NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

PROGRAM OF POSTGRADUATE STUDIES
“DATA SCIENCE AND INFORMATION TECHNOLOGIES”

SPECIALIZATION: BIOINFORMATICS – BIOMEDICAL DATA SCIENCE

MASTER’S THESIS

Development of explainable deep learning methods for deciphering transcription factor dynamics

PANAGIOTIS XIROPOTAMOS

SUPERVISORS:

Theodore Dalamagas, Research Director, Information Management Systems Institute, ATHENA Research Center

Georgios K. Georgakilas, Scientific Associate, Information Management Systems Institute, ATHENA Research Center

ATHENS

SEPTEMBER 2024

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

"ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΠΛΗΡΟΦΟΡΙΑΣ"

ΕΙΔΙΚΕΥΣΗ

“ΒΙΟΠΛΗΡΟΦΟΡΙΚΗ – ΕΠΙΣΤΗΜΗ ΒΙΟΙΑΤΡΙΚΩΝ ΔΕΔΟΜΕΝΩΝ”

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη μεθόδων επεξήγησης βαθιάς μάθησης για την αποκρυπτογράφηση της δυναμικής των μεταγραφικών παραγόντων

Παναγιώτης Ξηροπόταμος

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Θεόδωρος Δαλαμάγκας**, Διευθυντής Ερευνών, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

Γεώργιος Κ. Γεωργακίλας, Επιστημονικός Συνεργάτης, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2024

MASTER THESIS

Development of explainable deep learning methods for deciphering transcription factor dynamics

Panagiotis Xiropotamos

SRN: 7115152200038

SUPERVISORS: **Theodore Dalamagas**, Research Director, Information Management Systems Institute, ATHENA Research Center

Georgios K. Georgakilas, Scientific Associate, Information Management Systems Institute, ATHENA Research Center

EXAMINATION COMMITTEE: **Theodore Dalamagas,** Research Director,
Information Management Systems Institute,
ATHENA Research Center

Georgios K. Georgakilas, Scientific Associate,
Information Management Systems Institute,
ATHENA Research Center

Artemis G. Hatzigeorgiou, Professor of Bioinformatics,
Department of Computer Science and Biomedical Informatics,
University of Thessaly

SEPTEMBER 2024

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη μεθόδων επεξήγησης βαθιάς μάθησης για την αποκρυπτογράφηση της δυναμικής των μεταγραφικών παραγόντων

Παναγιώτης Ξηροπόταμος

Α.Μ.: 7115152200038

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

Γεώργιος Κ. Γεωργακίλας, Επιστημονικός Συνεργάτης, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ : **Θεόδωρος Δαλαμάγκας,** Διευθυντής Ερευνών, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

Γεώργιος Κ. Γεωργακίλας, Επιστημονικός Συνεργάτης, Ινστιτούτο Πληροφοριακών Συστημάτων, Ερευνητικό Κέντρο «ΑΘΗΝΑ»

Άρτεμις Γ. Χατζηγεωργίου, Καθηγήτρια Βιοπληροφορικής, Τμήμα Πληροφορικής με Εφαρμογές στη Βιοϊατρική, Πανεπιστήμιο Θεσσαλίας

ΣΕΠΤΕΜΒΡΙΟΣ 2024

ABSTRACT

Living cells require a differential activation of transcription factors (TFs) to establish cell type-discriminating gene expression. Binding affinity, timing, and combinatorial activity with other TFs at enhancer and promoter regions affect the expressed gene product at each state and while some of those associations have been deciphered through experimentation, a comprehensive understanding of the dynamic TF interactions during each cell state remains elusive. In this study, we utilized chromatin affinity data via ChIP-seq[3] protocol for TCF1 factor, which encompasses regulatory regions such as enhancers and promoters, where the factor binds and regulates gene expression, on a genome-wide scale for two cell types, Double positive T cells, which naturally produce TCF1, and NIH3T3 fibroblasts, which are induced to express TCF1. The binding profile was used as input to an in-house developed Convolutional Neural Network (CNN) model, trained to distinguish whether a sequence is accessible by TCF1 to either of the cell types, to therefore infer any biological background concerning the two systems. Our hypothesis posited that transcription factor motifs, pivotal in either state, underlie the key distinguishing features among diverse regulatory elements in the two different biological systems. Interpreting the first convolutional layers and especially the filter kernels provided us with insights into what the model grasped during training, to associate DNA sequences to cell types. The novel-developed interpretability modules provide insights into the TFs implicated in the modeling process, how important they are in the decision-making, their association with the studied systems, and most importantly the interplay of those learned representations, inferring the dynamics of those TFs among the two cell types. Among the results, binding sites of known cooperators of TCF1, such as JUNB and FOSL1 have been found among others, while the application of the interpretability modules on model snapshots during training has provided an enhanced view of the training process. By applying artificial intelligence to such complex biological questions of TF dynamic activity, we show the modeling capabilities of convolutional neural networks to identify TF binding sites and associate them to cell types only from the sequence itself, enabling us to further inspect the explainability opportunities that those models have to offer.

SUBJECT AREA: Explainable Convolutional Neural Networks, Genomic Modeling

KEYWORDS: CNN, explainability, Transcription factors, dynamics, bioinformatics, genomics

ΠΕΡΙΛΗΨΗ

Τα κύτταρα απαιτούν τη διαφορική ενεργοποίηση παραγόντων μεταγραφής (TFs) για να καθιερώσουν την χαρακτηριστική γονιδιακή έκφραση που τα διακρίνει. Καθώς η δέσμευση, ο χρόνος και η συνδυαστική δραστηριότητα με άλλους μεταγραφικούς παράγοντες στους ενισχυτές και τους υποκινητές των γονιδίων επηρεάζουν την έκφραση, και ενώ ορισμένες από αυτές τις συσχετίσεις έχουν αποκωδικοποιηθεί μέσω πειραμάτων, η κατανόηση των δυναμικών αλληλεπιδράσεων των παραγόντων μεταγραφής στον εκάστοτε κυτταρικό τύπο παραμένει ασαφής. Σε αυτήν τη μελέτη, χρησιμοποιούνται δεδομένα συμπλόκων παραγόντων - χρωματίνης μέσω του πρωτοκόλλου ChIP-seq για τον παράγοντα TCF1, τα οποία περιλαμβάνουν ρυθμιστικές περιοχές όπως ενισχυτές και υποκινητές, όπου ο παράγοντας δεσμεύεται και ρυθμίζει την έκφραση των γονιδίων σε ολόκληρο το γονιδίωμα, για δύο τύπους κυττάρων, τα διπλά θετικά Τ κύτταρα, τα οποία παράγουν φυσικά τον TCF1, και τους ινοβλάστες NIH3T3, οι οποίοι επάγονται να εκφράζουν το TCF1. Οι διαφορικές θέσεις δέσμευσης χρησιμοποιήθηκαν ως είσοδος σε ένα νευρωνικό δίκτυο συνέλιξης (CNN) που εκπαιδεύτηκε να διακρίνει αν μια αλληλουχία είναι προσβάσιμη από τον TCF1 για καθέναν από τους τύπους κυττάρων, ώστε να συναχθεί οποιοδήποτε βιολογικό υπόβαθρο που αφορά τα δύο συστήματα. Η υπόθεσή βασίζεται στο γεγονός ότι τα μοτίβα των συνεργατών παραγόντων μεταγραφής, τα οποία είναι καθοριστικά στις δύο καταστάσεις, βρίσκονται εκατέρωθεν του μοτίβου πρόσδεσης του TCF1 στα δύο διαφορετικά βιολογικά συστήματα. Η ανάπτυξη μεθόδων για την ερμηνεία των πρώτων επιπέδων συνέλιξης και ιδιαίτερα των φίλτρων παρείχε πληροφορίες για το τι έμαθε το μοντέλο κατά την εκπαίδευση, ώστε να μπορεί να συσχετίσει τις αλληλουχίες DNA με τους τύπους κυττάρων. Οι μέθοδοι επεξήγησης που αναπτύχθηκαν παρέχουν πληροφορίες για τους μεταγραφικούς παράγοντες που εμπλέκονται στη διαδικασία της μοντελοποίησης, τη σημασία τους στη λήψη αποφάσεων, τη συσχέτισή τους με τα μελετόμενα συστήματα και κυρίως τη δυναμική αλληλεπίδραση αυτών των αναπαραστάσεων που έμαθε το μοντέλο μεταξύ των δύο τύπων κυττάρων. Μεταξύ των αποτελεσμάτων, βρέθηκαν θέσεις δέσμευσης γνωστών συνεργατών του TCF1, όπως ο JUNB και ο FOSL1, μεταξύ άλλων, ενώ η εφαρμογή των μεθόδων επεξήγησης σε στιγμιότυπα του μοντέλου παρείχε πληροφορίες για την διαδικασία της εκπαίδευσης. Εφαρμόζοντας την τεχνητή νοημοσύνη σε ένα τόσο σύνθετο βιολογικό ερώτημα της δυναμικής δραστηριότητας των παραγόντων μεταγραφής, καταδεικνύονται οι δυνατότητες μοντελοποίησης των νευρωνικών δικτύων συνέλιξης να αναγνωρίζουν τις θέσεις δέσμευσης των συνεργατών παραγόντων μεταγραφής και να τις συσχετίζουν με τους τύπους κυττάρων μόνο από την ίδια την αλληλουχία, δίνοντάς τη δυνατότητα να εξεταστούν περαιτέρω οι δυνατότητες επεξήγησης που προσφέρουν αυτού του τύπου τα μοντέλα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: επεξήγηση νευρωνικών δικτύων συνέλιξης, γενομική μοντελοποίηση

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: CNN, επεξήγηση, μεταγραφικοί παράγοντες, δυναμική, βιοπληροφορική

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, Dr. Theodore Dalamagas, Professor in the “Data Science and Information Technologies” master's program and Research Director of the Information Management Systems Institute at ATHENA Research & Innovation Center for his support during both the master’s courses and my thesis.

I am also very grateful to my long-time supervisor, Dr. Georgios K. Georgakilas, Post-Doc Researcher and Scientific Associate at the same institute, who has been a dedicated and valuable mentor since my undergraduate studies and ongoing.

I would like to thank prof. Artemis G. Hatzigeorgiou, Professor of Bioinformatics at the Department of Computer Science and Biomedical Informatics at the University of Thessaly, for her collaboration and ıinvolvement with the examination committee.

I would like to thank my colleagues, Haris Manousaki and Charis Sinnis, for their continuous support during the past months.

I am grateful to my parents, sister, and friends for their support and encouragement in every step of my life.

Thank you!

Panagiotis Xiropotamos

September 2024

Table of Contents

1. Introduction

1.1 Biological Background

1.1.1 Differentiation process

Cell differentiation describes a biological process, in which stem or pluripotent cells develop into cell types with distinct molecular functions. In the early stages of development, all cells are at stem cell state and have the ability to differentiate into any cell type within the organism. Cells acquire unique molecular features during development , structures, and roles needed to form the micrroenvironment of tissues and at a higher level, organs[1]. This process is pivotal in embryonic development but also, in tissue repair and regeneration in the latter stages of development and adult life.

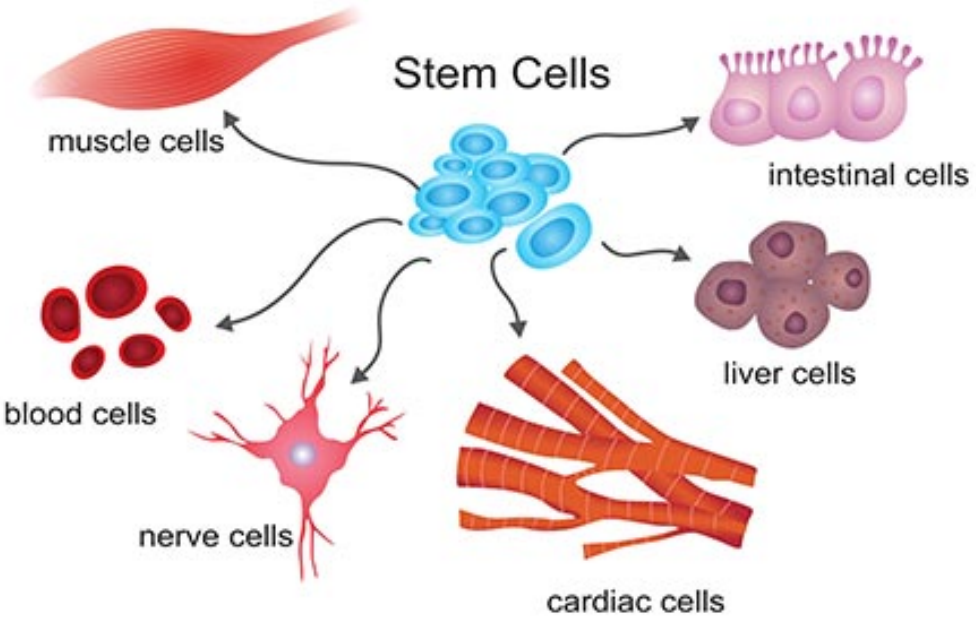


Figure 1-1. Graphical representation of cell differentiation.

Despite the fact that cells within an organism contain identical DNA, different cell types express distinct sets of genes, a phenomenon that characterizes the cells' unique functionality. The regulation of gene expression is needed for the cell differentiation as it is determined which genes are activated or silenced. The regulation lies on external molecular signals, signalling, and transcription factors, which bind to specific DNA sequences, called motifs, and control the gene expression. Epigenetic modifications,

including DNA methylation and histone modifications, influence gene expression by modifying the genome accessibility without changing the DNA sequence itself.

1.1.2 Gene expression patterns

The progression of differentiation lead cells to commit specific lineages, express genes that are necessary for their unique molecular functions. For instance, muscle cells activate genes involved in the contraction of muscles, while neurons express genes that have cruicial role in neuron signalling and transmission. External signals, namely growth factors and morphogens, also play a key role by guiding cells to their differentiation pathways and also trigger cascades of gene expression alterations, enabling the cell to adopt its final identity or environmental changes. [2], [3] The regulation of gene expression ensures the development and maintaining of cells into specific types, contributing to the complex structure and function of living organisms. Understanding the mechanisms of differentiation [4] and the role of gene expression is essential to decode the intricacies of development and maintenance of tissue structure.

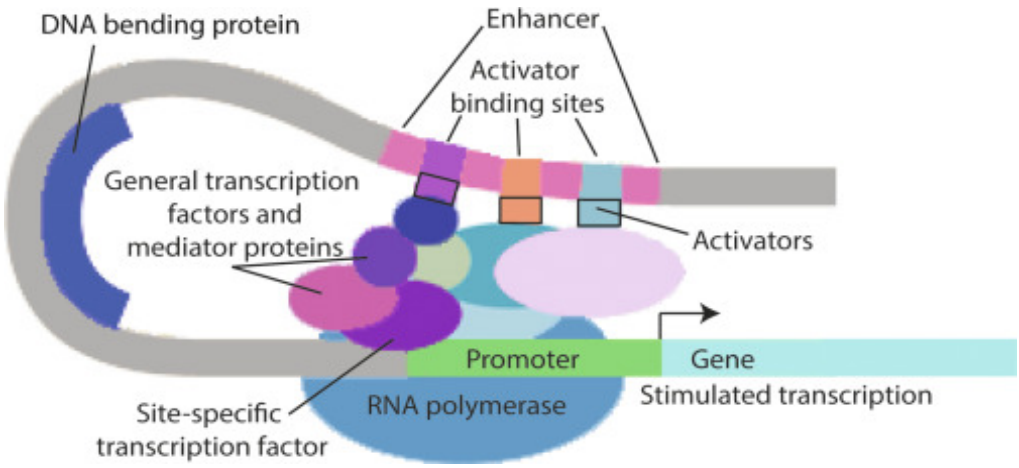


Figure 1-2. Molecular interactions involved in the regulation of gene expression.

1.1.3 Transcription factors

Transcription factors (TFs) are proteins that have a key role in controlling gene expression, binding to specific DNA sequences near or distal to the target genes. TFs function as molecular switches to determine whether a gene is expressed or silenced in a specific cell type and cell cycle state. This regulation is cruicial for controlling the process of cell differentiation and enabling cells to adopt specialized roles with their moleculat functions and structure, despite the fact that they share the same DNA across all cell types. [5]

Transcription factors bind to regions of DNA, distal or proximal regulatory elements, on promoters or enhancer regions, [6] which are located upstream or at a distance from the genes they regulate and either recruit or block the transcription complex, including RNA polymerase II enzyme, which is responsible for transcribing the gene into messenger RNA (mRNA). mRNA is then translated into proteins that carry out various cellular functions. In this way, transcription factors modulate the production of specific proteins or other gene products that are necessary for a cell's identity and function.[7]

The specificity of transcription factor binding is determined by the sequence of nucleotides in the DNA, as each TF recognize specific DNA motifs. However TF cooperativity supports their regulatory effect, as some TFs have no specific binding and are recruited by other factors. Several transcription factors collaborate at a single distal or proximal regulatory element, forming complexes that fine-tune the level of gene expression [8], [9]. Additionally, transcription factors can act as activators, promoting gene expression, or as repressors, inhibiting it. This combinatorial control allows for accurate regulation of gene expression, leading different cell types to express unique sets of genes even them having identical genetic material.

1.1.4 Epigenetics impact on TF binding

While the DNA sequence remains near identical to all cells of an organism, the epigenome—a collection of chemical modifications to the DNA and histone proteins—is dynamic in different cell types. These epigenetic changes create the landscape of accessibility of DNA to transcription factors and other regulatory proteins, therefore directly regulate the gene expression patterns of each cell.[10]

One of the most significant epigetic modifications is DNA methylation, which occurs at cytosine bases in regions called CpG islands [11], [12]. In case of heavy methylation of CpG islands, the corresponding gene region becomes less accessible to transcription factors and leads to silencing of the associated gene. In other case where areas of DNA are less methylated, the DNA sequence remains open, allowing transcription factors to bind and promote the expression of the gene product. Through those selective methylation patterns, cells can ensure that only the genes necessary for their specific molecular and structural functions are expressed while others are suppressed, maintaining their stability.

