to a smaller size, for every couple of base pairs each filter is indicated to be present or not in this layer. Further convolutional layers can then look for patterns in the presence of the previous filters. In the DNA context, this could be looking for TFBS motifs, and then scanning for combinations of these motifs. The way that CNN's can scan the genome makes them suited for functional genomics.

Neural networks in regulatory genomics can make predictions about the regulatory network of human DNA

Neural network models have already been used to make predictions about the regulatory network of human DNA. A paper from 2018 trains a model using ChIP-seq data to predict enhancers and promotors from a DNA sequence for the entire genome[20]. However, the paper itself mentions the challenge of validation for the model, a complete database of regulatory elements does not yet exist. Another paper from 2018 introduces the DeFine tool [49]. The DeFine tool is able to predict binding intensities of transcription factors to certain DNA sequences. It can also assess the effect of SNP's and other variants on the binding intensity of transcription factors. This model is also trained using ChIP-seq data. Deep learning on ChIP-seq data has also been used to predict the binding sequence of DNA binding proteins [3]. All of these established models use ChIP-seq data to train their models, the models mostly only predict one TF at a time, because ChIP-seq can only look at one TF at a time.

ATAC-seq data is imprecise compared to ChIP-seq data. In ATAC-seq it is very often unsure whether a peak should be seen as noise or as an actual result. For this reason, not many models that predict TFBS from ATAC-seq data exist yet. Two recent papers present models that attempt to achieve this.

The first paper introduces maxATAC [11]. It uses ATAC-seq data, transformed into footprint scores, combined with the sequence itself as an input. maxATAC trains 127 separate models for 127 different transcription factors. The output of a model is the predicted binding sites for one specific TF over the entire genome. The fact that this model can only predict one TF at a time is a big disadvantage, as the advantage of using ATAC-seq over ChIP-seq is that ATAC-seq can in theory detect all TFs at the same time, whereas ChIP-seq can only look at one TF at a time. A model that predicts all human TFs at the same time would thus be a big improvement.

The second paper introduces TAMC [53]. TAMC also predicts TFBS from ATAC-seq data. An interesting feature of TAMC is that the bias correction step of ATAC-seq data preparation is not needed for this tool, making it more of an end-to-end tool which saves time during the analysis. The model uses a 1D-CNN to predict the binding probability. The model combines footprinting scores from the TOBIAS pipeline[6] with data from the footprinting analysis by HINT-ATAC[21]. HINT-ATAC adds information about cleavage preferences of the Tn5 protein in different locations on the genome to the analysis. HINT-ATAC leaves out the bias correction steps of the models. This combined data is then put into the 1D-CNN to make predictions. TAMC performs a bit better than TOBIAS in predicting TFBS, but not by much. The most interesting feature of the model is how bias correcting for the ATAC-seq is not needed to still obtain meaningful results. However, it also requires 2 different models to be run as input.

Interpreting neural network predictions is crucial for biological relevance and understandability

Neural networks can give highly accurate predictions for complex systems. This is useful in the field of functional genomics. However, neural networks are lacking one crucial feature in this field. In functional genomics, it is often just as important to know why a gene or an enhancer is predicted to be relevant in a cell, as knowing which gene is predicted to be important. Because of the inherent complexity of neural networks, results can be hard to interpret or understand from a biological point of view. Luckily there are a couple of ways to make the results interpretable.

Feature importance scores are scores given to the inputs of a neural network, describing the influence on the output prediction for each node of the input. This can be done for every single input kernel the model receives. This is a way of interpreting which parts of the input were important for making the prediction. The most common and easy way to do this is by using backpropagation [41]. Backpropagation calculates the derivative of the loss function of the prediction for each node in the neural network. This essentially translates to the amount of influence each point has on the outcome of the network. Nodes with a very high or very negative derivative thus have a big influence on the outcome and could hint at the important features of the input. The other way of calculating feature importance scores is by changing each input node in turn and seeing the effect on the output [16]. This approach is more computationally demanding than calculating derivatives. However, when using DNA sequence as input, this approach is very similar to in vitro saturation mutagenesis where each base pair is edited to study the effects of each individual base [19]. This makes it interesting to compare to findings from these mutagenesis experiments. These feature importance scores can also be visualized, in image recognition, this is common practice to discover what part of the image each layer of the neural network can detect. However, for DNA and motifs, this is a bit more complicated. The same motif can play a role in many different places of the DNA, so it is not possible to just look at the importance scores of multiple samples and derive motifs directly from this, the motif might be

active in a different spot in different samples [14]. Also, if multiple motifs can activate a certain gene, losing one of these genes could still lead to the activation of the gene. If one of these motifs is edited the activity of the gene stays high, and the motif gets a low importance score, where it should have had a higher one. Combining important regions by aligning them has been shown to lead to better predictions of motifs[41].

Another, more recent method of interpreting neural networks is by using SHapely Additive exPlanations (SHAP). The paper by Lundberg and Lee first shows 6 existing models for explaining predictions and shows the high similarity between the models [23]. The paper then combines the models and proposes SHAP values as a unified approach to interpreting models. The shapely additive value is based on the accuracy, missingness and consistency of predictions. Accuracy is putting very similar input into the original model and into the prediction model leading to very similar results. Missingness is when a feature is missing from the input, this feature does not influence the prediction, meaning that only features that are present can influence the predictions. Consistency means that the contribution of a feature should not change in direction when the model changes. So a feature should not change from a positive to a negative contribution if the prediction model changes. SHAP values can use all these three qualities of good predictions. The authors say this is what makes SHAP values the superior measurement of feature importance compared to the others. The biggest problem with SHAP values is the high computational cost of the calculation of the values. A solution for this is approximating the SHAP values using fewer samples, this saves computational time but still gives a robust result. Multiple methods for estimating the SHAP values are presented by the paper. The paper suggests Shapley Kernel as a good way to approach the SHAP values, as this method can be applied to explain any model, unlike the other SHAP estimations presented. Some methods of estimating the SHAP values can only be used on logistic models, or only on neural networks. Kernel SHAP can always be used.

In this paper, machine learning methods and ATAC-seq data will be combined to answer biological questions. The main question that will be examined in this paper is whether footprint scores can predict where transcription factor binding sites are. This will be examined by researching whether a logistic regression on footprint scores can predict the location of TFBS, if a convolutional neural network using footprint scores as input can predict where TFBS are and which of the two gives better predictions. Simultaneously, the question of whether a neural network can extract extra information from the spatial distribution of footprint scores in the input windows will be examined. The question of where this potential extra information is located in the input windows will be answered by looking at different window sizes as an input for the neural networks.

To further examine the usefulness of footprinting analysis, the same questions will be answered using uncorrected read counts from ATAC-seq data as input for the models. The difference between the models using footprints and the models using read counts will show how much footprinting contributes to making TFBS predictions. The questions that will be answered are whether read counts in a logistic regression can predict where TFBS are, whether read counts in a CNN can predict where TFBS are and which of the two is better. Whether and which extra information the CNN can extract from the spatial distribution of read counts will be examined by looking at different window sizes as input for the model and comparing the results.

Lastly, these two approaches will be compared to two alternative ways of predicting TFBS locations. The first alternative method is the BINDetect program from the TOBIAS workflow, which can be seen as the current state of the art for predicting TFBS. By comparing to BINDetect the question is whether adding position weight matrices of known binding site motifs can improve predictions of TFBS. The other method of predicting TFBS location is by extracting the shap values of a neural

network that predicts which parts of the chromatin are open, and which are closed. Transcription factors have the ability to open up chromatin, so in theory, the parts that the model finds important for predicting open chromatin should overlap with binding sites for these transcription factors (see methods). All methods of predicting TFBS locations will be compared to each other to make a final conclusion about which method to analyze ATAC-seq data is best.

Methods

Unibind database as a golden standard

To gain insight into the value of footprinting analysis when predicting transcription factor binding sites, the footprint scores need to be compared with a golden standard. The reference for TFBS that will be used in this paper is the Unibind database[35]. Unibind is a database containing experimentally verified TFBS from thousands of ChIP-seq datasets, annotated for specific cell lines. This database is of course not complete, as new TFBS are still being discovered. For the rest of this paper, the Unibind entries for the HEPG2 cell line will be looked at.

