Wide scale analysis of transcription factor biases and specificity

by

Aseel R. Awdeh

Thesis submitted to the
University of Ottawa
in partial fulfillment of the requirements
for the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

There are approximately 30 trillion cells in the human body, and nearly every cell has the same genomic sequence. Yet, due to differential gene expression, we have around 200 distinct cell types each with varying functionalities. The cell type specific states are maintained via the binding of multiple regulatory proteins to different locations along the genome in a process known as transcriptional regulation. Additionally, disruptions to the transcriptional regulation process may lead to the development of disease. Hence, uncovering the complex interplay of protein-DNA interactions along the genome is of critical importance. The advent of technologies probing the genomic sequence, as well as the development of powerful computational modeling techniques to relate DNA sequences to molecular phenotype, has enabled the understanding of many molecular processes genome wide. However, these computational methods require significant adaptation to biological systems – to accurately and fully account for the biology behind the molecular processes, as well as the biases associated with the data generating systems and processes. In this thesis, we address three main issues that arise from the use of omics data, more specifically ChIP-seq data, when identifying regulatory proteins along the genome. The first part of the thesis involves the study of the biases and noise associated with ChIP-seq experiments. Each experiment is prone to noise and bias, and as such we propose the use of a customized set of weighted controls, instead of equally weighted controls, for each ChIP-seq experiment in the peak calling process to mitigate the noise and bias. To do this, we implement a peak calling algorithm, called Weighted Analysis of ChIP-seq (WACS), which is an extension of the well-known peak caller MACS2, to incorporate the weighted controls in the peak calling process. We show that our approach assists in a better approximation of the noise distribution in controls, and fundamentally improves our understanding of ChIP-seq signals and their biases. Another aspect we explore in this thesis is the ability to uncover cell type specificity of transcription factor binding from the ChIP-seq data. A transcription factor may bind to various parts of the genome in different cell types, due to modifications in the DNA-binding preferences of the transcription factor, or other mechanisms, such as chromatin accessibility or cooperative binding, thus leading to a "DNA signature" of differential binding. We develop a deep learning approach, called SigTFB (Signatures of TF Binding) and conduct a wide scale analysis of hundreds of transcription factors to identify and quantify the varying degrees of cell type specific DNA signatures of various transcription factors across cell types. We also assess the consistency of cell type specificity for a specific transcription factor when assayed by different antibodies. We show that many transcription factors are indeed cell type specific, while others are more general with lower cell type specificity. Finally, to further explain the biology behind a transcription factor's cell type specificity, or lack that of, we conduct a wide scale motif enrichment analysis of all transcription factors in question. We show that cell type specific transcription factors are typically associated with corresponding differences in motif enrichment and gene expression. Together, these contributions deepen our knowledge of transcription factor binding, and how experimental and cell type specific variations can be uncovered.

Acknowledgments

First and foremost, I would like to thank my supervisor, Theodore J. Perkins, and co-supervisor, Marcel Turcotte, for their ongoing guidance, expertise and support throughout the duration of my graduate career. I would also like to thank my parents and brothers. This would not have been possible without their unconditional love, support and constant encouragement. Thank you for believing in me. Finally, I would like to thank members of the Perkins lab and other OHRI members: Justin Chitpin, Ali Karimnezhad, Zeinab Mokhtari, Hina Bandukwala, Matt Tanner, Gareth Palidwor and Christopher J. Porter.

This work was supported by the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-06604 and RGPIN-2014-04195), the Queen Elizabeth II Graduate Scholarship in Science and Technology, the Ontario Graduate Scholarship and a Compute Canada (www.computecanada.ca) Resources-for-Research-Groups grant. None of the agencies that funded this work had any role in the design of the study, in the collection, analysis, and interpretation of data, or in writing this material.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Next generation sequencing	NGS
Deoxyribonucleic acid	DNA
Ribonucleic acid	RNA
Transcription Factor	TF
Chromatin immunoprecipitation followed by high throughput sequencing	ChIP-seq
Antibody	AB
RNA sequencing	RNA-seq
DNase I hypersensitive site sequencing	DNase-seq
Immunoglobulin G	IgG
Weighted Analysis of ChIP-seq	WACS
Polymerase Chain Reaction	PCR
Signatures of TF Binding	SigTFB
Messenger RNA	mRNA
Base Pair	bp
Cleavage Under Targets and Tagmentation	CUT&Tag
Protein Binding Microarray	PBM
High Throughput-Systematic Evolution of Ligands by Exponential Enrichment	HT-SELEX
Assay for Transposase-Accessible Chromatin with high-throughput sequencing	ATAC-seq
Irreproducible Discovery Rate	IDR
Independent and Identically Distributed	IID
False Discovery Rate	FDR
Position Weight Matrix	PWM
Position Specific Scoring Matrix	PSSM
Protein Binding Microarray	PBM
Binding Energy Estimation by Maximum Likelihood for Protein Binding Microarrays	BEEML-PBM
Machine Learning	ML
Deep Neural Network	DNN
Convolutional Neural Network	CNN
Recurrent Neural Network	RNN
Rectified Linear Unit	ReLU
Gated Recurrent Unit	GRU
Long Short-Term Memory	LSTM
Area Under the Precision-Recall Curve	AUPRC
Receiver Operating Characteristic	ROC
Area Under the receiver operating characteristic Curve	AUC
True Positive Rate	TPR
False Positive Rate	FPR

Multi-Task Learning	MTL
Transcription Factor Motif Discovery from Importance Scores	TF-MoDISco
Global Importance Analysis	GIA
Micrococcal nuclease digestion with deep sequencing	MNase-seq
Alternative splicing	AS
Model-based Analysis for ChIP-Seq	MACS
Non-negative least squares	NNLS
Find Individual Motif Occurrences	FIMO
CCCTC binding factor	CTCF
Estrogen receptors α	ERs
Human embryonic stem cells	hESC
Cell lines/types	CLs
Cell type specific	CL Specific
Cell type general	CL General
Activating Transcription Factor 7	ATF7
Cyclic AMP response element	CRE
Basic leucine zipper	bZIP
Multiple Expectation maximizations for Motif Elicitation	MEME
Alternative Splicing	AS

Chapter 1

Introduction

1.1 Overview

There has been ongoing international effort to completely sequence the human genome, with the yet most complete version of the genome published recently in 2022 [188]. With the use of next generation sequencing (NGS) technologies, scientists have been able to interpret the structure and functionality of the genetic information encoded in the genome. Information derived from sequencing either the entire genome, or targeted smaller regions of it, have led to novel scientific insights in the understanding of biology and disease, more specifically allowing for more personalized medicine and patient care.

The genome is a blueprint for protein synthesis. More particularly, according to the central dogma of molecular biology, the DNA (deoxyribonucleic acid) is transcribed into RNA (ribonucleic acid), which is then translated into a chain of amino acids. Yet, out of the 3 billion base pairs in the human genome, less than 2% are protein-coding regions, whilst the majority (98%) comprise some of the non-protein coding regions. Amongst many components, such as non-coding RNA genes, repeats, pseudo-genes and retro-transposons, the non-protein coding regions of the genome also consist of regulatory elements. These elements are responsible for the control of gene expression in diverse cellular contexts allowing for specialized functionalities per context. An example of key regulatory elements responsible for gene expression regulation are transcription factors (TFs). TFs are proteins that bind to specific short patterns of the DNA sequence, also known as motifs, that exist in the non-protein coding regions of the genome. Uncovering the differential binding preferences of a TF across various cell types and tissues, and understanding its impact on downstream genomic processes, is important in molecular biology. Mutations also occur in these non-protein coding regions. In fact, many disease associated variants are located within these non-coding regions [159], where they perturb TF binding sites, and ultimately complicate gene expression and cellular functionality.

The availability of functional genomic assays by international collaborative projects such as ENCODE [56] and the Roadmap Epigenomics Project [35] has allowed genome-wide profiling of many biochemical measurements of the non-coding, as well as the protein-

coding, regions of the genome. An example of a widely used assay is chromatin immunoprecipitation followed by sequencing (ChIP-seq), an experimental methodology used to identify protein-DNA interactions genome-wide, typically histone modifications or TF binding sites, in various cell types at different developmental phases [155, 179]. In ChIP-seq, proteins are extracted using a specific antibody (AB), and the DNA fragments attached to those proteins are then sequenced and mapped back to a reference genome to infer the binding sites. These binding sites contain the sequence motifs for the TFs in question. Examples of other genomic assays include RNA sequencing (RNA-seq) for measuring gene expression [254] and DNase I hypersensitive site sequencing (DNase-seq) for quantifying chromatin accessibility genome wide [229].

Understanding transcriptional regulation, and how mutations impact gene expression, is essential. Due to the many levels of complexity that exist between the TF activity and the phenotypic output, it is not simple, however. The main difficulty commences at identifying the DNA binding preferences of TFs. There are many features that could influence binding, and thus understanding the combinatorial impact of these features is essential for having a more comprehensive understanding of transcriptional regulation. Features that influence TF binding may be derived from contextual or epigenetic information encoded within the genomic sequence, or external intra- or inter-molecular TF interactions. Such factors may then alter the motifs or combinations of motifs recognized by a TF, as well as the specificity and sensitivity of binding, thus demonstrating complications in the use simple recognition models to represent binding [223]. For example, in addition to the intrinsic binding preference of a TF, factors such as the DNA shape, co-operative and competitive binding, chromatin modifications and alternative splicing may influence binding [223]. Some TFs may also have multiple DNA binding domains, and thus have the ability to recognize multiple motifs [104, 223]. Another level of complexity is introduced by the multifaceted nature of the experimental protocols, such as ChIP-seq, used to obtain TF binding sites – as they are prone to noise and bias at every step of the process. Accounting for noise and bias in ChIP-seq is essential for the prevention of false overestimation, or even underestimation, of TF binding sites.

The obstacles encountered when uncovering the functionality of the non-coding regions of the genome, as well as the availability of such large volumes of sequencing data, have motivated the use of computational machine learning approaches. Using as input raw DNA sequences or genomic assay profiles, such as ChIP-seq profiles of TF binding or DNase-seq profiles of chromatin accessibility, machine learning techniques are able to analyze large complex data to better understand underlying biological processes. Machine learning methods have been proposed to learn genomic features, such as chromatin accessibility, TF binding, histone methylation/acetylation, and gene expression. Moreover, machine learning will not only be able to identify regions of enrichment, expression or accessibility genome wide, but can also uncover the underlying DNA sequence patterns driving the selection of such features by the model.

The advent of these new technologies, such as high throughput sequencing and sophisticated machine learning models, have increased our understanding of transcriptional regulation. Ideally, we would want computational methods to fully represent the complex protein-DNA interactions. However, due to the biases and noises associated with

omics data, and the complexity of the underlying biological systems, there are yet some limitations that need to be addressed.

1.2 Problem Statement

In this thesis, I investigate three main themes of transcriptional regulation to achieve a better understanding of the underlying molecular mechanisms behind the complex protein-DNA interactions. The aim is to systematically explore distinct patterns observed in ChIP-seq data, in terms of bias and noise, as well as TF cell type specificity and motif enrichment.

1.2.1 Noise and Bias in ChIP-seq Experiments

Noise and bias are introduced at every step in the cascade of events leading to the generation of high throughput sequencing data. Noisy datasets, for example, may be caused by poor quality DNA, low concentration of DNA or inadequate primers, thus making it difficult to distinguish between background noise and the actual regions of enrichment. However, the use of statistical models on well-prepared samples of strong signal strength helps alleviate the problem. Bias, on the other hand, is a more complex matter, as it still exists even with high quality samples and statistical models. Bias may stem from the non-uniform structure of chromatin leading to the differential fragmentation of the genome, or sequence dependent polymerase chain reaction (PCR) amplification due to GC content bias, or the differential mappability of short reads due to coverage bias, or even variable antibody specificity in the case of ChIP-seq data [135, 162]. Accounting for these technical artifacts is essential for obtaining high quality reproducible results, as not accounting for bias produces misleading output.

The fundamental goal of this thesis is to better understand trancriptional regulation. Since ChIP-seq is a popular high-throughput genome profiling technology used in the study of TFs, I investigate bias in ChIP-seq data. According to the ENCODE and modENCODE consortia [56], one way to mitigate bias influence is through the integration of control datasets, such as input DNA or ChIP with Immunoglobulin G (IgG) antibody, in the ChIP-seq analysis. Various ChIP-seq datasets are differently biased, however. Depending on which controls are used, different regions of enrichment can be selected for the same ChIP-seq dataset. I implement a peak calling algorithm, Weighted Analysis of ChIP-Seq (WACS), to address the aforementioned limitations. Given a target experiment and a set of control experiments as input, the ideal contribution of each control experiment is determined using non negative least squares regression, so as to maximize the value of the target experiment. WACS shows the importance of smart bias removal methods via the use of a customized selection of weighted control datasets per ChIP-seq experiment.

1.2.2 Cell Type Specific DNA Signatures of Transcription Factor Binding

The cell type specificity of TFs is essential for the establishment and maintenance of gene regulation in distinct cellular contexts, and as such important in the study of phenotypic variation and disease. Most disease associated variants are located in the non-protein coding regions of the genome. Moreover, cell type specific TFs assist in biomarker identification of certain diseases [267]. This is possible because TFs can regulate the expression of cell or tissue marker genes, or inversely marker genes can influence TF activity. TFs could even act as cell markers themselves, such as TF GATA3 in the case of breast cancer. Thus, through the study of cell type specific TFs, a more personalized model of care can be produced to counteract the mechanisms that lead to disease development.

It is widely established that the same TF binds to multiple genomic regions in different cell types, hence corroborating the need for large scale TF-DNA interaction mapping projects, such as ENCODE [56]. Yet, a TF may bind to different binding sites along the genome due to disparities in chromatin accessibility, or other mechanisms, such as steric hindrance or cooperative binding, without any intrinsic difference in the DNA recognition preference. Databases, such as JASPAR [44] and HOMOCOCO [132], implicitly assume that DNA binding preferences of TFs are context independent, meaning factors that do not cause an inherent difference in binding preference are not accounted for. There are a few exceptions, however, where, for example, motifs for heterodimers are provided and distinct from their constituent TFs. Indeed, there are many documented studies showing how factors, such as complexing with other proteins, alternative splicing and cooperative binding, influence the TF-DNA binding preferences of a TF.

I develop a deep learning approach, called SigTFB (Signatures of TF Binding), to undertake the first wide-scale study of this phenomenon, creating an atlas of cell type specificity for hundreds of TFs. I examine TF-DNA interactions to determine the "DNA signatures" of differential binding of TFs. The term DNA signature is used to encompass not only the intrinsic binding preference of a TF, but other factors, such as differential chromatin accessibility, competitive and cooperative binding, and steric hindrance, that could influence the DNA binding preference of a TF. It is important to note that SigTFB is used to determine the degree, not the cause, of cell type specificity. The heterogeneity of ChIP-seq across various cell types is uncovered, and the cell type specificity, or lack that of, of TFs is investigated. A cell type specific TF is defined to be a TF that has different DNA signatures in various cell types, while a more general TF binds to the same signature regardless of cell type. Contrary to the popular use of deep learning solely for the prediction of TF binding genome wide, here a deep learning approach is used for the prediction of cell type specific DNA signatures of different TFs. More specifically, the degree of cell type specificity of TFs is characterized and quantified on a wide scale. The consistency in cell type specificity of a TF across cell types and even antibodies is also explored.

1.2.3 Motif Enrichment Analysis of Cell Type Specific Transcription Factors

Unraveling mechanisms that lead to the cell type specific nature of TFs is critical in the understanding of regulatory mechanisms and disease causing variants. In the third main contribution, I further investigate the cell type specificity atlas and the trained deep neural networks to understand which specific sequence features drive binding in different cell types. Thus, to identify what motif patterns lead to specific binding site selection by a TF, and conversely, what patterns are more general, and to explain why and how TFs bind to specific binding sites, a wide scale motif enrichment analysis of TFs and their corresponding cell types and tissues is conducted. The aim is to also determine whether the cell type specific predictions, attained earlier, are consistent with the motif enrichment and gene expression analyses.

1.3 Thesis Statement

Traditional computational approaches, be it for peak calling when analyzing ChIP-seq data or the prediction of TF binding sites using called peaks, have been proposed to obtain novel scientific insights on transcriptional regulation. However, due to the complexity of TF-DNA interactions, computational methods need improvement to accurately represent the underlying biological mechanisms. The main goal in this thesis is to address three main themes that arise in the study of transcriptional regulation – in terms of noise and bias in ChIP-seq experiments, and TF DNA-signatures and motif enrichment. The aim is to develop, apply and evaluate computational methods to better decipher the non-protein coding regions of the genome. The methods developed should accurately represent input ChIP-seq data, effectively process large volumes of complex data, and provide appealing visualization to communicate results. Overall, with minor modifications, these techniques should be adaptable to other high throughput sequencing datasets.

1.4 Contributions

Our main contributions consist of the development of machine learning approaches to analyze high throughput sequencing data, that is ChIP-seq, for a deeper and more comprehensive analysis of transcriptional regulation. The following is a list of contributions in the order that they appear.

1. A peak calling algorithm for smart control selection and weighting.
2. A wide scale evaluation of the smart bias removal peak calling method on 438 ChIP-seq datasets and 147 control datasets of the K562 cell line, and 20 ChIP-seq datasets for each of the A549, GM12878 and HepG2 cell lines. Improved specificity in peak calling compared to state-of-the-art methods is shown.

3. A systematic comparison of smart control construction versus random control selection in peak calling.

4. A systematic assessment on the number of smart controls used in peak calling.

5. A wide scale investigation of cell type specific DNA signatures of TF binding for 194 distinct TFs covering a total of 35 human cell types. Variations in the degrees of cell type specificity for the various TFs are detected and quantified.

6. A study on the consistency in cell type specificity for a specific TF across different antibodies.

7. A wide scale investigation of motif enrichment and gene expression in a total of 261 TF-AB and cell type combinations. Results show that differential motif patterns often correlate with a higher degree of cell type specificity, such as in the case of ATF7, while similar motif patterns across cell types for the same TF often relates to lower degrees of cell type specificity, such as the case of CTCF.

1.5 Thesis Outline

Chapter 2 surveys the literature and provides an overview on high throughput sequencing data, more specifically ChIP-seq data, transcriptional regulation, TF binding specificity, and finally deep learning in genomics. Chapter 3 introduces WACS, the weighted peak calling algorithm I developed to address bias in ChIP-seq data via the incorporation of a weighted customized selection of controls. Chapter 3 also compares WACS to other peak calling algorithms, and provides a wide scale analysis on motif enrichment and reproducibility. Chapter 4 proposes a deep learning approach implemented to explore the cell type specificity of TFs. Chapter 4 also conducts a wide scale analysis of TFs across various cell types and tissues to identify and quantify their degree of cell type specificity. Chapter 5 implements a wide scale motif enrichment and gene expression analysis to uncover the motif patterns driving the cell type specific nature of TFs. Chapter 6 summarizes the conclusions, outlines the limitations of proposed approaches, and proposes possible directions for future work.

Chapter 2

Literature Review

In this chapter, we review the literature encompassing both the biological and computational aspects of our aims. We first provide a simple explanation of the biology in terms of genes, gene expression and transcriptional regulation. We then discuss different high throughput sequencing technologies, explain the ChIP-seq methodology specifically in detail, and discuss methods used for representing TF binding specificity. We finally explore the application of deep learning in genomics, more specifically in the prediction of TF binding sites.

2.1 Biology

Human cells contain genetic information in the form of DNA. This DNA is a double-stranded nucleic acid consisting of nucleotides building blocks: adenine (A), guanine (G), cytosine (C) and thymine (T). The sequences of nucleotides represent various protein-coding genes, where a gene is a region in the DNA that contains the genetic instructions necessary to produce sequences of amino acids, known as proteins. More specifically, a gene consists of both protein-coding regions, equivalently exons, which translate into proteins, and non-coding regions, equivalently introns, which do not code for proteins. Depending on the order of nucleotides in a gene, different proteins with different functionalities are produced. There are also non-coding genes which never produce a protein product, but rather produce an RNA which may have some function in the cell. This is not our focus.

Analogous to a computer, a human cell consists of a set of instructions, more particularly genes, that need to be interpreted and decoded by the cell's hardware. The process of interpretation and conversion of the sequence of nucleotides into a protein constitutes the central dogma. It is also known as gene expression. Gene expression is carried out in two main stages: transcription and translation. Transcription involves the conversion of a DNA sequence of a gene into an RNA sequence. Transcription is carried out by enzymes, such as the RNA polymerase, which initiates transcription, and other regulatory proteins, such as TFs, which determine when and where proteins are synthesized. TFs, for example, are DNA-binding proteins which bind to specific DNA sequences along the genome, either

promoter regions near the transcription start site to initiate transcription, or enhancer regions to enhance or repress the transcription of the gene.

RNA sequences are single stranded nucleic acids, which similar to DNA, are made of nucleotides.The building blocks of RNA are the nucleotides adenine, guanine, cytosine, and uracil (U). Moreover, the RNA molecule produced by transcription is also composed of coding and non-coding regions. The RNA responsible for the translation of a nucleotide sequence into a protein is the messenger RNA (mRNA). Post transcription, in a process called splicing, non-coding regions are removed from pre-mRNA to produce mature mRNAs. In addition to mRNA, there are other types of RNA, such as transfer RNA, ribosomal RNA, microRNA and long noncoding RNA, that assist in the transcription and translation processes [140], and thus play a vital role in regulation of gene expression.

The second main step of gene expression is the translation of the mRNA sequences into proteins. Each set of three nucleotides in the mRNA forms a codon, and each codon corresponds to an amino acid, which is the building block of proteins. With the mRNA sequence acting as a template, the chain of amino acids is assembled in the same order as they appear to form a linear sequence of amino acids, known as a polypeptide. The ribosome and tRNA molecules assist in the translation process. The polypeptide then folds in specific manner to form a well defined three-dimensional structure with a unique functionality, which is also known as a protein. Additional post-translational modifications may be required to produce functional proteins. These may include, but are not limited to, the addition of phosphate groups in phosphorylation, or methyl groups in methylation, or acetyl functional groups in acetylation.

Variations in transcription or translation can impact gene expression. The binding ability of RNA polymerase to promoter regions, for example, may be affected by the binding of other competing TFs to the same promoter regions, which ultimately affects the rate of transcription by controlling the amount of mRNA produced from a specific gene [191]. The rate at which mRNA is processed, that is the speed at which non-coding regions are removed, can also effect the quantity of mature mRNAs, and hence proteins made. Moreoever, post-transcriptional events can control the amount of proteins produced, while post-translational events can modify the actual protein output.

So, how can gene expression lead to the expression of specific genes in specific cell types or tissues only when its corresponding protein is required? Albeit various cell types have the same genetic code, gene expression is regulated by TFs, co-factors and chromatin regulators to create and maintain cell type specific states in humans [66, 140]. Differences between cell types is often due to the exposure of cells to different extracellular signals from their immediate environments. For instance, during development, the graded activity of morphogen signals informs cells on their whereabouts in the organism, stimulating signaling pathways that turn on different sets of genes in different cells [208]. Other signaling molecules, such as those associated with disease, health and aging, are continuous throughout a cell's life cycle. Moreover, events within cells, including random molecular events, can influence cell behaviour. A change in the regulation of gene expression can lead to diseases, such as breast cancer and diabetes [140].

Understanding transcription, more specifically how and where TFs bind to specific DNA

sequences, is essential to understanding differentially expressed genes. Transcriptional regulation is the combinatorial result of the structural properties of the DNA-protein complex (i.e chromatin) and the TF-DNA interactions [191, 204]. Different TFs can have different DNA binding domains which allow them to bind to specific regions along the genome. However, there are many factors that can influence transcription. For example, chromatin remodeling complexes are usually recruited when a TF binds to the DNA. These complexes can either increase or decrease the accessibility of chromatin, making the binding sites along the genome more accessible or less accessible to TFs respectively. Additionally, histone and DNA modifications, referred to as the post-translational modifications to histone proteins and DNA respectively, can influence transcriptional activity through acetylation, methylation and phosphorylation. For example, the addition of the methyl group to a gene promoter region along a DNA strand may repress gene transcription [275].

2.2 High Throughput Sequencing

The discovery of DNA as genetic material in 1928 by Frederick Griffith has led to advancements in DNA-based technologies aiming to decode the blueprint of life [13]. Sequencing, or the ability to read and determine the order of nucleotides in a DNA sequence, has revolutionized biology. Understanding genomic sequences has led to better interpretation of the structure, functionality and meaning of the genetic information.

Early sequencing efforts brought about Sanger sequencing, also known as the chain termination method, in the 1970s [2]. Although successful in its ability to sequence DNA fragments or even full genomes, Sanger sequencing remains expensive in terms of time and cost, as only one sequence of up to 900 base pairs (bps) can be sequenced at a time. Additionally, the quality of sequencing deteriorates as sequencing length increases, and this method can only sequence one DNA fragment at a time. More recently, in the early 2000s, improved next generation sequencing (NGS) technologies, also known as massively parallel or short-read sequencing, that led to a greater magnitude of sequence data to be generated at low cost were introduced [27, 84, 160]. The emergence of NGS technologies has proven to be beneficial over traditional sequencing methods. Unlike Sanger sequencing, where only one DNA sequence is sequenced at a time, NGS allows the sequencing of millions of short DNA fragments simultaneously in a more cost effective and time efficient manner. Thus, an entire human genome of three billion nucleotides can be sequenced in a day using NGS. Other advantages include the ability to detect low frequency variants at higher sensitivity, as well as offering a more comprehensive genomic coverage and allowing for paired-end sequencing, where both ends of a fragment are sequenced. Examples of NGS technologies include Illumina, 454 pyrosequencing, Ion torrent and SOLiD.

The common steps shared across NGS platforms are library preparation, clonal amplification, sequencing and data analysis. Library preparation is a multi-step process which usually involves (1) the extraction and purification of DNA sequences from samples, (2) the fragmentation of the DNA sequences and (3) the ligation of adapters to both ends of the fragments. This is followed by the amplification of the adapter-ligated fragments via polymerase chain reaction (PCR), as well as bridge amplification in case of the Illumina

platform, to produce billions of short DNA fragments (or sequencing reads) ranging from 40 to 700 bp [89]. With the Illumina platform, sequencing is then carried out using either sequencing by synthesis, which involves the incorporation of flourescent nucleotides one at a time to a DNA template strand, or sequence by ligation, which involves the addition of multiple bases at a time. Other platforms, such as Ion Torrent, use a slightly different sequencing approach based on the principle of detecting electrical signals when a hydrogen ion is emitted during the addition of a nucleotide to the growing DNA strand.

There are many specialized applications of NGS that help uncover the mechanisms of gene regulation and cell adaptation to external and internal environments [27, 55, 134, 176, 228, 233]. These applications work not by sequencing all the DNA in the cell, but just certain parts of the DNA. For instance, in the next section, we will explore ChIP-seq, a commonly used application for identification of TF and histone binding sites. Briefly, ChIP-seq works by using an antibody to bind and pull a specific protein of interest out of a cellular mix, along with the DNA to which it is bound (see Figure 2.1). Sequencing that DNA then tells one what parts of the genome were bound by the protein. There are other alternatives to ChIP-seq, such as ChIP-exo [206] and Cleavage Under Targets and Tagmentation (CUT&Tag) [119]. ChIP-exo, unlike ChIP-seq, finds the binding intensity per base pair, while CUT&Tag does not need as much starting material as ChIP-seq. Other high throughput techniques that measure the sequence specificities of DNA-binding proteins include, but are not limited to, in vitro protein binding microarray (PBM) and high throughput-systematic evolution of ligands by exponential enrichment (HT-SELEX). NGS technologies can also quantify gene expression in a cell with RNA-seq by measuring RNA transcript levels, or measure chromatin accessibility genome wide with Assay for Transposase-Accessible Chromatin with high-throughput sequencing (ATAC-seq) or DNase-seq. However, the central focus of this thesis is ChIP-seq, so we describe this procedure and the analysis of its data in more detail in the next section.

2.3 ChIP-seq

One widely used technology by the scientific community is chromatin immunoprecipitation followed by next generation sequencing (ChIP-seq) [155, 179]. ChIP-seq analysis has allowed the extensive investigation of the structural and functional elements encoded in a genomic sequence. The main objective of ChIP-seq is the detection of protein-DNA binding sites and histone modifications genome wide in various cell lines and tissues.

In a ChIP-seq experiment, DNA is first crosslinked with proteins using formaldehyde or another chemical reagent to form DNA-protein complexes (Figure 2.1). Next is the process of chromatin fragmentation, where DNA is sheared to produce several DNA fragments, which are typically 100 to 500 bp in length. Then, in the chromatin immunoprecipitation step, a specific antibody is added to the complex to select the proteins of interest, along with the DNA to which they are crosslinked. In this case of a TF, this should ideally be an antibody that specifically binds that TF and no other proteins. While in ChIP-seq for a histone mark, this would be an antibody that specifically binds to only the chemically modified form of a histone protein, and not the unmodified protein version nor

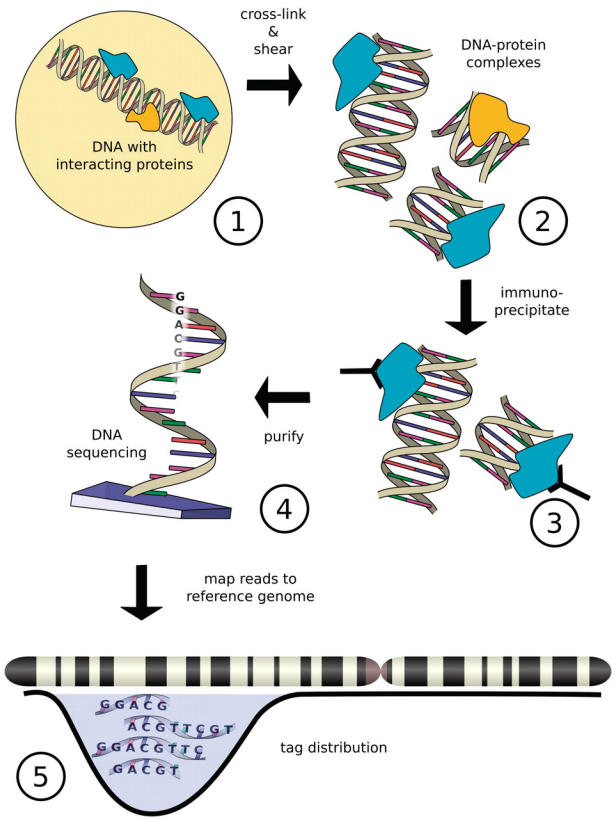


Figure 2.1: Workflow of ChIP-seq analysis. Chromatin in the nucleus (1) is cross-linked with proteins to produce DNA-protein complexes, then sheared to produce several DNA fragments, (2) followed by immunoprecipitation where specific antibodies are added to the DNA-protein complexes to select the proteins of interest. Short reads (3) are obtained using high throughput sequencing, (4) then mapped to the reference genome, (5) finally yielding a distribution of tags on the genome. Source: [239], used with permission conveyed through Copyright Clearance Center, Inc.

any other proteins. The active gene enhancer H3K27ac, for example, means the acetylation of the lysine residue at position 27 of the H3 histone protein. This is followed by DNA recovery and purification, where DNA is retrieved and recovered from the DNA-protein complexes to identify the protein bound sequences. The retrieved DNA fragments are then sequenced and mapped to a reference genome using mapping programs such as Bowtie [136] or Burrows-Wheeler Alignment [141], producing stacks of aligned reads across the genome. Some read pileups correspond to true regions of enrichment, also known as peaks, while others may be a result of the distortion of the ChIP-seq signal.

2.3.1 Peak Calling

The separation of signal from noise in the stacks of aligned reads is an important step in the analysis of ChIP-seq data. Peak calling is a computational method used to identify genomic regions enriched with aligned reads comparatively to a background signal. More

than 30 peak calling algorithms have been implemented, including but not limited to MACS2 [280], SICER [268, 272], F-Seq2 [38, 281], SISSR [115], QuEST [247] and HOMER [93]. Surveys that compare different ChIP-seq peak calling methods are outlined here [51, 109, 163, 189, 243, 259]. The overall peak calling approach often involves two main sub-problems: (1) the identification of regions of enrichment along the genome and (2) the statistical evaluation of the candidate regions to identify the true regions of enrichment (peaks). Prior to any peak calling method, however, is the preprocessing of the ChIP-seq data.

(i) Preprocessing Data

The biases and noise in ChIP-seq data, if not properly addressed, can greatly impact the interpretation of the output produced. Thus, due to the importance of developing effective computational peak calling methods, many methods have attempted to tackle such issues via the normalization of read counts, the removal of duplicate reads, and the incorporation of controls, to ultimately build better estimates of the ChIP-seq signal. In this section, we briefly describe each of the preprocessing steps used in the analysis of ChIP-seq data.

Background Signal: The incorporation of negative controls, or a background signal, is an essential step in the analysis of ChIP-seq. Noise and bias in ChIP-seq experiments typically increase the number of false positive peaks identified [51, 152, 189]. The use of controls, however, aids in the increase of sensitivity, which is the ability to correctly identify bound regions, and specificity, which is the ability to correctly identify unbound regions [152]. Moreover, negative controls correct for GC content bias that may occur when more (or fewer) reads map to GC-rich regions, and are often used for the detection of irregular relationships between the antibody and DNA sequence. Controls can be in the form of input DNA control, where a DNA sample that has not been immunoprecipitated is used, or a mock immunoprecipitated control, where a non-specific antibody, such as the immunoglobulin G (IgG) antibody, is used to produce random immunoprecipitated DNA fragments by binding to irrelevant proteins. In the case of no controls, peak calling methods typically model the background distribution as a Poisson or negative bionomial distribution. However, with the integration of controls, the enrichment of reads mapped to ChIP-seq relative to the control is often computed by either computing the difference or ratio of enrichment between the two. The control sample can also be used to compute the parameters of the Poisson or negative bionomial distribution. Furthermore, due to the biases that exist in ChIP-seq and controls datasets, other methods, such as CloudControl [97] and AIControl [98], have suggested the notion of smart customized controls to more accurately model the background signal of a ChIP-seq experiment. In AIControl, for example, ridge regression is used to fit the controls to the background signal of a ChIP-seq experiment.

Normalize Data: Normalization is a critical step in the analysis of ChIP-seq. It involves the transformation of independent ChIP-seq data sets from various sources, be it

different biological replicates or experimental conditions, for direct comparison [5, 65, 146]. Normalizing data accounts for differences in sequencing depth (the total read count). It also accounts for several sources of systematic bias that may arise due to the multi-faceted nature of the ChIP-seq pipeline. As previously mentioned, control datasets are often used in the peak detection process. ChIP-seq and control samples are typically sequenced at varying depths, and have different read distributions. Thus, after read counts are computed per genomic window in both samples, they are normalized for a more effective comparison. A simple strategy accounts for the total number of reads in each sample. Let N and M be the total number of reads in the ChIP-seq and control samples respectively. To linearly scale the control read count to be the same as the ChIP-seq read count, the control read density is multiplied by the scaling factor N/M. However, this method does not account for the non-uniform distribution of reads across the genome, and may incorrectly model the background distribution [65, 146]. Another method uses the notion that ChIP-seq is composed of both background and enrichment regions to compute a scaling factor that normalizes the background reads of the ChIP-seq signal with respect to the control signal [65].

Filter Duplicates: Duplicate reads are reads that map to the same region along the genome [68, 244]. Duplicates that arise due to repetitive DNA, or the over-amplification of genomic regions via PCR, often lead to false conclusions made about the true levels of enrichment. Biological duplicates may also occur due to the sequencing of DNA fragments originating from the same genomic region. Most peak calling approaches automatically remove or offer the option of removing duplicate reads. Duplicate reads can also be removed by the user with tools like DeDup [186] or SAMtools [142] either before mapping (based on sequence identity), or after mapping (based on identical mapping location). Although it may underestimate the ChIP-seq signal, duplicate removal has shown to significantly improve the output quality and downstream genomic analyses. Duplicate removal is often performed prior to building a signal profile.

Biological Replicates: The use of multiple biological samples in the assessment of reproducibility assists in the mitigation of bias in ChIP-seq [135, 152]. According to the ENCODE guidelines, each ChIP-seq experiment requires at least two independent biological ChIP-seq replicates [135]. With largely variable replicates, there is more inconsistency in the peaks detected, thus suggesting the need for improvement or reiteration of the experimental protocol. Consistency in peaks detected from different replicates gives reassurance in the experimental quality of the ChIP-seq used. More particularly, it is essential not only to test the consistency in peak detection between replicates, but to also merge information from replicates for further ChIP-seq analysis. Methods could simply compute the peak overlap between two independent biological replicates, or pool reads from the replicates together and call peaks from the pooled data. An extensively used methodology by ENCODE is the Irreproducible Discovery Rate (IDR) framework [135, 145]. Here, peaks are first called for each replicate and then sorted according to a measure of significance, such as p-value or q-value. The general idea is that two replicates should display high consistency in the most significant peaks detected, and lack it in the least significant peaks. Highly

significant peaks will more likely be actual regions of enrichment, while less significant ones will mostly be associated with noise. The transition of consistency between high to low significant peaks acts as an IDR threshold for the detection of reliable peaks.

(ii) Identification of Regions of Enrichment

Be it narrow peaks produced by TF binding, or broad peaks produced by histone modifications, peak calling methods commence by creating peak profiles for each chromosome. The simplest strategy involves the direct count of the number of mapped reads overlapping genomic intervals, and the identification of intervals with counts more than a set threshold as regions of enrichment. However, due to the complexity of ChIP-seq in terms of signal, experimental noise and artifacts, this method is ineffective. Additional aspects of ChIP-seq need to be recognized for better discrimination [51, 189].

Fragment length and Position Estimation: A typical initial step in peak calling methods is the approximation of the fragment length distribution [51, 189]. ChIP-seq reads only represent the DNA fragment ends, and not the full length fragments. Moreover, ChIP-seq experiments are typically single end-sequenced, suggesting that reads are equally likely to originate from either the forward or reverse strand. The strand specific nature of the read distribution results in a biomodal distribution around the true binding sites of the target protein, as seen in Figure 2.2. The estimated fragment size is equal to the distance between the two modes/peaks in the bimodal distribution. Fragment length estimation is often coupled with the shifting of reads towards the center of the binding site, and the extension of reads to represent the estimated fragment length. These initial steps help better locate the precise binding sites of proteins.

Build Signal Profile: Building a strand specific signal profile is an important step in ChIP-seq analysis [51, 189]. Smoothing techniques are used to smooth, or approximate, the read count densities. For example, MACS2 [280] and SISSR [115] slide a window of specified length across the genome and compute the total or average read count per region. Regions with read counts exceeding a specific threshold are identified and merged if consecutive. Fragments are extended and shifted prior to the sliding window scans in MACS2, unlike SISSR where the summit is detected prior to fragment extension. Other peak calling methods, such as QuEST [247] and F-Seq 2 [38], use kernel density estimation with a Gaussian kernel to model the read pileup across the genome. The derived probabilites at each base or genomic interval are proportional to the probability of finding mapped reads at that region.

(iii) Assessment of Candidate Regions

Candidate peaks are defined according to specific selection criteria, such as read count threshold or minimum enrichment ratio, p-value or q-value [51, 189]. A simple strategy is to identify candidate regions as regions with read counts more than a preset read count

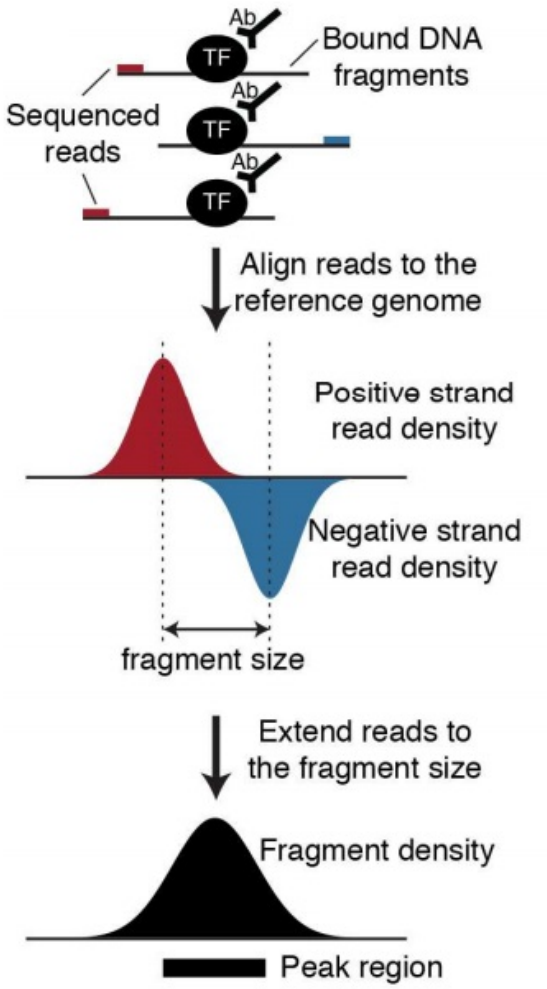


Figure 2.2: Overview of the ChIP-seq pipeline. Transcription factor binding sites display a characteristic bimodal distribution of the positive (forward) and negative (reverse) strand reads. Source: [25], used with permission conveyed through Copyright Clearance Center, Inc.

threshold. Alternatively, most peak callers adopt a statistical approach to determine the quality of peaks, such that p-values or q-values derived from the enrichment ratio relative to the control are typically used to assess quality of peaks. For example, in MACS2, the read distributions in each window in each of the control and ChIP-seq samples are modeled as separate Poisson distributions. The overall expected number of reads per window is based on the expected number of reads in the control and ChIP-seq, which is derived from their corresponding Poisson distributions. Each genomic region, or window, is evaluated independently, where the null hypothesis states that enrichment levels of the control and ChIP-seq are similar. A window is considered to be a peak if the Poisson p-value is less than the default value of 10^{-5}, which can be modified based on user preference. The multiple test problem arises, however, as thousands of hypotheses, i.e. peaks, are tested simultaneously. This increases the probability of detecting a significant peak by chance, as the number of false positive peaks, regions claimed to be peaks but are not peaks, increases. Hence, most peak calling methods correct for the false discovery rate (FDR) [28] using the Benjamini-Hochberg correction method.

2.3.2 Challenges in Peak Calling

As with any complex experiment, there are still many sources of noise and bias in ChIP-seq experiments that may negatively impact downstream genomic analyses. Errors may arise from sequencing, structural, biological or technical variations. This may include: low binding affinity of antibody used; differential fragmentation of DNA due to the non-uniform chromatin structure; differential amplification of DNA fragments due to PCR amplification bias; differential mappability of reads due to GC content bias; library contamination due to flaws in the experimental protocol; or sequencing errors due to defects in the sequencing technology used. More details on the different sources of error can be found in Chapter 3. Moreover, each peak calling algorithm adopts a specific protocol for peak detection. Feature variations in protocols may drive a difference in peak calling performance [51]. These features may include whether signals are detected per nucleotide or per bin along the genome, or whether or not ChIP-seq and control signals are explicitly combined, or whether there is an option to select multiple alternate window sizes instead of being biased by one pre-selected window size, or whether or not the signal is modeled as a distribution with parameters estimated from ChIP-seq and control data, or the choice of statistical test used to score candidate peaks. Finally, it is important to note, that scientists do not know the true gold standard – meaning that the actual regions of enrichment along the genome are unknown. Peak calling techniques provide a means of estimating what these regions are.

2.4 Representing TF Binding Specificity

Different methods have been proposed for the experimental determination of the representation of TF binding preferences [220, 222, 255]. In-vitro data, such as PBM or SELEX, and in-vivo data, such as ChIP-seq, incur bound regions, where a TF bindings to the genome, and unbound regions, where a TF does not bind. A TF's sequence specificity

can be deduced from either data type, where bound and unbound sequences are used to build a model that accurately describes the DNA binding preferences of that TF. Both in-vitro and in-vivo are not without limitations, however. With in-vivo data, other factors in addition to the intrinsic binding preference of the TF may influence binding, such as chromatin accessibility and co-binding, while in-vitro data is generally known for its poor detection of low specificity binding sites, as it is difficult to detect the binding of a TF to a site at low concentration.

Most DNA-binding proteins, such DNA-binding TFs, bind to specific sequences in the DNA. Others, such as histones, bind to the DNA in a non-specific manner. A binding protein TF can bind to multiple genomic regions, where different sequences may have different binding affinities. Thus, to characterize the TF binding site specificity of the collection of sequences, a consensus DNA sequence or a binding site motif is generally used. The binding site motif captures the binding information for the set of preferred bound sequences [235, 236]. One consensus sequence is suitable for highly specific DNA interacting proteins [223], such as restriction enzymes, however, may not represent all the possible binding preferences of the binding protein.

A weight matrix (see next section) is one way for depicting sequence specificity of a collection of sequences and allowing variants of the consensus sequence [220, 222, 223, 235, 236, 255]. However, many other sequence specific models (see following sections) have been proposed to portray the motifs and account for the varying binding affinities and sequence degeneracy (similar sequences with same function under certain conditions) of protein-DNA interactions [220, 222, 235, 236, 255]. The models are then used for the search and prediction of potential TF binding sites in the genome. It is important to note that the although the models attempt to closely match the sequence composition of the known/experimental binding sites, mismatches are allowed. Model selection is based on a balance between the number of mismatches and precision of the representation.

Position Weight Matrices and Related Models To learn the sequence specificities of a DNA-binding protein, a commonly employed model is the position weight matrix (PWM) (also called the position specific scoring matrix [PSSM]) [235, 236, 265]. Sequence specificity of a TF is typically determined from an aligned set of bound sequences, and background frequencies derived from control or unbound data. The matrix is of size N by 4, where N is the length of the protein-DNA binding site and 4 represents the four possible nucleotides: adenine (A), guanine (G), cytosine (C) and thymine (T).

For every base position along the aligned set of bound sequences, a weight for each corresponding nucleotide is given, depending on the nucleotide frequency, also known as absolute entropy, or relative nucleotide preference, also known as relative entropy, at that position. Entropy refers to how random or surprising the choice of a base is at a given position of a TF binding site. More particularly, absolute entropy is based on the bound sequences alone, such that for each base position in the alignment:

$$AbsoluteEntropy(p) = -\sum_{c \in A,C,G,T} p(c) \log_2 p(c)$$

where $p(c)$ refers to the fraction of bases that equal A, C, G or T. Absolute entropy has a maximum value when all the events are equally likely. The more skewed the distribution, the lower the absolute entropy. Relative entropy, on the other hand, takes into account the base distribution in the bound and control/unbound sequences. The relative entropy for each base position in the alignment is:

$$RelativeEntropy(p_1, p_2) = -\sum_{c \in A,C,G,T} p_1(c) \log_2 \frac{p_2(c)}{p_1(c)}$$

where $p_1(c)$ and $p_2(c)$ are the frequencies of how often a specific base A, C, G or T appears in the control/unbound and bound sequences respectively. At each base position of the aligned bound sequences, the weights represent the log likelihood ratio of how different the actual observation is from the expectation defined by the control data.

PWMs assume independent contributions of the positions in the binding site to the total score. Probabilities at each position along a sequence S are calculated independently, then multiplied together to produce the total score for the sequence. Sequence logos can then be used to visualize the PWMs [218], where the rows correspond to the four possible nucleotide possibilities (A, C, G, T) and the columns represent the different positions in the sequence (See logo-like representations of motifs in the mutation maps in Figure 5.1). The overall height of a given column in a sequence logo depicts the information content of that position, or the absolute entropy.

Sequences are scanned using the PWM to detect potential binding sites. A suitable scoring method and threshold are needed to determine whether a certain subsequence matches the PWM. A simple strategy involves sliding the PWM across a sequence, scoring each site on both the reverse and forward strands of the sequence, and comparing the score of the site along the sequence to a threshold. More sophisticated motif discovery methods have been proposed. An outline of the such approaches can be found here [59, 153].

Although PWMs have the advantage of assigning position specific weights and not equating all mismatches, as well as being simple to interpret and visualize, there are some limitations to this method. PWMs fail to capture gaps. Suppose, for instance, a TF binds to the two motifs ATCNTCG or ATCNNTCG, where N represents any nucleotide. A single PWM can not model the sequence specificities of both motifs, due to the varying length of the motifs and the inconsistent subsequence length between the ATC and TCG parts of the motif. Gapped motifs are common, and may arise from TFs that bind as a homodimer or heterodimer of two proteins complexed together. Moreover, PWMs may be unable to portray the isoformic characteristics of TFs. TF isoforms, such as Pax-5, may result in differential DNA binding preferences, thus requiring a separate PWM for each isoform. The primary Pax-5 motif usually consists of the paired domain and the transactivation domain [150]. Pax-5 isoforms, however, may only bind to the paired and not the transactivation domain, such as Pax-5d, or have gaps within the paired domain, such as Pax-5e. Similar to gapped motifs, one PWM can not capture the specificities of all TF isoforms in this case. Other constraints include the independence of the base positions, and at times, the failure to detect subtle sequence signals, such as cofactor binding sequences [4]. PWMs provide a sufficient approximation of the truth specificity for some TFs [236], however it is not adequate for others.

More complex methods have been proposed to model the sequence specificities of a protein. For instance, to incorporate dependencies between positions, some methods account for the contributions of all the dinucleotides [222], instead of individual nucleotides, along the binding site. Another example involves the use of probabilistic modeling methods based on Markov networks to account for contributions of neighboring positions [220]. Sets of features, which represent certain properties, are given weights and their impact on the protein-DNA interactions is measured.

k-mer Based Approach Another approach is the k-mer based method. One example of a k-mer based approach involves the construction of a compact universal protein binding microarray (PBM), which accounts for all possible k-mer variants [30, 31]. The goal is to generate all possible k-mer sequences, calculate an enrichment score of binding affinity per sequence, and finally select k-mer sequences with the highest enrichment scores. Unlike PWMs, no assumptions about position independence or gap lengths are made [255]. Moreover, k-mer based methods can capture more subtle preferences in binding site composition than PWMs [31]. However, one main disadvantage is the generation of several k-mers per sequence per transcription factor, making the approach both computationally and memory intensive.

Other Complex Approaches The methods discussed in the previous two sections may underfit the binding data due to failure to detect low binding affinity sequences. Thus, due to the development of improved in-vitro and in-vivo technologies, more complex approaches have been proposed to generate more realistic, compact and accurate models of TF binding specificity.

Some of these methods may explicitly infer PWM parameters from input data. One way this can be achieved is by fitting models directly to binding intensity measurements. In MatrixREDUCE [77], for example, a linear relationship between the sequences and the binding intensities is assumed, where the sequence specificity is directly inferred from the binding data by fitting a statistical model to the protein-DNA interactions. However, the exact relationship between sequences and binding measurements remains unknown, and the impact of experimental techniques on such association is unclear. As such, another way is to fit models by ranking the binding intensity measurements instead. Seed-and-Wobble [31], for instance, computes enrichment scores for each possible motif pattern variant of length N, where N is the length of the sequence, using a modified form of the Mann-Whitney U statistic, selects the optimal seed and a collection of single mismatch variants to the seed, and finally constructs a PWM by combining the enrichment scores of the selected variants using a Boltzmann distribution. Another popular method is BEEML-PBM (Binding Energy Estimation by Maximum Likelihood for PBMs), where weighted nonlinear least-squares regression is used to predict the binding energy of TF-DNA interactions.

Other approaches use machine learning (ML) techniques, such as support vector machines [4] or deep neural networks [6, 11, 46, 92, 120, 197, 276, 285], to indirectly infer PWM parameters. Some methods implement the classification of protein-bound and unbound sequences to estimate a TF's sequence specificity model, while others use regression

for the direct prediction of binding intensity measurements from binding data. The use of ML to learn the mapping between DNA sequences and binding intensities/classes has proven to be more successful for the interpretation of complex binding site preferences. Deep neural networks will be discussed in more detail in the next section.

Thorough assessments conducted to compare the different methodologies proposed to learn the sequence specificities of binding proteins can be found here [173, 255]. While the chosen sequence specificity model, be it k-mer, PWM or diculeotide based, is important, the nature of the TF, evaluation metric, parameter estimation and algorithm configuration all greatly impact the nature and quality of the results obtained.

2.5 Deep Learning in Genomics

Deep learning has strengthened our understanding of genomics in fields such as precision medicine and drug design. It has allowed high dimensional sequencing data, such as RNA-seq, DNase-seq, or ChIP-seq, to be fully utilized at higher potential and increased interpretability. For instance, deep learning was used by DeepChrome to predict gene expression from histone modifications [225], DeepVariants to predict genetic variants from NGS data [194], and to predict drug response in cancer patients [216]. A review of deep learning approaches used in genomics can be found here [69, 128, 245, 271].

Deep learning has many advantages over traditional machine learning methods in modeling sequence specificity [11]. Firstly, traditional machine learning approaches require domain knowledge expertise for feature definition and selection. Unlike these approaches, deep neural networks can directly learn from the genomic sequence alone without the need for predefined features. This step can greatly impact model performance – being also computationally and time intensive when dealing with high dimensional data. Moreover, since neural networks take raw data as input, they are capable of learning a suitable representation of the data by transforming low-level features into more complex, abstract or high-level forms. They are able to capture non-linear relationships between base pairs in the sequence and cover wider and multiple genomic regions [138]. It is important to note that architecture engineering and finding a meaningful loss function in deep learning is as expensive computationally and time consuming as feature engineering.

There are various architectures of neural networks used in different applications depending on problem design and data type. A fully connected network is made of a series of layers. Each layer consists of neurons, where each neuron is connected to all the neurons in the preceding layer. Training a fully connected feed forward network on high dimensional data is a challenge, as the number of parameters may exceed the number of training instances, making the model more prone to overfitting. As such, other model architectures, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have been used to reduce model complexity, and therefore the training time and the risk of overfitting [118, 129, 138]. The various network architectures also have different weight sharing schemes. Fully connected networks assume independent input features (where each feature has a different parameter), CNNs assume that the same patterns can be derived from subsets of input features, and RNNs are more natural for sequential data.

2.5.1 Architecture

The high dimensionality of genomic data has led to the use of deep learning approaches, such as CNNs and RNNs, instead of fully connected feed forward networks [6, 11, 276, 285]. Here, we focus on deep learning approaches used for the prediction of TF binding sites and/or chromatin accessibility genome-wide. The prediction of chromatin accessibility involves the prediction of whether or not regions are chromatin accessible, while TF binding prediction predicts whether or not regions are bound by a specific TF. Methods can be based on only CNNs, such as DeepBind [6], DeepSEA [285] and Basset [120], or a combination of both CNNs and RNNs, such as DeeperBind [92], DanQ [197] and DeepGRN [46].

Convolutional neural networks

A CNN is a type of neural network commonly known for its application in image classification and computer vision due to its capacity to analyze spatial information [276]. The intuition behind CNNs is that the classification of an input depends on local features within the input, rather than the entirety of the input. For instance, to classify an image of a dog, the CNN detects different dog features, such as eyes and paws, and the relationship between these feature, to ultimately be able to detect various types of dogs regardless of where they appear in the image. Similarly, in the analysis of genomic sequences for the prediction of TF binding, a certain DNA motif may be present at any location in the sequence, and it might be one of several variant motifs that the TF binds to. This is referred to as the sequence specificity of a TF, as previously discussed in Section 2.4, which measures how strongly a TF binds to various genomic regions relative to unbound regions. Two-class image classification is analogous to the sequence specificity modeling problem, where a 2D image with three color possibilities is similar to a 1D genomic sequence with the 4 possibilities A, C, G and T [276]. A window of the genomic sequence can be seen as an image. As such, CNNs are able to learn the local and global features of genomic data and are thus used to model the relationships between TFs and raw genomic sequences.

The three main types of layers of CNN architectures include the convolutional layer, the pooling layer and the fully-connected layer. The convolutional layer, also known as a motif scanner, is the first and main building block of a CNN, which is used for pattern detection and feature extraction. Each convolutional layer has multiple filters or kernels which scans the raw genomic sequences. This is akin to scanning the sequences using multiple PWMs. The weight sharing strategy in CNNs enables the network to capture local patterns in the sequence data and learn a single filter for analogous motifs which may appear at multiple locations in the sequences. A non-linear activation layer, such as the rectified linear unit function (ReLU), is then applied to each convolutional layer. This is then followed by a pooling layer which operates on each filter independently. Pooling reduces the spatial size of the filter by reducing the number of parameters.

DeepBind is one of the first pivotal methods which used CNNs for the identification of protein binding sites in DNA and RNA sequences [6]. It used a single convolutional layer, and trained multiple single task models – one for each TF and cell line combination.

DeepBind outperformed all the other traditional sequence specificity algorithms (previously discussed in Section 2.1.4) on both in vitro and in vivo data, and it was able to learn motif patterns associated with the binding sites and derive new sequence motifs. Increasing the number of convolutional layers or the use of local pooling has no significant impact on performance [276]. However, increasing the number of convolutional filters increases the performance; as such we set the number of convolutional filters as a hyperparameter in our model. Other methods, such as DeepSEA [285] and Basset [120], used CNNs to uncover regulatory features from genomic sequences to predict chromatin accessibility along the genome. DeepSEA and Basset both used a 3-layer CNN in a multi-task learning framework, where each task corresponded to a cell line or tissue. Despite the technical differences, both methods were trained on chromatin accessible data to predict the impact of single-nucleotide variants on regulatory regions, such as DNase hypersensitive sites, TF binding sites and histone marks. Other applications, which use CNNs, include but are not limited to, the prediction of Hi-C contact maps [71, 82], DNA methlaytion [202], gene expression [14, 225, 287] and alternative splicing [106, 148].

Recurrent neural networks

Recurrent neural networks (RNNs) are another class of neural networks which model the temporal relationships of sequential data [47]. Due to their outstanding performance in processing sequential data, RNNs have been used in many challenging domains, such as natural language processing, speech recognition and language translation. RNNs account for the positions of elements in the sequence and capture interactions between distant elements in the sequence. RNNs also allow inputs and outputs to be different in terms of type and length. For problem domains like those mentioned above, they provide a better representation of high dimensional data than fully connected networks, as they reduce the number of parameters in the model.

The mechanisms of forward and backpropogation in RNNs are slightly different than what is traditionally used [47]. In forward propogation, RNNs scan the data from left to right, and are recurrent because they perform the same task for every element in the sequence. This is unlike traditional forward passes applied in other network architectures, which treat each element in the input independently. RNNs pass the activation from one time step (one element) to the next. Thus, when making the prediction $y[t]$, it not only uses information from input $x[t]$ but also uses activation from $x[t-1]$. The same set of parameters are used at each time step – for every nucleotide in the sequence. In a standard unidirectional RNN, information earlier in the sequence is only used. However, in 1997, Schuster and Paliwal proposed the use of bidirectional recurrent neural networks to account for prior and future elements in the sequence [219].

RNNs use backpropogation through time to calculate gradients – meaning in the back-propogation step, the sequence is scanned from right to left. The term gradients refers to the errors made while training the network. Similar to traditional backpropogation methods, RNNs take the partial derivatives (gradients) with respect to parameters, and then update the parameters using gradient descent. However, unlike traditional methods, RNNs obtain the loss for each element (time step) in the sequence, then find the overall loss by

summing the losses of all the elements. Thus, all elements in the sequence are treated as if they have the same loss.

In a way, RNNs appear to be advantageous over CNNs as they are able to transfer information through arbitrarily long sequences [47]. However, standard RNNs only capture local dependencies. They suffer from the "short term memory" problem and are unable to capture long term dependencies. Very deep neural networks, including RNNs, suffer from the vanishing gradient problem, where the gradient decreases exponentially with the number of layers. Thus, updating the weight parameters with small gradients makes the update insignificant and training more difficult. To tackle the vanishing gradient problem and allow RNNs to capture long term dependencies along the sequence, gating mechanisms, such as the gated recurrent units (GRUs) [53, 54] and long short-term memory (LSTM) units [99], are used. An internal state in the network is updated continuously allowing the network to memorize long range dependencies of elements along the sequence. These mechanisms regulate and manage the flow of information between nodes in the neural network.