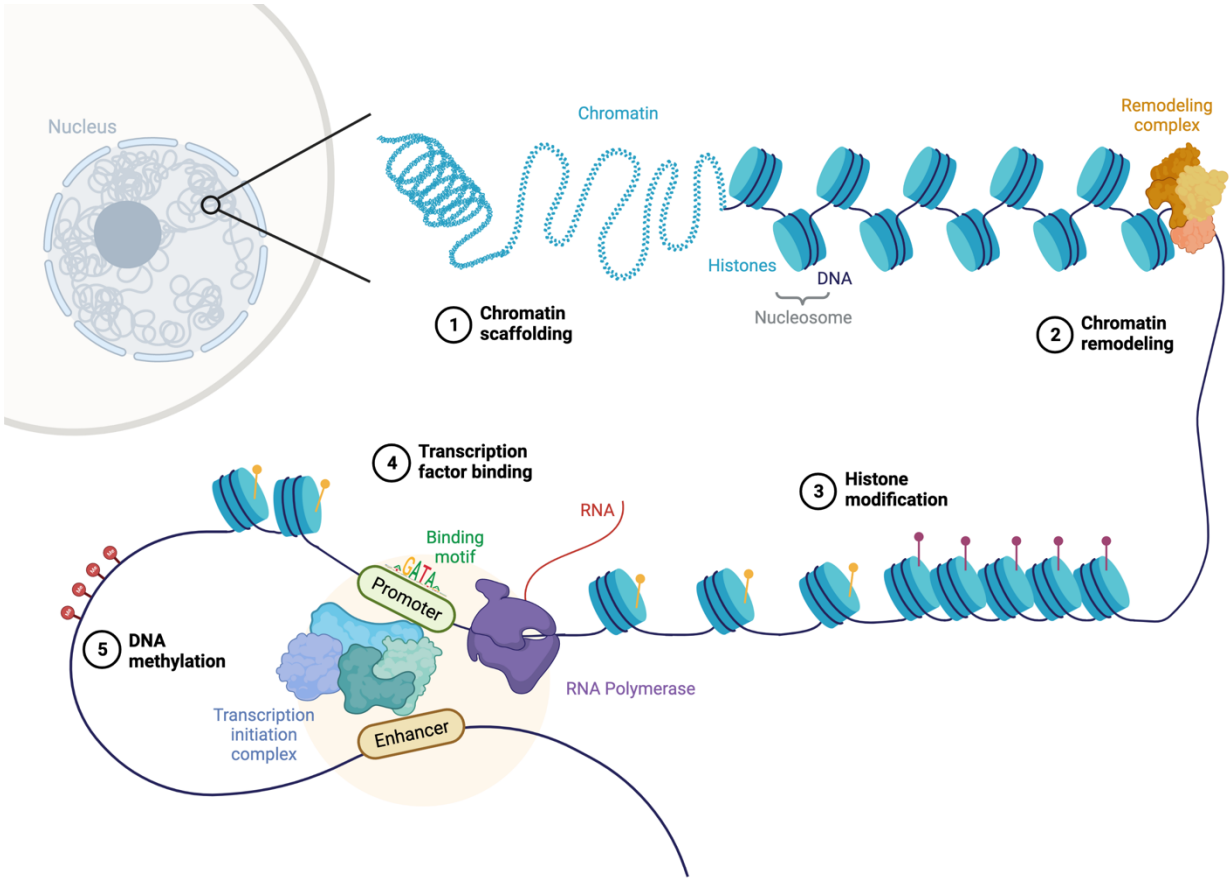


Figure 1-3. Multi-step regulation of gene expression through chromatin dynamics and modifications.

Another key epigenetic modification involves the chemical alteration of histone proteins, around which DNA is wrapped to form chromatin. Histones can be modified through processes such as acetylation, methylation, or phosphorylation, which affect how tight the DNA is perplexed around them. Acetylation of histones, for example, loosens the chromatin structure, making the DNA sequence more accessible to transcription factor binding. On the other hand, methylation of certain histone residues can either activate or repress gene expression, depending on the specific location of the modification.[13]

These epigenetic modifications create a dynamic chromatin structure that can either permit or block the binding of proteins and transcription factors to DNA. In differentiated cells the specific epigenetic structure of the genome allow transcription factors to recognize and bind to only the relevant regions of the genome to the cell's identity. For example, a transcription factor that activates muscle-specific genes will be able to bind to its target sequences only in muscle cells, where the chromatin is accessible, while the same regions in other cell types may be tightly packed and inaccessible due to those epigenetic modifications.[14], [15]

1.1.5 Methods for genome-wide accessibility profiling

1.1.5.1 ATAC-seq

ATAC-seq (Assay for Transposase-Accessible Chromatin followed by Sequencing) is a protocol used to study genome-wide chromatin accessibility. It identifies regions of the genome where the chromatin is loosely packed and allowing DNA-binding proteins such as transcription factors to bind. This technique uses a modified TN5 transposase, which inserts sequencing adapters into accessible DNA regions that can bind and cut. The desired fragments are then sequenced, revealing open chromatin sites that potentially indicate active regulatory elements. ATAC-seq is widely used to understand how chromatin structure influences gene regulation and expression. [16], [17], [18]

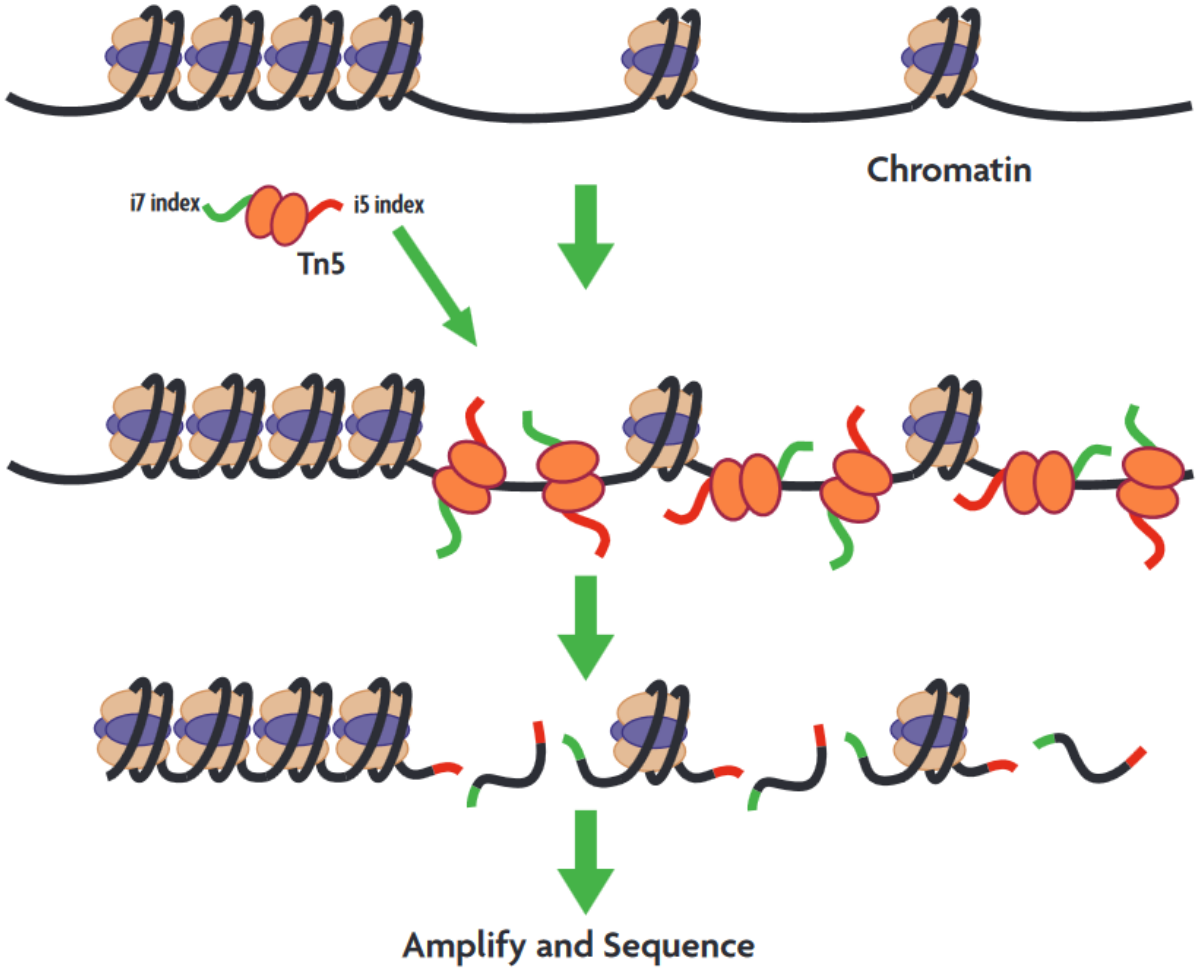


Figure 1-4. Overview of ATAC-sequencing protocol.

1.1.5.2 ChIP-seq

ChIP-seq (Chromatin Immunoprecipitation followed by Sequencing) is a protocol used to identify specific DNA-binding of proteins, including transcription factors and histone modifications, across the genome. It involves crosslinking proteins to DNA making stable complexes, fragmenting the DNA, and using specific antibody for the protein to separate the protein of interest along with its bound DNA. The DNA sequence associated with the protein is then sequenced to map the binding locations of the protein across the genome. ChIP-seq provides insights into how transcription factors, histones, and other chromatin-related proteins regulate gene expression by binding to specific genomic regions.[19], [20], [21]

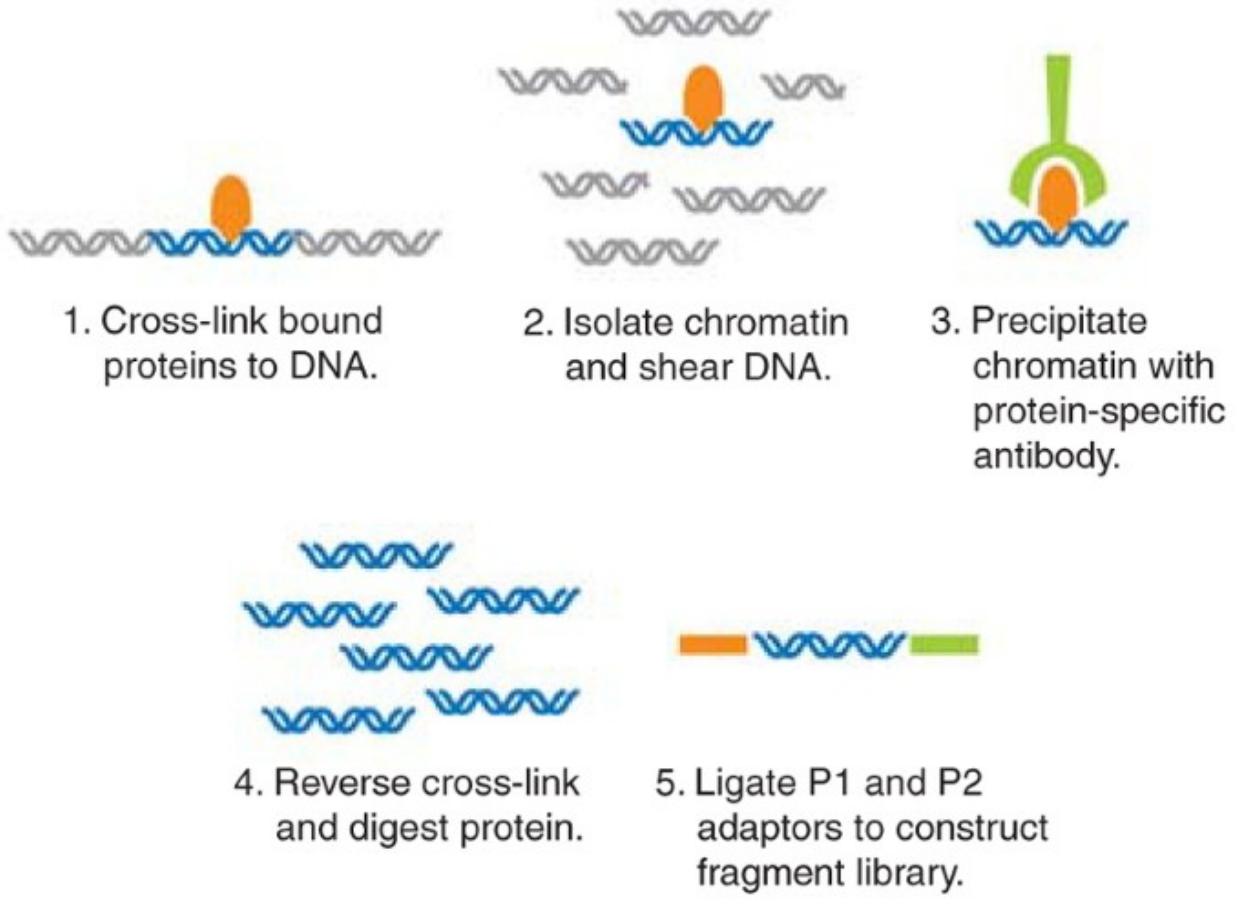


Figure 1-5. Overview of ChIP-sequencing protocol.

1.2 Problem statement

1.2.1 Need for a holistic view

Transcription factors' dynamics are highly interconnected and operate in a complex regulatory landscape shaped by chromatin structure and epigenetic modifications.A

holistic approach is essential to capture the full spectra of their regulatory effect and understand how gene expression is controlled in different cell types and conditions.

It is practically impossible to perform ChIP-seq experiments for all transcription factors (TFs) due to both cost and experimentation time needed. ChIP-seq requires specific antibodies for each TF, and producing high-quality antibodies for every TF is expensive and technically difficult, while in some cases there is no specific antibody. Additionally, ChIP-seq is a labor-intensive process, involving cell preparation, immunoprecipitation, sequencing, and data analysis for each TF. Given that there are one and a half thausand of TFs, the time, cost, and resources needed to conduct experiments for all of them across different conditions and cell types make it impractical. On the other hand, while ATAC-seq is much more efficient and less labor-intensive, it doesn't provide direct information about which specific TFs are binding to open chromatin regions. ATAC-seq identifies regions of accessible DNA, indicating where TFs could bind, but it doesn’t specify which TFs are bound or their precise binding motifs, or if any TF binds in general to those open chromatin regions. This makes it a less insightful tool for determining exact TF binding sites compared to ChIP-seq, which directly maps TF-DNA interactions.

1.3 In silico methods for modeling TF activity

Transcription factor binding is crucial for gene expression regulation, therefore a holistic view of TF interplay in specific cell states is essential for our understanding of the regulation of genes. Even though TF-ChIP-seq experiments provide a genome-wide view of the binding for a specific factor, a complete view would require a substantial investment in both human and financial resources due to the extensive experimentation involved. Those limitations could be mitigated by employing CNN models to model the binding of specific transcription factors and then applying explainability methods to retrieve the model-acquired knowledge for the cooperative action of other factors.[22], [23], [24]

1.4 Artificial Intelligence

Artificial intelligence (AI) provides robust modeling capacity, especially in recognizing patterns, making predictions, and extracting insights from large datasets.[25] AI models trained, have proven to solve complex problems in various domains, from image and speech recognition to natural language processing.

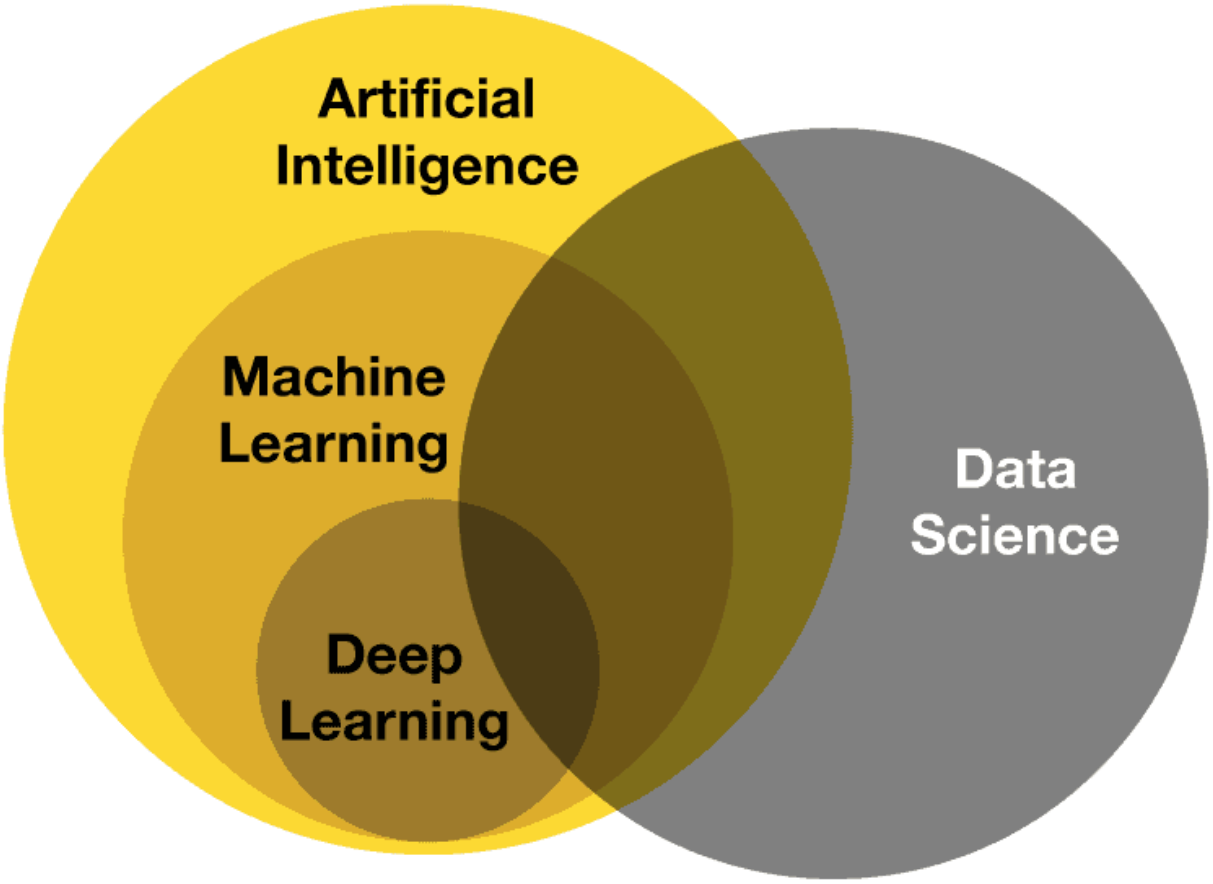


Figure 1-6. Venn diagram of artificial intelligence and data science.