Footprinting analysis

The Tobias package[6] was used to create footprint scores from the ATAC-seq data (Fig. 3). First the program 'ATACorrect' was used to correct the reads for Tn5 cleavage bias. The original ATAC-seq data is measured in the number of reads per base pair. After the cleavage corrections, the data is still continuous, and can now be interpreted as more cut sites than expected when the measurement is positive and fewer cut sites than expected when the measurement is negative. Next, the program 'ScoreBigWig' from the Tobias package is used to extract footprint scores from the corrected reads. The output data from this analysis can be interpreted as higher evidence of binding for a higher footprint score. For both programs, the standard settings were used. The programs require a .bed file containing the ATAC peaks. The ATAC-seq data used for all analyses comes from a HEPG2 cell line from the Encode database in .bam format, along with the called peaks from this dataset in a .bedgraph format[43]. The dataset used was built on chromosome version GRCh38.

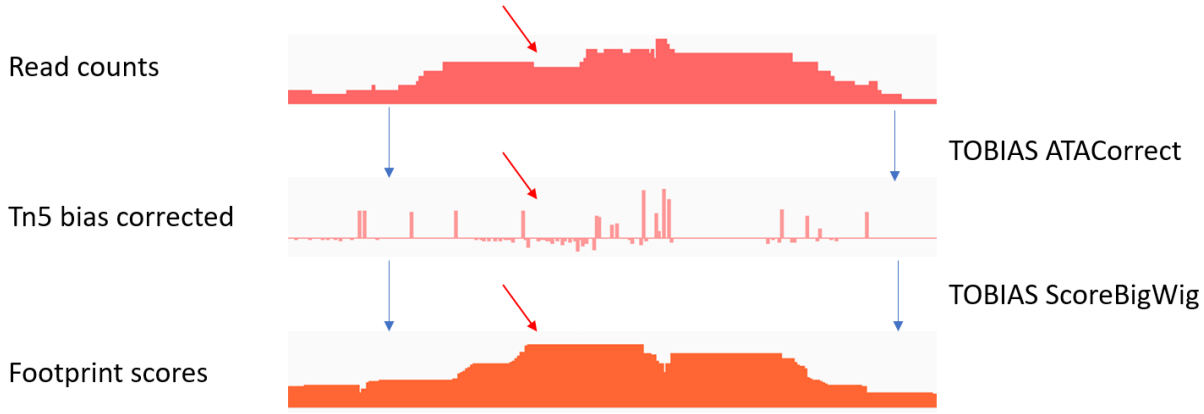

Figure 3. Overview of the bias correction and footprinting programs of the TOBIAS package. ATACcorrect corrects the read counts on Tn5 bias and translates the reads into a measurement of more or fewer cut sites than expected per position. ScoreBigWig uses the corrected data to do footprinting analysis. Footprint scores can be interpreted as a measurement of evidence of a protein binding there. The red arrow shows an example of a footprint, The area surrounding this part has more reads than the area itself, leading to a high footprint score.

Logistic regression

The logistic regression classifiers in this paper were trained using Scitkit learn[32] in python using the function 'sklearn.linear_model.LogisticRegression'. Version 1.1.0 of scikit learn was used. version 3.10.4 of python was used here and in the rest of the paper. After training the classifiers, the values for the intersect and the coefficient of the logistic regression can be extracted. To find the threshold above which the logistic regression classifier makes positive predictions, the intercept is divided by the negative of the coefficient: $threshold = \frac{intercept}{-coefficient}$. Links to the code can be found in the appendix.

Convolutional neural networks

Different CNN's are used in this paper. The original CNN prototype that predicts binary chromatin opening states from one-hot encoded DNA sequences is made by Kevin Kenna's lab, an overview of this model is given in figure 5. The other CNN's are an adaptation of this original one. These other CNN's take windows of values of the specified window size by 1 as an input, filled with either the footprints scores or the read counts. First the input goes through two convolutional layers with a kernel size 12 with ReLU activation. Then a max pooling layer is used of pool size 3. After this comes a dropout layer of 25%. After this comes another convolutional layer of kernel size 2, followed by another pooling and dropout layer. Then the output is flattened and put into two densely connected layers, the former of shape 15 with ReLU activation, and the latter of shape 2 with SoftMax activation. The y labels used to train the model are a tensor of two by one, representing the chance of the middle base pair of the input window (for the 500 base pair window, that is the 250th position) of overlapping with an entry in the Unibind database and one minus the chance of overlapping with Unibind. The models are trained for 10 epochs. The packages TensorFlow[1] and keras are used to create the models in R. Version 2.4.0 of TensorFlow and Keras were used. Version 3.6.1 of R was used. An overview of the model is given in figure 4.

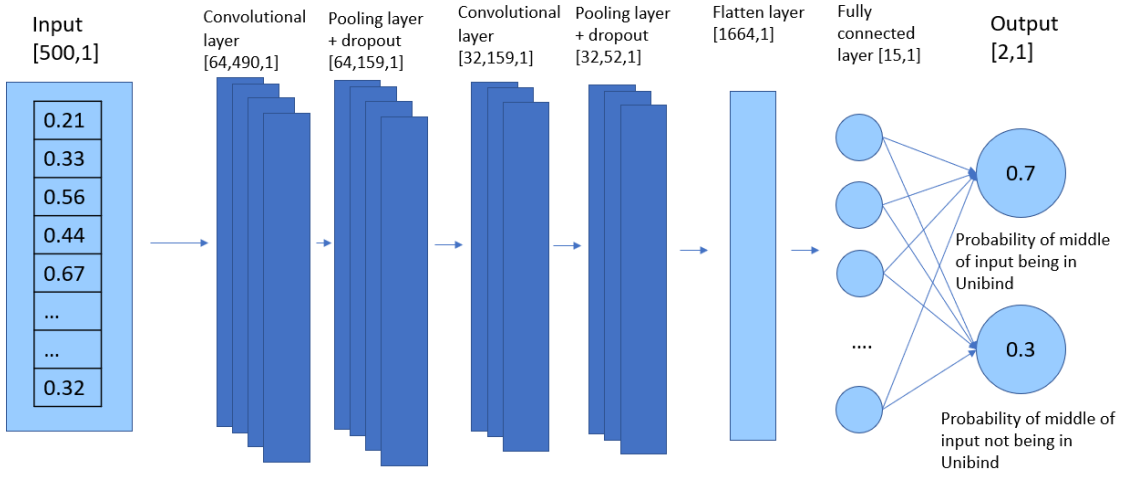

Figure 4. Example of the convolutional neural network used with a 500 base pair window. Each layer with its input shape is denoted at the top. The input values were either footprints scores or read counts. The input size is varied in the paper, and the rest of the model stayed the same. The output layer denotes the predicted probability of the middle base pair of the window being in Unibind and the predicted probability of the middle base pair not being in Unibind. Middle base pair being the (window size / 2)th base pair.

The original model that predicts binary openness of chromatin takes a tensor of the window size by four, filled with the one-hot-encoded DNA sequence of the windows as input (Fig 5). Each column here represents a different base, A, C, T or G. Each row represents a different position, a one is stated in each row in the corresponding column of the base present at that location, and the other values in that row are zero. Because of the bigger input size, one more convolutional layer is present in this model.

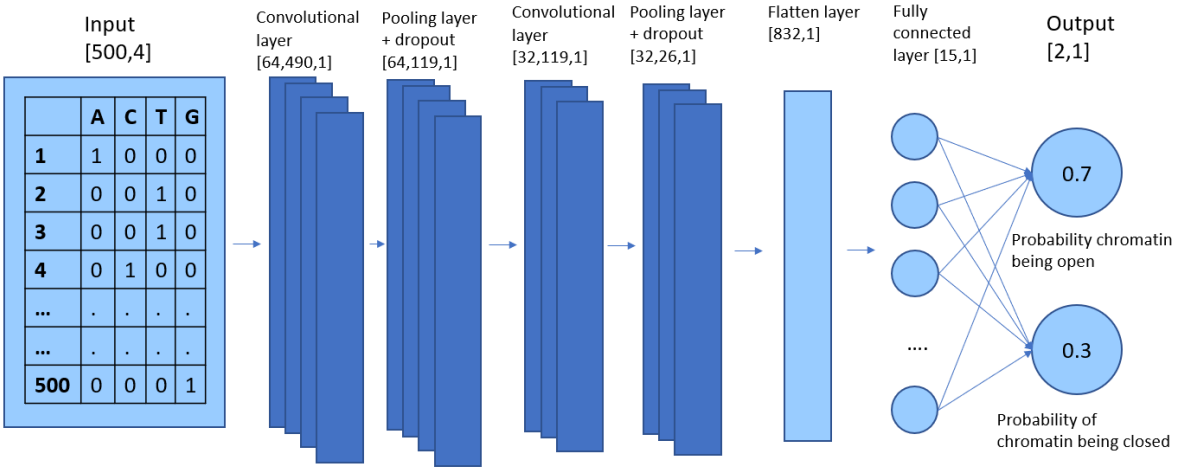

Figure 5. Overview of the original CNN that predicts binary chromatin opening states from one-hot encoded DNA sequences. Each layer with its input shape is denoted at the top. Input is a 500 by 4 tensor containing a one-hot-encoded DNA sequence of 500 base pairs. The input layer denotes the predicted probability of the chromatin in the input window being open and the predicted probability of the chromatin in the input window being closed.

