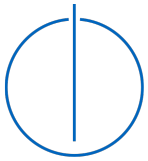
DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Machine Learning for Regulatory Genomics (ML4RG) Term Project

Authors:	Christos Georgakilas, Sinan Ergezer, Ahmet Yigit Dogan, Ugur Dura
Supervisor:	Prof. Julien Gagneur
Advisor:	Monika Heinzl, Vincent Loubiere
Submission Date:	17.07.2023



UNDERSTAND HOW PROMOTERS DNA SEQUENCE DIRECT TRANSCRIPTION THROUGH DEEP LEARNING ANALYSIS

Christos Geogakilas[1], *Sinan Ergezer*[1], *Ahmet Yigit Dogan*[1], *Ugur Dura*[1], *Monica Heinzl*[2], *Vincent Loubiere*[2]

[1]Technical University of Munich (TUM), Munich, Germany
[2]Research Institute of Molecular Pathology (IMP), Vienna, Austria

ABSTRACT

Understanding the complex regulatory mechanisms of gene expression is crucial in molecular biology. Promoter regions, especially core promoters, play a vital role in transcriptional regulation. Recent studies have revealed the diversity and complexity of core promoters, with various functional motifs contributing to their unique characteristics. Transcriptional cofactors (COFs) mediate regulatory signals between enhancer regions and core promoters, influencing gene expression. Leveraging deep learning, we developed a novel model to predict COF preferences from DNA sequences, aiming to uncover the underlying DNA motifs that shape COF-promoter specificities. The results of the models demonstrated their effectiveness in predicting TSSs (Transcription Start Site) and providing insights into COF-promoter interactions. Deep learning methods proved valuable in deciphering the complex interplay between DNA sequences and transcriptional regulation. This research contributes to our understanding of the relationship between DNA motifs and gene expression activity.

Index Terms— Promoters, Cofactors, Deep Learning, Promoter Motifs

1. INTRODUCTION

Deciphering the intricate regulatory mechanisms governing gene expression is a fundamental pursuit in molecular biology[1]. Central to this endeavor are promoter regions, which play a crucial role in orchestrating the precise activation or repression of genes. Core promoters, typically located upstream (60-120 base pairs) of the TSS, serve as essential platforms for the assembly of transcription factors and other auxiliary factors (e.g. cofactors), enabling the initiation of transcription and subsequent regulation of gene expression[2]. Unraveling the structure-function relationships of Human promoters is of particular significance due to their inherent complexity and diversity. The complexity may be attributed to the necessity of controlling the expression of thousands of protein-coding genes using a relatively small set of transcription factors[3]. The complexity and diversity of human promoters are particularly striking, possibly due to the need to control the expression of thousands of protein-coding genes with a limited number of transcription factors[4]. Moreover, the rapid evolution of primate promoters suggests weak selective constraints and further contributes to the intricate and diverse nature of human promoter structures[5]. Understanding the intricacies of human promoters is not only crucial for deciphering gene regulatory networks but also holds implications for studying genetic variants associated with rare Mendelian diseases and somatic mutations in cancer.

Previously, the core promoter was thought to be a universal component, functioning in a similar manner across all protein-coding genes[6]. However, there is now a growing belief that core promoters consist of multiple short DNA elements and motifs, typically ranging from 5 to 15 bases in length[7]. This diversity in the composition of core promoter regions leads to variations in their structure and functionality, making each core promoter unique in its characteristics[4]. Decades of research have identified various functional sequence motifs within the core promoter that contribute to its structure-function relationship[7]. The most well-known and extensively studied motif is the TATA box, originally believed to be universally present in RNA pol II core promoters[8]. However, with the advent of genome-wide TSS detection techniques based on high-throughput sequencing, it has become apparent that the structure of the core promoter is highly diverse and complex, lacking universal core promoter elements[9]. Only around 17% of eukaryotic core promoters are estimated to contain the TATA box, highlighting the tremendous heterogeneity within this regulatory region[10].

In the dynamic regulation of gene expression, COFs play a pivotal role, mediating the transmission of regulatory signals from enhancer regions to core promoters[11]. These COFs act as crucial intermediaries, influencing transcriptional activation and gene expression. In a study by Vanja Haberle et al. in 2019, the authors demonstrated differential activation of core promoters by different COFs, indicating distinct regulatory preferences or compatibility between COFs and specific types of core promoters[12]. This interplay between COFs and core promoters shapes the transcriptional landscape, facilitating the specific activation of target genes, alternative promoter usage, and the selection of distinct TSS[12]. Importantly, these findings establish the existence of distinct COF-core promoter compatibilities not only in *Drosophila* cells but also in Human cells, suggesting the potential conservation of these regulatory principles across species[12]. By unraveling the relationship and specificity between COFs and core promoters, these studies shed light on the common dependencies and specificities shared between COFs and core promoters in driving gene expression.

In recent years, sequence-based approaches leveraging deep learning have shown promise in promoter prediction. Deep convolutional neural networks (CNNs) have demonstrated exceptional performance in various fields, including omics data[2][13][14]. Deep learning models have been successfully applied in biological problems, such as branch point selection, DNA sequence quantification etc. [2]. For instance, In 2017, Umarov and Solovyev published CNNProm for promoter recognition, achieving high accuracy in discriminating short promoter sequences[15]. Then, in 2019, Umarov et al., improved the CNNProm and published CNNProm2 that works well with longer promoter sequences and predicts the

TSS positions also[16]. Besides that, in 2022, Almedida et al. published a deep-learning model called DeepSTARR to predict enhancer activity directly from the DNA sequence[17]. DeepSTARR managed to learn relevant transcription factor motifs and higher-order syntax rules, including the impact of motif-flanking sequence and inter-motif distances on enhancer activity[17]. These advancements highlight the power of deep learning methods in deciphering the intricate relationship between DNA motifs and gene expression activity.

Although Haberle et al. hinted at specific preferences between COF and promoters, how these specifities are encoded at the DNA level remains unclear[12]. Here, we devleoped a novel deep Learning model to predict these preferences directly from promoters' DNA sequence, hoping to identify the DNA motifs that shape COF/Promoter specificities.