1.4.1 Machine learning

Machine learning is the branch of computer science that studies the automated solving of complex problems by applying programming tools. Two key challenges in applying machine learning are the clear description of the problem for the detailed programming of the model and the appropriate processing and transformation of the data to be manageable by the computer. In the field of machine learning, three subcategories are distinguished according to the type of learning:

Supervised learning – The model is trained on a data set with known outputs – labels to find important features. Once trained it can predict the output on an unknown data set. Depending on the output type the model is applied to classify the input into categories or predict a continuous variable (regression).

Unsupervised learning – The model is trained on a data set without desired outputs. Based on the characteristics of each data, it identifies the most important ones and groups all the data based on similarity.

Reinforcement learning - The model through feedback carries out some tests. Through interaction with the environment, it learns a series of actions. It is a dynamic model that can adapt to changes in the environment.[26]

1.4.2 Deep learning

Deep learning is the implementation of artificial neural networks on a larger scale to solve more complex questions. Representation learning is the basic operating principle of deep learning networks, according to which the model is trained on a specific problem through examples, just as humans learn through experience. More specifically deep learning models combine features given to the input through the feature map or even extract them by itself through convolutional networks. Another advantage of deep learning networks is the combination of convolutional architectures, deep networks, and recurrent networks that can be created and trained in parallel.[27]

1.4.3 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a class of artificial deep learning networks. In recent years with the development of graphics cards, the field of CNNs has seen an advent, as with the advantages they bring over other architectures they have shown to outperform in almost every test. The key feature of CNNs is their ability to learn features from a raw input. [28]Other architectures require human extraction of features to analyze the data, adding bias to the equation of how right or wrong the features we extract are. CNNs during training, with learning representation, extract the data features on known input and use this knowledge to predict the output in a completely objective manner. The architecture of convolutional neural networks is structured by:

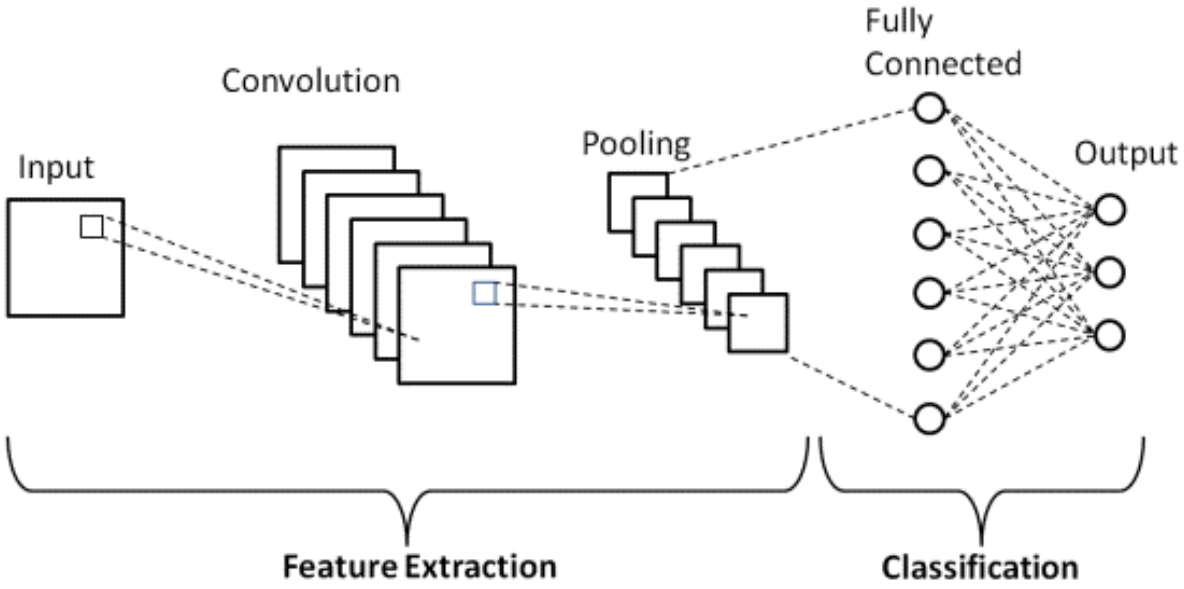


Figure 1-7. Schematic representation of a Convolutional neural network.

Convolution layer: The convolution layers consist of filters that always have smaller dimensions than the input, scanning all possible positions of the input samples and their similarity to the filter is compared. This results in a three-dimensional feature map with the first two dimensions being the dimensions of the input reduced by the dimensions of the filter, and the third being the number of filters. In other words, it is a map showing the similarity values of each filter at each possible position on the input. These filters during training are changed to depict features that help the model classify the input into a particular output. The output of the convolution layer can be used as input for a subsequent convolution layer just like in deep neural networks and in this way, the model learns combinations of features of the previous layer.

Pooling layer: This is a level that allows the reduction of computations since CNNs can become very complex. Pooling refers to reducing the dimensions of the output from the convolution plane to condense the information of the convolution. Pooling can be done in different ways to best fit the problem the model is solving and can be finding the highest, lowest, or average filter activation value within the pooling area.

Dropout layer: CNNs and DNNs in general due to their complexity tend to overfit, learning the training data to a great extent but not being able to generalize and use the knowledge acquired to make predictions on a held-out set. During dropout, random neurons and filters per layer are deactivated so that the model becomes more capable of combining different features of the input to make predictions.

Fully connected layer: Consists of one or more hidden layers in which most of the model calculations are done. The hidden layer consists of a set of artificial neurons and their role is to combine the information of the previous layers.

Output layer: The output layer is fully connected to every neuron from the previous layer. It consists of neurons with sigmoidal, softmax, or linear activation depending on the modeling task. Softmax and sigmoid are used for classification, while the linear activation function for regression models.

1.5 XAI – Explainable Artificial Intelligence

The explainability of deep neural networks (DNNs) refers to the ability to understand and interpret how these complex models make decisions or predictions. While DNNs are powerful tools for solving a wide range of problems, their complexity characterized by numerous layers and parameters leads to a "black box", making it difficult to grasp the rationale behind specific outputs.[29]

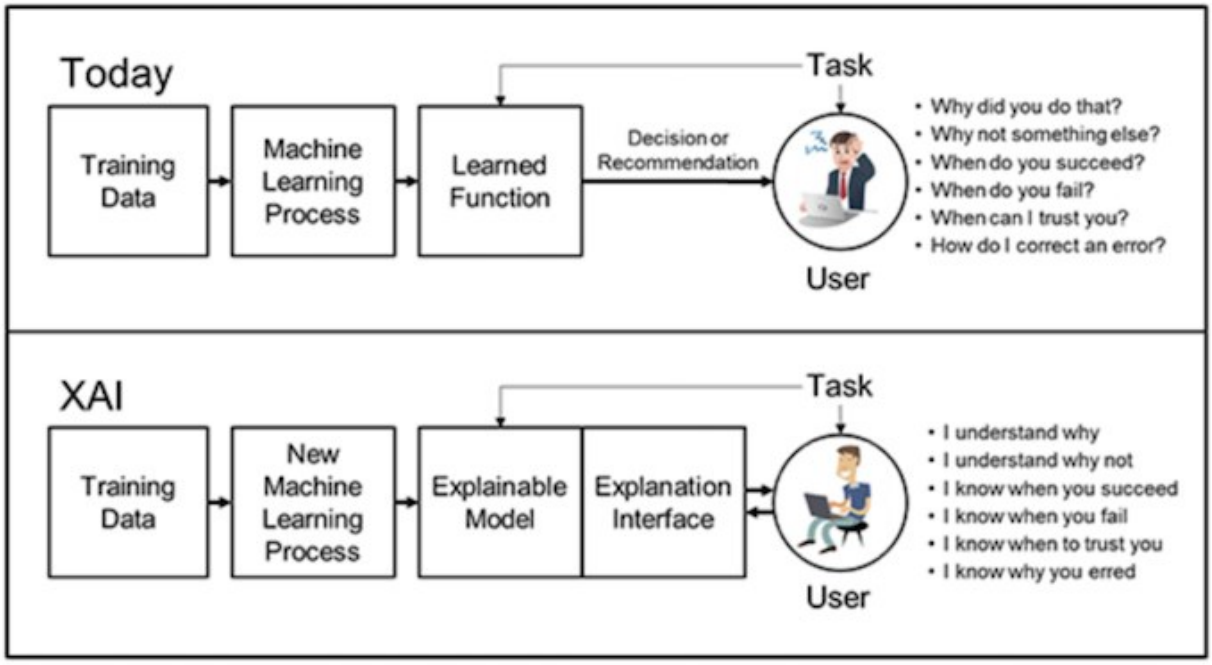
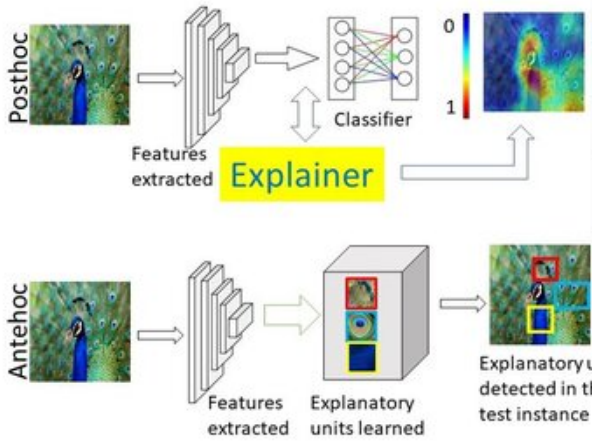


Figure 1-8. Comparison of traditional machine learning and Explainable AI.

1.5.1 Post-hoc vs ante-hoc

Post-hoc and Ante-hoc explainability approaches are used to describe the methods applied to machine learning models making them transparent and providing insights about their decision-making. Post-hoc approaches refer to explainability methods that are applied after the training of a model, treated as a “black box” due to its complexity. In the

case of a classifier with a convolutional layer that is trained to predict if an image contains a cat or dog, the use of SHAP values [30], [31], to identify the pixels of an image that contribute to its decision-making (features), is considered post-hoc. Ante-hoc approach refers to explainability methods that are incorporated into the model architecture during the model development process and are designed to be inherently interpretable. In the case of a linear regression model, the relationship between the data points is easy to understand and interpret through the models' coefficients. [31]



Aspect	Posthoc	Antehoc
Modifications to black-box architecture	Not Necessary	Necessary
Training/ Retraining	Not Necessary	Necessary
Faithfulness	Challenging	Guaranteed
Retaining black-box accuracy	Guaranteed	Challenging

Figure 1-9. Comparison of post and ante hoc interpretability methods.

Ante-hoc approaches are inherently faithful, as the transparent models align directly with their decision-making process, but they may compromise accuracy because of the prioritization of simplicity and interpretability over complexity. These models might also require retraining during development to balance accuracy and transparency. Post-hoc approaches preserve the accuracy of complex black-box models, as these methods are applied after the model is built and their structure is not modified, but explanations may not always be faithful, as they try to approximate the model's decision-making rather than directly trying to dissect its internal logic.[32]

1.5.2 Global vs local explainability

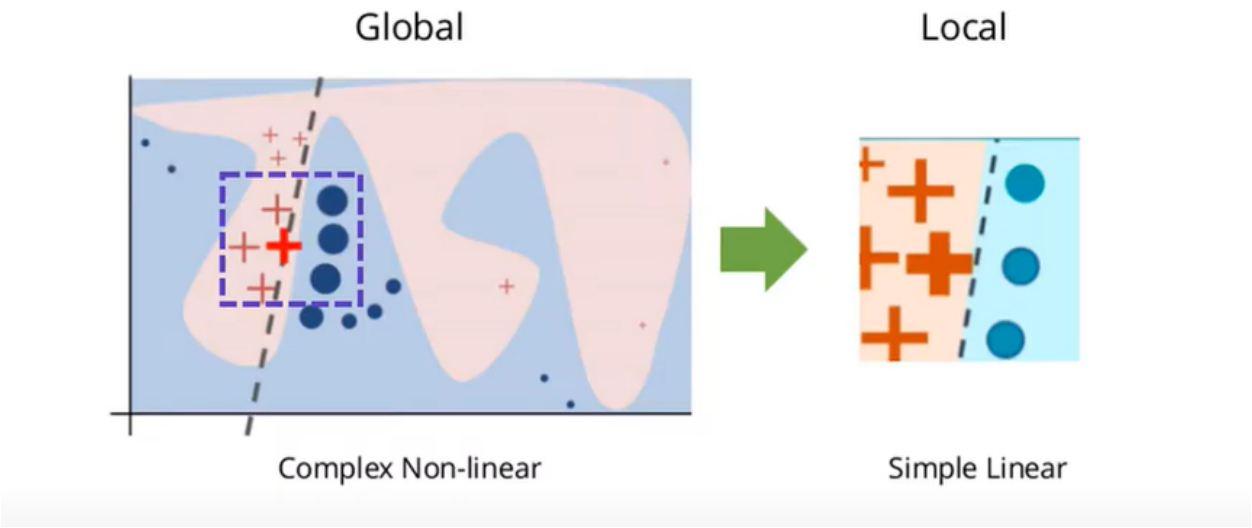


Figure 1-10. Comparison of local and global explainability approaches.

Global and local approaches refer to describe the field of view that a method is applied during explainability. Local explainability focuses on individual predictions of the model, trying to describe the decision-making of the model for a specific input. In the case of the CNN classifier of images for the detection of dogs and cats, the application of SHAP values will provide insights into areas of interest for a specific image, highlighting features that contribute to its prediction. Global explainability does not focus on specific events but tries to provide insights regarding the models' behavior, decision patterns, and feature interactions under any input. Regarding the example of the CNN model, visualization of the convolutional layers' kernels could provide a feature pool that the model has learned to identify in any input to make its predictions as edges and curves or more robust features such as ears and fur in the first and deeper layers respectively.[33]

1.6 Thesis proposal and objectives

The hypothesis behind this reasoning relies on the cooperativity of transcription factors during gene regulation, as they act in clusters to enable or disable the activity of specific genes. During this thesis, ChIP-seq data for the TFC1 transcription factor will be employed derived from two different systems, double positive T-cells that express naturally the factor and is pioneer in the t-cell development process, and fibroblasts (NIH3T3) that are induced to express TCF1[34]. In the two systems that differ in gene

expression and TCF1 binding, the methodology tries to model the binding of TCF1 and its cooperators in unique TCF1 binding sites across the two conditions. By modeling the binding, as the TCF1 motif is present in both sets, the model should learn to identify binding sites of cooperate factors that are located upward and downward the TCF1 binding site.

On the trained model, post-hoc and global explainability methods will be applied to extract the model-acquired knowledge about the binding of cooperators of the TCF1 between conditions. Interpretability modules include a motif detector that will visualize the filters of the first convolutional layer, where literature has proven that TFs motifs are stored[35], a filter assessment for the importance of each motif to the prediction, filter co-activation patterns and an enrichment module of filters to specific cell types. These modules should answer the questions -which TFs? -, -How important they are?-, -How they cooperate?-, and -In which cell types are they enriched?-, providing a holistic view of TF dynamics in a specific biological context. The outcomes of the modules will be then used to reconstruct the TF interplay across cell types on a plot representation that will captivate the model decision process and therefore the dynamic relations of TF across the two systems.

2. Related work

2.1 Related pipelines - examples and limitations

During the latent years with the advent of next-generation sequencing and GPUs, there has been an effort by research groups to infer transcription factor binding from sequence modeling via CNN and therefore the application of explainability methods. Most of the methodologies though are restricted to visualizing the filter kernels or implementing attention layers in the model's architecture to capture the TFs and their activity respectively. TF-MoDISco is a method for the identification of motifs using deep learning, by computing importance scores for each input as a local interpretability procedure. Those sequences of scores are segmented and then clustered to retrieve motifs that are nearly identical and important for the model predictions, providing no information about the filters' quality.[36] Basset on the other hand, is a model trained to predict chromatin accessibility based on sequence input and performs global and post hoc interpretability by visualizing filter kernel with the positive activation technique as described in the methods section, measures the effect of each filter, by disabling the output of each and assessing the performance of the model [37]. SATORI deep learning model that captures regulatory element interactions in genomic sequences, implementing CNN to capture the

motifs and a self-attention layer that is then used to infer the motifs' interactions as an ante hoc approach without post-processing of the model [35], [38]. Even though there have been efforts to describe the TF cooperativity by modeling genomic sequences, there is not yet an end-to-end approach that provides a clear view of both which TFs are important, how important they are for the decision-making of the model, and how they cooperate.