Hybrid Architectures

The sequential nature of genomic sequences has led to the integration of CNNs and RNNs for the prediction of TF binding sites along the genome in many approaches, such as DeeperBind [92], DanQ [197], FactorNet [198], and DeepGRN [46]. CNNs extract the local short range dependencies between nucleotides from the genomic sequences, while RNNs, more specifically LSTMs, capture the long range dependencies at a motif level and thus make use of information across the entire sequence [92].

In an assessment comparing the different deep learning approaches, Zhang et al. show that approaches using CNNs outperform hybrid approaches on ChIP-seq data [277]. The evaluation of the models was based on scalability, usability, performance and motif prediction. Similar results suggesting that simpler architectures outperform more complex ones were obtained by another evaluation paper [283]. However, deepRAM show contradictory results of hybrid models outperforming all other models, evaluation here was based on only the AUC metric [245]. The authors suggest that the combinations of RNNs and CNNs improves performance over simple CNNs. The trend can be seen more clearly with an increase in training data size. However, the use of simpler models leads to better interpretability. In summary, no clear conclusion can be made about which architecture design is better as it depends mostly on problem design, data type and data size.

2.5.2 Data Representation

Beyond the deep learning architectures, there are other factors to consider when modeling a DNA sequence for the prediction of TF binding sites. These include, but are not limited to, the input and output data representation, the length of input and class imbalance. ChIP-seq and DNaseq-seq are typically used to derive TF sequence specificity models. Many approaches [6, 197, 198] have turned to the ENCODE Project Consortium [56] and

the Roadmap Epigenomics Consortium [35] to extract ChIP-seq and DNase-seq data to detect regions of enrichment and regions of chromatin accessibility.

How the input and output data are represented in the different approaches varies however. Some methods, such as DeepBind [6], MTTFSite [286], Basset [120] and Wnuk et al. [261], directly use the extracted ChIP-seq or DNase-seq peaks to produce the positive and negative instances. For example, in DeepBind [6], ChIP-seq peaks for a specific TF and cell type combination are considered positive sequences – these are sequences that are bound by the TF in that specific cell type. The negative sequences are then produced by shuffling nucleotides in the positive sequences, in such a way that dinucleotide frequencies are preserved. In DeepBind, a binary classification model is trained for each TF and cell type combination with equal numbers of positive and negative sequences each of length 101bp. Similarly, Basset [120] predicts chromatin accessibility, and thus uses DNase-seq peaks as positive instances for the prediction problem. However, unlike DeepBind, Basset is formulated as a multi-task classification problem for the prediction of chromatin accessibility across 164 output nodes or cell types. Peaks from different cell types are greedily merged together to form 600bp sequences. To do this, peaks are first extended from midpoint to form peaks of length 600bp. Peaks that overlap by at least 200bp are then merged together. The activity of a peak is equal the union of peak activity across all 164 cell types. Negative instances are sequences inaccessible in some cell types, but accessible in others. Wnuk et al. [261] also uses the Basset approach for data set construction and training. However, unlike Basset, the dataset produced is based on both DNase-seq and RNA-seq data. The model predicts chromatin accessibility while being implicitly informed about tissue or cell state through gene expression. A vector of gene expression values is concatenated to the output of the convolutional layers of the network, in a similar way as ChromDragoNN [165].

Another method for the construction of input involves the division of the genome into bins, where depending on the problem formulation bins that are accessible or bound are given a value of 1, and 0 otherwise. DeepSea [285] and DanQ [197], for example, bin the genome into non overlapping 200 bp intervals. Due to their multi-task formulation, each bin is associated with a 919 length binary target vector of chromatin features. Chromatin features include chromatin accessibility, TF binding and histone modifications. Only bins with at least one TF binding event are included in the analysis, making up 17% of the genome. The 200 bp bins are then extended by 400 bp in each direction. Other approaches have proposed the use of overlapping bins. For instance, to predict binding of various TFs in a specific cell type, FactorNet [198] bins the genome into 200 bp intervals with 50 bp increments. The data is balanced by randomly selecting an equal number positive and negative bins per epoch. ChromDragoNN [165] and Bichrom [231] adopt a similar binning approach to FactorNet; dividing the genome into 200bp and 500bp bins respectively with steps of 50bps for the prediction of chromatin accessibility in various cell types. Finally, Phuycharoen et al. [192] proposed the setting of bin size as a hyperparameter varying between 200 and 2000 nucleotides, instead of having a set value for bin size.

There can also be variations in the output data representation. For example, it can be in the use of quantitative [253, 285] versus binary genomic profiles [6, 92, 197, 198] to represent TF binding. Instead of predicting 1 or 0 for whether or not a genomic region

is bound, DeFine [253], for example, formulates a regression model for the prediction of real-valued TF binding intensities. Another instance is the use of a many-to-many deep learning framework to obtain a single nucleotide base resolution [15, 143] versus a many-to-one framework [6, 92, 197, 198, 253, 285].

Additionally, some of these approaches, such as FactorNet [198], ChromDragoNN [165], MTTFSite [286] and Leopard [143], have explored the integration of multiple datasets, such as ChIP-seq for TF binding sites, DNase-seq for chromatin accessibility and RNA-seq for gene expression, to increase their understanding of the dynamics of transcriptional regulation. Moreover, all the previous methods mentioned in this section have included the reverse complements of the sequences in their analysis. All methods have also used one-hot encoding representations of their input data, such that each nucleotide is encoded as a binary vector, with $A = [1, 0, 0, 0]$, $C = [0, 1, 0, 0]$, $G = [0, 0, 1, 0]$, and $T = [0, 0, 0, 1]$.

2.5.3 Regularization

An essential issue in learning is the construction of learning algorithms that perform well not only on the training data, but on new unseen test data as well. One way to achieve this is by understanding the bias-variance trade-off of the model [83]. An algorithm with high bias, for example, has a strong preconception of the training data, and thus will underfit, or not properly fit, the training data. Conversely, an algorithm with high variance will precisely fit the training data, and often leads to the model learning the noise and irregularities of the training data, or overfitting. In both cases of underfitting and overfitting, the model has high generalization error, meaning it performs poorly on new unseen data. Underfitting is often addressed by increasing the complexity of the model, while overfitting typically requires the use of regularization methods.

Regularization refers to the use of any methodology explicitly designed for the reduction of the generalization error [83, 131]. A commonly used regularization method is weight regularization, such as L1 or L2 regularization, where a penality, computed based on model weights, is added to the loss objective function. Penalizing the model decreases the model complexity by shrinking the weights of the model. Another popular strategy is dropout, which involves the probabilistic removal of some outputs from the layers of the network [96]. Other methods include constraining model weights to a specific range, adding statistical noise to training data, increasing the amount of training data via data augmentation, or early stopping – that is stopping model training when the validation performance starts to decrease. Overall, regularization methods may either reduce model complexity or increase the number of training instances to combat overfitting. One method, or a combination of methods, are usually adopted for the regularization of deep learning architectures for the prediction of TF binding sites.

2.5.4 Loss Function

The loss function is central to any gradient-based learning algorithm. Loss functions measure the difference between expected and predicted output; the goal is to minimize the

loss. The gradients are derived using the output of the loss function to update the model parameters/weights. In the case of binary or multi-task models, the majority of the proposed approaches, such as DeepBind [6], Basset [120] and DeepSEA [285], have trained their models under the maximum likelihood framework using either cross-entropy or negative log-likelihood loss. Both loss functions produce equivalent results. In PyTorch [183], a popular machine learning framework, applying cross entropy to the final layer is equivalent to applying negative log-likelihood to the final log-softmax layer [182]. The cost function for binary cross entropy is:

$$C = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\sigma(a_i)) + (1 - y_i)\log(1 - \sigma(a_i))]$$

where σ is the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$, C is the cross-entropy loss, N is the number of instances, y_i is the actual label of instance i, and $\sigma(a_i)$ is the predicted probability of instance i. Each predicted probability $\sigma(a_i)$ is compared to the actual true label y_i.

2.5.5 Class Imbalance

Imbalanced classifications problems are those where one class accounts for the vast majority of instances [75, 107, 108]. Since TF binding or chromatin accessibility cover a relatively small portion of the genome in any given cell type, there is potential for high class imbalance. Deep learning methods rely on optimization methods for learning. With high class imbalance, errors from the minority class have little to no impact on the output of the loss function, as the model is mainly focusing on the majority class. One way to deal with class imbalance is via weighing the instances [75, 107, 108]. All instances usually contribute equally to the overall loss. Here, however instances are weighed according to the class they belong to. In the simplest approach, instances that belong to the minority class are individually weighted higher than instances from the majority class, so that collectivity minority and majority classes have the same total weight. This technique works best with multi-class or binary classification problems. However, it is more difficult to apply to multi-task/multi-label problems. Another class balancing technique involves the oversampling of the minority class or the undersampling of the majority class.

Several methods have been proposed to mitigate the imbalance problem faced specifically in the prediction of TF binding sites or accessible regions along the genome. Some methods, such as Basset [120], DanQ [197] and FactorNet [198], rely on evaluation metrics such as the area under the precision-recall curve (AUPRC) to deal with class imbalance when training the model (see next section). Data used represents actual genomic data with roughly 1 bound/accessible to 1000 unbound/inaccessible regions. In contrast, DeepSEA does not address class imbalance and only uses the area under the receiver operating characteristic curve (AUC) to train and evaluate their model [285]. The authors of DeepSEA claimed that the class imbalance problem is dealt with by their multitask model. However, DanQ in future work invalidated their claim. Other approaches, such as DeepBind [6] and MTTFSite [286], circumvent the class imbalance problem as they generate the same

number of negative unbound examples as positive bound examples. With DeepBind, an unbound region is constructed for each bound region via shuffling the dinucleotides in the corresponding bound region. MTTFSite, on the other hand, extracts unbound regions with similar dinucletiode compositions as bound regions from actual DNA sequences – making sure that the unbound region do not overlap any of the bound regions.

However, having the same number of bound and unbound regions does not reflect the actual genomic nature of the data. The predictive model may have poor generalization performance. Additionally, the sole reliance on AUPRC is an insufficient solution for this issue. Consequently, alternative approaches including the upsampling of bound or accessible regions [165, 192, 261], which leads to data duplication, or the downsampling of the unbound regions [170], which leads to data loss, have been proposed. Another sampling strategy takes into account the instances selected per training batch – where only bound/accessible regions or only inaccessible regions are selected per batch [231].

2.5.6 Performance Evaluation

The prediction of TF binding sites or chromatin accessible regions along the genome is usually depicted as a binary classification problem; that is each instance could either be bound/accessible or unbound/inaccessible. Different evaluation metrics could be used to evaluate model performance. The selection of the evaluation metric greatly depends on whether the data is balanced or not.

In DeepBind, for example, the area under the receiver operating characteristic (ROC) curve is used [6]. The ROC curve is a probability curve that displays the relationship between the true positive rate (TPR, also known as sensitivity or recall) and the false positive rate (FPR) at various thresholds. Sensitivity examines the fraction of truly predicted bound regions (true positives) out of all the bound regions, while FPR is the number of incorrectly predicted unbound regions (false positives) out of all the unbound regions (false positives and true negatives).

The aim is to have a higher sensitivity and lower false positive rate. In DeepBind, the input data per model is balanced, meaning the number of positive (bound) and negative (unbound) instances are equal. As such, the area under the ROC curve (AUC) is a good metric to use. However, in case of imbalanced data, which is usually the case with the TF binding prediction problem, the area under the precision-recall (AURPC) curve is usually used [120, 197, 198, 286]. The precision metric measures the ability to identify only bound/accessible regions. Mathematically, it is defined as the number of true positives divided by the number of predicted positives. In AUPRC, both precision and recall deal with the positive class – which is the minority class in the TF binding site prediction problem. The AUC metric, however, examines the true negatives – which is the majority class, as well as the positive class. As a result, some methods address the class imbalance issue by using the AURPC metric when training the model, as described in the previous section.

2.5.7 Transfer Learning and Multi-task learning

Transfer learning is typically used to save significant amount of computation, or due to the lack of data available to train the whole model. It is the process of using knowledge acquired from training a model on one task to solve another related task [171, 178]. This is contrary to traditional learning, where learning is isolated and knowledge is neither retained nor transfered between models. In transfer learning, a base network is first trained on a general available dataset, such as CIFAR-100. The learned features from the pre-trained model are then transferred to another network which is trained on another related dataset. Thus, the learning of the new task depends on previously learned tasks. There are two main transfer learning strategies. One involves the use of off-the-shelf pretrained models, where a pretrained network without the final layer is used. The other involves the fine tuning of the pretrained network, where not only is the final layer removed but select layers in the pretrained network are retrained.

Multi-task learning (MTL) is a form of inductive transfer which enables simultaneous learning of tasks in a model for an improved generalization performance [43, 214, 258]. Parameter sharing of hidden layers across all tasks enables the model to learn a shared and better representation of the related tasks and reduces the chances of overfitting. MTL increases the sample size used for training, meaning that there is more data to work with, and thus assists the model in focusing attention on shared features that matter to all tasks. There can also be two main learning methodologies in the multi-task formulation: fully shared model versus shared-private model [43, 214, 286]. Fully shared indicates the use of one model to extract features from all tasks such that the tasks share the same parameters, while shared-private proposes the use of both an independent model per task as well as a shared model for all the tasks.

There are many applications of MTL in genomics. For example, in methods such as DeepSea [285] and DanQ [197], the prediction problem is formulated as a multi-task problem for the simultaneous prediction of 919 chromatin features, such as TF binding, chromatin accessibility and histone modifications, per genomic region. As mentioned in section 2.5.2, a value of 1 is indicative of a bound or accessible region, and 0 otherwise. FactorNet [198], which is an extension of DanQ [197], trains a model for each cell type to predict binding across various TFs. In another multi-task formulation for the same prediction problem, AgentBind predicts the binding of a TF across a variety of cell types [282]. Similarly, as previously mentioned, Basset predicts chromatin accessibility for multiple cell lines. Thus far, the methods discussed use fully shared learning formulations. MTTFSite, on the other hand, uses a shared private model for the prediction of TF binding across various cell types [286]. Using such a formulation, the model is able to learn common features as well as cell type specific features, for ultimately the prediction of binding sites in cell types with no data.

The scarcity and expense of data collection and acquisition has also led to combining transfer learning with MTL as an alternative to traditional learning methods in the prediction of genomic features, such as chromatin accessibility and TF binding sites, along the genome. For example, for a more cell type specific prediction of chromatin accessibility, ChromDragoNN [165] adopted a two-step training transfer learning strategy. Firstly, a

multi-task ResNet model, where the pretrained model is trained on one hot encodings of input DNA sequences to predict chromatin accessibility across various cell types. This is followed by the fine-tuning of the pretrained ResNet model – the pretrained model is used to train the second stage model for the prediction of chromatin accessibility in a specific cell type. The convolutional layers of the pretrained model are selectively retrained – the layers may or may not be frozen during training. Similarly, Novakovsky et al. [170] utilize a two stage learning process for the prediction of TF binding sites across multiple TFs. A multi-task model is first trained to predict binding across various TFs. Then, in the fine-tuning step, the weights from pretrained multi-task model are used to initialize single-task models for the prediction of binding for individual TFs. Unlike ChromDragoNN, which uses transfer learning to learn cell type specific chromatin information, the authors here use transfer learning to obtain improved performance and more refined features of the binding motif for the TF in question.

Transfer learning can also be used as a method of regularization in genomics – such that hidden variables of a larger dataset are used to regularize training of a smaller dataset. For example, in the prediction of the differential binding of the TF MEIS across three tissues, Phuycharoen et al. [192] first formulate the problem as a regression task, where a convolutional model is trained on DNA sequences to predict amount of binding across all tissues. Then, to regularize the model, the weights of the pretrained model are transfered to another model for the classification of differentially bound regions across the tissues.

2.5.8 Hyperparameter Optimization

Hyperparameters are adjustable training variables that require to be manually set with pre-determined values before training. Different problems require different combinations of hyperparameters. Finding the best set of hyperparameters is a crucial step when training a model as it can greatly impact performance. Hyperparameter tuning (or hyperparameter optimization) is defined as finding the best configuration of hyperparameters for a machine learning model that will result in the best performance on the validation set [103, 270]. In a neural network, hyperparameters consist of model design variables, such as the number of hidden layers and the loss function, and training algorithm variables, which determine how the model is trained, such as the learning rate and dropout rate.

Hypeparameter tuning is formulated as an optimization problem, where given a set of hyperparameters, the objective is to obtain the best hyperparameter configuration to help minimize the loss or maximize the accuracy. A naive approach is to manually select and test hyperparameters iteratively until exhaustion or satisfaction. However, the lack of reproducibility, inefficiency when dealing with high dimensional data, and need for expert knowledge, have resulted in methods for the automatic calibration of hyperparameters [33, 34]. Grid search, for example, involves training and testing every possible combination of hyperparameters from a grid of hyperparameter values. This approach, however, results in exponential growth – becoming exponentially computationally intensive (in terms of time and resources) as the number of hyperparameters increases. An alternative method is random grid search which selects random combinations of hyperparameters from the grid

of hyperparameters [33, 34]. This improves the efficiency as good hyperparameters are often found in fewer iterations than exhaustive grid search.

Automatic hyperparameter calibration methods have been used in the prediction of TF binding sites. The hyperparameters can include the kernel size (motif detector length), number of filters (number of motif detectors), learning rate, learning momentum, batch size, dropout rate and the initial weight scale (how the weights are initialized) [6]. DeepBind, one of the first deep learning methods for TF binding, used a random grid search hyperparameter tuning approach. Here, 30 sets of random hyperparameters were first selected. Each hyperparameter set was then evaluated using 3-fold cross validation on the training set. The score of the hyperparameter set was the average of the three validation performance values. The hyperparameter set with the highest cross validation score was chosen, and the model was finally trained on the entire training data with the best hyperparameters. Other approaches, such as DeepRAM [245], DESSO [269], ChromdragoNN [165], Phuycharoen et al. [192] and Biochrom [231], use a similar hyperparameter optimization approach to DeepBind. In some methods, such as DeepSEA and MTTFSite, it was unclear what hyperparameter optimization method was used.

Grid and random search are not always optimal in terms of hyperparameter selection. With random search, for example, finding the best combination is not guaranteed since not all hyperparameter combinations are tested. Additionally, both grid search and random search are uninformed about past hyperparameter selections – thus making them less efficient. As a result, a more guided and efficient method known as Bayesian optimization is used [41]. Here, prior belief about the objective function is incorporated. The prior is updated with samples drawn from the objective function to get a better posterior approximation – that is a better hyperparameter configuration. Models, such as Gaussian processes, random forests and tree Parzen estimators, are often used to approximate the objective function, and referred to as the "surrogate" models. A surrogate model balances between exploring unexplored areas of the hyperparameter space and exploiting areas with the best hyperparameter configuration thus far. In general, Bayesian optimization is able to keep track of the hyperparameter tuning history of the model, and examine past hyperparameter configurations to choose the next configuration. Basset [120] and AI-TAC [157] also used Bayesian optimization for hyperparameter selection.

2.5.9 Interpretation Techniques

Transparency and trust, as well as robustness and reliability, are all important factors to consider when dealing with learning algorithms [9, 67]. The deployment of deep learning algorithms in a wide range of applications, such as self-driving cars, medical prognosis and image recognition, has made the explaining of their predictions of vital importance. However, it remains difficult to understand the inner workings of deep networks due to their black box nature. In genomics, for example, deep learning is used for the prediction of TF binding sites along the genome; how and why these specific sites are selected, as well as what the motif patterns are, remains unclear. As a result, recent progress to explain model predictions has been made by the development of various interpretation

techniques. In this section, we explain the different types of interpretation approaches used to explain and validate that deep neural networks (DNN) are learning biologically relevant representations in the prediction of TF binding sites.

Convolutional filters

A common and simple strategy to understand what a CNN is learning is to visualize the convolutional filters. As previously discussed, convolutional filters detect spatial patterns in the input data. CNNs learn multiple filters simultaneously which convolve feature maps or input data from previous layers. Typically, in image recognition, producing feature maps is straightforward – it involves visualizing weights produced after applying a filter to an input image. In genomics, however, more specifically the prediction of TF binding sites, visualizing filters consists of more steps.

Many approaches, such as DeepBind [6], Basset [120], DanQ [197], FactorNet [198], TBiNet [180], Novakovsky et al. [170] and AI-TAC [157], have used the method of visualizing convolutional filters to discover motifs learnt by their deep learning approaches. The aim is to identify the learned motifs associated with the filters. As such, each convolutional filter is converted to a position weight matrix (PWM). One way to do this is to first select sequences that activate each filter. Next, these sequences are used to construct position frequency matrices and PWMs for each filter. A tool such as Tomtom [90] is then usually used to search for motifs corresponding to the PWMs from a database of known motifs, such as JASPAR [217].

Here, filters from the first convolutional layer are usually assessed – as they deal directly with input data. A CNN that consists of fewer layers is forced to learn more interpretable whole motifs in the first layers [124]. With deeper models, a more distributed representation of the sequence motifs is learned by learning partial motifs at each layer. Thus, the visualization of filters from deeper layers is challenging and not yet explored.

Although visualizing convolutional filters is beneficial, solely depending on convolutional filters to interpret what the network is learning is insufficient. Studies have shown that architectural choices, such as max-pooling and filter size, impact the sequence representation of the learned motifs [124]. For example, the larger the max-pool size, the better representation of motifs is learned. As such, different approaches such as attribution based methods have been proposed.

Perturbation-based attribution methods

Another paradigm in the interpretability of DNNs are perturbation-based approaches, where the input of a model is modified and the change in output is measured. Modifications in input depend on the data type in question, and could thus include perturbing nucleotides in a DNA sequence, modifying pixels in an image or words in a sentence. Perturbation was first introduced in the world of image recognition by Zeiler and Fergus [274], where they occluded parts of input images using grey masks to reveal parts of images important for classification. Since then, perturbation-based methods have been improved

[78, 79, 105, 207, 211] and applied to other fields, such as natural language processing and genomics. A recent survey investigating the application of perturbation-based approaches to different data types, such as video, image and text, is outlined here [105].

The difference in output between original and perturbed input assists in highlighting the parts of the input essential for prediction. If important parts of the input are perturbed, the output of the model is greatly impacted. Conversely, if non-essential parts are perturbed, the output of the model is unaffected. The idea is to evaluate the impact of small modifications in input on output.

A common challenge across domains, such as image classification, natural language processing and video recognition, is computational efficiency. As the number of dimensions increases, the more challenging it is to perturb all possible positions as well as different combinations of positions in the input [105]. As such, selecting the optimal set of positions with greatest impact on output can be difficult. Another potential issue is its susceptibility to saturation, meaning some changes in input may not be reflected in the output, indicating that some inputs are irrelevant for prediction when actually relevant [8, 9, 221]. For instance, suppose a neuron encodes a logical OR for several of its input features. For many of its input patterns, changing any one of the features has no effect on the output, such that all features are deemed irrelevant (for that feature vector), although this is clearly untrue. However, due to its advantageous nature in being model agnostic, reliable and less noisy than other approaches, such as gradient-based approaches [9, 105, 226], perturbation-based methods have been used in various applications [9, 105].

Here, we are particularly interested in the application of perturbation-based methods in genomics, more specifically to infer TF binding sites or regions of accessibility along the genome. In a method known as in-silico mutagenesis, the gold standard for interpretability in genomics, approaches such as DeepBind [6] and DeepSEA [285] computationally assess the impact of perturbing every nucleotide in an input DNA sequence on the prediction of TF binding sites or chromatin accessibility respectively. To then compute the effect of nucleotide substitutions on output, one could use the difference between probability scores of perturbed and original sequences [6], or the log2 fold change of odds between the two scores [285]. Additionally, mutation maps are used to visualize the effect of all possible mutations across an input sequence [285]. The $4 \times n$ mutation map, where n is the length of the sequence and 4 is the number of DNA nucleotides: (A, C, G, T), conveys how important each position along the sequence is by showing the increase or decrease in probability score [6, 285].

Gradient-based attribution methods

For comparison purpose, we will briefly discuss other attribution-based approaches referred to as gradient-based approaches. These methods rely on the gradient of the output with respect to the input to compute importance scores per nucleotide per input position for a given sequence. Thus, with a single forward and backward pass through the network, gradient-based methods are able to produce a nucleotide-resolution map of the importance scores of all the nucleotides across the entire sequence (similar to mutation maps in the

previous section). In this class of approaches, the different algorithms proposed, such as gradients to inputs (saliency maps) [224], guided back-prop [230], integrated gradients [238], Deeplift [221] and SHAP [151], differ in the way gradients are computed.

Global interpretability methods

Thus far, the attribution-based methods discussed focus on the local interpretation of the input, meaning it focuses on interpreting one sequence at a time to get perturbation or importance scores at individual bases along the sequence. A positive score suggests that the base contributes to the binding of the TF, while a negative score indicates that the base inhibits binding.

However, analyzing one sequence at a time may give an incomplete picture. Moreover, most gradient-based approaches, except for Deeplift, fail the sanity check – which tests whether importance scores produced by the various methods change when the input is randomized and the network output changes. This suggests that these scores may be biologically irrelevant at times [3, 123, 226]. Thus, there has been interest in analyzing all motif patterns across all sequences [9, 124, 125, 221]. This is known as global interpretability – where the goal is to understand the overall behaviour of the model. When examining the patterns produced by all input sequences, bias and noise can be then be accounted for.

An example of a global interpretability method, discussed earlier in this section, is the visualization of individual convolutional filters. As previously mentioned, most methods usually assess filters from the first convolutional layer of the network, so the distributed representations of the motifs are rarely accounted for. Additionally, however, a network with fewer layers is able to learn full motif patterns in the first layers of the network [124].

Visualizing convolutional filters in deep CNNs may lead to hampered interpretability, as redundant partial motifs are often found. As result, other methods such TF-MoDISco (Transcription Factor Motif Discovery from Importance Scores) [221] and GIA (global importance analysis) [125, 126] have been proposed. In TF-MoDISco, for example, the first step is to obtain importance scores per nucleotide per sequence using one of the gradient-based approaches. Next, TF-MoDISco identifies segments (referred to as seqlets) from the input sequences with high importance scores, and clusters the seqlets into groups with similar patterns of activity. Despite the advantageous nature of TF-MoDISco in motif discovery [15, 157, 165, 170, 221], it was not used for aim 3 due to its extremely high memory requirement of 250 GB. Another approach, GIA, relies on perturbation-based methods. It begins by inserting certain motif patterns in the input sequences, producing two groups of sequences: synthetic sequences (with the motif pattern) and original sequences (with no motif pattern synthetically inserted). After performing in-silico mutagenesis on the original and synthetic sequences, GIA then quantifies the change in prediction across all input sequences caused by inserting the motif. That is, it finds the average effect of inserting the motif across input sequences.

2.6 Recap of knowledge gaps

Deciphering the complex interplay of protein-DNA interactions along the genome is of critical importance. The advent of high throughput sequencing technologies and the development of advanced computational methods have increased our understanding of molecular processes genome wide. There are limitations to these methods and technologies, however. In this thesis, we address three main issues that arise in the study of transcriptional regulation, more specifically in the use of ChIP-seq data to identify DNA-protein binding sites along the genome. The objective is to systematically explore distinct patterns observed in ChIP-seq data, in terms of bias and noise, as well as TF cell type specificity and motif enrichment.

The first part of the thesis involves the study of the biases and noise associated with ChIP-seq experiments. ChIP-seq is a popular high-throughput genome profiling technology used in the study of TFs. As with any high throughput sequencing technology, ChIP-seq data is prone to noise and bias. Accounting for these artifacts is essential for obtaining high quality reproducible data, as not accounting for bias produces misleading output. One way to alleviate the noise and bias in the data is via the incorporation of control data sets in the ChIP-seq analysis [72, 91, 164, 213, 243, 272, 280]. Whether biases in controls and ChIP-seq data are the same is not known, however. Different types of bias exist in different ChIP-seq experiments. Many existing peak calling methods do not select controls that best model the background signal of a ChIP-seq experiment, or estimate the ChIP-seq background signals. The use of different controls for the same ChIP-seq experiment can produce inconsistent results. Depending on which controls are used, different aspects of the ChIP-seq bias may be accounted for. Some studies, such as BIDCHIPS [200], CloudControl [97] and AIControl [98], have shown that different ChIP-seq datasets can be biased in different ways. They showed that enrichment analysis was improved via the integration of multiple control datasets through regression. BIDCHIPS, for example, re-prioritizes peaks already identified by another peak calling method. However, it accounts for only five types of non-binding background influences and does not provide a method to call peaks on the combined control. AIControl, a peak calling method which is an extension of CloudControl, integrates a set of publicly available controls datasets and uses ridge regression to model the background signal. The need for users to input controls is removed. However, users may want to input their own controls, and AIControl does not accommodate for that option. Moreover, as the number of datasets increases in ENCODE, the option to allow for controls as input in a weighted peak caller becomes more imperative to better represent the newly available data. Consequently, in Chapter 3, I introduce a peak calling algorithm, Weighted Analysis of ChIP-Seq (WACS), which utilizes "smart" controls to model the non-signal effect for a specific ChIP-seq experiment.

Another aspect I explore in this thesis is the ability to uncover cell type specificity of TF binding from the ChIP-seq data. TFs may bind to different regions of the genome in various cell types or tissues. Mechanisms, such as variations in chromatin accessibility, alternative splicing or cooperative binding, may influence a TF's binding preference, leading to a DNA signature of differential binding. I develop a deep learning approach, called SigTFB (Signatures of TF Binding), to detect and quantify the degree of cell type specific DNA

signatures for hundreds of TFs. The interest is in binding that makes itself shown through some kind of DNA sequence preference. It is important to note that the deep learning approaches discussed in this chapter address a different problem from that of SigTFB. Many methods, such as DeepBind [6] and DanQ [197], solely focus on the optimization of the prediction of TF binding sites. In SigTFB, however, we are not interested in performance in terms of peak prediction as the input peaks are already experimentally measured. Instead, the interest is in developing a supervised learning system to identify sequence features for a specific TF to distinguish binding in one cell type versus another. Moreover, although some deep learning methods, such as MTTFSite [286] and Phuycharoen et al. [192], also explore the differential binding of TFs across cell types, their problem formulation and objectives fundamentally differ. MTTFSite, for example, defines shared non-unique cell type instances as bound regions that overlap by at least 100bp across cell types, while the remaining that do not overlap as cell type specific. In the case of SigTFB, the model is given all instances as input and learns to differentiate between non-specific versus cell type specific instances. Unlike other methods, the SigTFB model is able to discriminate between shared and unique motifs in cell types from only bound regions. In SigTFB, negative unbound instances in one cell type are bound in others, while for MTTFSite [286] and Phuycharoen et al. [192], negative instances are unbound regions in all cell types. Moreover, SigTFB conducts wide scale analysis, where all TFs in ENCODE with at least more than 2 cell types available are explored, whereas MTTFSite [286] and Phuycharoen et al. [192] investigate TFs in a total of 5 and 3 cell types respectively.

Finally, to further explain the biology behind a TF's cell type specificity, a wide scale motif enrichment and gene expression analysis of the TFs in question is conducted. The aim here is to identify unique patterns that drive cell type specificity in certain cell types versus others for a specific TF.

Chapter 3

WACS: Improving ChIP-seq Peak Calling by Optimally Weighting Controls

This chapter includes the contents of "WACS: improving ChIP-seq peak calling by optimally weighting controls" by Awdeh, Turcotte and Perkins published in BMC Bioinformatics 2021 [16]. WACS source code is available at https://github.com/aawdeh/WACS.

3.1 Author Contributions

Aseel Awdeh devised a method to tackle the noise and bias in ChIP-seq, wrote the software implementation, conducted all the experiments and wrote the manuscript, all with input and guidance from Theodore J Perkins and Marcel Turcotte.

3.2 Overview

Chromatin immunoprecipitation followed by high throughput sequencing (ChIP-seq), initially introduced more than a decade ago, is widely used by the scientific community to detect protein/DNA binding and histone modifications across the genome. Every experiment is prone to noise and bias, and ChIP-seq experiments are no exception. To alleviate bias, the incorporation of control datasets in ChIP-seq analysis is an essential step. The controls are used to account for the background signal, while the remainder of the ChIP-seq signal captures true binding or histone modification. However, a recurrent issue is different types of bias in different ChIP-seq experiments. Depending on which controls are used, different aspects of ChIP-seq bias are better or worse accounted for, and peak calling can produce different results for the same ChIP-seq experiment. Consequently, generating "smart" controls, which model the non-signal effect for a specific ChIP-seq experiment, could enhance contrast and increase the reliability and reproducibility of the results. We

propose a peak calling algorithm, Weighted Analysis of ChIP-seq (WACS), which is an extension of the well-known peak caller MACS2. There are two main steps in WACS: First, weights are estimated for each control using non-negative least squares regression. The goal is to customize controls to model the noise distribution for each ChIP-seq experiment. This is then followed by peak calling. We demonstrate that WACS significantly outperforms MACS2 and AIControl, another recent algorithm for generating smart controls, in the detection of enriched regions along the genome, in terms of motif enrichment and reproducibility analyses. This ultimately improves our understanding of ChIP-seq controls and their biases, and shows that WACS results in a better approximation of the noise distribution in controls.

3.3 Background

High throughput sequencing technologies help in uncovering the mechanisms of gene regulation and cell adaptation to external and internal environments [26, 112]. One widely used technology is chromatin immunoprecipitation followed by next generation sequencing (ChIP-seq). It allows the genome-wide investigation of the structural and functional elements encoded in a genomic sequence, such as transcriptional regulatory elements. The main goal of a ChIP-seq experiment is the detection of protein-DNA binding sites and histone modifications genome-wide in various cell lines and tissues. Many peak calling methods have been proposed for the identification of regions of enrichment (putative binding sites) in ChIP-seq data [24, 133, 189, 243, 259].

Every experiment is prone to noise and bias, and ChIP-seq experiments are no exception. While some read pileups correspond to regions of true enrichment, others may be a result of the distortion of the ChIP-seq signal. Biased or noisy datasets (with a high number of false negative or false positive peaks) negatively impact downstream biological and computational analyses [156]. Thus, accounting for both noise and bias is important. Existing peak callers generally account for noise by assessing statistical significance under some statistical model. Bias is a more complicated subject and is usually addressed explicitly only via some control data to which the ChIP-seq is compared. We return to the issue of controls shortly.

There are many sources of bias in a ChIP-seq experiment. In the experimental design, for example, the quality of the experiment is predetermined by antibody and immunoprecipitation specificity. Low sensitivity, resulting from poor affinity to the target protein of interest, or low specificity, from cross reactivity with other unrelated proteins, degrades the quality of a ChIP-seq experiment [135]. The fragmentation step may also introduce bias [162]. Prior to immunoprecipitation, the DNA-protein complexes undergo fragmentation. However, due to the non-uniform nature of the chromatin structure (DNA), some regions are more densely packed (heterochromatin) than others and are thus more resistant to fragmentation. Less densely packed regions (euchromatin) will undergo more fragmentation. Another source of bias is mappability, which is the extent to which reads are uniquely mapped to regions along the genome [117, 162]. In an ideal situation, long enough reads

are used such that there is higher coverage and uniformity in coverage. However, in practice, read length is short and there are "ambiguous" reads that map to multiple regions. Such multiply mapped reads can either be retained (creating ambiguous ChIP-seq signal) or discarded (creating empty, unmappable regions), with either choice creating a different sort of bias. GC content bias [29, 242], introduced by PCR amplification or sequencing, also results in imbalanced coverage of reads along the genome. For example, in PCR amplification, both GC rich and GC poor fragments are underrepresented in sequencing data [29]. These variations in coverage can have a significant impact on the results obtained.

Systematic and experimental biases hinder the full potential of ChIP-seq analysis. Thus, the quality of the input samples is important, especially in large scale analysis where low quality datasets have greater effects [156, 166]. Consequently, more than a decade after ChIP-seq was introduced, the ENCODE and modENCODE consortia developed a set of ChIP-seq quality control metrics and guidelines to produce high quality reproducible data [135]. The protocols address all the stages of a ChIP-seq experiment, as bias and noise may be introduced at various stages, such as experimental design, execution, evaluation and storage methods [162].

One essential step for the alleviation of bias is the incorporation of control datasets in ChIP-seq analysis. It assists in the selection of true enrichment binding sites from false positives. Controls, such as input DNA and IgG, attempt to minimize the effects of immunoprecipitation, antibody imprecision, PCR-amplification, mappability bias, etc., and thereby increase the reliability of the results. In the input DNA, using the same conditions as the original ChIP-seq experiment, the DNA undergoes cross linkage and fragmentation. However, no antibody nor immunoprecipitation is used [135]. For the IgG control, sometimes referred to as a "mock" ChIP-seq experiment, all the same steps and conditions as the original ChIP-seq experiment are applied. However, a control antibody (not specific to the protein of interest) is adopted to interact with non-relevant genomic positions [135]. DNase-seq and ATAC-seq are used to tackle open chromatin regions. According to ENCODE [135], the input DNA and IgG controls should have a sequencing depth greater than or equal to the original ChIP-seq experiment. Higher sequencing depth is recommended since input DNA signals represent broader genomic chromatin regions than ChIP-seq [135, 162]. Other crucial factors addressed by the protocols include, but are not limited to, biological/technical replicates and library complexity.

Many existing peak calling algorithms allow testing enrichment compared to a control [72, 91, 164, 213, 243, 272, 280]. Whether biases in controls and ChIP-seq data are the same is not known, however. None of these methods selects a control or estimates background signals. Depending on which controls are selected and their nature, peak callers can produce different results (i.e., binding site positions) for the same ChIP-seq experiment. The BIDCHIPS [200], CloudControl [97] and AIControl [98] studies have shown that different ChIP-seq datasets can be biased in different ways. They address different biases in different ChIP-seq datasets via the integration of multiple control datasets through regression to improve enrichment analysis. There are some limitations to these studies, however.

For example, BIDCHIPS [200] has the ability to re-prioritize peaks already identified by another peak calling method. However, only five types of non-binding background

influences are accounted for and there are no mechanisms for de novo peak calling based on the combined control [200]. The Hiranuma et al. [97, 98] studies prove the advantage of using more controls to model the background signal. In CloudControl [97], the controls are subsampled in their regression fit proportional to their weights. This then allows the single customized control to be used as input to any peak calling method. However, the downsampling of the combined controls may introduce noise into the control signal.

AIControl [98], a peak calling framework, is an extension of CloudControl [97]. It integrates a group of publicly available control datasets and uses ridge regression to model the background signal. This eliminates the need for the user to input controls. However, some users may want to provide their own controls, and this is not accommodated. Additionally, the number of datasets in ENCODE increases with time, so allowing controls as input in a weighted peak caller is important to represent the newly available datasets and newly explored cell lines.

In this work, we introduce a peak calling algorithm, Weighted Analysis of ChIP-Seq (WACS), which utilizes "smart" controls to model the non-signal effect for a specific ChIP-seq experiment. WACS first estimates the weights for each input control, without requiring the fine-tuning of any parameters. Using the weighted controls, WACS then proceeds to detect regions of enrichment along the genome. WACS is an extension of MACS2.1.1 (Model-based Analysis for ChIP-Seq) [280], the most highly cited open source peak caller. Our development of WACS based on MACS2 allows researchers to use the weighted approach within a peak calling method with which they are familiar, and which has many refined features. Fragment length estimation/detection, read shifting, candidate peak identification, and peak assessment remain the same, while the construction of the control via the weighted combination of datasets is different. To allow for potentially large numbers of controls, we restructure the code invisibly for better memory footprint. We also correct a hashing bug in the pileup-computing code of MACS2, which becomes especially important when we have high read depth and/or many controls. (This bug has subsequently been corrected in the main MACS2 distribution as well.)

We evaluate WACS on a large collection of 90 ChIP-seq datasets and 147 control datasets from the K562 cell line in the ENCODE database [56]. To establish generalizability and study performance in a less expansive setting, we also investigate WACS on 20 ChIP-seq datasets for each of the A549, GM12878 and HepG2 cell lines. (The terms ChIP-seq and treatment are used interchangeably throughout this chapter.) We compare WACS to MACS2, as WACS is based on MACS2. We also compare WACS to AIControl, as it is the only other weighted peak caller which intellectually selects its controls. The results demonstrate the importance of smart bias removal methods and the use of customized control datasets for each ChIP-seq experiment, as the amount of bias varies across different ChIP-seq experiments. In the investigation of downstream genomic analysis, such as motif enrichment and reproducibility, the use of weighted controls in WACS shows a significant improvement in peak detection in comparison with the pooled unweighted controls in MACS2 and weighted controls in AIControl.

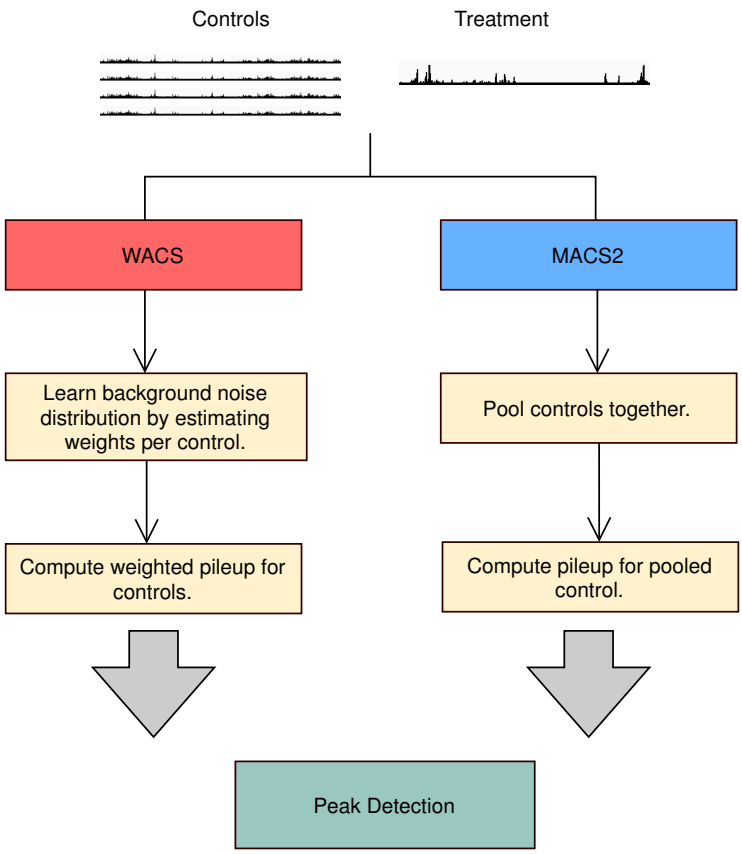


Figure 3.1: Flowcharts for WACS and MACS2. Both methods take controls and a treatment as input.

3.4 Results

3.4.1 WACS: A new algorithm for ChIP-seq peak calling with a weighted combination of controls

Our approach, WACS, estimates a background distribution by weighting controls, and ultimately identifies regions of enrichment along the genome (Figure 3.1 and 3.2). Below we describe the five major steps of the WACS algorithm. To implement WACS, we modified a well-known open source algorithm, MACS2. Because there is limited written description of how MACS2 works, we describe some parts of MACS2 to fully describe WACS. The WACS algorithm is summarized into two parts: Derive Weights (Algorithm 1) and Peak Detection (Algorithm 2).

3.4.2 Algorithm 1: Derive Weights

The control and treatment samples (in BAM format) are first preprocessed, as seen in Algorithm 1. Using SAMtools [142], we index, sort and optionally filter (remove duplicates

Algorithm 1 Derive Weights

Input: Control samples (BAM) and ChIP-seq sample (BAM)
Output: Weights per control

procedure DERIVEWEIGHTS
    Index, sort and filter the BAM files.
    Produce read counts per 200bp windows along genome.
    Normalize control and ChIP-seq counts by reads per billion.
    Compute control weights using non-negative linear least-squares regression

Algorithm 2 Peak Detection

Input Control samples (BAM), control weights, ChIP-seq (BAM)
Output Detected peaks

procedure PEAK DETECTION
    ▷ **Compute ChIP-seq pileup and associated values**
    Read treatment sample.
    Estimate/compute fragment length d based on treatment.
    treatmentPileup $\leftarrow$ Compute treatment pileup.
    $\lambda_{BG} \leftarrow treatReadcount \times d \div genomeSize$
    $LengthScales \leftarrow [d, 1kb, 10kb]$
    ▷ **Initialize control pileup to zero at each length scale.**
    for $< j = 1$ to $3 >$ **do**
        $ControlPileup[j] \leftarrow 0$
    ▷ **Read and accumulate each control.**
    for $<$ each control $i >$ **do**
        Read control i into $FixedWidthTrack_i$.
        ▷ **Loop over length scales.**
        for $< j = 1$ to $3 >$ **do**
            ▷ **Compute scale factor.**
            $sf \leftarrow d \div LengthScale[j] \times \left(\frac{treatReadcount}{controlReadcount_i}\right)$
            ▷ **Compute weighted pileup at this scale.**
            $currentPileup \leftarrow sf \times controlWeight_i \times$ BidirectionalExtendReads($FixedWidthTrack_i$, $LengthScales[j]$)
            ▷ **Add into growing control pileup.**
            $ControlPileup[j] \leftarrow ControlPileup[j] + currentPileup$
    ▷ **Find maximum control pileup.**
    $\lambda_{local} \leftarrow$ maximum of λ_{BG} and pointwise maximum of *ControlPileup* at three different length scales
    ▷ **Call peaks.**
    CallPeaks($treatmentPileup$, λ_{local})

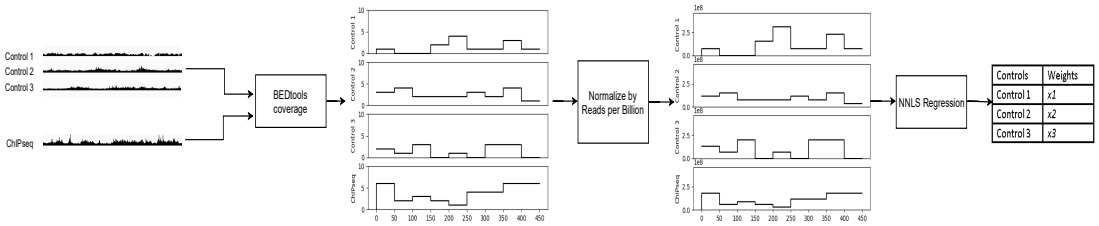


Figure 3.2: Flowchart for the estimation of weights per control.

from) the BAM files (line 2 in Algorithm 1). We then use BEDtools [199] to convert the BAM files of mapped reads into read counts per 200 base pair (bp) windows along the genome with 50 bp increments (line 3 in Algorithm 1).

Next, WACS normalizes the mapped reads per window for the preprocessed control and treatment samples. This ensures that the control and treatment samples are on the same scale. WACS applies reads per billion normalization to both the control and ChIP-seq samples (line 4 in Algorithm 1). For each sample m and window i:

$$n_{mi} \leftarrow r_{mi} \times 10^9 \div TotalReadCount_m$$

where r_{mi} is the read count in the window, n_{mi} is the normalized read count, and $TotalReadCount_m$ is the total number of reads in sample m. This effectively reproduces the normalization in MACS2, which linearly scales the control sample to the ChIP-seq sample. In what follows, we assume k total controls comprise samples 1 to k, and sample $k+1$ is the ChIP-seq data.

WACS then calculates the weights per input control (line 5 in Algorithm 1). WACS performs non-negative least squares (NNLS) to model the treatment dataset as a function of the controls. The overall objective of the regression is to find the values of the parameters (weights), that minimize the sum of squared differences between predictions and target values, with an additional constraint that allows only positive weights. Given n instances (windows), $y_i = n_{k+1,i}$ target values (one per window), $x_i = (n_{1i}, \ldots, n_{ki})$ feature vectors (one vector per window), a vector Θ of coefficient weights and a constant offset Θ_0, NNLS's objective function is:

$$\min_{\Theta,\Theta_0} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \Theta \cdot x_i - \Theta_0)^2$$
$$\text{subject to } \Theta \geq 0$$
$$\text{and } \Theta_0 \geq 0$$

To solve the NNLS regression we rely on the nnls module from scipy.optimize, part of the SciPy [250] package in Python. This produces a weighted control model for the treatment, with weights that indicate the relative importance of each control in modelling the treatment background signal. Zero weights are given to controls not required for modelling the treatment experiment. If there is one control, WACS and MACS2 produce the same output, as by default, the control in WACS gets a weight of exactly 1. The

controls can also be weighted by the user, instead of using NNLS to compute the weights of the controls.

3.4.3 Algorithm 2: Peak Detection

WACS is identical to MACS2 in its initial processing of the treatment sample, including: loading the mapped reads (line 2); estimation/calculation of fragment length d, which differs depending on whether the ChIP-seq reads are sequenced single-end or paired-end (line 3); and construction of the treatment pileup, which also differs for single-end or paired-end reads (line 4). Because these details have been described elsewhere, we do not repeat them here [73, 74, 280].

Where WACS differs substantially from MACS2 is how it reads in, processes, and combines the control samples. WACS reads the controls into memory one at a time, accumulating them into overall (weighted) control pileups at three different length scales: d, $1kb$ and $10kb$. The length scale is essentially the diameter of a Parzen-windows density estimator used to smooth the control reads. As each control is read in, it is smoothed, scaled so that its total reads are commensurate with the treatment, and further scaled by the control weight computed in Algorithm 1 (unless the user opts for unweighted controls). The function BidirectionExtendReads performs the actual smoothing, extending the read starts into intervals with diameter equal to the length scale. The smoothed and scaled control is added to the growing overall control at that length scale. In contrast, MACS2 reads all the control data in before beginning smoothing, which can create an unmanageable memory footprint when very many controls are being combined. Finally, WACS (as does MACS) creates an overall control pileup by taking the pointwise maximum of the "background" read density λ_{BG} and the control pileups computed at each length scale.

Finally, WACS calls peaks using the same mechanism as MACS2, which involves identifying candidate peaks and comparing the pileup heights at their summits with the control track. In the case of unweighted controls, WACS produces an identical control track to MACS2 and identical peak calls. However, when control samples are weighted differently, a different control track is produced and different peaks may be called. Each peak is associated with a p-value and a q-value, the latter accounting for multiple comparisons across the entire genome.

3.4.4 Duplicate removal.

Duplicate reads—multiple reads mapped to the same position on the genome—are often due to the overamplification of DNA fragments by PCR, which leads to the repeated sequencing of a DNA fragment. For WACS and MACS2, duplicate removal is optional. To produce more reliable peak calls, MACS2/WACS remove redundant reads at each genomic locus for both the treatment and control datasets [280]. The default number per genomic locus is determined by the sequencing depth, which refers to the number of reads mapped to a specific genomic location. However, when dealing with multiple controls, MACS2 performs duplicate removal after pooling reads. WACS does the same thing when used

in unweighted mode, for the sake of consistency with MACS2. In this case, apparent "duplicates" arising from different sequencing runs may be removed incorrectly, artificially flattening the control read distribution in high density areas. This phenomenon can be particularly prominent when hundreds of controls are being pooled. Thus, we recommend that users who want to perform de-duplication do so prior to feeding the mapped read files to MACS2 or WACS.

3.4.5 Average number of peaks per algorithm and average percentage overlap between algorithms.

To evaluate the performance of WACS with other methods, we downloaded ChIP-seq and control data for four cell lines: K562, A549, GM12878 and HepG2. For each ChIP-seq sample, we generated peaks under five conditions: (1) MACS2 with all the controls from the same cell line (All MACS2), (2) MACS2 with the matched ENCODE controls (Matched MACS2), (3) WACS with all the controls from the same cell line (WACS), (4) WACS with 10 randomly selected controls from the same cell line (WACS Random10) and (5) AIControl with its predefined controls (AIControl). We also used two methods to study the quality of peaks. "All Peaks" considers all the original peaks output by each method, whilst "Standardized" peaks normalizes the peaks output by each ChIP-seq sample by number of peaks and peak width. (See Methods)

In this section, we examine some basic statistics regarding the peaks generated by each algorithm and their corresponding pairwise overlap with the other peak calling methods. We focus on the K562 results in this and the following several subsections; results for additional cell lines are reported further below. This will help us understand how different the peak callers are. In Table 3.1, we report the average number of peaks output by each algorithm across the different ChIP-seq datasets for all peaks and standardized peaks.

We notice that AIControl outputs the largest number of peaks—over seven times as many peaks as WACS, and nearly 4 times as many as Matched MACS2. WACS outputs the smallest number of peaks on average. Matched MACS2 and All MACS2 output approximately the same number of peaks, and roughly twice as many as produced by WACS, while WACS Random10 generates a number of peaks intermediate between WACS and MACS2. For standardized peaks, however, all algorithms have the same number of peaks per dataset, which averages out to 12016.

In Table 3.2 and Table 3.3, we report the average percentage peak overlap between each pair of algorithms across the ChIP-seq datasets, for all peaks and standardized peaks respectively. More specifically, for every algorithm X (rows) and every other algorithm Y (columns), we computed the percentage of X's peaks overlapping any of Y's peaks for each of the 90 ChIP-seq datasets, and then averaged the percentages across the 90 datasets. When considering all peaks, for example, 27.1% of the peaks generated by WACS overlap with All MACS2 peaks. Most notably, less than 7% of the peaks generated by AIControl overlap with peaks generated by the other algorithms. For the other pairwise combinations, most overlaps are in the 30%-40% range. Conversely, in Table 3.3 for standardized peaks, we notice an almost symmetrical matrix with an increase in percentage overlap across all

algorithms, in comparison to Table 3.2. This is especially noticeable for AIControl, where approximately 25% of the AIControl peaks now overlap with peaks generated by other algorithms. All overlaps are in the range of 23%-43%.

The different number of peaks generated by each algorithm, and the resultant differences in percentage overlaps, highlight the importance of standardizing the peaks to remove the effect of the number of peaks in our analysis. Standardizing the peaks allows us to select the top quality peaks for comparison.

Table 3.1: Average number of peaks

Type	WACS	WACS Random10	Matched MACS2	All MACS2	AIControl
All Peaks	12457	17239	24422	26892	91113
Standardized	12016	12016	12016	12016	12016

Table 3.2: Average percentage of all peaks overlapping.

	WACS	WACS Random10	Matched MACS2	All MACS2	AIControl
WACS	-	37.3	29.0	27.1	29.0
WACS Random10	38.6	-	32.7	34.9	32.7
Matched MACS2	31.9	35.0	-	37.9	31.0
All MACS2	30.2	37.3	37.8	-	33.6
AIControl	6.5	6.3	5.0	5.5	-

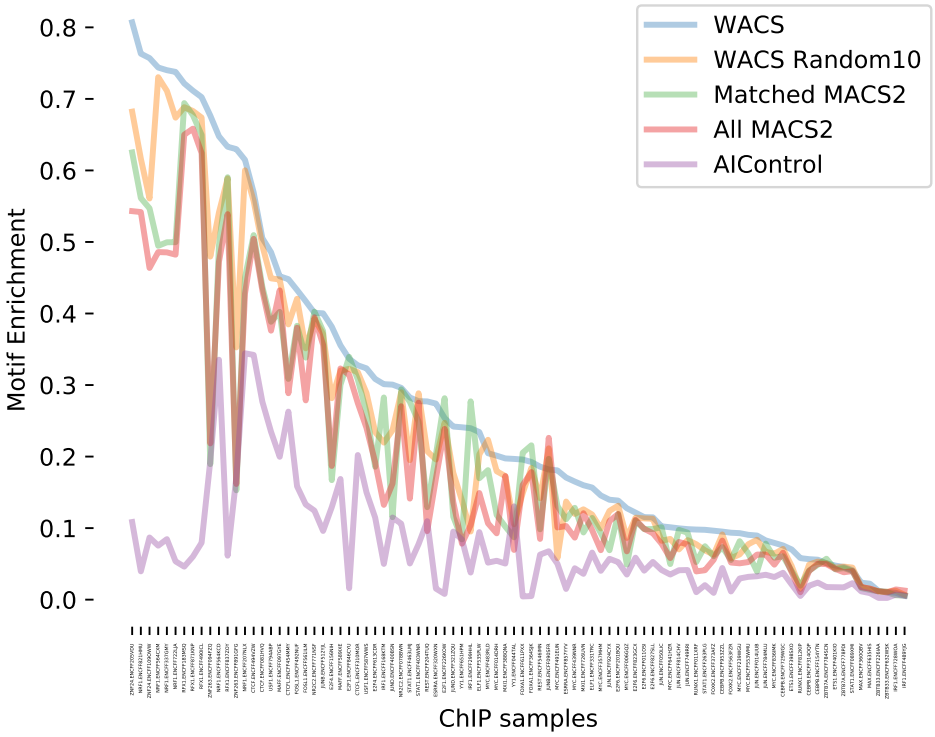
Table 3.3: Average percentage of standardized peaks overlapping.

	WACS	WACS Random10	Matched MACS2	All MACS2	AIControl
WACS	-	42.0	37.7	39.6	25.1
WACS Random10	42.0	-	39.5	42.7	25.0
Matched MACS2	37.7	39.4	-	41.7	23.5
All MACS2	39.6	42.7	41.2	-	24.3
AIControl	25.1	25.0	23.5	24.3	-

3.4.6 Peaks identified by WACS are more enriched for known sequence motifs.

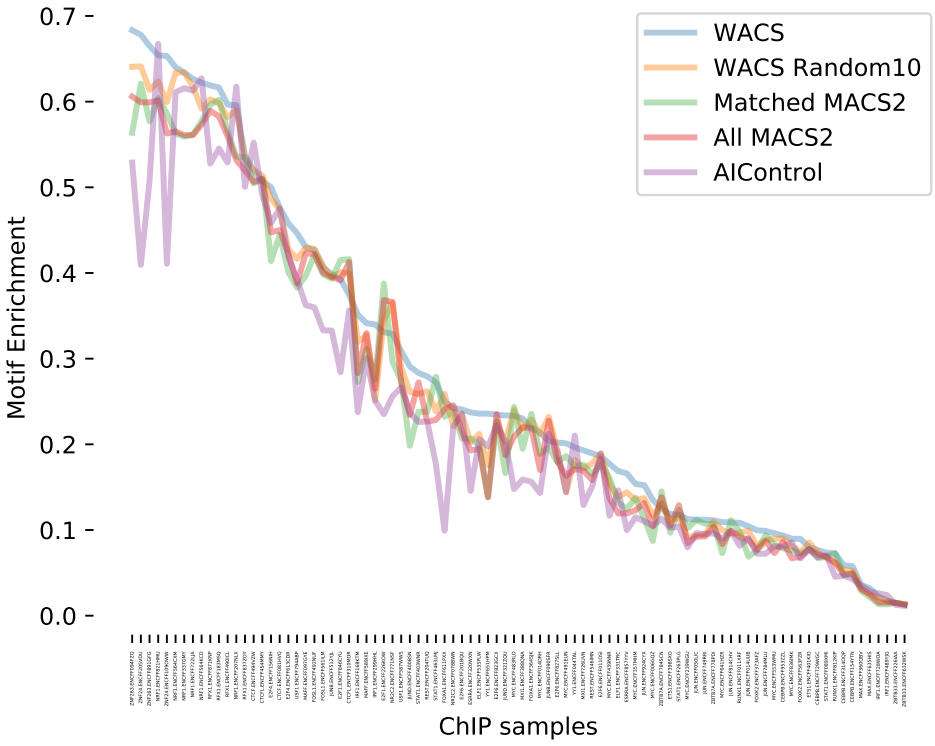
The purpose of ChIP-seq analysis is the identification of regions of enrichment, such as transcription factor (TF) binding sites, along the genome. Thus, DNA binding motifs for a TF tend to be enriched in genuine binding sites. To evaluate the performance of our method in comparison to MACS2 and AIControl, we performed motif enrichment analysis on the peaks. Adopting a similar method as in [98], we first used JASPAR to obtain position weight matrices (PWMs) for each unique TF [80]. Motifs in JASPAR are derived from in vitro assays, such as SELEX, and in vivo high throughput sequencing experiments, such as ChIP-seq or ChIP-exo [80]. (See Appendix Table A.5 for the PWM IDs per TF.) Using PWMs as input, we then used FIMO (Find Individual Motif Occurrences) [86] in the MEME suite [20] to scan the entire human genome GRCh38 and identify motif hits genome wide with a cutoff of 1e-5 to define significant matches. In our analysis, peaks with a motif are considered as true positives, while those lacking a motif hit are considered false positives. We quantify motif enrichment for a particular set of peaks as the precision, or equivalently the fraction of true positive peaks over total peaks.

