"Ultra Chrome"
Comprehensive Deep CNN model for predicting gene expression from histone modifications

Davide Ghiotto [†] 1236660

Abstract—**Histone modifications are one of the most important factors that effect gene regulation. Computational methods that predict gene expression from histone modification signals are highly beneficial for understanding their interaction effects in gene regulation. These tools can help in developing *epigenetic drugs* for diseases like cancer. Previous studies for quantifying the relationship between histone modifications and gene expression levels either failed to capture relations between different histone modifications or they developed multiple models that addressed the task from the point of view of a single cell type. This paper explains how I built a deep spatial convolutional neural network to classify gene expression using histone modification data as input. This model extends previous deep learning frameworks by tackling the task of predicting gene expression with a global perspective with respect to different cell types, without developing multiple distinct models. I propose *UltraChrome*, an enhanced version of *DeepChrome* (*Singh et al. (2016)* [1]), as a global model which slightly outperforms state-of-the-art models like deep temporal convolutions and attention-based recurrent neural networks.**

Index Terms—**Histone Modifications, Deep Learning, Convolutional Neural Networks, Recurrent Neural Networks.**

I. Introduction

Gene regulation and the Histone Code Hypothesis

Gene regulation is a complex process that controls gene expression to be high or low. One of the most important factor that influence gene regulation is histone modifications. Histones are basic proteins that prevent DNA damage by wrapping it up in tight chromatin packs. Their special function directly determines the spatial arrangement of the DNA causing the activation or deactivation of different binding locations for interacting proteins and ultimately leading to changes in gene regulation. The importance of these *epigenetic changes* are supported by evidence that particular modifications profiles are linked to cancer (*Bannister (2011)* [2]). However, unlike DNA mutation, these changes are reversible and this fact makes these studies interesting for the development of *epigenetic drugs*.

The role of histone modifications in gene regulation is expressed by the **Histone Code Hypothesis** (*Jenuwein et al. (2001)* [3]), which states that modifications of histone proteins (changes to DNA scaffolding) are directly involved in DNA transcription and gene regulation.

Previous studied already investigated the correlation between histone modifications and gene expression. The problem now has shifted to predicting if a gene is expressed or not solely based on histone modifications signals.

Contributions

In this paper I present *UltraChrome* a Deep CNN model that extends previous deep learning architectures and whose main contributions can be summarized as follows:

- To my best knowledge, is the first implementation of a Spatial (2D) Convolutional Neural Network used to tackle gene expression prediction tasks
- Generalize over the cell type leading to one unique and comprehensive model for all the available cell types
- Sightly improves previous state-of-the-art Deep Learning models if compared over all the available cell types

II. Related Work

In the last decade, different studies addressed the combinatorial interactions of histone modifications in gene regulation. An initial study by *Lim et al.(2009)* [4] confirmed the correlation between histone modifications and gene regulation.

Later *Dong et al.(2012)* [5] implemented **Random Forests** for classifying gene expression as high or low.

Most recent studies applied deep learning models to automatically learn combinatorial interactions among histone modifications. *Singh et al. (2016)* [1] introduced ***DeepChrome***, a unified CNN framework and the first deep learning implementation for gene expression prediction task. More important than the CNN model itself, the main contribution of *DeepChrome* was to create a representation of combinatorial interactions among histone modification signals. One important drawback of *DeepChrome* is the use of temporal (1D) convolution which takes into consideration only sequences and do not capture local patterns between corresponding histone modifications.

Later, *Singh et al. (2017)* [6] developed ***AttentiveChrome***: the first attention-based deep learning method applied to this particular task. The novelty of this work is about providing an interpretation of the deep learning black-box model by using the attention scores and visualizing what the model "sees" when making predictions.

However both of these implementation failed to tackle the predictive challenge from a global point of view and they specialized their deep learning models over one particular cell type at the time.