Biological experiments that capture the TF activity such as ChIP and ATAC-seq tend to be noisy but are used as ground truth on both training and testing of the models [35]. Improved predictions on unvalidated experimental benchmark datasets may not necessarily serve as a reliable way of comparing model performance thus model interpretability could provide insights on whether a model is reliable and captured any biological background. To overcome those issues, the proposed methodology will provide extensive insights into the features learned by the model, their importance, interactions, and enrichment in classes, clarifying whether a model has grasped a meaningful biological background.

3. Materials and Methods

3.1 Dataset generation

3.1.1 NGS introduction

Next Generation Sequencing is a parallel sequencing method that yields a vast amount of data. Depending on the sequencing technology, physicochemical phenomena during nucleotide binding are measured to determine the composition of DNA segments. Technology is used to determine the sequence of nucleotides in whole genomes or targeted regions of DNA or RNA. NGS has revolutionized biosciences, enabling laboratories to perform a wide variety of applications and study biological systems at a level never before possible.[39]

NGS involves several major steps in sequencing. For example, DNA NGS involves DNA fragmentation, library preparation, parallel sequencing, bioinformatics analysis, and variant/mutation annotation and interpretation.

DNA fragmentation is used to break the targeted DNA into many short segments, usually 100–300 bp in length. Different methods can be used to achieve this. DNA can be fragmented using mechanical methods, enzymatic digestion, or other methods. For

example, sonication can be used to break DNA into short segments. The short segments relevant to the targeted DNA sequences are pulled out using specific complementary probes of different designs. This method is usually referred to as hybridization capture assay. Another method involves polymerase chain reaction (PCR) amplification. In this method, many pairs of primers are used to amplify the targeted DNA segments using PCR. The PCR products serve as short segments of targeted DNA. This method is usually called amplicon assay. The DNA segments are then used for library preparation.

Massive parallel sequencing is performed using an NGS sequencer. The library is uploaded onto a sequencing matrix in a certain sequencer. Different sequencers have different sequencing matrices. For example, Illumina NGS sequencer uses flow cells and Ion Torrent NGS sequencer uses sequencing chips. However, its goal is the same, which is to allow parallel sequencing of all the DNA segments at the same time. The sequence information generated from such massive parallel sequencing is analyzed using bioinformatics software.[40]

Bioinformatics analysis is a process involving base calling, read alignment as preprocessing, and mapping procedure. During this process, the sequence information is compared to a reference genome, a sequence to identify the exact location of the targeted sequences. All information from each sequenced segment is pieced together to generate final sequencing results for the full length of the targeted DNA. The final sequencing results are then used for further interpretation according to each specific sequencing protocol.

3.1.2 Reference genome

The reference genome is a digitized version of the haploid genome of an organism. More specifically whole sequencing data DNA from several donors are combined with algorithms so that the genome reference is the most common genome version among donors [41]. The most recent version of the genome was obtained for data analysis reference for Mus Musculus mm39 from the UCSC database.

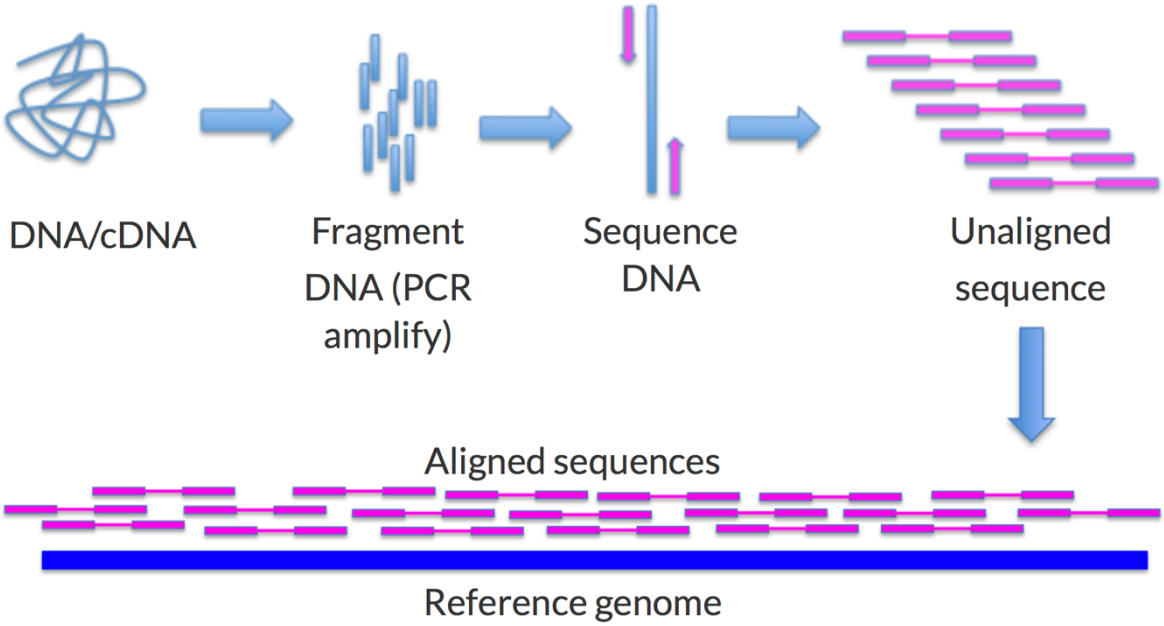


Figure 3-1. Generation of the reference genome.

3.1.3 Analysis of raw ChIP-seq data

For the analysis of ChIP-seq raw data aPEAch pipeline [42] was used to extract intervals of open chromatin for raw sequencing reads. As with any high-throughput experiment such as next-generation sequencing, a single assay is subject to considerable variability. For this reason, each cell type has two biological replicates (biological replicates) which measure the state of the chromatin, in order to limit the phenomenon of variability. In order to assess consistency between replicates, we need metrics that objectively assess the reproducibility of high-throughput assays. The sample processing algorithm uses the Irreproducibility Discovery Rate (IDR) test [43], which compares a pair of ordered lists of regions and assigns values that reflect their reproducibility. The basis of the assay's operation is that in two replicates measuring the same biological signal, the most significant regions, which are likely to be true signal, will have high consistency between replicates, while the less significant regions may be noise. In a list of ordered regions that contain important and unimportant the consistency varies. The change in consistency is an indicator of whether it is real or not a peak. Further analysis of open chromatin regions requires precision at the point with the highest signal, however the IDR assay gives a near-false peak after comparing the two replicates. It considers the area with the most signal in the merged from the two copies, the peak is in the middle of the two peaks of the peaks it consists of which is not arbitrary. As the peak is important for the continuation of the analysis, another algorithm was implemented in Python (version 3.9), which finds the real peak, based on the two copies. It uses the start-end coordinates of each merged

peak while to find the point of maximum signal it uses the alignments and measures the total coverage with reads per nucleotide from both copies.

Genome intervals that are identified to be statistically enriched at each condition are compared to find overlapping and unique regions among them to stratify the intervals into categories. Based on the hypothesis that the key differentiating factor among conditions is the binding sites of transcription factors that are unique to each category, overlapping regions are more probable to contain no significant biological background for the modeling process. Thus, unique to each condition genome intervals that are accessible are prompted to the dataset generation process, at which the sequence and the conservation score are retrieved, while the label encodes the accessibility of the interval. Custom code that leverages Bedtools [44] utilities performs the genome interval comparison to retrieve a file containing regions and their accessibility profile among classes, while for the dataset generation process the intervals are normalized for their length by minimizing or extending the window according to the summit of the ChIP- or ATAC- peak. Intervals are split to generate datasets based on user-defined input in three different manners, based on specific chromosomes, percentages of chromosomes, or percentages of the total dataset that is first shuffled to ensure evenly distributed intervals of each chromosome to the dataset splits in training, validation and test sets in percentages of 70-10-20 percentages respectively.

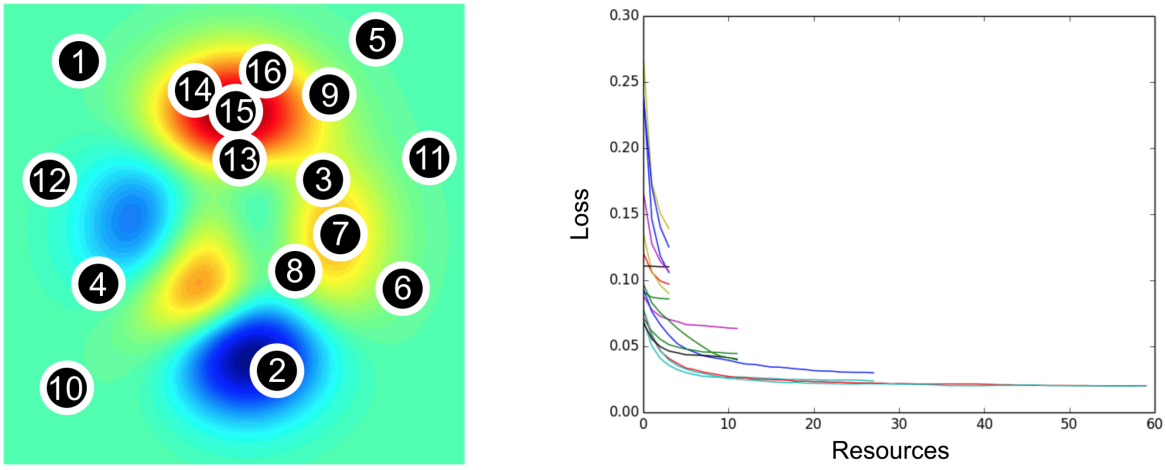
3.2 Model development

To captivate the underlying biology of the cooperative action of transcription factors on open chromatin regions, a convolutional neural network was developed using Keras and TensorFlow to model the differences among classes. The hypothesis supports that the differentiating factor among conditions is the effect of cell type-specific transcription factor binding that alters the gene expression, however, the action of specific transcription factors and the localization of patterns on the genome remains unknown. CNNs overcome this issue with their feature extraction modules, the convolutional layers, that are trainable and require no initialization. During training, the filter kernels are optimized to identify patterns in the input to captivate the differences in input between classes. The architecture of the model consists of two input branches, each containing three consecutive convolutional layers which are then flattened and concatenated into a feature vector. The feature vector is parsed to a fully connected branch consisting of one to three layers the output of which is an output layer of length equal to the classes. The convolutional layers consist of flexible length kernel size and kernel number, which is

determined during parameter selection, while as activation function the LeakyReLU has been utilized. For the fully connected branch, both the number of layers and the number of nodes of each layer is determined during parameter selection, while ReLU is being used as activation function. For the final output layer, the activation function is determined based on the modeling process, sigmoid or softmax if the modeling is multiclass or dual class respectively.

3.3 Parameter selection

A crucial step for the modeling process is to select the optimal parameters for kernel number and size, hidden layer configuration, learning rate, and batch size. To address this task two methods have been tested, grid search and hypermodel. During grid search, the training was performed on a nested loop to test various spectra for each parameter, and the best model was manually selected based on accuracy and loss curves and the absolute metric values. This procedure is both time-consuming and requires time to process all models that have been trained, introducing bias during model selection.



(a) Configuration Selection (b) Configuration Evaluation

Figure 3-2. (a) The heatmap shows the validation error over a two-dimensional search space with red corresponding to areas with lower validation error, (b) The plot shows the validation error as a function of the resources (epochs) allocated to each configuration.

To overcome this issue, the CNN is now coded as hypermodel and the hyperparameters as the number and size of filters, the number of neurons in the hidden layers but also the learning rate are initialized as a range of values, a searching space, that hyperband model

uses to identify the best combination during training. Hyperband [45] is a hyperparameter optimization algorithm that works by efficiently allocating resources across many hyperparameter configurations. It starts by randomly sampling many configurations and then runs each configuration with a small number of epochs. After that, it evaluates their performance and eliminates the worst-performing ones, keeping only the promising configurations. The remaining configurations are given more resources in subsequent rounds. This process, called successive halving, repeats until it narrows down to the best-performing configurations. Hyperband balances between exploring many different configurations and exploiting the best ones by progressively allocating more resources to the top-performing candidates. It's designed to save time and computational power by quickly discarding bad configurations and focusing on better ones.

3.4 Interpretability modules

The main objective of the proposed method does not rely on the prediction of newly fed sequences to the model for their accessibility between cell states, as the model has already been trained to the genome-wide profile of the TCF-1 accessible intervals. As the network has been trained and tested for its modeling capabilities, the significant biological insights that accompany TCF1 binding in different contexts has already been stored to the network weights, as it is capable of distinguishing plain sequences for their association to either of the cell states. To translate the knowledge acquired from the model, a set of post hoc interpretation modules have been developed. The interpretation modules rely mainly on the convolutional layers, that extract the differentiating features (motifs) among sequences and provide insights by annotating, evaluating their importance, the association to the different classes and co-activation status. This set of modules provides both a clear vision on the features that the model has learned through training and biological information regarding the biological system that has been through this modeling procedure.

3.4.1 Filter visualization

Filter visualization is a common step in model interpretability that is widely used in computer vision, to extract and study the features that a model has learned during training [46]. This step is performed by visualizing the filter kernels of the first convolutional layer, which captivates more simple characteristics of the input data and is easy for the user to

perceive. Further investigation in deeper layers requires more robust methods of interpretability and has not been applied to this study. Even though Kernel Visualization is a common procedure both in computer vision and in relevant studies that CNNs have been trained on DNA sequences, the exact method is not well standardized. Therefore, for the case of this study, four methods have been implemented and tested for their robustness.

3.4.1.1 Top activations method

The first method used to visualize the filter kernels is based on the activation of filters on subsequences of the input sequences. The first convolutional layer is isolated from the model and a forward pass of the test set is performed to calculate activation values for each filter and subsequence. As the layer includes leaky ReLU activation function, the values are 0 or close to zero for non-activating subsequences, while higher positive activation values indicate that subsequences are closer to the feature that the filter has learned. Therefore, for the visualization process, only highly activating subsequences for each filter are parsed to the next step to generate position weight matrices. Subsequences capable of activating each filter have the same length as the filter kernel and are used to calculate nucleotide frequencies for each position of the kernel. For each filter and at each position of the kernel, the nucleotide frequency is calculated by counting the number of distinct nucleotide instances and then dividing by the total number of the subsequences, generating an Nx4 matrix, where N is the length of the kernel and the sum of each column is 1.[37]

3.4.1.2 Generation of positive activating sequences

This method aims to avoid the use of the test set, as it could lack activating subsequences for some of the filters. This could be a result of an unbalanced split on the training and test set or due to the low number of instances of some features. The rationale is that based on the kernel values for each nucleotide, only positive values for each nucleotide and position in the kernel could result in highly positive activating subsequences. For each filter and position within the kernel if there is a positive weight for one or more nucleotides, they are stored, while in case where no positive weight is present for a position of the kernel, all nucleotides are stored to maintain the length of the kernel. Based

on the nucleotides that have positive weight, all possible subsequences are created to be used later for the generation of position frequency matrices as described above.

3.4.1.3 SOFTMAX method

This methodology is inspired by the above-described method which is based on generating subsequences to tackle the same issues of the filter visualization step. Rather than generating filter-activating subsequences, the kernel of each filter is normalized with a softmax function at each position to translate filter weights into probabilities. The positional probabilities for each filter mimic the method of generating position frequency matrices and are used to visualize filters as the sum of each position for all nucleotides sum to 1.

3.4.1.4 Randomization method

This method is used to assess the significance of each combination of filter and subsequence within the test set. Similar to the top activation method, a forward pass on the test set is performed on the isolated first convolutional layer to calculate the activation of each combination of filter and test set subsequence. Subsequently, in an iterative manner, the filter weights are randomized for new weights considering the mean and the standard deviation for each filter and position to create similar but random filters. Then a forward pass on the test set is performed to retrieve the activation values of the randomized filters on the same subsequences while in case the randomized activation is greater than the original one this instance is kept. This procedure is repeated for 100.000 times to ensure a valid sample size for the randomization process. The number of instances where the randomized activation for each subsequence and filter is greater than the original is divided by the total number of trials to calculate the significance level of each combination. The null hypothesis supports that those combinations are random, every similar kernel could provide greater activation than the original set, while if the combination is unique, it will not appear often during the randomization process. The calculated p-values for each combination are then grouped by filter and each filter is sorted in a descending manner an algorithm for elbow point detection is used to apply the p-value cutoff for each filter. This step is crucial, as the number of iterations does not allow the correction for false discovery rate. For each filter, the subsequences that do not overpass the p-value cutoff are used to create position frequency matrices.

3.4.1.5 MOTIF comparison to JAPSAR database

To relate the motifs acquired from the visualization step to transcription factors, a necessary step is to compare them with experimentally confirmed patterns of transcription factors. PFMs generated by any of the above-described methods are compared to the JASPAR 2022[47], [48], [49] database, which contains a curated, non-redundant set of profiles, derived from published collections of experimentally defined transcription factor binding sites for eukaryotes. This step is performed leveraging TOMTOM [50] from MEME [51]suite, an algorithm that compares one or more motifs against a database of known motifs. TOMTOM creates matches between query and database motifs, performs the alignment and reports the significance level of each pair. Based on this procedure, each filter is annotated if there is one or more matches to the database and is related to (a set) of transcription factors, providing insights into which transcription factor motifs are important for the model to differentiate sequences during the modeling process.