Figure 3.3a and 3.3b display the motif enrichment for each of the 90 ChIP-seq datasets for all peaks and standardized peaks respectively, when using WACS (blue line), WACS Random10 (yellow line), Matched MACS2 (green line), All MACS2 (red line) and AIControl (purple line). The ChIP-seq datasets have been sorted so that the WACS performance decreases from left to right. An immediate observation is that some ChIP-seq datasets result in much more motif-enriched peaks regardless of peak caller, while others have much less motif enrichment. This may have to do with factors such as specificity of the TF's



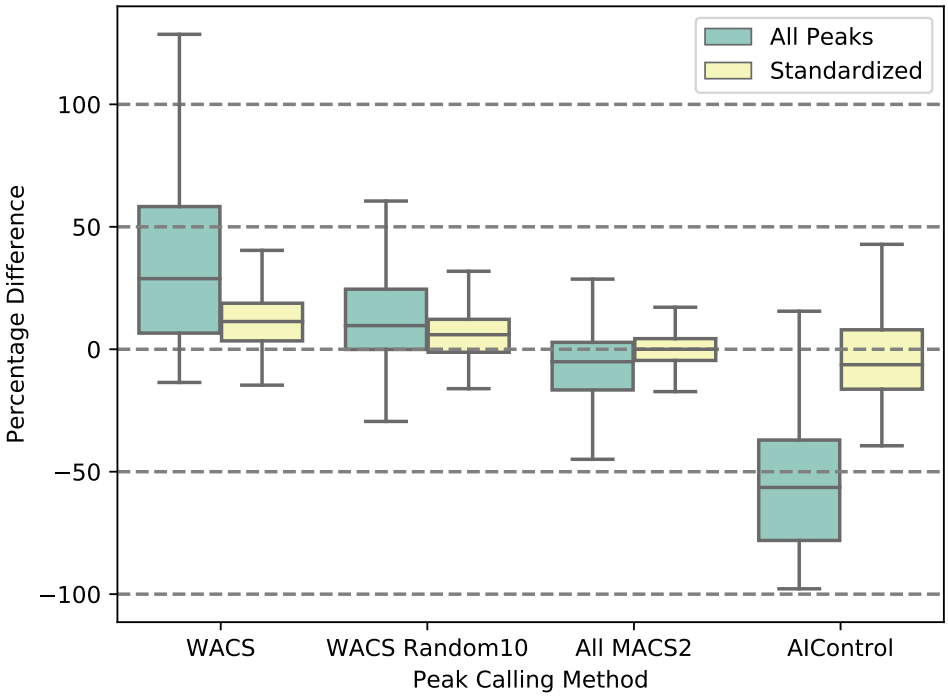
(a)

Figure 3.3: Motif enrichment of peaks found by five different peak calling approaches in 90 ChIP-seq samples. Motif enrichment is defined as the fraction of all peaks that contain at least one motif occurrence for the transcription factor in question. (a) Motif enrichment for all peaks.



(b)

Figure 3.3: (continued) Motif enrichment of peaks found by five different peak calling approaches in 90 ChIP-seq samples. Motif enrichment is defined as the fraction of all peaks that contain at least one motif occurrence for the transcription factor in question. (b) Motif enrichment for the standardized peaks.



(c)

Figure 3.3: (continued) Motif enrichment of peaks found by five different peak calling approaches in 90 ChIP-seq samples. Motif enrichment is defined as the fraction of all peaks that contain at least one motif occurrence for the transcription factor in question. (c) Distributions of percentages differences in motif enrichment relative to Matched MACS2. Box and whisker plots show the 0th, 25th, 50th, 75th and 100th percentiles.

Table 3.4: Number of datasets out of 90 where each algorithm's peaks show the highest motif enrichment, compared to the other algorithms.

Type	WACS	WACS Random10	Matched MACS2	All MACS2	AIControl
All Peaks	75	5	7	3	0
Standardized	61	8	10	4	7

DNA binding, the accuracy of the JASPAR PWM used for motif search, or the quality of the ChIP-seq dataset itself.

When analyzing all the peaks (Figure 3.3a), WACS is seen to outperform the other approaches the majority of the time—on 75 of 90 ChIP-seq samples in total. WACS Random10, All MACS2, and Matched MACS2 perform rather similarly, although we quantify this more carefully just below. AIControl performs the worst, with quite poor motif enrichment even in the datasets where all other algorithms perform very well. However, keeping in mind that AIControl tends to produce a large number of peaks, this could be a precision-recall sort of trade-off, in which the default behavior of AIControl is oriented towards the recall end of the spectrum. Indeed, when we examine the width- and number-standardized peaks (Figure 3.3b), the performance of all algorithms is much more similar. We still see a strong effect that some ChIP-seq datasets have peaks with much better motif enrichment than others. We also still see that WACS still performs best, although by a smaller margin and less often—it is the top performer in 61 of 90 datasets. Table 3.4 reports the number of times out of 90 that each algorithm's peaks show the best motif enrichment. By a proportion test, for either all or standardized peaks, WACS's fraction of times as the top performer is statistically significantly greater than the expected fraction of 1/5 if all five algorithms performed equally well, with a p-value of less than 10^{-5}. That WACS outperforms the other peak callers on the majority of the treatment samples even after standardization suggests that better motif enrichment is not a result of being more selective of the peaks, but that the peaks have inherently higher quality, at last as measured by motif enrichment.

To evaluate further the quantitative differences in motif enrichment, we computed the percentage differences relative to Matched MACS2—which is the method used by ENCODE and something of a "gold standard". Specifically, for each other algorithm and for each ChIP-seq dataset, we calculated the difference in motif enrichment, divided by the Matched MACS2 motif enrichment, and converted to a percentage. Figure 3.3c displays box plots of the percentages differences, for all peaks (green) and standardized peaks (yellow). For all the four methods, we observe that the standardized peaks in comparison to all peaks results in reduced dispersion and variability of the data. We will focus on standardized peaks in our discussion. For WACS, we notice a positive motif enrichment difference for most of the ChIP-seq datasets, with a mean improvement of 45% when all peaks are considered, or a more modest 14% when peaks are standardized. WACS Random10 also

shows improvements over Matched MACS2 on average, although they are not as large as the WACS improvements. Nevertheless, all four cases (WACS and WACS Random10 with all or standardized peaks) are statistically significantly greater than zero by one-sample t-tests, with p-values of less than 10^{-5}. All MACS2 performs similarly to Matched MACS2, as does AIControl when peaks are standardized, with none of the percent differences being statistically significantly different from zero. Without standardization, however, the full set of AIControl peaks is significantly worse on motif enrichment compared to Matched MACS2, with p-value less than 10^{-29}. Overall, these results again confirm the improved performance of WACS compared to other approaches, although standardization reduces its advantage.

Another method for evaluating motif enrichment is the area under the precision-recall curve (AUPRC) [98]. The AURPC is designed to compare algorithms on the same set of instances. Each algorithm, however, generates a different set of peaks for a specific ChIP-seq dataset. Thus, we believe precision is a more appropriate evaluation metric than AUPRC for this comparison. Nevertheless, for the purpose of comparison with AIControl [98], which uses the AUPRC metric, we performed the AUPRC analysis as well. Figure 3.4 shows an example precision-recall curve for the ChIP-seq dataset ENCFF109OWW with TF ZNF24, and Figure 3.5 shows the AUPRC for each of these ChIP-seq datasets when using standardized peaks. Using AUPRC, WACS outperforms WACS Random10, All MACS2, Matched MACS2 and AIControl on 73, 80, 78 and 81 of the 90 treatment samples respectively. These differences are statistically significant by a two-tailed sign test with p-value less than 10^{-5}.

3.4.7 Peaks identified by WACS are more reproducible.

Ideally, a ChIP-seq peak calling algorithm is able to reproducibly identify true regions of enrichment along the genome with no false positives. Reproducibility is most commonly measured by computing the percentage overlap of peaks between replicates [24, 133]. As described above, the K562 experiments we chose included exactly two ChIP-seq biological replicate samples in 45 distinct experiments (see Appendix Table A.1). Using the five different peak calling approaches, we called peaks for every sample, and evaluated the overlap between replicate samples. Overlaps means we took one replicate and computed the fraction of peaks that overlaps with the other replicate.

Figures 3.6a and 3.6b show the percentage overlap with all peaks and standardized peaks respectively for each of the ChIP-seq experiments when using WACS (blue line), WACS Random10 (yellow line), Matched MACS2 (green line), All MACS2 (red line) and AIControl (purple line). WACS has higher reproducibility than the other approaches on 26 of the 45 experiments when all peaks are considered, and on 28 of the 45 experiments with standardized peaks. These numbers are statistically significantly higher than expected under the null hypothesis that all algorithms perform equally, by a proportion test with p-value less than 10^{-4}. AIControl has lowest reproducibility of the five approaches, regardless of whether all peaks or standardized peaks are considered. See Table 3.5 for details on all five algorithms.

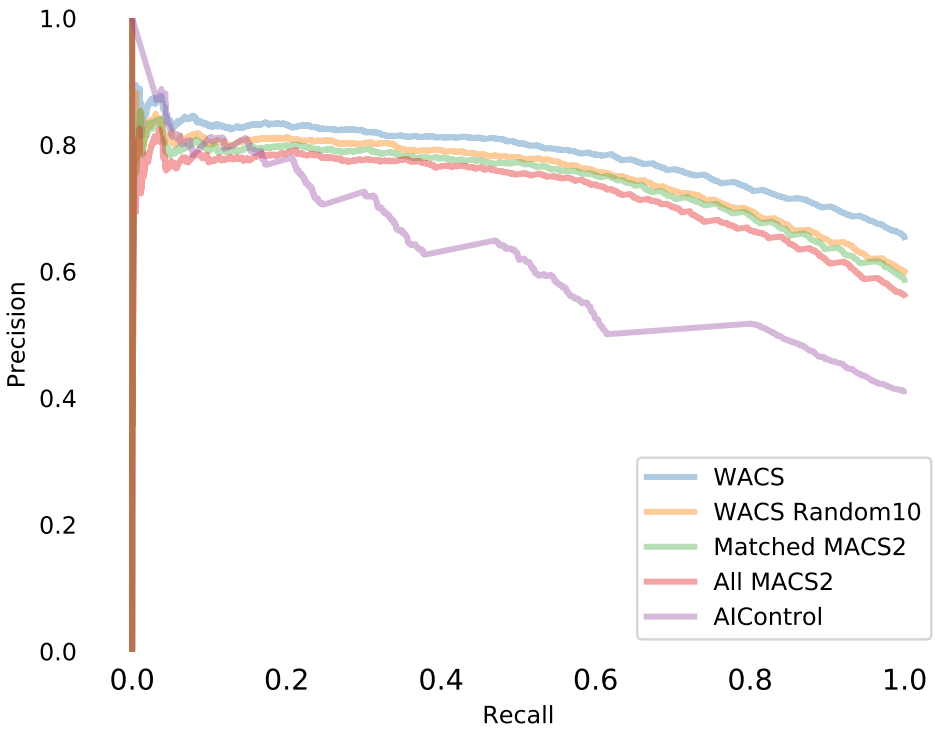


Figure 3.4: Example of precision recall curve for TF ZNF24 ChIP-seq dataset ENCFF109OWW.

Table 3.5: Number of experiments out of 45 for which each peak calling approach has the highest reproducibility between biological replicates.

Type	WACS	WACS Random10	Matched MACS2	All MACS2	AIControl
All Peaks	26	7	8	1	3
Standardized	28	7	5	4	1

To further investigate the quantitative differences in reproducibility, we computed the percentage differences in overlap relative to the overlap obtained by Matched MACS2. Figure 3.6c displays box plots of these percentage differences for all peaks (green) and standardized peaks (yellow). We notice a positive percentage difference in overlap for WACS, with 16% improved reproducibility for all peaks, or 5.6% for standardized peaks, on average. These differences are statistically significant by t-test with p-values of less than 10^{-3}. However, WACS Random10's performance is not statistically better than Matched MACS2, nor is All MACS2. AIControl has statistically significantly worse reproducibility both for all peaks ($p < 0.05$) and standardized peaks ($p < 10^{-12}$). The significance is borderline for the all peaks case, despite a large drop in mean, because of the high

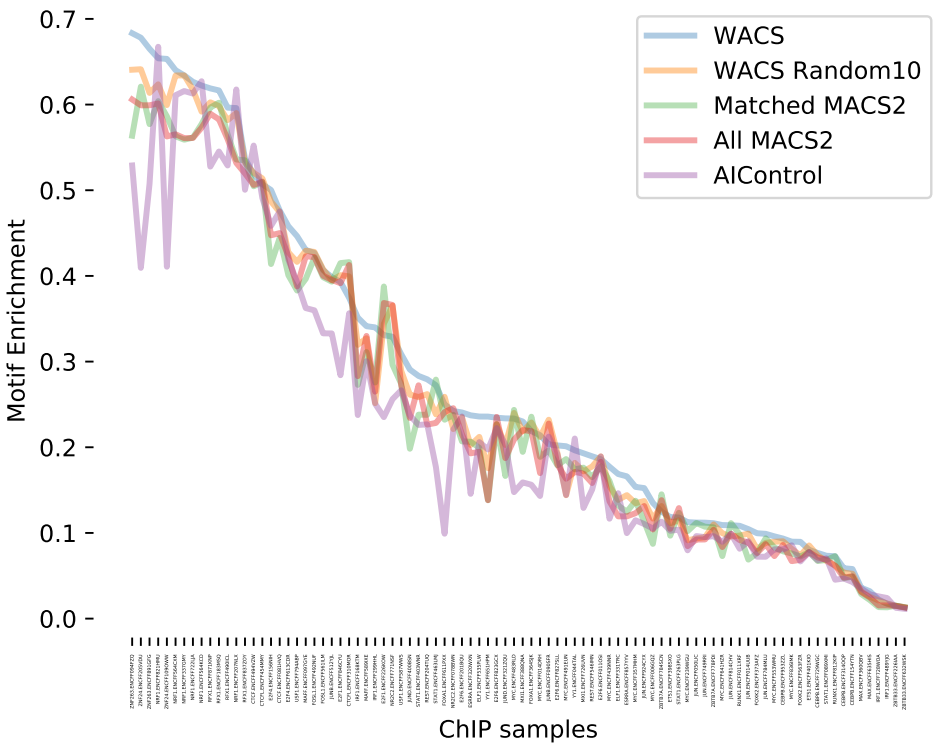


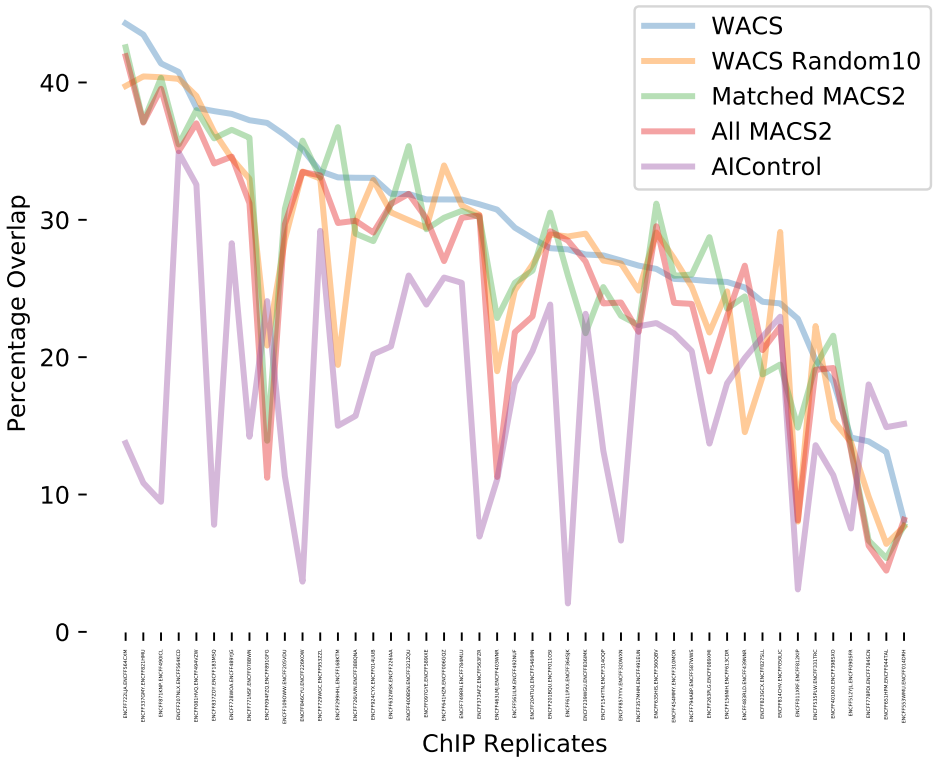
Figure 3.5: AUPRC for the K562 treatment samples.

variability in its performance. Thus, in this section and the previous section, we see compelling evidence that WACS produces higher quality peaks than the other approaches, as measured by both motif enrichment and reproducibility between replicates.

3.4.8 Controls used per treatment sample.

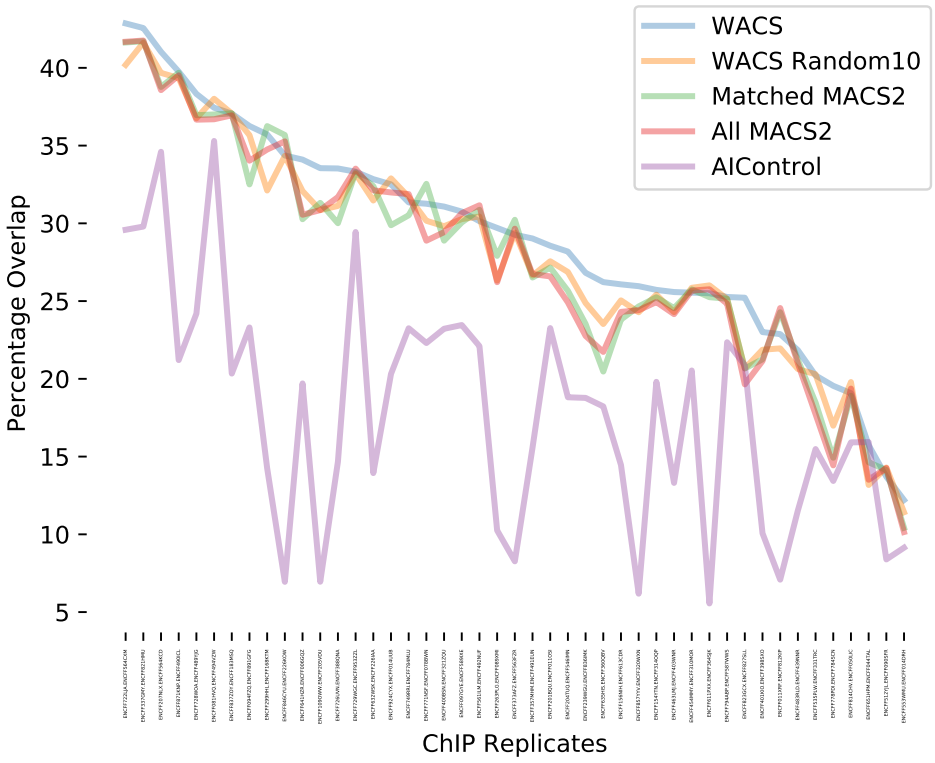
Our results (and other results [97, 98, 200]) for motif enrichment and reproducibility analysis suggest that smart controls offer superior background subtraction and peak-calling for ChIP-seq data. However, the standard practice remains to generate controls alongside each ChIP-seq experiment, or to match them on the basis of experimental details, such as cell/tissue type, read length and sequencer. If smart controls are to be used, it is unclear how many controls should be considered, and how many will end up in the smart control. It is unclear whether ENCODE matched controls are, in fact, the best choices or even among the controls selected by a smart control procedure.

Here, we aim to increase our understanding of the smart controls used to model the background signal. Figure 3.7 displays a matrix where the rows and columns represent the ChIP-seq and control datasets respectively. The blue color in the matrix represents the controls selected by WACS to fit each ChIP-seq dataset, the maroon color represents the ENCODE matched controls [56] and the magenta color represents the controls selected by both ENCODE and WACS.



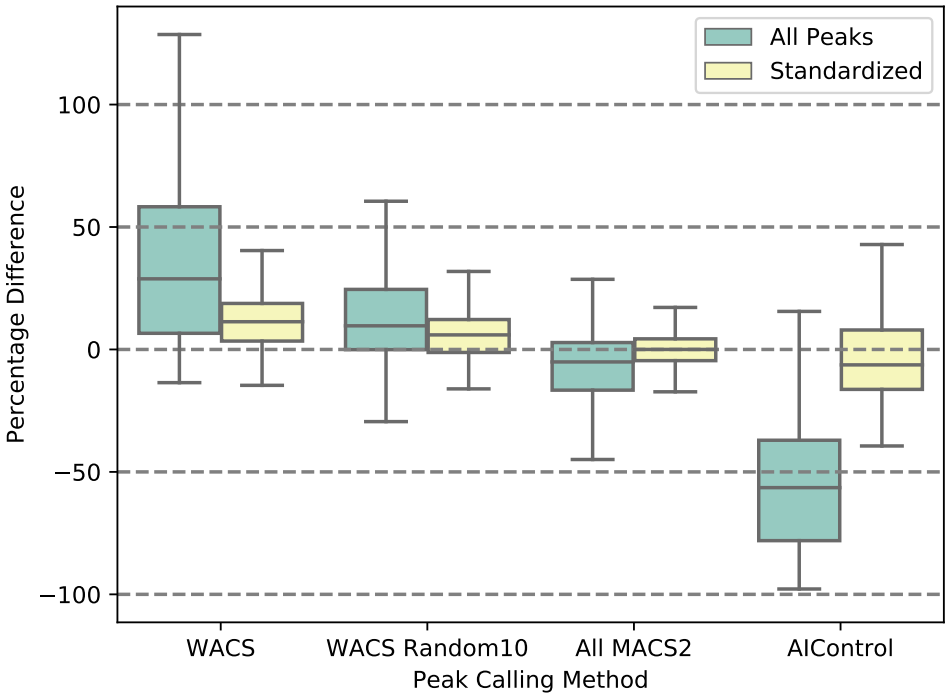
(a)

Figure 3.6: Reproducibility of peak calls between biological replicates. (a) Percentage overlap between replicates, for each of the five peak calling methods for 45 ChIP-seq experiments, when using all peaks.



(b)

Figure 3.6: (continued) Reproducibility of peak calls between biological replicates. (b) Percentage overlap between replicates, for each of the five peak calling methods for 45 ChIP-seq experiments, when using standardized peaks.



(c)

Figure 3.6: (continued) Reproducibility of peak calls between biological replicates. (c) Box plots of percentage difference in reproducibility relative to Matched MACS2.

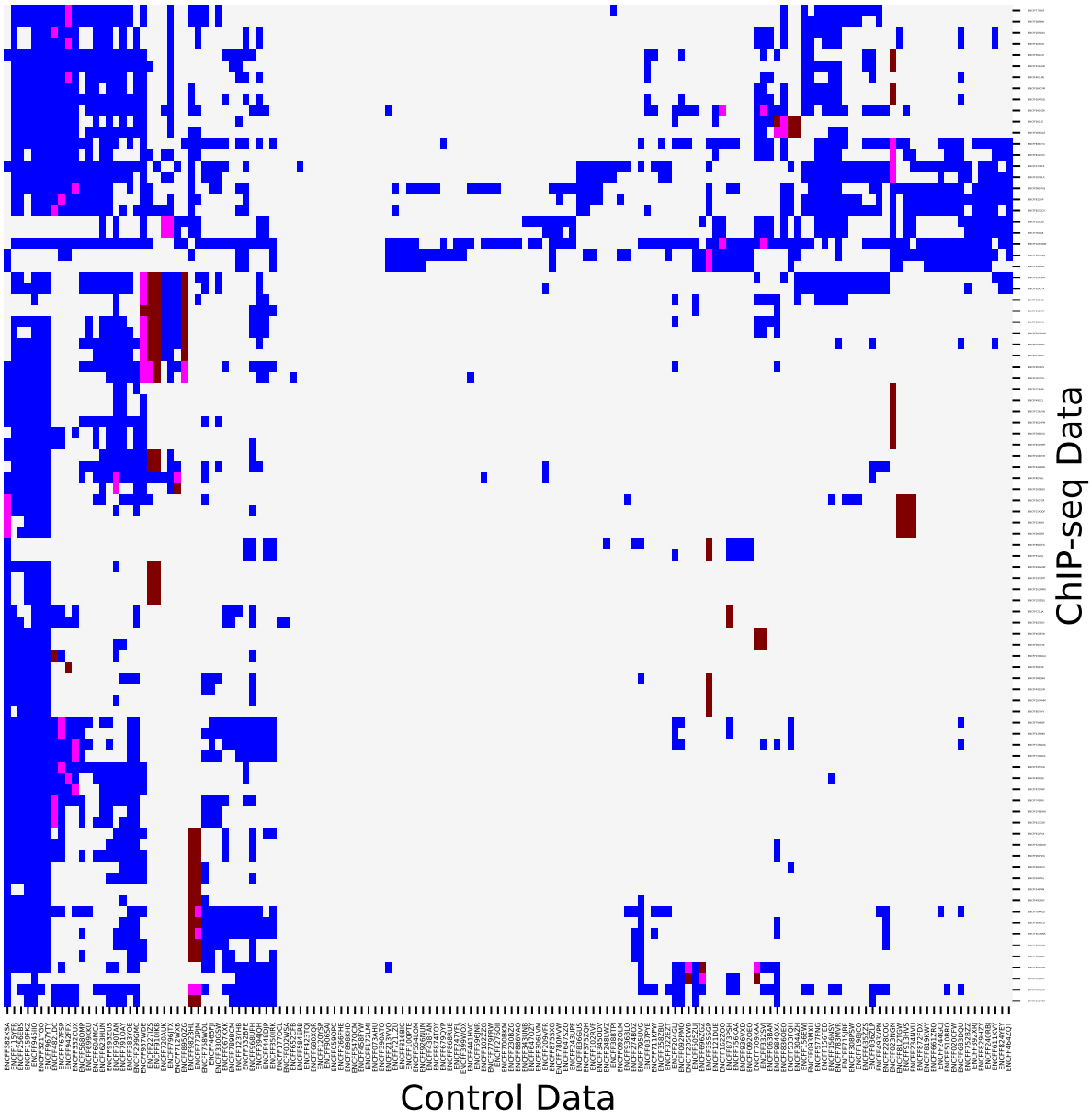


Figure 3.7: Comparison of controls used by WACS and ENCODE. The rows and columns correspond to the ChIP-seq and control experiments respectively. For each ChIP-seq dataset, the controls are given a *blue* color if they are used by WACS only, a *maroon* color if they are ENCODE matched controls only, and a *magenta* color if they are used by both ENCODE and WACS.

Let us first consider the WACS selected controls per ChIP-seq dataset (blue) in Figure 3.7. Different subsets of the 147 controls are required by WACS for each ChIP-seq dataset, but these form several coherent clusters, where groups of ChIP-seq datasets use relatively the same controls for modeling the background signal. For example, the 10 or so controls most towards the left of the diagram are used in modeling nearly all the ChIP-seq datasets'

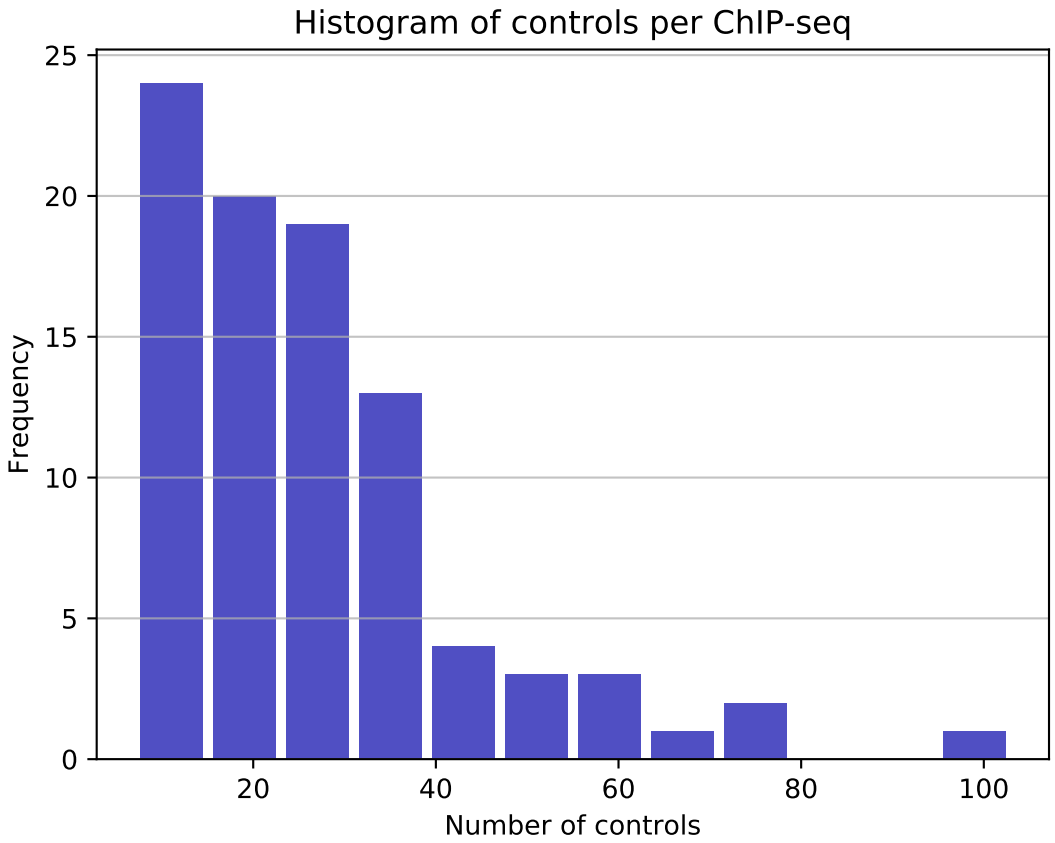


Figure 3.8: Histogram of the overall number of controls used per ChIP-seq dataset using WACS.

backgrounds. The next 10 controls are widely used, though less so, and are distinctly used by some of the ChIP-seqs towards the top. Conversely, there is a set of ChIP-seq datasets about near but not quite at the top of the matrix that rely on a large number of controls for modeling their background, whereas ChIP-seqs in the lower half rely almost solely on the leftmost controls.

Although each ChIP-seq's background is modeled by a unique combination of controls, a clear trend is that many controls are combined—approximately 26 on average. Figure 3.8 shows a histogram of the overall number of controls used by the ChIP-seq datasets using WACS.

For the ENCODE matched controls, we observe a range of 1 to 4 ENCODE matched controls per ChIP-seq dataset (maroon color in Figure 3.7). For 40 of the 90 ChIP-seq datasets (44%), none of the matched ENCODE controls are used to model the background signal in comparison to those used by WACS (rows with no magenta color in Figure 3.7). For example, 19 controls are used to model the background signal for the ChIP-seq dataset ENCFF651HPM in Figure 3.7, none of which are the matched ENCODE controls. For the remaining 56% of the ChIP-seq datasets, some of the ENCODE matched controls are also those selected by WACS, as seen in Figure 3.7 (magenta color), and there are 30 ChIP-

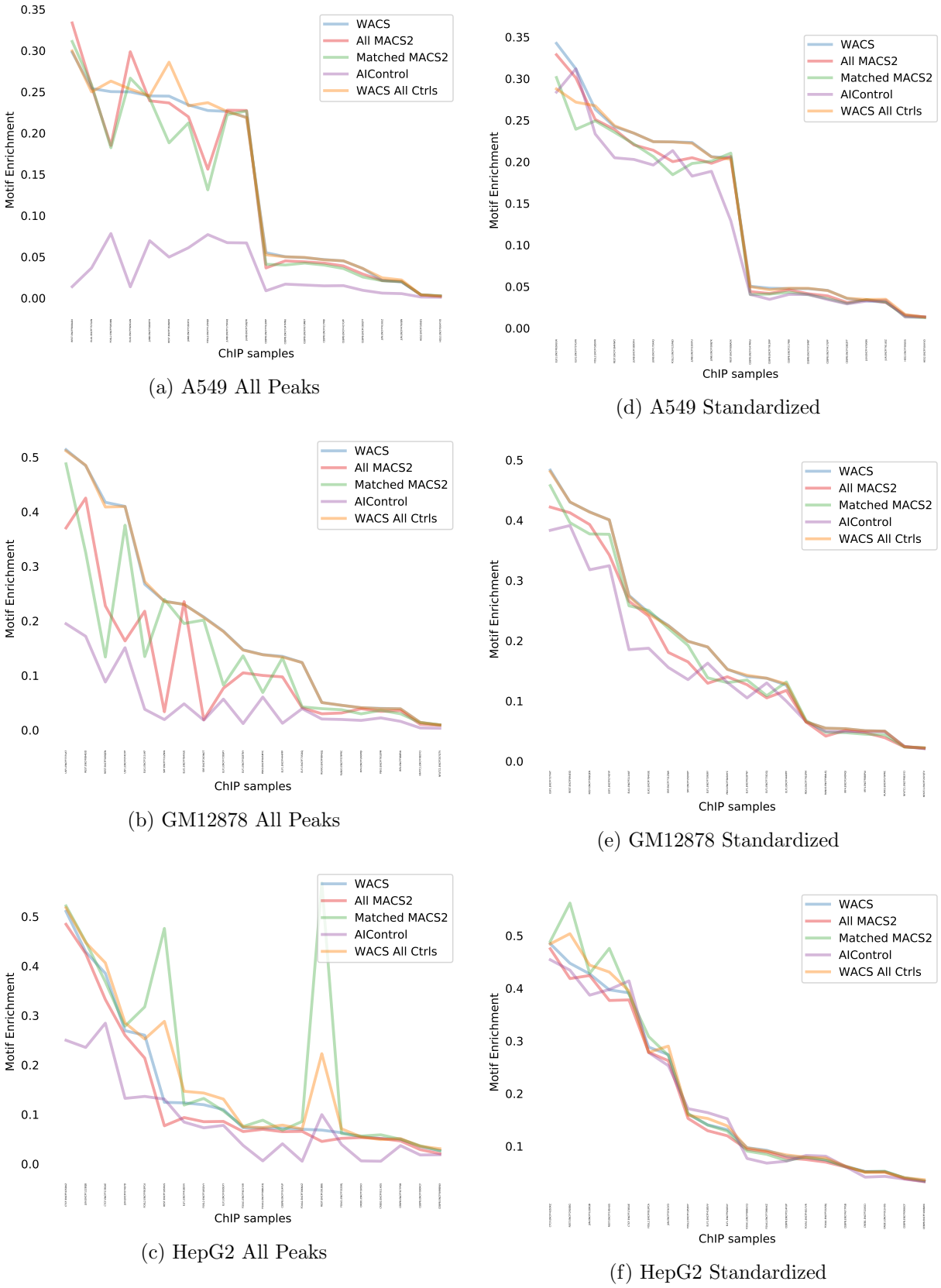
seq datasets that use all their matched ENCODE controls (in addition to other controls samples). It is not clear from manual examination nor straightforward statistical analysis what features of a control, or jointly of a control and a ChIP-seq dataset, might cause the control to be desirable for inclusion. Determining the distinguishing characteristics of the best controls for a given ChIP-seq, beyond their utility in our regression formulation, is an important topic for future research.

Additionally, we further investigate which features resulted in the inclusion or exclusion of a control by WACS for a specific ChIP-seq dataset. An instance is defined as each control and ChIP-seq dataset combination, and the target value is a boolean which indicates whether that control was selected for that specific ChIP-seq dataset. For each instance, we consider boolean features representing the similarity or difference between the ChIP-seq and control datasets. These include lab name, experimental release year and mapped read length. A value of 1 indicates that the feature is equivalent for both the ChIP-seq and control datasets, and 0 otherwise. We conduct an exact Fisher's test and found statistically significant results for each of these features with $p < 0.005$. (See Appendix Table A.7, A.8, A.9). However, these predictions are far from perfect, and future work needs to be conducted to establish what a 'good' control is.

3.4.9 Validation on additional cell lines

Here, we further evaluate WACS, MACS2 and AIControl on three other cell lines: A549, HepG2 and GM12878. We specifically explored 20 ChIP-seq and 18 control datasets for each cell line. (See Appendix Table A.4, A.5 and A.6 for accession codes of the samples.) We evaluated MACS2 with the ENCODE matched controls (Matched MACS2), MACS2 with the cell line specific controls (All MACS2), WACS with the cell line specific controls (WACS), WACS with the all controls across the three different cell lines (WACS AllCtrls), and AIControl with its predefined set of controls on ChIP-seq datasets (AIControl).

To evaluate the quality of the peaks generated by each method for each cell line, we first investigate motif enrichment. Figure 3.9 displays the motif enrichment for all and standardized peaks for each of the ChIP-experiments corresponding to each cell line, when using WACS (blue line), WACS AllCtrls (yellow line), All MACS2 (red line), Matched MACS2 (green line) and AIControl (purple line). AIControl across all cell lines, for all and standardized peaks, has the lowest motif enrichment. For the cell line A549, as seen in Figures 3.9a and 3.9d, WACS and WACS All Ctrls display the highest motif enrichment and have very similar performance. WACS and WACS All Ctrls outperform Matched MACS2, All MACS2 and AIControl on 14 treatment samples in total, as shown in Table 3.6. An equivalent trend is observed for the GM12878 cell line (Figures 3.9b and 3.9e). However, when using all peaks, WACS has the highest motif enrichment; WACS outperforms WACS All Ctrls, Matched MACS2, All MACS2 and AIControl on 15 treatment samples in total, as shown in Table 3.6. Additionally, for standardized peaks, for cell lines A549 and GM12878, we notice almost equivalent motif enrichment when using All MACS2 and Matched MACS2. For HepG2 with all peaks (Figure 3.9c), on the other hand, Matched MACS2 outperforms WACS, WACS All Ctrls, All MACS2 and AIControl on 11 treatment



(a) A549 All Peaks

(b) GM12878 All Peaks

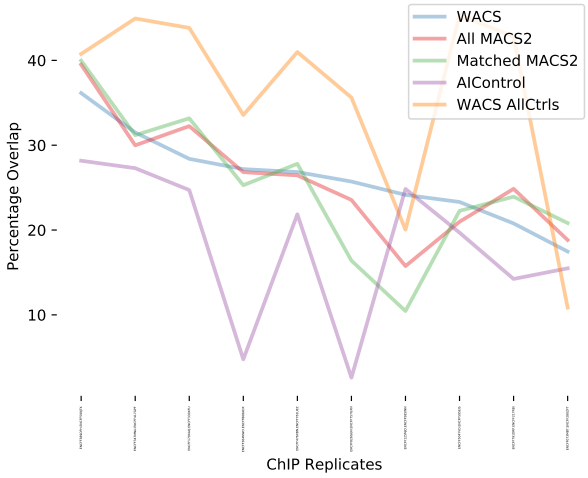
(c) HepG2 All Peaks

(d) A549 Standardized

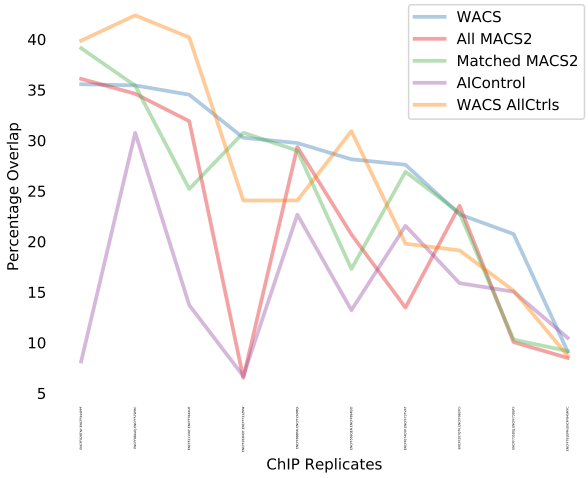
(e) GM12878 Standardized

(f) HepG2 Standardized

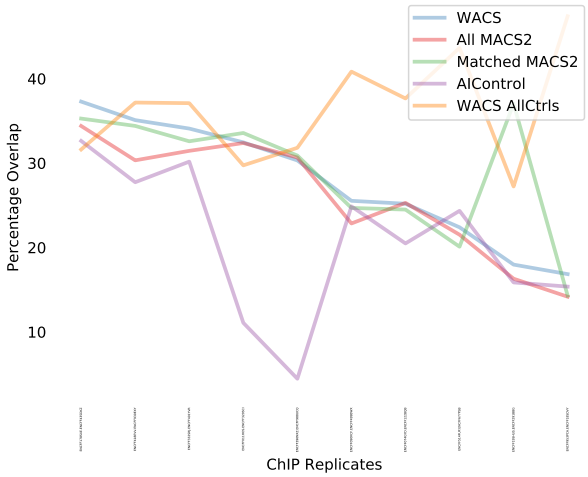
Figure 3.9: Motif enrichment of the peaks called by five methods for each of the three additional validation cell lines: A549 (*a* and *d*), GM12878 (*b* and *e*) and HepG2 (*c* and *f*).



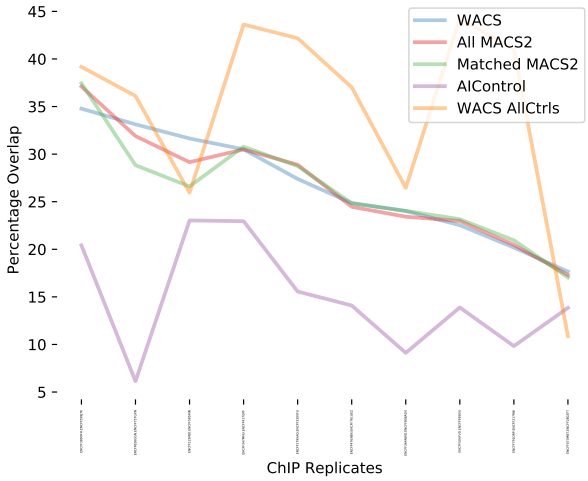
(a) A549 All Peaks



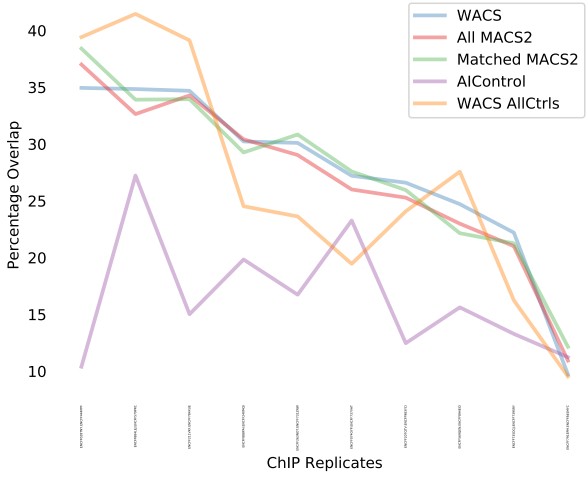
(b) GM12878 All Peaks



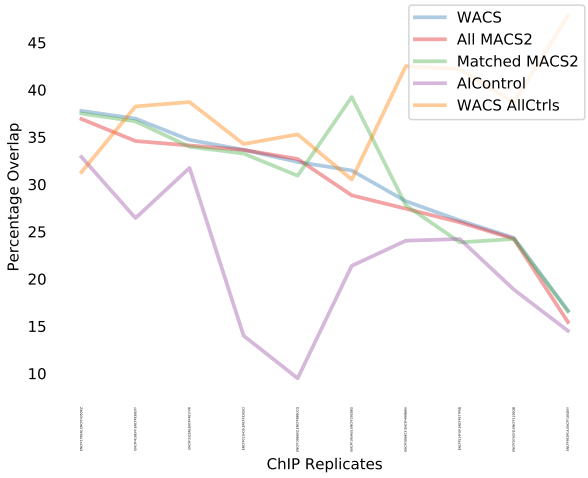
(c) HepG2 All Peaks



(d) A549 Standardized



(e) GM12878 Standardized



(f) HepG2 Standardized

Figure 3.10: Percentage overlap in peaks between biological replicates, for each of the five peak calling methods for each of the three additional validation cell lines: A549 (a and d), GM12878 (b and e) and HepG2 (c and f).

samples in total. For HepG2 with standardized peaks (Figure 3.9f), all methods display similar performance.

Finally, we explore the reproducibility of peaks in ChIP-seq replicates for each cell line. There are a total of 10 ChIP-seq experiments for each cell line, each with two replicates. Figure 3.10 show the percentage overlap with all and standardized peaks for each of the ChIP-seq experiments, when using WACS (blue line), WACS All Ctrls (yellow line), All MACS2 (red line), Matched MACS2 (green line) and AIControl (purple line). WACS All Ctrls outperforms WACS, Matched MACS2, All MACS2 and AIControl on all of the ChIP-seq datasets for all the three cell lines, A549, GM12878 and HepG2 for all and standardized peaks, as show in Table 3.7. Again, AIControl displays the lowest percentage overlap for A549, GM12878 and HepG2 for all and standardized peaks.

Moreover, we conduct a proportion test across all the three cell lines (A549, GM12878 and HepG2) for both motif enrichment and reproducibility. We notice that at times WACS outperforms the other peak calling methods, and other times WACS All Ctrls does. The high variability and small sample size leads to less significance when considering WACS or WACS All Ctrls separately. However, there is an overall positive favor in terms of WACS. For either all or standardized peaks, we observe that the fraction of datasets where both WACS and WACS All Ctrls have the highest motif enrichment and highest reproducibility is statistically significant with a p-value less than 10^{-5}.

3.5 Methods

We evaluated WACS, MACS2.1.1 [1] and AIControl [2] on data from the ENCODE consortium [55]. ENCODE ChIP-seq data is organized into "experiments", which typically comprise two or more ChIP-seq samples generated at the same time and under the same conditions. Experiments also have controls matched to the ChIP-seq samples, and peaks called for each of the ChIP-seq samples. The K562 cell line has the most data available, so we focused our empirical evaluation on that data. We identified experiments with precisely two ChIP-seq samples. We included ChIP-seq BAM files mapped to the GRCh38 genome with filtered alignments. We further restricted attention to TFs with position-weight matrices in JASPAR. By these criteria, we identified 90 ChIP-seq samples (in 45 experiments) for analysis. We also collected all available controls for the K562 cell lines, resulting in 147 control samples for our analysis. Finally, to test the generality of our results in other cell lines, we selected 20 ChIP-seq and 18 control samples for each of A549, GM12878, and HepG2 cell lines. See multiple cell lines/types (CLs). The full list of datasets can be found in Appendix Tables A.1, A.2, A.3, A.4, A.5 and A.6 for the accession codes of samples.

As seen in Figure 3.1 and 3.2, MACS2 pools the controls together for each ChIP-seq sample, whereas WACS estimates a weight for each control and computes a unique weighted control pileup for each ChIP-seq sample. AIControl uses a predefined set of publicly available controls [98]. We used two methods to evaluate the quality of the peaks generated

[1] https://github.com/taoliu/MACS
[2] https://github.com/hiranumn/AIControl.jl/

Table 3.6: Numbers of datasets for which each algorithm produces peaks with the best motif enrichment.

Cell Line	WACS	WACS AllCtrls	Matched MACS2	All MACS2	AIControl
All Peaks					
A549	7	7	2	4	0
GM12878	15	2	2	1	0
HepG2	0	9	11	0	0
Standardized					
A549	10	8	1	1	0
GM12878	11	5	3	0	1
HepG2	3	6	5	0	6

Table 3.7: Number of datasets for which each algorithm produces peaks with the greatest overlap between biological replicates.

Cell Line	WACS	WACS AllCtrls	Matched MACS2	All MACS2	AIControl
All Peaks					
A549	0	8	1	0	1
GM12878	3	4	1	1	1
HepG2	1	7	2	0	0
Standardized					
A549	2	8	0	0	0
GM12878	2	4	3	1	0
HepG2	1	8	1	0	0

by WACS, MACS2 and AIControl. One method considers all the original peaks output by each algorithm (called All Peaks). However, different peak callers can produce peaks in different locations based on the same data, and they can also produce different numbers of peaks. Thus, strictly for the purpose of comparison, we adopted the standardization procedure proposed by Hiranuma et al. [98], where the peak width and number of peaks are normalized for each treatment sample. First, the peak width is normalized by binning the peaks in 1000 base pair windows. For example, a peak at chromosome 1 from 14520 to 15420 is counted as two peaks covering bins 14000 to 15000 and 15000 to 16000. Next, the number of peaks for all five peak-calling conditions for the same dataset is normalized

by retaining the top n most statistically significant peaks, where n is the smallest number of peaks in any of the five width-standardized peak sets.

3.6 Discussion

In this paper, we provide a method, WACS, for improved peak-calling and increase our understanding of ChIP-seq data, controls and their biases. WACS is built on the pre-existing, widely-used and precise peak calling method in MACS2, but has been recoded internally for better efficiency with many simultaneous datasets, and provides weights per control for a more accurate background model. We showed that this form of "intelligent" control construction is beneficial for peak calling. It appears to better estimate background signal in ChIP-seq datasets, as evidenced by better motif enrichment and better reproducibility in the called peaks. We showed that the controls selected by WACS are not necessarily the matched ENCODE controls. Additionally, for most of the ChIP-seq datasets, many more than two controls are selected to model the background signal. These findings run contrary to typical practice, where typically one or a small number of controls are chosen by the experimenter, sometimes based simply on controls having been done simultaneously with the ChIP-seq experiments, without any analysis of whether the control really models well the ChIP-seq background. As noted also by Hiranuma et al. [98], intelligent control selection or construction allows researchers to use other controls non-specific to their ChIP-seq experiment to model the noise distribution. This can decrease cost, time and resources required to perform the ChIP-seq experiments.

Moreover, WACS is a more selective peak caller in comparison to the other peak calling methods – as it outputs the least number of peaks on average. We evaluate WACS using all peaks and standardized peaks and observe that WACS outperforms the other peak calling methods in both cases. However, the difference in performance when using standardized peaks is less than when using all peaks. This shows that the performance of the other peak calling methods improves after standardization. Thus, this suggests that WACS pro-actively removes lower quality peaks.

Hiranuma et al. [98] claim that AIControl is better at removing background noise than MACS2. However, our results suggest the contrary. This may be due to a number of reasons. First, Hiranuma et al. [98] uses a different and nonstandard evaluation method for reproducibility analysis. Whereas we adopted the widely used approach of looking at peak overlaps between biological replicates [24, 133], Hiranuma showed that AIControl had higher irreproducibility than MACS when applied to unrelated datasets. Furthermore, Hiranuma et al. applied MACS2 using only one matched control, while for our analysis, we used either all the ENCODE matched controls for a treatment sample or simply all controls from the same K562 cell line. In either case, the provision of multiple controls may have improved MACS2's performance.

In this work, we described using NNLS to fit a model of ChIP-seq background to control densities, but other formulations are possible. For example, we experimented with an instance-weighted NNLS formulation, to account for differing variances on the regression

targets y_i (the ChIP-seq read counts per window). We did not find any improvement in performance. However, results may depend on how one estimates target variances. Relatedly, performing regression on log-transformed read counts may be worth exploring. RNA-seq analysis tools such as DESeq2 [149] use log linear models for read counts and comparisons between conditions. It would also make sense to explore L1-penalized regression formulations, to explore trade-offs between the number of controls used to model background and the accuracy of the background model.

Future work will deal with a more thorough analysis of the weighted controls approach on other high throughput sequencing data, such as RNA-seq, and other cell lines. The weighted approach will be used to study the biases in RNA-seq data across different platforms, labs, cell types, tissues, etc. For example, RNA-seq is used to measure the difference in gene expression between tissues, where a tissue consists of a mixture of cell types. To generate a realistic control tissue, the weighted approach can be used to weight the cell types in the tissue to model the background signal. Also, in this analysis, we focused on sharp peaks, which are more generally found at protein-DNA binding sites. Thus, an analysis of other broader peaks, for example, will be conducted. Ultimately, our overall aim is to increase the fidelity of conclusions drawn from high-throughput sequencing datasets, each of which may be biased in different ways, and to take fuller advantage of the masses of data already published as a "reference" for interpreting new data.

3.7 Conclusion

We developed a peak calling method, WACS, which allows a mixture of weighted controls as input. The user inputs the controls. These controls can either be weighted by the user, or the weights can be computed by our regression approach. The latter systematically estimates the weights of the input controls to model the background signal for that ChIP-seq experiment. In the special case of equal weights which sum up to 1, the peaks output from WACS and MACS2 are identical. If different weights are allowed, the two algorithms have different outputs. WACS allows only positive weights for better interpretability of results. Negative weights are biologically difficult to interpret; as it does not add to the background signal. WACS proceeds to use this devised background signal to identify regions of enrichment along the genome. WACS is an extension of the most highly cited peak calling algorithm, MACS2 [280]. We conducted a comparison between WACS, MACS2 and AIControl to evaluate our method and the significance of the weighted controls. WACS significantly outperforms both MACS2 and AIControl in motif enrichment analysis and reproducibility analysis.

Chapter 4

SigTFB: Cell Type Specific DNA Signatures of Transcription Factor Binding

The majority of this chapter is extracted from "Cell Type Specific DNA Signatures of Transcription Factor Binding" by Awdeh, Turcotte and Perkins submitted for publication [17]. The source code for SigTFB involving the pre-processing of ChIP-seq data, classification and downstream genomic analysis is available at https://github.com/aawdeh/SigTFB. The data used is available at https://doi.org/10.20383/103.0605.

4.1 Author Contributions

Aseel Awdeh recognized the need for a better understanding of transcriptional regulation, devised a method to address the limitations, wrote the software implementation, conducted all the experiments and wrote the manuscript, all with input and guidance from Theodore J Perkins and Marcel Turcotte.

4.2 Overview

Transcription factors (TFs) bind to different parts of the genome in different types of cells. These differences may be due to alterations in the DNA-binding preferences of a TF itself, or mechanisms such as chromatin accessibility, steric hindrance, or competitive binding, that result in a DNA "signature" of differential binding. We propose a method called SigTFB (Signatures of TF Binding), based on deep learning, to detect and quantify cell type specificity in a TF's DNA-binding signature. We conduct a wide scale investigation of 194 distinct TFs across various cell types. We demonstrate the existence of cell type specificity in approximately 30% of the TFs. We stratify our analysis by different antibodies for the same TF, to rule out the possibility of certain technical artifacts, yet we find that

cell type specificity estimates are largely consistent when the same TF is assayed with different antibodies. Our comprehensive investigation provides a basis for further study of the mechanisms behind differences in TF-DNA binding in different cell types.

4.3 Background

Non-protein coding regions constitute approximately 98% of the human genome [56]. These regions contain complex instructions to regulate gene expression. Differential binding of transcription factors (TFs) to regulatory sites drives differential gene expression, allowing one genome to give rise to a diversity of cell types and tissues. Conversely, some cell types are defined by one or a few master regulatory TFs that they express. However, the same TF may bind different sites or even different preferred DNA sequences across these multiple cell types [40, 81, 94, 139, 192, 195, 232].

Many studies have shown the cell type specific nature of TF binding locations across multiple cell lines and tissues [40, 81, 94, 139, 167, 192, 195, 232, 252]. For example, Lee et al. studied the binding of the TFs MYC and CCCTC binding factor (CTCF) across 11 different human cell types, and found that both showed some degree of cell type specificity in their binding locations [139]. However, MYC had a much greater degree of cell type specificity than CTCF. Fewer than 25% of CTCF binding sites were cell type specific, while more than 87% were cell type specific for MYC. Additionally, binding sites unique to the cell type were associated with cell type specific functions, such as endothelial cell differentiation and positive regulation of developmental growth in fibroblast cells. Due to its higher degree of cell type specificity, MYC is mainly involved with cell type specific functionalities, while CTCF has a more consistent role across cell types. Cell type specific binding patterns have also been observed for other TFs. Estrogen receptors α (ERs), for instance, bind to distinct DNA patterns in cancerous lines, such as breast cancer and endometrial cancer [81]. The cell type specific sites typically have lower affinity to ERs which bind in conjuction with other TFs, unlike the shared sites which have high affinity for ERs. SOX2 is another example that displays dual modalities of cell type specific binding in human embryonic stem cells (hESC) [278], where the co-binding of SOX2 with PAX6 leads to hECS neural differentiation, and the co-occurrence of SOX2 with OCT4 results in self-renewing hECS. In another study, Wang et al. show the cell type specificity of HOXB9 in K562 cells in comparison to other cell types such as HepG2 and H1-hECS [252]. However, there has yet to be a comprehensive and quantitative analysis of cell type specificity across a broad range of TFs.

There are many factors that cause cell type specificity in TF binding. With the exception of pioneer transcription factors [273], TF binding usually occurs in more accessible regions along the genome. Thus, cell type specific chromatin accessibility is one mechanism directing TFs to different parts of the genome [167, 231, 232, 249]. Another factor is direct protein-protein interactions between TFs which lead to the formation of stable regulatory complexes. All constituents of the complex may bind directly to the genome, or tethering interactions may lead to one element of the complex binding to the genome. Complexing

with different DNA-binding partners may draw a TF to different parts of the genome. Alternatively, complexing with other proteins may alter the conformation of the protein and change its DNA-binding preferences [39, 76, 168, 192, 232, 252]. For example, PBX and MEIS paralogs have varying expressions across tissues, and their cooperative binding with HOX alters HOX binding specificity [40, 192]. An additional factor that influences the DNA-binding preferences of a TF protein is alternative splicing [147]. Different versions of the same TF, or TF isoforms, can either bind to different parts of the genome leading to differential gene expression, or bind to the same sites along the genome with different binding affinities resulting in varying amounts of gene expression [147, 150]. Other events, such as post-translational modifications, indirect cooperative binding or conversely steric hindrance, can also influence binding [23, 193].

While differential binding of TFs has been well established, here we seek to investigate in a deeper and more systematic way the possible mechanisms underlying differential binding. In particular, we seek to discriminate two major classes of mechanisms: those that are associated with DNA sequence binding signature and those that are not. Importantly, this is a distinct question from whether the same or different sites are bound in different cell types. For example, a TF might bind completely different sites in two cell types, and yet there may be no differentiating DNA sequence features at all. Mechanisms of differential binding that might not show a difference in DNA binding signature include: changes in TF expression level, so that fewer or more sites of the same type are detectably bound; changes in accessibility of sites of the same type; or even differences in data quality or depth, which alter our ability to detect binding. Conversely, mechanisms that would show a difference in DNA binding signature include: conformation-based changes in inherent TF-DNA preferences resulting from alternative splicing or post transcriptional modifications; complexing with different partners; or cooperation or hindrance with other TFs at regulatory sites.

A deeper understanding of genomics, including TF binding, has been achieved through the use of deep learning approaches. DeepBind is one of the first pivotal methods that used convolutional neural networks (CNNs) for the identification of protein binding sites in DNA and RNA sequences [6]. It used a single convolutional layer, and trained multiple single task models – one for each TF and cell line combination. Other methods, such as DeeperBind [92], DanQ [197] and DeepDRN [46], used a combination of both CNNs and recurrent neural networks (RNNs) for the prediction of TF binding sites. DeepSEA [285] and Basset [120] used CNNs to uncover regulatory features from genomic sequences to predict chromatin accessibility along the genome, and more specifically the impact of single nucleotide variants on regulatory regions, such as DNase hypersensitive sites, TF binding sites and histone marks.