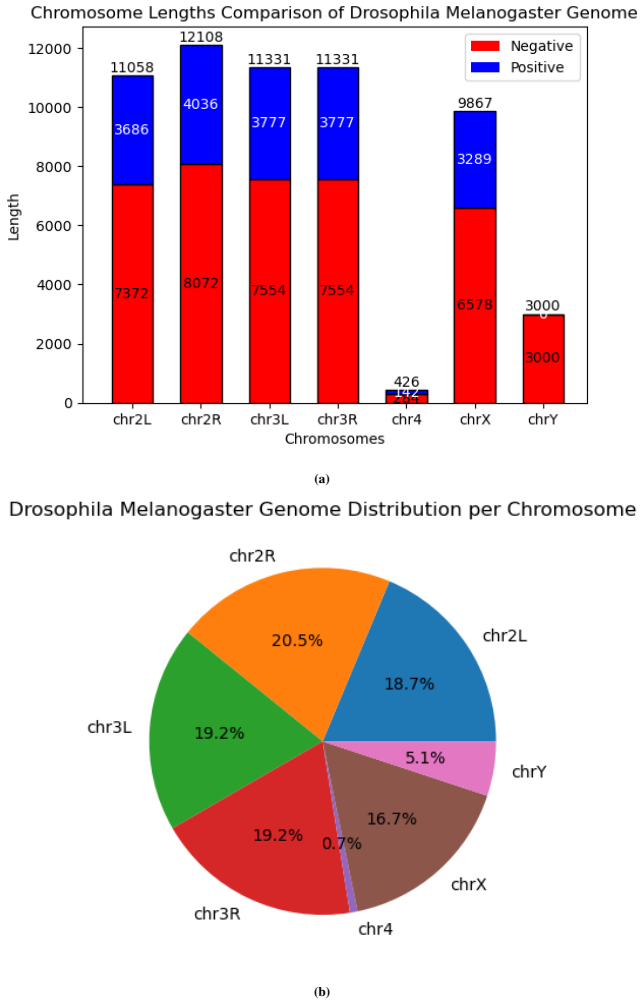


Fig. 1: Subfigure (a) shows negative (red) and positive (blue) length distribution of each chromosome of the Fruit fly. On top of each bar the total number of each entry indicated. Subfigure (b) shows the percentage distribution of each chromosome of the fruit fly genome. Both graph is plotted based on provided .bed file entries for TSS prediction.

2. MATERIALS AND METHOD

The construction of our model involved two primary components. First, we aimed to demonstrate the model's ability to accurately predict TSS along the *Drosophila* genomes, using only DNA sequence. Subsequently, we developed the core model, which predicts promoters' transcriptional response to COF binding, thus unveiling the relationship between promoter motifs COFs.

As mentioned earlier, IMP shared their experimental results with our team. They conducted a COF-STAP-Seq based experiment where each cofactor of interest was forcibly recruited next to a collection of Core-Promoters (CPs). It's important to clarify that each cofactor was tested separately in individual assays. To do this, they introduced two DNA pieces (plasmids) into the cells: the first plasmid pool contained the core-promoters flanked with an "UAS" recruiting sequence located just upstream, and the second plasmid produced the cofactor of interest fused to a GAL4-DNA binding domain. Upon transfection of these two plasmids, the cofactor of interest was produced and recruited next to different core-promoters, potentially triggering their transcription. By measuring the transcription of the promoters in the presence of different cofactors, we were able to assess their responsiveness. Each entry in the dataset was associated with cofactor expression, which included 15 different cofactors, including a control cofactor (Green Fluorescent Protein - GFP). The cofactors listed were p65, Nej_p300, Med25, Lpt, Med15, CG7154_Brd9, fs1h_Brd4, trr, trx, gfzf, Chro, CG15356_EMSY, Brd8, Mof, and the control GFP.

2.1. Data Preprocessing

At the outset, our analysis commenced by utilizing the .bed file provided by IMP. This file comprises 30,673 annotations for DNA sequences containing TSSs within the genome of the fruit fly, *Drosophila melanogaster* which were downloaded from the database Flybase [1]. The reference genome employed is Genome assembly Release 6 plus ISO1 MT [2] with the NCBI RefSeq ID: GCF_000001215.4.

To extract the corresponding DNA sequences, we parsed the dm6 Fasta file based on the TSS starting coordinates provided in the .bed file. By extending the sequences by +/- 124 bp from the TSS position, we obtained sequence entries of 249 bp in length. Each entry was assigned a unique ID and linked to its corresponding chromosome location and coordinates. Furthermore, utilizing the dm6 Fasta file, we obtained negative control sequences devoid of any TSS. This process involved dividing the entire *Drosophila melanogaster* genome into 249 bp bins. After eliminating any overlapping regions among the extended TSSs, negative control sequences were randomly sampled. Consequently, we amassed a total of 61,731 sample set for the TSS binary prediction model.

Regarding our second task's data processing, we received experimental results from IMP in the form of a CSV file. This file contained 72k *Drosophila* promoters, in response to the recruitment of 14 different COFs associated with TSSs of the *Drosophila melanogaster* genome. For this study, the CSV file was constructed based on the Apr. 2006 assembly of the *Drosophila melanogaster* genome (dm3, BDGP Release 5), which is the fruit fly genome release in 2014.[3]

[1] https://flybase.org
[2] *Drosophila melanogaster* Aug. 2014, BDGP Release 6 + ISO1 MT/dm6
[3] https://hgdownload.soe.ucsc.edu/goldenPath/dm3/bigZips/

Similar to our previous approach, we extracted the relevant sequences by parsing the dm3 Fasta file using the provided TSS starting coordinates in the CSV file. Since the CSV file contained both the starting and end sites, we parsed the fasta file using these exact coordinates, resulting in sequences that were 134 bp in length. Each entry was assigned a unique ID and associated with its respective chromosome location and coordinates.

$$Pseudocounts_K = \forall_K \in \{1, ..., n\} .b_K = min(b_K > 0) \quad (1)$$

$$Logcounts_J = \forall_J \in \{1, ..., n\} .a_j = log_2(a_j) \quad (2)$$

$$NormCounts_J = \forall_J \in \{1, ..., n\} .a_j = log_2(a_j) - log_2(GFP_j) \quad (3)$$

Importantly, CPs have a basal activity in the absence of functional COFs that need to be normalized out to infer COF/promoter preferences. To do so, we used transcriptions levels upon recruitment of the GFP protein, which has no impact on trasncription and will be used as a control. For each COF, normalized CPs activity was computed as $log2((COFcounts+pseudocount)/(GFPcounts+pseudocount))$ (eq. 2,eq. 3 and eq. 1). Of note, pseudocount were added to the data to avoid 0 ($pseudocount = min(counts > 0))(eq.1)$.This approach ensured accurate determination of COF / promoter preferences while effectively addressing issues related to zero values and expression noise.

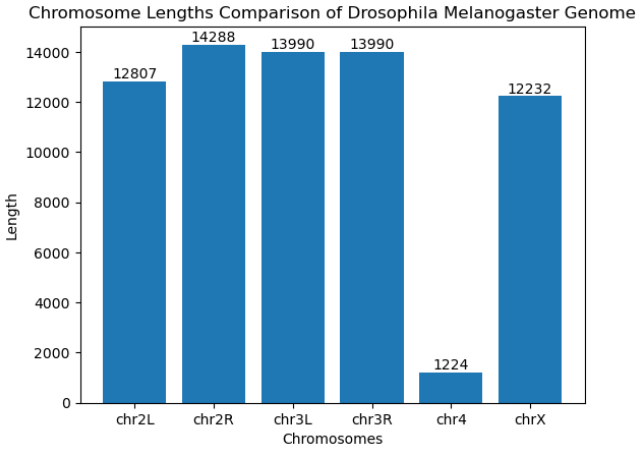


Fig. 2: Comparison of Chromosome Lengths in *Drosophila melanogaster* Genome. The bar plot displays the lengths of chromosomes based on the analysis of the given cofactor expression data. Each bar represents the length of a chromosome, and the corresponding values are indicated on top of each bar. The plot provides insights into the relative lengths of chromosomes.