3.4.2 Filter Importance

Filter importance is crucial for evaluating the significance of each learned feature from the model on the decision-making. This implementation is inspired by computer vision field, where similar approach is used to assess the importance of filters, by iteratively randomizing each filter at a time and then evaluating the model. More specifically, the model is first evaluated on the test set, and the loss is calculated and stored as the baseline. Then, each filter at a time is randomized while keeping the same standard deviation and mean of the filter weights to keep similar anticipating output of the first convolutional layer and not perturb further the model dynamics, and the loss is calculated. This procedure for each filter is performed 200 times to evaluate the importance on a wide spectrum of trials. The loss for each filter is projected on a boxplot which is ordered in a descending manner based on the loss mean, while the red dotted line on the y-axis stands for the baseline model. It is anticipated that filters that are important for the decision-making of the model, will perturb to a greater extent the loss of the model, meaning that the model will make more false decisions than perturbing a less important filter. In the latter case, the distribution of the loss is anticipated to be close to the baseline model.

3.4.3 Filter clustering

3.4.3.1 Hierarchical clustering method

Filter clustering aims on extracting co-activation patterns between filters among the sequences of the dataset. The activation map of the first convolutional layer is transformed into a two-dimensional array where rows represent test set inputs while columns represent filters. The combination of test set input and filter is a normalized activation value calculated as the mean activation of each filter on the sequence vector. The normalized activation provides insights into the uniqueness of the filter, as if the kernel does not identify specific patterns, it will activate across the sequence vector providing high average value column-wise. If a motif is abundant in a specific sequence but not in others, the normalized activation value will be differentially high on test set examples providing a different activation profile across the test set. The activation array is then transformed with the min-max normalization method to avoid parsing bias to the clustering step. The filters of the model are then clustered using Euclidean distance as a dissimilarity measure and complete link as clustering algorithm, using as data vector the activation map across test set for each filter. Clustering results are visualized in a circular hierarchy plot representing the higher clustering of the filters but also forming close clusters for those filters that have similar activation patterns, providing insights into the presence of cooperativity.

3.4.4 Filter enrichment

Filter enrichment module is performed for the filters of the first convolutional layer, leveraging the significant activated TFBS from the filter visualization step to identify whether a filter is enriched activated in one of the classes of the model. Significantly activated TFBS are annotated based on the label of the sequence for their association to one of the classes, using the same cutoff of p-value as in visualization. Then a contingency matrix for both NIH3T3 and DP is calculated based on the categories: A) activated TFBS in class X, B) not activated TFBS of class X, C) activated TFBS not in class X, D) not activated and not in class X. (X = NIH3T3 or DP). This contingency matrix is used to perform enrichment analysis with Fisher's exact test [52] and therefore compute the odds ratio of each filter being activated in a specific class and the significance level. Only significant trials (p-value <= 0.05) are used to plot the odds ratio (transformed with log()) to have a clear view on the filter enrichment among classes.

4. Results

4.1 Dataset preprocessing and generation

Datasets were derived from GEO and processed to retrieve the genome-wide areas of TCF1 binding across cell types. Raw sequencing data has undergone quality assessment and adapter trimming, with adapters found and removed in the Double Positive datasets, while in NIH3T3 raw reads were already sequencing adapter-free. Reads that passed the quality standards were then used for alignment, and for both of the datasets resulted in a high percentage of uniquely mapped reads (Table 1). Aligned reads were processed for coverage with MACS2 [53], including the control samples, to retrieve the intervals-peaks of TCF1 binding in both cases and specifically in NIH3T3 derived peaks the two replicates were tested for reproducibility. The two lists of intervals were compared to find the unique peaks per class and were normalized for their length according to the peak summit, to generate equal peaks 400 in length.

Table 4-1: ChIP-seq preprocessing and mapping metrics.

Sample Name	% Trimmed	Median Read Length	% Dups	% GC	% Aligned
DP_TCF1	0.2%	32 bp	30.7%	45%	76.8%
DP_input	0.2%	32 bp	13.8%	43%	77.3%
NIH3T3_TCF1_rep1	0.0%	74 bp	13.1%	44%	80.8%
NIH3T3_TCF1_rep2	0.0%	74 bp	11.9%	44%	84.6%
NIH3T3_empty_RV_repl1	0.0%	74 bp	11.8%	44%	80.4%
NIH3T3_empty_RV_repl2	0.0%	74 bp	13.4%	43%	79.2%

Unique intervals were processed with the data generation algorithm described above, to generate the model dataset. Each of the sequences was transformed into (4,400) NumPy arrays, while the label was represented by 0 for NIH3T3 and 1 for Double positive T-cells according to the accessibility of the binding site in either of the cell types. The resulting

35,536 uniquely accessible TCF1 binding sites for each class, that are split to 11,434 for Double positive and 24,102 for NIH3T3 were further split to generate the training validation and test set (Table 2).

Table 4-2: Training, Test and Validation of the unique peaks of NIH3T3 and DP samples.

	Training set	Test set	Validation set
NIH3T3	15,475	4,806	3,821
DP	7,269	2,301	1,864
Total	22,744	7,107	5,685

4.2 Hyperparameter tuning

Hyperparameter tuning was performed on the model using hyperband, a random-based method that implements successive halving to narrow down the best configurations. This procedure minimized the time for hyperparameter selection from weeks-scale using grid search to almost five hours with the hyperband implementation. As shown in Figure 4-1 the tuner performed an extensive search for the best combination of number and length of filters for the convolutional layers and number of nodes for the hidden layers, providing a list of models and their training history. Most of the results on the figure are not comparable, due to the learning rate and batch size differences, as both were part of the hyperparameter selection list. As the final model for the study, hyperband selected 35 filters with length 14 for the first convolutional layer, 21 filters with length 11 for the second convolutional layer and 11 filters with length 6 for the third and final layer of the feature extraction module of the model. The output of the convolutional module is then flattened and parsed to two hidden layers consisting of 100 and 80 neurons respectively. The

selected model architecture had balanced accuracy of 0.7968 and 0.7777 for the training and validation set respectively.

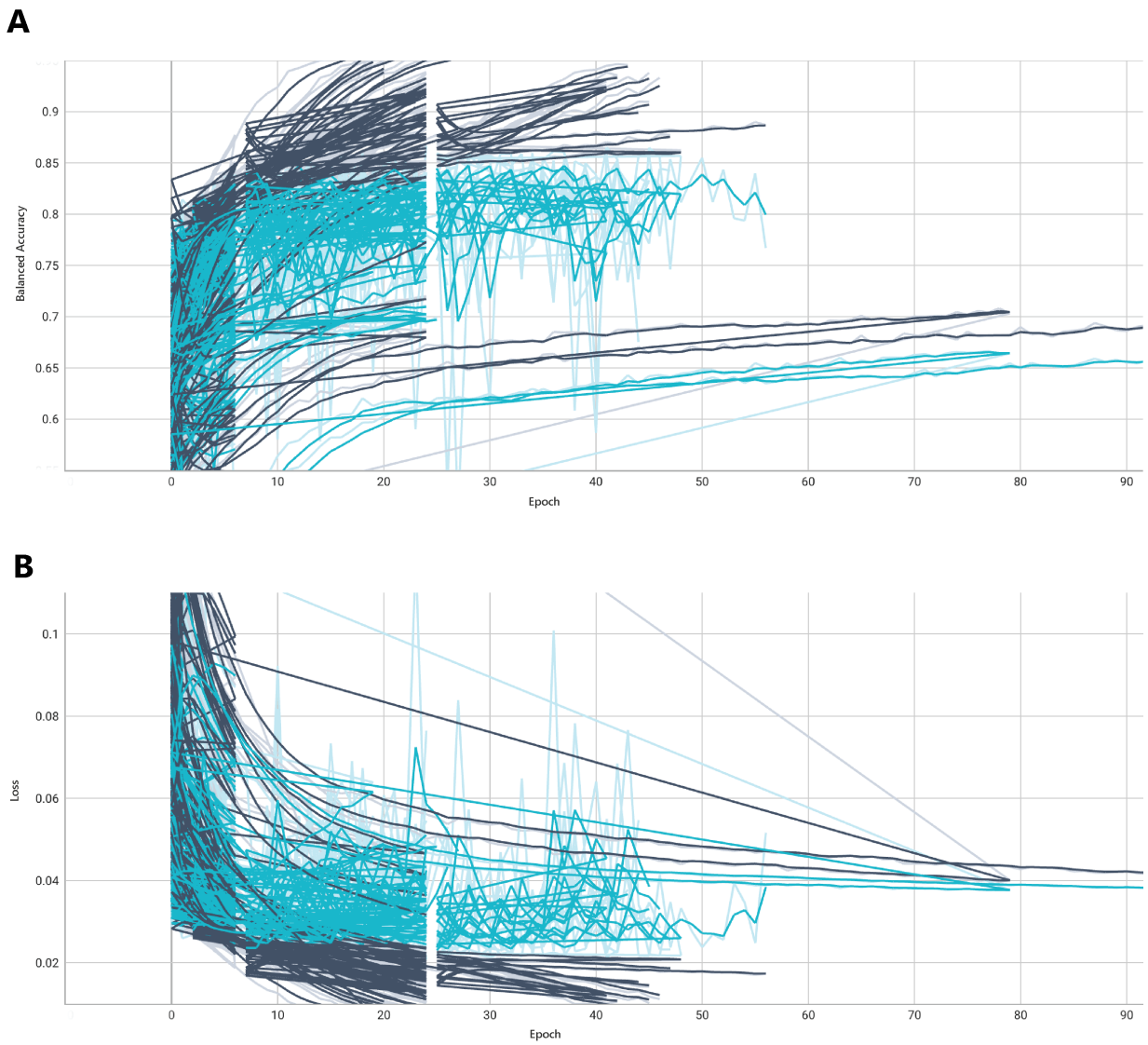


Figure 4-1. Trials of successive halving of Hyper Band and the representation of (A) balanced accuracy over epoch, (B) loss over epoch.

During the hyperparameter selection, 160 trials of different parameters for convolutional layer filter number and length supported that those parameters were not important enough, as at any combination the training resulted at least to a well-performing model with respect to the balanced accuracy. Even though most of the parameters produced good-performing models, those with shallow first convolutional layer, having 10 to 20 filters at most cases resulted in poor-performing models with an accuracy below 0.60. Additive to that, the length of the third convolutional layer's filters seemed to have impact

on the performance, suggesting that models with longer third convolutional layer length, >10, were more likely to result in poor-performing models.

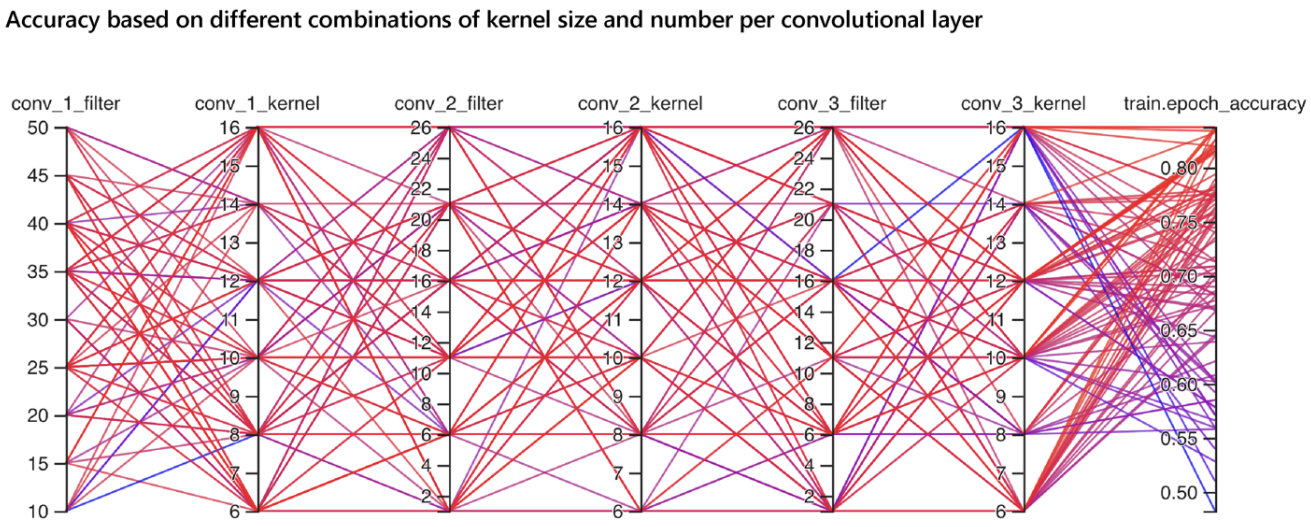


Figure 4-2. Parallel graph of hyperparameter combinations and the resulting balanced accuracy on the converged model during hyperparameter selection. (kernel refers to kernel length and filter to the number of filters).

4.3 Model training

After the parameter selection step, the final model configuration was again initialized and trained to find the optimal batch size and learning rate. This training process resulted in an even better-performing model with a balanced accuracy of 0.8287 and 0.8096 for training and validation set respectively. Beside training, a model snapshot was saved after each iteration, which was then used for benchmarking the interpretability modules. As shown in Figure 4-3 the selected model snapshots belong to different states on the training process and potentially could provide a range of results for each interpretability module and also insights on the training progression. The selected models are categorized based on the phase of the training that they belong to “starting” after one epoch of training, “early” post 5 epochs of training, “mid” after 50 epochs, and the “final” one that the model has converged.

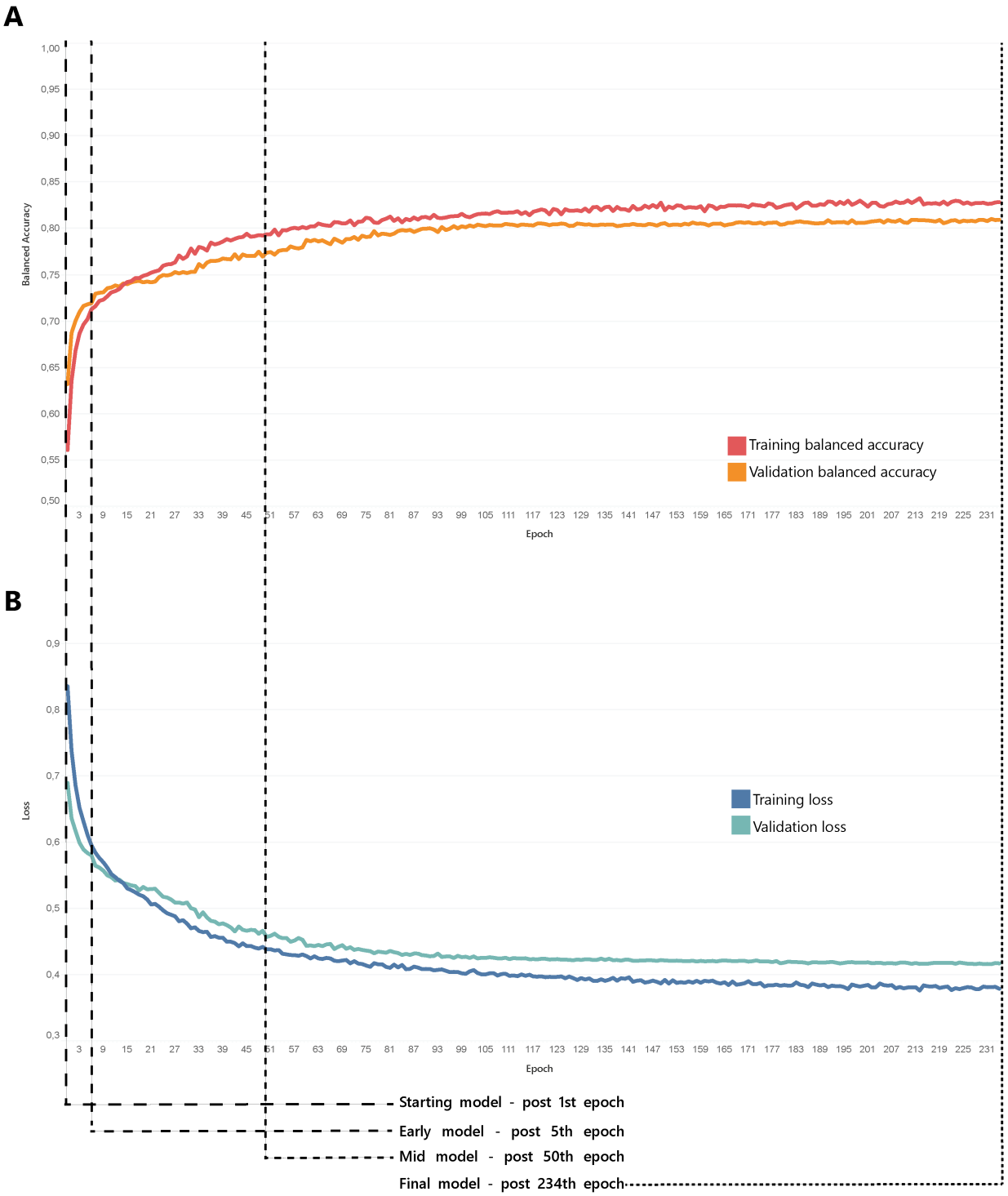
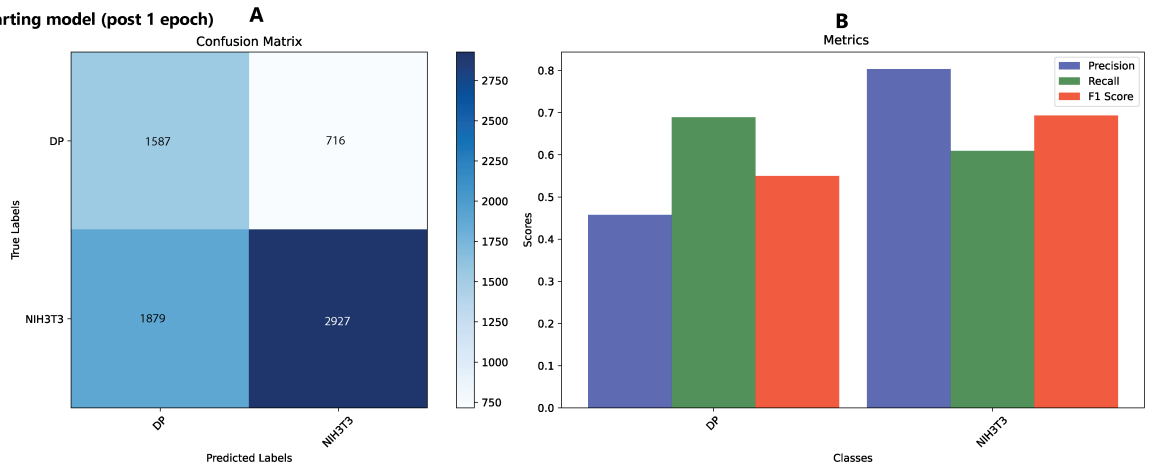


Figure 4-3. Training and validations Balanced Accuracy (A) and Loss (B) over epoch. The dotted lines represent the model snapshots.

