Methods to Improve Knowledge Transfer Efficiency for Data-limited Problems in Genomics

by

Ren Yi

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

September, 2021

Richard Bonneau

Kyunghyun Cho

Acknowledgements

I have spent more than half of my twenties in New York starting Fall 2015. During these years, I have probably questioned whether I would ever be able to get here – writing my thesis acknowledgment – way more than I should have. Things may seem difficult at times. But looking back, I can confidently say that it has been the most rewarding experience of my life so far, and I am grateful to have worked with and learned from many beautiful minds along the way.

I want to thank my advisors, Prof. Richard Bonneau and Prof. Kyunghyun Cho, for their mentorship, support, guidance, and patience throughout the years. I would also like to thank my thesis committee members, Prof. Dennis Shasha, Prof. Lakshminarayanan Subramanian, and Prof. Casey Greene, for keeping me on track and providing valuable feedback on my work.

I had the opportunity to collaborate with Prof. Dan Littman's immunology lab at the NYU medical school on several impactful genomics projects. I want to thank my experimental and computational collaborators from the Littman Lab: Prof. Emily Miraldi, Prof. Priya Issuree, Dr. Maria Pokrovskii, and Dr. Mo Xu. I had a fun and productive summer in 2019 working on DeepVariant in the Genomics team at Google Health. I want to thank my hosts at Google: Dr. Pi-Chuan Chang, Gunjan Baid, and Dr. Andrew Carroll. I also thank Dr. Mark DePristo for introducing me to the team at the very beginning.

I would like to thank all former and current Bonneau Lab and Cho Lab members for helping me along the way: Dr. Vlad Gligorijevic, Dr. Chris Jackson, Meet Barot, Omar Mahmood, Tymor Hamamsy, Daniel Berenberg, Dr. Jamie Morton, Giuseppe-Antonio Saldi, Michelle Badri,

Dr. Kostya Tchourine. I also would like to thank Shenglong Wang and other members of the NYU High Performance Computing team and the Flatiron Institute Scientific Computing team for providing excellent computational support for my research.

I also enjoyed getting to know many friends in New York who make me realize I'm never alone in this journey. I will try my best to name a few: Dr. Julia Koehler, Sreyas Mohan, Phu Mon Htut, Dima Taji, Dr. Lingdi Zhang, Dr. Shiwei Zheng, Dr. Emily Koo, Dr. Dayanne Castro, Zhouhan Chen, Xintian Han, Che Wang, and Yanqiu Wu.

My sincere gratitude goes to my parents, for their unconditional love and support throughout my whole life. Thank you for teaching me how to be independent at a young age and always encouraging me to pursue my dreams. Last but certainly not least, my love goes to David, who always remembers to drag me away from my desk once in a while so that I can appreciate the little things in life. Your love and encouragement have kept me sane throughout this journey.

Abstract

The recent advancement in computational genomics has largely benefited from the explosion of high-throughput genomic data and equal growth in biological databases. However, as more sequencing technologies become available and large genomic consortiums start to crowdsource data from larger cohorts of research groups, data heterogeneity has become an increasingly prominent issue. Data integration across multiple data sources and data modalities becomes particularly important for a greater number of biological systems. High-throughput omics data are typically highly skewed towards a small number of model organisms, factors, and conditions with which wet-lab experiments have higher success rates. It further introduces technical challenges when building machine learning models for problems with limited data. This thesis describes methods that improve knowledge transfer efficiency for learning data-limited problems through efficient task-specific feature representation in the multitask learning setting. We demonstrate the performance of our methods in two genomic problems – genetic variant calling and cell-type-specific transcription factor binding predictions.

Contents

List of Figures

List of Tables

List of Abbreviations

AE	autoencoder
ATAC-seq	Assay for Transposase-Accessible Chromatin using sequencing
ChIP-seq	chromatin immunoprecipitation followed by sequencing
CNN	convolutional neural network
CRF	linear-chain conditional random field
DNA-seq	DNA sequencing
DNase-seq	DNase I hypersensitive sites sequencing
ENCODE	Encyclopedia of DNA Elements
FFN	feedforward neural network
GIAB	Genome in a Bottle Consortium
GRU	gated recurrent unit
indel	insertion and deletion
MLP	multilayer perceptron
NGS	next-generation sequencing
PCR	polymerase chain reaction
PWM	position weight matrix
ReLU	rectified linear unit
RNA-seq	RNA sequencing
RNN	recurrent neural network

SNP	single-nucleotide polymorphism
TF	transcription factor
TRN	transcriptional regulatory network
WES	whole exome sequencing
WGS	whole genome sequencing

1 Introduction

The reductionist' approach for understanding biological phenomena–where cellular components are perturbed one at a time, and the responses are observed and analyzed–has been the workhorse of biology over the last century, resulting in numerous ground-breaking discoveries such as molecular machinery underlying apoptosis [Sulston and Horvitz 1977; Ellis and Horvitz 1986; Yuan et al. 1993; Hengartner and Horvitz 1994], the cellular origin of retroviral oncogenes [Stehelin et al. 1976], and restriction enzymes [Smith and Welcox 1970; Danna and Nathans 1971]. More recently, however, its role has gradually shifted towards experimentally validating *in silico* derived hypotheses amid rapid advancement in high-throughput sequencing technologies and machine learning algorithms.

As high-throughput biology data grow in ever-increasing volume, variety and complexity, we see a rapidly increasing number of research studies deriving conclusions from multimodal omics data, such as genomics, transcriptomics, proteomics, and metabolomics measurements [Sorokina et al. 2021; Ghosh et al. 2021; Schulte-Sasse et al. 2021]. It has been widely accepted that a comprehensive understanding of biological systems can only be derived from joint analyses of data from different sources and cellular levels [Joyce and Palsson 2006; Gomez-Cabrero et al. 2014; Li et al. 2018]. The goal of data integration, which consolidates data from disparate sources into one unified form, is therefore extracting additional biological knowledge that can not be otherwise obtained from any single dataset alone [Gligorijević and Pržulj 2015].

1.1 Motivations

1.1.1 The importance of data integration in biology

The motivation behind data integration in biology stems from the observation that biological systems work organically and cooperatively–on the cellular, tissue, and organism level–to maintain normal body function. More importantly, technical challenges can sometimes make it difficult to learn meaningful biological insights without proper data integration as a prerequisite.

Extracting valuable insights sometimes requires comparative analysis using data from multiple sources. For instance, integrating data from two public CRISPR-Cas9 cancer screens improves statistical power for identifying cancer lineage subtypes and unveils additional cancer biomarkers of gene dependency [Pacini et al. 2021]; integrating multiple networks, by learning both dataset-specific and conserved components among networks, improves transcriptional regulatory network inference accuracy [Castro et al. 2019]; integrating multiple protein-protein association networks using multimodal deep autoencoders improves protein function prediction efficiency [Gligorijević et al. 2018]; integrating DNA sequencing data from child-mother-father trios improves genetic variant calling performance compared to calling from individual datasets [Kolesnikov et al. 2021].

High-throughput sequencing data are sensitive to batch effects caused by non-biological factors, including differences in laboratory conditions, handling personnel, reagent lots, and technology platforms. It becomes particularly problematic when batch effects are correlated with the experimental outcome of interest, as it can potentially affect the validity of the research findings [Leek et al. 2010]. Data integration methods for removing batch effects focus on disentangling biological variables from technical variables (i.e., batch effects). Early methods designed to correct batch effect for microarray data have been reasonably effective at handling bulk and single-cell RNA sequencing data [Smyth and Speed 2003; Johnson et al. 2007b]. However, meth-

ods such as Harmony [Korsunsky et al. 2019] and LIGER [Welch et al. 2019] are more efficient at handling single-cell-specific batch correction problems.

More importantly, integrating and transferring knowledge from well-studied research problems is crucial for effectively learning data-limited problems, especially when working with high-throughput biological data. Over the last two decades or so, specialized biological databases have collected and generated a large number of omics samples, benchmarking datasets, and genome annotations for human [Bernstein et al. 2010; Network et al. 2012; Lonsdale et al. 2013; Consortium et al. 2015; Zook et al. 2018] and other model organisms [Bult et al. 2019; Ruzicka et al. 2019; Larkin et al. 2021; Cherry et al. 2012]. Take the Encyclopedia of DNA Elements (ENCODE) project [Moore et al. 2020] as an example. As of May 2021, ENCODE has collected 13828 next-generation sequencing datasets from 775 cellular conditions (i.e., cell lines, tissues, whole organisms, primary cell types, and *in vitro* differentiated cells), measuring many aspects of biological phenomena, including transcriptome, methylome, chromatin accessibility, chromatin interactions, and 3D chromatin structure. However, certain sequencing technologies, although cost-effective, are prone to sequencing and amplification errors and therefore cannot derive comprehensive insights without additional data from other sequencing platforms (Chapter 3 and Barbitoff et al. [2020]). Additionally, these datasets are typically highly concentrated towards a smaller collection of cellular conditions with which wet lab experiments have higher success rates (Chapter 4); we demonstrate this phenomenon by collecting all experiments from nine sequencing assays with the most number of datasets deposited in ENCODE and calculating the number of experiments per cellular conditions per assay (Figure 1.1).

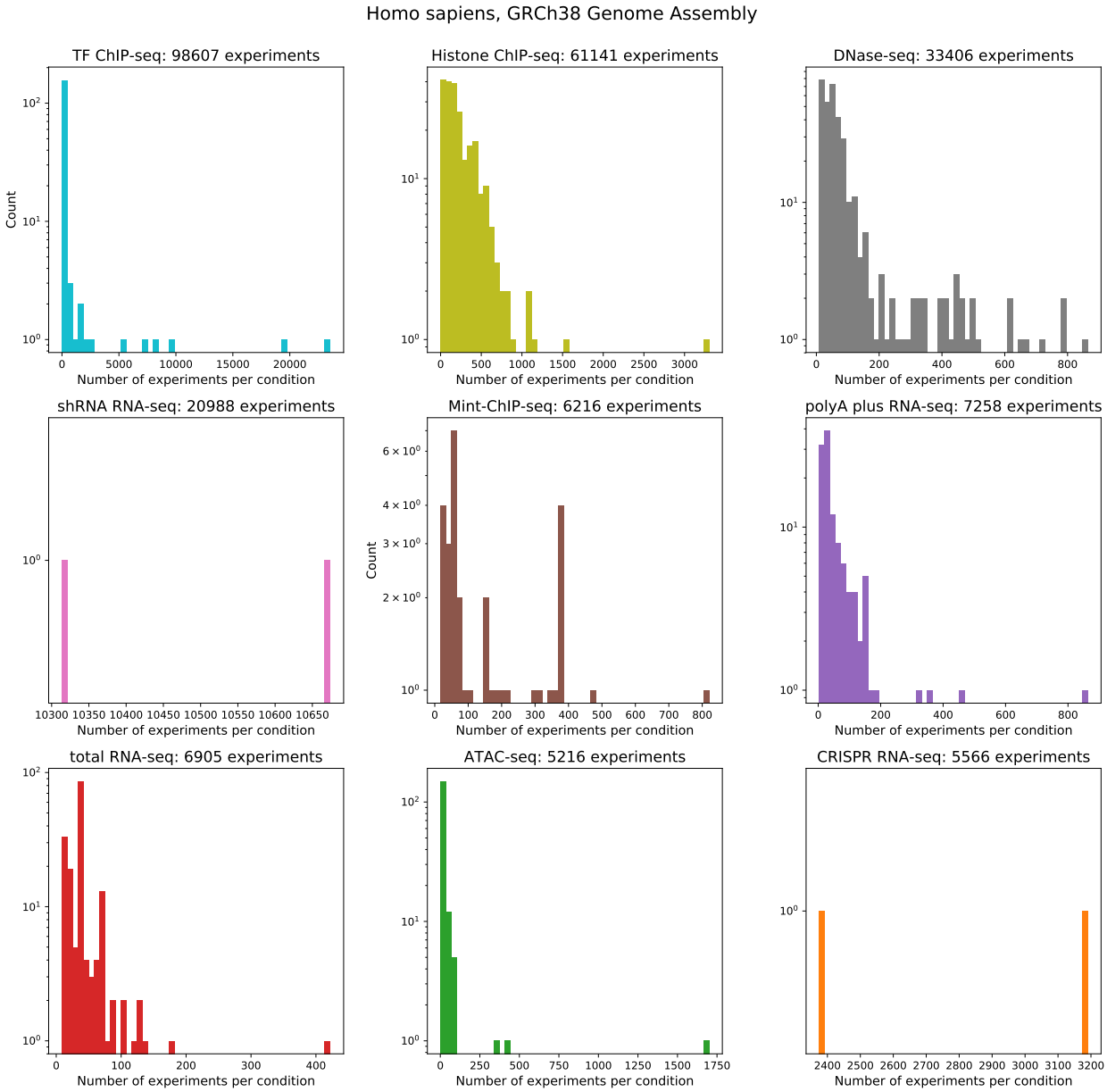


Figure 1.1: The number of experiments per cellular condition in ENCODE Consortium database stratified by assay type.

chromatin immunoprecipitation followed by sequencing (ChIP-seq) combines chromatin immunoprecipitation with DNA sequencing to identify binding sites of DNA-associated proteins [Johnson et al. 2007a]. Depending on the protein antibodies used in a particular ChIP-seq experiment, ChIP-seq can be used to detect binding sites for transcription factors (TF ChIP-seq) or histones (Histone ChIP-seq). Specialized ChIP-seq experiments conducted on low-input samples are called multiplexed ChIP-seq (Mint-ChIP-seq). (Continue on next page.)