At the end of the data processing, two methods were utilized to prepare the data: one-hot encoding and data loader preparation. First, parsed data was loaded and then converted to NumPy arrays.The data was divided into training, validation, and testing sets based on the row indices. 80% Training, 10% Validation and 10% Test ratios are selected. One-hot encoding was performed to convert the DNA sequence letters ("A," "C," "G," and "T") into categorical vectors. The codetable was created to map each letter to a corresponding index. Finally, the data and labels were converted to PyTorch tensors and sent to the specified device for computation. The data was organized into TensorDatasets and DataLoader objects for efficient batching during model training.

Similarly, for the cofactor model, the process involved loading the data, selecting the rows with the chromosome identifiers, shuffling the data, and dividing it into training and validation/testing sets(80% Training, 10% Validation and 10% Test ratios). However, in this case, a different subset of columns was chosen. One-hot encoding was again applied to convert the DNA sequence letters into categorical vectors using the codetable. The labels were converted to tensors of type float32, and the data and labels were sent to the specified device. Lastly, the data was organized into TensorDatasets and DataLoader objects for training, validation, and testing.

2.2. Data Analysis

In the .bed file, only canonical *Drosophila* chromosomes were included (2L, 2R, 3L, 3R, 4, X) containing 30,674 TSS sequences, corresponding to 19,578 unique genomic locations, as distinct gene isoforms can share a similar TSS. After cleaning the file from some duplicated sequences we ended up with 19578 sequences with a present TSS. On the other hand, while we were creating control sequences(negative TSS/ no TSS), since we have limited amount of data, we decided to sample twice the number of corresponding chromosome entries. In the end, we resulted with the distribution that is shown in fig. 1. Each chromosome consisted of 2/3 sequences without TSS and 1/3 sequences with TSS. In conclusion, we had 61731 entries for our data set that would be split for training, validating and testing the model. When the chart is examined on fig. 1, the number of entries for each chromosome distribution is balanced (except chromosome 4 and Y).

On the other hand, while creating the cofactor expression data set, IMP provided us with an xlsx file that contained 72000 entries. Parsing the data and extracting the important information resulted in 71592 entries. The entry distribution for each chromosome can be seen at fig. 2. When we normalize the data, the distribution of the values can be found in appendix. When we examined the mean likelihood expression of each cofactor throughout the data set as shown in fig. 10, we can clearly see that GFP has the lowest likelihood to be expressed. In addition to that, we can clearly see that the data set records bigger expression for some of the cofactors such as gfzf and trx. That could potentially cause the model to learn gfzf related motifs better and non-MOF related motifs better.

2.3. Model Construction

2.3.1. TSS Binary Predictor Model Construction

As previously mentioned, in the first stage of our work we implemented a binary prediction deep neural network architecture that, granted a DNA sequence of size 249, predicts whether the sequence contains a TSS or not. Thus, our model has to map the 249-bp-long sequence to one value that corresponds to the binary prediction for the TSS.

We based the implementation of our architecture on the state-of-the-art DeepSTARR model as it seemed to be fitting our needs for making accurate predictions based only on DNA sequences[17]. Additionally, the original DeepSTARR implementation is on Tensorflow, but we decided to implement it ourselves in PyTorch. The most important part of the network that we changed was the output,

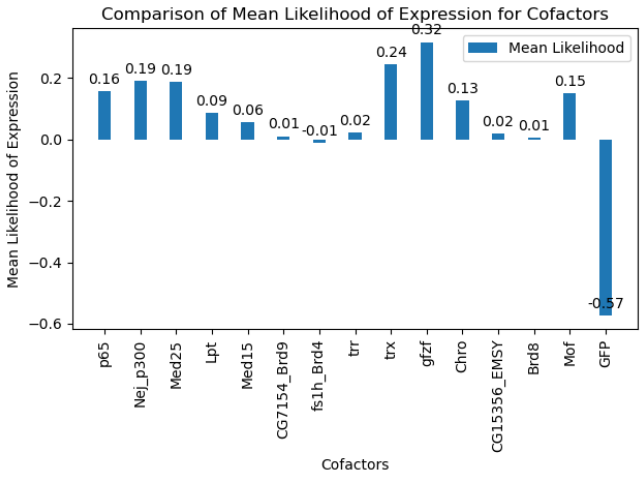


Fig. 3: Mean Likelihood of Expression for Cofactors. The plot illustrates the average likelihood of expression for each cofactor based on the dataset. Each bar represents the mean likelihood of expression, with a higher bar indicating a higher average likelihood. Data labels on top of each bar display the exact mean likelihood value. The plot provides insights into the relative likelihoods of expression for different cofactors, contributing to the understanding of their potential regulatory roles.

as DeepSTARR was made for a regression task whilst we used it for a binary prediction. The details about our architecture are given below:

Our architecture is consisted of 4 initial convolutional layers, each followed by a batch normalization layer, a ReLU non-linearity and a max-pooling layer. For all but the first one we also implement a dropout layer after the corresponding max-pool. The convolutional layers are used as feature extraction mechanisms as they manage to identify underlying features of the sequence data. More specifically, the first convolutional layers can extract local sequence features such as TF motifs and then later ones can find even more complex features like TF motif syntax. The sizes of the kernels and filters for the convolutional and max-pooling layers are the same as those in DeepSTARR, since it served as a great baseline for our experiments and it also intuitively made sense as our input sequences were of the same type and size. Adding some dropout layers as previously mentioned, helped with the regularization of our network in order to avoid very quick overfitting, especially since we didn't have a lot of data sequences(only 19,578 sequences that contains TSS).

Following the convolutional layers mentioned above and after having extracted the important features from our data, we have the classifier network part of our model. We flatten our feature tensor that has been generated from the convolutional layers before inputting it to the classifier. The classifier is basically made of 2 linear layers, each followed by a batch normalization layer, a ReLU non-linearity and a dropout and in the end a final linear layer with 1 output that gives the logit which will be used for the binary prediction. This logit is then passed through a sigmoid function which gives a probability value in the range of [0,1]. If this value is over 0.5 we predict that a TSS is present in the sequence, if it's under 0.5 we predict TSS absence. The linear layers manage to combine the features and patterns extracted by the previous convolutional layers in order to make accurate predictions.

To train our model we use the Adam optimizer and Binary Cross Entropy loss, as we have a binary prediction task. The sizes of the 2 linear layers and the probability of dropout for all dropout layers were tested as hyperparameters for the model and out of the tests the better performing ones were chosen. The final architecture we end up with isn't particularly heavy or complex and performs well on the task of binary prediction of TSS.

2.3.2. Cofactor-Promoter Motif Model Construction

In the second stage of our research, we developed a deep neural network architecture with the objective of predicting the expression values of various COFs given a DNA sequence of length 134. Our dataset initially comprised expression data for 14 COFs (excluding GFP) associated with each sequence. However, considering the insights from Haberle et al., who demonstrated that promoter sequences exhibit clustering behavior based on the occurrence of known Cofactor Protein motifs, we opted to focus on one COF per cluster[12]. This decision was motivated by the practicality and interpretability of the results. Notably, the clustering analysis revealed that the promoter sequences fall into five distinct groups, each characterized by a different motif occurrence profile. For instance, Group 1 displayed a strong enrichment of the TATA box motif. As a result, our model was designed to predict the expression levels of five specific COFs: p65, p300, gfzf, chro, and mof. These chosen COFs represent one COF from each cluster. Therefore, our model effectively maps the 134-base pair sequence to five expression values corresponding to the selected COFs. This approach allows us to gain insights into the specific COFs' involvement in the transcriptional response of promoters upon COF binding.