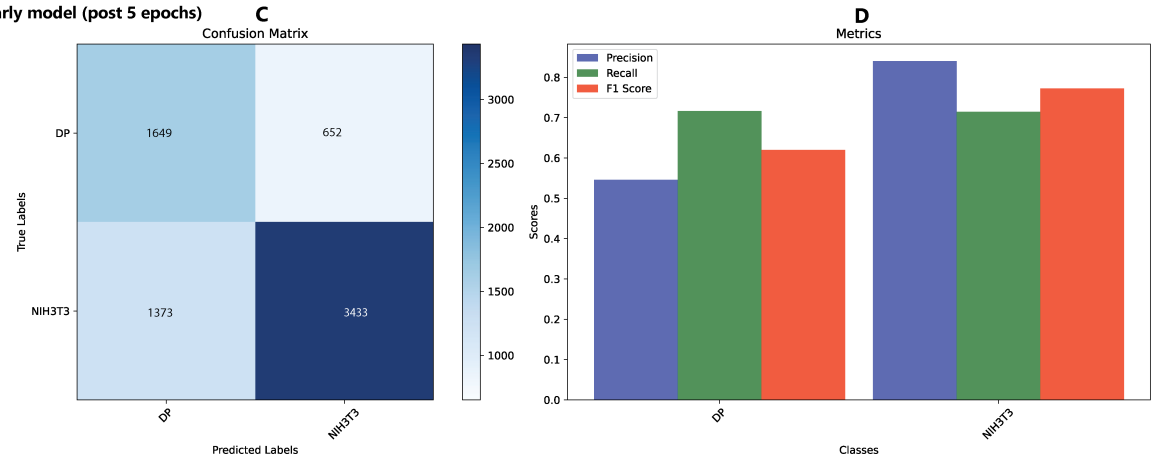
On Figure 4-4 is displayed a more detailed view of each of the four models' metrics. Starting model is close to random, with a large number of false negatives in each of the classes, while as the training progresses the number of false positives is minimized in the final model. The mid model is performing close enough to the final, with the number of

false positives being proportional to the size of each class and the recall being close for the two classes. The large number though in misclassifications is anticipated, as this type of data is often noisy due to the underlying biology. For example, two sequences may contain the same binding sites, and therefore the same features from the model, but being differentially accessible due to chromatin modifications, making more complex the decision making and resulting in misclassifications. Another interpretation could be the presence of the TCF1 binding motif only, preventing the model from extracting any other meaningful features from the sequence.

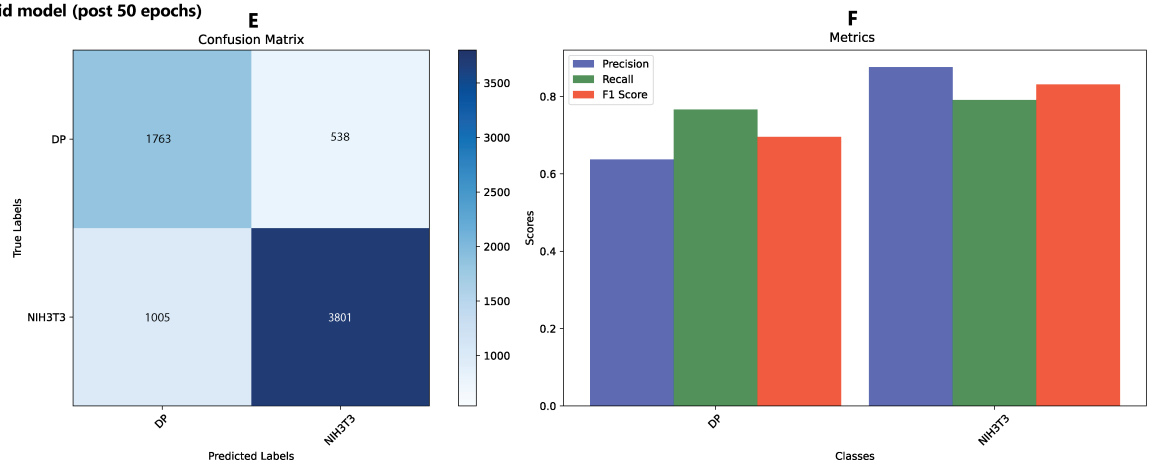
Starting model (post 1 epoch)



Early model (post 5 epochs)



Mid model (post 50 epochs)



Final model (post 234 epochs)

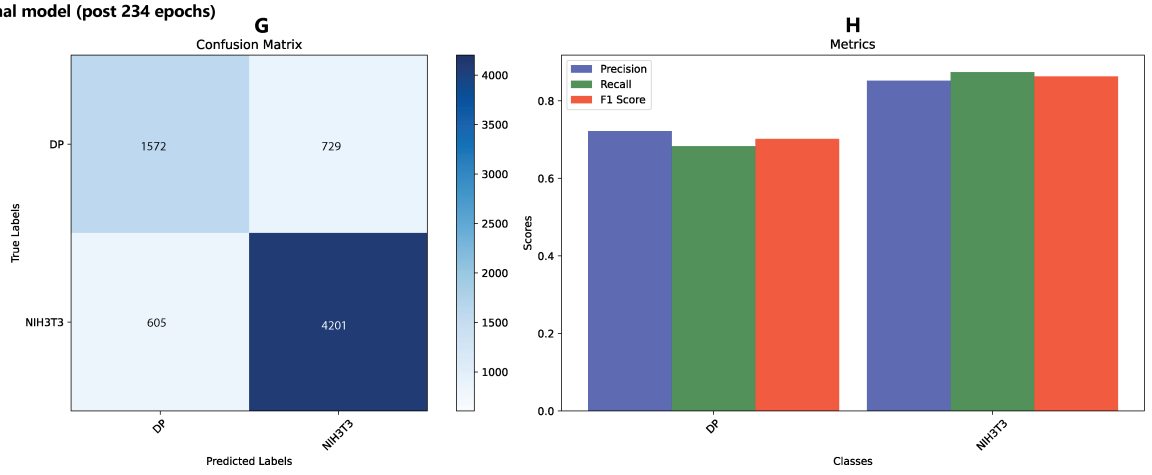


Figure 4-4. Confusion matrix for each of the selected snapshot models in panel (A), (C), (E), and (G) and evaluation metrics on (B), (D), (F), (H) for the starting, early, mid and final model respectively.

4.4 XAI modules

4.4.1 Filter visualization

4.4.1.1 Method testing

To evaluate the best-performing method for filter visualization, the four approaches have been tested on the final and well-performing model. This module is crucial for the annotation of the filters by comparing them to known TF motifs binding and therefore with the TFs. All methods provided significantly different results, from low quality to strict motifs, but the best performing were the top activation method and randomization. The Softmax method does not require input and, therefore provide faster results. Although all the motifs were degenerate (Figure 4-5) and could not be associated with any of the known TF binding motifs from Jaspar. Contrary to the softmax approach, the generation of positive activating sequences leads to more strict motifs, duo to the predetermined nucleotide per position of the motif, and could associate only three motifs out of 35 to known TFs. Top activating sequences generated motifs that are close to known motifs, with positions being either degenerate or have a strong favor of some specific nucleotides. However, this implementation has two major drawbacks, requires a user-defined threshold for filtering the top activations, and during trials, a minor change to this threshold affected the generated motifs. The randomization method provided solid motifs as those of the top activation method even though it requires a threshold for significance level. Due to the small number of randomization trials, the threshold was applied to p-value = 0.005, to minimize the false positive TFBS that were parsed to the motif generation step. This threshold though even with 10-fold changes did not change drastically the motifs' both appearance and cohesion to the known TF binding motifs. Therefore, the randomization method was selected as the best-performing filter visualization method and was tested during training.

Table 4-3: Number of filters generated by method, and matched to known TF motifs from Jaspar

method	Softmax	Positive activating sequences	Top activations	Randomization
Filters generated	35	35	35	35
Filters matched	0	4	9	10

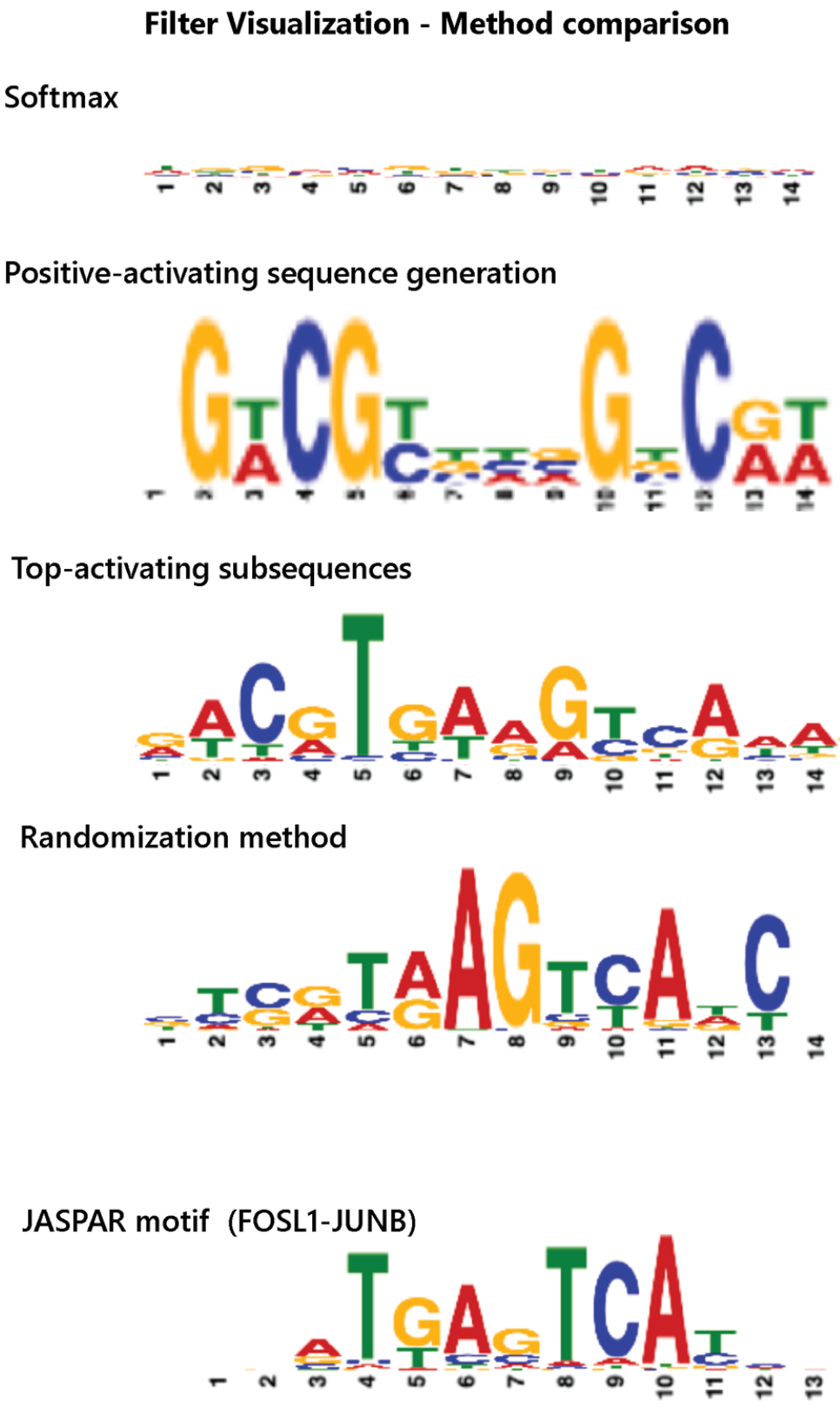


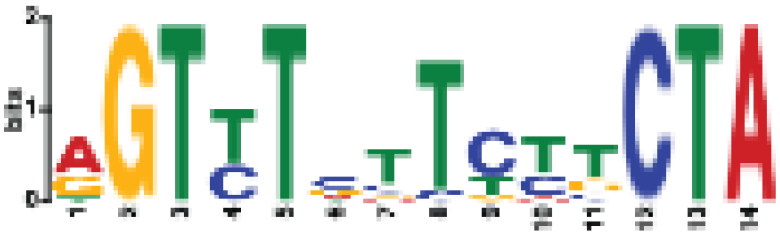
Figure 4-5. Comparison of visualization methods on the final model for filter 29.

4.4.1.2 Training progression results

As explained before, the randomization method was the best-performing filter visualization approach on the final model. An important step for the explainability of the model is to assess the training of the filters and how those are structured. Feature motifs were generated using the randomization method on the four model snapshots described above to evaluate the training process of each of the features. A significant insight into the training procedure posited that most motifs have been formed even after the first epoch of training, with small but crucial changes (Figure 4-6) during the rest epoch to form the final feature motifs. This finding is also strengthened based on the results of the enrichment module as described below.

Filter Visualization - Filter training

Filter 22 - Starting model



Filter 22 - Early model

Filter 22 - Mid model

Filter 22 - Final model

JASPAR motif (STAT1::STAT2)

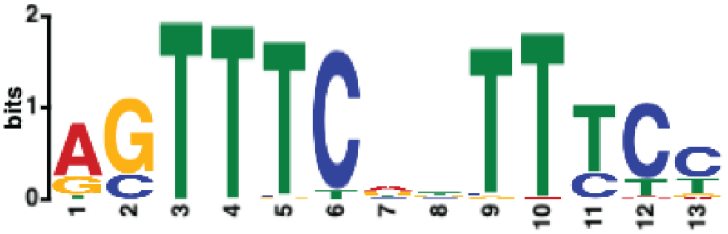


Figure 4-6. Training progression of filter 22, implementing randomization method for visualization. Also displayed is the STAT1::STAT2 complex, the Jaspar match.

4.4.2 Filter importance based on model perturbations

4.4.2.1 Training progression results

Filter importance module was run on the four model snapshots to evaluate the model's capability of learning meaningful representations on the input. As described in the methods section, importance is calculated by eliminating one filter at a time and then followed by measurement of the loss on the test set. Even though this procedure was aiming purely on ranking filters' importance, the application of the module on the early stages of the training provided also insights into the training procedure itself.

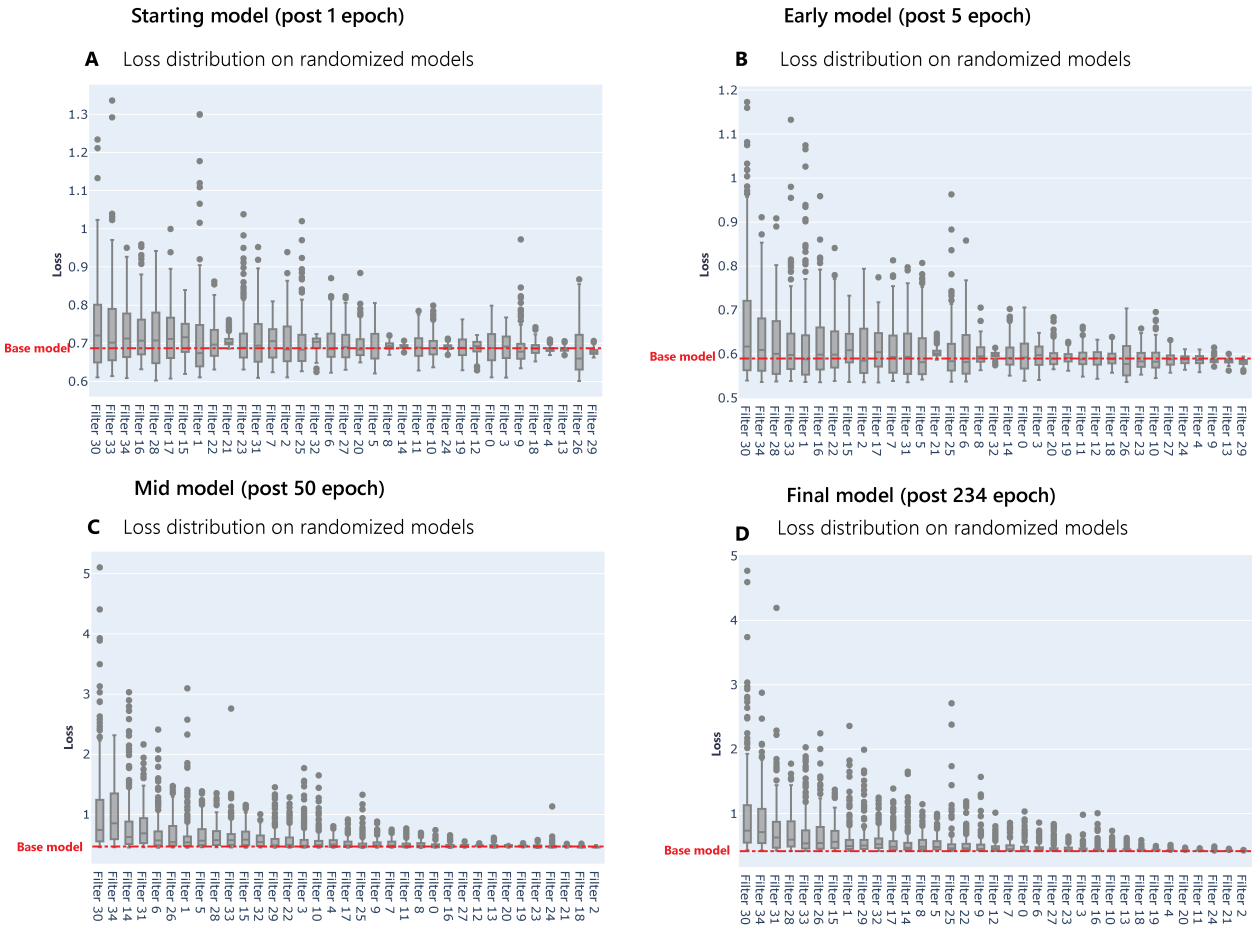


Figure 4-7. Distribution of loss for the perturbed models, indicating the effect of the filter elimination on each of the four model snapshots.