Figure 1.1: (Continued from the previous page.)
Both DNase I hypersensitive sites sequencing (DNase-seq) [Boyle et al. 2008] and Assay for Transposase-Accessible Chromatin using sequencing (ATAC-seq) [Buenrostro et al. 2013] access genome-wide chromatin accessibility (i.e., open chromatin regions). Open chromatin regions are identified by regions sensitive to cleavage by DNase I in DNase-seq, and by regions tagged by hypteractive Tn5 transposase in ATAC-seq. RNA sequencing (RNA-seq) is a transcriptome profiling technique that measures the presence and quantity of RNA in biological samples. PolyA+ RNA-seq measures the mRNA level in a cellular condition, whereas total RNA-seq measures both mRNA and ribosomal RNA levels. Both shRNA RNA-seq (small hairpin RNA knockdown followed by RNA-seq and CRISPR RNA-seq (CRISPR genome editing followed by RNA-seq) measure cellular transcriptome after blocking the target gene expression. shRNA silences gene expression by inhibiting the transcription of the target gene mRNA, while CRISPR knockout the target gene at the DNA level. All data included in this analysis are human samples mapped to GRCh38 genome assembly. Data are obtained from the ENCODE Consortium [Moore et al. 2020] and reflect the database status as of May 2021.

1.1.2 Data integration versus data augmentation for data-limited problems

Data augmentation refers to a set of data analysis techniques that increase the amount of data available to a problem of interest either by adding modified versions of existing data or by generating simulated data from existing ones [Shorten and Khoshgoftaar 2019]. These techniques help reduce overfitting when training machine learning models and have been widely applied to improve many machine learning problems [Van Dyk and Meng 2001; Wei and Zou 2019; Kobayashi 2018]. Take image augmentation as an example; basic geometric manipulations–such as flipping, scaling, mixing images [Inoue 2018] and random erasing [Zhong et al. 2020]–add new images by making small modifications to existing data. Generative modeling techniques, such as generative adversarial networks (GANs) and variational autoencoders (VAEs), also create synthetic images from existing ones [Goodfellow et al. 2014; Hsu et al. 2017; Karras et al. 2020; Antoniou et al. 2017b]. GANs-based image augmentation techniques are particularly useful for specialized computer vision problems where access to data is limited, such as medical image analysis [Yi et al. 2019b]. The quality of the synthesized images can be directly evaluated by accessing the resemblance of the generated instances to real-world images or can be indirectly quantified by the

accuracy achieved by downstream tasks (e.g., image classification and image caption generation).

Computational methods can be benchmarked using empirical and/or simulated data [Escalona et al. 2016]. Simulating biology data impose domain-specific problems as the true underlying data distribution is unknown, unlike many other machine learning problems mentioned previously. Empirical observations, which provide truth labels for many computational methods, can only capture a snapshot of the true data distribution. Nevertheless, many computational methods have been proposed to simulate DNA sequencing (DNA-seq) [Escalona et al. 2016], RNA-seq [Frazee et al. 2015; Zappia et al. 2017; Zhang et al. 2019; Gerard 2020], ChIP-seq [Datta et al. 2019; Subkhankulova et al. 2020; Zheng et al. 2021], and other types of next-generation sequencing (NGS) datasets.

Many simulation methods emulate the real-world scenarios by modeling a set of manually defined simulation parameters, such as polymerase chain reaction (PCR) bias [Angly et al. 2012], base-calling errors [Holtgrewe 2010], sequencing depth [Hu et al. 2012], and mutation rates [McElroy et al. 2012]. These parameters can be estimated from sequencing reads of related samples or provided by the users. Alternatively, additional signals can be added to the empirical datasets, resulting in simulated data that exhibit realistic attributes of real data [Gerard 2020]. However, manually curated simulation parameters can only explain a fraction of the variations in real-world data. Future collaborative efforts are necessary to standardize the evaluation pipeline of NGS simulation methods across available sequencing platforms [Earl et al. 2011].

1.2 Thesis outline

This thesis discusses data integration methods that improve knowledge transfer efficiency for learning data-limited problems in genomics. Chapter 1 introduces the concept of data integration and the importance of data integration when working with high-throughput omics data and machine learning. We realize this thesis covers a wide range of research topics in machine learn-

ing and genomics. We, therefore, provide the necessary background knowledge in Chapter 2. We apply our data integration methods to two genomic problems – genetic variant calling and cell-type-specific transcription factor binding prediction. Our findings are presented in Chapter 3 and Chapter 4. Conclusions and future directions are discussed in Chapter 5.

2 Background

2.1 Data integration

Data from different sources (e.g., DNA-seq data from different sequencing platforms) or covering different aspects of the same biological process (genome-wide chromatin accessibility, transcriptome, and methylation profiles from the same cellular condition) are commonly integrated to solve one or more biological tasks simultaneously. We can therefore break down the data integration problems in genomics into two basic forms: 1) the same type of feature from multiple sources are integrated to learn source-specific tasks or objectives, and 2) heterogeneous data measuring different aspects of the same biological phenomenon are integrated to learn the same task or objective.

Many strategies for data integration have been proposed, such as feature concatenation, Bayesian models, tree-based methods, and matrix factorization models. For more details on different types of data integration techniques for genomics, I refer readers to Gligorijević and Pržulj [2015], Li et al. [2018] and Zitnik et al. [2019]. This section will focus on introducing two model design techniques for data integration–multitask learning and transfer learning–that have become increasingly popular for designing deep learning models due to their efficiency in handling a large amount of heterogeneous data.

2.1.1 Multitask learning and multimodal models

The first basic form of data integration problems mentioned above can also be referred to as multitask learning, in which multiple tasks (e.g., classification or regression) are learned jointly to improve the generalization performance of all tasks. Some example tasks are predicting binding sites of DNA-binding proteins in different cell types, predicting genetic variants from different types of DNA-seq data, and predicting cell-type-specific molecular phenotypes (e.g., chromatin accessibility and gene expression) from DNA sequences. In the multitask learning setting, datasets from multiple related tasks typically follow distinct data distributions due to their different biological or technical properties. Therefore, leveraging the cross-dataset commonalities while recognizing the differences is crucial in order for the multitask training strategy to be successful. A typical multitask learning model contains a set of shared parameters that learn the common properties from input features of all tasks, as well as a set of task-specific parameters that learn task-specific features (Figure 2.1a). Compared to many single-task models, where each task is learned separately, multitask learning improves model generalization (by integrating data for multiple tasks) and prediction efficiency (by sharing parameters and computation among tasks).

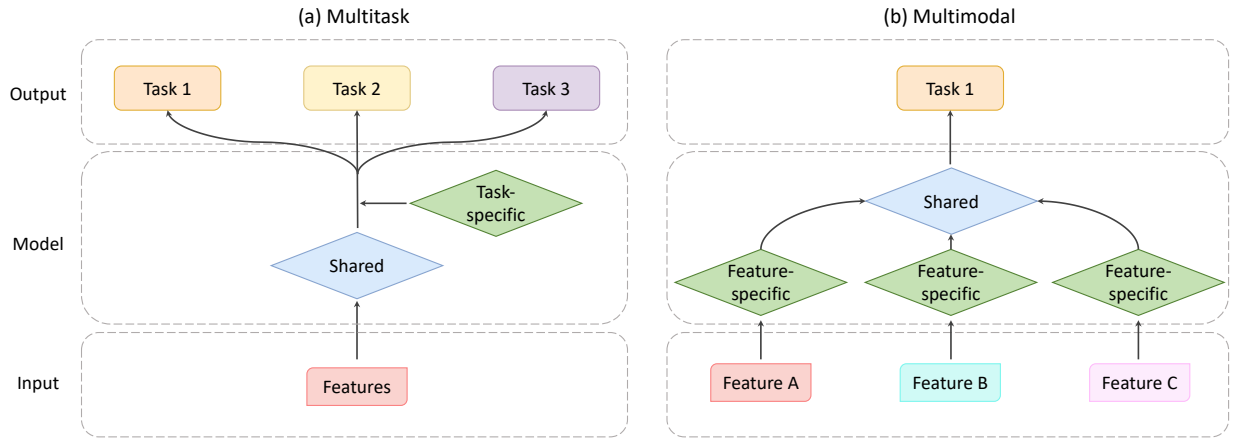


Figure 2.1: Multitask learning and multimodal models.

Similar to multitask learning, machine learning models can be extended to take multiple in-

put data modalities, each measuring a different aspect of the same phenomenon. For instance, cell-type-specific transcription factor binding sites can be predicted from the DNA sequence and chromatin accessibility data; the amino acid sequence, protein-protein interactions, and 3D protein structures can all be used as input features to predict protein functions. Apart from the scenario where raw input features are directly combined, multimodal models typically have a feature-specific component that learns to represent each type of feature separately, followed by a shared component that integrates the result of the feature-specific component and generates predictions (Figure 2.1b).

A potential problem for multimodal models, however, is incomplete data modalities. Biometric features such as height, weight, and age may not be available for all patients; chromatin accessibility data may be missing for a particular cell type of interest when predicting cell-type-specific transcription factor binding sites. Multimodal models have a severely limited scope of applications without proper mechanisms for handling missing features. One simple strategy for handling missing modalities involves assigning a special value to the missing features and manually excluding a subset of features periodically when training the model [Jaques et al. 2017b]. Another important research direction focuses on effective fusion of multimodal data. In addition to feature concatenation [Wang et al. 2017], methods have been proposed to either learn modality-specific factors [Liu et al. 2018] or reconstruct missing modalities through meta-learning [Ma et al. 2021].

2.1.2 Transfer learning

Data from a related problem can sometimes be used to facilitate learning of the target problem of interest. The research problem that focuses on transfering knowledge gained from the source task to the target task is referred to as transfer learning. Compared to training from scratch, a model pretrained on a related task converges faster, requires less training data, and in most cases, improves performance of the target task compared to a model trained from scratch [Eraslan et al.

2019]. Models pretrained on natural images [Deng et al. 2009] have been successfully adopted to a rich collection of medical image studies [Morid et al. 2020]. Models pretrained on images also improves training efficiency for calling genetic variants [Poplin et al. 2018a], despite the lack of resemblance between these two research problems. However, the benefit of pretraining diminishes with longer training time and more training data [Kornblith et al. 2019].

In the simplest case, transfer learning can be achieved by directly fine-turning a pretrained model using data from the target task. When the source and target tasks are related, however, characterising the task-specific features through a task-specific model component, in addition to a shared model component, may be beneficial for improving transfer learning performance (Figure 2.2). In practice, relative size of the shared and task-specific components will likely depend on the relatedness of the source and target tasks.