Deep learning has also been used to generalize across cell types, where common non-specific preferences of a TF across cell types are used to predict TF binding activity in cell types with little or no data available [143, 144, 196, 198, 286]. This is possible because a TF may have the same binding preferences in multiple cell types [44, 158]. Indeed, databases such as JASPAR [44], HOCOMOCO [132] and Transfac [158], are established based on the largely successful assumption that TF-DNA binding preferences can be specified independent of context, and can even be derived in-vitro, from SELEX or PBM experiments

[114, 237]. Yet, several studies, including [18, 40, 81, 94, 113, 139, 167, 192, 195, 210, 232, 252], show exceptions to the rule of invariant TF-DNA binding preferences, or demonstrate other differential DNA signatures in bound regions. In this paper, we propose a new deep learning approach for analyzing TF-DNA binding sites of a TF across a set of cell types. Although formulated as a binding site prediction problem, the purpose of the learning procedure is to detect and quantify the degree of cell type specificity in DNA-binding signatures for the TF across cell types. That is, given known bound regions for a TF across a potentially large number of different cell types (or tissues or conditions), we seek to quantify the extent to which the DNA sequences of those regions contain discriminative signals regarding cell type. This constitutes a first step towards understanding mechanisms that may underlie differential binding of TFs.

For a more comprehensive and thorough investigation of the cell type specificity of TFs, we develop a method called SigTFB (Signatures of TF Binding). We build upon previous deep learning studies in our own work. The network we use is based in part on the network used in DeepBind [6], augmenting it to differentiate and quantify cell type general versus cell type specific binding. Unlike most previous work, where the learning problem formulation is based on the discrimination of bound versus unbound regions, which may be genomically real unbound sites or sequence permuted sites, all the sites in our problem are bound. That is, sites are bound in different cell types, and our learning problem determines in which cell there is binding based on the DNA sequences. We focus on differences in "DNA signatures" of binding, rather than differences (or similarities) in the binding sites per se. We use the term DNA signatures to encompass the several factors (previously discussed) that could impact the DNA binding preferences of a TF. Our training procedure is inspired by that of ChromDragoNN [165], but uses a direct encoding of cell type in place of gene expression values. Moreover, similar to MTTFSite [286] and FactorNet [198], we utilize multi-task formulations to learn shared binding preferences of a TF in different cell types. Like Novakovsky et al. [170], which combines multi-task learning with transfer learning to predict TF binding in a specific cell type, we employ a multi-task transfer learning formulation for our learning problem. Our work is similar in intent to a recent study of the differential and cooperative binding nature of the two TFs, MEIS and HOXA2, in three mouse tissues [192], but different in the approach we employ and much larger in scale.

We conduct a large scale investigation of 194 TFs assayed by one or more antibodies (AB) across various cell types (for a total of 230 distinct TF-AB pairs), identifying TFs that show a difference in DNA-binding preference across multiple cell lines, and quantifying their degree of cell type specificity. We also investigate the consistency in the degree of specificity across the different antibodies for a specific TF, and explore the correlation between the number of cell types available for a specific TF and the degree of cell type specificity.

4.4 Results

4.4.1 A two-step deep learning model to study the differential binding of a transcription factor.

To study the differential binding of a TF, we study one TF and antibody (AB) at a time – each combination requiring ChIP-seq data from a minimum of two different cell types. Data from different ABs for the same TF are not mixed, as we hypothesized that varying binding affinities and the off-target binding effects of the different ABs may contribute to "false" differences between cell types. We collect all the ChIP-seq peak sequences from the different cell types for a specific TF-AB from the ENCODE consortium [56]. We identified 194 unique TFs assayed by ChIP-seq using the same AB in at least two cell types. As some TFs were assayed by multiple ABs in different cell types, there was a total of 230 TF-AB pairs with ChIP-seq data in multiple cell lines/types (CLs). The full list of datasets can be found in Appendix Table B.1. We obtained called peaks for each TF-AB-CL combination, and used human genome GRCh38/hg38 to extract 101 bp windows of DNA sequence around the summit of each peak (See Methods).

For each TF-AB pair, we define the following prediction problem: given the peak sequence and a one-hot encoding of cell type, can we predict whether that sequence was bound (i.e. was there a peak) in that cell type? We are not interested in performance in terms of peak prediction as the input peaks are already experimentally measured. Instead, we are interested in developing a supervised learning system to identify sequence features for a specific TF to distinguish binding in one cell type versus another.

We formulate the prediction of differential binding per TF-AB as a two step process using a deep learning architecture (see Figure 4.1). We trained a separate model for each of the 230 TF and AB combinations, covering 194 distinct TFs in a total of 35 human cell types. To construct the training set for each TF-AB, we merged ChIP-seq peaks that overlap by at least 30bp across the different corresponding cell types, adopting a similar approach to Basset [120]. The training set for each TF-AB corresponds to the collection of these peaks across the cell types. In stage 1 of training, each input instance comprises a 101 bp peak DNA sequence one-hot encoded into a 404-length binary vector. The output is a C length binary vector indicating in which cell types the peak is bound by the TF, with C being the number of cell types.

The first stage focuses on the sequence part of the input, and is trained as a multi-task classifier for the simultaneous prediction of bound versus unbound regions across various cell types. A rectifier activated convolution layer first transforms the input matrix into an output matrix. The rows in the output matrix correspond to convolutional kernels, where each kernel is a motif detector. This is followed by the max pooling layer which reduces the convolutional matrix into a vector of length equal to the number of convolutional kernels (N in Figure 4.1). The output vector is then passed to a fully connected layer, where it is then compared via the Basset loss function to the true target vector. This constitutes the stage 1 model, and presents the shared features across all cell types for that specific TF-AB.

In stage 2, each unified peak instance is expanded into $2 \times C$ instances, where in addition to the 404-length binary vector encoding the peak DNA sequence, there is a length C binary vector encoding cell type. In C of the instances, the cell type vector has one element set to 1 with the rest being zeros, specifying cell type, and a single binary output variable indicating binding or not in that cell type. These are referred to as "cell type specific" (CL Specific) instances. In the remaining instances, the cell type vector is set to all zeros, and the DNA input and binary output remain the same. This is a "cell type general" (CL General) instance, where the approximator is being essentially given a cell type specific instance, but the cell type information is hidden. Intuitively, the difference in prediction performance between the cell type specific and the cell type general cases conveys the degree to which the network is able to identify cell type specific DNA signatures that help in binding prediction.

The weights of the pre-trained multi-task model of stage 1 are used to initialize the parameters of the stage 2 models for the prediction of bound regions for a specific TF in a specified cell type. The cell type vector of length C defined previously is passed through a fully connected layer of length $P1$, and then concatenated with the convolutional output matrix from stage 1. The concatenated vector is passed through a fully connected layer of length Q, where the predicted output is then compared to the true target vector via the non-negative log likelihood loss function.

During both training and testing, instances are randomly chosen in mini-batches to have the same number of positive and negative instances from each cell type, and the same number of cell type specific and cell type general instances, avoiding any problems with class imbalance. We use the Ax hyperparameter optimization technique [22] along with 10-fold cross validation to produce an ensemble of 10 models. For each model, we compute a macro-average (across cell types) classification area under the receiver-operator characteristic curve (AUC) separately for cell type specific and cell type general instances. Finally, we take the difference of the two as a measure of learned cell type specificity, using a t-test across the 10 folds to assess statistical significance. A positive difference, where cell type specific predictions are more accurate than cell type general, is taken as evidence for the existence of a cell-type specific DNA signature in the binding sites of that TF.

4.4.2 ATF7 binding shows cell type specific DNA binding signatures.

To demonstrate our approach, we first focus on Activating Transcription Factor 7 (ATF7). As a member of the ATF family, ATF7 binds to the cyclic AMP response element (CRE) with the consensus DNA sequence "TGACGTCA" [49, 154]. Members of the ATF family are basic leucine zipper (bZIP) factors that complex with other bZIP factors to form homodimers or heterodimers [49, 85, 154, 161]. The ATF TFs exhibit varying functionalities in different tissues and cancerous cell lines, including tumour suppressive and oncogenic functions [49]. For instance, the deletion of ATF7 results in the spread of lymphoma [49]. Conversely, the activation of ATF7 in gastric or hepatocellular carcinoma promotes the proliferation of cancer cells. As such, ATF7 may be used as a biomarker for the early

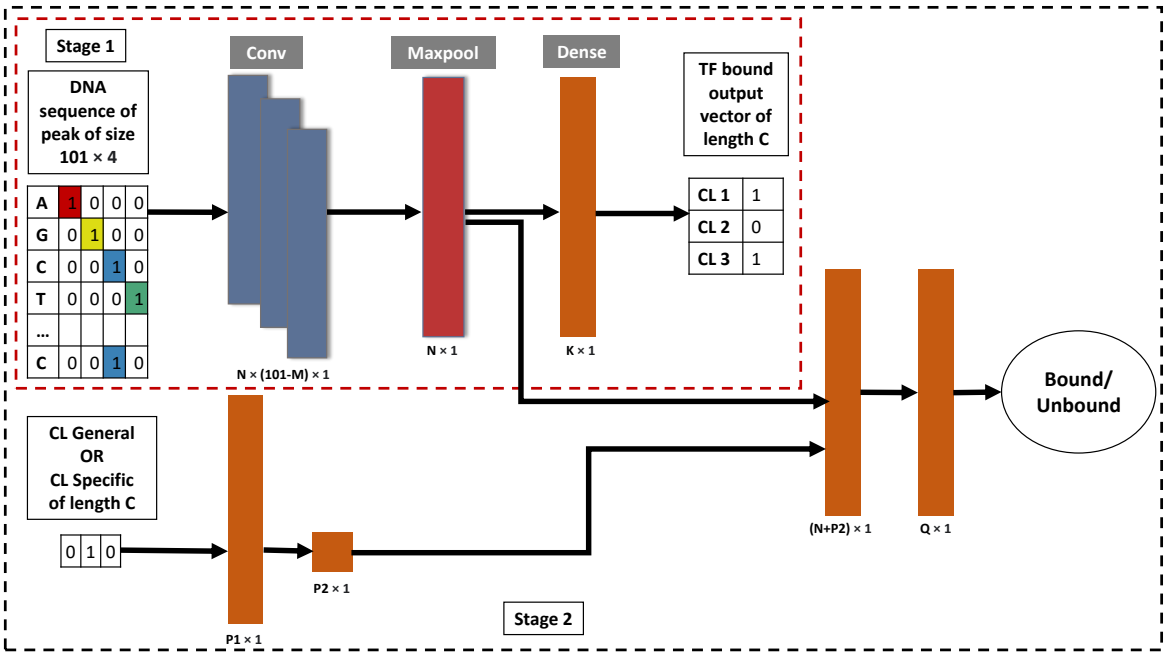
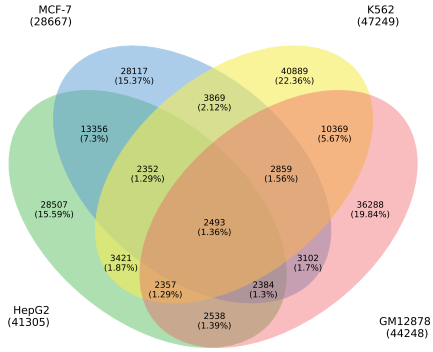


Figure 4.1: Simplified diagram of Stage 1 and Stage 2 Models. Stage 1 is shown in the red dashed box. Stage 2 is shown in the black dashed box. In Stage 1, the input instance is a one-hot encoded DNA sequence of size 101×4. This is passed through the convolutional layer (Conv) with N filters of size M, through a maxpool layer (Maxpool) of length N, a fully connected layer (Dense) of length K, then the output layer of length C to predict if the TF is bound or not in the different cell types ($C = 3$ in this case). Stage 2 takes cell type information of length C as input as well. This is first passed through fully connected layers of lengths $P1$ and $P2$, and then concatenated with the output of the convolutional layer from Stage 1. The concatenated output is passed through a fully connected layer of length Q to predict whether the sequence is bound or not in that cell type.

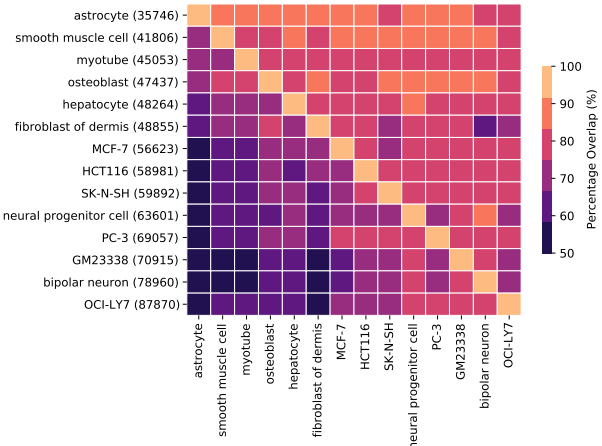
detection of tumours in liver and gastric cell lines. Due to the differences observed, we suspect ATF7 to bind to different places along the genome in different cell lines.

To investigate the activity of ATF7, we obtained ChIP-seq peaks from ENCODE [56] in the following four cell lines: GM12878, K562, HepG2 and MCF-7. The cancerous cell lines HepG2, MCF-7 and K562 correspond to liver hepatocellular carcinoma, breast cancer and myelogenous leukemia respectively. GM12878 is a non cancerous lymphoblastoid cell line. Figure 4.2a shows a Venn diagram of the peak overlaps between the four cell lines. The number of peaks per cell line are shown after the cell type name in brackets. We notice that a mere 1.36% of the total number of peaks across all four cell lines overlap, and the majority of the peaks are unique to one of the four cell lines. For example, 22.36% of the K562 peaks do not overlap with peaks from other cell lines. Moreover, we notice there is greater peak overlap between the pairs HepG2 and MCF-7, and GM12878 and K562.

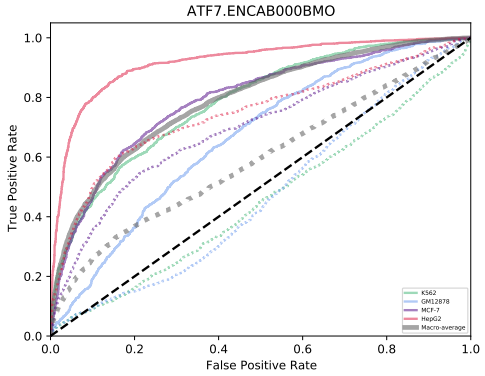
The lack of overlap between peaks in the four cell lines does not imply cell type specificity in DNA binding preference, as sequences in those peaks may be very similar. Differences in output may be due to dissimilarities in terms of noise, bias or even the number of peaks of the ChIP-seq experiments. For instance, HepG2 has over 40,000 peaks while



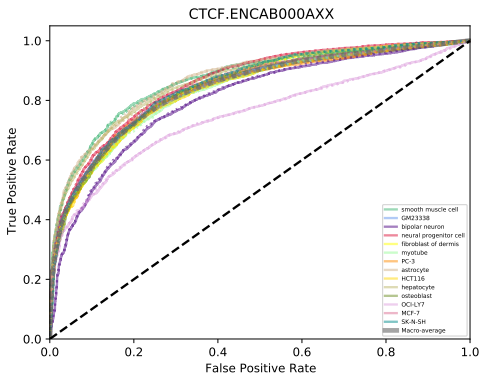
(a)



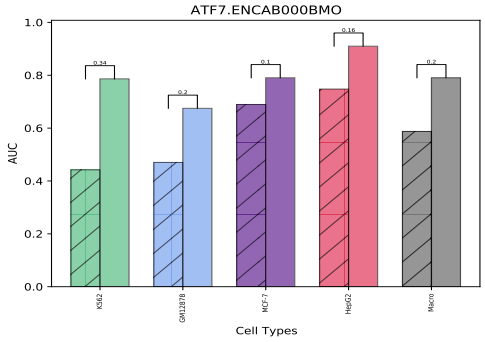
(d)



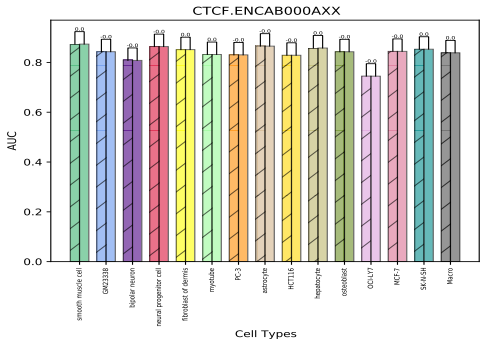
(b)



(e)



(c)



(f)

Figure 4.2: (a) Venn diagram of percentage overlap between cell lines for ATF7. (b) ROC curves per cell line cell line per condition: CL General (dashed line) and CL Specific (solid line) for ATF7. (c) AUC per cell line per condition: CL General (shaded) and CL Specific (not shaded) for ATF7. (d) Heatmap of percentage overlap between 14 cell lines in CTCF.ENCAB000AXX. (e) ROC curves per cell line cell line per condition: CL General (dashed line) and CL Specific (solid line) for CTCF. (f) AUC per cell line per condition: CL General (shaded) and CL Specific (not shaded) for CTCF.

MCF-7 has fewer than 30,000. Therefore, no more than 75% of HepG2 peaks could possibly overlap with MCF-7 peaks

To determine if there are cell type specific DNA signatures in the ATF7 peaks, we applied our deep learning method, SigTFB, as described in the previous section. Figure 4.2b shows the receiver operating characteristic (ROC) curves for each cell line with and without the cell line identity being provided, as well as averaged performance across all cell lines. The plot shows high variability in site prediction across across cell lines. Predictions for HepG2 (solid red curve) are significantly better than for MCF-7 and K562 (solid purple and green), which are better than for GM12878 (solid blue). Consequently, predictions are more accurate when the network is informed of cell type than when it is not (e.g. solid red versus dashed red curves). This trend is also true for the macro-averaged ROC curve (gray color in Figure 4.2b). Figure 4.2c shows the area under the ROC curve (AUC) per cell line per condition for the ATF7 TF, where the shaded and unshaded bars are CL General and CL Specific cases respectively. For each cell line, as well as the macro-averaged result, there is a clear difference between the two conditions. CL Specific classification outperforms CL General classification with a macro-averaged AUC difference of 0.2 ($p < 0.05$; one-sample t-test on AUC difference). Thus, we can conclude that the network has detected DNA signatures discriminating peaks in different cell types. We return to the question of exactly what those signatures are below. First, we examine another transcription factor, CTCF, in detail.

4.4.3 CTCF binding does not show cell type specific DNA binding signatures.

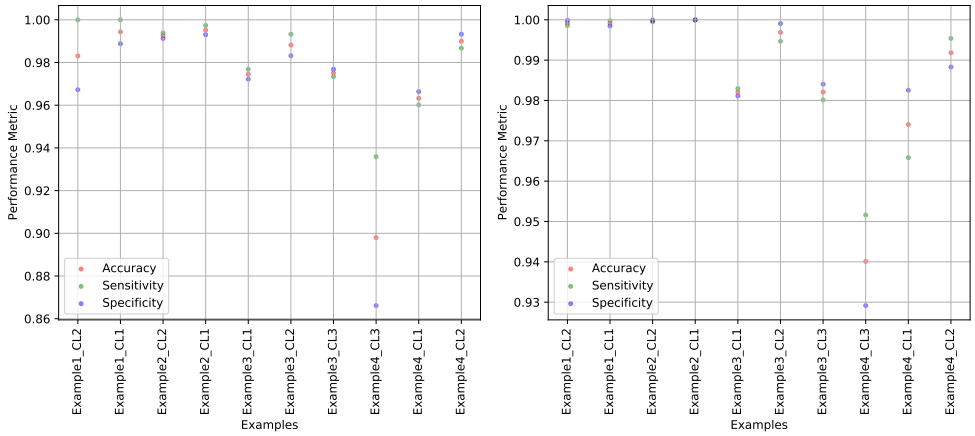
We next examine another transcription factor (TF), CCCTC-binding factor (CTCF). The CTCF binding domain is defined by ll zinc fingers, and is believed to be invariant across cell types – meaning that although the actual binding locations across cell types may vary, the CTCF DNA binding preferences remain the same. CTCF can function as a transcriptional repressor, transcriptional activator, or as an insulator barrier between genomic domains [48, 100, 122]. It also plays a key role in regulating the three-dimensional structure of chromatin. The function of CTCF is greatly dependent on its DNA binding partners.

To test these prior findings, we obtained ChIP-seq data for CTCF in 14 different cell types. These cell types include smooth muscle cell, GM23338, bipolar neuron, neural progenitor cell, fibroblast of dermis, myotube, PC-3, astrocyte, HCT116, hepatocyte, osteoblast, OCI-LY7, MCF-7 and SK-N-SH. The percentage overlap of ChIP-seq peaks between each pair of cell types is shown in Figure 4.2d, where each entry of the heatmap shows the percentage of peaks of the row's cell line overlapping peaks in the column's cell line. Additionally, the number of peaks per cell type are shown in brackets after the row cell type label. Overlap percentages range from approximately 50% to 90%, with an average of 77%. Cell types with fewer peaks tend to be better covered by cell types with more peaks, suggesting an element of peak detection power is at play. For instance, the astrocyte dataset has the fewest peaks at ≈37,000, which are more than 90% covered by the CTCF peaks in every other cell type – even distantly related cell types such as osteoblasts or

fibroblasts (first row in Figure 4.2d).

Figure 4.2d gives some intuition about the datasets. However, as seen for ATF7, a simple intersection analysis is not sufficient to determine cell type specificity. We further investigated the binding activity of CTCF by training SigTFB on CTCF and its 14 corresponding cell lines. Figure 4.2e shows the ROC curves for each of the cell types and the macro-averaged ROC across all cell types. Compared to ATF7, there is relatively little difference in binding site predictability across cell types and nearly no difference in predictability for a given cell type, with or without cell type identity information.

Cell types OCI-LY7 (lavender line) and bipolar neuron (indigo line) have the worst prediction performance, and also have the highest number of peaks. Possibly, a greater fraction of these peaks are not genuine, which would explain both inflated peak numbers and prediction difficulty. Figure 4.2f shows there is little too no difference in the area under the ROC curves (AUC) between CL Specific (solid line) and CL General (broken line) conditions for each cell line ($p > 0.5$; one-sample t-test on percentage differences). Consequently, these results illustrate the ubiquitous non cell type specific nature of CTCF DNA binding preferences. Importantly, they also demonstrate the specificity of SigTFB, in that it does not incorrectly report cell type specificity where there is none to be found.



(a) (b)

Figure 4.3: Plots of performance in terms of accuracy, sensitivity and specificity per synthetic example per cell type on simulated data for stage 1 models on (a) training and (b) test datasets. The x-axis corresponds to the examples, and the y-axis corresponds to the performance metrics.

4.4.4 Determining the degree of the cell type specificity of the different TFs.

Motivated by our results for ATF7 and CTCF, we expanded our study by investigating cell type specificity. Results on simulated data can be seen in Figure 4.3. Both balanced and imbalanced datasets are included in the analysis. Details regarding the simulated datasets,

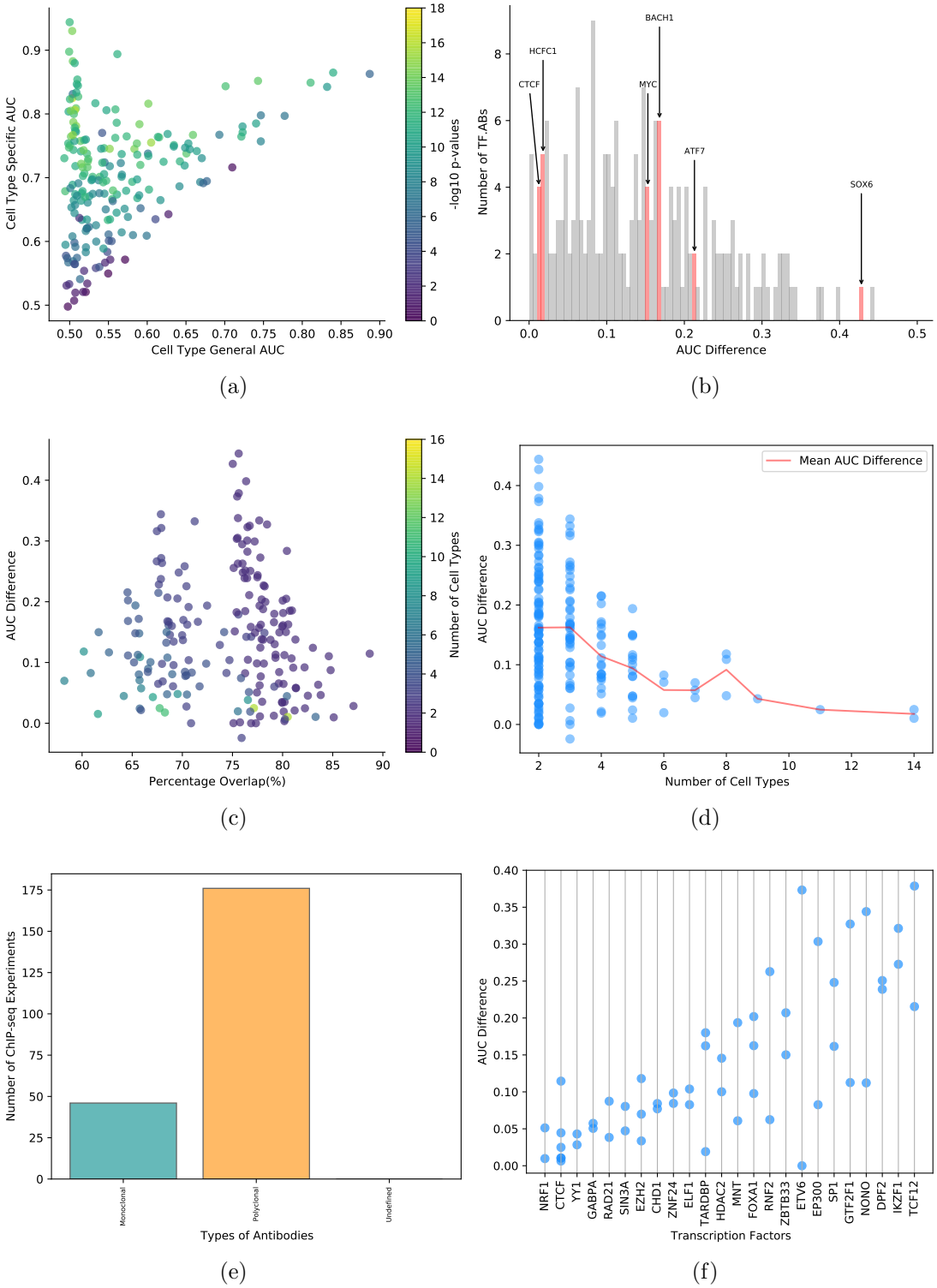


Figure 4.4: (a) Scatter plot of Cell Type General AUC versus Cell Type Specific AUC with the color gradient depending on the -log10 p-values. (b) Bar chart of AUC differences. (c) Scatter plot of percentage overlap(%) and AUC differences, with the color gradient depending on number of cell types per TF-AB. (d) Scatter plot of number of cell types versus AUC difference, with red line depicting mean AUC difference per cell type count. (e) Number of ChIP-seq experiments using polyclonal vs monoclonal antibodies. (f) Scatter plot of AUC differences for TFs with more than two ABs.

Simulated Datasets	Cell Line 1	Cell Line 2	Cell Line 3
Example 1	CTGCAG	CAGCTG	-
Example 2	CTGAG	TATATT	-
Example 3	CTGAG	TATATT	ACAC
Example 4	CTGAG	TATATT	ACAC

Table 4.1: Motifs per cell line per simulated example.

Simulated Datasets	Cell Line 1	Cell Line 2	Cell Line 3
Example 1	11916	7566	-
Example 2	35665	15722	-
Example 3	35665	15722	80959
Example 4	17665	14079	8004

Table 4.2: Number of instances per cell line per simulated example.

in terms of motifs and number of instances, present can be found in Tables 4.1 and 4.2 respectively. In Example 1, there are two cell lines each with a separate motif; in Example 2, there are also two cell lines but the dataset is more imbalanced than Example 1; in Example 3, there are three cell lines with some degree of class imbalance; and finally in Example 4, there are three cell lines, but data is a bit randomized, such that a motif present may not always indicate binding. In Figures 4.3 a and b, we plot different evaluation metrics, such as accuracy, specificity and sensitivity, across the simulated datasets for the training and test data sets respectively. We validate that our models are indeed learning on both balanced and imbalanced datasets, such that the minimum generalization performance is 0.93 across all examples and cell lines.

Now that we have established that SigTFB does indeed learn, we turn our attention back to our 230 TF-AB combinations. Figure 4.4a displays a scatter plot of the mean AUC of prediction when the network is (y-axis) or is not (x-axis) told what cell type it is predicting for, with the color gradient depending on the negative $\log 10$ p-values. Each point corresponds to a TF-AB combination. We observe a continuum of cell type specificity, where TFs with the least cell type specificity lie in the $x = y$ diagonal of the scatter plot. For these TFs, the cell type information does not improve prediction. The position of a point along the diagonal may depend on the extent to which there are shared common motifs for the TF across cell types, or even the extent to which the peaks themselves overlap across cell types. Conversely, points lying above the diagonal indicate that the network better predicts binding when informed of the cell type; these are TFs with the most significant cell type specificity. In other words, there are signals in the DNA sequence to which the network responds differently, depending on the cell type for which binding is being predicting. Points in the upper left corner correspond to TFs where cross-cell type prediction is virtually impossible, but is highly accurate for specific cell types. For such TFs, each cell type is expected to have specific DNA motifs that discriminate its binding sites.

Out of 230 TF-AB combinations, 204 TF-ABs have a p-value of less than 10^{-5}, suggesting that a majority of TFs have some degree of cell type DNA signatures in their binding sites. We observe that many TFs have cell type specific binding patterns at differing degrees of specificity, where the majority of the points (approximately 143) show relatively little predictability in the cell type general formulation (cell type general AUC<0.55). Of the 143, 134 have a statistically significant degree of cell type specificity, with 70 showing cell type specific AUC above 0.7. Figure 4.4b shows a histogram of the distribution of AUC differences between Cell Type General and Cell Type Specific across the different TF and AB combinations, with the bins containing ATF7, CTCF, HCFC1, BACH1, MYC and SOX6 highlighted in red. TFs that play a pivotal role in cancer either as oncogenes or suppressors, such as MYC [58, 234, 251, 266], BACH1 [60, 177, 279], ATF7 [49, 85, 154, 161], and SOX6 [88, 110], show a relatively higher cell type specificity than other TFs, such as CTCF [48, 100, 122] and HCFC1 [116, 121, 264], that are involved in chromatin regulation or other cellular processes. Appendix Table B.1 and Appendix Figures B.1 to B.10 show the AUC differences obtained for each TF-AB pair.

As explained above, the lack of overlap between binding sites in different cell types is not evidence per se of any differential DNA signature. We next examined whether there is any association between the two. Figure 4.4c plots the mean percentage peak overlap versus the mean AUC difference for each TF-AB, with the color gradient depending on the number of cell types and each point being a TF-AB combination. No clear relationship between the two variables is seen (Spearman correlation r=-0.1). With either a high percentage overlap between 75% and 80% or a lower percentage overlap between 65% and 70%, the AUC difference ranges between 0.44 and -0.02. This suggests that mean percentage overlap is not an indicator of cell type specificity.

We also investigate whether the number of cell types available for a specific TF-AB pair contributes to the varying degrees of cell type specificity. With the cell type count per pair ranging from 2 to 14, the differences observed could be attributed to the number of cell types. A higher number of cell types may increase the likelihood of detecting a difference in AUC – as the condition that results in cell type specificity may more likely be there. Conversely, with more cell types, the number of false positives may increase, thus reducing the possibility of finding a difference. With less cell types, on the other hand, a difference in AUC may be a result of the quality of ChIP-seq experiments. The scatter plot in Figure 4.4d shows the relationship between the number of cell types and AUC difference, with the red line depicting the mean AUC difference per cell type count. For the 101 TF-ABs with exactly two cell types, the AUC differences varies between 0.44 and -0.02, suggesting that the degree of cell type specificity is not impacted by the number of cell types when the count is equal to two. Similar trends are observed with cell type counts less than six. However, with a higher number of cell type, there is a limited number of experiments available, making it impracticable to draw any conclusions.

Out of the 194 distinct TFs we studied, 24 were assayed with multiple ABs. Lack of consistency across ABs due to different off-target biases or binding affinities may impact the TF's DNA signatures. Moreover, different ABs may have been used on different sets of cell types. Nevertheless, we may be reassured of the generality of our results if our measure of cell type specificity is consistent between different sets of experiments for the

same TF. For most of the TFs, 170 out of 194, only 1 AB is used (Figure 4.4e). However, there are 19 TFs with at least 2 ABs – all of which are polyclonal. Figure 4.4f shows a plot of the AUC differences between Cell Type Specific and Cell Type General for the 24 TFs with at least 2 ABs. Each point corresponds to a TF-AB combination, with TFs ordered by increasing average specificity. For example, CTCF was assayed with five different ABs, with a consistent result of very little cell type specificity – all five AUC differences are below 0.14. Conversely, several TFs show consistently high cell type specificity across multiple ABs, including: TCF12, SPI1, MNT, IKZF1 and DPF2. The least consistency is seen for the TF ETV6. Surprisingly, both datasets for ETV6 explore the same two cell lines GM12878 and K562, yet produce very differing results for cell type specificity: almost 0 for ENCAB000ABD and 0.37 for ENCAB997CJG. This may be due to differences in the actual ABs used, or could be a result of differences in the total number of peaks per dataset for each TF-AB combination.

4.5 Methods

Data accession and preprocessing

To identify the regions of enrichment along the genome for various TFs, we turned to the ENCODE project [56]. Because different antibodies for a TF can have different specificities or biases, we chose not to mix data from different antibodies. We identified 230 transcription factor-antibody (TF-AB) combinations that were generated using the ENCODE uniform processing pipeline for replicated ChIP-seq experiments [56]. We required experiments with data in at least two cell types. The pipeline maps pair-ended reads for each ChIP-seq experiment to GRCh38 and obtains filtered alignments. It then takes as input the filtered alignment BAM files from the replicates and the controls of the ChIP-seq experiment to detect peaks that pass the irreproducible discovery rate (IDR) at a threshold of 2% [56]. We used these peaks as input for our analyses.

For each TF-AB combination, we obtained the "experiment" level peaks across its various corresponding cell types. Then, similar to the greedy approach used by Basset [120] for chromatin accessibility, we repeatedly merged peaks from different experiments across various cell types if they overlapped by at least 30bp. The center of each merged peak is taken and extended by 50bp in each direction, such that the length of the intervals are 101bp [120]. There can be multiple experiments for the same TF-AB combination.

Training and test set construction

We train a different model for each of the 230 TF and AB combinations (see Figure 4.1). In stage one, described further below, the dataset per TF-AB pair is a multi-class dataset, where each of the C cell types assayed for that TF-AB is one of the possible classes. Each instance corresponds to a unified peak, as described in the previous section. A positive instance is a sequence which is bound in a specific cell type by the TF under study, meaning

they are regions that have passed the IDR threshold of 2%. Due to the merging of the peaks, a negative instance is a sequence not bound by that TF in a specified cell type. Because there can be multiple experiments for the same TF-AB pair in the same cell type, the target output is deemed positive if a peak was detected for at least one of the experiments in that cell type. The output is a binary vector of length C encoding which cell types the unified peak is bound, or not bound, in. There is at least one element that is bound (has a value of 1) in the output vector, and as many as all the elements bound, if that peak is found in all cell types.

For stage 2 training, there are $2 \times C$ instances per unified peak. For C of those instances, the input is a one-hot encoded DNA sequence representing the unified peak, as well as a binary vector one-hot encoding one of the C cell types. The output is 1 or 0 depending on whether that instance is bound in that cell type or not. This is the cell type specific case, where the model is informed about which cell type is being explored. For the remaining C instances, a vector of all zeros is used, instead of a one-hot encoded vector, with the same output and DNA sequence as before. This is the cell type general case, where we are not informing the model which cell type we are investigating. Here, in the concatenation step, we use a direct encoding of cell type, instead of the gene expression values used in ChromDragoNN [165].

We split each TF-AB dataset into 90% training and 10% testing. At both stages 1 and 2, the training dataset is then repeatedly divided into training and validation at a ratio of 8:2. At each iteration (per stage), we train the model on the training subset and observe the performance on the validation subset. We use Ax for optimal hyperparameter selection [22]. This is repeated 10 times for each stage for each TF-AB dataset. The hyperparameters for stage 1 training includes learning rate, weight decay, initial weight scales for the convolutional and fully connected layers, number of channels, batch size, and the number of epochs. PyTorch 1.5.0 (GPU) with the Adam optimizer is used for stage 1 training.

Many datasets have an unequal distribution of instances across cell types for a specific TF-AB. As a result, we observe poor predictive performance for the minority cell types – although the overall predictive performance across all cell types for that model may be high. To address the class imbalance issue, we develop a multi-label balanced sampler. During training or testing, each mini-batch selected has the same number of total instances per cell type, and each cell type has the same number of positive and negative instances. Thus, we ensure that high overall predictive performance is not biased towards certain cell types because of class imbalance across or within cell lines. Models with very poor predictive performance per cell type per TF-AB (with specificity or sensitivity less than 0.2), even after addressing the class imbalance issue, are filtered out.

We also filter out ChIP-seq experiments with fewer than 100 positive or 100 negative test sequences. If the size of the dataset is too small, then enrichment values may be too optimistic.

Neural network training

We formulate the training of differential binding per TF as a two step process (see Figure 4.1). All experiments were run on the Compute Canada platform. Models for each TF-AB were trained on a single GPU (NVIDIA (R) Cuda V11.0.194).

Stage 1. Stage 1 is formulated as a multi-task classification problem, such that for each input DNA sequence we predict whether or not its bound across multiple cell lines for a specific TF-AB. We use a modified version of DeepBind [6] of one hidden layer CNN with 404 binary inputs followed by a fully connected layer (Figure 4.1). Unlike DeepBind [6], the number of channels in our model is set as a hyperparameter. We also investigated the use of more complex models with more than one convolutional layer. However, the one convolutional layer gave the best results in terms of validation accuracy and loss. We use the Basset loss function, which was adopted by Basset [120] and ChromDragoNN [165], both of which predict chromatin accessibility across various cell types. Moreover, the 230 datasets obtained, with a total of of 194 unique TFs, are those that passed the filtering step in stage 1. That is, we discarded 20 datasets with a specificity or sensitivity of less than 0.2 in stage 1.

Stage 2 We then adopt a transfer learning multi-modal approach for stage 2, inspired by ChromDragoNN [165] (see Figure 4.1). In this stage, not only do we consider the genomic sequence of the peak, we also include cell type specific and cell type general information as input to the model. In stage 2 training, the genomic sequence is first input to the convolutional layer from stage 1, which is initialized using weights of the best pre-trained model with the optimized hyperparameters from stage 1. The convolutional layer from the stage 1 model is then concatenated with the output generated by the fully connected layer for either the cell type specific or cell type general data (refer to "CL General OR CL Specific" in Figure 4.1). The concatenated vector is then passed through a fully connected layer. Finally, the output node then determines whether or not this genomic sequence is bound in that specific cell type.

The negative log likelihood loss function with the SGD optimizer is used for stage 2. We use the same multi-label balanced sampler in stage 1 to address the class imbalance issue. Moreover, in addition to the stage 1 hyperparameters, stage 2 hyperparameters include number of neurons per layer before concatenation, number of neurons per layer after concatenation, freeze pretrained model, dropout probability and momentum rate. We use Ax for hyperparameter tuning [22]. The area under precision-recall curve (AURPC) is used to evaluate the primary performance of the stage 2 models.

4.6 Discussion

The complex structural and biochemical nature of protein-DNA interactions has made it difficult to fully understand how various factors influence transcriptional regulation and differential binding. We conducted a wide-scale investigation of TF and AB combinations across various cell types to identify and quantify differential binding preferences of TFs.

The DNA signatures constructed by our deep learning approach can account for many factors that could determine the binding preferences of a TF, such as its intrinsic binding preference, chromatin accessibility or co-binding factors. Different TFs display varying degrees of cell type specificity in their binding preference. For instance, our close examination of ATF7 found substantial cell type specificity, whereas we saw virtually none for CTCF. Such cell type specificity, or lack thereof, is also reflected in learned DNA sequence representations and motif enrichment analyses.

Differential binding analysis is one of the many key methods used for uncovering the relationship between TFs and the human genome in mechanisms, for example, leading to differential gene expression in tissue development or disease occurrence, or cooperative binding of TFs with other proteins and TFs. Binding site identification, or peak calling, of ChIP-seq can tell us whether a TF is binding the same or different sites in different cell types. However, we have shown here that peak overlap has negligible statistical association to the presence or absence of a DNA signature of differential binding.

Other deep learning approaches, such as MTTFSite [286] and Phuycharoen et al. [192], have also explored differential binding of TFs across cell types. While MTTFSite and Phuycharoen et al. adopt a similar learning framework to SigTFB in stage 1 training, in terms of using a multi-task model, their problem formulation and objective fundamentally differ. In MTTFSite, for example, prior to training, shared non-unique cell type instances are defined as bound regions across cell types that overlap by at least 100bp, while the remaining bound instances that do not overlap are cell type specific. In SigTFB, however, the model is given all instances as input and learns to differentiate non-specific versus cell type specific instances. The negative instances for a specific cell type in SigTFB are bound regions in other cell types, while in MTTFSite and Phuycharoen et al. negative instances are unbound regions in all cell types. SigTFB essentially learns to differentiate between shared and unique motifs in cell types from only bound regions. Additionally, the scale of the study differs. MTTFSite and Phuycharoen et al. investigate TFs in a total of 5 and 3 cell types respectively, while SigTFB explores all TFs in ENCODE with at least more than 2 cell types available, resulting in a total of 35 cell types across all TFs. Moreover, while MTTFSite trains a shared multi-task model on all cell types, then evaluates the network on previously defined cell type specific features, SigTFB trains a private network for each cell type in stage 2, using the weights from pretrained multi-task model of stage 1, to learn the cell type specific features of a cell type via the concatenation of one-hot cell type specific encoding. Through this approach, SigTFB demonstrates the varying degrees of TF specificity across cell types on wide scale, and shows the common non-specific preferences of a TF, as well as the unique cell type specific ones.

Similar to Novakovsky et al [170] and ChromDragoNN [165], we display the effectiveness of transfer learning in a multi-task deep learning framework for the prediction of binding profiles genome wide. Unlike these approaches, however, which mainly focus on cross cell type prediction, where models are trained on some cell types and tested on other cell types with limited data, we use transfer learning to acquire exclusive features per cell type. The multi-task setting in the first stage of learning allows the model to learn generalizable shared and unique features across cell types. Then, with the use of transfer learning, the model is constrained to learn cell type specific features, allowing the learning of a set

of motifs that cause cell type specificity. In addition to the type of learning used, data representation, the criteria chosen for model evaluation, and the hyperparameters selected are important factors we account for during the learning phase to achieve a more accurate prediction of binding profiles at cell type resolution.

Most deep learning approaches, such as DeepBind [6], MTTFSite [286] and DanQ [197], do not investigate the differences in ABs for the same TF when analyzing ChIP-seq experiments. We hypothesized that ABs could greatly influence the quality of ChIP-seq experiments. The polyclonal nature of ABs in ENCODE, for example, may result in ABs targeting the same protein to have different specificities, affinities and off-target binding. As a result, due to the lack of study on the consistency of functionality and performance of ABs across TFs ENCODE wide, we separate experiments from different ABs for a particular TF, and investigate the consistency in binding preference across different ABs for the same TF. Overall, we find consistency across ABs for most TFs ($\approx 88\%$), while for others the consistency is less apparent. Still, any lack of consistency in output may be due to other factors, such as the quality of the ChIP-seq dataset, the controls selected for peak calling, or the cell types available.

Information leakage occurs when the test data leaks into the training data. This leads to misleading high generalization performance as essentially the model is being tested on the same data it was trained on. Consequently, to avoid information leakage in SigTFB, the data is divided into training and test datasets prior to any training, and the test datasets are never touched while training. Additionally, we use 10-fold cross validation to repeatedly randomly divide our training dataset into training and validation – the model is trained on the training dataset and validated on the validation dataset. We also preprocess the training and testing datasets separately – preprocessing the overall dataset may influence the training set from the test set. However, although we used many methods to avoid information leakage, it can not be 100% guaranteed when dealing with ChIP-seq peaks. Peaks in the training and testing datasets may be correlated depending on the peak caller and the threshold used to determine whether a region is a peak or not.

The fundamental reasoning behind our study is that, by understanding how a deep neural network can code for cis-regulatory regions in differentially bound cell types, we can deduce the extent of cell type specificity of various regulatory proteins, and ultimately study their impact on downstream genomic analysis. Nonetheless, we acknowledge some limitations. First, the fact that a TF does not show cell type specificity in the cell types available from ENCODE does not imply that it will not show cell type specificity in other cell types. The human genome contains almost 1400 TFs [185], and despite the enormous effort of the ENCODE consortium, we found only 194 distinct TFs assayed in more than one cell type meeting our data set criteria. It is thus impossible to detect cell type specific binding for the vast majority of TFs, and it is uncertain whether other TFs may show specificity in other cell types. This underlines the importance of continued empirical study of TF binding in a wide range of cell types. A second limitation is that, despite best efforts, deep learning can at times fail to solve a prediction problem, even when a solution is possible in principle. There may be TFs for which we failed to detect a cell type specific signal, even when one is present. On the other hand, our careful checks against overfitting suggest that when a cell type specific signal is present, it is likely genuine, especially when

it is backed up by additional motif enrichment analyses. Thus, our results are best viewed as providing evidence for cell type specific DNA signatures in many TFs, while providing evidence against the same, without ruling it out, for other TFs. Thirdly, assumptions made regarding the network architecture, such as the 101 bp input sequence or fixed filter widths, may limit the learning capabilities of SigTFB. For instance, its inability to detect widely spaced motifs or motif pairs with fixed spacing, suggests that some DNA signatures relevant for cell type specificity may be possibly missing. Furthermore, in this work, we use ChIP-seq data due to its high availability and accessibility for multiple TFs and cell types. While ENCODE has many standards in place to ensure high data quality, other experimental approaches, such native ChIP [205] or ChIP-exo [206], may provide less noisy, higher resolution and more precise estimates of TF-DNA binding, and thus may ultimately improve the search for DNA signatures. Finally, while the degree of cell type specificity may be inferred from the TF-AB models, one can not directly infer the cause of such results. Thus, future work may further explain the rational behind such cell type specificity with a more in depth analysis of the different factors that cause cell type specificity in TF binding.

4.7 Conclusion

Many TFs are known to bind to different genomic sites in different cell types. Here, we demonstrated that for some of these TFs, different binding sites are associated with different DNA signatures, while others are not. We developed a deep learning prediction framework that is capable of detecting such DNA signatures. Our results provide a basis for future research into more specific molecular mechanisms that are reflected by these signatures.

Chapter 5

Motif Enrichment and Cell Type Specificity in Transcription Factors

The majority of this chapter is extracted from "Cell Type Specific DNA Signatures of Transcription Factor Binding" by Awdeh, Turcotte and Perkins submitted for publication [17]. The source code for SigTFB involving the pre-processing of ChIP-seq data, classification and downstream genomic analysis is available at https://github.com/aawdeh/SigTFB. The data used is available at https://doi.org/10.20383/103.0605.

5.1 Author Contributions

Aseel Awdeh recognized the need for a better understanding of transcriptional regulation, devised a method to address the limitations, wrote the software implementation, conducted all the experiments and wrote the manuscript, all with input and guidance from Theodore J Perkins and Marcel Turcotte.

5.2 Overview

The discovery of patterns from transcription factor (TF) binding sites is a known challenge in molecular biology. To further explain the biology behind a TF's cell type specificity, or lack thereof, we conduct a wide scale motif enrichment analysis of the 194 TFs in question. We show that the presence of alternate motifs correlates with a higher degree of cell type specificity in TFs, such as ATF7, while finding consistent motifs throughout is usually associated with the absence of cell type specificity in a TF, such as CTCF. In particular, we observe that several important TFs show distinct DNA binding signatures in different cancer cell types, which may point to important differences in modes of action. Moreover, we find that motif enrichment sometimes correlates with gene expression in TFs with higher cell type specificity.

5.3 Background

Motifs, the building blocks of regulatory information, are defined as short, recurring sequence patterns derived from DNA, RNA or protein sequences that share common biological properties [63, 64]. In protein binding experiments, such as in-vivo chromatin immunoprecipitation followed by sequencing (ChIP-seq) or in-vitro systematic evolution of ligands by exponential enrichment (SELEX), TFs bind to multiple positions along the genome in a sequence specific manner. The uncovering of the genomic occupancies, and consequently the motif patterns of TF binding sites, is essential for the characterization of TF proteins, and ultimately for the study and modeling of gene expression.

Sequence specific models for TF binding motifs are often deduced from data produced by genome wide high throughput sequencing technologies, such as ChIP-chip or ChIP-seq, or in-vitro sequencing technologies for a selection of TF binding sites, such as protein binding microarrays (PBMs) or SELEX. Several methods, most of which commence with the alignment of putative TF binding site sequences, have been proposed to represent the sequence specificity of TFs [174]. A single consensus DNA sequence, for example, can be used to model the binding preferences of highly specific interacting proteins [64, 223], such as restriction enzymes. The complexity of protein-DNA interactions, with regard to variations in binding affinity and sequence degeneracy, has become increasingly evident. As such, the inability of a consensus sequence to represent the binding preference variations of a TF has resulted in the use of regular expressions [64], position weight matrices (PWMs) [235, 236, 265] or other methods, such as dinucleotide models [222] or Markov models [220], to model the sequence specificity of TF binding. Despite the simplifying assumptions of PWMs, in terms of independent base positions and failure at times to detect low binding affinity sequences or important biological features, such as gaps and multiplicity, they are the community standard for providing a sufficient approximation of the true sequence specificity of TFs.

Several computational methods have been developed to either directly or indirectly infer PWM parameters from binding data to ultimately obtain realistic, compact and accurate models of TF binding specificity – a review of the different approaches can be found here [173, 255]. More recently, with the advent of deep learning and the plethora of high throughout sequencing data, deep learning models have been adapted to characterize DNA-binding protein specificity. Examples include, but are not limited to, DeepBind [6], DanQ [197] and FactorNet [198]. These approaches indirectly estimate the sequence specificity model of a TF via the classification of protein bound and unbound sequences. Through the mapping of DNA sequences to binding classes, deep learning approaches are able to more efficiently capture the complex binding preferences of a TF.

Explaining model predictions is a critical next step after the application of deep learning in genomics. It is essential, for example, to uncover how and why particular sites are selected and what motif patterns drive certain outputs. Various interpretation techniques have been deployed to justify and verify that deep learning approaches are indeed learning biologically relevant representations of TF binding sites. A simple strategy adopted by many [6, 120, 157, 170, 180, 197, 198] involves the visualization of convolutional filters

in convolutional neural networks, as they are known to detect local and global spatial patterns in genomic data. Another method projects the activation values of the neurons in the final fully connected layer into two dimensions using dimensionality reduction methods, such as the t-SNE algorithm [157]. Other approaches include perturbation based methods [6, 285], where the model input is altered and change in output is observed accordingly, and gradient based approaches [151, 221, 224, 230, 238], where the importance scores per nucleotide per input position along a sequence are computed using the gradients of the output with respect to the input.

Some interpretability techniques, such as visualizing convolutional filters or applying t-SNE on the final fully connected layer, provide a global assessment of the motif patterns found in all sequences. Others, such as perturbation and gradient based approaches, however, focus on interpreting one sequence at a time to obtain scores at individual base positions. However, the analysis of one sequence at a time may lead to an inadequate depiction of the motif patterns found. Thus, it is essential to analyze all motif patterns across all sequences. The motif discovery problem is formulated as deriving motif patterns at unknown positions with various lengths from a set of sequences. The derivation of motifs in a unsupervised manner may involve the use of enumeration based algorithms [19, 184, 190], which are based on the exhaustive enumeration and search of all possible k-mers in a set of sequences, or more sophisticated deterministic data driven probabilistic approaches, such as Multiple Expectation maximizations for Motif Elicitation (MEME) [21], or stochastic data driven probabilistic approaches, such as Gibbs Sampling [137]. Other motif enrichment algorithms, on the other hand, such as TomTom [90], conduct motif comparison and identify known motifs that exist at a statistically significant level in a set of sequences from databases, such as TRANSFAC [158] and JASPAR [44]. These databases provide a curated and systematically organized library of known TF binding preferences for TFs across various species.

In Chapter 4, we implemented a deep learning approach, called SigTFB (Signatures of TF Binding), to detect differences in DNA signature for a TF across cell types and quantify the degree of cell type specificity. The analysis was conducted on 194 distinct TFs. In this chapter, we aim to further elucidate the biology behind a TF's cell type specificity. We derive the motif patterns learned by the TF-AB models for both cases, cell type general and cell type specific. Cell type general is when the model is not informed of the cell type, while with cell type specific, the model is informed of cell type. We investigate the features or motif patterns driving cell type specificity, or lack that of, for a specific TF-AB model. The goal is to not only identify common motif patterns across cell types for a specific TF, but to also infer unique patterns that drive cell type specificity in certain cell types versus others for a specific TF. We build on SigTFB, developed in Chapter 4, and investigate the DNA signatures associated with cell type specificity in various TFs, mainly ATF7 and CTCF. We then conduct an overall assessment of motif enrichment across all TFs from different TF families. Multiple motif enrichment methods, such as in silico mutagensis, visualization of convolutional layers and t-SNE plots, are used to ensure consistency and reliability of results found.

5.4 Results

5.4.1 Determining the DNA signatures associated with cell type specificity for ATF7.

We look into the DNA signatures/motifs that cause the network to predict binding differently in general versus specific cell types for ATF7. Many approaches have been proposed for deriving important features from trained neural networks. In this section, we use several different methods to ensure consistency and for better interpretation of the model.

First, adopting a similar approach to AI-TAC [157], we use the t-SNE algorithm to present each ChIP-seq peak per cell type by its activation values in two dimensions across the 99 neurons of the final fully connected layer of the stage 2 model. Figures 5.1a and 5.1b show the ATF7 t-SNE plots for both cases of CL General and CL Specific respectively. Each point represents an instance, and is colored depending on which cell type to which it belongs. When predicting for general binding (Figure 5.1a), the DNA sequences, as encoded by the final network layer, appear as an undifferentiated mass with no obvious clustering structure, although the peaks from some cell lines do tend to be on one side or the other of the mass. Only bound instances per cell line are shown in Figure 5.1a. However, when CL information is provided, the network's representation of the sequences groups them perfectly by cell type (Figure 5.1b). Similar trends were observed when applying t-SNE to other cell type specific TFs, such as CREM and SP1, as seen in Figures 5.2 and 5.3.

We then explore the influence of the convolutional filters from the convolutional layer and their projections into this space. To do this, we converted the filters into PWMs and then used TomTom [90] to search for the PWMs in the JASPAR database [44] (See Methods). Our emphasis was on filters with significant similarities to known motifs of TFs in JASPAR; we found that filters without significant matches usually captured partial or variant motifs for known TFs. We note that filters could have more than one significant motif match, however, we focused on the single best match for simplicity. For example, ≈40% of the filters matched best to the JUND motif, a basic leucine zipper factor. Figure 5.1c shows that the JUND motif is present in many bound sequences across the four cell lines, although it appears more enriched in subclusters of the MCF-7 and HepG2 peaks. ATF7 and JUND both have basic leucine zipper domains, and are known to physically interact [61]. Another set of filters matched the SP2 motif. Figure 5.1d shows this motif is found in many fewer peaks, and primarily a subgroup of GM12878 peaks. Notably, C's and G's are enriched in the SP2 binding site motifs in this cell type, although a link between SP2 and ATF7 has yet to be established.

To further explore the DNA signatures the network learns, we use another interpretation technique called in silico mutagenesis (See Methods). We apply in silico mutagenesis to the ATF7 test sequences to infer TF binding sites per cell type. We compute the differences in network output when altering each element of the DNA input to each other possible nucleotide, and obtain mutation maps per sequence per cell type for both cases of CL Specific and CL General. The white color in the mutation maps indicates no change, while

blue or red refer to a drop or increase in predicted probability of binding respectively. Examples of mutation maps for different test sequences can be seen in Figures 5.1e to j. The mutation maps labeled "CL General" visualize the influence of the perturbed nucleotides along the sequence with cell type general information, while those labeled with the cell type name refer to the cell type specific state. Below the cell type name in the mutation maps is an indication of whether or not the sequence is a peak in that specific cell type. The network-predicted peak probability is included below that.

The sequence in Figure 5.1e yields the same result for both cases of cell type specific and cell type general across all the four cell types. The mutation maps highlight the subsequence "TGACGTCA", which is equivalent to the ATF7 motif profile in JASPAR. The network output is especially sensitive to the ATF7 motif when predicting for MCF-7 and HepG2, and less so for K562 and GM12878, although it still correctly predicts that peak for all cell types. Next, we focus on the sequences in Figure 5.1f, g and h. These sequences are peaks, and correctly predicted to be peaks, in MCF-7 and HepG2, and not peaks, and correctly predicted not to be peaks, in GM12878 and K562. In Figure 5.1f, the ATF2 motif of "ATGATGTCAT" is observed in HepG2 and MCF-7, but not detected as important in GM12878 and K562 nor cell type general. Indeed, altering some nucleotides in K562 and GM12878 actually increases the peak probability of the sequence. Other ATF7 motif variant patterns, such as "ATGACATCAT" and "TGATGCAAT", are observed with cell specific information in Figures 5.1 g and h respectively, where the motif pattern is clearly apparent in HepG2, barely apparent in MCF7, and non-existent in K562 and GM12878. Conversely, we observe alternate behavior in Figure 5.1i – where the sequence is considered to be a peak, and predicted to be a peak, in K562 and GM12878, and not a peak, and predicted not to be a peak, in HepG2 and MCF-7. No motif patterns are highlighted along the entire 101bp peak length, even though the network correctly predicts the peak for K562 and GM12878. It is not clear whether the network is predicting a peak by "default" and then suppressing that prediction in HepG2 and MCF-7 due to lack of a satisfactory motif or some competitive binding signal, or if there is some other multiply-present, redundant pattern that causes a positive prediction for GM12878 and K562–perhaps some co-binding factor with a motif very unlike the ATF family motifs.

To more systematically connect the important input sequence regions identified by in silico mutagenesis with known TF binding motifs, we extracted subsequences of length 31bp centered on the nucleotide with highest impact on prediction, and searched for enriched motifs using FIMO [86] (See Methods). Figure 5.1k (top) shows a heatmap of the ratio of the number significant motif hits to the total number of bound peaks per cell line per motif. We notice two main clusters of cell lines – each of which shows enrichment for a subset of motifs. As expected, the bZIP motifs, such as ATF2, ATF3, ATF4, ATF7, CREM and JUND are enriched in all four cell lines, with relatively more enrichment seen in HepG2 and MCF-7 as opposed to K562 and GM12878. Many TFs originate from TF families with similar DNA binding motifs, where motif enrichment is expected, but this does not necessarily denote involvement. Moreover, similar to the t-SNE output, we notice mild enrichment of SP2 (and other motifs, such as KLF3 and KLF4) in K562 and GM12878, and no enrichment in MCF-7 and HepG2.