Our implementation was again based on DeepSTARR as its convolutional approach fitted well with our objective of predicting the promoter activity based purely on DNA sequences. Again, we implemented our entire architecture in PyTorch. This architecture was more easily adjusted as our task was a regression of 5 values for each sequence, very similar to DeepSTARR's regression of 2 values (housekeeping and developmental enhancer activity)[12].

The architecture used is exactly the same as the one described above for the first task with the difference that instead of having one final linear layer with output of size 1, we have five final linear layers with output of size 1 as we now have a regression task of 5 values (expressions of 5 COFs) and not a binary classification. Additionally, we get rid of the Sigmoid layer as we don't want to apply any activation function on the outputs of our network. We use five output linear layers of output size 1 instead of one output linear layer of output size 5 to get the COF expression values as it gives the same results and it better suits the interpretability methods we will use for our model.

For this task it could possibly be beneficial to use different sizes of kernels and filters for the convolutional and max-pooling layers compared to the initial DeepSTARR since our input sequence is now of size 134 instead of 249 bps but this fine- tuning was out of the scope of the current project so we decided to move forward with the same values as DeepSTARR.

The model construction and training process utilized specific hyperparameters that were carefully selected to optimize performance. The batch size, which determines the number of examples processed in each iteration, was set to 128. The model was trained for 15 epochs, representing the number of complete passes through the

training dataset. A learning rate of 0.002 was employed, controlling the step size at which the model learns from the data. The convolutional layers employed different numbers of filters, with 256, 60, 60, and 120 filters in the respective layers. Additionally, the sizes of the filters were set to 7, 3, 5, and 3 for the corresponding layers. The model consisted of a single dense layer, with 256 neurons in each layer. To prevent overfitting, a dropout probability of 0.3 was applied. These hyperparameters were chosen through experimentation and tuning to optimize the model's performance in accurately predicting the desired outcomes.

To train our model we use the Adam optimizer and Mean Squared Error loss, as we now have a regression task. The loss function which is back-propagated through the network is calculated as the mean of all of the losses between the target values and the predicted values for all 5 COFs.

2.4. Model Interpretation

In the last part of the project, we wanted to interpret the models, in order to uncover promoter motifs needed for the prediction of the promoters' transcriptional response to the different COFs. Thus, We decided to unveil the promoter-COFs specifities by the underlying sequence patterns as this model is the one that could give us the most interesting and novel biological knowledge. The general idea of our approach and goal is: ***a)*** Gather the sequences that have high expression scores for a COF, ***b)*** Using an interpretation method, define at nucleotide level which parts of the input sequences are the most important for the network when predicting the final output value of the expression for that COF, ***c)***find motif occurrences with the subset of sequences and extract nucleotide contribution scores for these motif occurrences, ***d)*** identify which promoter motifs are present when each COF is able to strongly induce transcription. Among other very important biological knowledge, with these steps we can even get results about promoter motif and COF specificity in a fully computational approach.

The details about our interpretation method are described below: We are doing the model interpretation after training it on the regression task for the 5 COFs(p65, p300, gfzf, chro, and mof). We are using sequences from the validation set in order to do the interpretation.

For simplicity, we explain the method steps for one of the COFs in detail, but we apply the same method for all 5 COFs.

- Pick all the different sequences from the validation set that have a corresponding expression value of the COF which exceeds a threshold (threshold = 1)
- Feed these sequences to the trained model, getting the corresponding predictions for the COF expression
- Use the IntegratedGradients algorithm to calculate how each nucleotide in the input sequence contributes to the output value of the COF expression
- Load an R object provided to us by the IMP that contains 19 PWMs with known promoter motifs
- Find the location of these known motifs(TATA box, Ohler1...) in our sequence and compare with the calculated nucleotide contribution scores for the same location in the sequence

2.4.1. Pearson correlation coefficient (PCC)

In order to assess our model's prediction quality for the cofactor expressions, one of the main metrics examined during the study was the Pearson correlation coefficient (PCC). The main reason behind choosing PCC among the different types of correlation coefficients was that it provides a sense of linear correlation between two data sets. The correlation coefficient can be derived for two sets of sequential data (x and y) as follows shows in eq. 4.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{4}$$

The coefficient, r_{xy}, takes a value between -1 and 1, depending on the type of correlation. Negative values indicate an inverse proportionality, and vice versa, where the magnitude of the correlation gives a clue about the significance of the correlation. According to a rule of thumb scale, (*Table 1.*) provided by Hinkle et al.[18], the obtained magnitudes can be interpreted based on 5 levels of correlation between negligible correlation and very high correlation.

Size of Correlation	Interpretation
.90 to 1.00 (-.90 to -1.00)	Very high positive (negative) correlation
.70 to .90 (-.70 to -.90)	High positive (negative) correlation
.50 to .70 (-.50 to -.70)	Moderate positive (negative) correlation
.30 to .50 (-.30 to -.50)	Low positive (negative) correlation
.00 to .30 (.00 to -.30)	negligible correlation

Table 1: Pearson Correlation Coefficient significance scale suggested by Hinkle et al.[18].