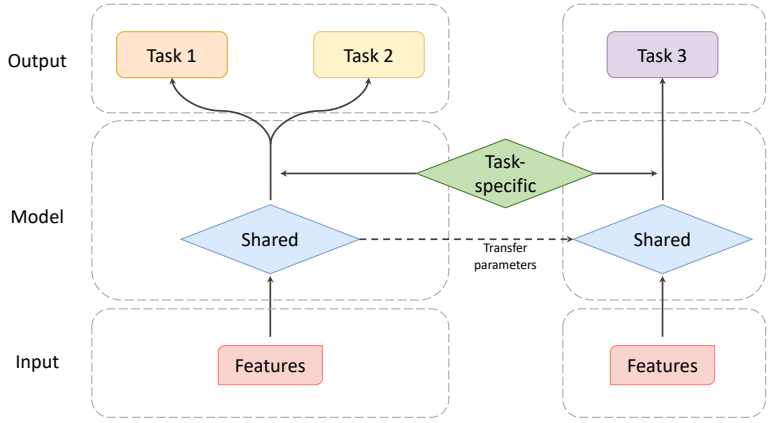


Figure 2.2: Transfer learning.

2.2 Deep learning

Deep learning belongs to the broad family of machine learning methods, which uses artificial neural networks to learn representations of data [LeCun et al. 2015]. The word "deep" in deep learning refers to the use of multiple layers of elementary operations in the network, each of

which takes the result from the previous layer as input to the next layer. It reduces the need to create handcrafted features because the stacking of neural network layers can help learn increasingly more complex features from input data. Different types of neural network layers also specialize in extracting features from data with distinct topology. Here I briefly introduce several common types of neural network layers, followed by discussing widely used deep neural network architectures for extracting complex features. My goal is to introduce the deep learning concepts that are important for the foundation of this thesis. I recommend Goodfellow et al. [2016] and Cho et al. [2014c], from which I get plenty of inspiration.

2.2.1 Common deep learning architectures

Neural networks can be roughly divided into two classes: feedforward neural networks (FFNs) and recurrent neural networks (RNNs). In feedforward networks, as the name suggests, information flows unidirectionally from the input x to the output y, possibly through some intermediate computations. In contrast, RNN contain feedback connections, where the output of the network is fed back to itself. Gradients for the FFN and RNN can be calculated through the generalized back-propagation algorithm [Rumelhart et al. 1986], although back-propagation through time is often used to refer to gradient calculation in recurrent neural networks due to its sequential nature.

Multilayer perceptron Multilayer perceptron (MLP) is a class of FFNs consists of fully connected layers – each node in one layer connects to every node in the next layer. An MLP contains one input layer, one output layer, and at least one hidden layer in between. Except for the input layer, each subsequent layer has a non-linear activation function. An MLP with one hidden layer can be represented as

$$y = \sigma(Wx + b) \tag{2.1}$$

where W and b are the parameters of the linear transformation, and σ is a non-linear activation function. The sigmoid and hyperbolic tanhgent functions were commonly used activation functions until the introduction of rectified linear unit (ReLU).

$$y = \max(0, x) \tag{2.2}$$

ReLU allows the model to obtain sparse representations and has fewer vanishing gradient problems than the sigmoid and hyperbolic tanhgent functions [Glorot et al. 2011].

Convolutional neural network Convolutional neural networks (CNNs) are a specialized type of FFN for processing data with grid-like structures, such as time-series data and image data. The convolution layer in a CNN uses filters (w) that perform the convolution operations to scan the input (x) with respect to its dimensions. Suppose for 1D time-series data, the time index t can only take on integer values. Then the discrete convolution operation ($*$) can be defined as

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a) \tag{2.3}$$

where the output $s(t)$ is often referred to as the feature map. In many CNN applications, a pooling layer is usually followed by a convolution layer, which reduces the dimensions of the output feature maps.

Recurrent neural network RNNs [Rumelhart et al. 1986] refers to a family of neural networks specialized to process sequential values $x^{<1>} \cdots x^{<t>} \cdots x^{<T>}$. In RNNs, each time step has a hidden state; the hidden state from the previous time step is used as the input to generate output for the current time step. At each time step t, the hidden state $h^{<t>}$ and the output $\hat{y}^{<t>}$

are defined as

$$
\begin{aligned}
h^{<t>} &= \sigma(W_{hh}h^{<t-1>} + W_{hx}x^{<t>}) \\
\hat{y}^{<t>} &= softmax(W_{yh}h^{<t>})
\end{aligned}
\tag{2.4}
$$

where W_{hh}, W_{hx}, W_{yh} are coefficients shared by all time steps and σ is an activation function. Such RNNs can process the input of any length. However, it fails to capture long-term dependencies due to vanishing (or exploding) gradient problems. To remedy this problem, Cho et al. [2014a] proposed a variant of RNN, called the gated recurrent unit (GRU). GRU defines several gates, each of which has the form

$$
\Gamma = \sigma_g(Wx^{<t>} + Uh^{<t-1>} + b) \tag{2.5}
$$

where W, U, b are coefficient specific to the gate and σ_g is the sigmoid function. Specifically, GRU uses two gates: the update gate Γu that controls how much information the previous hidden state should pass on to the next, and the reset gate Γr that controls how much information from the past should forget. The hidden state $h^{<t>}$ is updated by

$$
\begin{aligned}
\hat{h}^{<t>} &= \phi_h(W_h x^{<t>} + U_h(\Gamma_r \odot h^{<t-1>}) + b_h) \\
h^{<t>} &= (1 - \Gamma_u) \odot h^{<t-1>} + \Gamma_u \odot \hat{h}^{<t>}
\end{aligned}
\tag{2.6}
$$

where $\hat{h}^{<t>}$ is a memory cell parameterized by W_h, U_h and b_h, ϕ_h is the hyperbolic tangent function, and $\odot$ denotes the element-wise multiplication between two vectors.

2.2.2 Feature extraction

Feature extraction aims at deriving informative and non-redundant information from raw data. It is closely related to dimensionality reduction, as it becomes necessary when the dimensionality of the input data is too large or the input data are suspected of having redundant measurements.

One of the most used linear dimensionality reduction techniques is called principal component analysis (PCA). The goal of PCA is to transform the original variables to a new set of orthogo-

nal variables, called the principal components (PCs), that are ordered so that the first PC preserves the greatest variance in the original variables, the second PC preserves the second greatest variance, and so on [Jolliffe 1986]. PCA belongs to a class of unsupervised dimensionality reduction techniques, which also include random projection and independent component analysis (ICA). ICA aims to decompose original variables into independent non-Gaussian subcomponents that are statistically independent of each other [Hyvärinen 2013]. Important applications of ICA include the noisy speech recognition [Hsieh et al. 2009], and separating true signals from noise in EEG and fMRI scans [Winkler et al. 2011; McKeown et al. 2003]. The idea behind these algorithms also lays the foundation for signal transmission and data compression, where the goal is to preserve the most information possible from the original data with the least number of variables.

Another family of machine learning algorithms for feature extraction involves learning to represent input data or entities with low-dimensional vector embeddings. These vector embeddings can be learned using self-supervised learning, such as autoencoder (AE) [Kramer 1991]. An AE contains an encoder and a decoder. The encoder compresses the input data into a lower-dimensional latent vector representation, from which the decoder reconstructs the input data. An AE with one linear transformation is nearly equivalent to a PCA. However, using non-linear neural network architectures additionally allow AEs to learn complex features from the input data [Hinton and Salakhutdinov 2006]. Various regularization techniques have been proposed to ensure the learned representations are meaningful [Ng et al. 2011; Vincent et al. 2008; Rifai et al. 2011]. Variational autoencoders also allow the model to describe the latent representations in terms of probability distributions [Kingma and Welling 2013]. In fact, given the appropriate set of features, any entities can be represented by vector embeddings [Wu et al. 2018; Hamilton et al. 2018; Lerer et al. 2019]. For example, in content recommendation, a vector representation of a particular user's preference can be learned from a set of items the user likes (or clicks on); in text classification, a vector representation of a text label can be learned from a set of documents (or bags of words) that describe the text label; in knowledge graph link prediction, vector representa-

tions of nodes and edges can be learned from other nodes and edges in the graph. Such methods are commonly trained in the supervised fashion using the k-negative sampling strategy [Mikolov et al. 2013] that minimizes the distance between the learned embeddings and the correct class label while maximizing the distance between the learned embeddings and k incorrect class labels.

2.3 Genetic variant

If history is passed down through generations in the form of language, what would be the language of human evolution? I sometimes like to think that the answer lies in our genetic code – the over 3 billion base pairs (bp) of nucleotides that constitute our genome. Although we cannot ignore the contribution of environmental factors, our genome plays a vital role in determining our phenotypic traits, susceptibility to diseases, and in some cases, our habits and behaviors [Breed and Sanchez 2010]. For example, genomic factors are involved in nine of the ten leading causes of death in the United States[1]; though no single gene is thought to be responsible, Schizophrenia tends to run mostly in families [Gejman et al. 2010].

The first two versions of the human reference genomes were published in 2001 [Consortium et al. 2001; Venter et al. 2001]. However, no two people's genomes are identical, and the changes in the genetic makeup of an individual's genome compared to the human reference genome are referred to as the genetic variant. It is believed that any two human beings are 99.9% identical in terms of our DNA. The 0.1% (or 3 million) bp that remain contain the important genetic variants that contribute to the astonishing variety of individual differences in terms of appearance, disease susceptibility, and aptitude in athletics, math, music, and more.

[1] https://www.cdc.gov/nchs/fastats/leading-causes-of-death.htm

2.3.1 Types of genetic variants

Genetic variants can be broadly divided into two categories according to the size of the genetic changes: small-scale sequence variation under 1 kilobase (kb) and large-scale structural variation longer than 1 kb. Sequence variation includes base-pair substitution, and insertion and deletion (indel). Structural variation includes copy number variation (CNV) and chromosomal rearrangement.

Base-pair substitution It is the most abundant and most studied class of genetic variations among individuals. The two types of base-pair substitution include single-nucleotide polymorphism (SNP) and single-nucleotide variant (SNV); both refer to the substitution of a single nucleotide at a specific genomic location. SNP defers from SNV in that SNP exclusively refers to germline mutation, whereas SNV can be somatic.

Indel Insertion or deletion of bases in the genome is referred to as indel. Within the protein-coding regions of the genome, an indel likely causes frameshift mutation unless the size of the indel is a multiple of 3. In human, the indel frequency is also considerably lower than that of SNP. Calling indel variants, therefore, presents a harder machine learning problem compared to calling SNP variants.

CNV and chromosomal rearrangement CNV refers to the concept that the number of copies of a particular gene varies from one individual to another, resulting from sections of the DNA getting duplicated or deleted. Copy number of a gene affects the transcription and subsequently translation of a particular gene. Although the relationship between protein levels and copy number varies, CNVs that alter the level of proteins controlling critical cellular functions (e.g., housekeeping genes, dosage-sensitive genes [Riggs et al. 2012], and genes associated with Mendelian disorders [Amberger et al. 2019]) can lead to disease susceptibility [Zarrei et al. 2015]. In some lit-

erature, CNV is also referred to as the imbalanced chromosomal rearrangement that changes the copies of the part of affected chromosomes through either deletion or insertion. Chromosomal rearrangement refers to the type of mutations that changes the structure of the chromosomes. The other type of rearrangement, called the balanced chromosomal rearrangement, includes inversion and reciprocal translocation. Both types of balanced chromosomal rearrangement change the chromosomal gene order, with the inversion flipping the chromosome 180 degrees and the reciprocal translocation exchanging genetic materials in two chromosomes or two parts of the same chromosome [Griffiths et al. 1999].

2.3.2 DNA sequencing data for calling genetic variants

Genome-wide detection of genetic variants can be achieved by either DNA microarray or DNA-seq, although nowadays, the former is mainly used in genome-wide association studies (see Section 2.3.4) to identify common SNPs among thousands of individuals in the population of interest. DNA-seq can be conducted on the whole genome level (whole genome sequencing) or on the whole exome level (whole exome sequencing) that covers the 1-2% of the genome that codes for proteins. Large sequencing companies–such as Illumina, Pacific Bioscience (PacBio), and Oxford Nanopore Technologies (Nanopore)–independently developed multiple DNA-seq platforms. The choice of platforms will likely depend on the type of variants on which the research projects are conducted.

Currently, the most widely used technology is Illumina's sequencing by synthesis. This platform generates short reads (up to 300 bp) with a very low error rate and is commonly used to detect short SNP and indel variants. Both PacBio and Nanopore belong to the third generation sequencing technologies for sequencing long DNA reads [Amarasinghe et al. 2020], and are better suited for detecting large structural variants. Read length on average is around 10–16 kb for PacBio, and 10–30 kb for Nanopore. However, longer read length is accompanied by a higher error rate, and post-sequencing error correction [Fu et al. 2019; Zhang et al. 2020] is mandatory

for various downstream analyses.