In addition to scanning the test sequences using FIMO, we obtain the RNA-seq gene

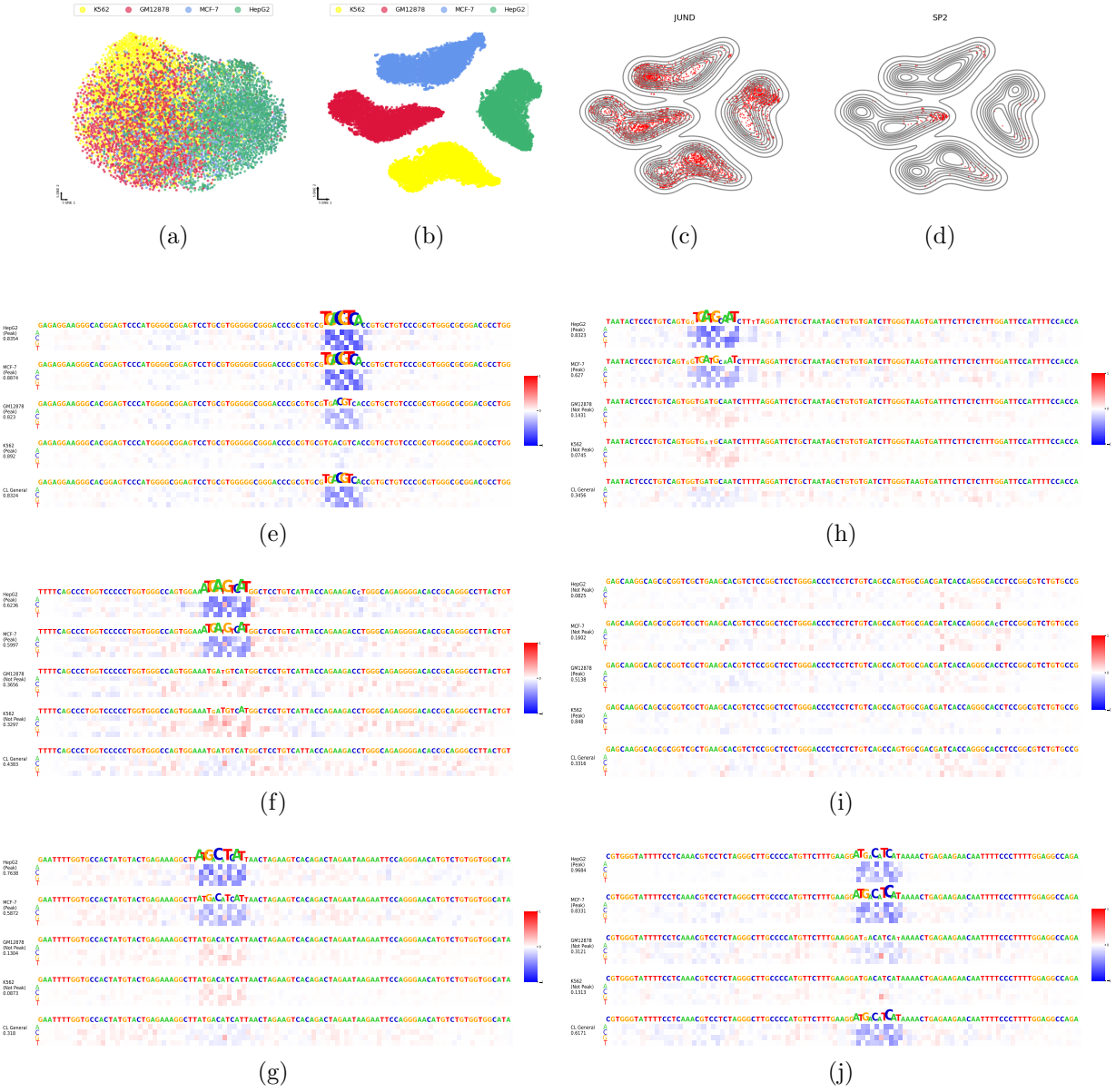
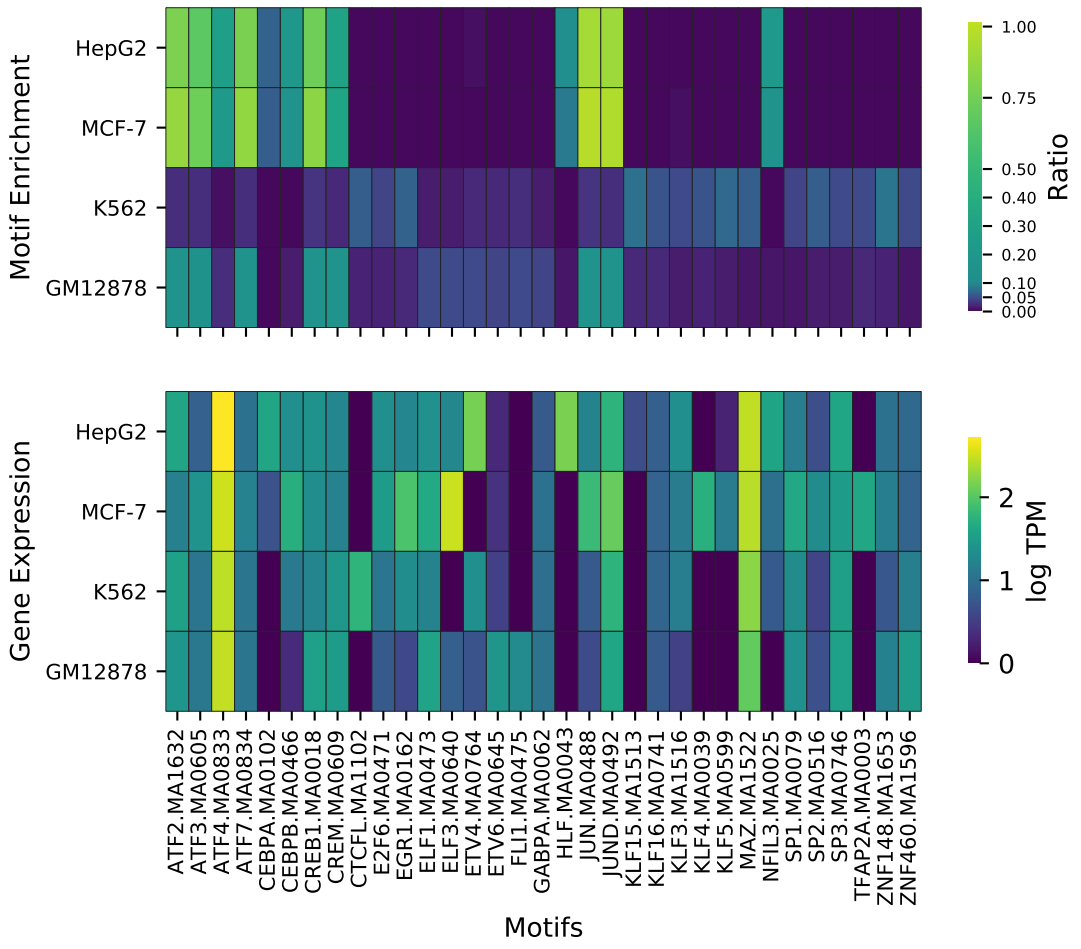
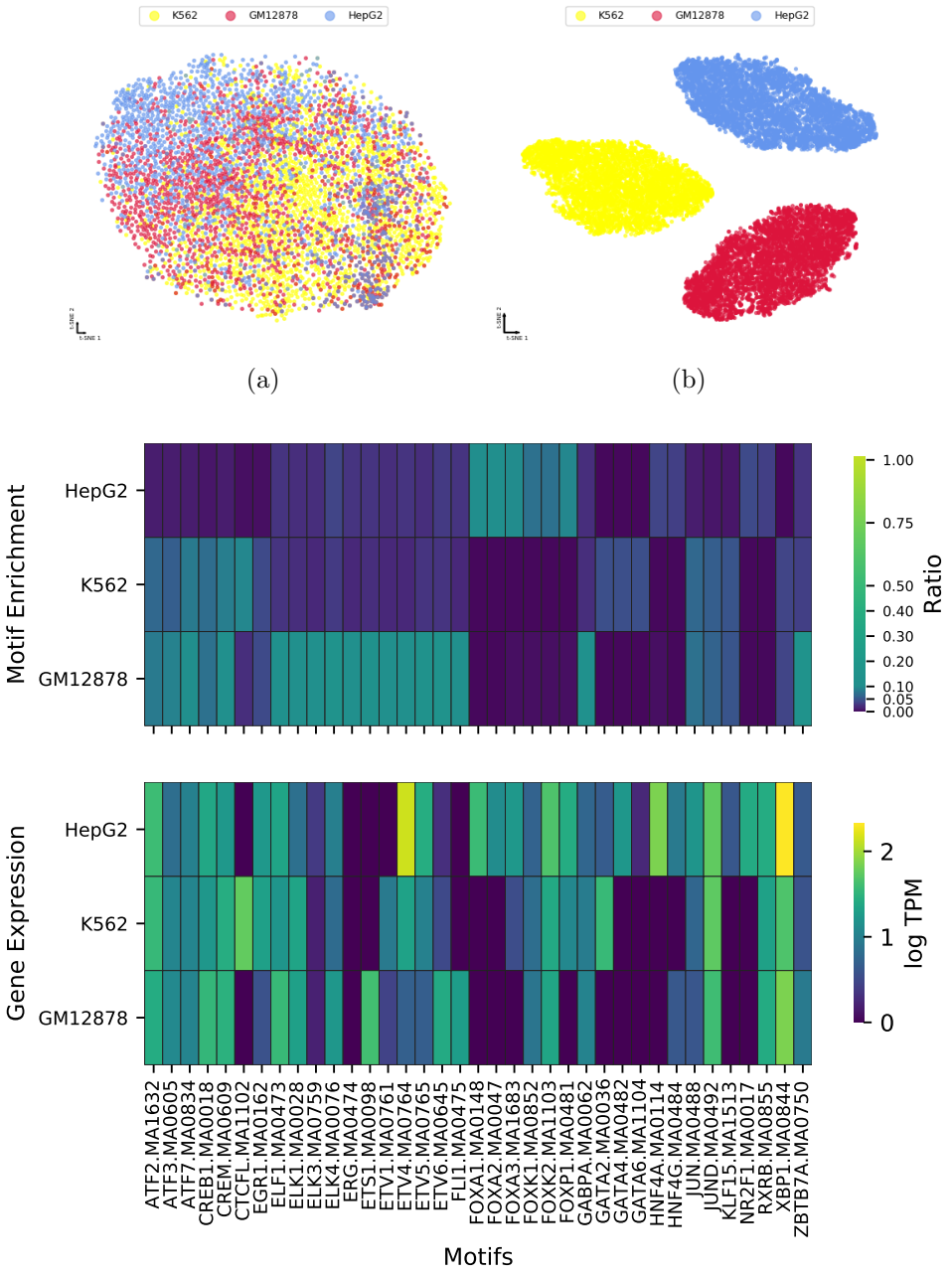


Figure 5.1: Motif enrichment for ATF7.ENCAB000BMO (a) t-SNE plot with cell type general concatenation. (b) t-SNE plot with cell type specific information. (c) and (d) are projections of the motifs JUND and SPI1 respectively onto the cell type specific t-SNE plots. (e), (f), (g), (h), (i) and (j) are mutation maps generated using in silico mutagenesis on different test sequences.



(k)

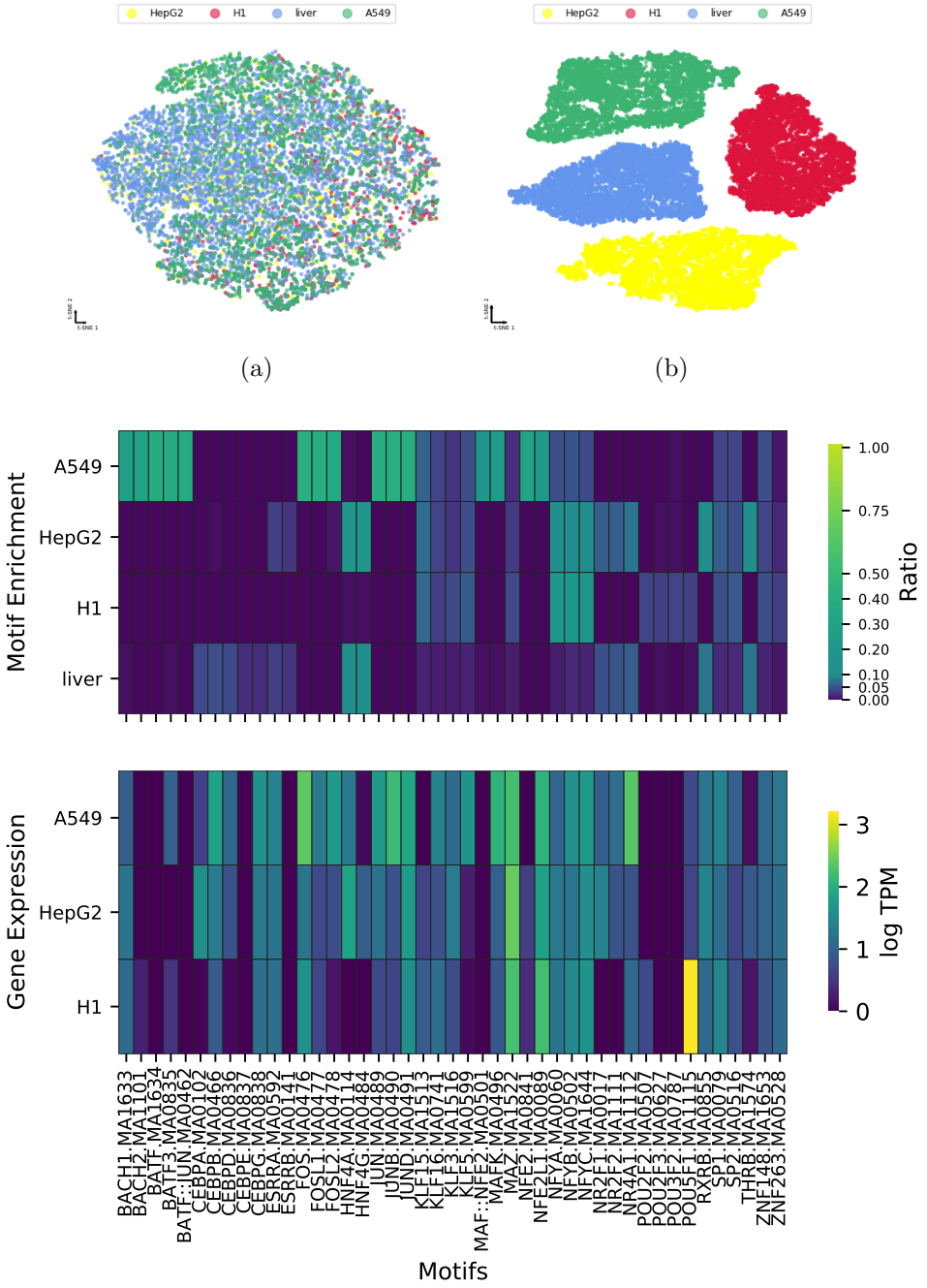
Figure 5.1: (continued) Motif enrichment for ATF7.ENCAB000BMO. (k) Heatmaps of motif enrichment (top) and gene expression (bottom) per cell line across motifs.



(a) (b)

(c) CREM.ENCAB000AAT

Figure 5.2: Motif enrichment of CREM.ENCAB000AAT. (a) t-SNE plot with cell type general concatenation. (b) t-SNE plot with cell type specific information. (c) Heatmaps of motif enrichment (top) and gene expression (bottom) per cell line across motifs.



(c) SP1.ENCAB000AKY

Figure 5.3: Motif enrichment of SP1.ENCAB000AKY. (a) t-SNE plot with cell type general concatenation. (b) t-SNE plot with cell type specific information. (c) Heatmaps of motif enrichment (top) and gene expression (bottom) per cell line across motifs.

expression for the motifs across the four cell types (See Methods), as shown in Figure 5.1k (bottom). For many TFs, such as ATF4, CEBPA, CEBPB, CTCFL, JUN and JUND, there are similar patterns in both motif enrichment and gene expression – meaning motifs that are enriched across the four cell types are also expressed in the corresponding cell lines. However, there are TFs showing motif enrichment unrelated to expression. Results may be due to similar motifs showing enrichment for something not expressed (such as KLF4), co-expression with other TFs without binding, or competitive binding with other factors. SP2, for example, is expressed in HepG2 and MCF-7, but its motif is not enriched in those peaks.

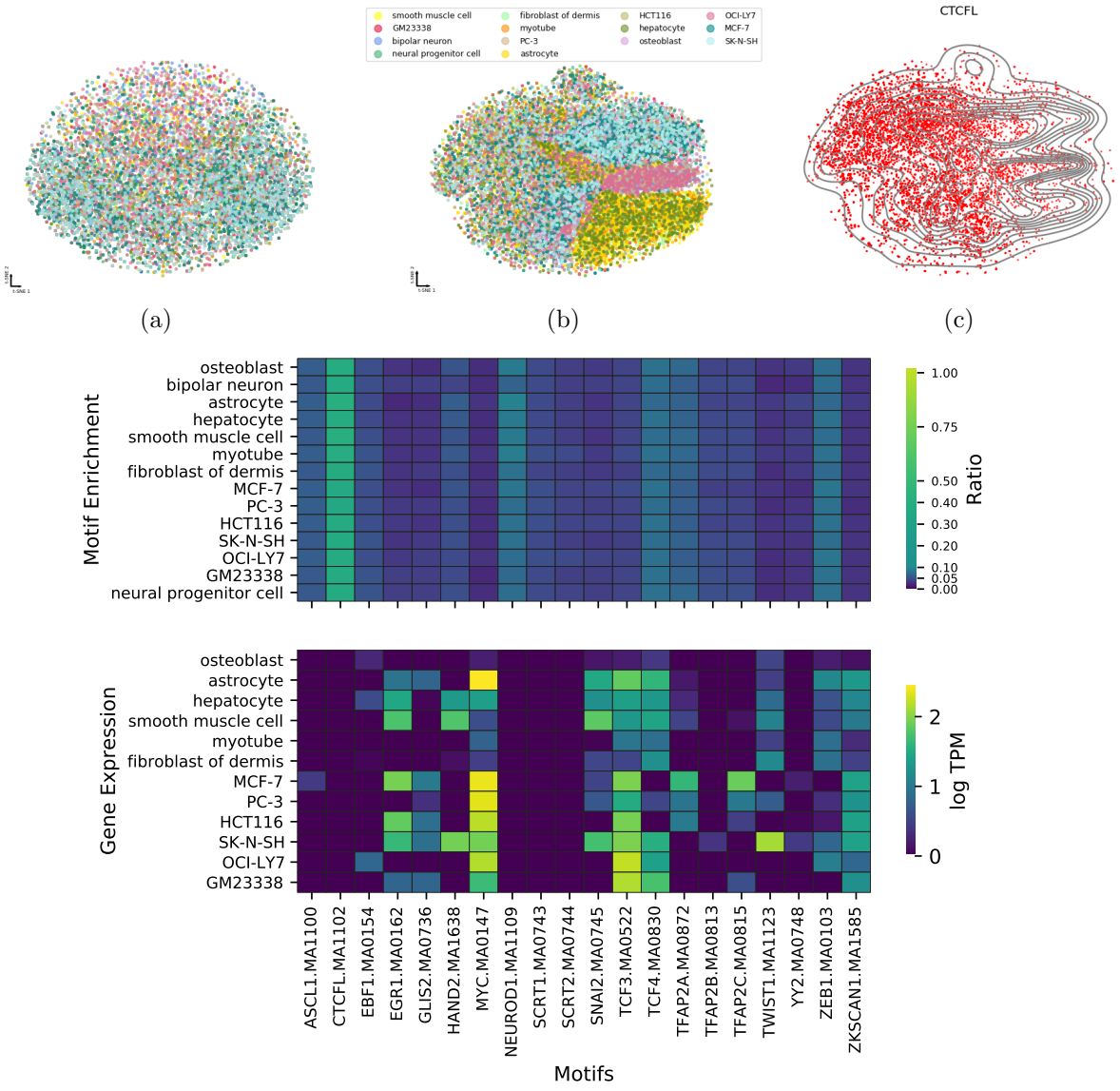
5.4.2 Determining the DNA signature associated with CTCF binding.

We next explore the cell type specificity, or lack thereof, in CTCF binding. Similar to our previous analysis on ATF7, we first apply the t-SNE algorithm on the last fully connected layer produced per cell line from the pretrained stage 2 model. We compare both cases of CL General and CL Specific for the TF CTCF and AB ENCAB000AXX (Figures 5.4a and b). The plot in Figure 5.4a has no clear structure, similar to what was observed for ATF7. However, in Figure 5.4b, there are distinct clusters of DNA sequences – not from individual cell types, but from groups of cell types. For instance, the yellow/green/orange cluster towards the bottom comprises primarily peaks from smooth muscle cells, astrocytes, myotubes, and hepatocytes. Some clusters are divided into two parts. The cluster encompassing the yellow, green and orange points, for example, is found in two different locations in the t-SNE plot. This is also true for the top cluster of purple, blue and brown points. Some degree of cell type clustering is apparent, however the cell type groupings of sequences are not as clearly evident as in ATF7. Only positive instances per cell type are displayed in the plot, however. Thus, a separation of some positive instances does not necessarily indicate that the model is performing well on the negative instances.

We also examined the filters in the first convolutional layer of the network to discover what known TF PWMs they resembled. In Figure 5.4c, we show the non-uniform distribution of CTCFL. CTCFL (CCCTC-binding factor like) is a paralog of CTCF, and thus both share the highly conserved zinc finger DNA binding domain [32, 62]. However, unlike CTCF, CTCFL is known to be more cell type specific [32, 62]. We notice that CTCFL avoids the lower right yellow/green clusters and the upper center purple/blue/green cluster. The cell types in the yellow/green clusters include the smooth muscle cell, myotube, astrocyte and hepatocyte.

Next, we use in silico mutagensis and FIMO to identify motifs enriched across the 14 cell types, as seen in Figure 5.4d (top). Unsurprisingly, different cell types show very similar patterns of motif enrichment, with the highest enrichment observed for CTCFL, and relatively high enrichment for other motifs, such as NEUROD1 and ZEB1. Additionally, similar to ATF7, we visualize gene expression per motif per cell line (Figure 5.4d (bottom)).

Analogous behavior is observed for CTCF using another AB ENCAB000AFR with a smaller number of cell types, as seen in Figure 5.5. With five cell types as opposed to 14,



(d) CTCF.ENCAB000AXX

Figure 5.4: Motif enrichment for CTCF.ENCAB000AXX. (a) t-SNE plot with cell type general concatenation. (b) t-SNE plot with cell type specific information. (c) Projections of CTCFL motif onto the cell type specific t-SNE plot. (d) Heatmaps of motif enrichment (top) and gene expression (bottom) per cell line across motifs.

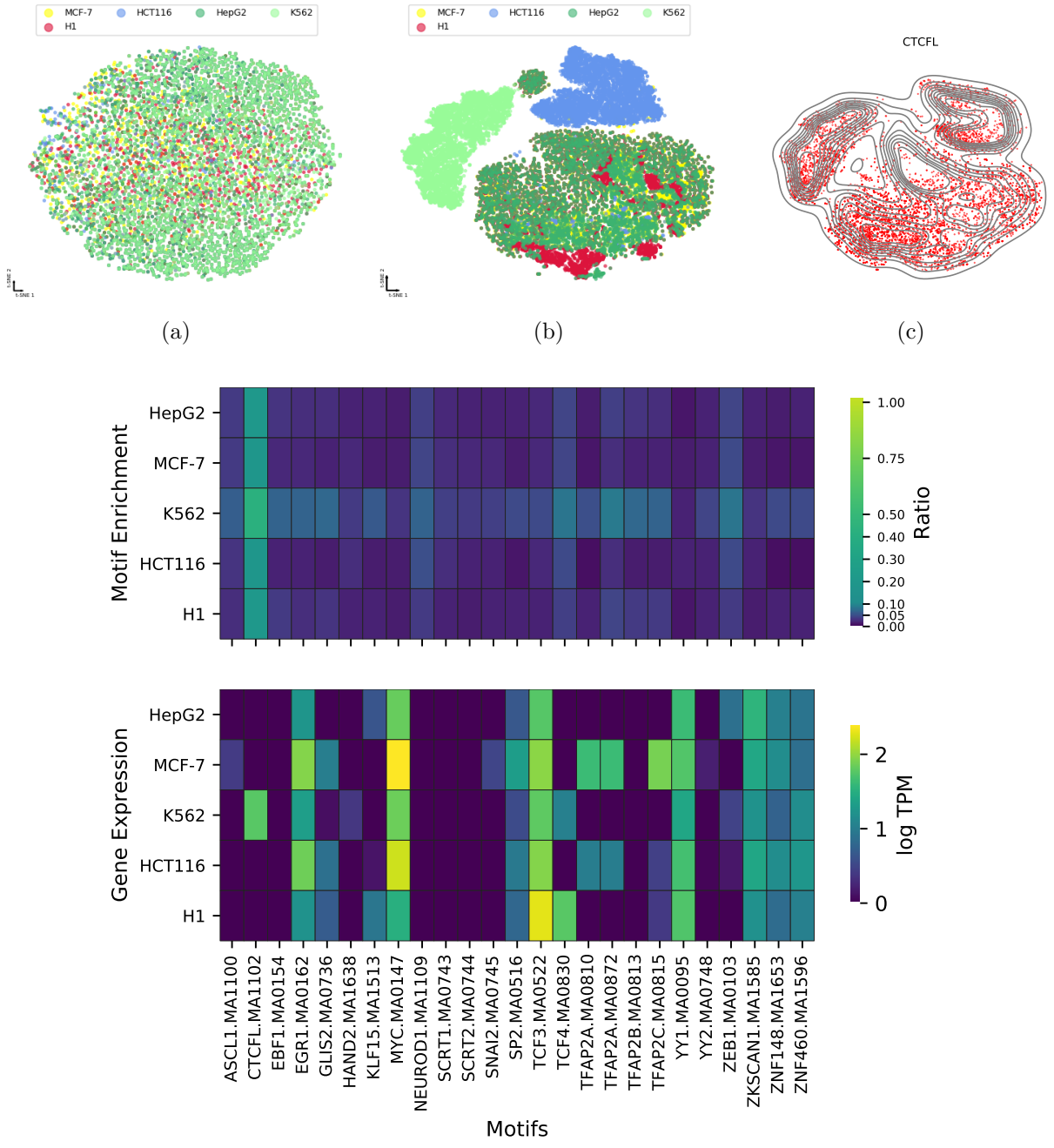


Figure 5.5: Motif enrichment of CTCF.ENCAB000AFR. (a) t-SNE plot with cell type general concatenation. (b) t-SNE plot with cell type specific information. (c) Projections of CTCFL motif onto the cell type specific t-SNE plot. (d) Heatmaps of motif enrichment (top) and gene expression (bottom) per cell line across motifs.

clustering using the cell type specific concatenation does not produce a distinct cluster for each cell type, rather it produces distinct subgroups of cell types that may be more cell type specific than others.

5.4.3 Learned motifs across TF families.

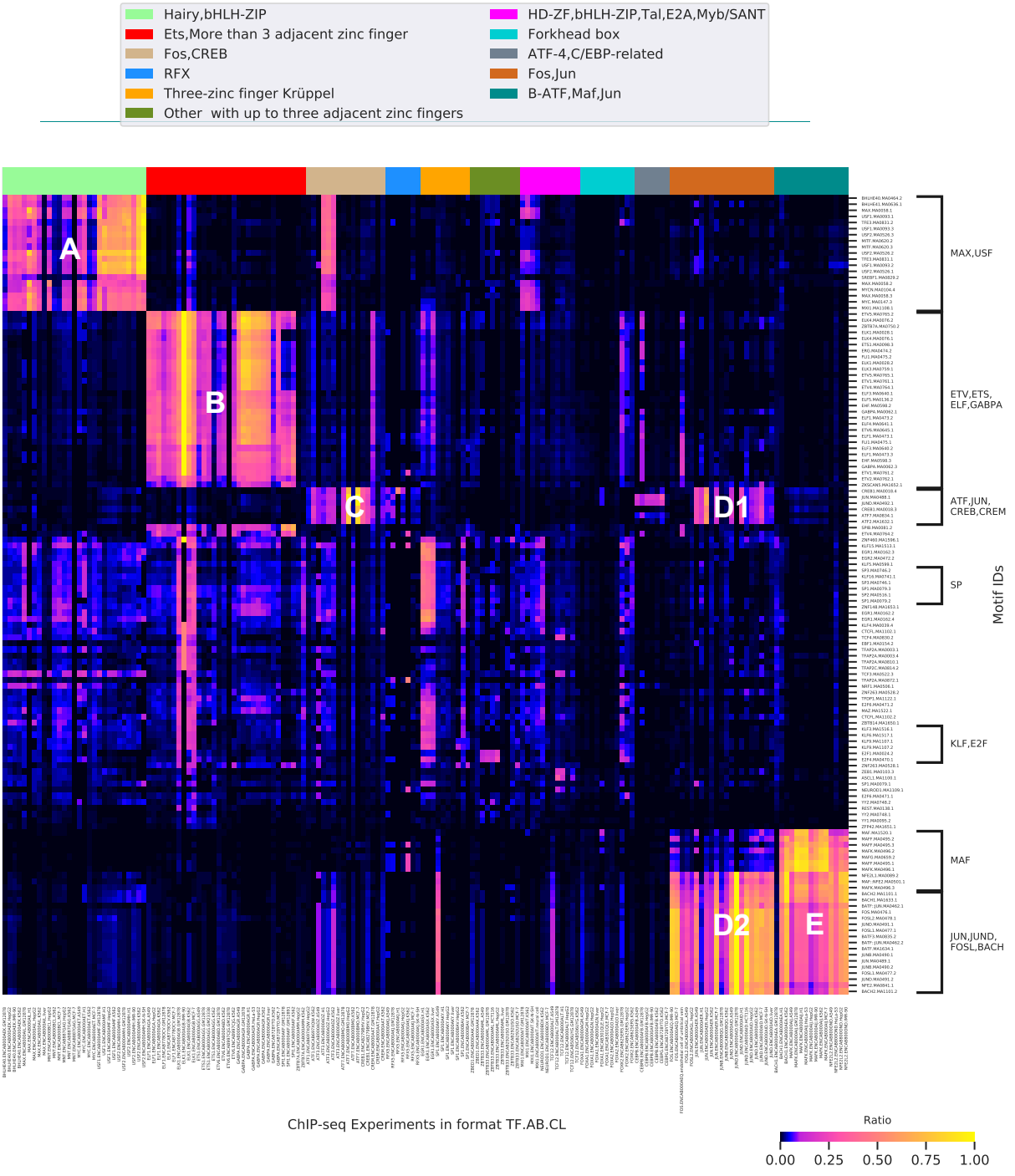
In this section, we examine the motif enrichment for various TF and antibodies (ABs). As previously discussed, we apply in silico mutagenesis and FIMO on all TF-AB models (See Methods). TFs from the same families have identical DNA-binding domains [7, 80]. As a result, we focus on homologous TFs that are part of TF families, as provided by the hierarchical clustering of the DNA binding domains in the JASPAR database [45, 80, 170]. This gives a total of 261 TF-AB and cell type combinations.

Heatmap of motif enrichment: Figure 5.6a visualizes the ratio of significant motif hits identified by FIMO to the total number of binding sites per TF-AB and cell type. The columns in the heatmap correspond to the ChIP-seq experiments in the form TF-AB-CL. The rows correspond to known motifs from JASPAR [44]. Each group of ChIP-seq experiments (columns) is part of a TF family, and is given a unique color as shown at the top of the heatmap. Different clusters of motifs are especially enriched in different clusters of ChIP-seq experiments (labeled A, B, C, D1, D2, E).

To more clearly visualize the trends observed, Figures 5.6b-g display enlarged heatmaps of each cluster. Cluster A, for example, consists of the TFs BHLHE40, MAX, MNT, MYC, USF1 and USF2, which are part of the bHLH-ZIP and Hairy TF family. We notice uniform patterns of enrichment across motifs and cell types in USF1 and USF2, meaning the level of motif enrichment across their corresponding cell types does not change. This indicates a lack of cell type specificity, and is consistent with the low degrees of cell type specificity of less than 0.1 (Appendix Table B.1). MAX, however, has a higher degree of cell type specificity of 0.15. This is due to the very low levels of enrichment motifs in liver, while the other four cell types have a higher and more constant level of motif enrichment. Similar behaviours can be seen for MNT with either antibody, and MYC. The TFs in this family, do not all show similar levels of enrichment across the motifs. For example, although the motif SREBF1 is enriched in all TFs, it is overall more enriched in BHLHE40, USF1 and USF2 in comparison to MAX, MNT and MYC. This may because the TFs MAX, MNT and MYX suppress SREBF1.

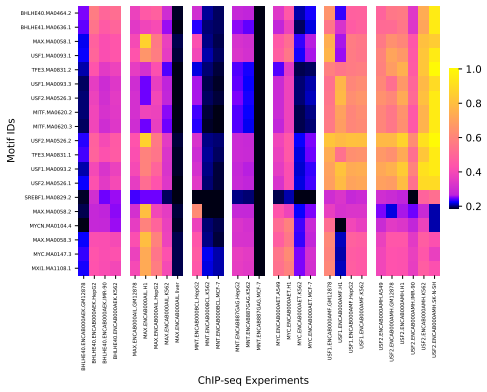
Similar trends of varying enrichment levels across different cell lines for the same TF and AB combination are also observed in the other clusters. For example, in cluster B for ETS1, we observe that most of the motifs (ELK1, ETV5, ETV4, GABPA, etc.) are enriched in the cell types A549, GM12878 and GM23338, but not in K562. As such, K562 has the highest difference in AUC between cell type specific ($\approx$0.68) and cell type general ($\approx$0.56) concatenation, in comparison to the other cell types.

The Fos/Jun TF family has two main clusters, D1 and D2. Cluster D2 (Figure 5.6f) shows varying levels of enrichment in most members of the Fos/Jun TF family across

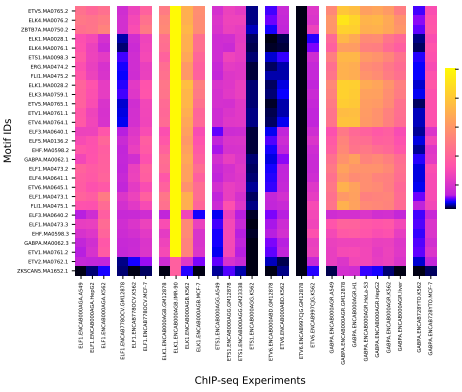


(a)

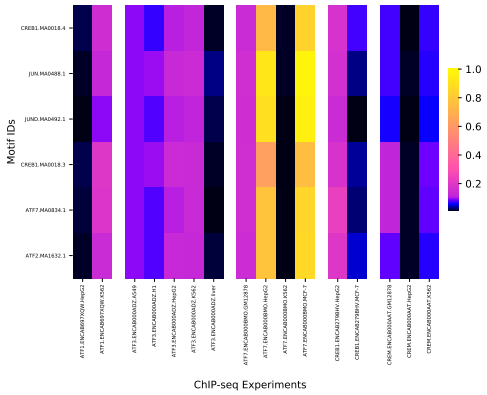
Figure 5.6: Overall motif enrichment. (a) Heatmap of motif enrichment. Each column corresponds to a TF-AB and cell type combination, and each row corresponds to a motif ID. Each TF family is given a unique color as seen at the top of the heatmap.



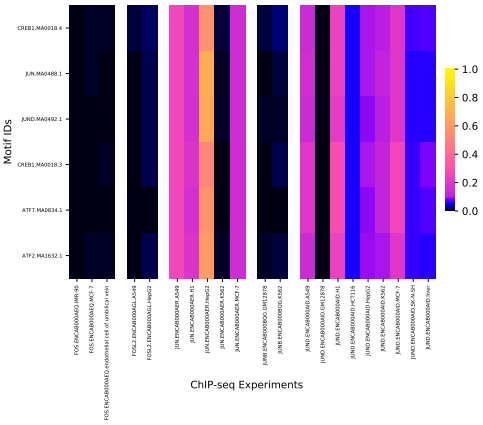
(b) Cluster A



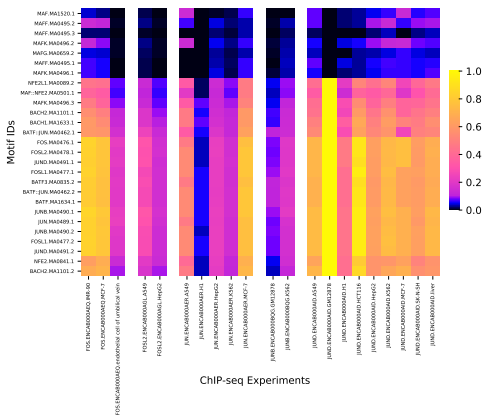
(c) Cluster B



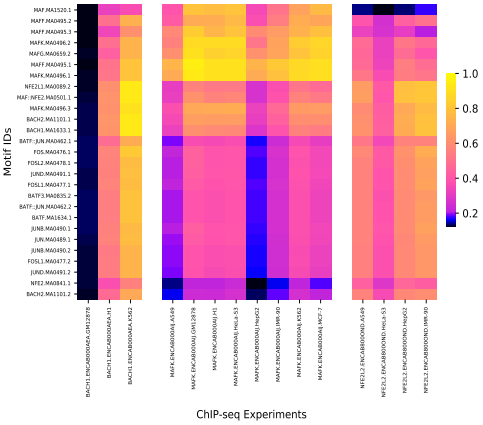
(d) Cluster C



(e) Cluster D1



(f) Cluster D2



(g) Cluster E

Figure 5.6: (continued) Overall motif enrichment. (b), (c), (d), (e), (f) and (g) are zoomed in heatmaps for clusters A, B, C, D1, D2 and E.

motifs from the MAF-related family, BACH, FOS/FOSL and JUN/JUND. For instance, the MAF motifs at the top of the heatmap show relatively little enrichment compared to the other motifs below, such as BACH1 and FOSL1, with some cell lines, such as JUN.ENCAB000AER.A549, having more enrichment for these motifs than others. Moreover, the cell lines, JUN.ENCAB000AER.H1 and JUNB.ENCAB000BQG.GM12878, have the lowest motif enrichment compared to other cell lines in the cluster. There are also observable differences between cells types for FOS, FOSL2 and JUN, which is consistent with the moderate level of cell type specificity obtained by SigTFB. Cluster D1 (Figure 5.6e), on the other hand, displays different trends for another set of motifs. Notably, JUN.ENCAB000AER.H1, which lacks enrichment for D2 motifs, shows significant enrichment for D1 motifs, including JUN and JUND motif variants that are different between blocks D1 and D2. Additionally, in all the cell types for FOS and FOSL2 in cluster D1, there is little to no enrichment for motifs CREB1, CREM, ATF7, ATF2 and ATF3.

Another group is cluster E, in Figure 5.6g, which consists of the TFs BACH1, MAFK and NFE2L2, part of the B-ATF/Jun/Maf TF family. For the ChIP-seq experiments MAFK and NFE2L2, we observe varying patterns of enrichment across their corresponding cell types. These results are consistent with the AUC differences obtained earlier of more than or equal to 0.1 – which indicates a higher degree of cell type specificity. MAFKs are part of the small MAF TF group, and are a member of the bZIP family. They are known for their role in gene expression regulation in multiple cellular processes. There is no sufficient evidence to suggest any cell type specific functionality, or lack that of, of MAFK [36]. Different, more apparent, trends are observed across cell types for BACH1. Cell types K562 and H1 show high levels of enrichment for all motifs, while cell type GM12878 displays no motif enrichment.

Ideally, ChIP-seq experiments with different ABs for the same TF identify similar peaks, and therefore similar DNA signatures in those peaks. Although Figure 5.6 shows consistency in motif enrichment across ABs for the same TF, exceptions to this notion do exist. Examples of consistent levels of enrichment in the same cell types for different ABs can be seen in MNT.ENCAB000BCL and MNT.ENCAB887GAG in Cluster A (Figure 5.6b), and ELF1.ENCAB000AGA and ELF1.ENCAB778OCV in Cluster B (Figure 5.6c). A lack of consistency in enrichment can be seen in Cluster B, however, more specifically in ETV6.ENCAB000ABD and ETV6.ENCAB997CJG for cell type GM12878. This can also seen in the drastic differences in AUC difference obtained for each of TF-AB, as discussed earlier.

Networks of motif enrichment: Figure 5.7 provides a graphical representation of the most significant motif hits (ratio$>$ 0.5) in cell types: A549, K562 and HepG2. Each node in the network represents a TF/motif. Directed arrows point from TFs representing ChIP-seq experiments (or TF-AB combinations) to known motifs enriched in that experiment. Enrichment values from different versions of the same motif are averaged. Arrows are colored according to which cell type(s) show enrichment. Appendix Figure C.1 displays motif networks for other cell types (GM12878, MCF-7, liver, H1, HeLa-S3, HCT116, IMR-90, SK-N-SH).

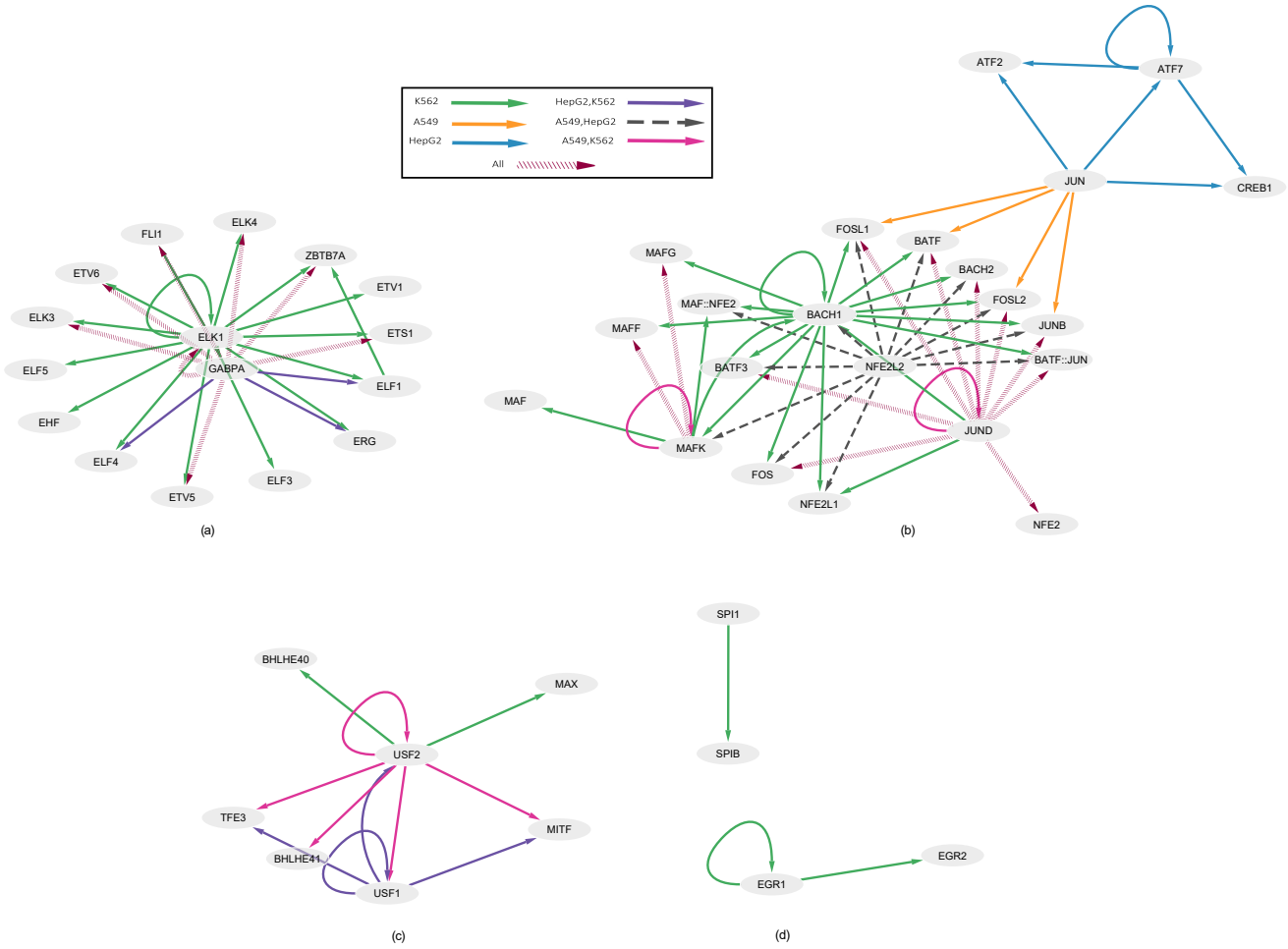


Figure 5.7: Motif enrichment network in the cancerous cell types: A549, K562 and HepG2. Each node represents a TF/motif, and the directed edges point from TFs of ChIP-seq experiments to known motifs enriched in that experiment. The various edge colors correspond to whether a motif is enriched in only one cell type, two cell types, or all three.

The three selected cell types are cancerous: A549 is lung carcinoma, HepG2 is hepatocellular carcinoma and K562 is leukemia. There are three main groups of TFs/motifs in Figure 5.7 (labeled a, b and c). Figure 5.7a, for example, is composed of TFs from the E26 transformation-specific (ETS) TF family. We notice enrichment of many ETS motifs, such as ETS1, ETV6 and ELK3, in all of three cell types for the GABPA TF (dashed maroon edges in Figure 5.7a). Abnormal levels of ETS motifs are often associated with cancer development and progression [101, 227]. Additionally, although many of the ETS motifs are enriched in all three cell types, ELF4, ELF1 and ERG are only enriched in K562 and HepG2, and not in A549 (purple edges in Figure 5.7a).

Figure 5.7b constitutes the FOS, JUN, MAF and ATF sub-families – part of the activator protein-1 (AP-1) TF family. We notice the enrichment of the AP-1 motifs: FOS, BATF3, FOSL1, BATF, FOSL2 and JUNB, in all three cell types for the JUND TF (dashed maroon edges in Figure 5.7b). AP-1 TFs play an important role in cancers and impact the proliferation, differentiation and apoptosis of cancerous cells [262]. While in most cancers such as breast [209, 246] and lung [111, 203] cancers, the AP-1 TFs act as oncogenes, in other cancers, such as leukemia, they may either act as oncogenes [37, 70] or tumor suppressors [175, 181, 240]. For instance, in spite of ChIP-seq data being available for the JUN TF in all cell types, we notice a difference in enrichment across the cell types. There is no significant enrichment of any motif for the JUN TF in K562, while in HepG2 and A549 the AP-1 TFs are enriched. This may be because JUN is suppressed in K562. Another example is BACH1, which shows enrichment for many of the AP-1 motifs, but only in K562.

Similar behaviours of variations in motif enrichment across cell types for a specific TF is shown in Figure 5.7c for the upstream stimulatory factors, USF1 and USF2. For example, the motifs TFE3, BHLHE41, USF1, USF2 and MITF are enriched in A549 and K562 for the TF USF2, while BHLHE40 and MAX are only enriched in K562 and not in A549. It is also important to keep in mind that ChIP-seq experiments may not be available for all three cell types. Thus, having enrichment only in one cell type may be due to ChIP-seq data being only available for that TF in that specific cell type, such as the case of TFs SPI1 and EGR1 in cell type K562 in Figure 5.7d.

Visualizing convolutional filters: Another method used to understand the motifs identified in each TF-AB model is the visualization of the convolutional filters (See Methods). We compute three parameters per filter per model: (a) the overall filter influence, (b) the number of activated sequences per filter, and (c) the q-value of the motif found by TomTom [90] for that filter. The overall filter influence is the difference in prediction with and without the filter when concatenating cell type specific information. It should be noted that the concatenation of cell type general or cell type specific information occurs after the computation of the convolutional layer activation values (as seen in Figure 4.1). As such, regardless of what the concatenation is per model, the PWMs/filters stay the same.

Figure 5.8 is a scatter plot which combines the three parameters per TF-AB across all TF-ABs. The y-axis and x-axis correspond to the ChIP-seq experiments in the form

"TF.AB" and the overall filter influence per filter per model respectively. The TF-ABs are arranged in alphabetical order. Each point in the scatter plot represents a filter for a specific TF-AB model. Only filters with significant q-values from TomTom are shown. The color gradient signifies the -log10 q-value of the TomTom match, while the size of the markers represents the fraction of the activated sequences. We notice varying numbers of unique filters per model across the ChIP-seq experiments. In some cases, such as 5 out 6 of the CTCF experiments, only one significant filter (CTCFL) is identified. Additionally, in some cases, the greatest fraction of sequences activate the shared motifs across cell types for a TF-AB. For ATF7, for example, a large portion of the sequences activate the JUND (bZIP) filter which is a common motif across the four cell types.

5.5 Methods

5.5.1 Feature attribution with in silico mutagensis and FIMO

We apply in silico mutagensis on test sequences per cell type for each of the 230 TF-AB pre-trained models. Using in silico mutagensis, we change each nucleotide to every other possible value and observe the output of each perturbed sequence. From the mutation map per sequence, we then extract a 31bp subsequence with the largest impact on output. That is, we select the nucleotide along the sequence that is most negatively influenced by perturbation, and extend it by 15bp in each direction.

To analyze the subsequences per cell type per TF-AB model, we then use FIMO 5.0.3 to search for motifs in the subsequences using known JASPAR human motifs [44] that are based on at least 1000 sites and had log p-values of at least 100. This gives us a total of 400 JASPAR motifs. For each cell type per TF-AB, and each motif, we find the ratio of the number of significant motif hits identified by FIMO to the number of total peaks for that cell type. We then find the top 20 motifs with the highest ratio. By using this approach, we account for enrichment as well as the number of peaks per cell type per TF-AB. We finally take the union of the these motifs across the cell types to construct the enrichment and expression heatmaps in Figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6.

Moreover, we obtain RNA-seq expression for the 35 distinct cell types across the 230 TF-AB pairs. The same cell type may appear in more than one TF-AB pair. We extract gene level expression data, both polyA and total RNAseq, in the form of transcripts per million (TPM) per cell type from ENCODE [56] – data that was generated using the uniform processing pipeline on unperturbed cell types. In the event that there is more than one RNA-seq experiment, be it polyA or total RNA-seq, available per cell type, the average TPM per gene per cell type is taken.

5.5.2 Filter Visualization

Motifs are derived from filters in the first convolutional layer of the network, using a similar approach to AI-TAC [157]. Each filter for each TF-AB model is converted to a PWM.

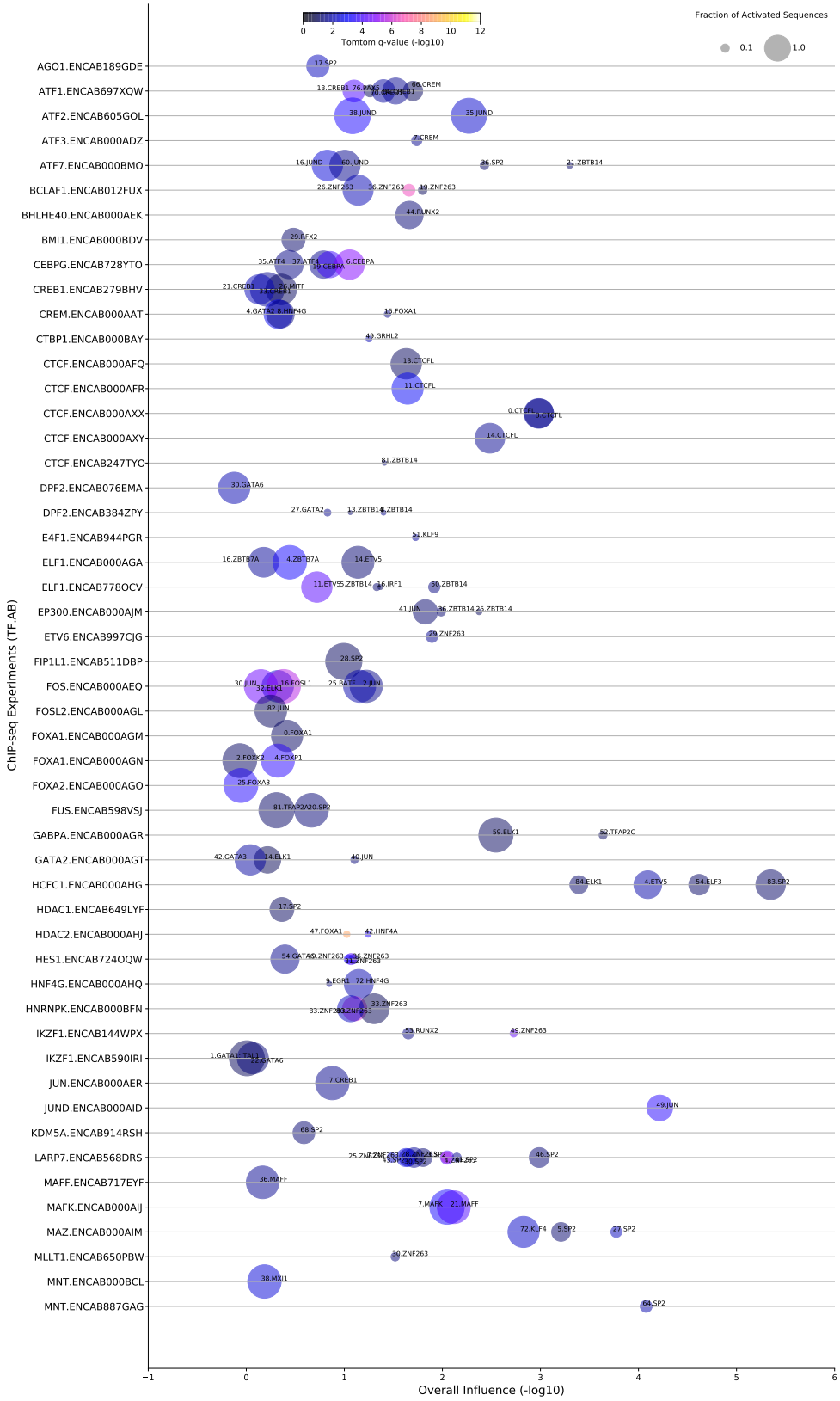


Figure 5.8: Visualization of filters per TF-AB model, with the color gradient depending on the -log10 q-values obtained from TomTom. The y-axis represents the TF-AB models, and the x-axis represents the overall filter influence. The size of the marker corresponds to the fraction of activated sequences for the filter.

Figure 5.8: (continued) Visualization of filters per TF-AB model, with the color gradient depending on the -log10 q-values obtained from TomTom. The y-axis represents the TF-AB models, and the x-axis represents the overall filter influence. The size of the marker corresponds to the fraction of activated sequences

Across all sequences for a specific TF-AB combination, subsequences of length 19bp with activation values greater than the set activation threshold are selected – subsequences that activate the filter are selected. Next, we construct a position frequency matrix from the aligned subsequences, and convert the position frequency matrix to a PWM using a uniform background nucleotide composition of 0.25. Since the number of filters per TF-AB model is set as a hyperparameter during training, this analysis yields a varying number of PWMs per cell type per TF-AB to capture the motifs detected by the filters. We assign known motifs to the PWMs derived from the filters in the first layer of the networks. The PWMs are compared to human motifs in the JASPAR database [44] using a quantifying tool called TomTom [90], and PWMs/motifs with significant -log10 q-values are shown. The fraction of sequences that activate each filter per model is also reported.

5.5.3 Filter Influence

We measure and quantify the impact of each filter per TF-AB for all models. To do this, we first remove each filter, one at a time, and replace all its activation values with the average activation value of that filter. Next, for all instances for a TF-AB model, we measure the difference in prediction before and after removal for that specific filter. That is, we use the modified filter instead of the original filter, and obtain a modified prediction output vector for the modified filter. The overall filter influence for each TF-AB model is defined as the squared mean difference between predictions with and without that filter.

5.6 Discussion

Understanding the inner workings of deep learning models is a challenge due to the non-linear nested structure and complexity of deep networks. In the previous chapter, we developed a deep learning tool called SigTFB to detect and quantify cell type specificity of TFs in different cell types, and showed how different TFs exhibit varying extents of cell type specificity in their binding preference. Here, we aim to further understand the differentiating motifs that lead to cell type specificity of a TF across cell types, and to also uncover the shared non unique motifs that result in the cell type general case.

We conduct an overall wide-scale motif enrichment analyses on TFs from different TF families, and display the varying levels of motif enrichment per family. Different interpretability techniques are used to uncover motif patterns corresponding to each of TF-AB models. We find that cell type specificity, or lack thereof, is often reflected in learned DNA sequence representations and motif enrichment analyses. A TF with a higher degree of cell type specificity typically corresponds to differentiating levels of motif enrichment across cell types, while a TF with low cell type specificity has relatively the same levels of motif enrichment across its cell types. Moreover, within the same TF family, most TFs display similar levels of enrichment for some motifs as opposed to others. Motif enrichment per cell type is also visualized as a network, where the most significant motifs enriched in the different experiments are shown. The networks display how and what TFs and motifs group

together for each cell type. Upon further investigation, we notice that the grouping of the different TFs and motifs is often validated by recent work in the literature. The different interpretability and visualization techniques provide a general reassurance of consistency in output.

Interpretability methods are used to understand and confirm that SigTFB is learning biologically relevant representations. A key challenge of all interpretability techniques is the absence of suitable quantitative validation methods to evaluate the quality of motif representations inferred. Moreover, understanding the limitations associated with these approaches is essential to avoid misleading interpretations of results. Drawbacks of perturbation based methods, for example, include the interpretation of one sequence at a time, and the underestimation of the importance of input features due to the saturation problem. The perturbation based method used in this chapter also involves a more computationally feasible approach, which changes one nucleotide at a time, instead of a subset of input features, which although better is computationally inefficient. Albeit a global interpretability approach, the visualization of convolutional filters in the first layer may impede interpretability in the case of deep neural networks, as deeper networks typically learn redundant and partial motifs at each layer. However, networks with fewer layers, such as SigTFB, are able to learn full motif patterns from the first layers of network [124]. It is also important to note that in SigTFB, convolutional filter activation values are set before concatenating cell type specific or cell type general information. Thus, while visualizing convolutional filters for SigTFB does not help in investigating a TF's cell type specificity, it provides a general overview of what motifs the filters represent. For future work, other interpretation techniques, such as gradient based approaches, may also be used to further uncover the motif patterns resulting in cell type specificity of TFs. However, limitations of such approaches in terms of noisy gradients and saturation also need to be accounted for. Finally, it is important to note that our confirmatory motif enrichment analysis is limited by the current state of knowledge. Like ENCODE, JASPAR includes data on a relatively small fraction of all human TFs. Not all motifs are in JASPAR, and not having matches may not necessarily be because there are not any significant matches, but because there are not any matches in the JASPAR database for this motif, which emphasizes the importance of further effort in that domain. Other databases, such as CIS-BP [256], may also be explored for further analysis.

The reasoning behind the varying degrees of cell type specificity of various TF-AB models is difficult to directly infer. Thus, future work may further explain the rational behind such cell type specificity with a more in depth analysis of the different factors that cause cell type specificity in TF binding. Additionally, when analyzing the t-SNE plots for some TF-ABs, such as CTCF, we observe distinct subgroups of cell lines within clusters than may be more cell type specific than others. As such for future work, one might explore the cell type specificity in terms of a group of cell lines within a group, rather one cell type at a time. Also, when examining the relationship between motif enrichment and gene expression, we notice that some motifs may enriched but not expressed, and vice versa. Future work may involve investigating why such behavior is observed across all cell lines.

Our work further extends Chapter 4 by elucidating the biology behind a TF's cell

specificity, or lack that of, via the use of in silico mutagensis and various other motif enrichment and interpretability methods. Our goal is to uncover the motif patterns that drive a difference in DNA signatures for TFs across various cell types, and to derive overall patterns of enrichment per cell type or per TF.

5.7 Conclusion

Determining the motif patterns that drive the model to produce certain predictions is essential for unravelling the black box nature of deep learning models. Here, we demonstrated the relationship between differential motif patterns and high cell type specificity, and further identified the specific cell types and motifs per TF-AB that may cause the varying degrees of cell type specificity, or lack that of. We conducted a wide scale motif enrichment and interpretability analyses to uncover the patterns driving cell type specificity in the different TF-AB models. Our results lay the foundation for further exploration of the reasons behind a transcriptional regulator's cell type signature in terms of motif enrichment and gene expression.