Extraction of Shap values

After the CNN's were trained, the shap values were extracted using the DeepExplainer function from the shap package in python[23]. Version 0.39.0 of the shap package was used. The function was used on all base pairs of all inputs of the test set, resulting a bedgraph file with shap values for the entire ATAC-peaks in the test chromosomes. The code for extracting shap values was created by Kevin Kenna's lab and adapted for this paper.

Training and testing of classifiers

All methods are trained on the same training data points and tested on the same testing data points. The ATAC-sequencing reads and annotated ATAC-peaks of a HEPG2 dataset obtained from the Encode database[43] were used. 500.000 random genomic positions were sampled from taken from this dataset, in order to have a big dataset and keep calculation running swiftly. Of these random positions, 127.814 positions overlapped with a Unibind region, this is the positive dataset. The rest of the data points was reduced to equal the number of points in Unibind to keep the training set balanced, forming the negative dataset. The data was then split into train and test data points. Chromosomes 8,9,13 and 14 were used to test on, the rest as training data. This resulted in 17451 test points for the positive and negative classes, being Unibind-membership and non-Unibind-membership, and 110363 positive and negative data points for the training data.

Results

Footprinting analysis turns read counts from ATAC-seq into interpretable results for predicting binding events

To compare the footprint scores with the Unibind database a logistic regression was used. A higher footprint score should mean a higher chance of binding, so a logistic regression classifier is expected to predict binding sites for a specific cell type well, as a logistic regression looks only at the values of the input, so the logistic regression should be able to divided datapoints into bound and unbound. All footprint score peaks higher than the threshold, calculated as stated in the methods, are seen as bound when using a logistic regression. An example of this is given in figure 6.

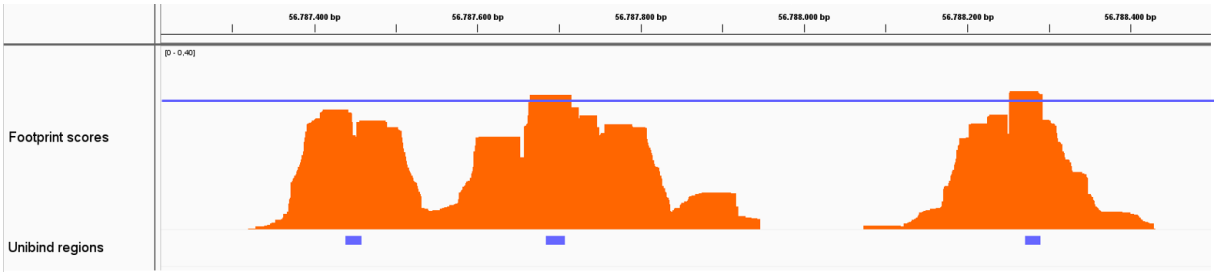

Figure 6. Footprint score track with logistic regression threshold. Orange bumps represent footprint scores. The blue threshold line is calculated from the intercept and coefficient from the logistic regression and is at 0.29. The threshold is calculated as mentioned in the methods. The lowest track displays annotated Unibind regions for this cell type

Using a logistic regression on footprint scores is a method that is being used already, the 'BINDetect' program of the Tobias package also does this. The program first scans the ATAC-peaks for known binding site motifs, then runs a logistic regression on the footprint score underlying these motifs to decide which motifs will be seen as bound, and which as unbound. The Logistic regression on footprint scores gives okay predictions for currently bound binding sites. As can be seen in figure 7A, the AUC for this classifier is 0.75.

Footprinting analysis classifies TFBS better than SHAP values from an open chromatin predicting neural network when used in a logistic regression.

Another way of predicting TFBS is by training a convolutional neural to classify open chromatin regions of the genome based on the DNA sequence. The model used uses the DNA sequences of ATAC-seq peaks as a positive input and random genomic sequences of the same size outside of the ATAC peaks as a negative input. This model is explained in the methods and in figure 5.

Certain transcription factors can cause the chromatin to open[55], making the DNA accessible for the transcriptional machinery. So in theory, the model could learn which parts of the DNA contribute most to opening up the chromatin and thus give a prediction for where the transcription factor binding sites are that cause the chromatin to open up. The DNA regions that contribute most to the prediction of the model are extracted by looking at Shap values given to certain positions in the input windows. The DNA windows that have the highest Shap values here could be seen as potential TFBS that help in opening up the chromatin. This method should only uncover TFBS that are relevant for the opening of chromatin, so there should also be TFBS that get low shap values if the corresponding bound transcription factor does not contribute to the chromatin being open or not.

Similar to the footprint analysis, a logistic regression was also done on the Shap values to predict Unibind regions for the HEPG2 cell line. High enough shap values should also lead to the classification of bound TF's, just like high footprint scores lead to the classification of bound TF's. The shap logistic regression seems to be underperforming, it gives an AUC of 0.56, which is very low when compared to the AUC of 0.75 of the footprint logistic regression. As mentioned before, the low performance could be due to the fact that this model should find specific TFBS that only help in opening the chromatin, whereas Unibind contains all sorts of TFBS.