2.3.3 Genetic variant callers

Genetic variant calling refers to the identification of genetic variations from individuals' genomes using DNA sequencing data. The starting point of many genetic variant callers is the DNA sequencing reads from an individual. These reads are first aligned to the reference genome. The read alignment is then provided to a machine learning model to predict three types of genotype likelihood: homozygous reference allele (hom ref), heterozygous allele (het), and homozygous alternative (hom alt) (Figure 2.3). Read alignment is typically carried out by standard alignment software such as BWA [Li and Durbin 2009] and Bowtie2 [Langmead and Salzberg 2012], although local realignment has also been proposed by several methods to improve variant calling accuracy further [DePristo et al. 2011b; Poplin et al. 2018a].

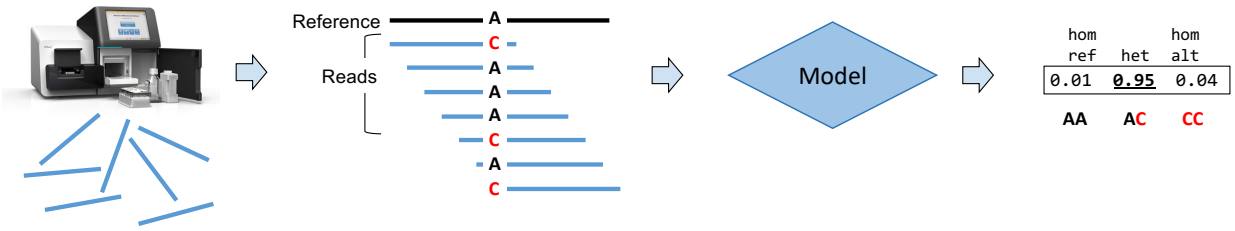


Figure 2.3: A typical variant calling pipeline.

Genome Analysis Toolkit HaplotypeCaller [Poplin et al. 2018b], Strelka2 [Kim et al. 2018], Freebayes [Garrison and Marth 2012] are all state-of-the-art germline genetic variant calling methods. Rapid development in deep learning techniques has allowed the development of deep neural network-based variant callers [Poplin et al. 2018a; Luo et al. 2019]. The InceptionV3 [Szegedy et al. 2016a] architecture, originally proposed for the image classification problem, has also been adopted by DeepVariant, a germline variant caller for short SNP and indel variants. DeepVariant consistently shows the best performance compared to many other state-of-the-art methods [Abasov et al. 2021]. Major concerns, if any, about the DeepVariant model center around

the intuition behind the choice of model architecture, as natural images show low resemblance to variant read alignment features. Luo et al. [2019] later proposed a much smaller CNN model called Clairvoyante. Clairvoyante contains 1.6 million parameters, 13-times fewer than the DeepVariant InceptionV3 model, which has 24 million parameters. Deep neural networks, such as InceptionV3, can be more effective at feature representations. However, models with fewer parameters require fewer data to train and can potentially be more effective for learning specialized variant calling problems with limited data.

High confidence benchmarking datasets and standardized variant caller evaluation pipelines have greatly facilitated the translation of variant calling methods to routine research and clinical practice. Global Alliance for Genomics and Health has also recommended the best practice for benchmarking germline small-variant calls, and developed hap.py for evaluating variant callers' performance and stratifying performance by variant type and genome context [Krusche et al. 2019]. Both the Genome in a Bottle Consortium (GIAB) and the National Institute of Standards have published benchmark variant calls for small variants [Zook et al. 2016, 2018] for several deeply sequenced human genomes. Zook et al. have also recently developed a benchmark set for structural variants. However, this structural variant benchmark set mainly includes germline deletions and insertions. Best practices for evaluating structural variants and high-confidence variant calls for complex structural variants–including inversions, duplications, and large CNVs–are still under active research and development [Zook et al. 2020].

2.3.4 Linking genetic variants to phenotypic traits

Accurately calling genetic variants is merely the first step towards identifying functional associations between variants and complex phenotypic traits. It is important to note that most genetic variants achieve significant frequencies in the human population simply by chance, and they do not contribute to phenotypic variations [Kimura et al. 1968; Frazer et al. 2009]. For example, there are roughly 3-5 million SNPs in each person's genome; but trait association has only been

identified in a small fraction of these SNPs [Buniello et al. 2019].

The current advancement in the variant-phenotype association can be largely attributed to the development of genome-wide association studies (GWAS, McCarthy et al. [2008]). GWAS detects associations between genetic variants and trait status through genotyping hundreds of thousands to millions of individuals in the population of interest. As of May 2021, 5037 GWAS has been published. These studies identified 160,065 SNPs and 257,351 SNP-trait associations; among the identified associations, 55,058 of them have reached a genome-wide significance threshold (p-value $\leq 5.0 \times 10^{-8}$) [Buniello et al. 2019]. These studies provide useful insights into understanding the cellular mechanisms that contribute to one's disease susceptibility and how clinical care and therapies can be optimized based on individuals' genotypes [Tam et al. 2019]. For example, the first GWAS, published by Edwards et al. in 2005, identified that the $\text{Try}^{402} \rightarrow \text{His}^{402}$ protein polymorphism in the gene encoding complement factor H significantly increases the risk of age-related mascular degeneration. GWAS identified variants could be used to inform drug selection and drug dosage: the Clinical Pharmacogenitics Implementation Consortium has established guidelines for pegylated inteferon-α-based treatment regimens for chronic interferons C virus infection based on the IL28 genotype [Muir et al. 2014], as a SNP identified near the IL28 gene increases patients' response rate to the pegylated interferon-α and ribavirin therapy [Ge et al. 2009].

GWAS identifies associations between genetic variants and phenotypic traits. However, the effect of the genetic variants on transcription and gene expression remains to be carefully characterized to finely-map the regulatory potential of common and rare variants. The effect of genetic variants on gene expressions can be identified using a method called the expression quantitative trait loci (eQTLs) mapping. A typical eQTL workflow includes collecting hundreds or thousands of gene expression datasets (either through microarray or through RNA-seq) and subsequently identify genetic variants whose presence (and the number of copies) affect the level of gene expressions [Westra and Franke 2014]. Large-scale eQTL studies have collected gene expression

data in human [Trynka et al. 2011; Grundberg et al. 2012; Zhu et al. 2016] and many other organisms [Keurentjes et al. 2007; Viñuela et al. 2012; Hasin-Brumshtein et al. 2014; Fair et al. 2020].

Such studies, although providing valuable information connecting genetic variants to their relevance to disease, are generally limited to common variants that have matched expression data in relevant tissues and cell types, which can be infeasible or difficult to obtain. *In silico* prediction of molecular phenotypes from biological sequences, therefore, has emerged as a cost-effective way to facilitate quantitative trait loci identification. Several deep learning frameworks–including DeepSea [Zhou and Troyanskaya 2015], Basenji [Kelley et al. 2018b], ExPecto [Zhou et al. 2018], Basenji2 [Kelley et al. 2018a], and Enformer [Avsec et al. 2021a]–have been proposed to predict molecular phenotypes–including transcription factor binding, histone modification, chromatin accessibility, and gene expression–from DNA sequences. Compared to earlier ones, newer methods significantly improve models' capacity in handling 1) longer DNA sequences, 2) more complex model architectures, 3) higher prediction resolution, and 4) a larger set of molecular phenotypes as target labels. The high-resolution predictions generated from these models have shown to be beneficial in disentangling causal variants from associations, which has been historically difficult to pinpoint due to linkage disequilibrium that causes nonrandom association of variants at different loci [Slatkin 2008].

Molecular phenotype datasets used to provide target labels for training these models are unanimously mapped to the human reference genome. The effect of the genetic variants can be generated at prediction time by providing the model with input DNA sequences containing the minor alleles. However, there is no direct gold standard for evaluating the prediction accuracy as generating molecular phenotype datasets containing the minor alleles can be laborious and costly. Community efforts to create benchmarking datasets are necessary to systematically measure the efficacy of *in silico* molecular phenotype prediction methods. Additionally, the current design of the model architectures makes it difficult to expand predictions beyond the training set of molecular phenotypes. Future work is necessary to address this shortcoming, e.g., via representation

learning of cell types and assays [Avsec et al. 2021a], to make such models more transfer-learning-friendly.

2.4 Transcriptional regulation

Although I advocate for the freedom of speech when it comes to democracy and social justice, life forms would probably be very chaotic if all the genes in our bodies are free to "express" themselves. Different tissues and cell types express different sets of genes during normal organism development to carry out their designated function. Differential gene expression among tissues and cell types is accomplished primarily through complex regulation of gene transcription, in which a segment of DNA is converted into RNA. Transcription is tightly regulated by multiple pieces of cellular machinery, including the formation of promoter initiation complex at the transcription start site, the recruitment of transcription factors and enhancers, and RNA transcripts' elongation. One of the essential contributors to transcription regulation is transcription factors (TFs). TFs orchestrate the regulation of transcription by binding to specific short DNA sequences primarily at the promoter and enhancer regions of their target genes and subsequently activate and repress gene expression.

Transcription regulation requires cooperative activities of many intracellular and extracellular factors, both spatially and temporally. Here I attempt to provide a brief introduction to the basics of transcription regulation, emphasizing the involvement of TFs during this process. Our current understanding of TFs is based on decades of experimental and computational research on their binding specificities and functional properties. I hope to provide sufficient biological background to motivate future computational modeling of TF binding and functions inspired by their biological properties.

2.4.1 Regulation at the level of chromatin state

Chromatin consists of DNA and histone proteins. One hundred and forty-seven bp DNA sequences wrap around a histone octamer and form the chromatin's basic functional unit called the nucleosome. Each histone octamer consists of two copies of each of the four positively-charged histone proteins H2A, H2B, H3, and H4. These histones, therefore, attract negatively charged DNA, forming a tightly wrapped histone-DNA complex. Post-translational modifications, primarily located at the (N)-terminus of the histone proteins [Luger et al. 1997], alter chromatin structure, recruit chromatin remodeling enzymes, and subsequently affect chromatin accessibility, transcription, and other DNA processes [Bannister and Kouzarides 2011]. Common histone modifications include acetylation, methylation, phosphorylation, SOMOylation, and ubiquitination. Genome-wide histone modifications can be profiled using ChIP-seq (Histone ChIP-seq, Figure 1.1)

Acetylation of lysine (K) residues in histone proteins reduces their positive charge, weakening histone-DNA interactions and increasing the propensity for gene transcription. For example, acetylation at the 9th (H3K9ac) and the 27th (H3K27ac) lysines of H3 has been widely accepted as a marker for activate promoters [Karmodiya et al. 2012] and enhancers [Creyghton et al. 2010]. Compared to acetylation, methylation of histones have more dynamic functional implications; methylation can increase or decrease gene transcription, depending on the specific amino acid and the number of added methyl groups [Tessarz and Kouzarides 2014; Hyun et al. 2017]. Trimethylation of the 4th lysine in H3 (H3Kme3) is commonly associated with actively transcribed promoters [Liang et al. 2004]. Methylation in H3K9, H3K20, and H3K27 increases gene silencing and chromatin compaction and is a major contributor to the X chromosome inactivation in early female embryonic development in mammals [Kohlmaier et al. 2004; Brinkman et al. 2006].

Various histone modifications unpack the chromatin, making it accessible to the transcription apparatus [Wang et al. 2012] that orchestrates the transcriptional response. Therefore, open

chromatin regions provide important insight into which part of the genome are hotspots for the regulation of gene expression [Klein and Hainer 2020]. High-throughput assays for measuring DNA accessibility and nucleosome positioning include DNase-seq [Song and Crawford 2010], FAIRE-seq [Giresi et al. 2007], MNase-seq [Henikoff et al. 2011], and ATAC-seq [Buenrostro et al. 2013], with DNase-seq and ATAC-seq being the most widely used. Exposed DNA structures are subject to degradation by enzymes, such as Deoxyribonuclease I (DNase I). Therefore, DNase I hypersensitive sites in the genome can be captured and sequenced to identify accessible chromatin regions. Alternatively, in ATAC-seq, hyperactive Tn5 transposase simultaneously fragments and tags exposed DNA, which can be subsequently sequenced to identify open chromatin. ATAC-seq has overtaken DNase-seq as the preferred assay to profile chromatin accessibility, as it requires fewer steps and input materials. ATAC-seq has also been adopted to single-cell sequencing [Chen et al. 2019] to identify accessible chromatin with much-improved granularity.