As the hypothesis posed, the filter elimination process indicated a decreasing impact on the model's loss after the randomization of each filter on the final model, supporting that filters have different impact on the model's performance (Figure 4-7-D). On the converged model, all trials including filter randomization produced a worse-performing model based on loss with respect to the baseline model. The impact on the performance is divergent

among filters, indicating differences in the importance of each formed filter, and spans from a high level of model degeneration, thus more important, to even zero impact for filters that have not learned a meaningful representation for the classification task. On the mid model, filters seem to have similar influence on the model (Figure 4-7-C), but the hierarchy is not as well defined as in the final model. The application of the importance module on the starting and early models concluded in a different behavior regarding both the hierarchy and the perturbed model's performance. Hierarchy as expected was not well defined (Figure 4-7-A, B), although the ranking remains similar to the final model's results, at least for the most important ones. Performance-wise, starting and early models that have been perturbed during the randomization process resulted in some iterations in better models than the baseline model with respect to the loss, while filter visualization highlighted that most filters learn representations of the input from the first even epoch and do not alter significantly during training. The combination of those two insights could indicate that features have been successfully extracted even from the first epoch, and during the rest of the training procedure, the classifier is optimized to associate them to the output label.

4.4.3 Filter enrichment to model classes

4.4.3.1 Training progression results

Filter enrichment among classes served as a measure of association for the learned features, to either of the classes (Figure 4-8). This should indicate filters that are being activated more to some or both classes and therefore meaning specific transcription factor activity. For the stronger associated filters to either of the classes, log odds ratio > 0.3, the filters remain enriched to the same class, while for poorly associated filters, their enrichment is more fluid and changes during training, as the kernels are finetuned to the modeling problem. Out of 35 filters, 28 are significantly enriched (Fisher's exact test p value < 0.05) to either of the classes in all four tested snapshot models, while the remaining were not significantly associated. An indicative insight into the training procedure is the association of all 28 filters to either of the classes, as the models try to identify motifs in the input sequences to distinguish them to their respective classes.

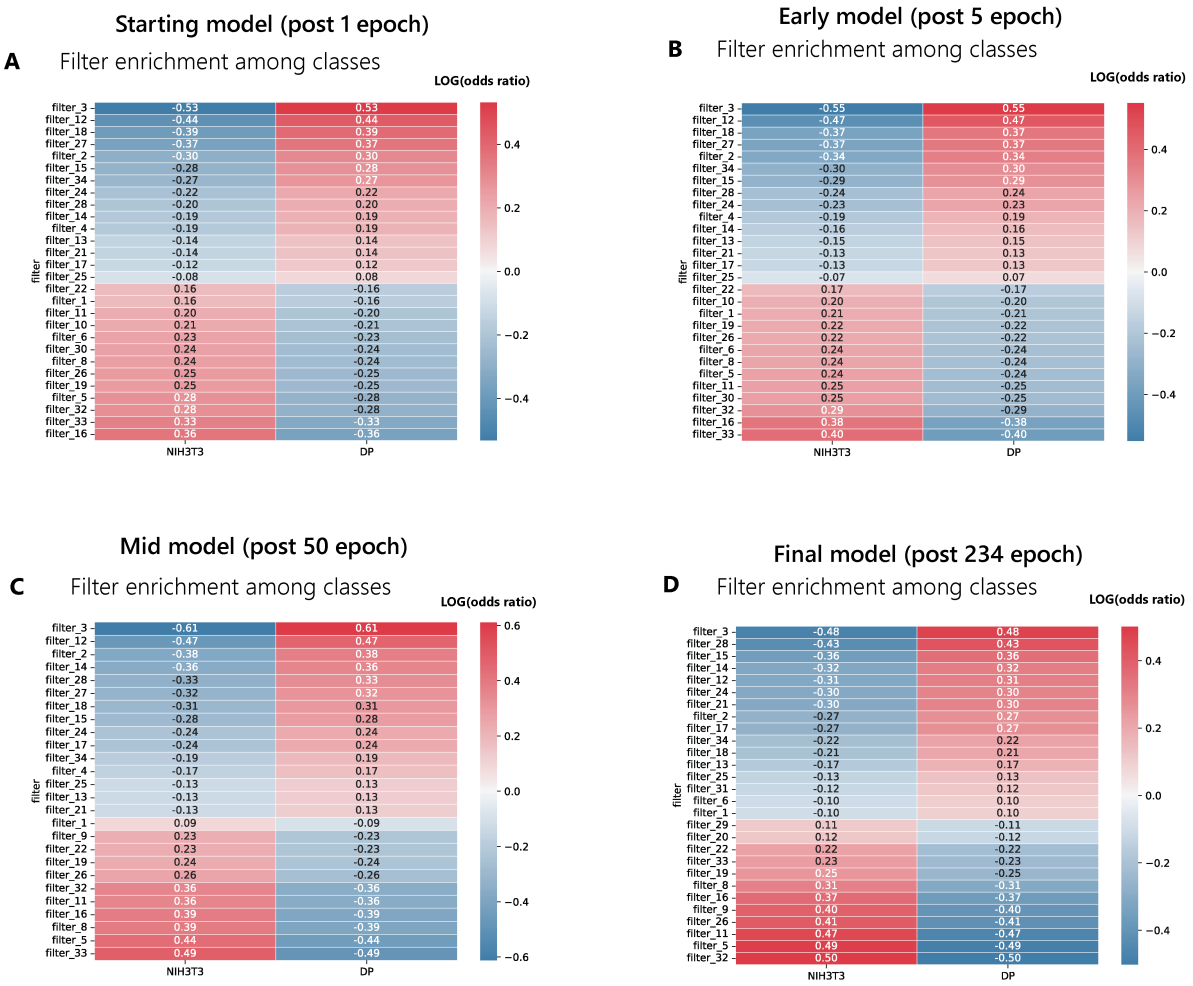


Figure 4-8. Statistically enriched (p-value < 0.05) filters in each cell type by model. Each cell represents the log(Odds ratio). Red color indicates enrichment, while blue indicates no enrichment to the class.

4.4.4 Clustering of filters based on activation profile

4.4.4.1 Training progression results

Clustering of filters was performed to identify coactivation patterns of filters, and therefore infer transcription factor dynamics. Hierarchical clustering was applied on the activation profile as described in the methods section for all four model snapshots as an approach to visualize how those dynamics are altered during training (Figure 4-9). This step is interconnected to the first convolutional layer only, as is filter visualization and enrichment, and is anticipated to provide similar general insights for the training process. Indeed, on each of the clustering results, there are formed four big clusters and the membership of each cluster is not altered significantly during training, as each formed

cluster maintain nearly the same filters with some minor alterations that are expected as the features learned are fine-tuned during the training procedure.

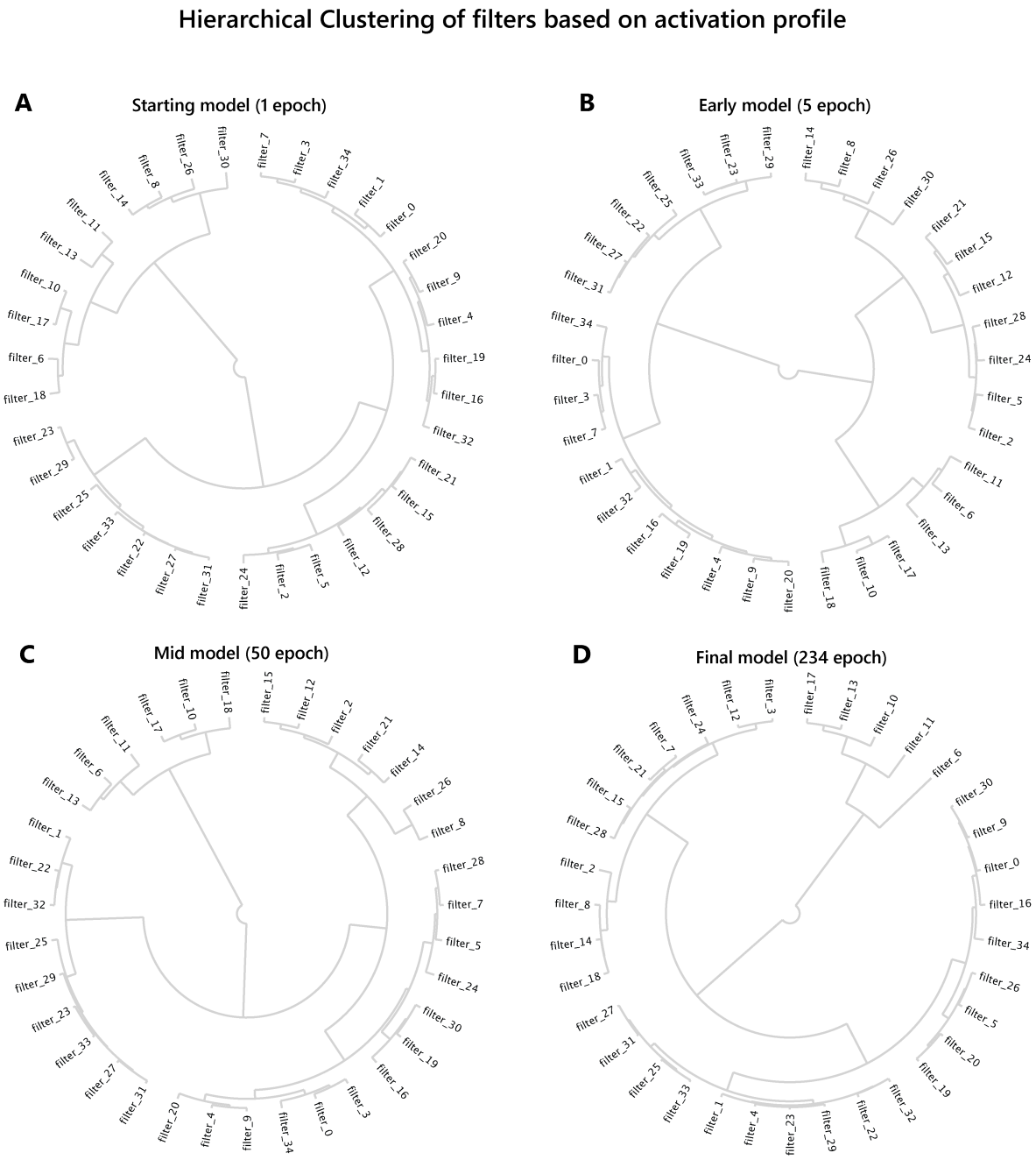


Figure 4-9. Hierarchical clustering of filters based on the normalized activation values across the test set.

4.4.5 Holistic view of the model-acquired knowledge

During training of the model to identify whether a sequence that TCF1 binds uniquely on either double positive T cells or NIH3T3 fibroblasts, the model has learned background biological information regarding TCF1 cooperators. Explainable modules provide insights

into how important those cooperators are for the model's decision-making, which they are, and how the motifs that those TFs bind are enriched in either class. Individual modules' results provide valuable information on both the biological background captured during the modeling process and the model's decision-making itself, yet a holistic view of the acquired knowledge remains crucial. To combine the extracted knowledge a circular plot has been crafted with the dynamics of the learned representations, TF motifs, as template. Each of the filters is then annotated on whether they are enriched to one of the classes or not (p-value> 0.05) and the motif derived from the visualization module. For those filters that an association has been identified by TOMTOM during motif comparison, the most similar transcription factor binding motif has been used to relate the filter to a transcription factor. This ensemble representation (Figure 4-10) provides a clear view of the learned motifs, how they are associated with transcription factors, whether each of the filters is enriched to a class, and the dynamic activation of those motifs, as it is crafted on the hierarchy.

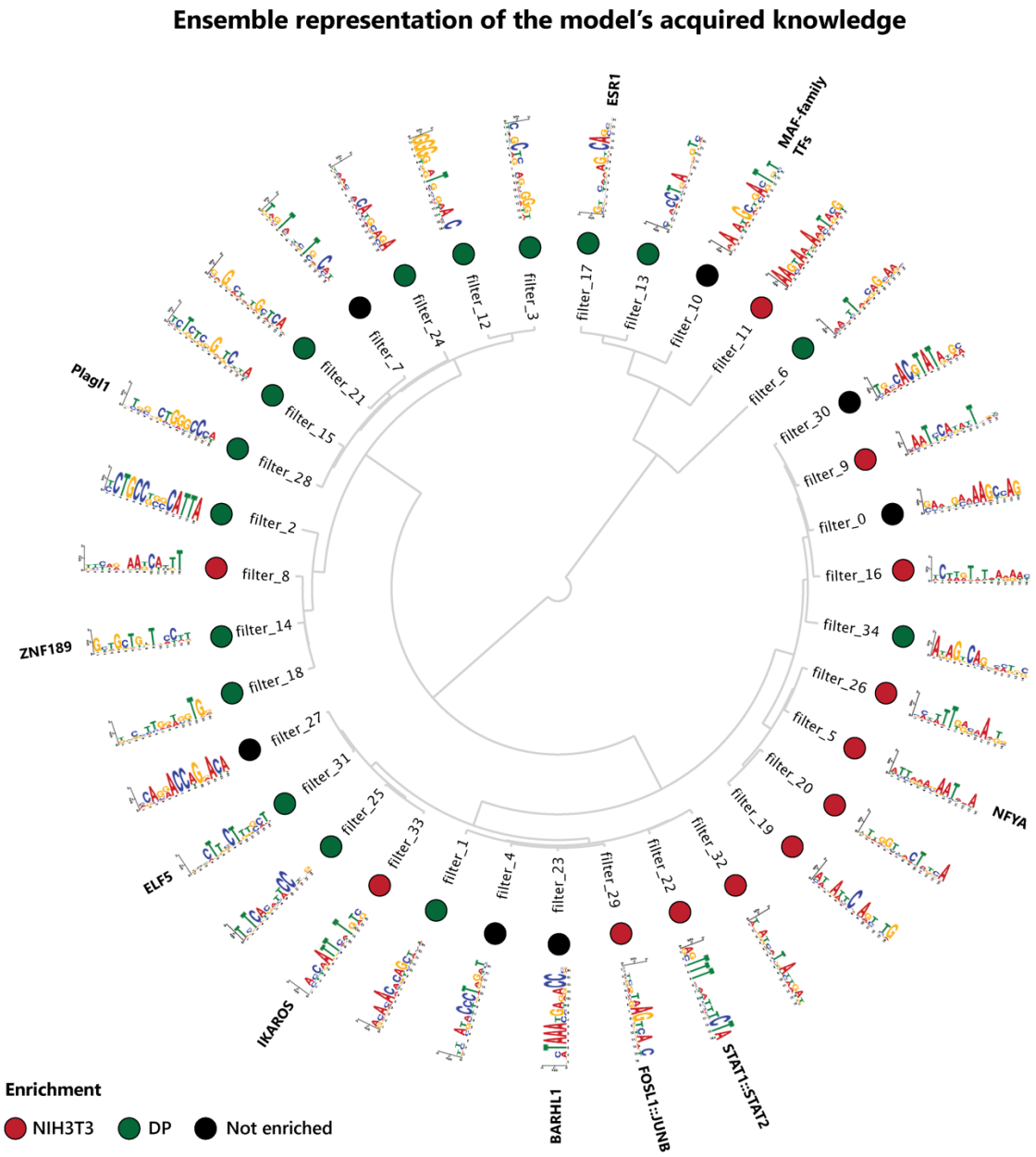


Figure 4-10. Ensemble representation of the model's acquired knowledge. Hierarchical clustering of filters infers the filter dynamics. Each filter is annotated for enrichment in classes: red – enrichment to NIH3T3, green – enrichment to Double positive T-cells, and black – not enriched. Further annotation of filters includes the visualization with randomization method and also projected the best matches of each filter to TFs based on TOMTOM results.

5. Conclusion

5.1 Genomic modelling provides insights on the TF dynamics

During the study of differential accessible binding sites of TCF1 in double positive T-cells and NIH3T3 fibroblasts leveraging the modeling capabilities of CNN models, the final goal was to decipher TF dynamics captured in the model's trainable parameters, weights, and biases. To achieve this, a well-trained model that classified sequences for their

accessibility in either cell type was created, while performing relatively well, considering the noisy nature of ChIP-seq data, meaning that it captured biological background in TCF1 binding regions that differ in the two classes. This comprehensive model is used for further examination, to extract biological relevant insights regarding the TF activity, leveraging the explainability modules. Indeed, filter visualization provided insights into the TFs implicated in either of the classes, by comparing the learned motif to experimentally derived TF motifs. IKAROS [54] and the complex FOSL1::JUNB [55] were among the anticipated findings, as they have, combined with TCF1, a dominant role in the t-cell development process, while also general transcription factors were detected, namely ZNF189 and MAF transcription factor family. The visualization of filters supports the hypothesis that the model has identified motifs of TFs and therefore the rest of the modules provide an enhanced view of their association to the model's decision-making and the classes themselves. Clustering of the filter activations provides a clear view of the TF interactions and dynamics, as the motifs have shown that inter-cluster members do not have close motifs, and each motif is representative to different sequence feature. The enrichment process, annotated the filters within the clusters, associating them to either of the classes. This annotation process resulted in two out of the four clusters being mainly formed from filters that are activated statistically more to either of the classes and therefore better describe the TF-activity schema of the examined cell types surrounding TCF1 binding sites. The importance assessment of the filters prioritized the learned features based on their impact on the decision-making of the model, while supported our understanding on the training process.