Chapter 6

Conclusion

A key challenge in human genomics is deciphering the functionalities of the non-coding regions in the human genome. In this thesis, we discussed methods related to enhancing the overall efficiency and reliability of machine learning models applied for scientific innovation in regulatory genomics. It commences with an investigation on bias and noise in ChIP-seq experiments to the analysis on the heterogeneity of ChIP-seq and the DNA signatures of TFs. Ultimately, and ideally, the aim is to build models that accurately and effectively represent the biology behind molecular processes, as well as to tackle the noise and bias associated in the generation and processing of ChIP-seq data.

The first aspect of the thesis encompasses the analysis of bias and noise associated with ChIP-seq experiments. Many errors in peak calling, more specifically of type I error, arise from a combination of bias and noise. We develop a peak calling algorithm, WACS, which is an extension of a pre-existing commonly used peak calling method called MACS2. The goal of WACS is to alleviate the bias influence associated with each ChIP-seq experiment via the incorporation of a weighted customized selection of controls. WACS provides a smart control construction method, which proves to be beneficial for peak calling and further downstream genomic analysis, such as motif enrichment and reproduciblity. Additionally, WACS proves to be a more selective peak calling method compared to other peak callers.

Another dimension we investigate is the ability to reveal the cell type specificity of transcription factors from ChIP-seq data itself. This challenge is addressed two-folds from a computational perspective. We first use a deep learning approach to uncover the heterogeneity in the DNA sequences underlying ChIP-seq peaks. We conduct a wide scale analysis of TFs to identify and quantify the extent of cell type specificity of various TFs enriched in different cell types and tissues. We demonstrate the existence of cell type specificity in many TFs, and the absence of cell type specificity in other TFs. We also explore the consistency of cell type specificity when having different ABs for the same TF. Next, in the second part of our analysis, after demonstrating the cell type specificity of various TFs, we further elucidate the biology behind the existence or absence of cell type specificity in a TF. We conduct a wide scale motif enrichment analysis of all TF and AB combinations. Using several motif enrichment analysis techniques, we demonstrate the there is a consistency in cell type specific predictions, achieved earlier, and motif enrichment.

Overall, in this thesis, we tackle different issues for a better and more concrete understanding of transcription regulation.

6.1 Limitations and Future Work

The approaches implemented in this thesis have enhanced our knowledge in regulatory genomics. Nevertheless, there are some limitations, in terms of technology, methods and data, which are essential to acknowledge. In this section, we discuss limitations that arise in the analysis of ChIP-seq data and the use of deep learning in genomics.

6.1.1 Analysis of ChIP-seq

We will first focus on obstacles associated with ChIP-seq analysis. As previously discussed, ChIP-seq technology is a gold standard used for the detection of TF binding sites and histone modifications genome-wide. However, as with every high throughput sequencing technology, it is not without flaws. Systematic and procedural artifacts arise at each step of the multi-step ChIP-seq protocol, impeding the utilization of ChIP-seq to its fullest potential. One way to alleviate bias found in different ChIP-seq experiments is via the use of WACS, as discussed in Chapter 3. However, due to the multi-faceted nature of ChIP-seq, there may still be some biases and noise that have not been addressed. Thus, future directions may include a further investigation in the types of bias or noise present before and after the use of weighted controls. Similar to BIDCHIPS [200], a recent method implemented in our lab, the ChIP-seq signal could be perceived to have varying quantifiable, and thus measurable, sources of bias. Are there specific biases alleviated by the use of WACS? Which biases still exist regardless even after using WACS? A systematic comparison of multiple peak callers can then be conducted to quantify biases found by different methods for a ChIP-seq experiment. Moreover, more recently, the use of controls in peak calling was questioned [281]. This was based on the assumption that high correlation between control and treatment samples suggests that the control signal can be derived from the treatment sample, making the control sample redundant. Albeit an interesting observation, experiments were only conducted on simulated ChIP-seq data, not real genomic data, and the correlation between the control and treatment samples varies in real data. Future work may involve investigating the degree to which controls are required by different real ChIP-seq experiments across various TFs and cell types (or tissues).

Peaks are a mixture of true regions of enrichment and the background signal. Ideally, for more accurate output, WACS would fit the controls to the background signal of the ChIP-seq experiment. That is not the case, however, as the regression model in WACS does not know which portion of the signal is background. As such, some regression targets are incorrect as they represent the actual regions of enrichment. It can be thought of as a chicken and an egg problem – where a peak caller is first needed to identify the regions that are not peaks (background signal of the ChIP-seq), and a weighted peak caller (WACS) is then required to estimate weights per control to model the background signal of the

ChIP-seq experiment needed for peak detection. Thus, information about the peaks prior to weight control estimation is needed. This information is then used for peak detection again. For future work, an iterative process may be developed in an attempt to set up the regression to focus on the background signal of the ChIP-seq data. The approach may involve estimating the weights per control, finding the peak regions in the ChIP-seq data, and censoring those peak regions from the ChIP-seq signal. This can then be repeated multiple times until all peak regions from the ChIP-seq signal are removed. Finally, it is important to note that the influence of regression on peak regions will most probably be very low – as 99% of regions along the genome are not peaks and only a small fraction are considered to be actual peaks.

As with any other high throughput sequencing technology, the mapping of reads to a reference genome is an important step in determining the location of reads along the genome. In this thesis, the latest reference genome GRCh38/hg38 is used. Following mapping, the number of reads per region along the genome is calculated and regions of enrichment are identified. However, reads may originate from cancerous cell lines, such as K562, suggesting that there may be mutations in cancerous genomes not accounted for in the standard reference genome. Thus, the use of the same reference genome for mapping in all cell types and tissues may serve as a limitation to the methods implemented in this thesis, as it remains unclear what impact such mutations will have on read mapping and peak calling. While recent work has shown that the reference genome used indeed impacts peak calling and further downstream genomic analyses [87, 263], it is yet unclear how this would impact our WACS or SigTFB results. Future research may involve the exploration of the diversity of mutations present in ENCODE, via the construction of customized reference genomes for cancerous cell lines using reference guided de novo genome assembly, then the use of WACS or any other peak calling method to uncover any output variation.

Another shortcoming may arise in the actual peak calling step in the analysis of ChIP-seq. As mentioned in a recent paper published by our lab [52], many peak calling algorithms, MACS2 and WACS included, commit the statistical sin of using the same data twice, and thus may produce p-values that can not be trusted. As such, how peak callers define p-values is important, as incorrect values could produce misleading results. Additional research may involve the integration of both methods, RECAP [52] and WACS, to further explore peaks produced after p-value correction and smart bias removal. More specifically, future work could explore if is there a difference in peaks produced when accounting for bias using WACS whilst correcting for the p-values.

Regardless of how perfectly conducted the sample preparation and fragmentation steps are in the ChIP-seq workflow, the reproducibility and robustness of ChIP-seq data is highly dependent on the specificity and performance of ABs. This means that the binding preferences of a TF may vary depending on the AB used – due to varying specificities and off target-binding effects. Additionally, the polyclonal nature of ABs in ENCODE leads to irreproducibility, extensive batch variation and cross-reactivity [42, 212, 248]. This denotes that batches of the same AB may vary in quality. Thus, our understanding of ChIP-seq data may be hampered, or enhanced, depending on AB quality.

Due to the aforementioned limitations of the traditional ChIP-sequencing protocol,

other technologies, such as ChIP-exo [206] and Cleavage Under Targets and Tagmentation (CUT&Tag) [119], have been proposed. ChIP-exo, for example, achieves relatively lower noise and improves base pair resolution in comparison to ChIP-seq. Moreover, in CUT&Tag, the chromatin preparation and fragmentation methods of traditional ChIP-seq are eliminated, thus removing the need for a large amount of starting material and alleviating bias associated with these ChIP-seq steps. CUT&Tag also allows for single-cell genomic analysis, as it needs a relatively smaller number of cells to produce results. Additional research may include the investigation and comparison of bias and noise associated with these new methods. Will the incorporation of smart controls assist in peak calling as it does with ChIP-seq? Due to the novelty and recentness of such proposed methods, however, they are still not as widely used or as thoroughly investigated as ChIP-seq.

6.1.2 Deep Learning in Genomics

We discussed limitations that may arise with the use of ChIP-seq data, more specifically in terms of WACS, thus far. Since ChIP-seq data is used throughput this thesis, the same limitations may apply to the other approaches implemented. Next, we discuss current challenges encountered with the application of supervised learning in genomics research [128, 257], more specifically with the use of deep learning to uncover cell type specificity of various regulatory proteins.

Machine learning models and data representation methods are developed with assumptions that at times can be biologically inaccurate. Hence, input and output representations of data when using machine learning to model a specific problem formulation in genomics may impact model performance. For example, in regards to input data, there are many ways to define positive and negative instances for a specific TF. Due to the merging of ChIP-seq peaks across various cell types for a TF in Chapter 4, our notion of negative instances are those that are bound in one cell type, but not bound in others. Thus, all our negative instances are essentially bound in at least one cell type for a specific TF-AB. The notion of negative peaks differs in other problem formulations, where the genome is binned [165, 197, 198, 231, 285], or negative peaks are generated by the di-nucleotide shuffling of positive peaks [6]. In each scenario, there are factors that can bias our results. In the case of binning the genome, for example, some regions along the genome are considered "dark" regions, where reads can not be mapped to. Thus, if not removed from the input dataset, these regions will always be unbound across all cell types. Conversely, there are ENCODE blacklist regions where many reads typically map. They are believed to be due, at least in part, to errors in the reference genome, where multiple distinct genomic sites/sequences have been represented as one. In our problem formulation, ChIP-seq peaks from various cell types for a specific TF are merged together. The merged peaks are derived from different ChIP-seq experiments, which could vary in quality and originate from different labs. Although integrating multiple datasets gives models greater predictive power, the heterogeneity of the data may introduce biases. Thus, such biases are important to acknowledge when building your dataset. Another issue encountered in the representation of input data is the assumption that instances are independent and identically distributed (IID) [257]. In this thesis, our model is trained on ChIP-seq peaks generated by the ENCODE uniform

processing pipeline. The pipeline constitutes of many steps that lead to the generation of dependent data, such as the spatial smoothing of control signals, the computation of p-values and q-values, IDR analysis and fragment length estimation.

The performance of deep learning models is conditional on model design and hyperparameter choices. As such, an essential yet challenging step in deep learning is the selection of an optimal hyperparameter configuration and model architecture when training models. While Bayesian optimization is used to tune hyperparameters, some hyperparameters are explicitly initialized and not altered during the learning process. An example is input sequence length, which is set to 101bp in this thesis. Depending on model architecture, and type of data representation selected, whether its binning the genome or merging peaks, different sequence lengths can be selected. The length of input sequence may impact our learning experience, however. Longer input sequences may have more motif patterns, while shorter sequences may contain less features. While TF binding sites, or motifs, typically range from 6 to 20bp, the length of input sequences usually used in the prediction of TF binding sites ranges between 101 and 2000bp. Sequence lengths of up to 600bp were attempted in this thesis, yet similar performance was observed (data not shown). Ideally, sequence length should be tuned as well during optimization. Another data representation option that may impact performance is reverse-complement augmentation, or lack that of [284]. Due to complementary base pairing, it is unknown whether the regulatory protein is bound on the forward or reverse-complement strand of the double-stranded DNA. Thus, the ambiguity is typically managed by supplying both sequence strands as input. However, using a siamese architecture, we observed no difference in performance for our specific prediction problem when incorporating both orientations, and thus retreated to a standard architecture instead. The network architecture may also impact performance. We opted for the use of a simple architecture with a single CNN layer to support the motif enrichment analysis step. With this architecture, we avoid partial diffused patterns and can still identify co-occurring motifs. However, spatial information, more particularly the relationship, in terms of order and dependency, between the various motif patterns, is lost. Future work may include using a more complex architecture of CNN+LSTM or attention mechanisms to improve the detection of cell type specificity. One could also use exponential action functions, instead of ReLUs, and validate the claim of better sequence motif extraction [127].

Another common challenge faced by most large scale biological data is the imbalanced classification problem. In the prediction of TF binding sites, for example, the number of unbound regions along the genome overwhelmingly exceeds the number of protein bound regions. Some deep learning methods proposed for the prediction of TF binding sites do not explicitly address the class imbalance problem. The results produced by these methods could, thus, be misleading, as learning algorithms will be biased towards the majority class. Additionally, a substantial inflation of generalization performance is common when dealing with class imbalance in a multi-tasking setting; hence it may be important to explore performance per class per model. In the multitask formulation implemented in this thesis, we propose a balancing approach to not only balance the number of instances across cell types, but to also balance the number of positive and negative instances per cell line. Other class balancing techniques were attempted, yet were not as successful.

Moreover, although the class balancing method succeeded for most of the TF and AB combinations, it performed poorly in a few cases. In the inadequately performing models, we notice severe imbalance in the dataset with the minority class having less than 50 positive instances. Further research could explore models where class imbalance correction failed, and devise a more efficient approach to address the class imbalance problem.

Additionally, while our deep learning approach was successful on the majority of the TF and AB combinations, there are a few that failed. Poor performance may be due to lack of sufficient data or distributional differences in training and testing sets. A further investigation of the reasons behind such failures is important for a better understanding of our models. To navigate distributional differences, elaborate approaches, such as transfer learning and adversarial learning, could be used. However, lack of data may not be as easily tackled, as performing a ChIP-seq experiment is costly and time consuming. Deep learning models require sufficient data to learn, and will overfit on small datasets. A solution might be to first use deep learning for cross cell type prediction, and imputation on cell types with less data, then use our method to uncover cell type specificity.

Due to its black box nature and complexity, the inner workings of deep learning networks are difficult to understand [9, 169, 241]. Thus, interpretability techniques are used to understand and verify that deep learning models are learning biologically relevant representations, or motifs. The lack in understanding of the limitations associated with these approaches can produce a misleading sense of confidence in the output. A major issue is the lack of validation performance techniques to evaluate how well the interpretation of a model is, such that only a qualitative evaluation of the methods is conducted. In addition, most interpretation techniques, such as perturbation-based methods, only assess one example at a time. Thus, an understanding of the general behaviour of the model is needed. Moreover, when considering perturbation-based methods in this thesis, we used a feasible computational approach which perturbs one nucleotide at a time. However, ideally, a closer to optimal yet computationally inefficient evaluation, tests all subsets of input features. The importance of features can also be underestimated due to saturation problem. Although not used in this thesis, gradient based approaches can also be evaluated for the uncovering of cell type specificity for a specific TF-AB. However, they are not without limitations, as they are also impacted by the underestimation of features due to saturation, as well as noisy gradients.

SigTFB serves as a first step in the investigation of cell type specific DNA signatures. Factors, such as the intrinsic binding preference of a TF, alternative splicing, chromatin accessibility, co-binding and competitive binding, may impact a TF's DNA binding preference. For a comprehensive understanding of the mechanistic details behind the causes of cell type specificity, all different factors need to be accounted for – to ultimately decompose the DNA signature into factors that impact TF binding. A logical next step, for instance, may involve exploring whether chromatin accessibility causes the differential binding preference observed. Can we determine whether it is chromatin accessibility causing the cell type specific binding of the TF? Or is there another influencing factor, such as the DNA sequence preference change, making the TF cell type specific? To address whether or not the cell type specificity is mainly due to chromatin accessibility, we incorporated ATAC-seq data into our models (not shown here). We used ChIP-seq data of TF binding sites

with chromatin accessibility, as well as chromatin accessibility alone, to predict cell type specificity. Our results showed no significant improvement in performance when including chromatin accessible data – that strong cell type specificity in many TFs is not primarily governed by chromatin accessibility. However, previous work has used and shown the benefits of using chromatin accessible data to predict TF binding sites [120, 165, 197, 261, 285]. More work is required for a better understanding of the results obtained. Differences in results may be due to differences in problem, model, or even data formulation. For example, not all accessible sites are bound, not all inaccessible sites are unbound, specifically in the case of pioneer TFs, and an accessible region may contain more than one binding site. Moreover, the use of TF binding sites to predict accessibility may be limiting as most bound sites are typically accessible. However, not all accessible regions are bound, as such using accessibility to predict bound vs unbound regions of a TF is a reasonable approach. It is also important to note that a different data formulation method is used in approaches such as ChromDragoNN [165], where the genome is binned into intervals – resulting in some unbound and inaccessible bins across cell types. In SigTFB, all instances are bound in at least one cell type, increasing the likelihood of being accessible.

Alternative splicing (AS) is another factor that could influence a TF's DNA signature. Different versions of the same TF, TF isoforms, can be produced via AS. TF isoforms may have different protein structures, or even interact with different co-factors, and thus both could result in different DNA binding preferences of the isoform. Future directions may include a comprehensive investigation of AS from ChIP-seq and RNA-seq data to uncover and quantify the degree to which TF isoforms exist in ChIP-seq data. With the vast amount of ChIP-seq data available, can we detect whether TF isoforms exist from the TF's DNA binding preferences, which are the TF binding sites derived from ChIP-seq, using RNA-seq data? Can we then quantify the level at which each TF isoform exists in each cell type or tissue? In recent advances, we attempted to address these questions via the incorporation of transcript and exon level quantifications of various TFs from bulk RNA-seq data (from ENCODE) to our model – as different TF isoforms are made of different exons, and each isoform represents a transcript. Instead of concatenating the cell type encoding, as seen in SigTFB, we included exon and transcript level quantifications. However, our results were inconclusive (not shown here). Uncovering TF isoforms from short read bulk data is difficult. The recent emergence of third generation long read and single cell sequencing technologies suggests that alternative datasets, such as long-read single cell RNA-seq, may better assist in the analysis of TF isoforms. Long reads are able to better capture the full length isoforms, with the splice sites and transcription start and end sites, while single cell data can provide better insights at higher resolution of individual cells or tissues than conventional bulk sequencing. Moreover, it would be interesting to see how much information can be derived from short read data compared to long read data – more particularly how well can we predict isoforms from shorter read RNA-seq data if we use long read RNA-seq data as the ground truth. Similar work comparing short and long read RNA-seq can be found here [50, 102].

The behaviour of a TF in a specific cell type, tissue or condition, is impacted by the many interactions the TF participates in. The primary motif of an assayed TF does not typically explain numerous TF binding events. The sole focus on one interaction (factor)

over others may lead to information loss. Other causes of differential cell type specific DNA signatures of TFs, such as co-operative and competitive binding, should also be considered. In recent work, BindVAE [130], an unsupervised learning approach based on a variational autoencoder, decomposes input DNA sequences obtained from ATAC-seq or HT-SELEX to derive latent cell type specific features of TFs, such as those related to cooperative binding and the genomic regions surrounding the binding sites. This method can be extended to include different type of data, such as ChIP-seq, RNA-seq and ATAC-seq, to decompose the ChIP-seq signal further to obtain the impact of the discussed influencing factors on the cell type specific DNA signatures of TFs. After the identification and characterization of each influencing factor, future work may include a large scale analysis of how the different combinations of factors impact DNA signatures of various TFs in various cell types. In this study, we would assess the extent at which each factor impacts the DNA binding preference of the TF – which factors are more involved in the TF's cell type specific state. Are there some influencing factors that effect a group of TFs more than others? Can we detect and quantify the impact of each influencing factor to model the DNA signature of each TF in each cell type? Can we develop a hierarchy of TFs and cell types to determine which are more influenced than others? If, for example, a TF is known to be cell type specific, is there a specific influencing factor more involved in determining its DNA signature?

For a more cell type specific characterization of TFs, other cis-regulatory modules, such as enhancers, silencers and insulators, may also be accounted for. Enhancers, for instance, are known to drive cell type specific gene expression. They are short regions along the genome (≈500bp) bound to by multiple TFs, known as activator proteins, to increase the probability of a specific distant gene to be transcribed [10, 187, 201]. The activation of enhancers loops the chromatin in such a way that the enhancers become in proximity with the promoters they modulate. Future work may include uncovering the promoter-enhancer interactions involved in activating certain cell type specific TFs, and identifying which of these TFs have enhancer activating roles. The promoter-enhancer interactions typically fold the chromatin – thus including the 3D features of the genome in our analysis will be an important step in the study of TF cell type specific DNA signatures. To uncover the enhancer regions at high resolution, datasets such as nascent RNA sequencing data [57, 260] or micrococcal nuclease digestion with deep sequencing (MNase-seq) [95] may be be used. Nascent RNA sequencing, for example, detects and quantifies known genes and active enhancers simultaneously, assaying all active TFs in a cell or tissue in one experiment, instead of one TF at a time with ChIP-seq. In more recent work, transcriptional regulatory networks with both enhancer and genes, instead of gene regulatory networks of only co-regulated gene, were constructed using nascent RNA-seq data [215]. For a more cell type specific visualization of each TF, future work may include the generation of transcriptional regulatory networks for each TF and cell type. Moreover, it would be interesting to further investigate the different types of genes, be it housekeeping or more cell type specific genes, involved with the varying degrees of cell type specific TFs by exploring the Gene Ontology (GO) consortium [1, 12], for example.

We propose some strategies in an attempt to explain the phenomenon of cell type specific DNA signatures of TFs. SigTFB, as well as the other proposed approaches, could also be applied to other types of data, such as epigenetic data on histone modifications to

uncover which cell type specific motifs drive epigenetic differences, or chromatin accessible data to identify which cell type specific patterns are more accessible. Furthermore, instead of only focusing on the overall AUC differences across entire experiments for a specific TF-AB as in SigTFB, future work may include having site specific scores of cell type specificity, where each genomic site is labeled with a cell type general or cell type specific score. Smart peak calling and the exploration of cell type specific DNA signatures can also be combined, where WACS is applied to ChIP-seq data for more efficient peak calling and SigTFB is then evaluated on the called peaks. Is there a difference in the SigTFB output when using WACS versus MACS2? More specifically, does the peak caller used impact the TFs derived as cell type specific. Moreover, in this thesis, we acknowledge the tentative impact of the antibody used on output. As such, further work may include predicting the AB sequence for each TF in ENCODE using tools such as AbLang [172], for example, and then validating the output sequence with the actual AB used. Finally, with the emergence of the vast amount of data types in attempt to decipher the genomic landscape, one might wonder about how much of the data is actually required to uncover different aspects of the genome?

Appendix A

WACS: Appendix

Appendix A provides the supplementary information for Chapter 3 on WACS. This includes the list of ChIP-seq experiments for the K562 (Tables A.1 and A.2), A549 (Table A.4), GM12878 (Table A.5) and HepG2 (Table A.6) cell types. It also includes details on TFs (Table A.3), lab (Table A.7), year (Table A.8) and mapped read length (Table A.9) for the various ChIP-seq experiments used.

Table A.1: Table for the 45 ChIP-seq experiments and their corresponding ChIP-seq replicate samples and TFs for the K562 cell line from the ENCODE database used in our analysis.

Experiment	Replicates	TF
ENCSR000BKQ	ENCFF401KIO, ENCFF398SXO	ETS1
ENCSR000BKT	ENCFF794ABP, ENCFF587WWS	USF1
ENCSR000BKU	ENCFF044TAL, ENCFF651HPM	YY1
ENCSR000BLI	ENCFF823GCX, ENCFF827SLL	E2F6
ENCSR000BMD	ENCFF331TRC, ENCFF535PLW	ELF1
ENCSR000BME	ENCFF784SCN, ENCFF778PDI	ZBTB7A
ENCSR000BMV	ENCFF492NUF, ENCFF561ILM	FOSL1
ENCSR000BMW	ENCFF204TUQ, ENCFF546IMN	REST
ENCSR000BNK	ENCFF454MMY, ENCFF310MOR	CTCFL
ENCSR000BRQ	ENCFF154YTN, ENCFF314OQP	CEBPB
ENCSR000DWE	ENCFF081HVQ, ENCFF494VZW	CTCF
ENCSR000EFS	ENCFF924CYX, ENCFF014UUB	JUN
ENCSR000EFV	ENCFF635HIS, ENCFF360QBV	MAX
ENCSR000EGI	ENCFF589IXE, ENCFF097GYE	MAFF
ENCSR000EGJ	ENCFF641HZR, ENCFF006GQZ	MYC
ENCSR000EGK	ENCFF168KTM, ENCFF299HHL	IRF1
ENCSR000EGN	ENCFF400BSN, ENCFF321ZQU	JUND
ENCSR000EGS	ENCFF239WGU, ENCFF836IMK	MYC
ENCSR000EGT	ENCFF489YJG, ENCFF728WOA	IRF1

ENCSR000EGZ	ENCFF726UVN, ENCFF388QNA	MXI1
ENCSR000EHE	ENCFF953ZZL, ENCFF729WGC	CEBPB
ENCSR000EWH	ENCFF771NSF, ENCFF078BWN	NR2C2
ENCSR000EWJ	ENCFF201BQU, ENCFF011OSI	E2F6
ENCSR000EWL	ENCFF613CDR, ENCFF156NIH	E2F4
ENCSR000EWN	ENCFF891GFG, ENCFF094FZQ	ZNF263
ENCSR000EZT	ENCFF749RRI, ENCFF784MLU	JUN
ENCSR000EZU	ENCFF014DRH, ENCFF553WMU	MYC
ENCSR000EZV	ENCFF357MHM, ENCFF491EUN	MYC
ENCSR000EZW	ENCFF814CHV, ENCFF050LIC	JUN
ENCSR000FAU	ENCFF089XMI, ENCFF263PLG	STAT1
ENCSR000FAV	ENCFF463LMJ, ENCFF403WNR	STAT1
ENCSR000FAZ	ENCFF439NNR, ENCFF483RLD	MYC
ENCSR041AXL	ENCFF871KNP, ENCFF490ICL	RFX1
ENCSR099NCH	ENCFF205VDU, ENCFF109OWW	ZNF24
ENCSR486IFJ	ENCFF857YYV, ENCFF320WXN	ESRRA
ENCSR494TDU	ENCFF337GMY, ENCFF821HMU	NRF1
ENCSR508DQA	ENCFF563FZR, ENCFF373AFZ	FOXK2
ENCSR563LLO	ENCFF846CYU, ENCFF226KOW	E2F1
ENCSR588AKU	ENCFF011XRF, ENCFF812KIP	RUNX1
ENCSR795IYP	ENCFF512YJL, ENCFF090SFR	JUNB
ENCSR819LHG	ENCFF364SJK, ENCFF611PXX	FOXA1
ENCSR837EYC	ENCFF564KCD, ENCFF207NLX	NRF1
ENCSR876GXA	ENCFF632WSK, ENCFF226IAA	ZBTB33
ENCSR968GIB	ENCFF837ZOY, ENCFF183MSQ	RFX1
ENCSR998AJK	ENCFF722LJA,ENCFF564CXM	NRF1

Table A.2: Table for the 90 ChIP-seq samples and their corresponding control samples for the K562 cell line from the ENCODE database used in our analysis.

ChIP-seq	Controls
ENCFF310MOR	ENCFF772PJM, ENCFF982BHL
ENCFF263PLG	ENCFF332CUX
ENCFF109OWW	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF953ZZL	ENCFF023NGN
ENCFF439NNR	ENCFF767FSP
ENCFF589IXE	ENCFF023NGN
ENCFF183MSQ	ENCFF712WXB, ENCFF790TAN
ENCFF491EUN	ENCFF332CUX
ENCFF094FZQ	ENCFF355SGP
ENCFF827SLL	ENCFF812TGW, ENCFF234NVU, ENCFF913HVS, ENCFF382XSA
ENCFF078BWN	ENCFF355SGP
ENCFF611PXX	ENCFF796JTX, ENCFF720AUK
ENCFF641HZR	ENCFF023NGN
ENCFF226KOW	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF097GYE	ENCFF023NGN
ENCFF239WGU	ENCFF942FFX
ENCFF492NUF	ENCFF772PJM, ENCFF982BHL
ENCFF564KCD	ENCFF709XAA, ENCFF332SVJ
ENCFF201BQU	ENCFF355SGP
ENCFF044TAL	ENCFF772PJM, ENCFF982BHL
ENCFF836IMK	ENCFF942FFX
ENCFF635HIS	ENCFF023NGN
ENCFF398SXO	ENCFF772PJM, ENCFF982BHL
ENCFF204TUQ	ENCFF772PJM, ENCFF982BHL
ENCFF561ILM	ENCFF772PJM, ENCFF982BHL
ENCFF651HPM	ENCFF772PJM, ENCFF982BHL
ENCFF331TRC	ENCFF812TGW, ENCFF234NVU, ENCFF913HVS, ENCFF382XSA
ENCFF050LIC	ENCFF482LDC
ENCFF360QBV	ENCFF023NGN
ENCFF546IMN	ENCFF772PJM, ENCFF982BHL
ENCFF553WMU	ENCFF482LDC
ENCFF821HMU	ENCFF227IZS, ENCFF910IKB
ENCFF400BSN	ENCFF023NGN
ENCFF403WNR	ENCFF767FSP
ENCFF512YJL	ENCFF162ZOO, ENCFF332SVJ
ENCFF337GMY	ENCFF227IZS, ENCFF910IKB
ENCFF778PDI	ENCFF772PJM, ENCFF982BHL
ENCFF784SCN	ENCFF772PJM, ENCFF982BHL
ENCFF299HHL	ENCFF942FFX

ENCFF729WGC	ENCFF023NGN
ENCFF564CXM	ENCFF227IZS, ENCFF910IKB
ENCFF784MLU	ENCFF942FFX
ENCFF846CYU	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF156NIH	ENCFF355SGP
ENCFF494VZW	ENCFF873PSH
ENCFF205VDU	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF871KNP	ENCFF227IZS, ENCFF910IKB
ENCFF321ZQU	ENCFF023NGN
ENCFF728WOA	ENCFF482LDC
ENCFF154YTN	ENCFF304AZH, ENCFF984QXA, ENCFF533FQH, ENCFF836OEO
ENCFF089XMI	ENCFF332CUX
ENCFF454MMY	ENCFF772PJM, ENCFF982BHL
ENCFF011OSI	ENCFF355SGP
ENCFF794ABP	ENCFF772PJM, ENCFF982BHL
ENCFF613CDR	ENCFF355SGP
ENCFF081HVQ	ENCFF873PSH
ENCFF168KTM	ENCFF942FFX
ENCFF837ZOY	ENCFF712WXB, ENCFF790TAN
ENCFF483RLD	ENCFF767FSP
ENCFF891GFG	ENCFF355SGP
ENCFF226IAA	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF090SFR	ENCFF162ZOO, ENCFF332SVJ
ENCFF006GQZ	ENCFF023NGN
ENCFF563FZR	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF857YYV	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF722LJA	ENCFF227IZS, ENCFF910IKB
ENCFF924CYX	ENCFF023NGN
ENCFF823GCX	ENCFF812TGW, ENCFF234NVU, ENCFF913HVS, ENCFF382XSA
ENCFF490ICL	ENCFF227IZS, ENCFF910IKB
ENCFF011XRF	ENCFF285EWB, ENCFF696ZGZ, ENCFF709XAA
ENCFF314OQP	ENCFF304AZH, ENCFF984QXA, ENCFF533FQH, ENCFF836OEO
ENCFF489YJG	ENCFF482LDC
ENCFF726UVN	ENCFF023NGN
ENCFF771NSF	ENCFF355SGP
ENCFF535PLW	ENCFF812TGW, ENCFF234NVU, ENCFF913HVS, ENCFF382XSA
ENCFF207NLX	ENCFF709XAA, ENCFF332SVJ
ENCFF814CHV	ENCFF482LDC
ENCFF373AFZ	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF632WSK	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF587WWS	ENCFF772PJM, ENCFF982BHL
ENCFF388QNA	ENCFF023NGN

ENCFF357MHM	ENCFF332CUX
ENCFF401KIO	ENCFF772PJM, ENCFF982BHL
ENCFF014DRH	ENCFF482LDC
ENCFF364SJK	ENCFF796JTX, ENCFF720AUK
ENCFF320WXN	ENCFF895QZG, ENCFF227IZS, ENCFF910IKB, ENCFF937WDE
ENCFF749RRI	ENCFF942FFX
ENCFF014UUB	ENCFF023NGN
ENCFF463LMJ	ENCFF767FSP
ENCFF812KIP	ENCFF285EWB, ENCFF696ZGZ, ENCFF709XAA

Table A.3: Table for the transcription factors (TFs) and their corresponding motif ID from JASPAR for 45 ChIP-seq experiments.

TF	ID
MXI1	MA1108.1
E2F4	MA0470.1
MAFF	MA0495.1
CEBPB	MA0466.1
JUNB	MA0490.1
CTCFL	MA1102.1
YY1	MA0095.2
USF1	MA0093.2
REST	MA0138.2
ESRRA	MA0592.1
ELF1	MA0473.1
STAT1	MA0137.2
ZNF24	MA1124.1
ZBTB7A	MA0750.2
IRF1	MA0050.2
JUN	MA0488.1
CTCF	MA0139.1
NR2C2	MA0504.1
NRF1	MA0506.1
FOXA1	MA0148.1
FOXK2	MA1103.1
JUND	MA0491.1
E2F1	MA0024.2
E2F6	MA0471.1
ZNF263	MA0528.1
ZBTB33	MA0527.1
RUNX1	MA0002.2
MAX	MA0058.2
FOSL1	MA0477.1

Table A.4: Table for the ChIP-seq experiments and their corresponding ChIP-seq replicate samples, TFs and controls for the A549 cell line from the ENCODE database used in our analysis.

Experiment	Replicates	TF	Controls
ENCSR182OZC	ENCFF791DRP ENCFF217RBI	CEBPB	ENCFF634ULC ENCFF632UPH ENCFF368OTV
ENCSR375BUB	ENCFF073MBT ENCFF280ZFT	CEBPB	ENCFF214UMU ENCFF773DUX ENCFF408NFU
ENCSR606ZTC	ENCFF347MNU ENCFF417GPF	CEBPB	ENCFF455UAB ENCFF887YTT ENCFF081TBO
ENCSR623KNM	ENCFF757GXN ENCFF826GGN	ELK1	ENCFF949XNJ ENCFF918AJW
ENCSR000BQO	ENCFF125MJO ENCFF585INN	FOSL2	ENCFF656HEF
ENCSR419TWL	ENCFF504YVD ENCFF595EIS	HES2	ENCFF634ULC ENCFF632UPH ENCFF368OTV
ENCSR991VVW	ENCFF476XBN ENCFF761UEZ	JUN	ENCFF193ABY ENCFF222ACA ENCFF639UDD
ENCSR269RPR	ENCFF179XAQ ENCFF330XFU	JUNB	ENCFF171YYX ENCFF631DES ENCFF298EPS
ENCSR431LRW	ENCFF599JTK ENCFF389OFH	JUNB	ENCFF653HKQ ENCFF097CSC ENCFF987XCE
ENCSR892DRK	ENCFF364NWO ENCFF808ADX	REST	ENCFF572IKT ENCFF714AMB

Table A.5: Table for the ChIP-seq experiments and their corresponding ChIP-seq replicate samples, TFs and controls for the GM12878 cell line from the ENCODE database used in our analysis.

Experiment	Replicates	TF	Controls
ENCSR841NDX	ENCFF028TNY ENCFF444PPF	ELF1	ENCFF666ATR ENCFF322NTO
ENCSR000BMB	ENCFF739SRY ENCFF735DGJ	ELF1	ENCFF488YYE
ENCSR000DZB	ENCFF211VKF ENCFF784XUE	ELK1	ENCFF477ZKJ ENCFF450WED ENCFF824NQO ENCFF710SMS ENCFF579QDW ENCFF813LMQ
ENCSR000BGY	ENCFF240MQI ENCFF888PAI	IRF4	ENCFF562HPN ENCFF100EIH ENCFF438FFV
ENCSR000BQL	ENCFF983YCI ENCFF207QTV	NFATC1	ENCFF754WTG ENCFF966AVZ ENCFF537DAJ
ENCSR000BGR	ENCFF791EPM ENCFF845MYC	PBX3	ENCFF562HPN ENCFF100EIH ENCFF438FFV
ENCSR000BQS	ENCFF894EID ENCFF569QEN	REST	ENCFF430ZCF ENCFF100EIH ENCFF438FFV ENCFF562HPN
ENCSR000BRI	ENCFF884LEJ ENCFF579PRC	RUNX3	ENCFF754WTG ENCFF966AVZ ENCFF537DAJ
ENCSR000BGE	ENCFF263NOT ENCFF731ZNW	SRF	ENCFF862QZT ENCFF289ONG
ENCSR000BGI	ENCFF737VAT ENCFF074OYP	USF	ENCFF862QZT ENCFF289ONG

Table A.6: Table for the ChIP-seq experiments and their corresponding ChIP-seq replicate samples, TFs and controls for the HepG2 cell line from the ENCODE database used in our analysis.

Experiment	Replicates	TF	Controls
ENCSR000BQI	ENCFF090KCF ENCFF499BWX	CEBPB	ENCFF175NMQ ENCFF285LVE
ENCSR000EEE	ENCFF677PSB ENCFF514FUP	CEBPB	ENCFF165KZY
ENCSR000BIE	ENCFF435DKZ ENCFF178SXE	CTCF	ENCFF175NMQ ENCFF285LVE
ENCSR112ALD	ENCFF011HOS ENCFF320SCI	CREB1	ENCFF950AXC ENCFF190EPQ
ENCSR267DFA	ENCFF396NXZ ENCFF988UCQ	FOXA1	ENCFF950AXC ENCFF190EPQ
ENCSR000BMO	ENCFF332SRJ ENCFF401YVR	FOXA1	ENCFF175NMQ ENCFF193YIO ENCFF285LVE ENCFF943DZB
ENCSR000BHP	ENCFF953PCA ENCFF185DVY	FOSL2	ENCFF943DZB ENCFF193YIO
ENCSR000BMZ	ENCFF930EXY ENCFF418EVV	ELF1	ENCFF175NMQ ENCFF285LVE
ENCSR000BJL	ENCFF195HUS ENCFF291BBG	REST	ENCFF249PQD ENCFF741TQN
ENCSR000EEK	ENCFF074GYD ENCFF122BQB	JUN	ENCFF165KZY

Table A.7: Lab

Features	Controls Used	Controls Not Used	Row Total
Same Lab	376	2208	2584
Not Same Lab	1709	8397	10106
Column Total	2085	10605	12690

Table A.8: Year

Features	Controls Used	Controls Not Used	Row Total
Same Year	617	1945	2562
Not Same Year	1468	8660	10128
Column Total	2085	10605	12690

Table A.9: Mapped Read Length

Features	Controls Used	Controls Not Used	Row Total
Same Length	375	1541	1916
Not Same Length	1710	9064	10774
Column Total	2085	10605	12690

Appendix B

SigTFB: Appendix

Appendix B provides the supplementary information for Chapter 4 on SigTFB. This includes the list of TF-AB combinations used in the chapter, as well as respective details on ENCODE accession code, corresponding cell types and the AUC difference obtained (Table B.1). The cell type names for the corresponding ENCODE cell type code can be found in Table B.2. Plots in Figures B.1, B.2, B.3, B.4, B.5, B.6, B.7, B.8, B.9 and B.10 display the AUC differences obtained from evaluation on test dataset per cell type per TF-AB combination.

Table B.1: TF-AB combinations and their corresponding AUC differences (score), cell types (CL ID) and actual ENCODE experiment used (ENCODE Experiments). Refer to SI Table 2 for the cell type names corresponding to the "CL ID".

TF	AB	Score	CL ID	ENCODE Experiments
AGO1	ENCAB189GDE	0.06	EFO:0002067 EFO:0001187	ENCFF627BHP ENCFF100VYA
ARNT	ENCAB608UKB	0.06	EFO:0002067 EFO:0002784	ENCFF758RQJ ENCFF655EFA
ASH2L	ENCAB947YBH	0.01	EFO:0001187 EFO:0002784	ENCFF096XRG ENCFF638IUM
ATF1	ENCAB697XQW	0.2	EFO:0002067 EFO:0001187	ENCFF914IWA ENCFF715PLB
ATF2	ENCAB605GOL	0.14	EFO:0002067 EFO:0001187 EFO:0002784	ENCFF210HTZ ENCFF803FHN ENCFF089BQU

ATF3	ENCAB000ADZ	0.19	EFO:0001086 EFO:0002067 EFO:0001187 UBERON:0002107 EFO:0003042	ENCFF487GLV ENCFF137OEY ENCFF851UTY ENCFF467WOR ENCFF782SGI ENCFF146URA
ATF7	ENCAB000BMO	0.21	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF760ZVI ENCFF371SJR ENCFF495PWL ENCFF498YGH
BACH1	ENCAB000AEA	0.17	EFO:0002067 EFO:0002784 EFO:0003042	ENCFF725YZH ENCFF543FNN ENCFF851YHG
BCL3	ENCAB000AEG	0.18	EFO:0001086 EFO:0002784	ENCFF093ZAB ENCFF247MHT
BCLAF1	ENCAB012FUX	0.23	EFO:0002067 EFO:0002784	ENCFF381ZDU ENCFF054DTJ
BHLHE40	ENCAB000AEK	0.08	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001196	ENCFF567GON ENCFF863ATX ENCFF477JTV ENCFF370ZNL ENCFF622HGF
BMI1	ENCAB000BDV	0.14	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF352DRR ENCFF414LXZ ENCFF592LPO
BRD4	ENCAB782ZNQ	0.01	EFO:0002067 EFO:0001187	ENCFF806CQB ENCFF736GHL
CBX5	ENCAB150NID	0.0	EFO:0002067 EFO:0002784	ENCFF403TAE ENCFF417SVR
CBX8	ENCAB000AEX	0.08	EFO:0001086 EFO:0002067	ENCFF330OCU ENCFF210GJE
CEBPB	ENCAB000AFB	0.11	EFO:0001086 EFO:0002067 EFO:0001187 EFO:0001196 EFO:0002784	ENCFF786YYI ENCFF047UIF ENCFF757KYL ENCFF862DXR ENCFF915ZYE ENCFF813LOW ENCFF321KQD
CEBPG	ENCAB728YTO	0.11	EFO:0002067 EFO:0001203	ENCFF930PBH ENCFF086CSF ENCFF797MRW

CHD1	ENCAB000AFE	0.08	EFO:0002784 EFO:0003042	ENCFF863CTN ENCFF806HXY ENCFF549ODQ
CHD1	ENCAB000AFF	0.08	EFO:0001196 EFO:0001203	ENCFF510QXG ENCFF730UAD
CHD2	ENCAB000AFG	0.1	EFO:0001086 EFO:0001187 EFO:0003072 EFO:0002784 EFO:0003042	ENCFF310IDS ENCFF546AYN ENCFF181XMM ENCFF245UXM ENCFF068MEO
CHD4	ENCAB276UJU	0.07	EFO:0001086 EFO:0001187	ENCFF766YPH ENCFF148ABR
CREB1	ENCAB279BHV	0.06	EFO:0001187 EFO:0001203	ENCFF495PCJ ENCFF550TXR
CREM	ENCAB000AAT	0.17	EFO:0002067 EFO:0001187 EFO:0002784	ENCFF091YID ENCFF290UGF ENCFF021XJN
CTBP1	ENCAB000BAY	0.23	EFO:0002067 EFO:0001203	ENCFF456MGR ENCFF349UTF
CTCF	ENCAB000AFQ	0.01	EFO:0001086 EFO:0002067 EFO:0003072 EFO:0001196 EFO:0002784	ENCFF396BZQ ENCFF307XFM ENCFF960ZGP ENCFF540DWT ENCFF646TUX
CTCF	ENCAB000AFR	0.04	EFO:0002824 EFO:0001203 EFO:0002067 EFO:0001187 EFO:0003042	ENCFF549PGC ENCFF543WTP ENCFF821AQO ENCFF119XFJ ENCFF942TCG
CTCF	ENCAB000AXU	0.01	UBERON:0002107 CL:0002319 EFO:0003072	ENCFF049UCF ENCFF143HEE ENCFF372JOV

CTCF	ENCAB000AXX	0.01	CL:0000103 NTR:0000711 CL:0000192 CL:0002551 CL:0002372 EFO:0002824 CL:0000062 EFO:0003072 CL:0000182 EFO:0006711 EFO:0002074 EFO:0007950 CL:0000127 EFO:0001203	ENCFF203ZIS ENCFF798RFA ENCFF719TNH ENCFF560GGY ENCFF232FXZ ENCFF186NOM ENCFF476DVJ ENCFF685KTA ENCFF141MTA ENCFF518MQA ENCFF744PXO ENCFF148BSH ENCFF960XTR ENCFF846FYU
CTCF	ENCAB000AXY	0.02	EFO:0001086 CL:0000236 EFO:0002067 CL:0001054 EFO:0001187 CL:0000312 UBERON:0002106 UBERON:0001264 EFO:0002791 UBERON:0002048 CL:0002327 CL:0002553 EFO:0002784 EFO:0001203	ENCFF505MGI ENCFF068YLN ENCFF139RCX ENCFF502CZS ENCFF861NDU ENCFF954FAQ ENCFF028IIR ENCFF356LIU ENCFF612ZUY ENCFF519CXF ENCFF300XXC ENCFF910TER ENCFF785NTC ENCFF535MZG ENCFF628EUU ENCFF685HMV ENCFF615GTV
CTCF	ENCAB247TYO	0.11	CL:0000103 EFO:0007950	ENCFF322WKG ENCFF904CNB
DDX20	ENCAB398QAO	0.16	EFO:0002067 EFO:0001203	ENCFF536LKB ENCFF089GNH
DPF2	ENCAB076EMA	0.24	EFO:0002067 EFO:0001203	ENCFF217ZTP ENCFF042AWM

DPF2	ENCAB384ZPY	0.25	EFO:0002067 EFO:0002784	ENCFF537VKZ ENCFF771IAW
E2F4	ENCAB000AFV	0.2	EFO:0002067 EFO:0002784	ENCFF687SFB ENCFF225TLP
E2F8	ENCAB224FFQ	0.1	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF171WWF ENCFF412GFI ENCFF072VGV
E4F1	ENCAB944PGR	0.23	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF347USC ENCFF752KNU ENCFF035GFS
EGR1	ENCAB000ASX	0.0	UBERON:0002107 EFO:0002067 EFO:0003042	ENCFF477ANT ENCFF561OGS ENCFF617JQS ENCFF808WST
EHMT2	ENCAB282XQE	0.13	EFO:0001086 EFO:0002067 EFO:0001187	ENCFF413RQL ENCFF682XPD ENCFF199OOU
ELF1	ENCAB000AGA	0.1	EFO:0001086 EFO:0002067 EFO:0001187	ENCFF935ZUW ENCFF463GCH ENCFF840RWO
ELF1	ENCAB778OCV	0.08	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF617ZLL ENCFF020UCD ENCFF948CPI
ELK1	ENCAB000AGB	0.02	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF119SCQ ENCFF408TWV ENCFF432AQP
EP300	ENCAB000AJM	0.08	EFO:0002067 CL:0002319 EFO:0001187 EFO:0002791 EFO:0002784 EFO:0003042	ENCFF755HCK ENCFF924KFU ENCFF459ARL ENCFF674QCU ENCFF834UVX ENCFF865UDD ENCFF510FUM
EP300	ENCAB000AJO	0.3	EFO:0001187 EFO:0002784	ENCFF080HJX ENCFF806JJS
ESRRA	ENCAB000AGE	0.15	EFO:0001086 EFO:0002067 EFO:0002784 EFO:0001203	ENCFF722LJP ENCFF592GWM ENCFF541DRZ ENCFF558UWY

ETS1	ENCAB000AGG	0.06	EFO:0007950 EFO:0001086 EFO:0002067 EFO:0002784	ENCFF896WFR ENCFF461PRP ENCFF511AZU ENCFF980VOD
ETV6	ENCAB000ABD	-0.0	EFO:0002067 EFO:0002784	ENCFF426GSY ENCFF745ANU
ETV6	ENCAB997CJG	0.37	EFO:0002067 EFO:0002784	ENCFF658SGJ ENCFF116AMK
EZH2	ENCAB000AGH	0.12	CL:0002551 EFO:0001187 CL:0000515 EFO:0002791 CL:0002327 CL:0002553 EFO:0002784 CL:0000236	ENCFF340LPI ENCFF260KLJ ENCFF504QZJ ENCFF279MNV ENCFF420KMT ENCFF615NYO ENCFF128OWK ENCFF434OEY
EZH2	ENCAB913HCF	0.03	CL:0000182 EFO:0005723	ENCFF976SAN ENCFF324UNA
EZH2 phospho T487	ENCAB000BKV	0.07	NTR:0000711 EFO:0002074 CL:0000103 EFO:0005723	ENCFF314ZKR ENCFF320REA ENCFF689GPW ENCFF687VHB
FIP1L1	ENCAB511DBP	0.12	EFO:0002067 EFO:0001187	ENCFF084DTV ENCFF031LBW
FOS	ENCAB000AEQ	0.13	CL:0002618 EFO:0001196 EFO:0001203	ENCFF217ZMF ENCFF327GZX ENCFF170POB
FOSL2	ENCAB000AGL	0.16	EFO:0001086 EFO:0001187	ENCFF808RWZ ENCFF054ESU
FOXA1	ENCAB000AGM	0.16	EFO:0001086 EFO:0001187	ENCFF152BOT ENCFF297HAX ENCFF167BKY
FOXA1	ENCAB000AGN	0.2	EFO:0001187 UBERON:0002107	ENCFF872MGU ENCFF324QGE ENCFF951VPZ
FOXA1	ENCAB301QLE	0.1	EFO:0001187 EFO:0001203	ENCFF160RLI ENCFF367TQC
FOXA2	ENCAB000AGO	0.11	EFO:0001187 UBERON:0002107	ENCFF184NAC ENCFF168JLI ENCFF293LRQ

FOXK2	ENCAB625ERS	0.17	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF899MQW ENCFF990MTR ENCFF490EQR ENCFF315CHX
FUS	ENCAB598VSJ	0.03	EFO:0002067 EFO:0001187	ENCFF216YZI ENCFF688ARM
GABPA	ENCAB000AGR	0.06	EFO:0001086 EFO:0002067 EFO:0001187 UBERON:0002107 EFO:0002791 EFO:0002784 EFO:0003042	ENCFF520GJC ENCFF124HAC ENCFF225GFQ ENCFF054HJA ENCFF091UDB ENCFF946ACA ENCFF280YAF ENCFF344XWK
GABPA	ENCAB728YTO	0.05	EFO:0002067 EFO:0001203	ENCFF620TOF ENCFF535BMV
GATA2	ENCAB000AGT	0.3	EFO:0002067 CL:0002618	ENCFF173TXA ENCFF987YIJ
GATAD2B	ENCAB939ONI	0.17	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF046BRP ENCFF298AIX ENCFF569CMJ
GTF2F1	ENCAB000AHE	0.11	EFO:0002067 EFO:0001187 EFO:0002791 EFO:0003042 EFO:0001203	ENCFF876GXQ ENCFF394JHN ENCFF493PRB ENCFF478HYJ ENCFF343QQE
GTF2F1	ENCAB108WPN	0.33	EFO:0002067 EFO:0001187	ENCFF843UHP ENCFF599TWF
HCFC1	ENCAB000AHG	0.02	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF722QBB ENCFF167RXK ENCFF401IAI ENCFF485SRU
HDAC1	ENCAB649LYF	0.16	EFO:0002067 EFO:0001187	ENCFF069KPS ENCFF557WXK
HDAC2	ENCAB000AHI	0.1	EFO:0001086 EFO:0002067 EFO:0003042	ENCFF363GSV ENCFF497YNJ ENCFF814DAF
HDAC2	ENCAB000AHJ	0.15	EFO:0002067 EFO:0001187 EFO:0003042	ENCFF589GSN ENCFF009IVJ ENCFF741IMY

HDGF	ENCAB173GGB	0.17	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF442WRJ ENCFF161SFU ENCFF575WFB
HES1	ENCAB724OQW	0.31	EFO:0002067 EFO:0001203	ENCFF010OOE ENCFF144OPN
HNF4A	ENCAB000AHP	0.15	EFO:0001187 UBERON:0002107	ENCFF072CXB ENCFF837QHJ ENCFF905JAC
HNF4G	ENCAB000AHQ	0.09	EFO:0001187 UBERON:0002107	ENCFF497MUF ENCFF086CTA
HNRNPK	ENCAB000BFN	0.09	EFO:0002067 EFO:0001187	ENCFF828KXG ENCFF984QUV
HNRNPL	ENCAB000AVZ	0.23	EFO:0002067 EFO:0001187	ENCFF984ESZ ENCFF039CUI
HNRNPLL	ENCAB698BTS	0.12	EFO:0002067 EFO:0001187	ENCFF662WPN ENCFF890KTX
IKZF1	ENCAB144WPX	0.32	EFO:0002067 EFO:0001187 EFO:0002784	ENCFF969BZA ENCFF968NOG ENCFF994OQH
IKZF1	ENCAB590IRI	0.27	EFO:0002067 EFO:0002784	ENCFF018NNF ENCFF785BTP
JUN	ENCAB000AER	0.12	EFO:0002067 EFO:0003042 EFO:0001187 EFO:0001203	ENCFF907UNK ENCFF331IUK ENCFF312GEN ENCFF394CEC ENCFF167WUZ ENCFF881AVX ENCFF672LKE ENCFF032UMW
JUNB	ENCAB000BQG	0.0	EFO:0002067 EFO:0002784	ENCFF478XNA ENCFF739XTO

JUND	ENCAB000AID	0.02	EFO:0001086 EFO:0002824 EFO:0002067 EFO:0001187 UBERON:0002107 EFO:0003072 EFO:0003042 EFO:0002784 EFO:0001203	ENCFF587VEY ENCFF569ZCY ENCFF213EYD ENCFF873DJD ENCFF998KDQ ENCFF246HKM ENCFF539GRW ENCFF420PED ENCFF187QQB ENCFF430PEI ENCFF443HNU ENCFF646IUA ENCFF229COM
KDM1A	ENCAB000AIH	0.11	EFO:0001086 EFO:0001187 EFO:0003042	ENCFF768FGG ENCFF316CBQ ENCFF562OAN
KDM5A	ENCAB914RSH	0.06	EFO:0001086 EFO:0001187	ENCFF149INM ENCFF334HKG
LARP7	ENCAB568DRS	0.32	EFO:0002067 EFO:0002784	ENCFF305SLO ENCFF668ZTJ
MAFF	ENCAB717EYF	0.14	EFO:0002067 EFO:0002791 EFO:0001187	ENCFF498MGH ENCFF493TIR ENCFF672LKL
MAFK	ENCAB000AIJ	0.11	EFO:0001086 EFO:0001203 EFO:0002067 EFO:0001187 EFO:0002791 EFO:0001196 EFO:0002784 EFO:0003042	ENCFF813WJW ENCFF171OJF ENCFF873SVI ENCFF351VGZ ENCFF186AWV ENCFF712RIS ENCFF328IZQ ENCFF893SCL
MAX	ENCAB000AIL	0.15	EFO:0002067 EFO:0001187 UBERON:0002107 EFO:0002784 EFO:0003042	ENCFF762LKG ENCFF140PUO ENCFF270NAL ENCFF493ZMX ENCFF618VMC ENCFF669BQN ENCFF900NVQ

MAZ	ENCAB000AIM	0.05	EFO:0001086 EFO:0001187 EFO:0001196 EFO:0002784 EFO:0001203	ENCFF666YGQ ENCFF144TBQ ENCFF661NNJ ENCFF916AJB ENCFF348STZ
MBD2	ENCAB000BQP	0.13	EFO:0002067 EFO:0001203	ENCFF464QAL ENCFF617QSK
MLLT1	ENCAB650PBW	0.16	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF578NMN ENCFF125MEN ENCFF010AIG
MNT	ENCAB000BCL	0.19	EFO:0002067 EFO:0001187 EFO:0001203	ENCFF562FMQ ENCFF432GSK ENCFF459DYU
MNT	ENCAB887GAG	0.06	EFO:0002067 EFO:0001187	ENCFF454QQD ENCFF482JSR
MTA1	ENCAB000BCN	0.24	EFO:0002067 EFO:0001203	ENCFF801KEW ENCFF225VFR
MTA3	ENCAB000BML	0.24	EFO:0002067 EFO:0001203	ENCFF083AZM ENCFF459XLR
MXI1	ENCAB000AIT	0.08	EFO:0002067 CL:0002319 EFO:0002784 EFO:0003072	ENCFF255WJM ENCFF199HGX ENCFF243QTL ENCFF116RCK
MYC	ENCAB000AET	0.15	EFO:0001086 EFO:0002067 EFO:0003042 EFO:0001203	ENCFF392JJN ENCFF542GMN ENCFF339AQP ENCFF605WXD ENCFF370EQJ ENCFF492XUU ENCFF300OKR ENCFF527EGF ENCFF658XME ENCFF700TLG
NANOG	ENCAB000AIX	0.28	EFO:0007950 EFO:0003042	ENCFF621PFM ENCFF794GVQ
NCOR1	ENCAB805KAO	0.26	EFO:0002067 EFO:0001187	ENCFF638IIC ENCFF616RSZ
NEUROD1	ENCAB000BDX	0.28	EFO:0002067 EFO:0001203	ENCFF059LJD ENCFF755APC
NFATC3	ENCAB375FHW	0.15	EFO:0002067 EFO:0002784	ENCFF430JFH ENCFF704PDA

NFE2L2	ENCAB800OND	0.1	EFO:0001086 EFO:0002791 EFO:0001187 EFO:0001196	ENCFF882YLO ENCFF305KIK ENCFF474PPT ENCFF418TUX
NFXL1	ENCAB208FGX	0.26	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF329STX ENCFF860IXB ENCFF927DIO
NFYB	ENCAB000AJD	0.11	EFO:0002067 EFO:0002784	ENCFF510NDO ENCFF009NXK
NKRF	ENCAB893CIV	0.14	EFO:0002067 EFO:0002784	ENCFF084NXU ENCFF520QSR
NONO	ENCAB097MPS	0.11	EFO:0002067 EFO:0001187	ENCFF823CQK ENCFF420QKI
NONO	ENCAB349QGP	0.34	EFO:0002067 EFO:0001203	ENCFF515YFU ENCFF800CDQ
NR2C1	ENCAB324ARN	0.18	EFO:0002067 EFO:0002784	ENCFF462AKP ENCFF023XHV
NR2F1	ENCAB000ATZ	0.2	EFO:0002067 EFO:0002784	ENCFF363IQN ENCFF531KOV
NR2F2	ENCAB000AJH	0.24	EFO:0002067 UBERON:0002107	ENCFF118HUH ENCFF819WNB ENCFF379TVQ
NR2F6	ENCAB854ATP	0.19	EFO:0002067 EFO:0001187	ENCFF350CKI ENCFF194VBK
NRF1	ENCAB000AJI	0.05	EFO:0001203 EFO:0001187 EFO:0002784 EFO:0003042	ENCFF652BRY ENCFF407IVS ENCFF418DKQ ENCFF269RME
NRF1	ENCAB000BLM	0.01	EFO:0002067 EFO:0001187	ENCFF313RFR ENCFF626VDA ENCFF543STN
PAX5	ENCAB000AJS	0.18	EFO:0002785 EFO:0002786 EFO:0002784	ENCFF997VAB ENCFF196JGP ENCFF987CQF
PAX8	ENCAB000BOS	0.34	EFO:0002784 EFO:0001203	ENCFF473UHQ ENCFF992JWY
PBX2	ENCAB697XQW	0.11	EFO:0002067 EFO:0001187	ENCFF925OBR ENCFF709YWO
PCBP1	ENCAB000BFX	0.0	EFO:0002067 EFO:0001187	ENCFF467RYH ENCFF487WAN
PCBP2	ENCAB000BFY	0.33	EFO:0002067 EFO:0001187	ENCFF642XRH ENCFF941XZW