[†] Department of Mathematics, University of Padova, email: d.ghiotto@studenti.unipd.it

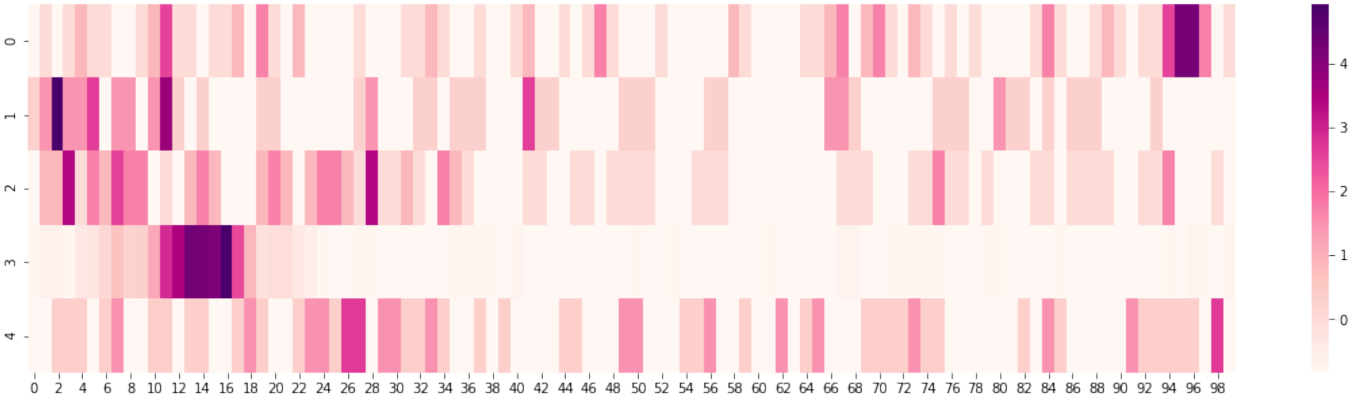


Fig. 1: Example of the matrix representation of the input

III. Processing Pipeline

The input data consists of 56 cells types each of them containing different genes represented by 5 different histone modifications over 100 bins.

Starting from the raw input generated and formatted by *Singh et al. (2016)* [1] I transformed the 5×100 matrix in one *3D* tensor of shape $5 \times 100 \times 1$ in order to correctly feed it to the convolution layer at the beginning of the deep learning model. After the convolution I placed some regularization layers and then a multi-layer architecture. The last step is simply a *softmax* operation to reduce the computation into a binary classification task.

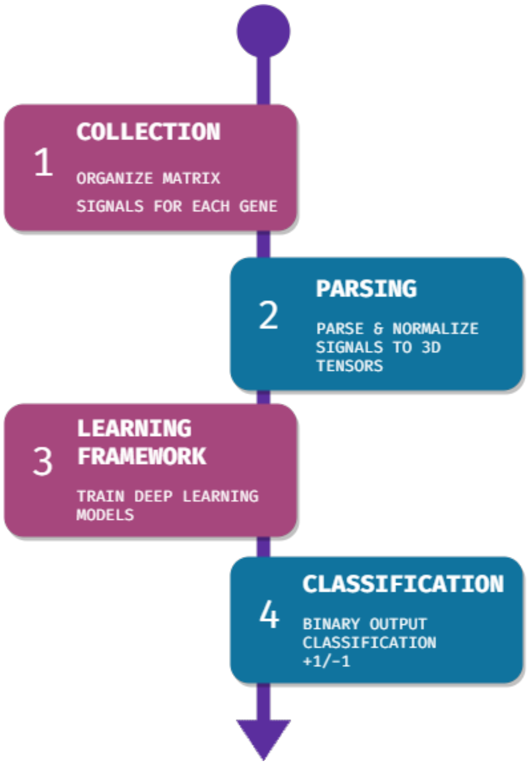


Fig. 2: Processing pipeline

The processing pipeline is summarized by the diagram Fig.2. The are 4 main steps: data collection of histone modifications signals; parsing the data into custom format for deep learning models; training with learning framework of choice; binary classification output that is gene expression high or low.

In some implementations of my models I merged all the 56 cell types datasets into one comprehensive dataset to train a *global* model. This strategy will be detailed in Sec.VI.

IV. Signals and Features

My setup is identical to *Singh et al. (2016)* [1]. Starting from the REMC[1] database, they divided the 10 000 basepair (bp) DNA region (65000 bp) around the transcription start site (TSS) of each gene into bins of length 100 bp. Each bin includes 100 bp long adjacent positions surrounding the TSS of a gene. In total, they consider five core histone modification marks from REMC database as reported in *Kundaje et al. (2015) [7]*. These five histone modifications are selected as they are uniformly profiled across all cell-types considered in that study. This makes the input for each gene a 5×100 matrix, where columns represent bins and rows represent histone modifications. For each bin, they report the value of all 5 histone signals as the input features for that bin. A visual example of an input matrix of an expressed gene is reported in Fig.1.