3. RESULTS

3.1. Binary Prediction of TSSs

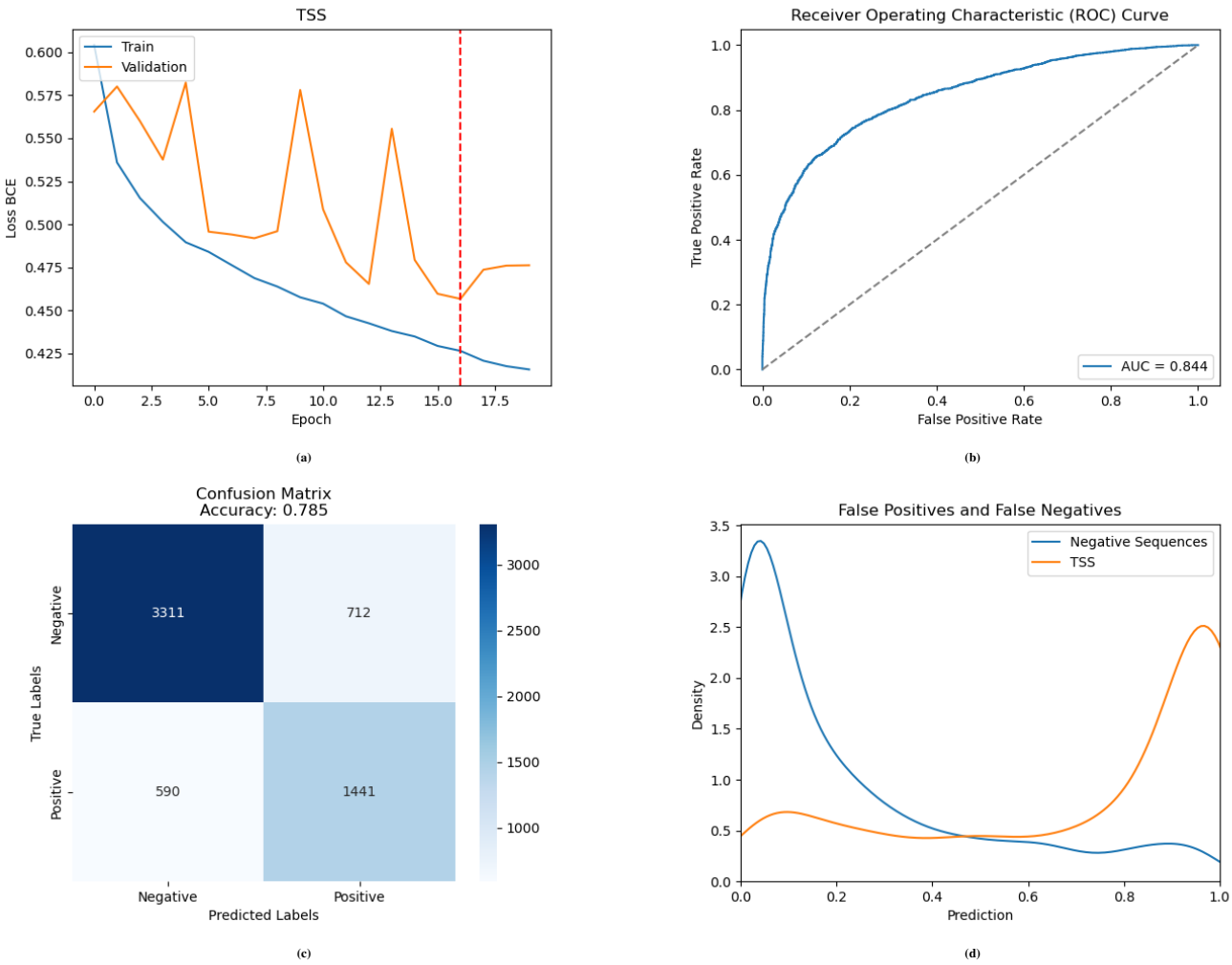


Fig. 4: Model Evaluation and Training Performance. This figure presents multiple subplots assessing the performance and training progress of the model. (a) The upper-left subplot depicts the training and validation loss curves over epochs, providing insights into the convergence of the model. (b) The upper-right subplot showcases the Receiver Operating Characteristic (ROC) curve, illustrating the trade-off between true positive rate and false positive rate and quantifying the model's discriminatory power. (c) The lower-left subplot displays the Confusion Matrix, highlighting the classification accuracy and the distribution of true positive, true negative, false positive, and false negative predictions. (d) The lower-right subplot exhibits the density plot comparing the distribution of false positives and false negatives. The combined figure offers a comprehensive analysis of model performance, training progression, and diagnostic metrics, contributing to the understanding and assessment of the developed model.

Examining fig. 4, the performance evaluation of our developed model yielded promising results, as shown in Figure 4. Figure 4a depicting the training and validation loss curves demonstrated convergence, with a validation loss of 0.475 and a training loss of 0.425. Early stopping suggested 15 epochs as the optimal point, indicating that the model neither overfits nor underfits the data. Figure 4b displayed the ROC curve, with an AUC of 0.844. This value indicates a reasonably good discriminatory power of the model in distinguishing between true positive and false positive predictions. Moving to Figure 4c, the confusion matrix showcased an overall accuracy of 0.785. Specifically, it revealed 3311 true negatives, 712 false positives, 590 false negatives, and 1441 true positives, providing insight into the model's ability to correctly identify negative and positive instances. Lastly, Figure 4d presented the density plot comparing the distribution of false positives and false negatives. The plot revealed that the model successfully predicted both negative sequences and positive sequences. However, there was a slight tendency to predict non-TSS sequences (false positives) more frequently than sequences with TSS (false negatives). Collectively, these results demonstrate the effectiveness of our model in predicting the presence of TSSs and provide valuable insights into its performance and predictive tendencies.

3.2. Cofactor-Promoter Motif Specificity Prediction

[]

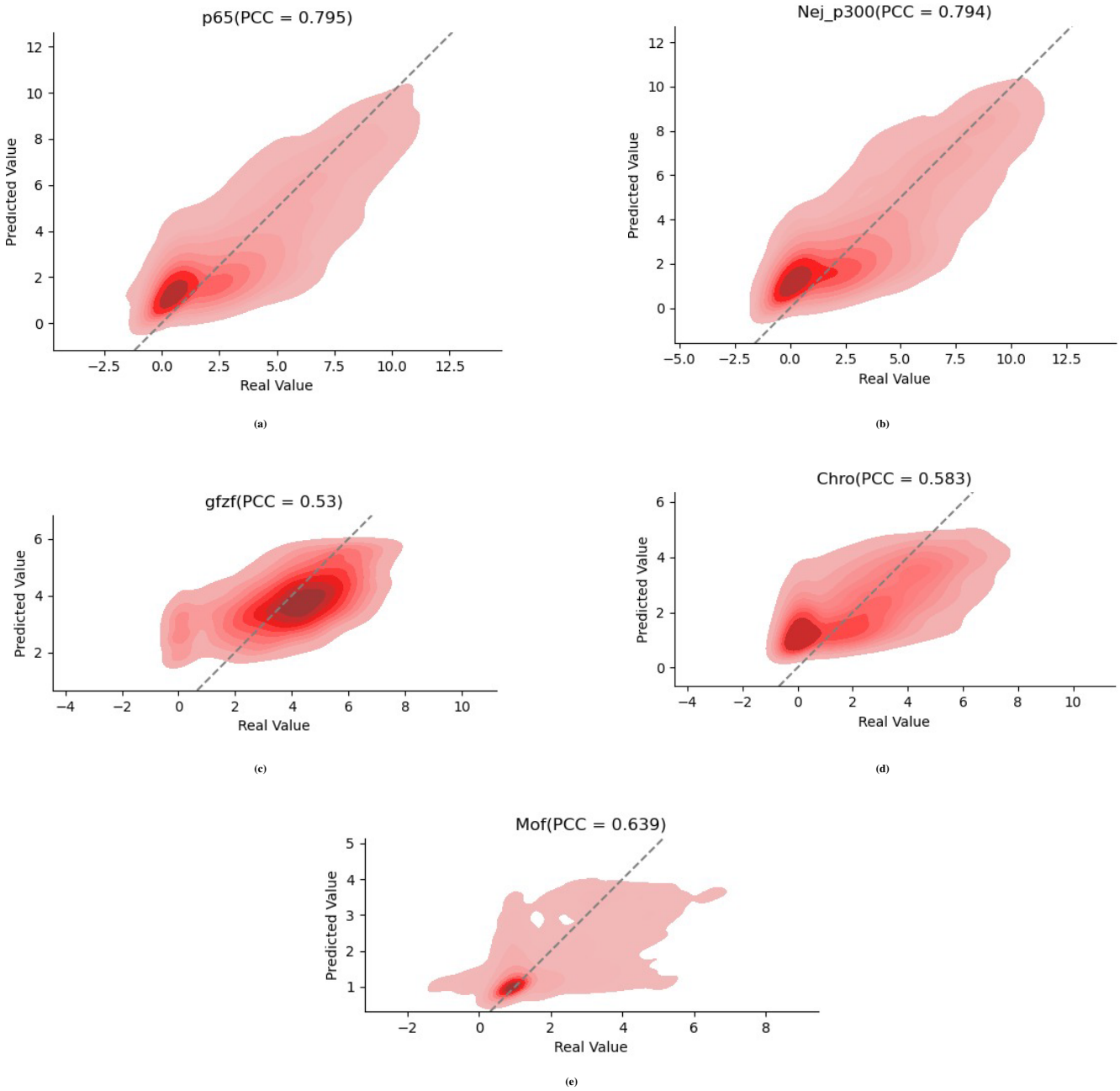


Fig. 5: Real expression values vs. predicted expression values for (a) p65, (b) Nej_p300, (c) gfzf, (d) Chro, and (e) Mof.