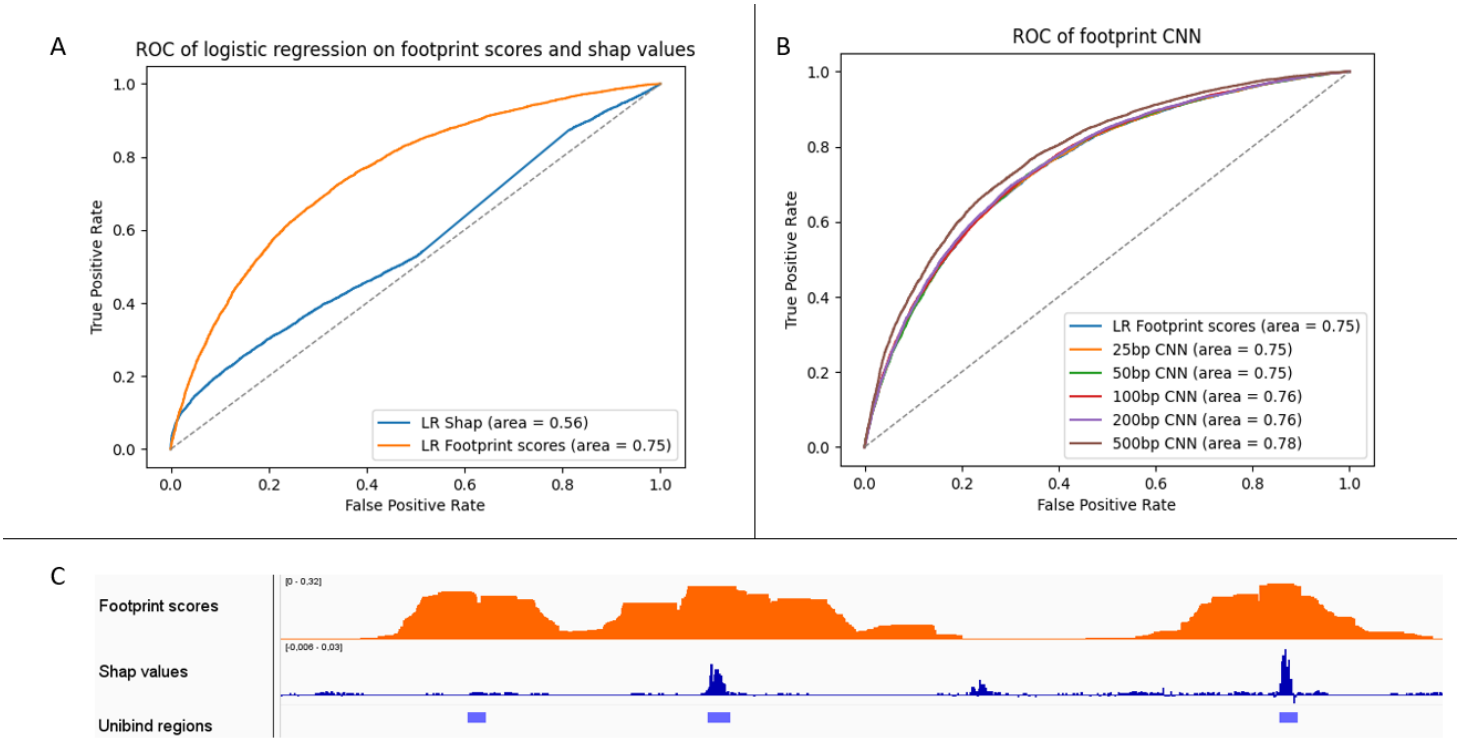

Figure 7. ROC curves of Unibind classifiers using footprint scores and Shap values. A. ROC curve of the performance of the logistic regression classifier on footprints scores and the logistic regression classifier on Shap values from the open chromatin model. AUC scores are noted in the legend. **B.** ROC curve of the performance of the convolutional neural network that takes footprint scores as an input. AUC curves for inputs of window sizes of 25, 50, 100, 200 and 500 base pairs are shown, as well as the AUC for logistic regression on footprint scores. AUC scores are noted in the legend. **C.** example genomic tracks showing footprint scores in orange, Shap values of the open chromatin model in dark blue and annotated Unibind regions for the HEPG2 cell line in light blue.

The footprint score CNN extracts little additional spatial information from footprint scores when predicting TFBS

The predictions made by logistic regressions are simple in nature. The height of the signal is the only factor that influences a prediction being made. In order to capture spatial information hidden in the distribution of the footprint scores, we reasoned that a convolutional network can be used to predict TFBS. A model was trained that takes as input the footprint scores over a window of a fixed size, and predict whether the base pair in the middle of this window is in Unibind or not, being the base pair at (window size / 2). The model can use information from around the base pair it is trying to predict in order to improve performance. The performance of this model will show whether extra information is hidden around the footprint score of the middle base pair. An input of different window sizes was

tested in order to discover where the extra information in spatial distribution around the middle base pair was located. The model was trained and tested with a window size of 25, 50, 100, 200 and 500 base pairs. An overview of this model is in figure 4.

The CNN barely improves on the performance of the logistic regression when predicting TFBS. Meaning that little extra information is present around the middle base pair in the window. This can be seen in figure 7B. All different CNN's here give an AUC between 0.75 and 0.78, compared to the 0.75 of the logistic regression, this is not a big improvement. The performance between the models that use different window sizes does not differ much. Performance of the model starts increasing slowly when windows of 500 bp are used.

The model is trained and tested on windows around random positions taken from the ATAC peaks. However, the model can also be used to make a prediction for every base pair within the ATAC-peaks. In a real-world context, the model would be used in this way in order to find the potential TFBS. To speed up calculations this prediction was made using a sliding window that moves every 5 base pairs, so for every 5 base pairs a prediction of unbind membership was made. The model performs better when looking at the ROC curve in figure 8A, with an AUC ranging from 0.82 to 0.84, because of the higher amount of negative samples in this test set. There is a big part of the ATAC-peaks that have a low footprint score and are not part of the Unibind database, these samples are easier to predict correctly. In the previous test set, there were as many positive as negative samples, which makes it harder to make a good prediction. The window size does again not influence the performance of the model much.

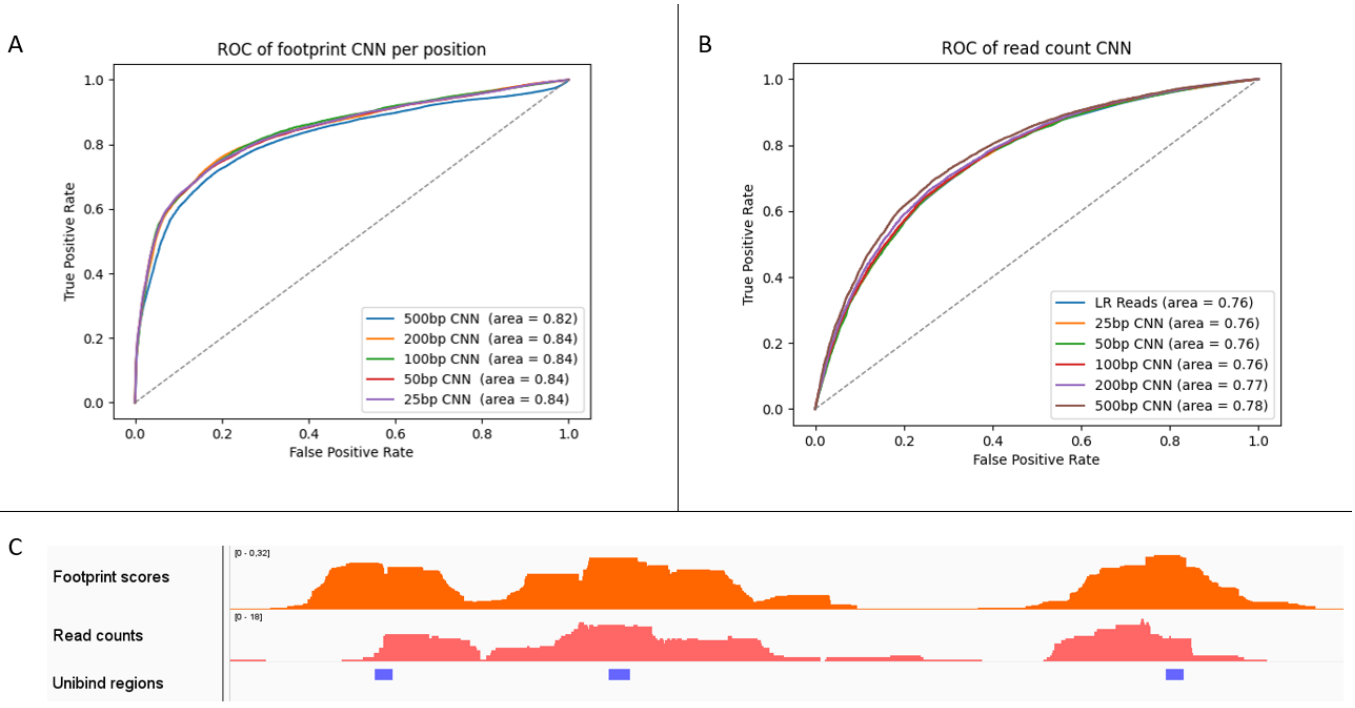

Figure 8. ROC curves of CNN Unibind classifiers using footprint scores and read counts. A. ROC curve of the performance of the footprint CNN when classifying every 5 base pairs within the ATAC-peaks. Shown for the 5 different window sizes. AUC scores are noted in the legend. **B.** ROC curve of the performance of the convolutional neural network that takes read counts as an input. AUC curves for inputs of window sizes of 25, 50, 100, 200 and 500 base pairs are shown, as well as the AUC for logistic

regression on read counts. AUC scores are noted in the legend. **C.** example genomic tracks showing footprint scores in orange, read counts of the ATAC-seq data in pink and annotated Unibind regions for the HEPG2 cell line in light blue.

A CNN using read counts as input performs predicts TFBS as well as a CNN using footprint scores as input

To examine the benefits of footprinting of ATAC-seq data in the prediction of TFBS, the CNN was also trained with raw sequencing read counts as input for the model. The reads are not corrected for Tn5 bias. The CNN works exactly the same, except for the difference in input data. As can be seen in figure 8B , the CNN using read counts performs just as well as the CNN using footprint scores, showing an AUC ranging from 0.76 to 0.78. This is similar to the performance of the footprint score CNN.

This shows that for the prediction of TFBS with a CNN, the footprint analysis and Tn5 bias correction provide no added benefit. Also, the logistic regression on the read counts performs as well as the logistic regression on footprint scores, with an AUC of 0.76 and 0.75 respectively. This shows that no improvement in prediction quality is obtained when footprinting analysis is done on the data. It may be that the CNN learns to take over some of the functionality of the bias correction and footprinting analysis, giving an explanation as to why both CNN's perform similarly. However, this would not explain why the two logistic regressions perform similarly as well.

The performance of the logistic regression and the CNN model on read counts is also similar. A small increase in performance can be seen when the window size of the model is large, however, the difference in AUC is small enough to conclude that the most important information should be in the central part of the windows, nothing is more informative for the prediction than the magnitude of the ATAC-seq signal at a given position.