5.2 Training progression

The developed explainable modules extract the biological background concerning the TCF1 binding, while enforcing our understanding regarding the training process. The results of the modules on the four model snapshots have a very different development, based on the model sections used during interpretability. Filter visualization, enrichment, and clustering are limited to the first convolutional layer while filter importance accounts the entire model. The layer-interpretation modules have very similar results, even from the first epoch implying solid feature extraction that is made early in the training process with minor but critical optimization of the kernels for the rest of the training, as the filter visualization has shown. Considering the results of the layer-based methods above, the filter importance module provided a clearer view of how the model associated the learned features to classes. Trying to decipher the development of the filter importance results, the entangled view on the perturbed model's performance, in combination with the sold

feature extraction from the very first step, supports that the model performs feature extraction before associating them to classes, as anticipated, but most of the training procedure concerns the classifier training with minor adjustments to the feature extraction module of the model.

5.3 Strengths and limitations

Deep learning approaches for the modeling of transcription factor binding, such as BASSET and SATORI, have highlighted the capabilities of such models to capture the binding motifs of TFs and their dynamics. The application of this methodology through the study resulted in a relatively good performance for the prediction of accessible sequence that TCF1 binds between classes. Most methodologies that try to infer transcription factor dynamics through genomic modeling are restricted to filter visualization and the extraction of dynamics but do not provide a more comprehensive view on the model's decision-making process. In this study the novel approach on explainability lies in all the modules, as filter importance, clustering and enrichment have not yet been applied to any study in the literature, while the method for filter visualization is also novel and has been benchmarked against several visualization methods, including one of the most used, the top activation method. Even though the modeling capabilities are shown during this study, deep learning models are susceptible to the dataset. ChIP-datasets tend to be noisy, including experimentally derived binding sites of TFs, and do not guarantee that there are binding sites of other TFs subsequently, therefore providing zero biological background for the modeling. Additive to that and as explained in the introduction, accessibility of the genome could be affected in various manners, therefore two sequences could have the same binding sites for TFs, but being differentially activated confuse the training procedure. Another limitation of the genomic modeling interpretability refers to the datasets used and how those are interpreted, as even though the datasets stem from genome-wide assays, the capturing biology differs among transcription factors, and therefore a comprehensive view on all the TFs and dynamics in a system could not be inferred, rather than is a broader view of the captured biological background on the modeling task. For such model to be effective, the modeling task should be thoroughly thought and contain a strong difference embedded in the dataset, so the model can then identify those differentiative factors among the sequences.

5.4 Future work

The application of explainability approaches on a DNA-trained convolutional neural network has provided valuable results, supporting our understanding on the model's

decision-making. Those results should serve as template for expanding further our knowledge on the model training procedure and investigating at a deeper level the dynamics of TFs captured from the model. There is room for further explainability in deeper convolutional layers, which could lead to deciphering of patterns of filter activations and therefore a better view on the TF dynamic activity. The whole methodology should be applied on a set of different modeling tasks to validate its capabilities in explaining the trained models, including multi-class and multi-label classification tasks and regression. Those trials should indicate how those explainability methods generalize to a broader set of modeling tasks, while also providing any valuable perspectives on the training procedure.

6. Discussion

In this study a CNN model was applied to decipher transcription factor dynamics, examining TCF1 binding sites in double-positive T-cells and NIH3T3 fibroblasts. The model performed well on classifying sequences by leveraging explainability modules for filter visualization, importance extracting, clustering, and enrichment providing detailed insights into TF activity and dynamics bridging the gap of similar approaches found in the literature. Notable findings included the identification of key TF motifs such as IKAROS and FOSL1::JUNB, which align with their known roles in T-cell development, and validate the model's biological relevance**.** The focus on the first convolutional layer limited the depth of interpretability, suggesting the need for future research to explore deeper convolutional layers for an enhanced view on the learned representations and their effect on the model's decision-making. The application of the proposed methodology to diverse genomic contexts, leveraging different input data such as ATAC-seq and DNase-seq data will validate the capabilities of the method to generalize, while the modeling should be tested on different tasks, namely multi-class and regression, to enable the interpretation of more complex tasks. Overall, while the study demonstrates the potential of CNNs in genomic modeling, further validation across various datasets and biological contexts is necessary to fully capture TF dynamics and interactions.

Figure List

Table list

Abbreviations

CNN	Convolutional Neural Network
DNN	Deep Neural Network
TF	Transcription Factor
XAI	Explainable Artificial Intelligence
AI	Artificial Intelligence
SHAP	Shapley value
ML	Machine Learning
DP	Double Positive
ChIP	Chromatin Immunoprecipitation
ATAC	Assay for Transposase-Accessible Chromatin
seq	Sequencing
TCF1	T Cell Factor 1

References

[1] H. Holtzer, H. Weintraub, R. Mayne, and B. Mochan, “Chapter 6 The Cell Cycle, Cell Lineages, and Cell Differentiation*,” vol. 7, A. A. Moscona and A. Monroy, Eds., in Current Topics in Developmental Biology, vol. 7. , Academic Press, 1972, pp. 229–256. doi: https://doi.org/10.1016/S0070-2153(08)60073-3.

[2] H. W. Brock and C. L. Fisher, “Maintenance of gene expression patterns,” *Developmental Dynamics*, vol. 232, no. 3, pp. 633–655, 2005, doi: https://doi.org/10.1002/dvdy.20298.

[3] O. Q. H. Zinani, K. Keseroğlu, and E. M. Özbudak, “Regulatory mechanisms ensuring coordinated expression of functionally related genes,” Jan. 01, 2022, *Elsevier Ltd*. doi: 10.1016/j.tig.2021.07.008.

[4] H. Zhang and Z. Z. Wang, “Mechanisms that mediate stem cell self-renewal and differentiation,” *J Cell Biochem*, vol. 103, no. 3, pp. 709–718, 2008, doi: https://doi.org/10.1002/jcb.21460.

[5] D. S. Latchman, “Transcription factors: An overview,” *Int J Biochem Cell Biol*, vol. 29, no. 12, pp. 1305–1312, 1997, doi: https://doi.org/10.1016/S1357-2725(97)00085-X.

[6] Y. Pan, C.-J. Tsai, B. Ma, and R. Nussinov, “Mechanisms of transcription factor selectivity,” *Trends in Genetics*, vol. 26, no. 2, pp. 75–83, Feb. 2010, doi: 10.1016/j.tig.2009.12.003.

[7] A. J. Warren, “Eukaryotic transcription factors,” *Curr Opin Struct Biol*, vol. 12, no. 1, pp. 107–114, 2002, doi: https://doi.org/10.1016/S0959-440X(02)00296-8.

[8] A. de Mendoza and A. Sebé-Pedrós, “Origin and evolution of eukaryotic transcription factors,” *Curr Opin Genet Dev*, vol. 58–59, pp. 25–32, 2019, doi: https://doi.org/10.1016/j.gde.2019.07.010.

[9] S. K. Burley and K. Kamada, “Transcription factor complexes,” *Curr Opin Struct Biol*, vol. 12, no. 2, pp. 225–230, 2002, doi: https://doi.org/10.1016/S0959-440X(02)00314-7.

[10] T. R. Hughes and S. A. Lambert, “Transcription factors read epigenetics,” *Science (1979)*, vol. 356, no. 6337, pp. 489–490, 2017, doi: 10.1126/science.aan2927.

[11] L.-Y. Zhao, J. Song, Y. Liu, C.-X. Song, and C. Yi, “Mapping the epigenetic modifications of DNA and RNA,” *Protein Cell*, vol. 11, no. 11, pp. 792–808, Nov. 2020, doi: 10.1007/s13238-020-00733-7.

[12] A. M. Deaton and A. Bird, “CpG islands and the regulation of transcription,” *Genes Dev*, vol. 25, no. 10, pp. 1010–1022, May 2011, doi: 10.1101/gad.2037511.

[13] T. Jenuwein and C. D. Allis, “Translating the Histone Code,” *Science (1979)*, vol. 293, no. 5532, pp. 1074–1080, Aug. 2001, doi: 10.1126/science.1063127.

[14] A. Balsalobre and J. Drouin, “Pioneer factors as master regulators of the epigenome and cell fate,” *Nat Rev Mol Cell Biol*, vol. 23, no. 7, pp. 449–464, 2022, doi: 10.1038/s41580-022-00464-z.

[15] M. Shirvaliloo, “The Landscape of Histone Modifications in Epigenomics Since 2020,” *Epigenomics*, vol. 14, no. 23, pp. 1465–1477, Dec. 2022, doi: 10.2217/epi-2022-0437.

[16] F. C. Grandi, H. Modi, L. Kampman, and M. R. Corces, “Chromatin accessibility profiling by ATAC-seq,” Jun. 01, 2022, *Nature Research*. doi: 10.1038/s41596-022-00692-9.

[17] F. Yan, D. R. Powell, D. J. Curtis, and N. C. Wong, “From reads to insight: A hitchhiker’s guide to ATAC-seq data analysis,” Feb. 03, 2020, *BioMed Central*. doi: 10.1186/s13059-020-1929-3.

[18] L. Luo, M. Gribskov, and S. Wang, “Bibliometric review of ATAC-Seq and its application in gene expression,” May 01, 2022, *Oxford University Press*. doi: 10.1093/bib/bbac061.

[19] P. J. Park, “ChIP-seq: Advantages and challenges of a maturing technology,” Oct. 2009. doi: 10.1038/nrg2641.

[20] P. Gade and D. V. Kalvakolanu, “Chromatin immunoprecipitation assay as a tool for analyzing transcription factor activity,” *Methods in Molecular Biology*, vol. 809, pp. 85–104, 2012, doi: 10.1007/978-1-61779-376-9_6.

[21] R. Mundade, H. G. Ozer, H. Wei, L. Prabhu, and T. Lu, “Role of ChIP-seq in the discovery of transcription factor binding sites, differential gene regulation

mechanism, epigenetic marks and beyond," Sep. 15, 2014, *Landes Bioscience*. doi: 10.4161/15384101.2014.949201.

[22] Y. He, Z. Shen, Q. Zhang, S. Wang, and D. S. Huang, "A survey on deep learning in DNA/RNA motif mining," Jul. 01, 2021, *Oxford University Press*. doi: 10.1093/bib/bbaa229.

[23] I. G. P. M. De-Signed Research; A, "Deep learning of immune cell differentiation," vol. 117, 2011, doi: 10.1073/pnas.2011795117/-/DCSupplemental.

[24] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modeling techniques for genomics," *Nat Rev Genet*, vol. 20, no. 7, pp. 389–403, 2019, doi: 10.1038/s41576-019-0122-6.

[25] Y. Xu *et al.*, "Artificial intelligence: A powerful paradigm for scientific research," Nov. 28, 2021, *Cell Press*. doi: 10.1016/j.xinn.2021.100179.

[26] E. Morales and H. J. Escalante, "A brief introduction to supervised, unsupervised, and reinforcement learning," 2022, pp. 111–129. doi: 10.1016/B978-0-12-820125-1.00017-8.

[27] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," *SN Comput Sci*, vol. 2, no. 6, p. 420, 2021, doi: 10.1007/s42979-021-00815-1.

[28] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," May 27, 2015, *Nature Publishing Group*. doi: 10.1038/nature14539.

[29] F. Doshi-Velez and B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning," Feb. 2017, [Online]. Available: http://arxiv.org/abs/1702.08608

[30] S. M. Lundberg, P. G. Allen, and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions." [Online]. Available: https://github.com/slundberg/shap

[31] C. O. Retzlaff *et al.*, "Post-hoc vs ante-hoc explanations: xAI design guidelines for data scientists," *Cogn Syst Res*, vol. 86, p. 101243, 2024, doi: https://doi.org/10.1016/j.cogsys.2024.101243.

[32] V. Kamakshi and N. C. Krishnan, "Explainable Image Classification: The Journey So Far and the Road Ahead," Sep. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/ai4030033.

[33] S. Bassan, G. Amir, and G. Katz, “Local vs. Global Interpretability: A Computational Complexity Perspective,” Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.02981

[34] J. L. Johnson *et al.*, “Lineage-Determining Transcription Factor TCF-1 Initiates the Epigenetic Identity of T Cells,” *Immunity*, vol. 48, no. 2, pp. 243-257.e10, 2018, doi: https://doi.org/10.1016/j.immuni.2018.01.012.

[35] P. K. Koo and M. Ploenzke, “Deep learning for inferring transcription factor binding sites,” Feb. 01, 2020, *Elsevier Ltd*. doi: 10.1016/j.coisb.2020.04.001.

[36] A. Shrikumar *et al.*, “Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5,” Oct. 2018, [Online]. Available: http://arxiv.org/abs/1811.00416

[37] D. R. Kelley, J. Snoek, and J. L. Rinn, “Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks,” *Genome Res*, vol. 26, no. 7, pp. 990–999, Jul. 2016, doi: 10.1101/gr.200535.115.

[38] F. Ullah and A. Ben-Hur, “A self-attention model for inferring cooperativity between regulatory features,” *Nucleic Acids Res*, vol. 49, no. 13, pp. e77–e77, Jul. 2021, doi: 10.1093/nar/gkab349.

[39] H. P. J. Buermans and J. T. den Dunnen, “Next generation sequencing technology: Advances and applications,” *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, vol. 1842, no. 10, pp. 1932–1941, 2014, doi: https://doi.org/10.1016/j.bbadis.2014.06.015.

[40] H. Satam *et al.*, “Next-Generation Sequencing Technology: Current Trends and Advancements,” Jul. 01, 2023, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/biology12070997.

[41] D. M. Church *et al.*, “Modernizing reference genome assemblies,” *PLoS Biol*, vol. 9, no. 7, Jul. 2011, doi: 10.1371/journal.pbio.1001091.

[42] P. Xiropotamos *et al.*, “aPEAch: Automated Pipeline for End-to-End Analysis of Epigenomic and Transcriptomic Data,” *Biology (Basel)*, vol. 13, no. 7, p. 492, Jul. 2024, doi: 10.3390/biology13070492.

[43] Q. Li, J. B. Brown, H. Huang, and P. J. Bickel, “Measuring reproducibility of high-throughput experiments,” *Annals of Applied Statistics*, vol. 5, no. 3, pp. 1752–1779, Sep. 2011, doi: 10.1214/11-AOAS466.

[44] A. R. Quinlan and I. M. Hall, “BEDTools: A flexible suite of utilities for comparing genomic features,” *Bioinformatics*, vol. 26, no. 6, pp. 841–842, Jan. 2010, doi: 10.1093/bioinformatics/btq033.

[45] L. Li, K. Jamieson, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” 2018. [Online]. Available: http://jmlr.org/papers/v18/16-558.html.

[46] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, “Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning,” *Nat Biotechnol*, vol. 33, no. 8, pp. 831–838, 2015, doi: 10.1038/nbt.3300.

[47] A. Sandelin, W. Alkema, P. Engström, W. W. Wasserman, and B. Lenhard, “JASPAR: An open-access database for eukaryotic transcription factor binding profiles,” *Nucleic Acids Res*, vol. 32, no. DATABASE ISS., Jan. 2004, doi: 10.1093/nar/gkh012.

[48] J. A. Castro-Mondragon *et al.*, “JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles,” *Nucleic Acids Res*, vol. 50, no. D1, pp. D165–D173, Jan. 2022, doi: 10.1093/nar/gkab1113.

[49] I. Rauluseviciute *et al.*, “JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles,” *Nucleic Acids Res*, vol. 52, no. D1, pp. D174–D182, Jan. 2024, doi: 10.1093/nar/gkad1059.

[50] S. Gupta, J. A. Stamatoyannopoulos, T. L. Bailey, and W. S. Noble, “Quantifying similarity between motifs,” *Genome Biol*, vol. 8, no. 2, Feb. 2007, doi: 10.1186/gb-2007-8-2-r24.

[51] T. L. Bailey, J. Johnson, C. E. Grant, and W. S. Noble, “The MEME Suite,” *Nucleic Acids Res*, vol. 43, no. W1, pp. W39–W49, 2015, doi: 10.1093/nar/gkv416.

[52] P. Sprent, “Fisher Exact Test,” in *International Encyclopedia of Statistical Science*, M. Lovric, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 524–525. doi: 10.1007/978-3-642-04898-2_253.

[53] Y. Zhang *et al.*, “Model-based analysis of ChIP-Seq (MACS),” *Genome Biol*, vol. 9, no. 9, Sep. 2008, doi: 10.1186/gb-2008-9-9-r137.

[54] H. Hosokawa and E. V. Rothenberg, “How transcription factors drive choice of the T cell fate,” Mar. 01, 2021, *Nature Research*. doi: 10.1038/s41577-020-00426-6.

[55] A. Shetty *et al.*, “A systematic comparison of FOSL1, FOSL2 and BATF-mediated transcriptional regulation during early human Th17 differentiation,” *Nucleic Acids Res*, vol. 50, no. 9, pp. 4938–4958, May 2022, doi: 10.1093/nar/gkac256.