The 5 plots included in *Figure 5.* provide a good basis to compare the predicted expression values by the model to the ones in the test set. These smooth scatter plots are obtained by converting basic scatter plots of predicted and real values to more interpretable, heatmap-like representations, where the density of observations is indicated by tones of red (a deep red region includes a high number of observations, and lighter red parts include a smaller number of observations). They are also used due to the excessive number of samples (7144 observations) to avoid stacked observations. In the best scenario, one would expect the predictions to be same as the real values, thus, drawing a simple $x = y$ line (dashed lines) is a good way to check the tendencies of model for each cofactor. The overall performance of the model can be checked by interpreting the deviations and spreading behavior of red regions, the deepest ones being in the first place, from the target ($x = y$ in this case).

3.3. Biological Interpretation of Cofactor-Promoter Motif Specificity

[]

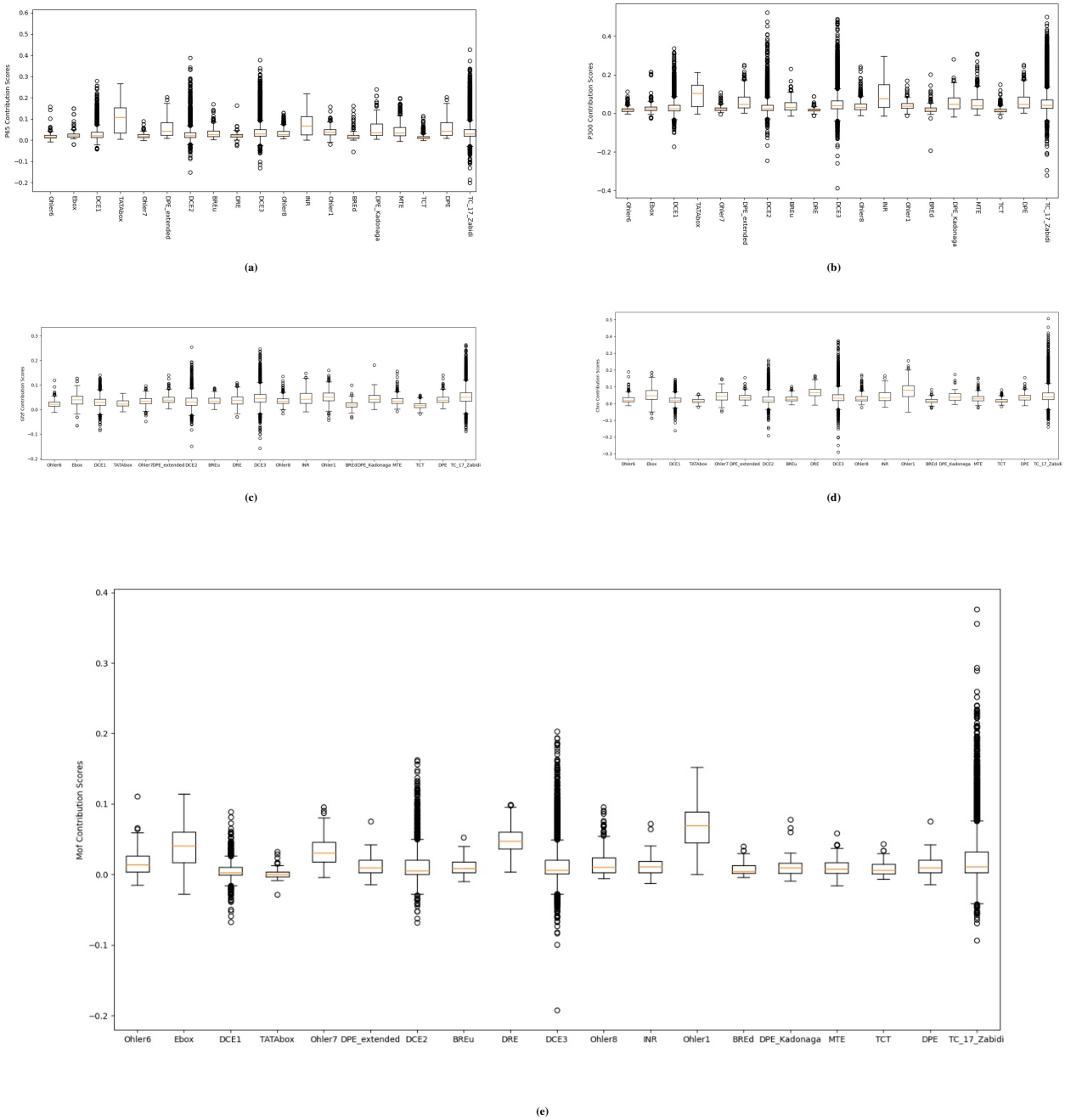


Fig. 6: Contribution scores of promoter Motifs found in the input test sequences. These values are extracted with an interpretation algorithm for the test sequences from trained model weights for (a) p65, (b) Nej_p300, (c) gfzf, (d) Chro, and (e) Mof.

4. DISCUSSION

According to Figure 4, the binary TSS prediction model demonstrates the ability to distinguish DNA sequences based on the presence or absence of TSS. This is supported by the reliable performance indicated by the confusion matrix (Figure 4c). However, upon closer examination of the confusion matrix (Figure 4c) and the density distribution (Figure 4d), a noticeable trend emerges. The model exhibits a higher proficiency in predicting non-TSS sequences compared to sequences containing TSS. This bias may be attributed to the uneven distribution of data and limitations faced during training. Additionally, the density plot reveals instances where the model's predictions display a degree of uncertainty, as evidenced by the prediction certainty between 0.2 and 0.8. To address these limitations, several suggestions can be considered. Pretrained language models have proven their effectiveness in various biological applications. Reddy et al. (2023) have reviewed strategies for predicting promoter-driven gene expression using language models[19]. For example, the state-of-the-art TSSNote-CyaPromBERT, trained on the human genome, could be utilized for transfer learning to initialize our model weights and improve predictions[20]. Further experiments can be hold to compare the models.

On the other hand,upon analyzing the results of the cofactor prediction model and mapping the obtained coefficients to corresponding intervals, it becomes evident that the model exhibits predictive capability for the cofactors p65 and Nej_p300, displaying a high degree of correlation with the target values. Furthermore, the model demonstrates the ability to provide outputs for each cofactor of interest, albeit to a moderate extent of linear relationship.