Here I formulate the gene expression prediction as a binary classification task. Specifically, the outputs of my models are labels 1 and -1, representing gene expression level as high or low, respectively. Following what done by *Singh et al. (2016)* [1] I split the dataset into training (6601), validation (6601) and test (6600).

In addition to this procedure I standardized the dataset using mean and standard deviation of the training set. I applied this scaling to improve the performance of the neural networks training algorithm.

[1] REMC: http://www.roadmapepigenomics.org/

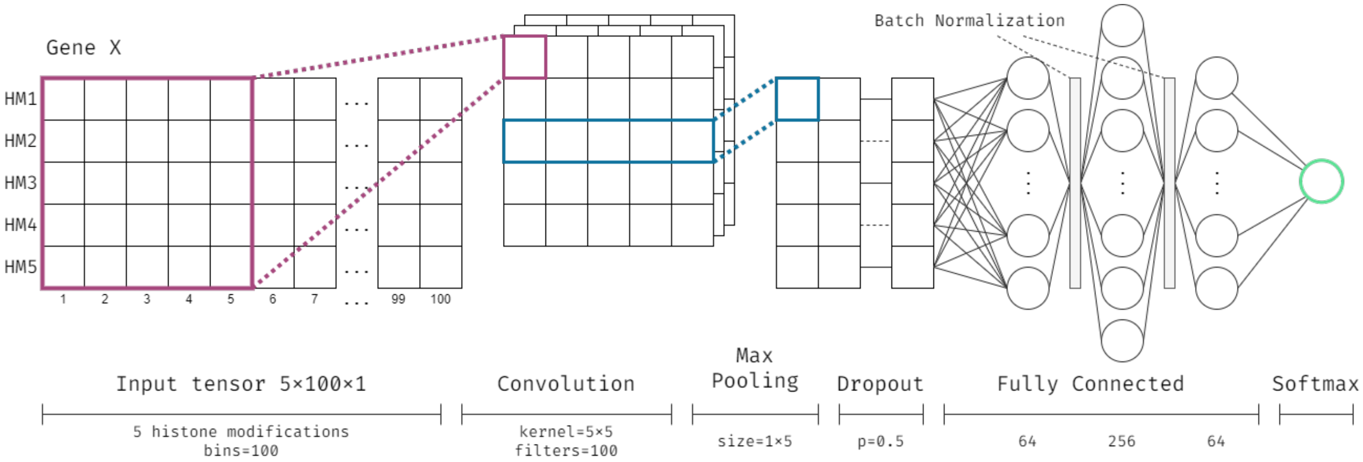


Fig. 3: *UltraChrome* Architecture

V. Learning Framework

UltraChrome architecture is composed of different layers and the diagram in Fig.3 provides a visual reference of the model, that can be summarized as follows:

- **Input layer**: the input is a 3D tensor of shape $5 \times 100 \times 1$.
- **Convolution**: I applied a spatial (2D) convolution with F filters with a kernel size of 5. This operation performs a sliding window over all bins positions and across all 5 histone modifications. The convolution enables the model to extract visual, local and global, features of the distribution of the histone modifications.
- **Pooling**: to preserve local features I inserted a *max pool* layer of shape (1×5): the particular choice of the shape aimed to capture the variability just over the bins and not also across the different modifications.
- **Dropout**: I inserted a *dropout* layer with a rate of 0.5 after the pooling to help the generalization ability of the model and reduce overfitting.
- **Flattening**: this flattening layer is a necessary step to correctly format the input for the following layers.
- **Fully Connected layers**: the core of the learning is achieved by exploiting a multi layer, *fully connected*, architecture composed of 3 different layers with N_1, N_2, N_3 neurons each of them with *relu* activation function to produce non-linearities.
- **Batch normalization**: in order to reduce the effects of the vanishing gradient problem I inserted 2 *batch normalization* layers between the 3 *fully connected* layers.
- **Output layer**: the final layer is just a *softmax* layer with one neuron that maps to the desired output in the form of a probability score.
- **Loss function**: I used the *binary cross-entropy* loss to properly match the binary classification nature of the problem.