In figure 9, all classifiers that were used are summarized. The logistic regression on read counts and on footprint scores, the CNN using read counts and the CNN using footprint scores as input, BINDetect and the logistic regression on shap values from the open chromatin CNN are compared. The highest AUC's come from the CNN's that use 500 base pair input windows, with either read counts or footprints, with an AUC of 0.78.

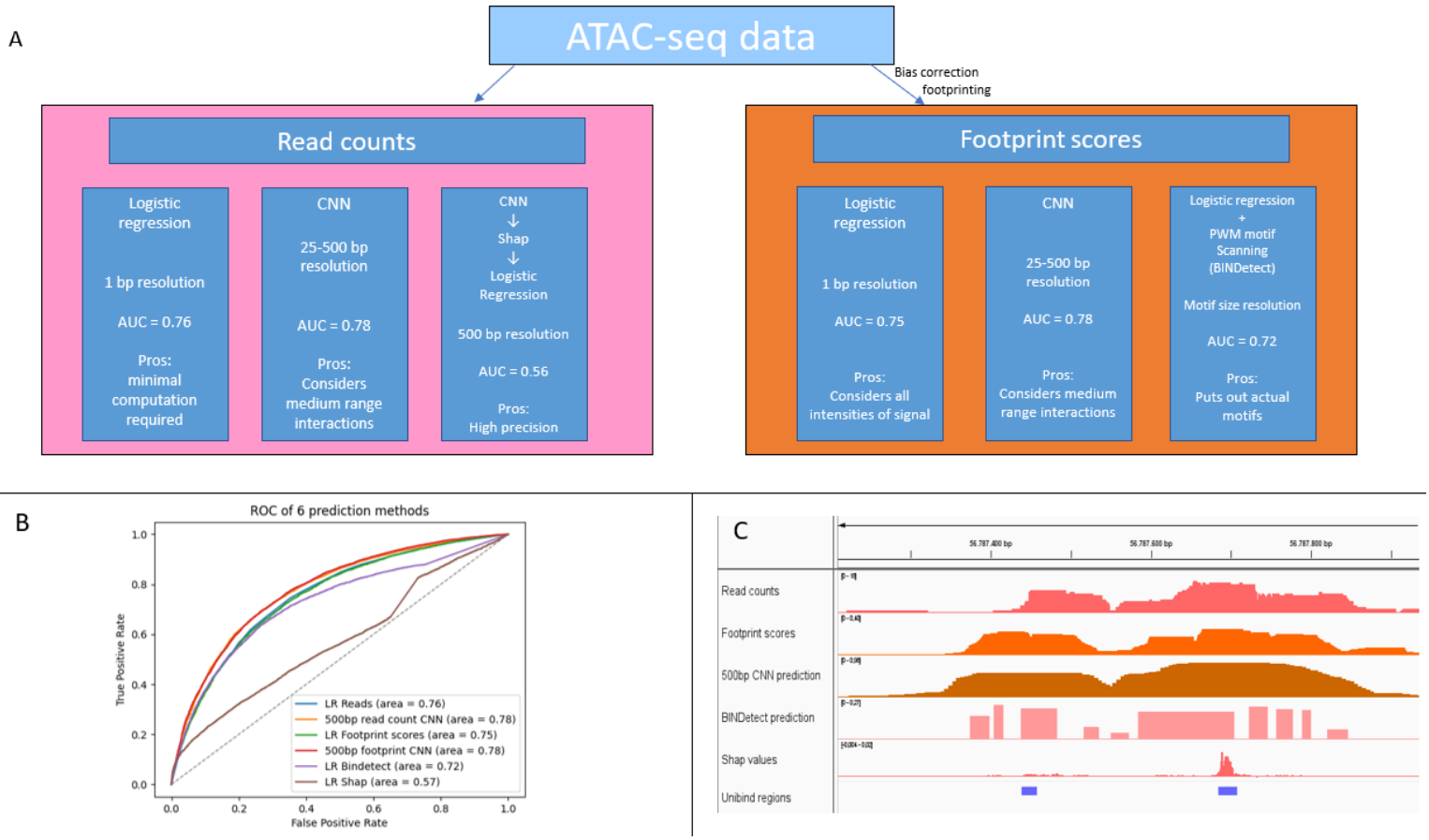

Figure 9. Summary of all used classifiers of Unibind membership. A. summary table of the 6 different methods split into methods that use read counts and methods that use footprint scores. For each method, the resolution or input size, the AUC and a benefit of the method are shown **B.** ROC curve of the performance of all different Unibind membership classifiers. For the CNN's the window size of 500 is shown. AUC scores are noted in the legend. **C.** example genomic tracks, from top to bottom: Read counts, footprint scores, predictions of Unibind membership made by the 500 bp footprint CNN, predictions of TFBS made by BINDetect, Shap values from the open chromatin CNN and Annotated Unibind regions for the HEPG2 cell line.

Conclusion

Six different methods to predict transcription factor binding sites were presented and discussed in this paper. An overview of the methods is given in figure 9A. The performance of all classifiers is summarized in figure 9B. The most obvious conclusion to be made is that the footprinting analysis does not boost performance in any of the classifiers used. Logistic regression on footprints performs similarly to logistic regression on read counts when predicting TFBS. Both show an AUC of around 0.75. The same story goes for the neural networks. Footprint scores are defined by TOBIAS as an estimation of evidence of binding. It is fair to say that in these classifiers the footprint scores fail to serve that purpose. In the TOBIAS paper, the method is presented as very promising, however as we see here that is not so much the case. The paper itself also compares footprint scores to read counts as a predictor for the location of TFBS in the supplemental information, here the footprints perform a little better than the read counts when used as a classifier, an improvement of around 0.03 in AUC. So this finding is in line with the TOBIAS paper.

However, footprinting scores have some good things going for them. Footprint scores are easier to interpret than raw read counts, footprint scores get scaled, making them easier to compare over the entire genome. Other than that, the read counts are noisier than the footprint scores, because there are small numbers of reads outside of the called peaks. Inside the called peaks the noise is reduced in footprint scores. As can be seen in figure 9C, footprints are smoother peaks than read counts are. Also, ATAC-peaks with relatively low read counts compared to the rest of the reads can still receive a relatively high footprint score because of the shape in the Tn5 bias-corrected read counts. These features would be lost when looking at read counts only. The qualitative difference in footprint scores and read counts is shown in figure 10, this shows the footprint scores and read counts rescaled to be between 0 and 1. If the footprint scores would only be a rescaling of read counts, the dots should form a diagonal line through the middle of the graph and the r-squared value should be high. A different shape in the points is seen and the r-squared value is 0.676, pointing to a difference in values. r-squared is calculated using Sci-kit learn[32]. These parts of the peaks that get raised in footprint scores even though having low read counts, do not boost the performance of the classifiers enough to make the footprinting perform better than the read counts in any classifier, but it does show there is a trade-off between the two methods for which parts are looked at. In theory, both methods give up performance in some areas and gain performance in other areas. The footprint scores should be able to consider low peaks and high peaks in the ATAC-data similarly, whereas read counts do theoretically not do this, but in practice, this does not lead to an advantage when using footprint scores as a classifier. The classifiers that use read counts as input performed just as well as the classifiers that use footprint scores.

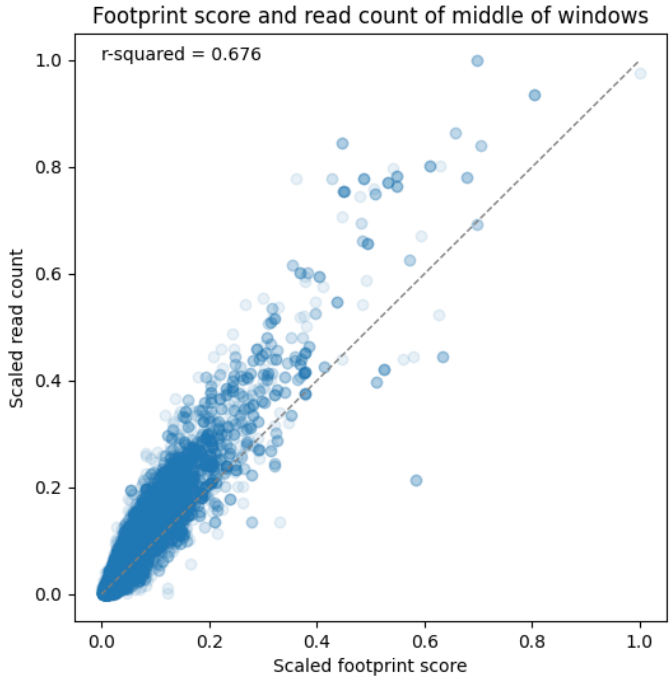

Figure 10. comparison of read count values and footprint scores of the middle of the test set windows. Scatterplot of read count values and footprint scores from the middle (250th) base pair of the test set windows. Both values are scaled between 0 and 1. (x = (x – min(x)) / (max(x) – min(x)). R-squared value of the linear model between the scaled footprints and the scaled read counts denoted at the top.