To double check the significance of the linear relation between target expression values and predicted expression values, a convenient further check would be their scatter plots. The plots in *Figure 5.* depict to what extent the predicted values fit the real ones.

Cofactor	PCC
p65	0.795
Nej_p300	0.794
gfzf	0.530
Chro	0.583
Mof	0.639

Table 2: Pearson Correlation Coefficient of predicted and real expression values of cofactors of interest.

The highly linear relationship for Nej_p300 and p65 is quite obvious in the plots. The slight misfit to the regression line in the plots of gfzf and Chro can also be seen, when compared to Nej_p300 and p65, even a tiny non-linearity for gfzf can be in question. When it comes to Mof, the regression line seems to be fitting perfectly to the most dense area of the observations, yet relatively a small number of predicted values prevent its correlation coefficient to be higher than 0.7 due to their high deviation from real values.

Observing the predicted interpretation box plots of our best trained model on the test set provided in Fig. 6for each of the 5 COFs and comparing them to the experimental results for the 5 COFs provided by Fig. 7 we observe that our model almost accurately predicts the found experimental results regarding cofactor-motif specificity. For COFs that our model achieves higher PCC values and thus better prediction ability we see that the specificity is even more apparent. We believe that this finding in a fully computational way is a very important contribution of our work as to our knowledge it hasn't been formulated before. For example, for the p65 COF we observe that the median of the contribution score values in the boxplot for the TATA Box and INR motif are the highest, something also observed in the experimental data. This suggests that this specific cofactor displays specificity for these two distinct types of core promoters. Also it indicates that it can possibly be used to predict the responsiveness of different COFs on yet untested sequences.

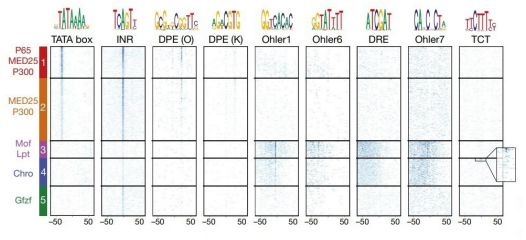


Fig. 7: The plot is obtained from Haberla et al. 2019's publication [12]. Different core promoter (CP) groups, which are preferentially activated by distinct transcriptional cofactors (COFs), exhibit diverse CP motifs. The presence of established CP motifs specific to Drosophila CPs is observed within five different groups based on their responsiveness to COFs in STAP-seq (Extended Data Fig. 6). Within each group, CPs are arranged in descending order of STAP-seq tag count for the most potent corresponding COFs (on the left). Additionally, the occurrence of TCT is highlighted in the top 10% of CPs belonging to group 4 (inset).

5. CONCLUSION

In conclusion, this research paper focuses on two primary components: the prediction of TSSs along *Drosophila* genomes using DNA sequence, and the prediction of promoters' transcriptional response to COF binding, revealing the relationship between promoter motifs and COFs. The paper describes the data preprocessing steps for both tasks, including the extraction of DNA sequences, the creation of negative control sequences, and the normalization of COF expression levels.

For the TSS binary prediction task, a deep neural network architecture based on DeepSTARR is implemented. The model consists of convolutional layers for feature extraction and linear layers for classification, using a sigmoid function to predict TSS presence or absence. The model is trained using the Adam optimizer and Binary Cross Entropy loss.

For the COF-promoter motif prediction task, a similar deep neural network architecture based on DeepSTARR is employed, but with five final linear layers for regression of the expression values of specific COFs. The model is trained using the Adam optimizer and Mean Squared Error loss.

To interpret the models and uncover promoter motifs related to COF binding, an interpretation method is applied. It involves selecting sequences with high expression scores for a COF, calculating nucleotide contribution scores using the IntegratedGradients algorithm, and comparing known promoter motifs with the calculated scores to identify motif occurrences and their contributions to COF expression.

The models' performance is evaluated using metrics such as Pearson correlation coefficient (PCC), training and validation loss curves, ROC curve, confusion matrix, and density plots. The TSS binary prediction model demonstrates the ability to distinguish sequences based on TSS presence or absence, although it shows a bias towards predicting non-TSS sequences. The COF-promoter motif prediction model exhibits a moderate linear relationship with the expression values of specific COFs, with higher correlations observed for p65 and Nej p300. Finally, the COF model seems to be able to learn important biological knowledge, which to our knowledge has been found previously only in experiments regarding COF-Promoter Motif specificity.[12]. This suggests that it can possibly be used to predict the responsiveness of different COFs on yet untested sequences.

Overall, this research paper presents a comprehensive approach to predict TSSs and investigate the relationship between promoter motifs and COFs. The developed models show promising predictive performance and provide valuable biological insights into transcriptional regulation in *Drosophila* genomes. Future work could explore transfer learning from pretrained language models and further experiments to improve the models' predictions. Additionally, the model's performance on predicting more COFs could be tested.

6. REFERENCES

[1] Bernd Pulverer, "Getting specific: Sequence-specific DNA-binding transcription factors," *Nat. Rev. Mol. Cell Biol.*, vol. 6, no. S1, pp. S12–S12, 2005.

[2] Mhaned Oubounyt, Zakaria Louadi, Hilal Tayara, and Kil To Chong, "DeePromoter: Robust promoter predictor using deep learning," *Front. Genet.*, vol. 10, pp. 286, 2019.

[3] Long Vo Ngoc, George A Kassavetis, and James T Kadonaga, "The RNA polymerase II core promoter in drosophila," *Genetics*, vol. 212, no. 1, pp. 13–24, 2019.

[4] Sarah N Mapelli, Sara Napoli, Giuseppina Pisignano, Ramon Garcia-Escudero, Giuseppina M Carbone, and Carlo V Catapano, "Deciphering the complexity of human non-coding promoter-proximal transcriptome," *Bioinformatics*, vol. 35, no. 15, pp. 2529–2534, 2019.

[5] Han Liang, Yeong-Shin Lin, and Wen-Hsiung Li, "Fast evolution of core promoters in primate genomes," *Mol. Biol. Evol.*, vol. 25, no. 6, pp. 1239–1244, 2008.

[6] Anna Sloutskin, Hila Shir-Shapira, Richard N Freiman, and Tamar Juven-Gershon, "The core promoter is a regulatory hub for developmental gene expression," *Front. Cell Dev. Biol.*, vol. 9, pp. 666508, 2021.

[7] Ananda L Roy and Dinah S Singer, "Core promoters in transcription: old problem, new insights," *Trends Biochem. Sci.*, vol. 40, no. 3, pp. 165–171, 2015.

[8] James D Watson, Tania A Baker, and Stephen P Bell, *Molecular biology of the gene*, Addison Wesley, London, England, 5 edition, 2003.

[9] Andreas Albersmeier, Katharina Pfeifer-Sancar, Christian Rückert, and Jörn Kalinowski, "Genome-wide determination of transcription start sites reveals new insights into promoter structures in the actinomycete corynebacterium glutamicum," *J. Biotechnol.*, vol. 257, pp. 99–109, 2017.