VI. Results

A. Validation of Hyper-parameters

In order to select the best model I chose the hyper parameters validating the models on the validation set. I tested different options and values:

- Number of filters: $\{50|100\}$
- Number of neurons per layer:
  - *Low* : $[32, 128, 32]$ (3 layers)
  - *High* : $[64, 256, 64]$ (3 layers)
  - *Deep* : $[64, 256, 256, 64]$ (4 layers)
- Dropout: $\{Yes|No\}$
- Batch Normalization: $\{Yes|No\}$

In the Tab. 1 are reported a compact subset of the most significative results of the validation phase.

Hyper-parameters				**AUC validation set**		
Filters	Neurons	Dropout	BN	Min	Max	Mean
50	Low	No	No	0.653	0.935	0.764
100	Low	No	No	0.653	**0.939**	0.765
100	High	Yes	No	**0.654**	0.933	0.763
100	High	Yes	Yes	0.652	0.937	**0.766**
100	Deep	Yes	Yes	0.648	0.937	0.765

TABLE 1: Hyper parameter tuning for *UltraChrome* on validation set

As we can see, the best combination is obtain with $F = 100$ filters, high number of neurons $(64, 256, 64$ respectively), dropout with probability 0.5 and batch normalization. However the differences are not large from one combination to another: small changes in the second part of the architecture (the fully connected multi-layers) are not so relevant on the overall results, suggesting that the main contribution maybe come from the convolution itself rather than the particular deep model. Moreover, the last combination reported, with a deeper architecture, did not lead to better results, but only increased the complexity of the network.

B. *Other models*

While developing my main model *UltraChrome* I tested other models:

- **Random Forest Classifier (RFC)** with 150 estimators and max depth of 20. This model was used in one of the first attempts to investigate correlation between histone modifications and gene expression, as reported in this paper by *Dong et al. (2012) [5]*.
- **Base Neural Network (BNN)**: simple *Deep Neural Network* with two *fully connected* layers with 64 and 32 neurons respectively. This naive model was used as a baseline for more complex architectures.
- **ChromeR (CR)**: *Recurrent Neural Network* with a *Bidirectional LSTM* implementation. In particular, after validation, I used 128 units for the *LSTM* layer. After the *Bidirectional* layer I added a *Dropout* layer with a probability 0.5 to overcome overfitting, then I flattened the data and fed them to two *fully connected* layers with 625 and 125 neurons respectively and *relu* activation function.

C. *Global and individual models*

As mentioned in the introduction I, one of my contributions is also related to the type of training approach: I trained my models over the full dataset, unaware of the different cell types (the same principle applies to validation and test set). However, in order to compare the original baselines of *DeepChrome* and *AttentiveChrome* I also trained my neural networks in an individual way: for each cell type I trained a model and then I validated it and tested it only on its sets. These two configurations could be summarized as *global* and *individual*.

Moreover, implementing *DeepChrome* model myself allowed me to reproduce both configurations.

Individual configurations led to better results in terms of AUC score but required to differentiate for the cell type also at test time: this particular setup poses a strong hypothesis on the problem formulation and execution, in the sense that we must obtain the cell type separately before testing the unseen data with our model.

D. *Test Set AUC*

In the Tab 2 are reported the AUC scores for all the implemented methods and the two baselines from previous related work.

As we can see from the results, *AttentiveChrome* is the best *individual* model in terms of mean AUC. However, reporting just the mean AUC averaged over all 56 cell types is not so interesting as we look at only one single score. This is the reason why I also trained my global model individually for all the cell types. This arrangement puts my algorithm on the same level as *DeepChrome* and *AttentiveChrome*, enabling me to discriminate the single improvement with single-cell precision.