2.4.2 Regulation through transcription factors

Transcription factors (TFs) are a set of DNA binding proteins involved in regulating transcription initiation and elongation. More than six percent of human genes are believed to be TFs or cofactors [Lambert et al. 2018]. Each TF contains at least one DNA-binding domain (DBD), with most TFs containing at least one of the two DBD types – C2H2-ZFs and Homeodomains [Lambert et al. 2018]. TF binding to DNA exhibits strong sequence specificity. *In vitro* TF binding sites can be profiled through high-throughput assays including protein binding microarray (PBM) and high-throughput systematic evolution of ligands through exponential enrichment (HT-SELEX). These assays generate large amount of short DNA sequences preferred by a given TF, which is then summarized as TF binding motifs. Many *de novo* motif discovery programs, such as MEME [Bailey 1994; Bailey and Elkan 1995; Bailey et al. 2006, 2009], BEEML-PBM [Zhao and Stormo 2011], and Seed-and-Wobble [Berger et al. 2006], summarize TF sequence specificity in motif position weight matrices (PWMs). A number of TF motif databases, including Cis-BP [Weirauch et al.

2014], TRANSFAC [Wingender et al. 1996, 2000], JASPAR [Sandelin et al. 2004; Fornes et al. 2020], and UniPROBE [Newburger and Bulyk 2009; Hume et al. 2015], have collected a large number of TF motifs in many model organisms. It is important to note that although motifs represented as PWMs are easily interpretable, heavy information loss cannot be avoided when condensing TF binding specificity in a 2D matrix. Evidence suggests newer deep learning techniques, such as CNNs [Alipanahi et al. 2015; Avsec et al. 2021b] and vector embeddings (Yuan et al. [2019] and Section 2.2.2), are more effective at capturing complex TF sequence preferences.

Most human TFs contain at least one effector domains that recruit cofactors to regulate transcription. Depending on the type of effector domains a particular TF possesses, it can 1) bind to the basal transcription machinery, 2) bind to other TFs, or 3) recruit histone and chromatin remodeling enzymes [Frietze and Farnham 2011]. Some TFs interact with the core promoter regions, while others can be recruited to distal enhancers and interact with the promoter-bound proteins via the looping mechanisms [Frietze and Farnham 2011]. It is believed that very few TFs occupy most of their motifs *in vivo*. Many transient and low-affinity binding sites also do not exhibit motif enrichment. Many TFs work cooperatively to achieve desired cellular function. Cooperative TF binding event also affects TF sequence preferences [Jolma et al. 2015]. Although most TFs primarily bind to open chromatin regions, pioneer transcription factors can efficiently bind to nucleosomal DNA to affect the stability of the nucleosome [Zhu et al. 2018]. Therefore, assays for identifying open chromatin can only explain a fraction of the *in vivo* TF binding landscape.

2.4.3 Computational modeling of TF functions

TFs control the expression patterns of target genes by first binding to regions containing promoters, distal enhancers, and/or other regulatory elements. However, functional interactions between TFs and target genes are further complicated by TF concentrations and co-occurrence of other TFs. TF binding to DNA is the first step towards TFs' functional regulation of target gene

expressions. Functional connections between TFs and genes combine to form a transcriptional regulatory network (TRN), represented as a directed graph. Determining the valid TRN is necessary to explain how genetic variants can lead to disease susceptibility [Hu et al. 2016], how variation at the genetic level leads to selectable phenotypic variation [Peter and Davidson 2011], and how to re-engineer organisms to produce industrial chemicals and enzymes efficiently [Huang et al. 2017]. Many machine learning models have been proposed to infer genome-wide TRN through gene expression data and prior knowledge of the network structure [Liao et al. 2003; Faith et al. 2007; Marbach et al. 2012; Kamimoto et al. 2020; Pratapa et al. 2020; Gibbs et al. 2021].

Curated databases of regulator-gene interactions culled from domain-specific literature are an excellent source for prior networks. While some model systems have excellent databases of known interactions, these resources are unavailable for most organisms and cell types [Gibbs et al. 2021]. In these cases, using chromatin accessibility in combination with the known DNA-binding preferences for TFs to identify putative target genes is a viable alternative. Miraldi et al. [2019] generated prior networks by identifying open chromatin regions ± 10 kb of the transcription start sites enriched with TF motifs collected from multiple motif databases. Kamimoto et al. [2020] built on top of this method and further constrained the regulatory regions to be within promoters and enhancers. However, networks generated from these motif-derived prior networks still perform considerably worse than the literature-derived ones [Gibbs et al. 2021]. Automatically generating literature-derived prior network by coupling sentence-based text mining [Han et al. 2015, 2018] with named-entity linking [Hachey et al. 2013] can potentially provide better solutions to this problem from a different angle.

3 | Calling Genetic Variants from Whole Exome Sequencing Data

3.1 Introduction

Next-generation sequencing (NGS) measures a genome by repeated, semi-random sampling of short (76-300bp) fragments that have a 1% base error rate. NGS can be used to sample the whole genome or can attempt to target coverage to the whole exome, the 1-2% of genome which codes for proteins and their bordering regions. Whole exome sequencing (WES) is a cost-effective method for identifying interpretable, causal variants in Mendelian disorders [Bamshad et al. 2011] (Figure 3.1).

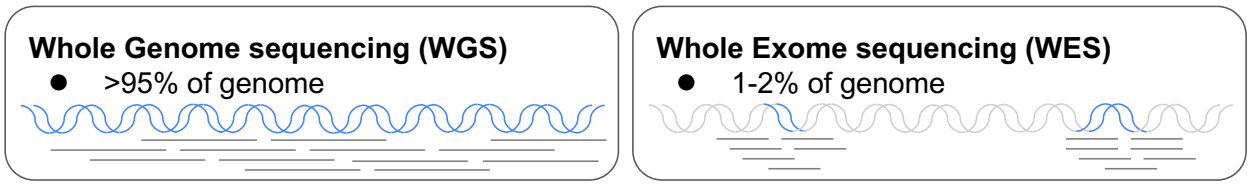


Figure 3.1: Genome coverage comparison between whole genome and whole exome sequencing.

After sequencing, variant calling analyzes these fragments relative to a reference genome to identify the genomic positions that distinguish an individual sample [DePristo et al. 2011a; Garrison and Marth 2012; Luo et al. 2019; Kim et al. 2018]. Machine learning approaches to variant calling [Luo et al. 2019; Poplin et al. 2018a] have demonstrated best-in-class accuracy, benefitting

from training sets created by extensively sequencing the well-characterized GIAB samples [Zook et al. 2016, 2018].

WES must contend with greater sources of error (e.g. variation in capture efficiency and greater GC bias [Meienberg et al. 2016]), and WES samples generate less training data since WES covers less of the genome. Although there are a great deal of publicly available WES data, very few of them are generated on the GIAB truth sets needed for training and evaluating variant calling models. In this work, we investigate approaches that allow machine learning to benefit from the substantially larger body of whole genome sequencing (WGS) training data while retaining specialized learning from WES training data.

We use DeepVariant [Poplin et al. 2018a] as the foundation for this investigation. DeepVariant performs variant calling in four steps: 1) scanning through NGS read alignments to find candidate variants, 2) local reassembly of reads to reference and candidate variant haplotypes, 3) creation of a six-channel pileup image that represents the bases, base quality, mapping quality, strand, and support for reference or variant haplotype over a 221 bp window, and 4) using an InceptionV3 [Szegedy et al. 2016a] deep neural network to predict the genotype at the candidate position (Figure 3.2).

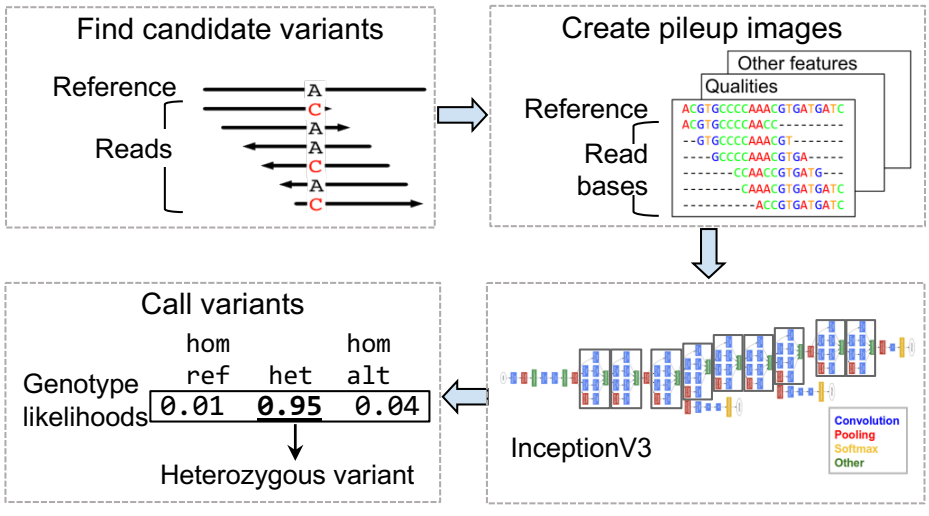


Figure 3.2: The DeepVariant workflow.

	WGS	WES
Train	320, 662, 815	17, 402, 861
Tune	2, 435, 712	631, 261

Table 3.1: The number of examples in DeepVariant production datasets.

Currently, separate DeepVariant models are trained for WGS and WES data, termed WGS and WES models, respectively. However, due to the inherent low region coverage, the exome contains far fewer variants (2×10^5) than the genome (4×10^6), resulting in far fewer training examples from WES data compared to WGS (Table 3.1). Variant calling performance (measured by F1 score) achieved using only the WES data is also considerably lower and less stable than that of the WGS (Figure 3.3).

Figure 3.3: DeepVariant performance using only WES or WGS data. F1 scores are calculated based on single nucleotide polymorphisms (SNPs) and indels (insertions and deletions) prediction accuracy. The experimental dataset used for the comparison is described in Section 3.4.1.

3.2 Related Work

Deep neural networks require large amounts of data to achieve high accuracy in computer vision [Deng et al. 2009; Krizhevsky et al. 2009], natural language processing [Rajpurkar et al. 2016; Williams et al. 2017], and genomics [Consortium et al. 2004; Harrow et al. 2012; Chèneby et al. 2017] tasks. Data augmentation techniques borrow from data-rich problems or generate adversarial examples. Image augmentation generates new examples by adding random noise and

transformations to existing images [Simard et al. 2003; Cubuk et al. 2018]. This process is extended by generative adversarial networks [Goodfellow et al. 2014; Karras et al. 2019; Antoniou et al. 2017a], which are especially useful for highly-skewed data and uncommon cases [Yi et al. 2019b].

Few methods have been proposed to generate adversarial examples for variant calling, as this research topic has been blessed with the abundance of WGS data. For example, the GIAB Consortium ([Zook et al. 2016, 2018]) has provided valuable NGS data as well as high-confidence truth sets for 7 extensively characterized genomes (HG001-GH007). However, method development for specialized problems, such as variant calling from WES and nanopore sequencing data, often experience major hurdles due to low data availability. The incomplete understanding of sequencing error profile and genome content forces strategies to semi-simulate data [Torracinta et al. 2016], but the faithfulness with which these approximate real-world data has not been comprehensively evaluated.

3.3 Methods

We establish the baseline for training WES models–training from WES data alone (*WES Only*), and we consider three training strategies for improving DeepVariant accuracy on WES data (Figure 3.4).

We first investigate two naive strategies for leveraging WGS data for training WES a DeepVariant WES model: 1) training a model from a combination of WGS and WES data (*WGS + WES*, Figure 3.4a), and 2) warmstarting a WES model from a trained WGS model (*warmstart WGS*, Figure 3.4b).