[10] Venkata Rajesh Yella and Manju Bansal, "DNA structural features of eukaryotic TATA-containing and TATA-less promoters," *FEBS Open Bio*, vol. 7, no. 3, pp. 324–334, 2017.

[11] Muhammad A Zabidi and Alexander Stark, "Regulatory enhancer-core-promoter communication via transcription factors and cofactors," *Trends Genet.*, vol. 32, no. 12, pp. 801–814, 2016.

[12] Vanja Haberle, Cosmas D Arnold, Michaela Pagani, Martina Rath, Katharina Schernhuber, and Alexander Stark, "Transcriptional cofactors display specificity for distinct types of core promoters," *Nature*, vol. 570, no. 7759, pp. 122–126, 2019.

[13] Zhong Zhuang, Xiaotong Shen, and Wei Pan, "A simple convolutional neural network for prediction of enhancer–promoter interactions with DNA sequence data," *Bioinformatics*, vol. 35, no. 17, pp. 2899–2906, 2019.

[14] Yoojoong Kim and Minhyeok Lee, "Deep learning approaches for lncRNA-mediated mechanisms: A comprehensive review of recent developments," *Int. J. Mol. Sci.*, vol. 24, no. 12, 2023.

[15] Ramzan Kh Umarov and Victor V Solovyev, "Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks," *PLoS One*, vol. 12, no. 2, pp. e0171410, 2017.

[16] Ramzan Umarov, Hiroyuki Kuwahara, Yu Li, Xin Gao, and Victor Solovyev, "Promoter analysis and prediction in the human genome using sequence-based deep learning models," *Bioinformatics*, vol. 35, no. 16, pp. 2730–2737, 2019.

[17] Bernardo P de Almeida, Franziska Reiter, Michaela Pagani, and Alexander Stark, "DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers," *Nat. Genet.*, vol. 54, no. 5, pp. 613–624, 2022.

[18] Dennis E Hinkle, William Wiersma, and Stephen G Jurs, *Workbook for Hinkle/Wiersma/jurs' applied statistics for the behavioral sciences, 5th*, Wadsworth Publishing, Belmont, CA, 5 edition, 2002.

[19] Aniketh Janardhan Reddy, Michael H Herschl, Sathvik Kolli, Amy X Lu, Xinyang Geng, Aviral Kumar, Patrick D Hsu, Sergey Levine, and Nilah M Ioannidis, "Pretraining strategies for effective promoter-driven gene expression prediction," *bioRxivorg*, 2023.

[20] Dung Hoang Anh Mai, Linh Thanh Nguyen, and Eun Yeol Lee, "TSSNote-CyaPromBERT: Development of an integrated platform for highly accurate promoter prediction and visualization of synechococcus sp. and synechocystis sp. through a state-of-the-art natural language processing model BERT," *Front. Genet.*, vol. 13, 2022.

Appendix

Source Code

Please find the source code on GitHub: GitHub Repository

Plots

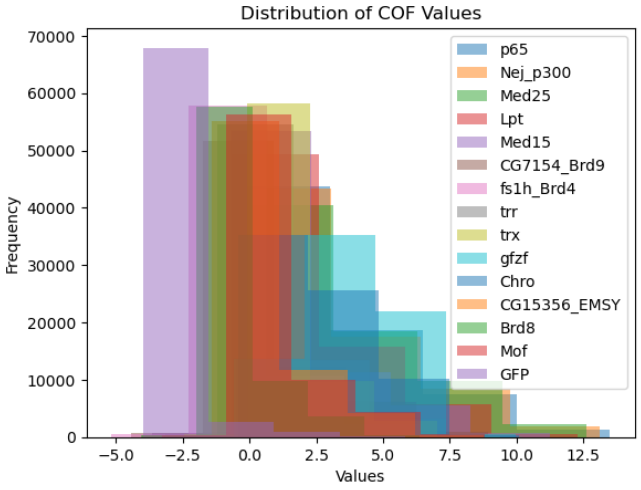


Fig. 8: Distribution of COF Values. The histogram plot showcases the distribution of values for each COF column based on the given cofactor expression data. Each histogram represents the frequency distribution of values, with the x-axis indicating the values and the y-axis representing the frequency of occurrence. The plot provides an overview of the distribution patterns and the range of values for each COF, contributing to the analysis of their expression levels.

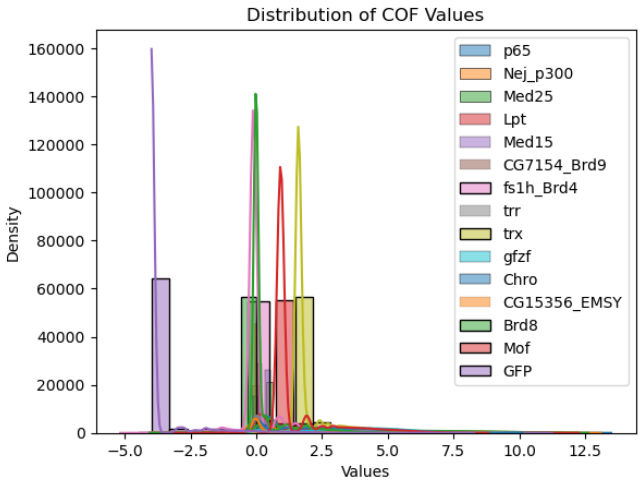


Fig. 9: Distribution of COF Values with KDE Scatter. The plot showcases the distribution of values for each COF column based on on the given cofactor expression data. Each COF's distribution is represented by a combination of histogram and KDE (Kernel Density Estimate) scatter plot. The histogram depicts the density of values along the x-axis, while the KDE scatter plot provides an estimate of the underlying probability density function. The plot offers insights into the distribution patterns and the relative density of values for each COF, facilitating the analysis of their expression levels and variability.

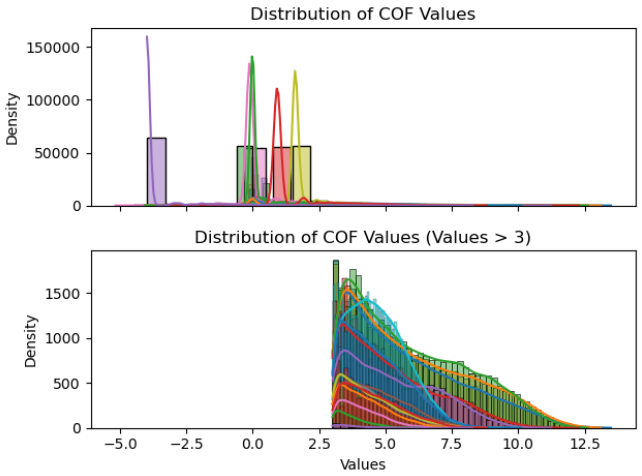


Fig. 10: Distribution of COF Values with Threshold. The plot presents the distribution of values for each COF column based on on the given cofactor expression data. The upper subplot displays the density of values using histograms and KDE (Kernel Density Estimate) scatter plots. The lower subplot highlights values exceeding the threshold of 3, providing a focused view of the distribution for those values. The shared x-axis allows for direct comparison between the overall distribution and the subset of values above the threshold. The plot aids in the assessment of the distribution patterns and the relative density of COF values, assisting in the analysis of their expression levels and the identification of high-value instances