Model	**Training type**	**AUC**
Random Forest Classifier	*Global*	0.7942
Base Neural Network	*Global*	0.7930
ChromeR	*Global*	0.7961
DeepChrome	*Global*	0.8032
UltraChrome	*Global*	0.8045
Base Neural Network	*Individual*	0.7986
ChromeR	*Individual*	0.8009
DeepChrome	*Individual*	0.8068
AttentiveChrome	*Individual*	**0.8115**
UltraChrome	*Individual*	0.8114

TABLE 2: AUC scores comparison on test set

The plot Fig. 4 shows the performance in terms of AUC of the best models for each of the 56 different cell types. The data is organized as follows: first I sorted the cell types from highest to lowest referring to *AttentiveChrome* AUC scores; second I reported the other models scores following the same ordering imposed by *AttentiveChrome*. This arrangement has two benefits:

- we can visualize and compare single AUC scores between different models
- we can identify an underlying trend common to all the models: gene expression on some cell types are easier to predict than others

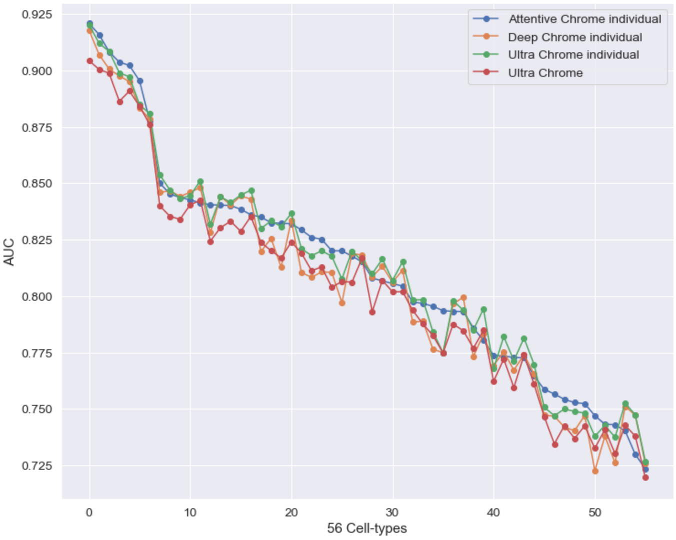


Fig. 4: AUC score on test set

With reference to Fig. 4 we can see that individual *UltraChrome* is almost always better than its global version, and it compares well with both *DeepChrome* and *AttentiveChrome*.However, from the figure is not very clear which algorithm emerge as the best, taking into consideration single cell scores.

E. Improvements Matrix

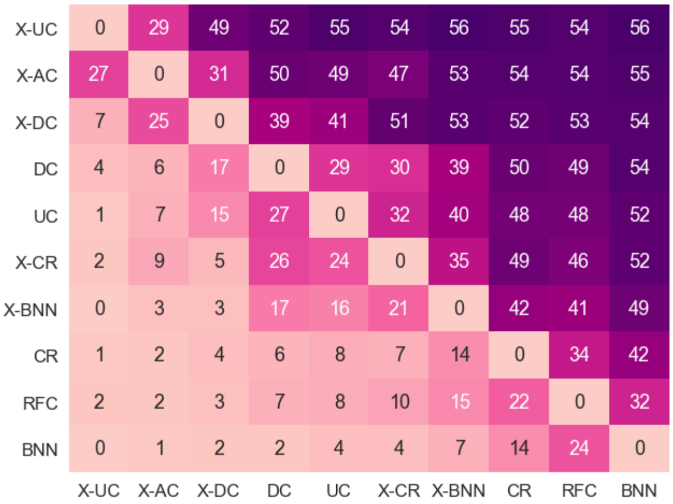
To graphically visualize the number of improvements from one model to another I built an improvements matrix, defined as follows:

$$\boldsymbol{IM} = [x_{ij}] \tag{1}$$

where:

- $i, j \in \mathcal{M}$
- $\mathcal{M}$: models set
- x_{ij} : # improvements of model i over model j.

The matrix defined by Eq. (1) is reported in the Fig. (5) as an *heatmap* for all the models available and it is properly sorted with the best performing model at the top and the darker a row the better a model compares to others.
The models set $\mathcal{M}$ contains the following models: **X-UC** (individual *UltraChrome*), **X-AC** (individual *AttentiveChrome*), **X-DC** (individual *DeepChrome*), **DC** (*DeepChrome*), **UC** (*UltraChrome*), **X-CR** (individual *ChromeR*), **X-BNN** (individual *Base NN*), **CR** (*ChromeR*), **RFC** (*Random Forest Classifier*), **BNN** (*Base NN*).