We further introduce an additional low-dimensional vector to DeepVariant to capture sequencing types (*SeqType*, Figure 3.4c). At the DeepVariant feature generation step, a 6-channel pileup image (i.e., feature matrix) p is generated for each candidate variant example. In the *Seq-*

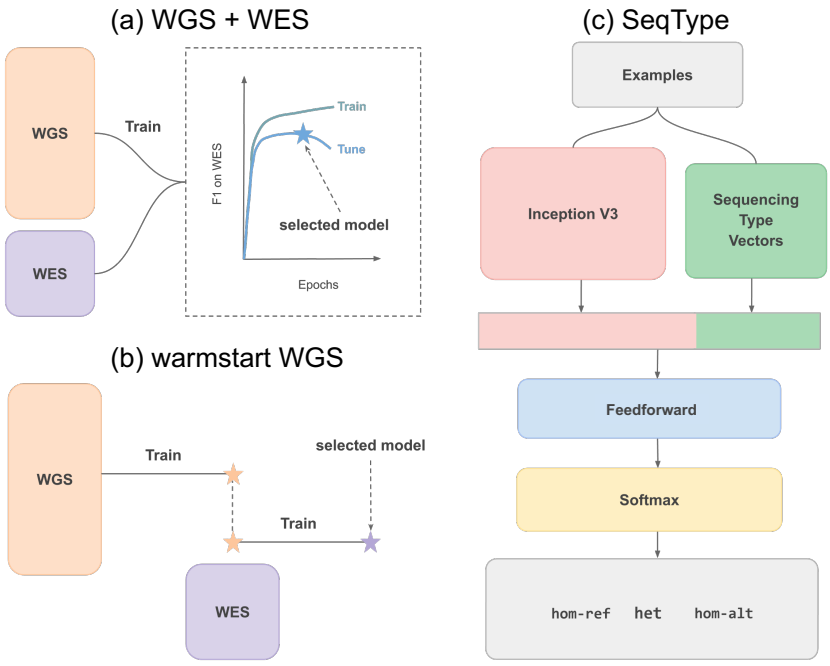


Figure 3.4: Three training strategies for improving DeepVariant accuracy on WES data.

Type approach, we additionally associate each example with its sequencing type vector $s \in \mathbb{R}^m$, which is a randomly initialized for the two data types (WES or WGS). Similar to DeepVariant, p is provided to the InceptionV3 [Szegedy et al. 2016b] to learn the hidden vector μ, the output of the InceptionV3 PreLogit layer. The final feature vector ω is generated by

$$\omega = \mu \oplus \nu \tag{3.1}$$

where $\oplus$ represents the concatenation operation. ω is then provided to a feedforward neural network (FFN) followed by a softmax layer to produce the final predicted genotype probabilities $\hat{y}$. The feedforward network consists of two linear transformations (parameterized by W_1, b_1, W_2, and b_2) with a ReLU activation and a layer normalization [Ba et al. 2016] in between.

$$\text{FFN}(\omega) = \text{LayerNorm}(\max(0, W_1\omega + b_1))W_2 + b_2 \tag{3.2}$$

We set the sequencing type vector dimenstion $m = 200$. The model is trained by minimizing the negative conditional log-likelihood across N training examples.

$$\mathcal{L} = -\sum_{i=1}^{N} \log(\hat{y}_i | p_i, s_i) \tag{3.3}$$

Training loss is minimized using the Adam [Kingma and Ba 2014] optimizer, just like the production DeepVariant model.

3.4 Experimental Design

3.4.1 Data

We use a reduced set of DeepVariant's production dataset to minimize data heterogeneity (Table 3.2). This experimental dataset contains three PCR-free WGS BAM files sequenced on Illumina HiSeq2500 and 18 WES BAM files sequenced on Illumina HiSeq4000.

	WGS	WES
Train	37, 106, 930	2, 641, 013
Tune	1, 024, 080	94, 149

Table 3.2: The number of examples proposed by DeepVariant using the experimental dataset.

The GIAB truth sets [Zook et al. 2016, 2018] provide labels for training and evaluation. We use HG001 samples for training and hold out HG002 for evaluation. This is the same training and evaluation strategy used for DeepVariant. The training set for HG001 is the v3.3.2 truth set, while the evaluation set for HG002 uses the v4-beta truth set newly available for only this sample.

3.4.2 Experimental setup

For each experiment, the checkpoint that achieves the highest F1 score on the tuning set within the first 2 million steps is selected as the best model checkpoint. The experiments are performed on TPUs [Jouppi et al. 2017]. We follow the DeepVariant WES case study[1] to evaluate the model performance using fully held-out HG002 WES sample available from GIAB [Zook et al. 2016]. For each training strategy, the prediction F1 scores reported here are based on 5 replicated training runs using the same parameter configurations and different random seeds. Variant predictions are bootstrapped 100 times. These bootstrap samples are used to perform statistical analyses, and p-values are calculated based on student's t-test.

3.5 Results

We first evaluate two strategies–*WGS + WES* and *warmstart WGS*–for adding training examples from WGS relative to a *WES only* baseline (Figure 3.5). Both strategies improve DeepVariant F1 scores ($p_{\text{WGS + WES}} = 3.3 \times 10^{-173}$, $p_{\text{warmstart WGS}} = 5.4 \times 10^{-153}$). Additionally, the addition of WGS data reduces the variability in model performance across replicated experiments.

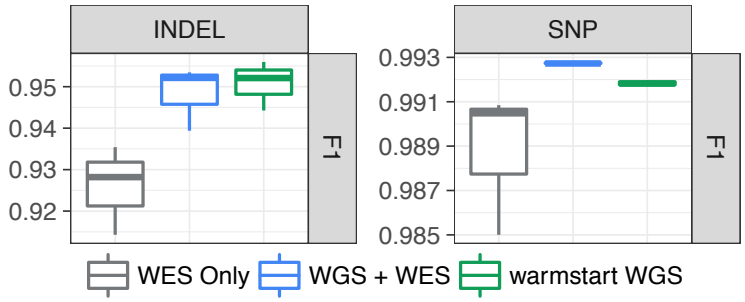


Figure 3.5: Performance of two training strategies for directly adding training examples for WGS. We separately report performance on indels and SNPs, evaluated using the whole exome truth set from sample HG002.

We stratify performance by variant type. SNPs are substitutions that do not change the se-

[1] https://github.com/google/deepvariant/blob/r0.8/docs/deepvariant-exome-case-study.md

quence length, while indels introduce insertions or deletions. Indel variants are harder to accurately predict ([Zook et al. 2016], Figure 3.3), especially in WES due to additional biases in coverage of GC-rich and poor regions [Meienberg et al. 2016]. More importantly, F1 scores achieved by *WGS + WES* are significantly different from *warmstart WGS*, also indicating that calling genetic variants from the WES data presents a different problem compared to that from the WGS data. We additionally randomly selected 10k DeepVariant proposed examples to investigate WGS and WES data distributions (Figure 3.6). WES data contains more high quality reads compared to WGS (Figure 3.6, left). DeepVariant randomly selects 100 reads to construct a pileup image when more than 100 reads are aligned to a particular 221 bp window centered around a particular candidate variant site. Due to the mandatory PCR amplification step in exome sequencing, WES data have significantly more examples that have more than 100 reads mapped to them (Figure 3.6, right). These findings suggest that WGS and WES data exihibit very different sample distributions, and therefore, methods such as SeqType are necessary for capturing sequencing type-specific features.

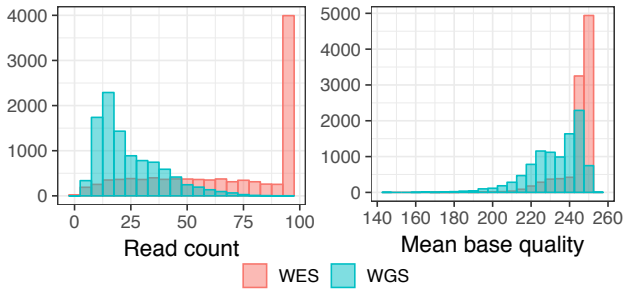


Figure 3.6: Histograms of read count and mean base quality per example, collected from 10k DeepVariant examples.

We then evaluate DeepVariant performance after adding a sequencing type feature vector (Figure 3.5). The *SeqType* model is trained on *WGS + WES* data configuration. *SeqType* significantly improves indels and SNPs F1 scores as opposed to *WES only* ($p_{\text{SeqType}} < 4.1 \times 10^{-288}$). Compared with three other methods, *SeqType* reduces the total number of prediction errors by 6% - 38% on indels (*WGS + WES*: 6%; *warmstart WGS*: 13%; and *WES only*: 38%), and 0.74% - 36%

on SNPs (*WGS + WES*: 0.74%; *warmstart WGS*: 12%; and *WES only*: 36%). We also note a further reduction in the variability of trained model accuracy on indels.

Training strategy	Variant type	F1	Total errors
SeqType	Indel	$\mathbf{0.955 \pm 2.237 \times 10^{-3}}$	$\mathbf{260 \pm 13.0}$
WGS + WES	Indel	$0.948 \pm 7.766 \times 10^{-3}$	299 ± 45.2
warmstart WGS	Indel	$0.951 \pm 5.964 \times 10^{-3}$	277 ± 32.9
WES Only	Indel	$0.926 \pm 1.075 \times 10^{-2}$	418 ± 54.6
SeqType	SNP	$\mathbf{0.99281 \pm 7.314 \times 10^{-5}}$	$\mathbf{538 \pm 5.5}$
WGS + WES	SNP	$0.99275 \pm 3.955 \times 10^{-5}$	542 ± 2.9
warmstart WGS	SNP	$0.99184 \pm 1.342 \times 10^{-4}$	612 ± 10.1
WES Only	SNP	$0.98878 \pm 3.273 \times 10^{-3}$	843 ± 248.1

Table 3.3: Comparing *SeqType* training performance with the *WGS + WES*, *warmstart WGS* and *WES Only*. Mean ± standard deviation of the F1 and total errors reported here are calculated based on replicated experiments using the same parameter configurations and different random seeds.

We further measure performance of each model on progressively harder test sets by randomly downsamping the coverage of the WES samples (Figure 3.7). *WGS + WES* and *warmstart WGS* both outperform *WES only*. *WGS + WES* shows higher SNPs F1 scores across all downsample fractions tested, whereas both *WGS + WES* and *warmstart WGS* remain roughly the same for indels. Adding the sequencing type feature further improves indels F1 scores, while matching or slightly improving SNPs F1 measures. This result is consistent across all downsample fractions.

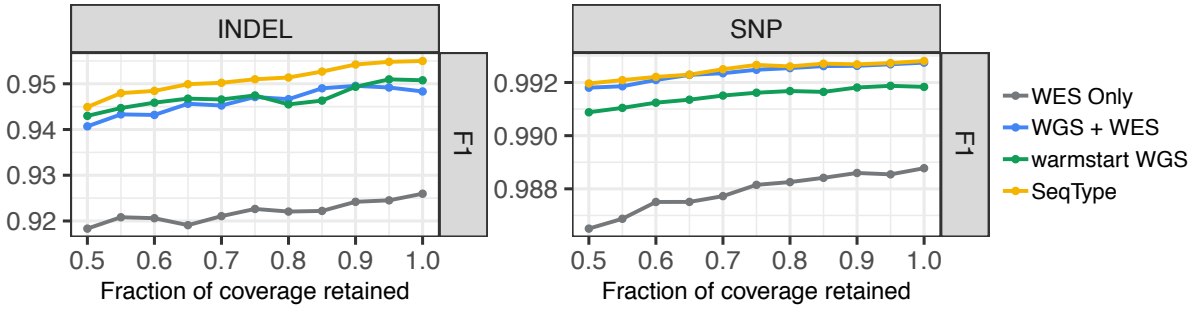


Figure 3.7: Model performances across with different fractions of coverage retained.

3.6 Conclusion

Variant calling has become increasingly beneficial for research and clinical diagnoses [Yang et al. 2014; Xue et al. 2015]. Here we present three data augmentation strategies to improve genetic variant calling from WES data. We show that incorporating WGS data during training by 1) jointly training on WGS and WES data, and 2) warmstarting the WES model from a WGS model improve accuracy on WES data. Since WGS and WES data come from different distributions, we observe further improvements by 3) jointly training on WGS and WES data and including the sequencing type information through a low-dimensional feature vector. This approach shows the most improvement on indels. All three approaches are robust to downsampling and perform well on lower-coverage data.

The sequencing type information can be encoded using fewer dimensions and does not necessarily need to be learned. We experiment with two other variations of the *SeqType* method: 1) trainable vectors of 100 dimensions, and 2) replacing the trainable vectors with constant vectors, where all values are 0 for WGS data or 1 for WES data. Our preliminary results suggest neither of these attempts successfully improves prediction accuracy. These observations indicate it is beneficial to use trainable vectors to distinguish sequencing types as these vectors can potentially learn to encode unique sequencing type features.