The convolutional networks used in the paper did not give the expected results. For both the footprint and read count networks, the logistic regression on the same data performed just as well. A logistic regression could be seen as analogous to a 1 bp window network model, as the only information the model would get is the value of the read count or footprint score at that centre position. Because TFBS motifs have a length of about 15 base pairs[4], it would be expected that the 25 bp neural networks could use the information for the entire motif to make better predictions than models that only use the centre base pair, this is not what was found in the models. The models only start to perform slightly better once the window size has been increased to 500 base pairs. When the window size starts getting this large, information about more distal elements such as enhancers and insulators could start to be seen by the model, potentially helping in making predictions. Bigger window sizes than 500 base pairs could maybe lead to better binding site predictions, this could be worth examining in future work. Not much is known yet about the exact workings of these long-range interactions, making it more difficult to examine. The CNN's used here were not able to extract much extra information from the spatial distribution of either footprint scores or read counts. The fact that the models performed slightly better with a window size of 500 shows that the small amount of extra information is not close to the central (250^{th}) base pair, but most likely is at least 100 base pairs away from the centre. A paper that does use long interaction to improve predictions of gene expression is the paper by Avsec *et al.* (2021)[5]. The model presented in this paper can integrate interactions up to 100 kilobases away from the centre into the predictions and improves on current methods of predicting gene expression. This shows that there is knowledge to be gained by studying these long-range interactions.

The Shap values that come from the open chromatin prediction network look very promising when visually inspected. They very often overlap exactly with the entries in the Unibind database (fig 9C), but even more often they miss the entries in the database. It looks like the precision of these predictions are good, but the problem lies in the sensitivity. The explanation is that the convolutional model most likely learns specific motifs, as shown by the shape of the Shap value peaks in figure 9C and gets very proficient at recognizing these. However, the motifs that are not learnt are also never recognized by the model. The BINDetect program from the TOBIAS package uses a list of motifs to make a prediction, there the motifs are combined with footprint scores to make the predictions. In the future, something similar could be done with the convolutional network in an attempt to increase the predictions. The BINDetect itself program does not perform better than the logistic regression on read counts or on footprint scores. This shows that adding motif position weight matrices does not make for better predictions. However, the scanning for motifs at the same time as doing the classifying does make for results that are easier to interpret and investigate further. Other papers do show a use for adding predetermined TFBS motifs to a model, for example, the paper by Ma *et al.* (2018)[24] links known motifs to each node of a layer to make the model transparent and more interpretable.

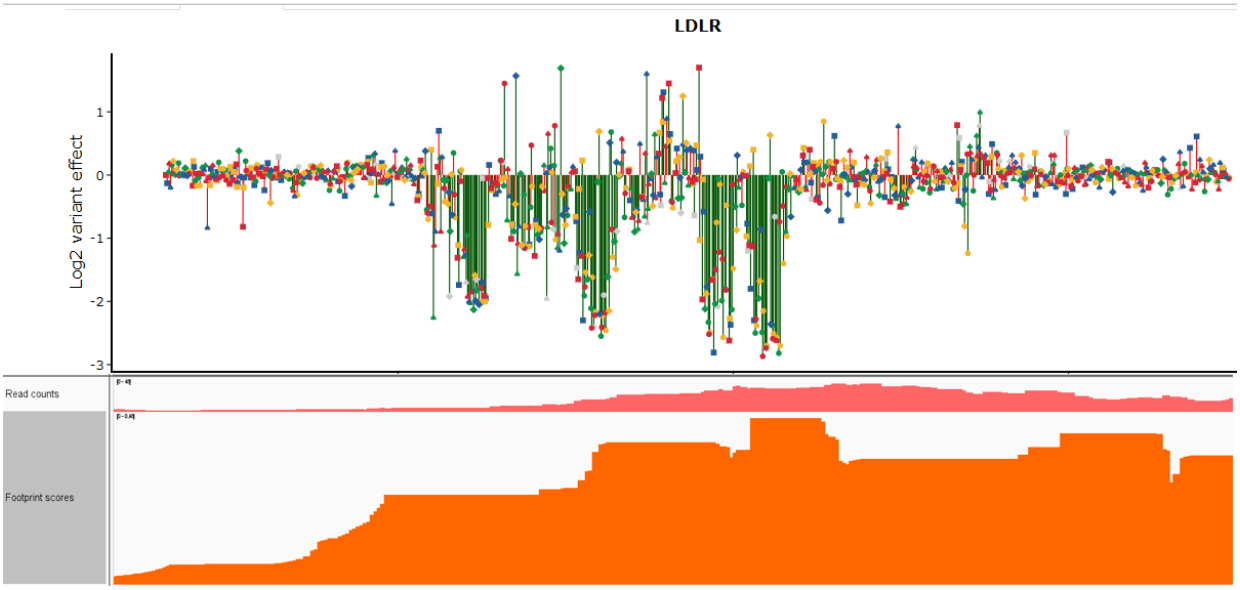

Figure 11. Mutagenesis plot aligned with footprint score track. The upper part of the figure shows the mutagenesis assay results from Kircher *et al* (2019)[19]. low points on the graph mean expression of the reporter gene decreased after a mutation in this position. High points on the graph mean expression of the reporter gene increased after a mutation in that position. The lower part of the figure shows the corresponding footprint scores in orange and the corresponding read counts in pink.

The computational methods used and discussed in this paper did not perform as expected, footprinting did not increase performance when predicting TFBS locations, neural networks added minimal information to the classification, the method of BINDetect that uses motif position weight matrices did not perform better and the use of shap values from the open chromatin model to predict TFBS did not pan out great. The logistic regression on read counts was in the end the method that worked best and is also the method that required the least amount of computational techniques. Maybe the future for ATAC-seq data is not in computational methods, after all, maybe the future lies in advancing the experimental technique itself to reduce noise and overcome bias. On a more optimistic note, the computational methods do all show some individual benefits. For example, the shap values from the open chromatin model show high precision in predicting the size and location of the motifs. The footprinting analysis is capable of considering every present ATAC-peak, small or big. Convolutional neural networks improve on performance slightly when using bigger window sizes, there may be more to be gained here when examined further. Maybe a combined usage of computational methods gives a better perspective on TFBS, but work is still to be done.

As an alternative to the in silico methods discussed in this paper, saturation mutagenesis is an in vitro assay that can be used to look for TFBS. In saturation mutagenesis, as done in the paper by Kircher *et al.* (2019)[19], known regulatory elements are mutated base by base after which the effect of the mutations in living cells is studied. Each base pair is mutated in every other base pair to study all possible effects. Mutations that influence the expression of the reporter gene significantly are clearly visible in a mutagenesis plot and show sizes similar to known TFBS. In figure 11, the mutagenesis result is compared with a footprint score track. Some clear overlap between the two methods is visible, a striking advantage of mutagenesis is the clarity of where TFBS start and stop, something that footprinting scores are yet unable to detect. Sadly, Mutagenesis data is harder to obtain than ATAC-seq data. For this reason, no big databases for mutagenesis data exist yet, Unibind is way more extensive than any mutagenesis TFBS datasets. The paper by Kircher *et al.* (2019) only covers 21 regulatory elements, whereas ATAC-seq data covers the entire genome. Some hopes for the future could be more mutagenesis data, as it should be the most accurate estimation of active regulatory

elements. The pitfalls for mutagenesis are that multiple binding sites could need to be active in combination to have an effect, as mutagenesis only mutates one base pair at a time, these events will not be captured by mutagenesis analysis. Combining the missing information from mutagenesis and clear peaks from ATAC-seq analysis could again lead to hints for which regulatory elements cooperate on the genome. More mutagenesis data would lead to clear answers, however, the power of ATAC-seq data lies in the fact that it is so easy and quick to obtain, so the perfect answers would be in understanding and using ATAC-seq data better instead of hoping for more mutagenesis data.