	X-UC	X-AC	X-DC	DC	UC	X-CR	X-BNN	CR	RFC	BNN
X-UC	0	29	49	52	55	54	56	55	54	56
X-AC	27	0	31	50	49	47	53	54	54	55
X-DC	7	25	0	39	41	51	53	52	53	54
DC	4	6	17	0	29	30	39	50	49	54
UC	1	7	15	27	0	32	40	48	48	52
X-CR	2	9	5	26	24	0	35	49	46	52
X-BNN	0	3	3	17	16	21	0	42	41	49
CR	1	2	4	6	8	7	14	0	34	42
RFC	2	2	3	7	8	10	15	22	0	32
BNN	0	1	2	2	4	4	7	14	24	0

Fig. 5: Improvements Matrix

Observing the improvements over all the cell types, individual *UltraChrome* (**X-UC**) comes as best model even if compared to *AttentiveChrome* (**X-AC**). However, if we look at global *UltraChrome* (**UC**) the results are not as good, but the difference in the problem setup generates an unfair comparison: the individual models are specifically trained, validated and tested on one cell type at the time; instead *UltraChrome* reaches good AUC scores in a global scenario.
There are several implications to these results:

- global models incorporate all the cell types characteristics and could describe the phenomenon as a whole
- lower probability of overfitting the model as we increase both the complexity of the task and the amount of training data
- better generalization ability over new cell types

VII. Concluding Remarks

In this paper I have presented *UltraChrome* a Deep Convolutional Neural Network that can effectively predict the gene expression by simultaneously analyzing spatial distribution of histone signals between all 5 kinds of modifications. Moreover, the model is cell type independent and do not need to know the type at test time, improving the generalization ability of the model and extending the previous models with this global approach.
The spatial convolution of *UltraChrome* intercepts patterns and interactions between different modifications. This characteristic is vital to better capture the gene expression mechanism: as reported by *Bannister et al.* [2], histone modifications can positively or negatively affect other modifications, a phenomenon called *crosstalk*. In the future, as new cell types and their relative histone modifications are added to the database we may be interested in instantly predict the gene expression without training a specific model for the new cell type. In this scenario a global model like *UltraChrome* can be useful to assign an initial indication of the gene expression to be refined later on with an individual and cell-specific model.
Future developments could improve the interpretability of *UltraChrome* model: the deep convolution internally maps the input features and extracts meaningful relations, but still miss a way to describe the biology underpinning these processes, a relevant question from an epigenetic point of view. Another improvement could be the integration of new histone modifications in addition to the 5 of the current setup. The new modifications could contribute to shed a light in more complex relations and *crosstalk* effects.

A. Project Remarks

The main difficulty was that the domain was very specific and required me some time and effort to properly understand the context of the problem. On the other hand, the project was also instructive. First, I had the chance to develop from scratch deep learning models and test several architectures and hyperparameters. Second, I was forced to build a complete learning framework, starting from data collection all the way to the communication of the results. In particular I reckon to have been very useful to think about new ways of communicate my results in an effective and elegant manner.

References

[1] G. R. Ritambhara Singh, Jack Lanchantin and Y. Qi, “DeepChrome: deep-learning for predicting gene expression from histone modifications,” *Bioinformatics*, pp. i639–i648, 2016.

[2] T. K. Andrew J Bannister, “Regulation of chromatin by histone modifications,” *Cell Research*, pp. 381–395, 2011.

[3] A. C. Jenuwein T, “Translating the histone code,” *Science*, pp. 1074–1080, 2001.

[4] K. L. B. Pek S Lim, Kristine Hardy, “Defining the chromatin signature of inducible genes in T cells,” *Genome Biology*, p. R107, 2009.

[5] A. K. Xianjun Dong, Melissa C Greven, “Modeling gene expression using chromatin features in various cellular contexts,” *Genome Biology*, p. R53, 2012.

[6] G. R. Ritambhara Singh, Jack Lanchantin and Y. Qi, “Attend and Predict: Understanding Gene Regulation by Selective Attention on Chromatin,” *Adv Neural Inf Process Syst*, pp. 6785–6795, 2017.

[7] M. K. Anshul Kundaje, “Integrative analysis of 111 reference human epigenomes,” *Nature*, pp. 317–330, 2015.