The *SeqType* method naturally extends from the concept of embeddings, which refer to a set of representation techniques commonly used in natural language processing [Mikolov et al. 2013; Radford et al. 2019; Devlin et al. 2018a; Liu et al. 2019] and genomics [Asgari and Mofrad 2015a; Gligorijević et al. 2018; Yuan et al. 2019]. Unlike other embedding methods which focus on dimension reduction, *SeqType* vector embeddings are trained to learn abstract features of their corresponding data types. We believe this method can be readily applied to other data augmentation problems. For instance, variant callers trained on Illumina NGS data may not generalize well to Pacific Biosciences data due to their vastly different sequencing and error profiles [Ching et al.

2018]. Despite both being Illumina high-capacity sequencers, HiSeq and NovaSeq reads have noticeably different alignment characteristics. We hypothesize learning sequencer-specific embeddings will be particularly useful in these scenarios, as the embeddings can potentially capture features unique to each sequencing platform.

4 Improving Multitask Transcription Factor Binding Site Prediction with Base-pair Resolution

4.1 Introduction

Genome-wide modeling of non-coding DNA sequence function is among the most fundamental and yet challenging tasks in biology. Transcriptional regulation is orchestrated by transcription factors (TFs), whose binding to DNA initiates a series of signaling cascades that ultimately determine both the rate of transcription of their target genes, and the underlying DNA functions. Both the cell-type-specific chromatin state and the DNA sequence affect the interactions between TFs and DNA *in vivo* [Vaquerizas et al. 2009]. Experimentally determining cell-type-specific TF binding sites is made possible through high-throughput techniques such as chromatin immunoprecipitation followed by sequencing (ChIP-seq) [Johnson et al. 2007a]. Due to experimental limitations, however, it is infeasible to perform ChIP-seq (or related single-TF-focused experiments) on all TFs across all cell types and organisms [Ching et al. 2018]. Therefore, computational methods for accurately predicting *in vivo* TF binding sites are essential for studying TF functions, especially for less well-known TFs and cell types.

Multiple community crowdsourcing challenges have been organized by the DREAM Consor-

tium[1] to find the best computational methods for predicting TF binding sites in both *in vitro* and *in vivo* settings [Weirauch et al. 2013; DREAM 2017]. These challenges also set the community standard for both processing data and evaluating methods. However, top-performing methods from these challenges have revealed key limitations in the current TF binding prediction community. Generalizing predictions beyond the training panels of cell types and TFs can potentially benefit from multitask learning and increased prediction resolution. However, many existing methods still use shallow single-task models. Predictions generated from these methods typically have low resolution, and they cannot achieve competitive performance for prediction or binding regions shorter than 50 base pairs (bp), although the actual TF binding sites are considerably shorter [Stewart et al. 2012].

4.1.1 Related work

Early TF binding prediction methods such as MEME [Bailey 1994; Bailey et al. 2006] focused on deriving interpretable TF motif position weight matrices (PWMs) that characterize TF sequence specificity. Amid rapid advancement in machine learning, however, growing evidence has suggested that sequence specificity can be more accurately captured through more abstract feature extraction techniques. For example, a method called DeepBind [Alipanahi et al. 2015] used a convolutional neural network (CNN) to extract TF binding patterns from DNA sequences. Several modifications to DeepBind subsequently improved model architecture [Hassanzadeh and Wang 2016] as well as prediction resolution [Salekin et al. 2018]. Yuan et al. developed BindSpace, which embeds TF-bound sequences into a common high-dimensional space. Embedding methods like BindSpace belong to a class of representation learning techniques commonly used in natural language processing [Mikolov et al. 2013; Devlin et al. 2018b] and genomics [Asgari and Mofrad 2015b; Yi et al. 2019a] for mapping entities to vectors of real numbers. New methods also explicitly model protein binding sites with multiple binding mode predictors [Gfeller et al. 2011], and

[1] http://dreamchallenges.org/about-dream/

the effect of sequence variants on non-coding DNA functions at scale [Zhou and Troyanskaya 2015; Zhou et al. 2018; Kelley et al. 2018a].

In general, the DNA sequence at a potential TF binding site is just the beginning of the full DNA-function picture, and the state of the surrounding chromosome, the TF and TF-cofactor expressions, and other contextual factors play an equally large role. This multitude of factors changes substantially from cell type to cell type. *In vivo* TF binding site prediction therefore requires cell-type-specific data such as chromatin accessibility and histone modifications. CNN as well as TF- and cell-type-specific embedding vectors have both been used to learn cell-type-specific TF binding profiles from DNA sequences and DNase-seq data [Qin and Feng 2017]. The DREAM Consortium also initiated the ENCODE-DREAM challenge to systematically evaluate methods for predicting *in vivo* TF binding sites [DREAM 2017]. Apart from carefully designed model architectures, top-ranking methods in this challenge also rely on extensively curated feature sets. One such method, called Catchitt [Keilwagen et al. 2019], achieves top performance by leveraging a wide range of features including DNA sequences, genome annotations, TF motifs, DNase-seq, and RNA-seq.

4.1.2 Current limitations

Compendium databases such as ENCODE [Moore et al. 2020] and Remap [Chèneby et al. 2020] have collected ChIP-seq data for a large collection of TFs in a handful of well-studied cell types and organisms [Ching et al. 2018]. Within a single organism, however, the ENCODE TF ChIP-seq collection is highly skewed towards only a few TFs in a small collection of well-characterized cell lines and primary cell types (Figure. 4.1). Knowledge transfer from well-known cell types and TFs are crucial for understanding less-studied cell types and TFs.

Top-performing methods from the ENCODE-DREAM Challenge typically adopt the single-task learning approach. For example, Catchitt [Keilwagen et al. 2019] trains one model per TF

[2] https://www.encodeproject.org/

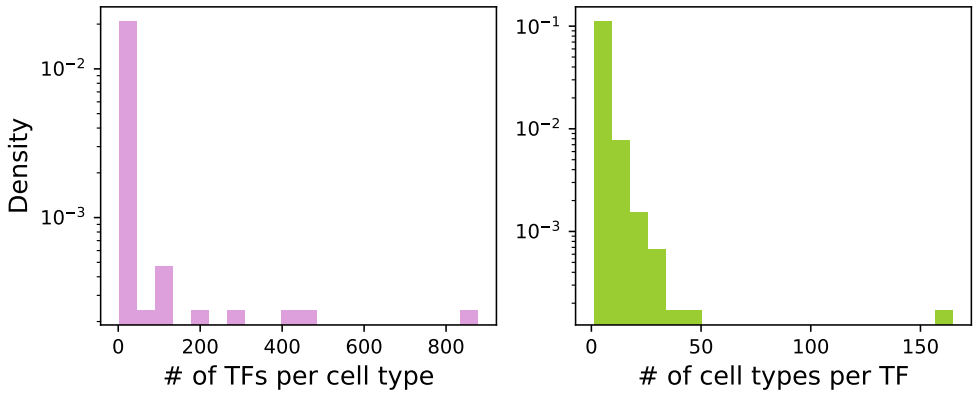


Figure 4.1: ENCODE TF ChIP-seq experiments grouped by TFs and cell types.
Left: the number of TF ChIP-seq experiments per cell type. Right: the number of cell types in which a TF has ChIP-seq experiments. Both histograms skew towards the left, indicating that most cell types only have ChIP-seq data from a small number of TFs and most TFs only have ChIP-seq data in a small number of cell types. Data obtained from ENCODE Consortium[2] and reflects the database status as of December 2020.

and cell type. Cross cell-type transfer predictions are achieved by providing a trained model with input features from a new cell type. This approach can be highly unreliable as the chromatin landscapes between the trained and predicted cell types can be drastically different [Calderon et al. 2019] and these differences can be functionally important [Sijacic et al. 2018]. Alternatively, each model can be trained on one TF using cell-type-specific data across multiple cell types of interest [Quang and Xie 2019]. However, such models tend to assign high binding probabilities to common binding sites among training cell types without proper mechanisms to distinguish cell types. Very few methods have adopted the multitask learning approach in which data from multiple cell types and TFs are trained jointly in order to improve the overall model performance. One multitask solution [Zhou and Troyanskaya 2015; Zhou et al. 2018; Kelley et al. 2018a] involves training a multiclass classifier on DNA sequences, where each class represents the occurrence of binding sites for one TF in one cell type. This solution is suboptimal as it cannot generalize predictions beyond the training TF and cell type pairs.

Sequence context affects TF binding affinity [Siggers and Gordân 2014], and increasing context size can improve TF binding site prediction [Zhou and Troyanskaya 2015]. TF binding sites

are typically only 4-20 bp long [Stewart et al. 2012]; increasing the prediction resolution – the number of predictions a model makes given an input sequence – is therefore beneficial for experimental validation as well as *de novo* motif discovery. An ideal TF binding site prediction strategy therefore involves high context size and high resolution. However, instead of identifying precise TF binding locations, existing methods mainly focus on determining the presence of binding sites. Predictions from these models therefore suffer from either low resolution or low context size, depending on the input sequence length.

In this work, we address the above challenges by introducing NetTIME (Network for TF binding Inference with Multitask-based condition Embeddings), a multitask learning framework for base-pair resolution prediction of cell-type-specific TF binding sites. NetTIME jointly trains multiple cell types and TFs, and effectively distinguishes different conditions using cell-type-specific and TF-specific embedding vectors. It achieves base-pair resolution and accepts input sequences up to 1 kb.

4.2 Approach

4.2.1 Feature and label generation

The ENCODE Consortium has published a large collection of TF ChIP-seq data, all of which are generated and processed using the same standardized pipelines [Moore et al. 2020]. We therefore collect our TF binding target labels from ENCODE to minimize data heterogeneity. Each replicated ENCODE ChIP-seq experiment has two biological replicates, from which two sets of peaks – conserved and relaxed – are derived; peaks in both sets are highly reproducible between replicates [ENCODE 2020].

Compared to the relaxed peak set, the conserved peak set is generated with a more stringent threshold, and is generally used to provide target labels. However, the conserved peak set

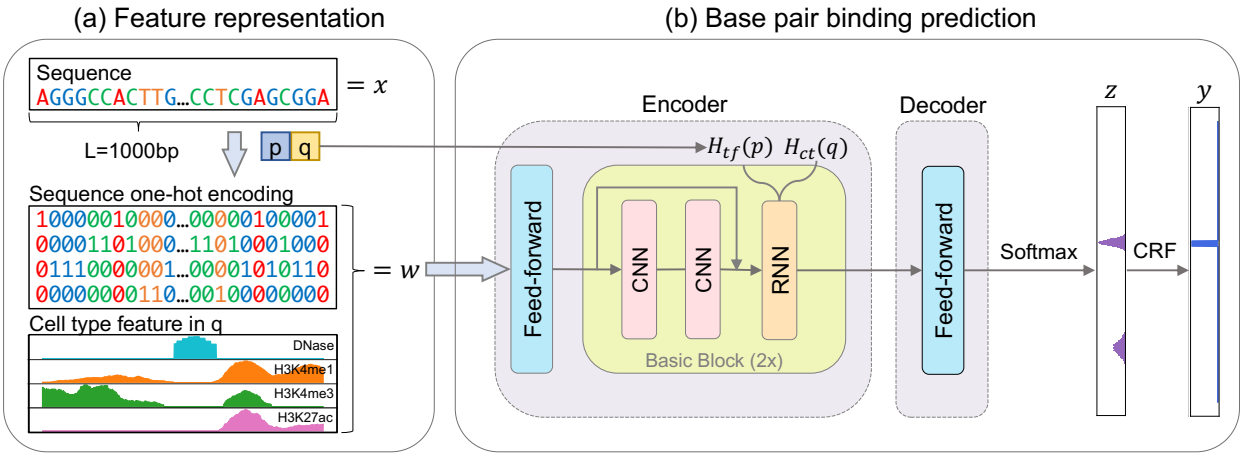


Figure 4.2: Schematic method overview.
(a) Constructing feature vector w from input sequence x, TF label p and cell type label q. w consists of the sequence one-hot encoding, and a set of cell-type-specific features – DNase-seq signals (cyan), and H3K4me1 (orange), H3K4me3 (green) and H3K4ac (pink) histone ChIP-seq signals – in cell type q.
(b) Feature vector w, TF label p and cell type label q are provided to the NetTIME neural network to predict base-pair resolution binding probability z. An additional CRF classifier is trained to predict binary binding event y from z.

usually contains too few peaks to train the model efficiently. Therefore, we use both conserved and relaxed peak sets to provide target labels for training, and the conserved peak set alone for evaluating model performance.