An important limitation to be considered for this paper is the Unibind database. The database was used for every analysis in this paper, so the completeness and robustness of the database are important. Only the ATAC-seq data of a HEPG2 cell line was looked at in this paper, so only the Unibind entries for HEPG2 cells were considered. It could be that the Unibind database is more or less accurate for other cell lines if those cell lines have been studied more or less often in ChIP-seq studies. So it could also be that our results would be different for other cell lines. The Unibind database contains 164 HEPG2 datasets and for example 31 HeLa cell datasets and 60 HEK293 datasets, so a difference in coverage per cell line is present in the Unibind database. In total Unibind covers 9654 ChIP-seq datasets, of which 4659 are human datasets. Unibind currently contains 841 different transcription factors in its database. Older papers have already shown that more than 1500 human transcription factors exist[50]. The database is not complete, this is a big limitation of the analyses in this paper. The incompleteness of the database could explain a lot of false positives that would be true positives in the classifiers. Alternatives to Unibind do exist, but these use ChIP-seq data as well[12,54]. The performance of these computational methods will not improve when using these other databases if they are not more complete.

Key acronyms

ATAC-seq	Assay for transposase accessible chromatin sequencing
ChIP-seq	chromatin immunoprecipitation sequencing
TF	transcription factor
TFBS	transcription factor binding site
LR	logistic regression
CNN	convolutional neural network
ROC curve	Receiver operating characteristic curve
AUC	area under the curve

Appendix: Code

Parts of the code that was used can be found on https://github.com/thijs32/Investigation-of-footprinting-analysis

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., … Zheng, X. (n.d.). *TensorFlow: A system for large-scale machine learning*.
2. Adey, A., Morrison, H. G., Asan, Xun, X., Kitzman, J. O., Turner, E. H., Stackhouse, B., MacKenzie, A. P., Caruccio, N. C., Zhang, X., & Shendure, J. (2010). Rapid, low-input, low-bias construction of shotgun fragment libraries by high-density in vitro transposition. *Genome Biology*, *11*(12), R119. https://doi.org/10.1186/gb-2010-11-12-r119
3. Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, *33*(8), Article 8. https://doi.org/10.1038/nbt.3300
4. Aptekmann, A. A., Bulavka, D., Nadra, A. D., & Sánchez, I. E. (2022). Transcription factor specificity limits the number of DNA-binding motifs. *PLOS ONE*, *17*(1), e0263307. https://doi.org/10.1371/journal.pone.0263307

5. Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., & Zeitlinger, J. (2021). Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, *53*(3), Article 3. https://doi.org/10.1038/s41588-021-00782-6
6. Bentsen, M., Goymann, P., Schultheis, H., Klee, K., Petrova, A., Wiegandt, R., Fust, A., Preussner, J., Kuenne, C., Braun, T., Kim, J., & Looso, M. (2020). ATAC-seq footprinting unravels kinetics of transcription factor binding during zygotic genome activation. *Nature Communications*, *11*(1), Article 1. https://doi.org/10.1038/s41467-020-18035-1
7. Boyle, A. P., Davis, S., Shulha, H. P., Meltzer, P., Margulies, E. H., Weng, Z., Furey, T. S., & Crawford, G. E. (2008). High-Resolution Mapping and Characterization of Open Chromatin across the Genome. *Cell*, *132*(2), 311–322. https://doi.org/10.1016/j.cell.2007.12.014
8. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., & Greenleaf, W. J. (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods*, *10*(12), Article 12. https://doi.org/10.1038/nmeth.2688
9. Buenrostro, J. D., Wu, B., Chang, H. Y., & Greenleaf, W. J. (2015). ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. *Current Protocols in Molecular Biology*, *109*(1), 21.29.1-21.29.9. https://doi.org/10.1002/0471142727.mb2129s109
10. Buenrostro, J. D., Wu, B., Litzenburger, U. M., Ruff, D., Gonzales, M. L., Snyder, M. P., Chang, H. Y., & Greenleaf, W. J. (2015). Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, *523*(7561), Article 7561. https://doi.org/10.1038/nature14590
11. Cazares, T., Rizvi, F. W., Iyer, B., Chen, X., Kotliar, M., Wayman, J. A., Bejjani, A., Donmez, O., Wronowski, B., Parameswaran, S., Kottyan, L. C., Barski, A., Weirauch, M. T., Prasath, V. S., & Miraldi, E. R. (2022). *maxATAC: Genome-scale transcription-factor binding prediction from ATAC-seq with deep neural networks* (p. 2022.01.28.478235). bioRxiv. https://doi.org/10.1101/2022.01.28.478235
12. Chen, D., Jiang, S., Ma, X., & Li, F. (2017). TFBSbank: A platform to dissect the big data of protein–DNA interaction in human and model species. *Nucleic Acids Research*, *45*(D1), D151–D157. https://doi.org/10.1093/nar/gkw1035
13. Crawford, G. E., Davis, S., Scacheri, P. C., Renaud, G., Halawi, M. J., Erdos, M. R., Green, R., Meltzer, P. S., Wolfsberg, T. G., & Collins, F. S. (2006). DNase-chip: A high-resolution method to identify DNase I hypersensitive sites using tiled microarrays. *Nature Methods*, *3*(7), Article 7. https://doi.org/10.1038/nmeth888
14. Eraslan, G., Avsec, Ž., Gagneur, J., & Theis, F. J. (2019). Deep learning: New computational modelling techniques for genomics. *Nature Reviews Genetics*, *20*(7), 389–403. https://doi.org/10.1038/s41576-019-0122-6
15. Goryshin, I. Y., & Reznikoff, W. S. (1998). Tn5 in Vitro Transposition *. *Journal of Biological Chemistry*, *273*(13), 7367–7374. https://doi.org/10.1074/jbc.273.13.7367
16. Kelley, D. R., Snoek, J., & Rinn, J. L. (2016). Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, *26*(7), 990–999. https://doi.org/10.1101/gr.200535.115
17. Kellis e.a. (n.d.). *Defining functional DNA elements in the human genome*. PNAS. Retrieved March 6, 2022, from https://www.pnas.org/doi/abs/10.1073/pnas.1318948111
18. Kelsey, G., Stegle, O., & Reik, W. (2017). Single-cell epigenomics: Recording the past and predicting the future. *Science*, *358*(6359), 69–75. https://doi.org/10.1126/science.aan6826
19. Kircher, M., Xiong, C., Martin, B., Schubach, M., Inoue, F., Bell, R. J. A., Costello, J. F., Shendure, J., & Ahituv, N. (2019). Saturation mutagenesis of twenty disease-associated regulatory elements at single base-pair resolution. *Nature Communications*, *10*(1), Article 1. https://doi.org/10.1038/s41467-019-11526-w
20. Li, Y., Shi, W., & Wasserman, W. W. (2018). Genome-wide prediction of cis-regulatory regions using supervised deep learning methods. *BMC Bioinformatics*, *19*(1), 202. https://doi.org/10.1186/s12859-018-2187-1
21. Li, Z., Schulz, M. H., Look, T., Begemann, M., Zenke, M., & Costa, I. G. (2019). Identification of transcription factor binding sites using ATAC-seq. *Genome Biology*, *20*(1), 45. https://doi.org/10.1186/s13059-019-1642-2
22. Lu, Z., Hofmeister, B. T., Vollmers, C., DuBois, R. M., & Schmitz, R. J. (2017). Combining ATAC-seq with nuclei sorting for discovery of cis-regulatory regions in plant genomes. *Nucleic Acids Research*, *45*(6), e41. https://doi.org/10.1093/nar/gkw1179

23. Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html
24. Ma, J., Yu, M. K., Fong, S., Ono, K., Sage, E., Demchak, B., Sharan, R., & Ideker, T. (2018). Using deep learning to model the hierarchical structure and function of a cell. *Nature Methods*, *15*(4), Article 4. https://doi.org/10.1038/nmeth.4627
25. Martins, A. L., Walavalkar, N. M., Anderson, W. D., Zang, C., & Guertin, M. J. (2018). Universal correction of enzymatic sequence bias reveals molecular signatures of protein/DNA interactions. *Nucleic Acids Research*, *46*(2), e9. https://doi.org/10.1093/nar/gkx1053
26. Meuleman, W., Muratov, A., Rynes, E., Halow, J., Lee, K., Bates, D., Diegel, M., Dunn, D., Neri, F., Teodosiadis, A., Reynolds, A., Haugen, E., Nelson, J., Johnson, A., Frerker, M., Buckley, M., Sandstrom, R., Vierstra, J., Kaul, R., & Stamatoyannopoulos, J. (2020). Index and biological spectrum of human DNase I hypersensitive sites. *Nature*, *584*(7820), Article 7820. https://doi.org/10.1038/s41586-020-2559-3
27. Meyer, C. A., & Liu, X. S. (2014). Identifying and mitigating bias in next-generation sequencing methods for chromatin biology. *Nature Reviews Genetics*, *15*(11), Article 11. https://doi.org/10.1038/nrg3788
28. Minnoye, L., Taskiran, I. I., Mauduit, D., Fazio, M., Aerschot, L. V., Hulselmans, G., Christiaens, V., Makhzami, S., Seltenhammer, M., Karras, P., Primot, A., Cadieu, E., Rooijen, E. van, Marine, J.-C., Egidy, G., Ghanem, G. E., Zon, L., Wouters, J., & Aerts, S. (2020). Cross-species analysis of enhancer logic using deep learning. *Genome Research*, gr.260844.120. https://doi.org/10.1101/gr.260844.120
29. Moore, J. E., Purcaro, M. J., Pratt, H. E., Epstein, C. B., Shoresh, N., Adrian, J., Kawli, T., Davis, C. A., Dobin, A., Kaul, R., Halow, J., Van Nostrand, E. L., Freese, P., Gorkin, D. U., Shen, Y., He, Y., Mackiewicz, M., Pauli-Behn, F., Williams, B. A., … Weng, Z. (2020). Expanded encyclopaedias of DNA elements in the human and mouse genomes. *Nature*, *583*(7818), Article 7818. https://doi.org/10.1038/s41586-020-2493-4
30. Nielsen, A. A. K., & Voigt, C. A. (2018). Deep learning to predict the lab-of-origin of engineered DNA. *Nature Communications*, *9*(1), Article 1. https://doi.org/10.1038/s41467-018-05378-z
31. Noonan, J. P., & McCallion, A. S. (2010). Genomics of long-range regulatory elements. *Annual Review of Genomics and Human Genetics*, *11*, 1–23. https://doi.org/10.1146/annurev-genom-082509-141651
32. 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., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.
33. Pennacchio, L. A., Bickmore, W., Dean, A., Nobrega, M. A., & Bejerano, G. (2013). Enhancers: Five essential questions. *Nature Reviews. Genetics*, *14*(4), 288–295. https://doi.org/10.1038/nrg3458
34. Pott, S., & Lieb, J. D. (2015). Single-cell ATAC-seq: Strength in numbers. *Genome Biology*, *16*(1), 172. https://doi.org/10.1186/s13059-015-0737-7
35. Puig, R. R., Boddie, P., Khan, A., Castro-Mondragon, J. A., & Mathelier, A. (2021). UniBind: Maps of high-confidence direct TF-DNA interactions across nine species. *BMC Genomics*, *22*(1), 482. https://doi.org/10.1186/s12864-021-07760-6
36. Qin, Q., Mei, S., Wu, Q., Sun, H., Li, L., Taing, L., Chen, S., Li, F., Liu, T., Zang, C., Xu, H., Chen, Y., Meyer, C. A., Zhang, Y., Brown, M., Long, H. W., & Liu, X. S. (2016). ChiLin: A comprehensive ChIP-seq and DNase-seq quality control and analysis pipeline. *BMC Bioinformatics*, *17*(1), 404. https://doi.org/10.1186/s12859-016-1274-4
37. Qin, Z. S., Yu, J., Shen, J., Maher, C. A., Hu, M., Kalyana-Sundaram, S., Yu, J., & Chinnaiyan, A. M. (2010). HPeak: An HMM-based algorithm for defining read-enriched regions in ChIP-Seq data. *BMC Bioinformatics*, *11*(1), 369. https://doi.org/10.1186/1471-2105-11-369
38. Quach, B., & Furey, T. S. (2017). DeFCoM: Analysis and modeling of transcription factor binding sites using a motif-centric genomic footprinter. *Bioinformatics*, *33*(7), 956–963. https://doi.org/10.1093/bioinformatics/btw740
39. Reznikoff, W. S. (2003). Tn5 as a model for understanding DNA transposition. *Molecular Microbiology*, *47*(5), 1199–1206. https://doi.org/10.1046/j.1365-2958.2003.03382.x
40. Rye, M. B., Sætrom, P., & Drabløs, F. (2011). A manually curated ChIP-seq benchmark demonstrates room for improvement in current peak-finder programs. *Nucleic Acids Research*, *39*(4), e25. https://doi.org/10.1093/nar/gkq1187

41. Shrikumar, A., Greenside, P., Shcherbina, A., & Kundaje, A. (2017). Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. *ArXiv:1605.01713 [Cs]*. http://arxiv.org/abs/1605.01713
42. Shrikumar, A., Tian, K., Avsec, Ž., Shcherbina, A., Banerjee, A., Sharmin, M., Nair, S., & Kundaje, A. (2020). Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5. *ArXiv:1811.00416 [Cs, q-Bio, Stat]*. http://arxiv.org/abs/1811.00416
43. Snyder, M. (2020). *ENCSR042AWH* [Data set]. Stanford University. https://doi.org/10.17989/ENCSR042AWH
44. Song, L., & Crawford, G. E. (2010). DNase-seq: A high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells. *Cold Spring Harbor Protocols*, *2010*(2), pdb.prot5384. https://doi.org/10.1101/pdb.prot5384
45. Stalder, J., Larsen, A., Engel, J. D., Dolan, M., Groudine, M., & Weintraub, H. (1980). Tissue-specific DNA cleavages in the globin chromatin domain introduced by DNAase I. *Cell*, *20*(2), 451–460. https://doi.org/10.1016/0092-8674(80)90631-5
46. Tarbell, E. D., & Liu, T. (2019). HMMRATAC: A Hidden Markov ModeleR for ATAC-seq. *Nucleic Acids Research*, *47*(16), e91. https://doi.org/10.1093/nar/gkz533
47. Vierstra, J., Lazar, J., Sandstrom, R., Halow, J., Lee, K., Bates, D., Diegel, M., Dunn, D., Neri, F., Haugen, E., Rynes, E., Reynolds, A., Nelson, J., Johnson, A., Frerker, M., Buckley, M., Kaul, R., Meuleman, W., & Stamatoyannopoulos, J. A. (2020). Global reference mapping of human transcription factor footprints. *Nature*, *583*(7818), Article 7818. https://doi.org/10.1038/s41586-020-2528-x
48. Voss, T. C., & Hager, G. L. (2014). Dynamic regulation of transcriptional states by chromatin and transcription factors. *Nature Reviews Genetics*, *15*(2), Article 2. https://doi.org/10.1038/nrg3623
49. Wang, M., Tai, C., E, W., & Wei, L. (2018). DeFine: Deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants. *Nucleic Acids Research*, *46*(11), e69. https://doi.org/10.1093/nar/gky215
50. Wingender, E., Schoeps, T., Haubrock, M., Krull, M., & Dönitz, J. (2018). TFClass: Expanding the classification of human transcription factors to their mammalian orthologs. *Nucleic Acids Research*, *46*(D1), D343–D347. https://doi.org/10.1093/nar/gkx987
51. Wolter, F., & Puchta, H. (2018). Application of CRISPR/Cas to Understand Cis- and Trans-Regulatory Elements in Plants. In N. Yamaguchi (Ed.), *Plant Transcription Factors: Methods and Protocols* (pp. 23–40). Springer. https://doi.org/10.1007/978-1-4939-8657-6_2
52. Yan, F., Powell, D. R., Curtis, D. J., & Wong, N. C. (2020). From reads to insight: A hitchhiker's guide to ATAC-seq data analysis. *Genome Biology*, *21*(1), 22. https://doi.org/10.1186/s13059-020-1929-3
53. Yang, T., & Henao, R. (2022). *TAMC: A deep-learning approach to predict motif-centric transcriptional factor binding activity based on ATAC-seq profile* (p. 2022.02.15.480482). bioRxiv. https://doi.org/10.1101/2022.02.15.480482
54. Yevshin, I., Sharipov, R., Valeev, T., Kel, A., & Kolpakov, F. (2017). GTRD: A database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Research*, *45*(D1), D61–D67. https://doi.org/10.1093/nar/gkw951
55. Zhao, Y., Zheng, D., & Cvekl, A. (2019). Profiling of chromatin accessibility and identification of general cis-regulatory mechanisms that control two ocular lens differentiation pathways. *Epigenetics & Chromatin*, *12*(1), 27. https://doi.org/10.1186/s13072-019-0272-y