PHF8	ENCAB757LUG	0.06	EFO:0001086 EFO:0001187	ENCFF907WHF ENCFF202WIO
POLR2A	ENCAB000AOC	0.02	EFO:0001086 EFO:0002824 EFO:0001203 EFO:0002067 EFO:0001187 EFO:0002786 EFO:0002791 CL:0002618 EFO:0002785 EFO:0002784 EFO:0003042	ENCFF246QVY ENCFF271RGE ENCFF403ZEO ENCFF422HDN ENCFF964EVA ENCFF565SUC ENCFF387VGY ENCFF455ZLJ ENCFF021HUZ ENCFF099NYA ENCFF741JES ENCFF798PUX ENCFF730DLS ENCFF664KTN ENCFF668VIK ENCFF915LKZ ENCFF182YZG
POLR2A phosphoS2	ENCAB000AOB	0.02	EFO:0001086 EFO:0002067 EFO:0001187 EFO:0002784	ENCFF652YVO ENCFF156MIR ENCFF266OPF ENCFF847DXY
POLR2G	ENCAB392AZR	0.19	EFO:0002067 EFO:0001187	ENCFF283CUY ENCFF551IJP
PRPF4	ENCAB624FBH	0.4	EFO:0002067 EFO:0001187	ENCFF417RQZ ENCFF908QCS
PTBP1	ENCAB553IRN	0.25	EFO:0002067 EFO:0001187	ENCFF875ZPV ENCFF917HXV
RAD21	ENCAB000AKG	0.04	EFO:0001086 CL:0002319 EFO:0001187 UBERON:0002107 EFO:0003072 EFO:0001196 EFO:0002784 EFO:0003042	ENCFF557OCR ENCFF897QCA ENCFF654EGO ENCFF454TRL ENCFF895JAW ENCFF295GOD ENCFF255FRL ENCFF874VFZ ENCFF060IVS ENCFF229WFR ENCFF093XOJ ENCFF315BSV

RAD21	ENCAB697XQW	0.09	EFO:0001187 EFO:0001203	ENCFF330VPL ENCFF081TVG
RAD51	ENCAB406ZKT	0.13	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF859MBC ENCFF996NBR ENCFF091AYX ENCFF740OPF
RB1	ENCAB000BFB	0.02	EFO:0002067 EFO:0002784	ENCFF034OSV ENCFF328QZM
RBBP5	ENCAB000AKH	0.03	EFO:0002067 EFO:0003042	ENCFF607WCG ENCFF666PCE
RBFOX2	ENCAB592TEY	0.05	EFO:0002067 EFO:0001187	ENCFF871YRG ENCFF232ASB
RBM22	ENCAB476CFS	0.32	EFO:0002067 EFO:0001187	ENCFF420IBN ENCFF305WYD
RBM39	ENCAB975CWB	0.25	EFO:0002067 EFO:0001187	ENCFF420ALF ENCFF503DIK
RCOR1	ENCAB000AFK	0.15	EFO:0003072 EFO:0001187 EFO:0002784 EFO:0001196	ENCFF073ADA ENCFF470ZMK ENCFF987VKU ENCFF139EBY
REST	ENCAB000AJK	0.05	EFO:0001086 EFO:0002067 EFO:0001187 EFO:0003072 UBERON:0002107 EFO:0002791 EFO:0002784 EFO:0003042	ENCFF023ZUW ENCFF208NUB ENCFF107EWI ENCFF403CAJ ENCFF540FXB ENCFF313CII ENCFF669XCW ENCFF986RRJ ENCFF796YFZ ENCFF274BBE ENCFF288XHG ENCFF178WRO
RFX5	ENCAB000AKJ	0.04	EFO:0001086 EFO:0001203 EFO:0002067 EFO:0001187 EFO:0003072 EFO:0002784 EFO:0003042	ENCFF502JJJ ENCFF062WBN ENCFF259LNG ENCFF201YKU ENCFF179WDI ENCFF059GWW ENCFF103MPW
RNF2	ENCAB000BEA	0.06	EFO:0002067 EFO:0001187	ENCFF380SYL ENCFF820LKT

RNF2	ENCAB790JUW	0.26	EFO:0002067 EFO:0003042	ENCFF462AZY ENCFF283MNG
RXRA	ENCAB000AKN	0.2	EFO:0003042 EFO:0001187 EFO:0002784 UBERON:0002107	ENCFF313BDA ENCFF105TFM ENCFF430SIE ENCFF572MCI ENCFF201KGJ
SAP30	ENCAB000AKO	0.15	EFO:0002067 EFO:0003042	ENCFF103RHL ENCFF193TFR
SIN3A	ENCAB000AKR	0.05	EFO:0001086 EFO:0001203 EFO:0002067 EFO:0002784 EFO:0003042	ENCFF514BGQ ENCFF802JAN ENCFF050CYK ENCFF567BJI ENCFF220RUS
SIN3A	ENCAB000AKS	0.08	EFO:0001086 EFO:0002067 EFO:0001187 EFO:0003072 EFO:0003042	ENCFF663RUS ENCFF407VGB ENCFF635YMI ENCFF708HTR ENCFF905VZD
SIN3B	ENCAB194LUB	0.19	EFO:0002067 EFO:0001187	ENCFF193DQZ ENCFF543INR
SIX5	ENCAB000AKV	0.07	EFO:0001086 EFO:0002067 EFO:0002784 EFO:0003042	ENCFF864TFH ENCFF189NMX ENCFF644BNN ENCFF247LOF
SKIL	ENCAB000BNW	0.3	EFO:0002067 EFO:0002784	ENCFF254QDM ENCFF903KEI
SMAD1	ENCAB000AYH	0.16	EFO:0002067 EFO:0002784	ENCFF987PGY ENCFF084BUP
SMAD5	ENCAB000AAG	0.02	EFO:0002067 EFO:0002784	ENCFF855SJG ENCFF069AAY
SMARCA4	ENCAB000AEO	0.26	CL:0000103 EFO:0002067	ENCFF703NAE ENCFF482JUI
SMARCA5	ENCAB528BVW	0.21	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF052STI ENCFF481TNF ENCFF618JNX
SMARCC2	ENCAB313DWJ	0.18	EFO:0002067 EFO:0001187	ENCFF150NHK ENCFF751ZVX
SMARCE1	ENCAB550CKA	0.27	EFO:0002067 EFO:0001203	ENCFF435SZS ENCFF761NKP

SMC3	ENCAB000AKX	0.02	EFO:0001086 EFO:0002067 CL:0002319 EFO:0001187 EFO:0001196 EFO:0002784	ENCFF035YWE ENCFF572RPI ENCFF944KJO ENCFF380ZXB ENCFF256LDD ENCFF175UEE
SNIP1	ENCAB027LVC	0.14	EFO:0002067 EFO:0001203	ENCFF529BDW ENCFF455HWV
SOX6	ENCAB000BOG	0.43	EFO:0002067 EFO:0001187	ENCFF944LNI ENCFF431STY
SP1	ENCAB000AKY	0.16	UBERON:0002107 EFO:0001086 EFO:0001187 EFO:0003042	ENCFF404OSB ENCFF500JFI ENCFF175VXL ENCFF433EFF ENCFF978TMH
SP1	ENCAB000BAV	0.25	EFO:0002067 EFO:0001187 EFO:0001203	ENCFF577EMC ENCFF452LDK ENCFF735WMX
SPI1	ENCAB000AKF	0.15	EFO:0002067 EFO:0002785 EFO:0002784	ENCFF414ECK ENCFF744AGB ENCFF071ZMW
SREBF1	ENCAB000ALC	0.14	EFO:0001086 EFO:0002067 EFO:0001203	ENCFF275WAD ENCFF777MYW ENCFF624DDK
SRF	ENCAB000ALE	0.14	EFO:0002784 EFO:0003042	ENCFF345IDL ENCFF182IFE ENCFF829SEJ
STAT1	ENCAB000ALF	0.16	EFO:0002067 EFO:0002784	ENCFF323QQU ENCFF747ICD ENCFF431NLF ENCFF646MXG
STAT5A	ENCAB000ALI	0.29	EFO:0002067 EFO:0002784	ENCFF383YEA ENCFF517IXK
SUZ12	ENCAB000BEB	-0.02	EFO:0002067 EFO:0001187	ENCFF856HYC ENCFF239LRW

TAF1	ENCAB000ALM	0.02	EFO:0001086 EFO:0001203 EFO:0002067 EFO:0001187 EFO:0002786 EFO:0003072 UBERON:0002107 EFO:0002791 EFO:0002785 EFO:0002784 EFO:0003042	ENCFF033PLJ ENCFF762MGC ENCFF234TBW ENCFF471NIK ENCFF453TIB ENCFF278XOE ENCFF540AAP ENCFF870SFJ ENCFF886KDK ENCFF214OJW ENCFF423CTO
TARDBP	ENCAB000AUF	0.02	EFO:0002067 EFO:0002784	ENCFF641AXD ENCFF871LZM
TARDBP	ENCAB000BAX	0.16	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF909RMQ ENCFF668JHK ENCFF233RBO
TARDBP	ENCAB057RGG	0.18	EFO:0002067 EFO:0001187	ENCFF448YOS ENCFF696QPP
TBL1XR1	ENCAB000ALP	0.1	EFO:0002067 EFO:0001187 EFO:0002784	ENCFF126KGW ENCFF392JWA ENCFF239WFN
TBP	ENCAB000ALR	0.08	EFO:0002067 EFO:0001187 EFO:0002791 EFO:0002784 EFO:0003042	ENCFF370YGS ENCFF748YXF ENCFF302RQH ENCFF534GKQ ENCFF896UZB
TCF12	ENCAB000ALT	0.22	EFO:0001086 EFO:0001187 EFO:0002784 EFO:0003042	ENCFF740HPV ENCFF768VSH ENCFF299JYV ENCFF228CDD
TCF12	ENCAB506UYG	0.38	EFO:0002067 EFO:0002784	ENCFF897RYA ENCFF912LXU
TCF7	ENCAB000ACR	0.27	EFO:0002067 EFO:0001187 EFO:0002784	ENCFF152RNE ENCFF512IAI ENCFF928MIN

TOE1	ENCAB755SML	0.19	EFO:0002067 EFO:0001187 EFO:0001203	ENCFF539FVQ ENCFF014WCO ENCFF144VMM
TRIM22	ENCAB000BNM	0.33	EFO:0001187 EFO:0002784 EFO:0001203	ENCFF063GDN ENCFF452VLA ENCFF830TFU ENCFF552WAH
U2AF1	ENCAB298WVV	0.33	EFO:0002067 EFO:0001187	ENCFF034KUO ENCFF482DRO
USF1	ENCAB000AMF	0.08	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0003042	ENCFF701QXK ENCFF717KGR ENCFF914IFQ ENCFF699HXL
USF2	ENCAB000AMH	0.07	EFO:0001086 EFO:0002067 EFO:0001196 EFO:0002784 EFO:0003042	ENCFF514SWA ENCFF425FVY ENCFF593EOW ENCFF938BOJ ENCFF710JBU
XRCC5	ENCAB308AOH	0.04	EFO:0002067 EFO:0001187	ENCFF929TWP ENCFF790ZAQ
YBX1	ENCAB493UWX	0.17	EFO:0001187 EFO:0002784 EFO:0001203	ENCFF332FUE ENCFF247VVK ENCFF500RBO
YY1	ENCAB000ANS	0.03	EFO:0002067 EFO:0002786	ENCFF072IHJ ENCFF635XCI ENCFF024TJO
YY1	ENCAB000ANT	0.04	EFO:0001086 EFO:0002824 EFO:0002067 EFO:0001187 EFO:0003072 UBERON:0002107 EFO:0002785 EFO:0002784 EFO:0003042	ENCFF953BTB ENCFF613DTQ ENCFF363UWP ENCFF094BQZ ENCFF177YDT ENCFF509GYP ENCFF538VYU ENCFF223MUF ENCFF838VFX ENCFF459TWF
ZBED1	ENCAB000AAK	0.21	EFO:0002067 EFO:0002784	ENCFF388TYU ENCFF630FLK

ZBTB33	ENCAB000AML	0.15	EFO:0001086 EFO:0002824 EFO:0001187 UBERON:0002107 EFO:0002784	ENCFF422MCZ ENCFF593ZJA ENCFF943WRA ENCFF773OQL ENCFF727ZIT ENCFF882UHR
ZBTB33	ENCAB292USO	0.21	EFO:0002067 EFO:0002784 EFO:0001203	ENCFF556STK ENCFF780WLS ENCFF475DID
ZBTB40	ENCAB373DVF	0.08	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF932XEU ENCFF624WDI ENCFF088LZZ ENCFF084IUW
ZBTB7A	ENCAB000AMM	0.07	EFO:0002067 EFO:0001187	ENCFF953JQD ENCFF245LRG
ZC3H11A	ENCAB000AMN	0.44	EFO:0001086 EFO:0002067	ENCFF478PGJ ENCFF415SIS
ZFP36	ENCAB118PND	0.09	EFO:0001086 EFO:0002067 EFO:0001187 EFO:0002791 EFO:0002784	ENCFF166GKK ENCFF429XQI ENCFF137JHO ENCFF224WII ENCFF763HPQ
ZFX	ENCAB657HDP	0.05	EFO:0001203 EFO:0002824	ENCFF215SIC ENCFF775BWJ
ZHX1	ENCAB361RPF	0.04	EFO:0002067 EFO:0002791	ENCFF267DZF ENCFF495BPY
ZHX2	ENCAB000ATW	0.17	EFO:0001187 EFO:0001203	ENCFF694ZRC ENCFF964KDQ
ZKSCAN1	ENCAB000AMP	0.26	EFO:0002067 EFO:0001187 EFO:0001203	ENCFF687REM ENCFF704VDI ENCFF721NEC
ZMYM3	ENCAB426WVA	0.19	EFO:0002067 EFO:0001187	ENCFF195IFB ENCFF769SEZ
ZNF143	ENCAB000AMR	0.06	EFO:0002067 EFO:0002784 EFO:0003042	ENCFF933WSP ENCFF700GZI ENCFF193POQ ENCFF153TQR
ZNF207	ENCAB000BNU	0.24	EFO:0002784 EFO:0001203	ENCFF676BIG ENCFF621ZSK

ZNF217	ENCAB182PZR	0.15	EFO:0002784 EFO:0001203	ENCFF200SLC ENCFF620RPM
ZNF24	ENCAB060JJI	0.1	EFO:0002067 EFO:0001187	ENCFF858WPR ENCFF723JDW
ZNF24	ENCAB198YAJ	0.08	EFO:0002067 EFO:0001187 EFO:0002784 EFO:0001203	ENCFF619BFO ENCFF313HBL ENCFF260CBQ ENCFF904QAD
ZNF282	ENCAB503NQV	0.0	EFO:0002067 EFO:0001187	ENCFF596JDS ENCFF482XNG
ZNF592	ENCAB438BKV	0.1	EFO:0002067 EFO:0001203	ENCFF972UGK ENCFF541HRT
ZNF687	ENCAB146FZU	0.14	EFO:0002784 EFO:0001203	ENCFF137BRA ENCFF329QYZ
ZSCAN29	ENCAB211EDR	0.1	EFO:0002067 EFO:0002784	ENCFF214NJL ENCFF979GFF

ENCODE Name	Cell Type
CL:0000062	osteoblast
CL:0000103	bipolar neuron
CL:0000127	astrocyte
CL:0000182	hepatocyte
CL:0000192	smooth muscle cell
CL:0002319	neural cell
CL:0002372	myotube
CL:0002551	fibroblast of dermis
CL:0002618	endothelial cell of umbilical vein
EFO:0001086	A549
EFO:0001187	HepG2
EFO:0001196	IMR-90
EFO:0001203	MCF-7
EFO:0002067	K562
EFO:0002074	PC-3
EFO:0002784	GM12878
EFO:0002785	GM12891
EFO:0002791	HeLa-S3
EFO:0002824	HCT116
EFO:0003042	H1
EFO:0003072	SK-N-SH
EFO:0006711	OCI-LY7
EFO:0007950	GM23338
UBERON:0002107	liver
NTR:0000711	neural progenitor cell

Table B.2: Cell type names and their corresponding ENCODE representations.

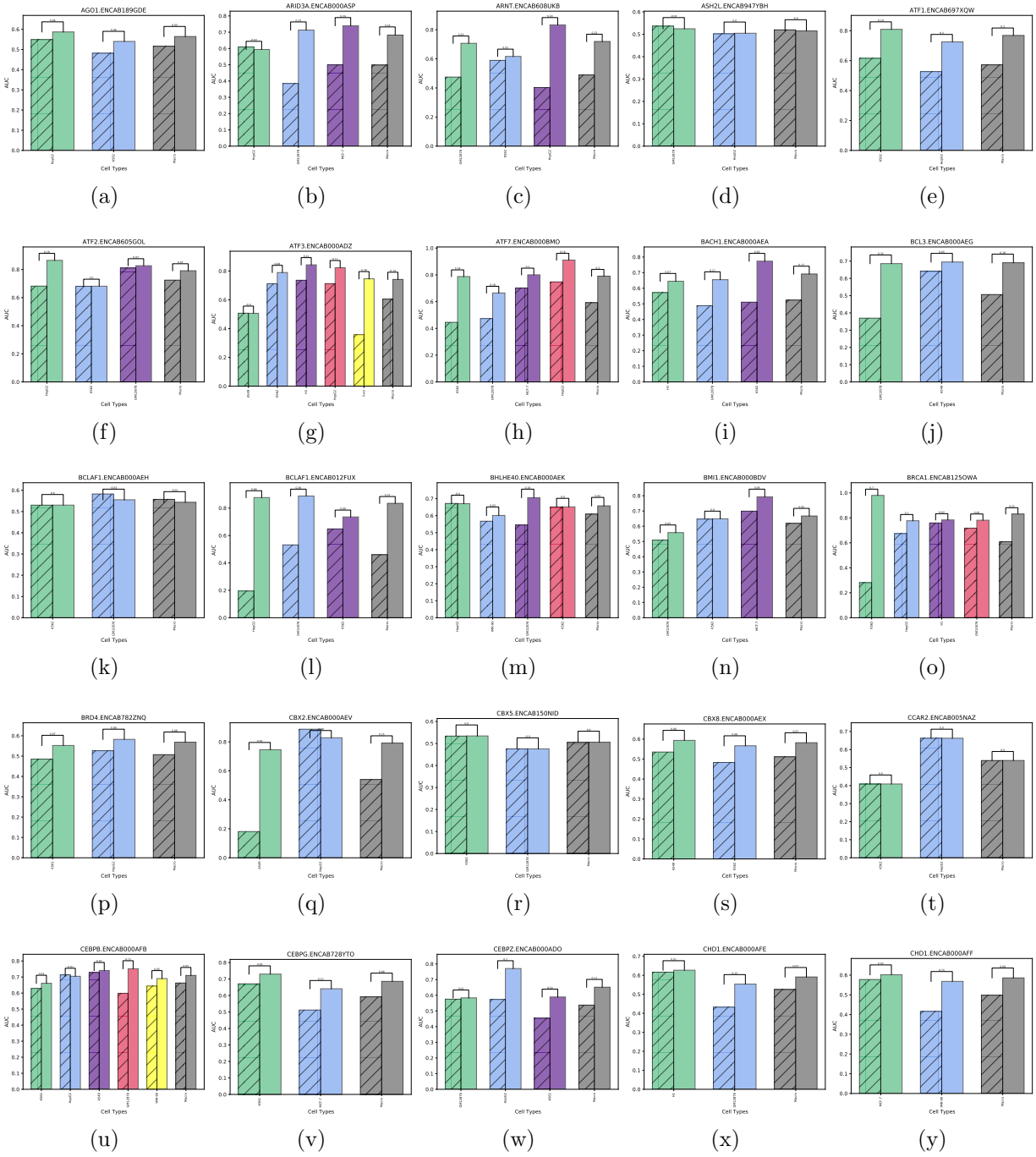
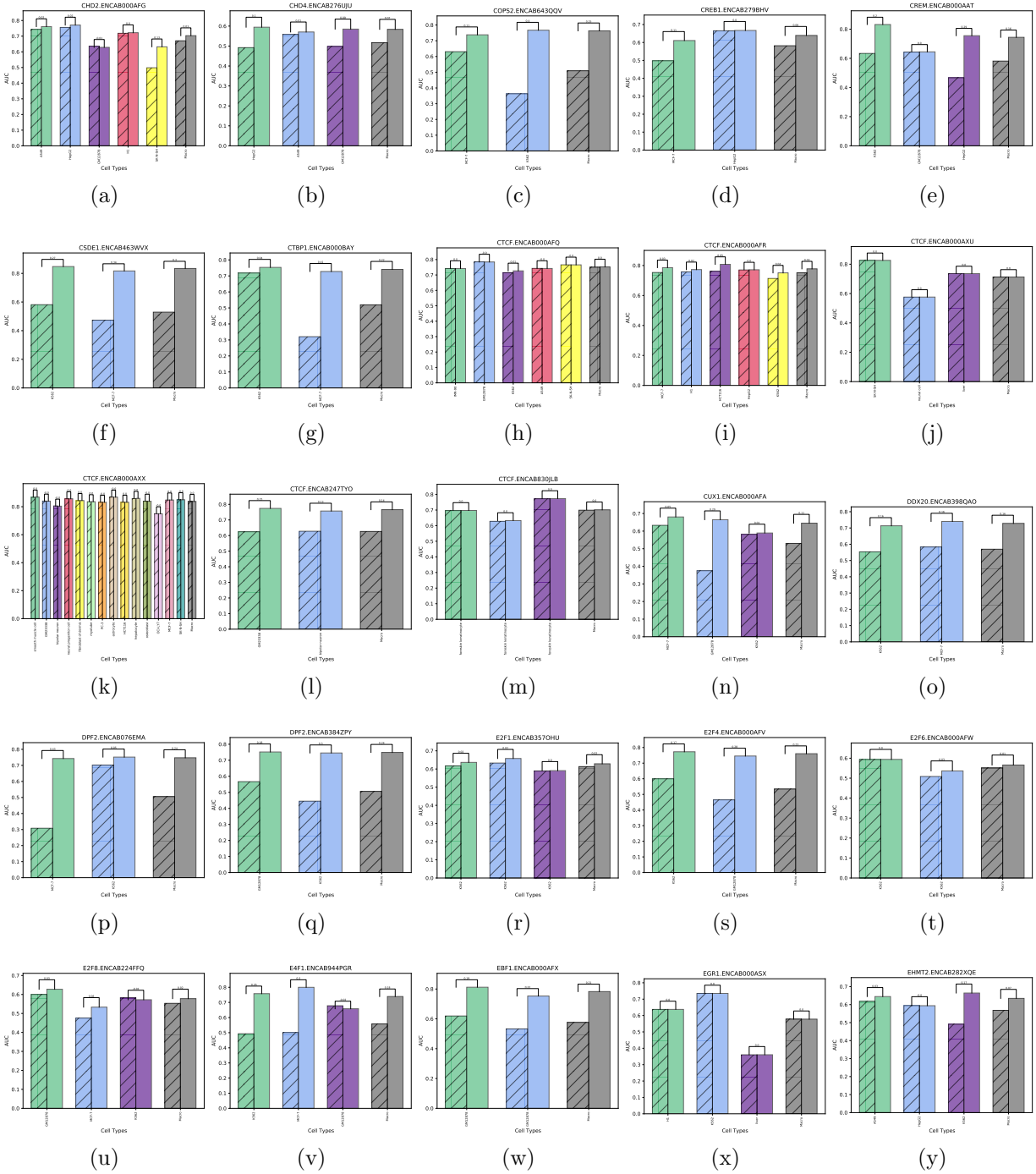


Figure B.1: Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Figure B.2: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.

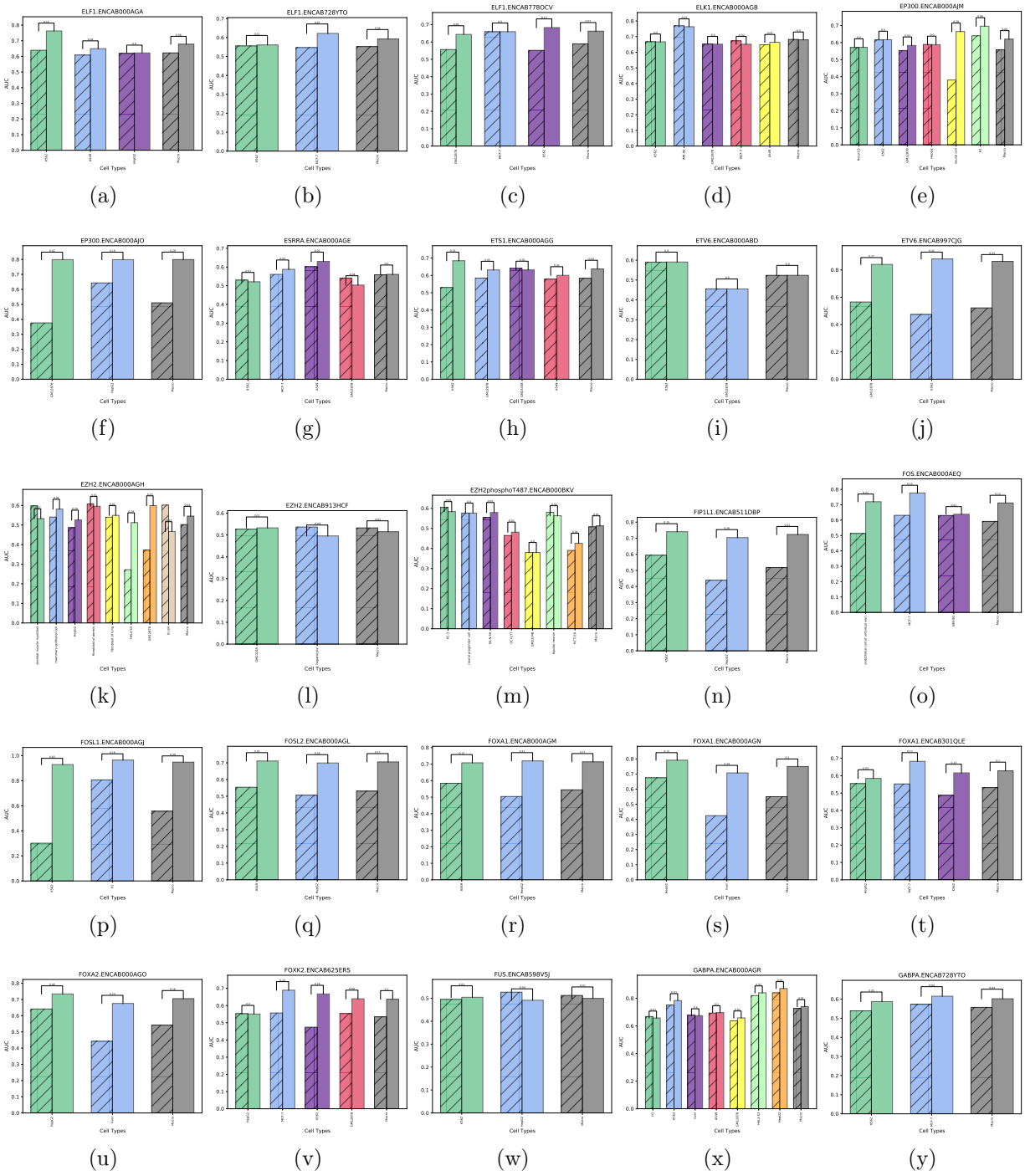


Figure B.3: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.

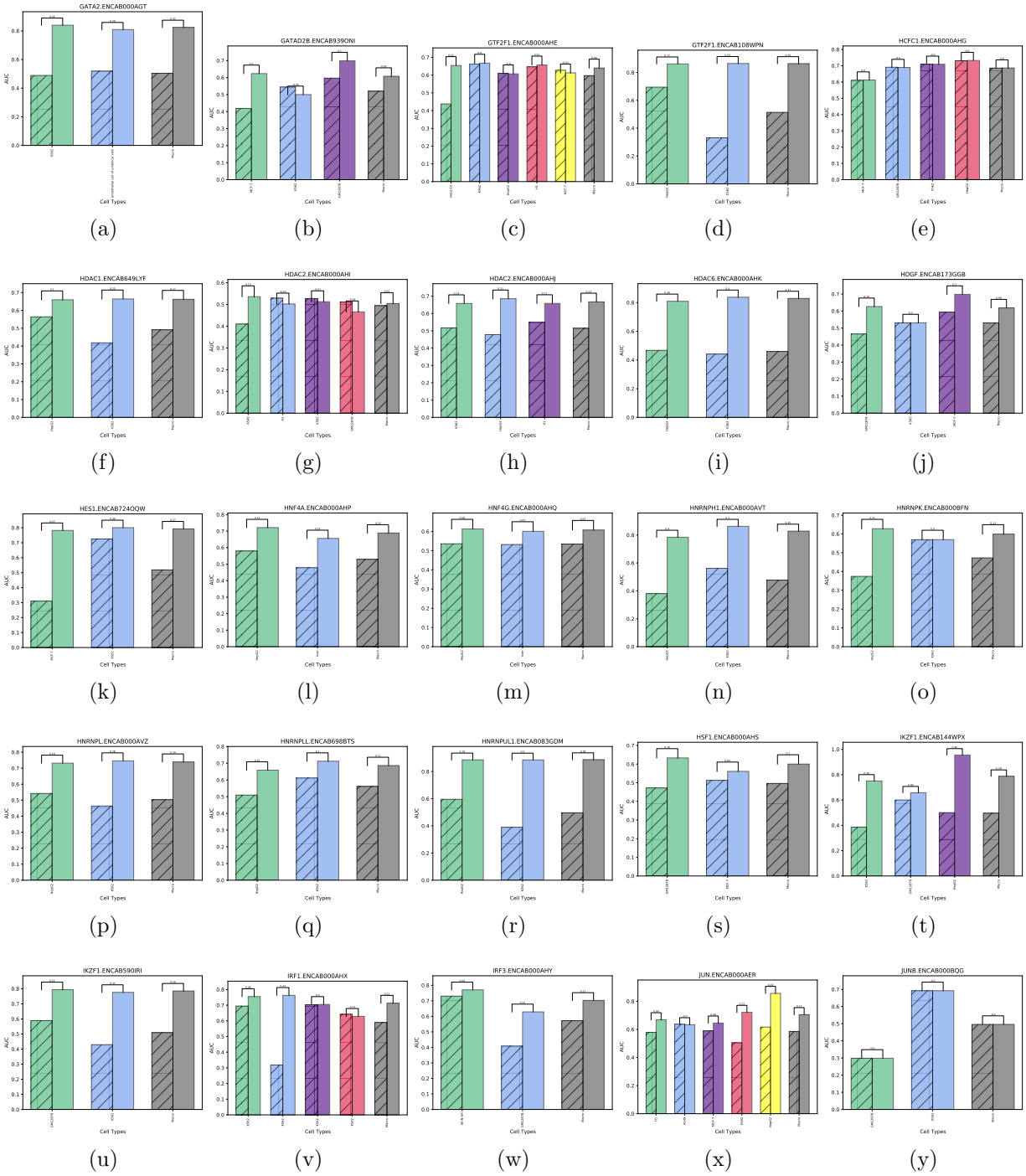
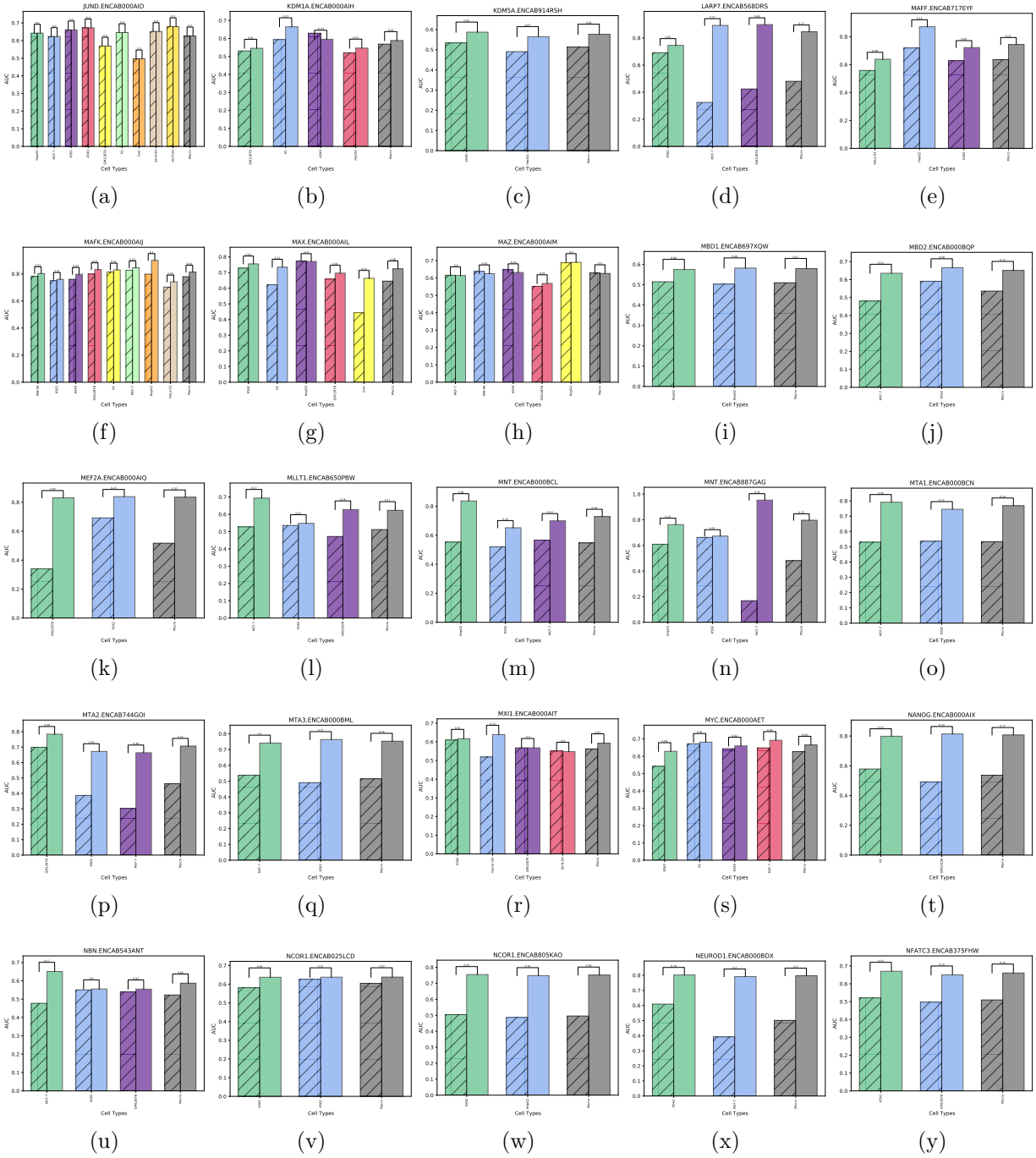


Figure B.4: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Figure B.5: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.

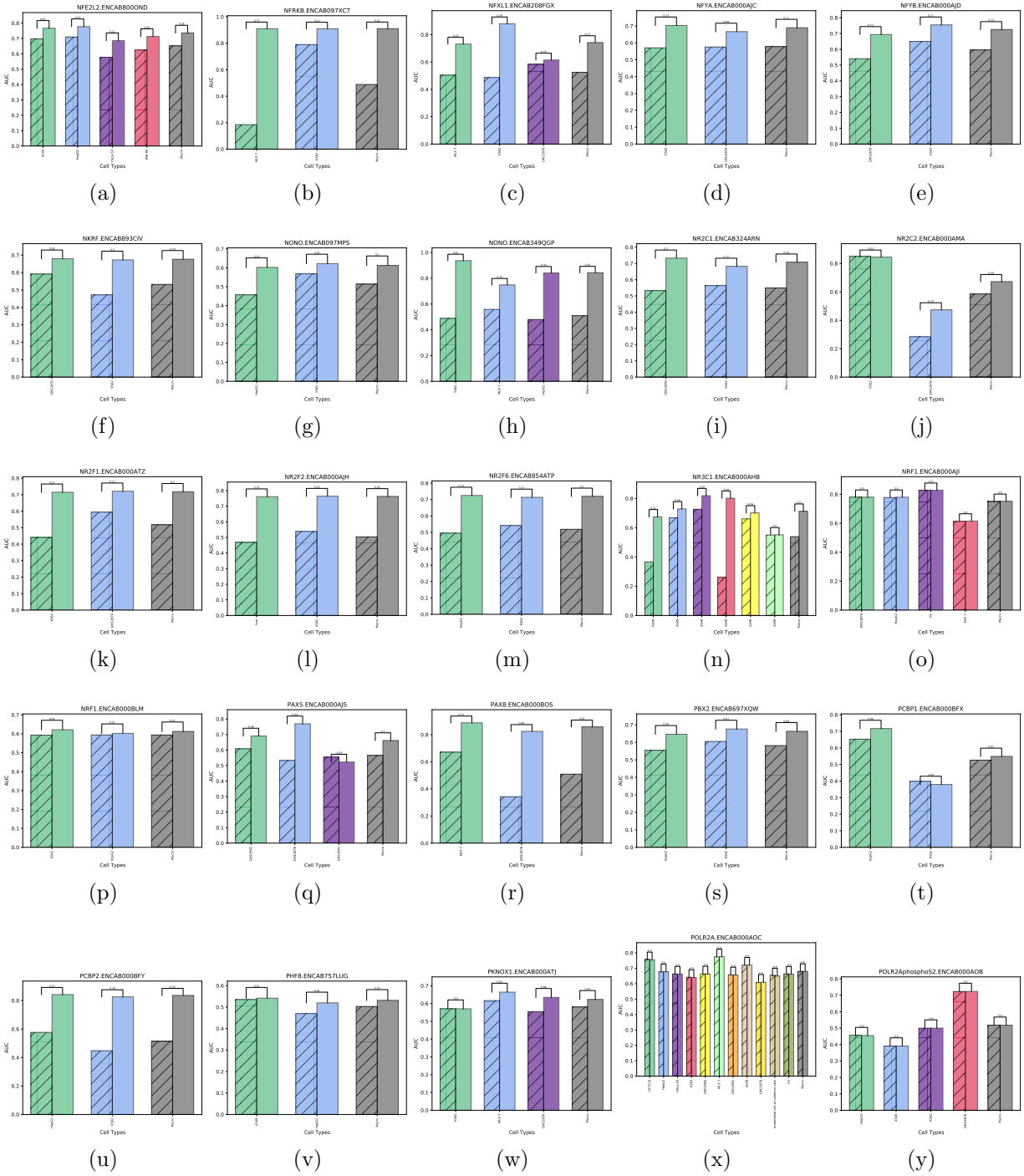
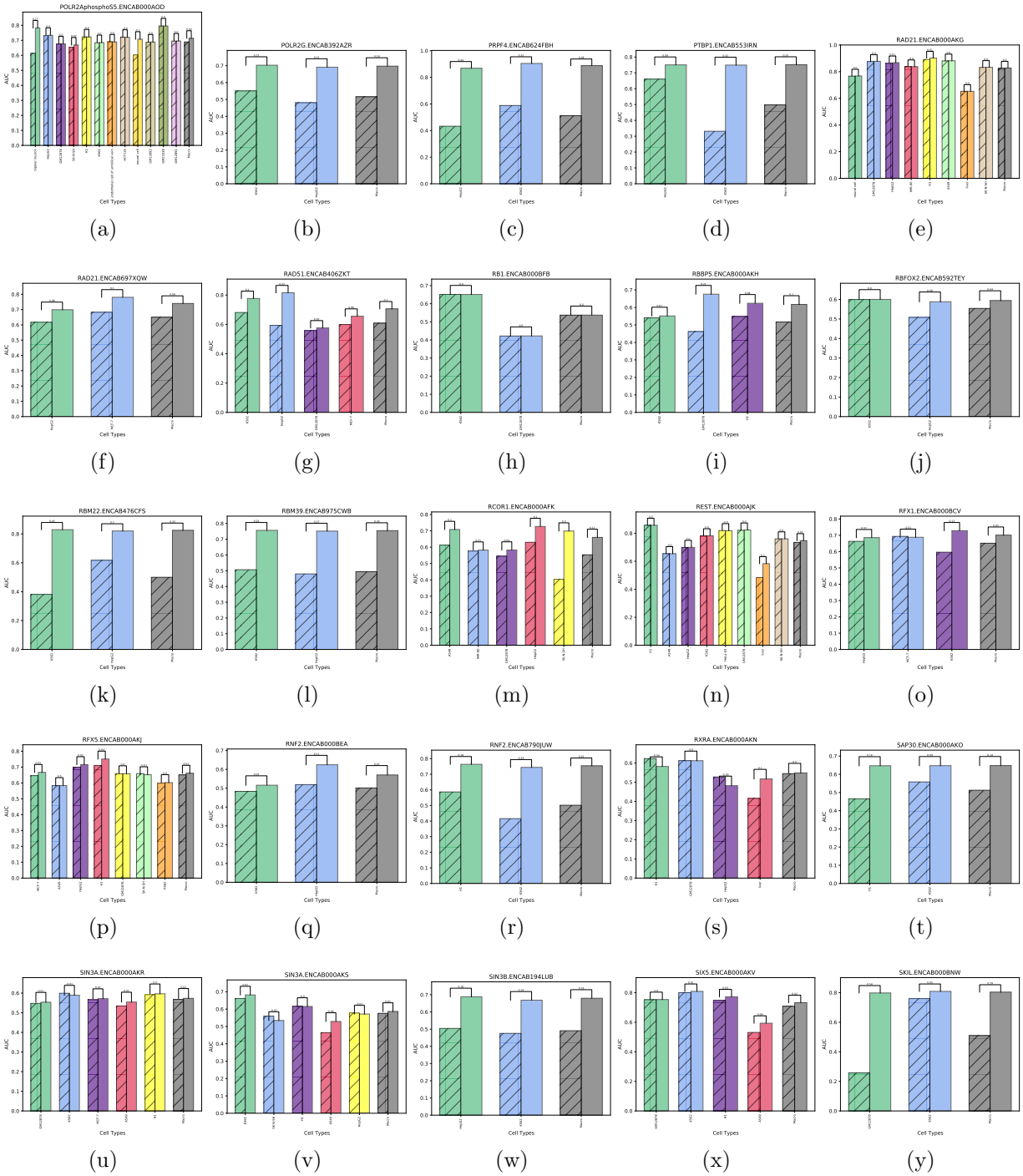


Figure B.6: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.



(a) (b) (c) (d) (e)

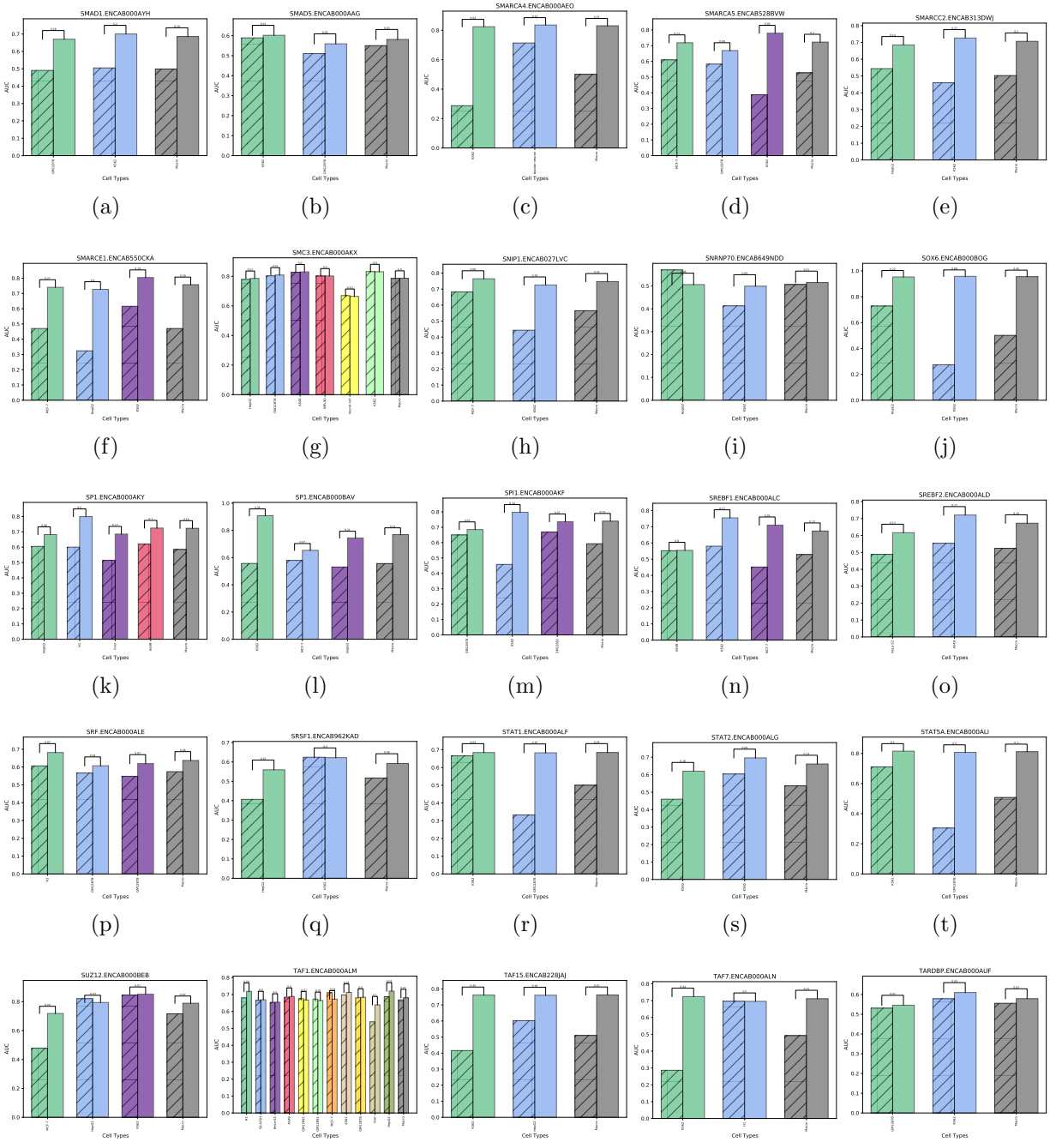
(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Figure B.7: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

(u) (v) (w) (x) (y)

Figure B.8: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.

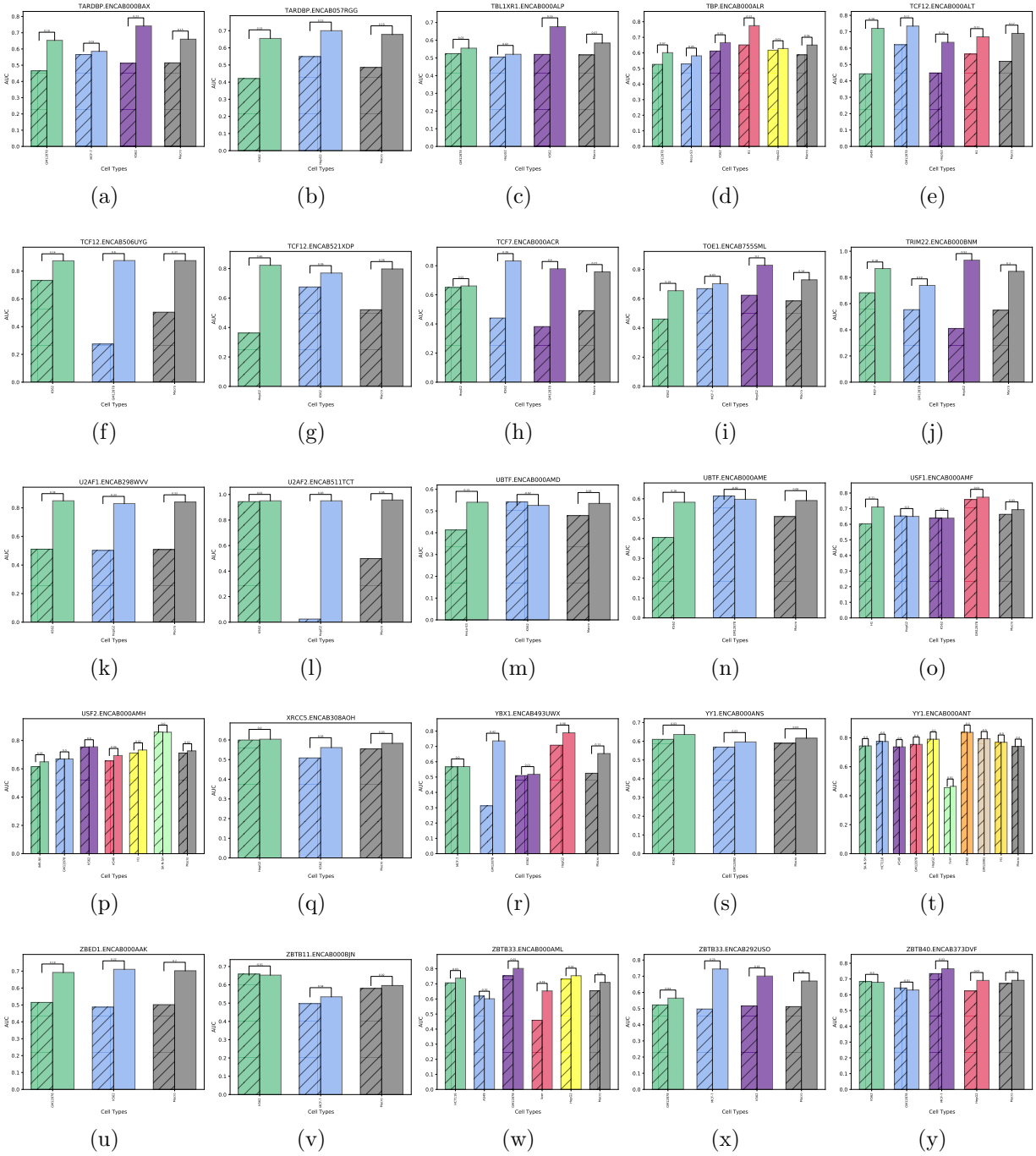
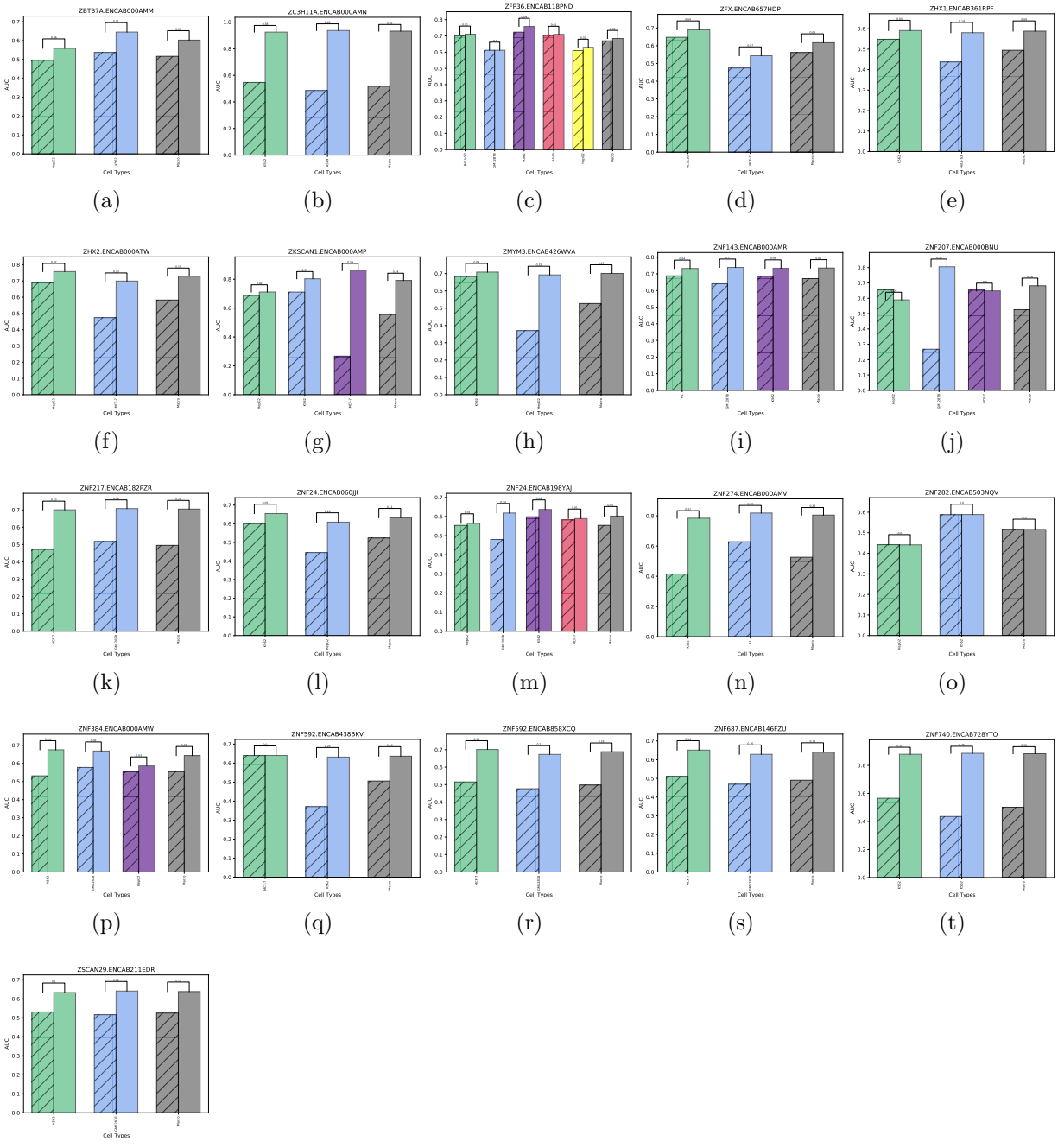


Figure B.9: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

(p) (q) (r) (s) (t)

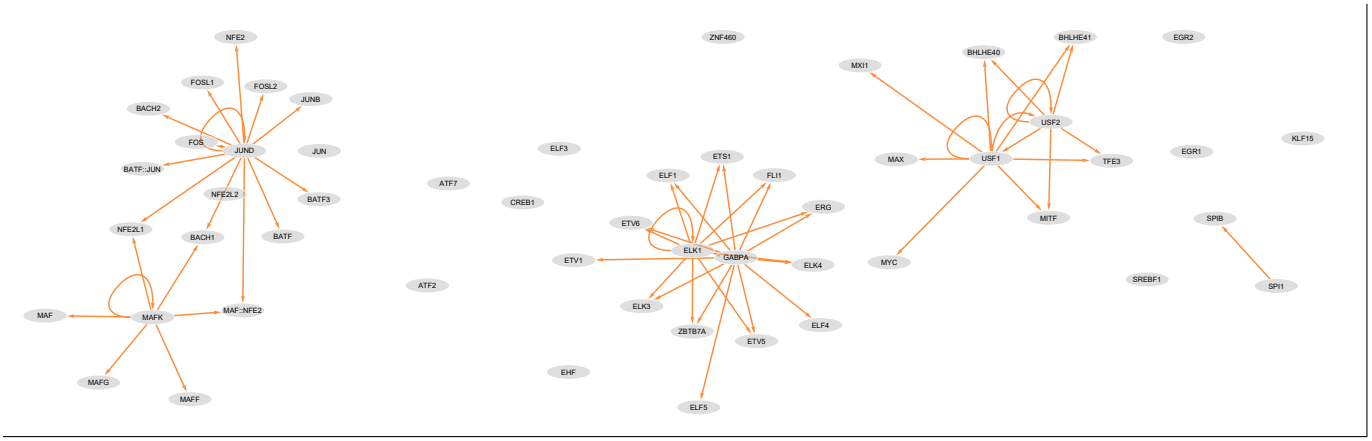
(u)

Figure B.10: (continued) Bar plots of AUC differences per cell type per TF-AB for cell type specific (shaded) and cell type general (dashed) cases on the test dataset.

Appendix C

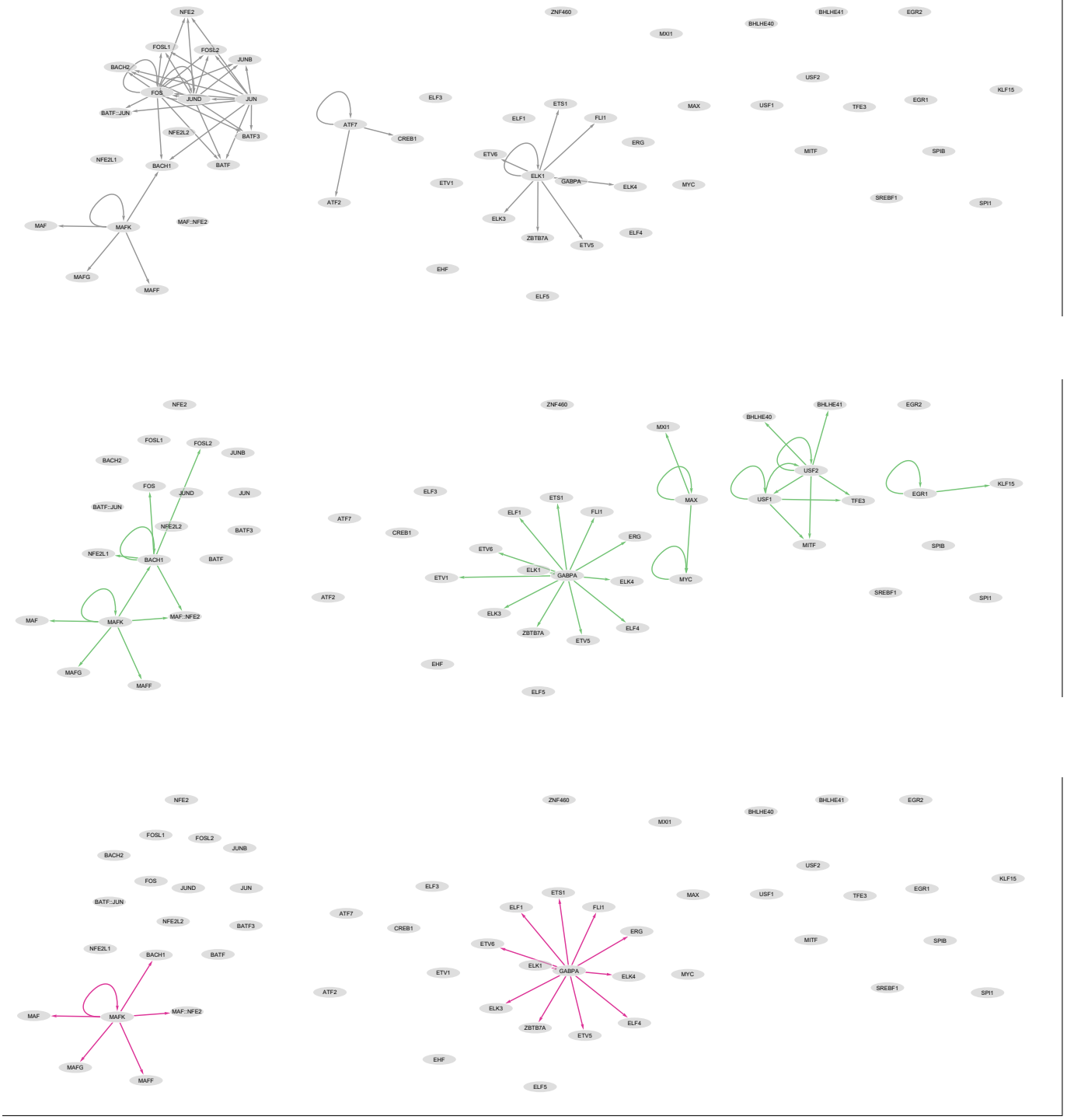
Motif Enrichment: Appendix

Appendix C provides the supplementary information for Chapter 5 on "Motif Enrichment and Cell Type Specificity in Transcription Factors". This includes figures of motif enrichment networks for various cell types, such as GM12878, MCF-7, H1, HeLa-S3, HCT116,



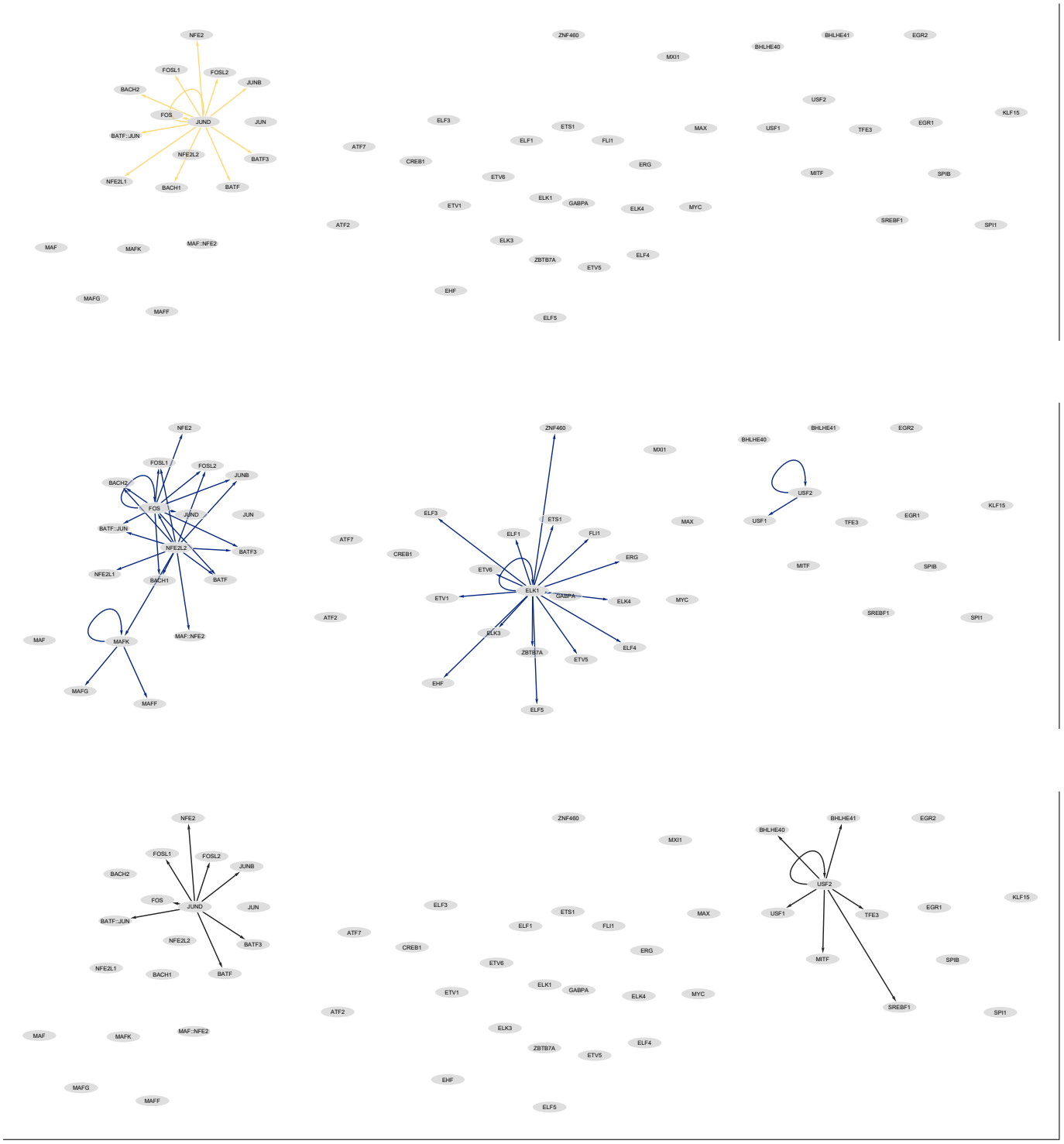
(a) GM12878

Figure C.1: Motif enrichment networks for cell type (a) GM12878.



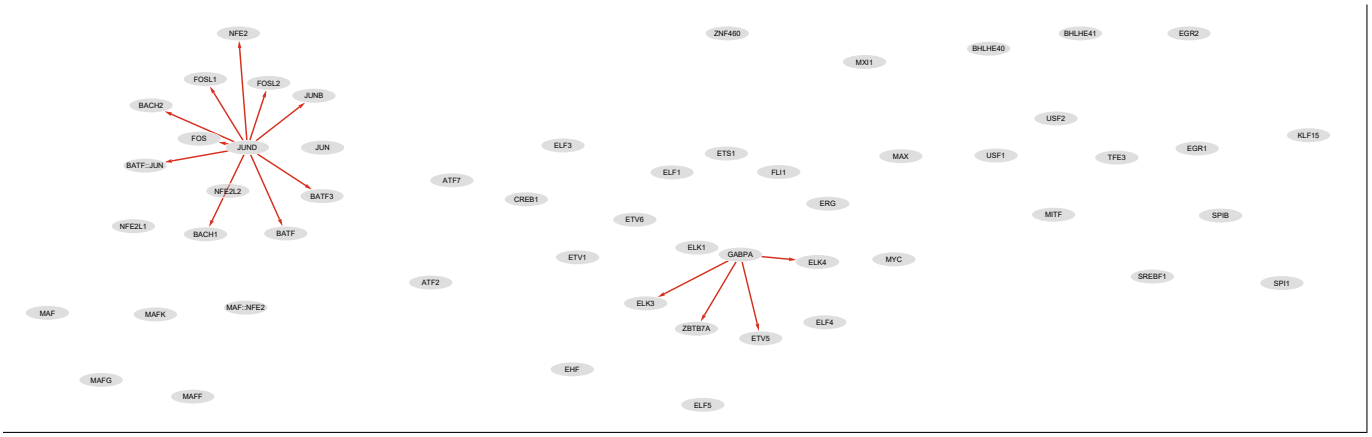
(d) HeLa-S3

Figure C.1: (continued) Motif enrichment networks for each of cell types (b) MCF-7, (c) H1 and (d) HeLa-S3.



(g) SK-N-SH

Figure C.1: (continued) Motif enrichment networks for each of cell types (e) HCT116, (f) IMR-90 and (g) SK-N-SH.



(h) Liver

Figure C.1: (continued) Motif enrichment networks for cell type (h) Liver.

References

[1] The Gene Ontology resource: enriching a GOld mine. *Nucleic Acids Research*, 49(D1):D325–D334, 2021.

[2] J Adams. DNA sequencing technologies. *Nature Education*, 2008.

[3] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in Neural Information Processing Systems*, 31, 2018.

[4] Phaedra Agius, Aaron Arvey, William Chang, William Stafford Noble, and Christina Leslie. High resolution models of transcription factor-DNA affinities improve in vitro and in vivo binding predictions. *PLOS Computational Biology*, 6(9):e1000916, 2010.

[5] Jelena Aleksic, Sarah Carl, and Michaela Frye. Beyond library size: a field guide to NGS normalization. *bioRxiv*, page 006403, 2014.

[6] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature Biotechnology*, 33(8):831–838, 2015.

[7] Giovanna Ambrosini, Ilya Vorontsov, Dmitry Penzar, Romain Groux, Oriol Fornes, Daria D Nikolaeva, Benoit Ballester, Jan Grau, Ivo Grosse, Vsevolod Makeev, et al. Insights gained from a comprehensive all-against-all transcription factor binding motif benchmarking study. *Genome Biology*, 21(1):1–18, 2020.

[8] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

[9] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Gradient-based attribution methods. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 169–191. Springer, 2019.

[10] Robin Andersson and Albin Sandelin. Determinants of enhancer and promoter activities of regulatory elements. *Nature Reviews Genetics*, 21(2):71–87, 2020.

[11] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7):878, 2016.

[12] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

[13] Oswald T Avery, Colin M MacLeod, and Maclyn McCarty. Studies on the chemical nature of the substance inducing transformation of pneumococcal types: induction of transformation by a desoxyribonucleic acid fraction isolated from pneumococcus type III. *The Journal of Experimental Medicine*, 79(2):137–158, 1944.

[14] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, 2021.

[15] Žiga Avsec, Melanie Weilert, Avanti Shrikumar, Sabrina Krueger, Amr Alexandari, Khyati Dalal, Robin Fropf, Charles McAnany, Julien Gagneur, Anshul Kundaje, et al. Base-resolution models of transcription-factor binding reveal soft motif syntax. *Nature Genetics*, 53(3):354–366, 2021.

[16] Aseel Awdeh, Marcel Turcotte, and Theodore J Perkins. WACS: improving ChIP-seq peak calling by optimally weighting controls. *BMC Bioinformatics*, 22(1):1–21, 2021.

[17] Aseel Awdeh, Marcel Turcotte, and Theodore J. Perkins. Cell type specific DNA signatures of transcription factor binding. *bioRxiv*, 2022.

[18] Gwenael Badis, Michael F Berger, Anthony A Philippakis, Shaheynoor Talukder, Andrew R Gehrke, Savina A Jaeger, Esther T Chan, Genita Metzler, Anastasia Vedenko, Xiaoyu Chen, et al. Diversity and complexity in DNA recognition by transcription factors. *Science*, 324(5935):1720–1723, 2009.

[19] Timothy L Bailey. DREME: motif discovery in transcription factor ChIP-seq data. *Bioinformatics*, 27(12):1653–1659, 2011.

[20] Timothy L Bailey, Mikael Boden, Fabian A Buske, Martin Frith, Charles E Grant, Luca Clementi, Jingyuan Ren, Wilfred W Li, and William S Noble. MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl_2):W202–W208, 2009.

[21] Timothy L Bailey and Charles Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Machine Learning*, 21(1):51–80, 1995.

[22] Maximilian Balandat, Brian Karrer, Daniel R Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. Botorch: Programmable bayesian optimization in pytorch. *arxiv e-prints*, pages arXiv–1910, 2019.

[23] Nilanjana Banerjee and Michael Q Zhang. Identifying cooperativity among transcription factors controlling the cell cycle in yeast. *Nucleic Acids Research*, 31(23):7024–7031, 2003.

[24] A. F. Bardet, Q. He, J. Zeitlinger, and A. Stark. A computational pipeline for comparative ChIP-seq analyses. *Nature Protocols*, 7(1):45, 2012.

[25] Anaïs F Bardet, Jonas Steinmann, Sangeeta Bafna, Juergen A Knoblich, Julia Zeitlinger, and Alexander Stark. Identification of transcription factor binding sites from ChIP-seq data at high resolution. *Bioinformatics*, 29(21):2705–2713, 2013.

[26] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, 2007.

[27] Sam Behjati and Patrick S Tarpey. What is next generation sequencing? *Archives of Disease in Childhood-Education and Practice*, 98(6):236–238, 2013.

[28] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

[29] Yuval Benjamini and Terence P Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 40(10):e72–e72, 2012.

[30] Michael F Berger and Martha L Bulyk. Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. *Nature Protocols*, 4(3):393–411, 2009.

[31] Michael F Berger, Anthony A Philippakis, Aaron M Qureshi, Fangxue S He, Preston W Estep 3rd, and Martha L Bulyk. Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotechnology*, 24(11):1429, 2006.

[32] Philipp Bergmaier, Oliver Weth, Sven Dienstbach, Thomas Boettger, Niels Galjart, Marco Mernberger, Marek Bartkuhn, and Rainer Renkawitz. Choice of binding sites for CTCFL compared to CTCF is driven by chromatin and by sequence preference. *Nucleic Acids Research*, 46(14):7097–7107, 2018.

[33] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, 24, 2011.

[34] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2), 2012.

[35] Bradley E Bernstein, John A Stamatoyannopoulos, Joseph F Costello, Bing Ren, Aleksandar Milosavljevic, Alexander Meissner, Manolis Kellis, Marco A Marra, Arthur L Beaudet, Joseph R Ecker, et al. The NIH roadmap epigenomics mapping consortium. *Nature Biotechnology*, 28(10):1045–1048, 2010.

[36] Volker Blank. Small Maf proteins in mammalian gene control: mere dimerization partners or dynamic transcriptional regulators? *Journal of Molecular Biology*, 376(4):913–925, 2008.

[37] Marzenna Blonska, Yifan Zhu, Hubert H Chuang, M James You, Kranthi Kunkalla, Francisco Vega, and Xin Lin. JUN-regulated genes promote interaction of diffuse large B-cell lymphoma with the microenvironment. *Blood, The Journal of the American Society of Hematology*, 125(6):981–991, 2015.

[38] Alan P Boyle, Justin Guinney, Gregory E Crawford, and Terrence S Furey. F-Seq: a feature density estimator for high-throughput sequence tags. *Bioinformatics*, 24(21):2537–2538, 2008.

[39] Marjorie Brand, Jeffrey A Ranish, Nicolas T Kummer, Joan Hamilton, Kazuhiko Igarashi, Claire Francastel, Tian H Chi, Gerald R Crabtree, Ruedi Aebersold, and Mark Groudine. Dynamic changes in transcription factor complexes during erythroid differentiation revealed by quantitative proteomics. *Nature Structural & Molecular Biology*, 11(1):73–80, 2004.

[40] Laure Bridoux, Peyman Zarrineh, Joshua Mallen, Mike Phuycharoen, Victor Latorre, Frank Ladam, Marta Losa, Syed Murtuza Baker, Charles Sagerstrom, Kimberly A Mace, et al. HOX paralogs selectively convert binding of ubiquitous transcription factors into tissue-specific patterns of enhancer activation. *PLOS Genetics*, 16(12):e1009162, 2020.

[41] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[42] Michele Busby, Catherine Xue, Catherine Li, Yossi Farjoun, Elizabeth Gienger, Ido Yofe, Adrianne Gladden, Charles B Epstein, Evan M Cornett, Scott B Rothbart, et al. Systematic comparison of monoclonal versus polyclonal antibodies for mapping histone modifications by ChIP-seq. *Epigenetics & Chromatin*, 9(1):1–16, 2016.

[43] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[44] Jaime A Castro-Mondragon, Rafael Riudavets-Puig, Ieva Rauluseviciute, Roza Berhanu Lemma, Laura Turchi, Romain Blanc-Mathieu, Jeremy Lucas, Paul Boddie, Aziz Khan, Nicolás Manosalva Pérez, et al. JASPAR 2022: the 9th release of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 50(D1):D165–D173, 2022.

[45] Jaime Abraham Castro-Mondragon, Sébastien Jaeger, Denis Thieffry, Morgane Thomas-Chollier, and Jacques Van Helden. RSAT matrix-clustering: dynamic exploration and redundancy reduction of transcription factor binding motif collections. *Nucleic Acids Research*, 45(13):e119–e119, 2017.

[46] Chen Chen, Jie Hou, Xiaowen Shi, Hua Yang, James A Birchler, and Jianlin Cheng. DeepGRN: prediction of transcription factor binding site across cell-types using attention-based deep neural networks. *BMC Bioinformatics*, 22(1):1–18, 2021.

[47] Gang Chen. A gentle tutorial of recurrent neural network with error backpropagation. *arXiv preprint arXiv:1610.02583*, 2016.

[48] Hebing Chen, Yao Tian, Wenjie Shu, Xiaochen Bo, and Shengqi Wang. Comprehensive identification and annotation of cell type-specific and ubiquitous CTCF-binding sites in the human genome. *PLOS ONE*, 7(7):e41374, 2012.

[49] Meilin Chen, Yijun Liu, Yuqin Yang, Yanbing Qiu, Zhicheng Wang, Xiaoxu Li, and Wenling Zhang. Emerging roles of activating transcription factor (ATF) family members in tumourigenesis and immunity: Implications in cancer immunotherapy. *Genes & Diseases*, 2021.

[50] Ying Chen, Nadia M Davidson, Yuk Kei Wan, Harshil Patel, Fei Yao, Hwee Meng Low, Christopher Hendra, Laura Watten, Andre Sim, Chelsea Sawyer, et al. A systematic benchmark of Nanopore long read RNA sequencing for transcript level analysis in human cell lines. *BioRxiv*, 2021.

[51] Yiwen Chen, Nicolas Negre, Qunhua Li, Joanna O Mieczkowska, Matthew Slattery, Tao Liu, Yong Zhang, Tae-Kyung Kim, Housheng Hansen He, Jennifer Zieba, et al. Systematic evaluation of factors influencing ChIP-seq fidelity. *Nature Methods*, 9(6):609–614, 2012.

[52] Justin G Chitpin, Aseel Awdeh, and Theodore J Perkins. RECAP reveals the true statistical significance of ChIP-seq peak calls. *Bioinformatics*, 35(19):3592–3598, 2019.

[53] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[54] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[55] ENCODE Project Consortium et al. The ENCODE (encyclopedia of DNA elements) project. *Science*, 306(5696):636–640, 2004.

[56] ENCODE Project Consortium et al. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57, 2012.

[57] Leighton J Core, Joshua J Waterfall, and John T Lis. Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. *Science*, 322(5909):1845–1848, 2008.

[58] Chi V Dang. MYC on the path to cancer. *Cell*, 149(1):22–35, 2012.

[59] Modan K Das and Ho-Kwok Dai. A survey of DNA motif finding algorithms. *BMC Bioinformatics*, 8(7):1–13, 2007.

[60] Sadaf Davudian, Behzad Mansoori, Neda Shajari, Ali Mohammadi, and Behzad Baradaran. BACH1, the master regulator gene: A novel candidate target for cancer therapy. *Gene*, 588(1):30–37, 2016.

[61] Fabienne De Graeve, Anne Bahr, Kanaga T Sabapathy, Charlotte Hauss, Erwin F Wagner, Claude Kedinger, and Bruno Chatton. Role of the ATFa/JNK2 complex in jun activation. *Oncogene*, 18(23):3491–3500, 1999.

[62] Roxanne E Debaugny and Jane A Skok. CTCF and CTCFL in cancer. *Current Opinion in Genetics & Development*, 61:44–52, 2020.

[63] Patrik D'haeseleer. How does DNA sequence motif discovery work? *Nature Biotechnology*, 24(8):959–961, 2006.

[64] Patrik D'haeseleer. What are DNA sequence motifs? *Nature Biotechnology*, 24(4):423–425, 2006.

[65] Aaron Diaz, Kiyoub Park, Daniel A Lim, and Jun S Song. Normalization, bias correction, and peak calling for ChIP-seq. *Statistical Applications in Genetics and Molecular Biology*, 11(3), 2012.

[66] Sarah Djebali, Carrie A Davis, Angelika Merkel, Alex Dobin, Timo Lassmann, Ali Mortazavi, Andrea Tanzer, Julien Lagarde, Wei Lin, Felix Schlesinger, et al. Landscape of transcription in human cells. *Nature*, 489(7414):101–108, 2012.

[67] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

[68] Mikhail G Dozmorov, Indra Adrianto, Cory B Giles, Edmund Glass, Stuart B Glenn, Courtney Montgomery, Kathy L Sivils, Lorin E Olson, Tomoaki Iwayama, Willard M Freeman, et al. Detrimental effects of duplicate reads and low complexity regions on RNA-and ChIP-seq data. In *BMC Bioinformatics*, volume 16, pages 1–11. BioMed Central, 2015.

[69] Gökcen Eraslan, Žiga Avsec, Julien Gagneur, and Fabian J Theis. Deep learning: new computational modelling techniques for genomics. *Nature Reviews Genetics*, 20(7):389–403, 2019.

[70] F Fan, MH Bashari, E Morelli, G Tonon, S Malvestiti, S Vallet, M Jarahian, A Seckinger, D Hose, L Bakiri, et al. The AP-1 transcription factor JUNB is essential for multiple myeloma cell proliferation and drug resistance in the bone marrow microenvironment. *Leukemia*, 31(7):1570–1581, 2017.

[71] Pau Farré, Alexandre Heurteau, Olivier Cuvier, and Eldon Emberly. Dense neural networks for predicting chromatin conformation. *BMC Bioinformatics*, 19(1):1–12, 2018.

[72] Anthony P Fejes, Gordon Robertson, Mikhail Bilenky, Richard Varhol, Matthew Bainbridge, and Steven JM Jones. FindPeaks 3.1: a tool for identifying areas of enrichment from massively parallel short-read sequencing technology. *Bioinformatics*, 24(15):1729–1730, 2008.

[73] Jianxing Feng, Tao Liu, Bo Qin, Yong Zhang, and Xiaole Shirley Liu. Identifying ChIP-seq enrichment using MACS. *Nature Protocols*, 7(9):1728–1740, 2012.

[74] Jianxing Feng, Tao Liu, and Yong Zhang. Using MACS to identify peaks from ChIP-Seq data. *Current Protocols in Bioinformatics*, 34(1):2–14, 2011.

[75] Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

[76] Theresa M Filtz, Walter K Vogel, and Mark Leid. Regulation of transcription factor activity by interconnected post-translational modifications. *Trends in Pharmacological Sciences*, 35(2):76–85, 2014.

[77] Barrett C Foat, Alexandre V Morozov, and Harmen J Bussemaker. Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, 22(14):e141–e149, 2006.

[78] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. pages 2950–2958, 2019.

[79] Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. pages 3429–3437, 2017.

[80] Oriol Fornes, Jaime A Castro-Mondragon, Aziz Khan, Robin Van der Lee, Xi Zhang, Phillip A Richmond, Bhavi P Modi, Solenne Correard, Marius Gheorghe, Damir Baranašić, et al. JASPAR 2020: update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 48(D1):D87–D92, 2020.

[81] Jason Gertz, Daniel Savic, Katherine E Varley, E Christopher Partridge, Alexias Safi, Preti Jain, Gregory M Cooper, Timothy E Reddy, Gregory E Crawford, and Richard M Myers. Distinct properties of cell-type-specific and shared transcription factor binding sites. *Molecular Cell*, 52(1):25–36, 2013.

[82] Haiyan Gong, Yi Yang, Sichen Zhang, Minghong Li, and Xiaotong Zhang. Application of Hi-C and other omics data analysis in human cancer and cell differentiation research. *Computational and Structural Biotechnology Journal*, 2021.

[83] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[84] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333–351, 2016.

[85] Malgorzata Gozdecka and Wolfgang Breitwieser. The roles of ATF2 (activating transcription factor 2) in tumorigenesis. *Biochemical Society Transactions*, 40(1):230–234, 2012.

[86] Charles E Grant, Timothy L Bailey, and William Stafford Noble. FIMO: scanning for occurrences of a given motif. *Bioinformatics*, 27(7):1017–1018, 2011.

[87] Cristian Groza, Tony Kwan, Nicole Soranzo, Tomi Pastinen, and Guillaume Bourque. Personalized and graph genomes reveal missing signal in epigenomic data. *Genome Biology*, 21(1):1–22, 2020.

[88] Xiaodong Guo, Mei Yang, Hao Gu, Jingmin Zhao, and Lin Zou. Decreased expression of SOX6 confers a poor prognosis in hepatocellular carcinoma. *Cancer Epidemiology*, 37(5):732–736, 2013.

[89] Nidhi Gupta and Vijay K Verma. Next-generation sequencing and its application: empowering in public health beyond reality. In *Microbial Technology for the Welfare of Society*, pages 313–341. Springer, 2019.

[90] Shobhit Gupta, John A Stamatoyannopoulos, Timothy L Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome Biology*, 8(2):1–9, 2007.

[91] Arif Harmanci, Joel Rozowsky, and Mark Gerstein. MUSIC: identification of enriched regions in ChIP-Seq experiments using a mappability-corrected multiscale signal processing framework. *Genome Biology*, 15(10):1–15, 2014.

[92] Hamid Reza Hassanzadeh and May D Wang. DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 178–183. IEEE, 2016.

[93] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C Lin, Peter Laslo, Jason X Cheng, Cornelis Murre, Harinder Singh, and Christopher K Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell*, 38(4):576–589, 2010.

[94] Sven Heinz, Casey E Romanoski, Christopher Benner, and Christopher K Glass. The selection and function of cell type-specific enhancers. *Nature Reviews Molecular Cell Biology*, 16(3):144–154, 2015.

[95] Jorja G Henikoff, Jason A Belsky, Kristina Krassovsky, David M MacAlpine, and Steven Henikoff. Epigenome characterization at single base-pair resolution. *Proceedings of the National Academy of Sciences*, 108(45):18318–18323, 2011.

[96] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[97] Naozumi Hiranuma, Scott Lundberg, and Su-In Lee. CloudControl: Leveraging many public ChIP-seq control experiments to better remove background noise. In *Proceedings of the 7th ACM international conference on bioinformatics, computational biology, and health informatics*, pages 191–199, 2016.

[98] Naozumi Hiranuma, Scott M Lundberg, and Su-In Lee. AIControl: replacing matched control experiments with machine learning improves ChIP-seq peak identification. *Nucleic Acids Research*, 47(10):e58–e58, 2019.

[99] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[100] Sjoerd Johannes Bastiaan Holwerda and Wouter de Laat. CTCF: the protein, the binding partners, the binding sites and their chromatin loops. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 368(1620):20120369, 2013.

[101] Tien Hsu, Maria Trojanowska, and Dennis K Watson. ETS proteins in biological control and cancer. *Journal of Cellular Biochemistry*, 91(5):896–903, 2004.

[102] Yu Hu, Li Fang, Xuelian Chen, Jiang F Zhong, Mingyao Li, and Kai Wang. LIQA: long-read isoform quantification and analysis. *Genome Biology*, 22(1):1–21, 2021.

[103] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[104] Sachi Inukai, Kian Hong Kock, and Martha L Bulyk. Transcription factor–DNA binding: beyond binding site motifs. *Current Opinion in Genetics & Development*, 43:110–119, 2017.

[105] Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, 150:228–234, 2021.

[106] Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F McRae, Siavash Fazel Darbandi, David Knowles, Yang I Li, Jack A Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B Schwartz, et al. Predicting splicing from primary sequence with deep learning. *Cell*, 176(3):535–548, 2019.

[107] Nathalie Japkowicz. Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, Algorithms, and Applications*, pages 187–206, 2013.

[108] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[109] Hyeongrin Jeon, Hyunji Lee, Byunghee Kang, Insoon Jang, and Tae-Young Roh. Comparative analysis of commonly used peak calling programs for ChIP-Seq analysis. *Genomics & Informatics*, 18(4), 2020.

[110] Weiliang Jiang, Qiongying Yuan, Yuanye Jiang, Li Huang, Congying Chen, Guoyong Hu, Rong Wan, Xingpeng Wang, and Lijuan Yang. Identification of SOX6 as a regulator of pancreatic cancer development. *Journal of Cellular and Molecular Medicine*, 22(3):1864–1872, 2018.

[111] Xiaoyan Jiang, Hui Xie, Yingyu Dou, Jing Yuan, Da Zeng, and Songshu Xiao. Expression and function of FRA1 protein in tumors. *Molecular Biology Reports*, 47(1):737–752, 2020.

[112] David S Johnson, Ali Mortazavi, Richard M Myers, and Barbara Wold. Genome-wide mapping of in vivo protein-DNA interactions. *Science*, 316(5830):1497–1502, 2007.

[113] Arttu Jolma, Teemu Kivioja, Jarkko Toivonen, Lu Cheng, Gonghong Wei, Martin Enge, Mikko Taipale, Juan M Vaquerizas, Jian Yan, Mikko J Sillanpää, et al. Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Research*, 20(6):861–873, 2010.

[114] Arttu Jolma, Jian Yan, Thomas Whitington, Jarkko Toivonen, Kazuhiro R Nitta, Pasi Rastas, Ekaterina Morgunova, Martin Enge, Mikko Taipale, Gonghong Wei, et al. DNA-binding specificities of human transcription factors. *Cell*, 152(1-2):327–339, 2013.

[115] Raja Jothi, Suresh Cuddapah, Artem Barski, Kairong Cui, and Keji Zhao. Genome-wide identification of in vivo protein-DNA binding sites from ChIP-Seq data. *Nucleic Acids Research*, 36(16):5221–5231, 2008.

[116] Eric Julien and Winship Herr. Proteolytic processing is necessary to separate and ensure proper cell growth and cytokinesis functions of HCF-1. *The EMBO Journal*, 22(10):2360–2369, 2003.

[117] Mehran Karimzadeh, Carl Ernst, Anshul Kundaje, and Michael M Hoffman. Umap and Bismap: quantifying genome and methylome mappability. *Nucleic Acids Research*, 46(20):e120–e120, 2018.

[118] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[119] Hatice S Kaya-Okur, Steven J Wu, Christine A Codomo, Erica S Pledger, Terri D Bryson, Jorja G Henikoff, Kami Ahmad, and Steven Henikoff. CUT&Tag for efficient epigenomic profiling of small samples and single cells. *Nature Communications*, 10(1):1–10, 2019.

[120] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Research*, 26(7):990–999, 2016.

[121] Bharat Khurana and Thomas M Kristie. A protein sequestering system reveals control of cellular programs by the transcriptional coactivator HCF-1. *Journal of Biological Chemistry*, 279(32):33673–33683, 2004.

[122] Somi Kim, Nam-Kyung Yu, and Bong-Kiun Kaang. CTCF as a multifunctional protein in genome regulation and gene expression. *Experimental & Molecular Medicine*, 47(6):e166–e166, 2015.

[123] Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. pages 267–280. Springer, 2019.

[124] Peter K Koo and Sean R Eddy. Representation learning of genomic sequence motifs with convolutional neural networks. *PLOS Computational Biology*, 15(12):e1007560, 2019.

[125] Peter K Koo, Antonio Majdandzic, Matthew Ploenzke, Praveen Anand, and Steffan B Paul. Global importance analysis: An interpretability method to quantify importance of genomic features in deep neural networks. *PLOS Computational Biology*, 17(5):e1008925, 2021.

[126] Peter K Koo and Matt Ploenzke. Deep learning for inferring transcription factor binding sites. *Current Opinion in Systems Biology*, 19:16–23, 2020.

[127] Peter K Koo and Matt Ploenzke. Improving representations of genomic sequence motifs in convolutional networks with exponential activations. *Nature Machine Intelligence*, 3(3):258–266, 2021.

[128] Lefteris Koumakis. Deep learning models in genomics; are we there yet? *Computational and Structural Biotechnology Journal*, 18:1466–1473, 2020.

[129] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[130] Meghana Kshirsagar, Han Yuan, Juan Lavista Ferres, and Christina S Leslie. Dirichlet variational autoencoders for de novo motif discovery from accessible chromatin. *bioRxiv*, 2021.

[131] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[132] Ivan V Kulakovskiy, Ilya E Vorontsov, Ivan S Yevshin, Ruslan N Sharipov, Alla D Fedorova, Eugene I Rumynskiy, Yulia A Medvedeva, Arturo Magana-Mora, Vladimir B Bajic, Dmitry A Papatsenko, et al. HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-seq analysis. *Nucleic Acids Research*, 46(D1):D252–D259, 2018.

[133] Teemu D Laajala, Sunil Raghav, Soile Tuomela, Riitta Lahesmaa, Tero Aittokallio, and Laura L Elo. A practical comparison of methods for detecting transcription factor binding sites in ChIP-seq experiments. *BMC Genomics*, 10(1):1–15, 2009.

[134] Mong-Hsun Tsai Lai. Common applications of next-generation sequencing technologies in genomic research. 2013.

[135] Stephen G Landt, Georgi K Marinov, Anshul Kundaje, Pouya Kheradpour, Florencia Pauli, Serafim Batzoglou, Bradley E Bernstein, Peter Bickel, James B Brown, Philip Cayting, et al. ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Research*, 22(9):1813–1831, 2012.

[136] Ben Langmead. Aligning short sequencing reads with Bowtie. *Current Protocols in Bioinformatics*, 32(1):11–7, 2010.

[137] Charles E Lawrence, Stephen F Altschul, Mark S Boguski, Jun S Liu, Andrew F Neuwald, and John C Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[138] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[139] Bum-Kyu Lee, Akshay A Bhinge, Anna Battenhouse, Ryan M McDaniell, Zheng Liu, Lingyun Song, Yunyun Ni, Ewan Birney, Jason D Lieb, Terrence S Furey, et al. Cell-type specific and combinatorial usage of diverse transcription factors revealed by genome-wide binding studies in multiple human cells. *Genome Research*, 22(1):9–24, 2012.

[140] Tong Ihn Lee and Richard A Young. Transcriptional regulation and its misregulation in disease. *Cell*, 152(6):1237–1251, 2013.

[141] Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[142] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 2009.

[143] Hongyang Li and Yuanfang Guan. Fast decoding cell type–specific transcription factor binding landscape at single-nucleotide resolution. *Genome Research*, 31(4):721–731, 2021.

[144] Hongyang Li, Daniel Quang, and Yuanfang Guan. Anchor: trans-cell type prediction of transcription factor binding sites. *Genome Research*, 29(2):281–292, 2019.

[145] Qunhua Li, James B Brown, Haiyan Huang, and Peter J Bickel. Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics*, 5(3):1752–1779, 2011.

[146] Kun Liang and Sündüz Keleş. Normalization of ChIP-seq data with control. *BMC Bioinformatics*, 13(1):1–10, 2012.

[147] A Javier Lopez. Developmental role of transcription factor isoforms generated by alternative splicing. *Developmental Biology*, 172(2):396–411, 1995.

[148] Zakaria Louadi, Mhaned Oubounyt, Hilal Tayara, and Kil To Chong. Deep splicing code: Classifying alternative splicing events using deep learning. *Genes*, 10(8):587, 2019.

[149] M. I Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):550, 2014.

[150] Marina Lowen, Gail Scott, and Patty Zwollo. Functional analyses of two alternative isoforms of the transcription factor Pax-5. *Journal of Biological Chemistry*, 276(45):42565–42574, 2001.

[151] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[152] Wenxiu Ma and Wing Hung Wong. The analysis of ChIP-Seq data. In *Methods in enzymology*, volume 497, pages 51–73. Elsevier, 2011.

[153] Kenzie D MacIsaac and Ernest Fraenkel. Practical strategies for discovering regulatory DNA sequence motifs. *PLOS Computational Biology*, 2(4):e36, 2006.

[154] Toshio Maekawa, Seungjoon Kim, Daisuke Nakai, Chieko Makino, Tsuyoshi Takagi, Hiroo Ogura, Kazuyuki Yamada, Bruno Chatton, and Shunsuke Ishii. Social isolation stress induces ATF-7 phosphorylation and impairs silencing of the 5-HT 5B receptor gene. *The EMBO Journal*, 29(1):196–208, 2010.

[155] Elaine R Mardis. ChIP-seq: welcome to the new frontier. *Nature Methods*, 4(8):613–614, 2007.

[156] Georgi K Marinov, Anshul Kundaje, Peter J Park, and Barbara J Wold. Large-scale quality analysis of published ChIP-seq data. *G3: Genes, Genomes, Genetics*, 4(2):209–223, 2014.

[157] Alexandra Maslova, Ricardo N Ramirez, Ke Ma, Hugo Schmutz, Chendi Wang, Curtis Fox, Bernard Ng, Christophe Benoist, Sara Mostafavi, et al. Deep learning of immune cell differentiation. *Proceedings of the National Academy of Sciences*, 117(41):25655–25666, 2020.

[158] Volker Matys, Olga V Kel-Margoulis, Ellen Fricke, Ines Liebich, Sigrid Land, A Barre-Dirrie, Ingmar Reuter, D Chekmenev, Mathias Krull, Klaus Hornischer, et al. TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. *Nucleic Acids Research*, 34(suppl_1):D108–D110, 2006.

[159] Matthew T Maurano, Richard Humbert, Eric Rynes, Robert E Thurman, Eric Haugen, Hao Wang, Alex P Reynolds, Richard Sandstrom, Hongzhu Qu, Jennifer Brody, et al. Systematic localization of common disease-associated variation in regulatory DNA. *Science*, 337(6099):1190–1195, 2012.

[160] B Meera Krishna, Munawwar Ali Khan, and Shams Tabrez Khan. Next-generation sequencing (NGS) platforms: an exciting era of genome sequence analysis. In *Microbial Genomics in Sustainable Agroecosystems*, pages 89–109. Springer, 2019.

[161] Bartolomeus J Meijer, Francesca P Giugliano, Bart Baan, Jonathan HM van der Meer, Sander Meisner, Manon van Roest, Pim J Koelink, Ruben J de Boer, Nic Jones, Wolfgang Breitwieser, et al. ATF2 and ATF7 are critical mediators of intestinal epithelial repair. *Cellular and Molecular Gastroenterology and Hepatology*, 10(1):23–42, 2020.

[162] Clifford A Meyer and X Shirley Liu. Identifying and mitigating bias in next-generation sequencing methods for chromatin biology. *Nature Reviews Genetics*, 15(11):709–721, 2014.

[163] Mariann Micsinai, Fabio Parisi, Francesco Strino, Patrik Asp, Brian D Dynlacht, and Yuval Kluger. Picking ChIP-seq peak detectors for analyzing chromatin modification experiments. *Nucleic Acids Research*, 40(9):e70–e70, 2012.

[164] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, 5(7):621–628, 2008.

[165] Surag Nair, Daniel S Kim, Jacob Perricone, and Anshul Kundaje. Integrating regulatory DNA sequence and gene expression to predict genome-wide chromatin accessibility across cellular contexts. *Bioinformatics*, 35(14):i108–i116, 2019.

[166] Ryuichiro Nakato and Katsuhiko Shirahige. Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation. *Briefings in Bioinformatics*, 18(2):279–290, 2017.

[167] Shane Neph, Jeff Vierstra, Andrew B Stergachis, Alex P Reynolds, Eric Haugen, Benjamin Vernot, Robert E Thurman, Sam John, Richard Sandstrom, Audra K Johnson, et al. An expansive human regulatory lexicon encoded in transcription factor footprints. *Nature*, 489(7414):83–90, 2012.

[168] Yumin Nie, Chuanjun Shu, and Xiao Sun. Cooperative binding of transcription factors in the human genome. *Genomics*, 112(5):3427–3434, 2020.

[169] Ian E Nielsen, Dimah Dera, Ghulam Rasool, Nidhal Bouaynaya, and Ravi P Ramachandran. Robust explainability: A tutorial on gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:2107.11400*, 2021.

[170] Gherman Novakovsky, Manu Saraswat, Oriol Fornes, Sara Mostafavi, and Wyeth W Wasserman. Biologically relevant transfer learning improves transcription factor binding prediction. *Genome Biology*, 22(1):1–25, 2021.

[171] Emilio Soria Olivas, Jos David Mart Guerrero, Marcelino Martinez-Sober, Jose Rafael Magdalena-Benedito, L Serrano, et al. *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI Global, 2009.

[172] Tobias H Olsen, Iain H Moal, and Charlotte M Deane. AbLang: an antibody language model for completing antibody sequences. *Bioinformatics Advances*, 2(1), 06 2022. vbac046.

[173] Yaron Orenstein and Ron Shamir. A comparative analysis of transcription factor binding models learned from PBM, HT-SELEX and ChIP data. *Nucleic Acids Research*, 42(8):e63–e63, 2014.

[174] Yaron Orenstein and Ron Shamir. Modeling protein–DNA binding via high-throughput in vitro technologies. *Briefings in Functional Genomics*, 16(3):171–180, 2017.

[175] Rene G Ott, Olivia Simma, Karoline Kollmann, Eva Weisz, Eva-Maria Zebedin, Marina Schorpp-Kistner, Gerwin Heller, S Zöchbauer, Erwin F Wagner, Michael Freissmuth, et al. JUNB is a gatekeeper for B-lymphoid leukemia. *Oncogene*, 26(33):4863–4871, 2007.

[176] Stephan Pabinger, Andreas Dander, Maria Fischer, Rene Snajder, Michael Sperk, Mirjana Efremova, Birgit Krabichler, Michael R Speicher, Johannes Zschocke, and Zlatko Trajanoski. A survey of tools for variant analysis of next-generation genome sequencing data. *Briefings in Bioinformatics*, 15(2):256–278, 2014.

[177] Joselyn Padilla and Jiyoung Lee. A novel therapeutic target, BACH1, regulates cancer metabolism. *Cells*, 10(3):634, 2021.

[178] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[179] Peter J Park. ChIP-seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10):669–680, 2009.

[180] Sungjoon Park, Yookyung Koh, Hwisang Jeon, Hyunjae Kim, Yoonsun Yeo, and Jaewoo Kang. Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Scientific Reports*, 10(1):1–10, 2020.

[181] Emmanuelle Passegué, Wolfram Jochum, Marina Schorpp-Kistner, Uta Möhle-Steinlein, and Erwin F Wagner. Chronic myeloid leukemia with increased granulocyte progenitors in mice lacking JUNB expression in the myeloid lineage. *Cell*, 104(1):21–32, 2001.

[182] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, 2017.

[183] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[184] Giulio Pavesi, Giancarlo Mauri, and Graziano Pesole. An algorithm for finding signals of unknown length in DNA sequences. *Bioinformatics*, 17(suppl_1):S207–S214, 2001.

[185] Dov A Pechenick, Joshua L Payne, and Jason H Moore. Phenotypic robustness and the assortativity signature of human transcription factor networks. *PLOS Computational Biology*, 10(8):e1003780, 2014.

[186] Alexander Peltzer, Günter Jäger, Alexander Herbig, Alexander Seitz, Christian Kniep, Johannes Krause, and Kay Nieselt. EAGER: efficient ancient genome reconstruction. *Genome Biology*, 17(1):1–14, 2016.

[187] Len A Pennacchio, Wendy Bickmore, Ann Dean, Marcelo A Nobrega, and Gill Bejerano. Enhancers: five essential questions. *Nature Reviews Genetics*, 14(4):288–295, 2013.

[188] Elizabeth Pennisi. Most complete human genome yet is revealed. *Science*, 376(6588):15–16, 2022.

[189] Shirley Pepke, Barbara Wold, and Ali Mortazavi. Computation for ChIP-seq and RNA-seq studies. *Nature Methods*, 6(11):S22–S32, 2009.

[190] Pavel A Pevzner, Sing-Hoi Sze, et al. Combinatorial approaches to finding subtle signals in DNA sequences. In *ISMB*, volume 8, pages 269–278, 2000.

[191] Theresa Phillips. Regulation of transcription and gene expression in eukaryotes. *Nature Education*, 1(1):199, 2008.

[192] Mike Phuycharoen, Peyman Zarrineh, Laure Bridoux, Shilu Amin, Marta Losa, Ke Chen, Nicoletta Bobola, and Magnus Rattray. Uncovering tissue-specific binding features from differential deep learning. *Nucleic Acids Research*, 48(5):e27–e27, 2020.

[193] Yitzhak Pilpel, Priya Sudarsanam, and George M Church. Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, 29(2):153–159, 2001.

[194] Ryan Poplin, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan Newburger, Jojo Dijamco, Nam Nguyen, Pegah T Afshar, et al. A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology*, 36(10):983–987, 2018.

[195] Damiano Porcelli, Bettina Fischer, Steven Russell, and Robert White. Chromatin accessibility plays a key role in selective targeting of HOX proteins. *Genome Biology*, 20(1):1–19, 2019.

[196] Qian Qin and Jianxing Feng. Imputation for transcription factor binding predictions based on deep learning. *PLOS Computational Biology*, 13(2):e1005403, 2017.

[197] Daniel Quang and Xiaohui Xie. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Research*, 44(11):e107–e107, 2016.

[198] Daniel Quang and Xiaohui Xie. FactorNet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 166:40–47, 2019.

[199] Aaron R Quinlan and Ira M Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[200] Parameswaran Ramachandran, Gareth A Palidwor, and Theodore J Perkins. BIDCHIPS: bias decomposition and removal from ChIP-seq data clarifies true binding signal and its functional correlates. *Epigenetics & Chromatin*, 8(1):1–16, 2015.

[201] Satyanarayan Rao, Kami Ahmad, and Srinivas Ramachandran. Cooperative binding between distant transcription factors is a hallmark of active enhancers. *Molecular Cell*, 81(8):1651–1665, 2021.

[202] S Rauschert, K Raubenheimer, PE Melton, and RC Huang. Machine learning and clinical epigenetics: a review of challenges for diagnosis and classification. *Clinical Epigenetics*, 12(1):1–11, 2020.

[203] Sekhar PM Reddy and Brooke T Mossman. Role and regulation of activator protein-1 in toxicant-induced responses of the lung. *American Journal of Physiology-Lung Cellular and Molecular Physiology*, 283(6):L1161–L1178, 2002.

[204] Attila Reményi, Hans R Schöler, and Matthias Wilmanns. Combinatorial control of gene expression. *Nature Structural & Molecular Biology*, 11(9):812–815, 2004.

[205] Bing Ren, François Robert, John J Wyrick, Oscar Aparicio, Ezra G Jennings, Itamar Simon, Julia Zeitlinger, Jorg Schreiber, Nancy Hannett, Elenita Kanin, et al. Genome-wide location and function of DNA binding proteins. *Science*, 290(5500):2306–2309, 2000.

[206] Ho Sung Rhee and B Franklin Pugh. Comprehensive genome-wide protein-DNA interactions detected at single-nucleotide resolution. *Cell*, 147(6):1408–1419, 2011.

[207] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. pages 1135–1144, 2016.

[208] Katherine W Rogers and Alexander F Schier. Morphogen gradients: from generation to interpretation. *Annual Review of Cell and Developmental Biology*, 27(1):377–407, 2011.

[209] M Rohini, A Haritha Menon, and N Selvamurugan. Role of activating transcription factor 3 and its interacting proteins under physiological and pathological conditions. *International Journal of Biological Macromolecules*, 120:310–317, 2018.

[210] Remo Rohs, Xiangshu Jin, Sean M West, Rohit Joshi, Barry Honig, and Richard S Mann. Origins of specificity in protein-DNA recognition. *Annual Review of Biochemistry*, 79:233, 2010.

[211] Ribana Roscher, Bastian Bohn, Marco F Duarte, and Jochen Garcke. Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216, 2020.

[212] Ananda L Roy, Elizabeth L Wilder, and James M Anderson. Validation of antibodies: Lessons learned from the common fund protein capture reagents program. *Science Advances*, 7(46):eabl7148, 2021.

[213] Joel Rozowsky, Ghia Euskirchen, Raymond K Auerbach, Zhengdong D Zhang, Theodore Gibson, Robert Bjornson, Nicholas Carriero, Michael Snyder, and Mark B Gerstein. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature Biotechnology*, 27(1):66, 2009.

[214] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[215] Zachary Maas Robin Dowell Rutendo Sigauke, Lynn Sanford. Regulatory network inference using nascent RNA sequencing data. Madison, Wisconsin. ISMB 2022 Poster Session.

[216] Theodore Sakellaropoulos, Konstantinos Vougas, Sonali Narang, Filippos Koinis, Athanassios Kotsinas, Alexander Polyzos, Tyler J Moss, Sarina Piha-Paul, Hua Zhou, Eleni Kardala, et al. A deep learning framework for predicting response to therapy in cancer. *Cell Reports*, 29(11):3367–3373, 2019.

[217] Albin Sandelin, Wynand Alkema, Pär Engström, Wyeth W Wasserman, and Boris Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Research*, 32(suppl_1):D91–D94, 2004.

[218] Thomas D Schneider and R Michael Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, 1990.

[219] M. Schuster and K.K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[220] Eilon Sharon, Shai Lubliner, and Eran Segal. A feature-based approach to modeling protein–DNA interactions. *PLOS Computational Biology*, 4(8):e1000154, 2008.

[221] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. pages 3145–3153. PMLR, 2017.

[222] Rahul Siddharthan. Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PLOS ONE*, 5(3):e9722, 2010.

[223] Trevor Siggers and Raluca Gordân. Protein–DNA binding: complexities and multi-protein codes. *Nucleic Acids Research*, 42(4):2099–2111, 2014.

[224] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[225] Ritambhara Singh, Jack Lanchantin, Gabriel Robins, and Yanjun Qi. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics*, 32(17):i639–i648, 2016.

[226] Leon Sixt, Maximilian Granz, and Tim Landgraf. When explanations lie: why modified BP attribution fails. *CoRR abs/1912.09818*, 2019.

[227] Gina M Sizemore, Jason R Pitarresi, Subhasree Balakrishnan, and Michael C Ostrowski. The ETS family of oncogenic transcription factors in solid tumours. *Nature Reviews Cancer*, 17(6):337–351, 2017.

[228] Barton E Slatko, Andrew F Gardner, and Frederick M Ausubel. Overview of next-generation sequencing technologies. *Current Protocols in Molecular Biology*, 122(1):e59, 2018.

[229] Lingyun Song and Gregory E Crawford. DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells. *Cold Spring Harbor Protocols*, 2010(2):pdb–prot5384, 2010.

[230] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[231] Divyanshi Srivastava, Begüm Aydin, Esteban O Mazzoni, and Shaun Mahony. An interpretable bimodal neural network characterizes the sequence and preexisting chromatin predictors of induced transcription factor binding. *Genome Biology*, 22(1):1–25, 2021.

[232] Divyanshi Srivastava and Shaun Mahony. Sequence and chromatin determinants of transcription factor binding and the establishment of cell type-specific binding patterns. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1863(6):194443, 2020.

[233] Lincoln D Stein. An introduction to the informatics of "Next-Generation" Sequencing. *Current Protocols in Bioinformatics*, 36(1):11–1, 2011.

[234] Zachary E Stine, Zandra E Walton, Brian J Altman, Annie L Hsieh, and Chi V Dang. MYC, metabolism, and cancer. *Cancer Discovery*, 5(10):1024–1039, 2015.

[235] Gary D Stormo. DNA binding sites: representation and discovery. *Bioinformatics*, 16(1):16–23, 2000.

[236] Gary D Stormo. Modeling the specificity of protein-DNA interactions. *Quantitative Biology*, 1(2):115, 2013.

[237] Gary D Stormo and Yue Zhao. Determining the specificity of protein–DNA interactions. *Nature Reviews Genetics*, 11(11):751–760, 2010.

[238] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. pages 3319–3328. PMLR, 2017.

[239] Adam M Szalkowski and Christoph D Schmid. Rapid innovation in ChIP-seq peak-calling algorithms is outdistancing benchmarking efforts. *Briefings in Bioinformatics*, 12(6):626–633, 2011.

[240] Agnieszka P Szremska, Lukas Kenner, Eva Weisz, Rene G Ott, Emmanuelle Passegué, Michaela Artwohl, Michael Freissmuth, Renate Stoxreiter, Hans-Christian Theussl, Sabina Baumgartner Parzer, et al. JUNB inhibits proliferation and transformation in B-lymphoid cells. *Blood*, 102(12):4159–4165, 2003.

[241] Amlan Talukder, Clayton Barham, Xiaoman Li, and Haiyan Hu. Interpretation of deep learning in genomics and epigenomics. *Briefings in Bioinformatics*, 22(3):bbaa177, 2021.

[242] Mingxiang Teng and Rafael A Irizarry. Accounting for GC-content bias reduces systematic errors and batch effects in ChIP-seq data. *Genome Research*, 27(11):1930–1938, 2017.

[243] Reuben Thomas, Sean Thomas, Alisha K Holloway, and Katherine S Pollard. Features that define the best ChIP-seq peak calling algorithms. *Briefings in Bioinformatics*, 18(3):441–450, 2017.

[244] Shulan Tian, Shuxia Peng, Michael Kalmbach, Krutika S Gaonkar, Aditya Bhagwate, Wei Ding, Jeanette Eckel-Passow, Huihuang Yan, and Susan L Slager. Identification of factors associated with duplicate rate in ChIP-seq data. *PLOS ONE*, 14(4):e0214723, 2019.

[245] Ameni Trabelsi, Mohamed Chaabane, and Asa Ben-Hur. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. *Bioinformatics*, 35(14):i269–i277, 2019.

[246] Shivtia Trop-Steinberg and Yehudit Azar. AP-1 expression and its clinical relevance in immune disorders and cancer. *The American Journal of the Medical Sciences*, 353(5):474–483, 2017.

[247] Anton Valouev, David S Johnson, Andreas Sundquist, Catherine Medina, Elizabeth Anton, Serafim Batzoglou, Richard M Myers, and Arend Sidow. Genome-wide analysis of transcription factor binding sites based on ChIP-Seq data. *Nature Methods*, 5(9):829–834, 2008.

[248] Anand Venkataraman, Kun Yang, Jose Irizarry, Mark Mackiewicz, Paolo Mita, Zheng Kuang, Lin Xue, Devlina Ghosh, Shuang Liu, Pedro Ramos, et al. A toolbox of immunoprecipitation-grade monoclonal antibodies to human transcription factors. *Nature Methods*, 15(5):330–338, 2018.

[249] Jeff Vierstra, John Lazar, Richard Sandstrom, Jessica Halow, Kristen Lee, Daniel Bates, Morgan Diegel, Douglas Dunn, Fidencio Neri, Eric Haugen, et al. Global reference mapping of human transcription factor footprints. *Nature*, 583(7818):729–736, 2020.

[250] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[251] Kanchan Vishnoi, Navin Viswakarma, Ajay Rana, and Basabi Rana. Transcription factors in cancer development and therapy. *Cancers*, 12(8):2296, 2020.

[252] Jie Wang, Jiali Zhuang, Sowmya Iyer, XinYing Lin, Troy W Whitfield, Melissa C Greven, Brian G Pierce, Xianjun Dong, Anshul Kundaje, Yong Cheng, et al. Sequence features and chromatin structure around the genomic regions bound by 119 human transcription factors. *Genome Research*, 22(9):1798–1812, 2012.

[253] Meng Wang, Cheng Tai, Weinan E, and Liping Wei. DeFine: deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants. *Nucleic Acids Research*, 46(11):e69–e69, 2018.

[254] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[255] Matthew T Weirauch, Atina Cote, Raquel Norel, Matti Annala, Yue Zhao, Todd R Riley, Julio Saez-Rodriguez, Thomas Cokelaer, Anastasia Vedenko, Shaheynoor

Talukder, et al. Evaluation of methods for modeling transcription factor sequence specificity. *Nature Biotechnology*, 31(2):126–134, 2013.

[256] Matthew T Weirauch, Ally Yang, Mihai Albu, Atina G Cote, Alejandro Montenegro-Montero, Philipp Drewe, Hamed S Najafabadi, Samuel A Lambert, Ishminder Mann, Kate Cook, et al. Determination and inference of eukaryotic transcription factor sequence specificity. *Cell*, 158(6):1431–1443, 2014.

[257] Sean Whalen, Jacob Schreiber, William S Noble, and Katherine S Pollard. Navigating the pitfalls of applying machine learning in genomics. *Nature Reviews Genetics*, pages 1–13, 2021.

[258] Christian Widmer and Gunnar Rätsch. Multitask learning in computational biology. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 207–216. JMLR Workshop and Conference Proceedings, 2012.

[259] Elizabeth G Wilbanks and Marc T Facciotti. Evaluation of algorithm performance in ChIP-seq peak detection. *PLOS ONE*, 5(7):e11471, 2010.

[260] Erin M Wissink, Anniina Vihervaara, Nathaniel D Tippens, and John T Lis. Nascent RNA analyses: tracking transcription and its regulation. *Nature Reviews Genetics*, 20(12):705–723, 2019.

[261] Kamil Wnuk, Jeremi Sudol, Kevin B Givechian, Patrick Soon-Shiong, Shahrooz Rabizadeh, Christopher Szeto, and Charles Vaske. Deep learning implicitly handles tissue specific phenomena to predict tumor DNA accessibility and immune activity. *iScience*, 20:119–136, 2019.

[262] Zuoqiao Wu, Mary Nicoll, and Robert J Ingham. AP-1 family transcription factors: a diverse family of proteins that regulate varied cellular activities in classical hodgkin lymphoma and ALK+ ALCL. *Experimental Hematology & Oncology*, 10(1):1–12, 2021.

[263] Phillip Wulfridge, Ben Langmead, Andrew P Feinberg, and Kasper D Hansen. Analyzing whole genome bisulfite sequencing data from highly divergent genotypes. *Nucleic Acids Research*, 47(19):e117–e117, 2019.

[264] Joanna Wysocka, Patrick T Reilly, and Winship Herr. Loss of HCF-1–chromatin association precedes temperature-induced growth arrest of tsBN67 cells. *Molecular and Cellular Biology*, 21(11):3820–3829, 2001.

[265] Xuhua Xia. Position weight matrix, Gibbs sampler, and the associated significance tests in motif characterization and prediction. *Scientifica*, 2012, 2012.

[266] Jinhua Xu, Yinghua Chen, and Olufunmilayo I Olopade. MYC and breast cancer. *Genes & Cancer*, 1(6):629–640, 2010.

[267] Mingcong Xu, Xuefeng Bai, Bo Ai, Guorui Zhang, Chao Song, Jun Zhao, Yuezhu Wang, Ling Wei, Fengcui Qian, Yanyu Li, et al. TF-Marker: a comprehensive manually curated database for transcription factors and related markers in specific cell and tissue types in human. *Nucleic Acids Research*, 50(D1):D402–D412, 2022.

[268] Shiliyang Xu, Sean Grullon, Kai Ge, and Weiqun Peng. Spatial clustering for identification of chip-enriched regions (SICER) to map regions of histone methylation patterns in embryonic stem cells. In *Stem Cell Transcriptional Networks*, pages 97–111. Springer, 2014.

[269] Jinyu Yang, Anjun Ma, Adam D Hoppe, Cankun Wang, Yang Li, Chi Zhang, Yan Wang, Bingqiang Liu, and Qin Ma. Prediction of regulatory motifs from human chip-sequencing data using a deep learning framework. *Nucleic Acids Research*, 47(15):7809–7824, 2019.

[270] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[271] Tianwei Yue and Haohan Wang. Deep learning for genomics: A concise overview. *arXiv preprint arXiv:1802.00810*, 2018.

[272] Chongzhi Zang, Dustin E Schones, Chen Zeng, Kairong Cui, Keji Zhao, and Weiqun Peng. A clustering approach for identification of enriched domains from histone modification ChIP-Seq data. *Bioinformatics*, 25(15):1952–1958, 2009.

[273] Kenneth S Zaret and Jason S Carroll. Pioneer transcription factors: establishing competence for gene expression. *Genes & Development*, 25(21):2227–2241, 2011.

[274] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[275] Assaf Zemach, Ivy E McDaniel, Pedro Silva, and Daniel Zilberman. Genome-wide evolutionary analysis of eukaryotic DNA methylation. *Science*, 328(5980):916–919, 2010.

[276] Haoyang Zeng, Matthew D Edwards, Ge Liu, and David K Gifford. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

[277] Shuangquan Zhang, Anjun Ma, Jing Zhao, Dong Xu, Qin Ma, and Yan Wang. Assessing deep learning methods in cis-regulatory motif finding based on genomic sequencing data. *Briefings in Bioinformatics*, 23(1):bbab374, 2022.

[278] Shuchen Zhang, Emma Bell, Huihan Zhi, Sarah Brown, Siti AM Imran, Véronique Azuara, and Wei Cui. OCT4 and PAX6 determine the dual function of SOX2 in human ESCs as a key pluripotent or neural factor. *Stem Cell Research & Therapy*, 10(1):1–14, 2019.

[279] Xinyue Zhang, Jieyu Guo, Xiangxiang Wei, Cong Niu, Mengping Jia, Qinhan Li, and Dan Meng. BACH1: function, regulation, and involvement in disease. *Oxidative Medicine and Cellular Longevity*, 2018, 2018.

[280] Yong Zhang, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, et al. Model-based analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):1–9, 2008.

[281] Nanxiang Zhao and Alan P Boyle. F-Seq2: improving the feature density based peak caller with dynamic statistics. *NAR Genomics and Bioinformatics*, 3(1):lqab012, 2021.

[282] An Zheng, Michael Lamkin, Cynthia Wu, Hao Su, and Melissa Gymrek. Deep neural networks identify context-specific determinants of transcription factor binding affinity. *bioRxiv*, 2020.

[283] Chenlin Zhou, Xiaoqin Yang, Yiyang Sun, Hongyao Yu, Yong Zhang, and Ying Jin. Comprehensive profiling reveals mechanisms of SOX2-mediated cell fate specification in human ESCs and NPCs. *Cell Research*, 26(2):171–189, 2016.

[284] Hannah Zhou, Avanti Shrikumar, and Anshul Kundaje. Towards a better understanding of reverse-complement equivariance for deep learning models in regulatory genomics. *bioRxiv*, pages 2020–11, 2021.

[285] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods*, 12(10):931–934, 2015.

[286] Jiyun Zhou, Qin Lu, Lin Gui, Ruifeng Xu, Yunfei Long, and Hongpeng Wang. MT-TFsite: cross-cell type tf binding site prediction by using multi-task learning. *Bioinformatics*, 35(24):5067–5077, 2019.

[287] Jan Zrimec, Christoph S Börlin, Filip Buric, Azam Sheikh Muhammad, Rhongzen Chen, Verena Siewers, Vilhelm Verendel, Jens Nielsen, Mats Töpel, and Aleksej Zelezniak. Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. *Nature Communications*, 11(1):1–16, 2020.