To collect target labels that cover a wide range of cellular conditions and binding patterns, we first select 7 cell types. The 7 cell types include 3 cancer cell types, 3 normal cell types and 1 stem cell type. The 22 TFs include 17 TFs from 7 TF protein families as well as 5 functionally related TFs. Conserved and relaxed peak sets are collected from 71 ENCODE replicated ChIP-seq experiments conducted on our cell types and TFs of interest. Each of these TF ChIP-seq experiments is henceforth referred to as a condition. All peaks from these conditions form a set of information-rich regions where at least one TF of interest is bound. We generate samples by selecting non-overlapping L-bp genomic windows from this information-rich set, where L is the context size. We set the context size $L = 1000$ as it was previously shown to improve TF binding prediction performance [Zhou and Troyanskaya 2015].

In vivo TF binding sites are affected by DNA sequences and the cell-type-specific chromatin landscapes. In addition to using DNase-seq, which maps chromatin accessibility, we collect ChIP-seq data for 3 types of histone modifications to form our cell-type-specific feature set. The histone modifications we include are H3K4me1, H3K4me3 and H3K27ac, which are often associated with enhancers [Rada-Iglesias 2018], promoters [Benayoun et al. 2014] and active enhancers [Creyghton et al. 2010], respectively.

4.2.2 Methods

NetTIME performs TF binding predictions in three steps: 1) generating the feature vector $\mathbf{w} = (w_1, \cdots, w_L)$ given a TF label p, a cell type label q, and a sample DNA sequence $\mathbf{x} = (x_1, \cdots, x_L)$ where each $x_l \in \{A, C, G, T\}$, 2) training a neural network to predict base pair resolution binding probabilities $\mathbf{z} = (z_1, \cdots, z_L)$, and 3) converting binding probabilities to binary binding decisions $\mathbf{y} = (y_1, \cdots, y_L)$ of p in q by either setting a probability threshold or additionally training a linear-chain conditional random field (CRF) classifier (Figure 4.2).

4.2.2.1 Feature representation

We construct the feature vector $\mathbf{w} \in \mathbb{R}^{K \times L}$ from $\mathbf{x} \in \mathbb{R}^{L}$, where K represents the number of features. Different types of features are independently stacked along the first dimension. For each element in $\mathbf{w}$, w_l is the concatenation of the one-hot encoding of the DNA sequence $O(x_l)$, and the cell-type-specific feature $C(x_l)$ (Figure 4.2a).

$$\forall l \in [1, L], \; w_l = \begin{bmatrix} O(x_l) \\ C(x_l) \end{bmatrix} \tag{4.1}$$

High-dimensional embedding vectors can be trained to distinguish different conditions as well as implicitly learning condition-specific features, and are therefore preferred by many machine

learning models over one-dimensional condition labels [Yi et al. 2019a; Qin and Feng 2017; Yuan et al. 2019]. Given TF label p and cell type label q, NetTIME learns the TF- and cell-type-specific embeddings $H_{tf}(p) \in \mathbb{R}^d$ and $H_{ct}(q) \in \mathbb{R}^{d'}$, where $d = d' = 50$.

4.2.2.2 Binding probability prediction

NetTIME adopts an encoder-decoder structure similar to that of neural machine translation models [Sutskever et al. 2014; Cho et al. 2014b; Vaswani et al. 2017] (Figures 4.2b, A.1):

Encoder: the model encoder maps the input feature $\mathbf{w}$ to a hidden vector $\mathbf{h} \in \mathbb{R}^{2d \times L}$. The main structure of the encoder, called the Basic Block, consists of a CNN followed by a recurrent neural network (RNN). CNN uses multiple short convolution kernels to extract local binding motifs, whereas bi-directional RNN is effective at capturing long-range TF-DNA interactions. We choose the ResBlock structure introduced by ResNet [He et al. 2016] as our CNN, as it has become a standard approach for training deep neural networks [Vaswani et al. 2017; Huang et al. 2018]. Traditional RNNs are challenging to train due to the vanishing gradient problem [Hochreiter and Schmidhuber 1997]. We therefore use the bi-directional gated recurrent unit (bi-GRU) [Cho et al. 2014a], a variant of RNN proposed to address the above challenge. The hidden state of bi-GRU is initialized by concatenating the embedding vectors $H_{tf}(t)$ and $H_{ct}(c)$.

Decoder: the model decoder converts the hidden vector $\mathbf{h}$ to binding probabilities $\mathbf{z}$. The conversion is achieved through a fully connected feedforward neural network (FFN), as the relationship between $\mathbf{h}$ and $\mathbf{z}$ may not be trivial. A softmax function subsequently transforms the decoder output to the binding probabilities.

4.2.2.3 Training

We train the model by minimizing the negative conditional log-likelihood of $\mathbf{z}$:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L}\log\ p(z_l^n|x_l^n, p, q) \tag{4.2}$$

where N denotes the number of training samples. The loss function is optimized by the Adam optimizer [Kingma and Ba 2014].

4.2.2.4 Binding event classification

Binary binding events $\mathbf{y}$ can be directly derived from $\mathbf{z}$ by setting a probability threshold $b \in (0, 1)$ such that

$$\forall l \in [1, L],\ y_l = \begin{cases} 1, & \text{if } z_l \geq b \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

Alternatively, a linear-chain CRF classifier can be trained to achieve the same goal. It computes the conditional probability of $\mathbf{y}$ given $\mathbf{z}$, defined as the following:

$$p(\mathbf{y}|\mathbf{z}) = \frac{1}{Z(\mathbf{z})}exp(\sum_{l=1}^{L}(z_l)_{y_l} + \sum_{l=1}^{L}V_{y_l,y_{l+1}}) \tag{4.4}$$

where

1. $Z(\mathbf{z})$ is a normalization factor,
2. $V \in \mathbb{R}^{p\times p}$ is a transition matrix, where p denotes the number of classes of the classification problem and each $V_{i,j}$ represents the transition probability from class label i to j,
3. $\sum_{l=1}^{L}(z_l)_{y_l}$ calculates the likelihood of y_l given z_l, and

4. $\sum_{l=1}^{L} V_{y_l, y_{l+1}}$ measures the likelihood of $y_l + 1$ given y_l.

In CRF, the class label at position l affects the classification at position $l+1$ [Sutton and McCallum 2010]. This is potentially beneficial for TF binding site classification as positions adjacent to a putative binding site are also highly likely to be occupied by TFs. We train the CRF by minimizing $-\log p(\mathbf{y}|\mathbf{z})$ over all training samples. The Adam optimizer [Kingma and Ba 2014] is used to update the parameter V.

4.2.3 Model selection

We follow the guideline provided by the ENCODE-DREAM Challenge [DREAM 2017] to perform data split as well as model selection whenever possible. Training, validation and test data are split according to chromosomes (Supplementary Table A.1). We use the Area Under the Precision-Recall curve (AUPR) score to select the best neural network model checkpoint.

To access how well our model predictions recover the positive binding sites in the truth target labels, we evaluate classifiers' performance according to Intersection Over Union (IOU) score. Suppose P and T are sets of predicted and target binding sites, respectively. Then

$$IOU = \frac{P \cap T}{P \cup T} \tag{4.5}$$

We test 300 random probability thresholds and select the best threshold, i.e., the threshold that achieves the highest IOU score in the validation set. We also train a CRF using predictions generated from the best neural network checkpoint. The best CRF checkpoint is selected according to average loss on the validation set. Model performance reported here is evaluated using the test set.

4.3 Results

4.3.1 Multitask learning improves performance by increasing data availability.

NetTIME can be trained using data from a single condition (single-task learning) or multiple conditions (multitask learning). Jointly training multiple conditions potentially allows the model to use data more efficiently and improves model generalization [Caruana 1997]. We therefore evaluate the effectiveness of multitask learning when jointly training multiple related conditions. For this analysis, we choose three TFs from the JUN family that exhibit overlapping functions: JUN, JUNB and JUND [Mechta-Grigoriou et al. 2001]. Combining multiple cell types of JUND allows the multitask learning model to significantly outperform the single-task learning models, each of which is trained with one JUND condition (Figure 4.3a). Jointly training multiple JUN family TFs further improves performance compared to training each JUN family TF separately (Figure 4.3b). However, we do not observe improved performance when subsampling the multitask models' training data to match the number of samples in the corresponding single-task models (Figure 4.3).

This indicates that the multitask learning strategy is more efficient due to the increased data available to multitask models rather than to the increased data diversity. Similar results are also observed when the same analysis is performed on three unrelated TFs (Figure 4.4).

4.3.2 Supervised predictions made by NetTIME achieves superior performance

Our feature set includes DNA sequence, and cell-type-specific features including DNase-seq and three types of histone ChIP-seq. In practice, however, data for these features are not always

Figure 4.3: Performance comparison between multitask learning and single-task learning approaches using JUN family TFs.
Models are trained with datasets from **(a)** JUND across multiple cell types, and **(b)** multiple TFs in the JUN family across multiple cell types. MTL: multitask learning; MTL-sampled: multitask learning training data that has been subsampled to match the number of samples in the corresponding single-task models; STL: single-task learning. The right panels in (a) and (b) are the averaged AUPR of the models shown in the corresponding left panels. Error bars represent standard error of the mean across all training conditions.

available for the conditions of interest. Additionally, TF motif enrichment has often been used by existing methods to provide TF binding sequence specificity information [Keilwagen et al. 2019; Quang and Xie 2019]. We therefore evaluate the quality of our model predictions when we vary the types of input features available during training.

We first train separate models after removing cell-type-specific features using training data from all conditions mentioned in Section 4.2.1. Model prediction accuracy is evaluated in the supervised fashion using test data from the same set of conditions. The addition of cell type features significantly improves NetTIME performance. However, adding TF motif enrichment features (Section A.1.1.1), either in addition to DNA sequence features or in addition to both sequence and cell type features, reduces prediction accuracy (Table 4.1). Despite exhibiting high sequence

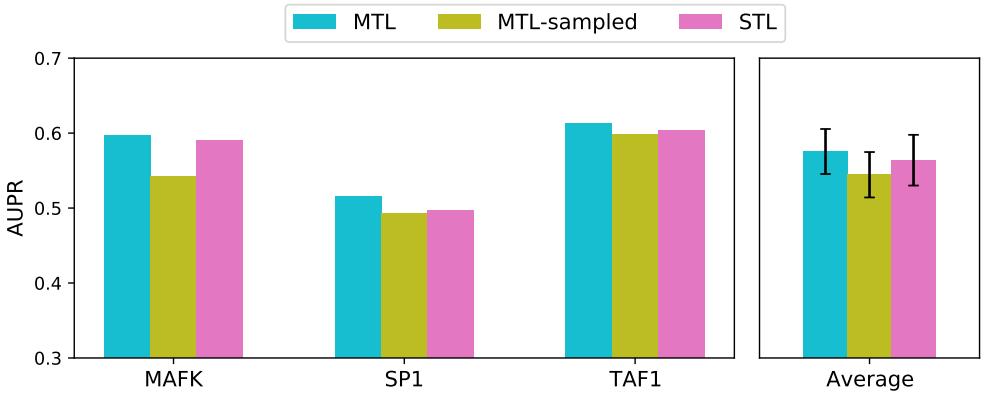


Figure 4.4: Performance comparison between multitask learning and single-task learning approaches using three functionally unrelated TFs.
Models are trained with data from MAFK, SP1 and TAF1 across multiple cell types. Left panel shows the average performance measured by AUPR across multiple cell types of the same TF. Right panel averages performance across multiple TFs shown in the left panel. We observe marginal benefit to using MTL strategy over STL when training unrelated TFs. However, the average AUPR score achieved by MTL-sampled is noticeably lower compared to STL. This additional analysis suggests that the relatedness among jointly trained conditions plays an important role in determining the effectiveness of multitask learning models. Increased data availability improves model generalization, although too much data heterogeneity reduces model predictability.

specificity *in vitro*, TF binding sites *in vivo* correlate poorly with TF motif enrichment [Chen et al. 2017]. Motif qualities in TF motif databases vary significantly depending on available binding data and motif search algorithms. Nevertheless, TF motifs have been the gold standard for TF binding site analyses due to their interpretability and scale. TF motif enrichment features are likely redundant when our model can effectively capture TF binding sequence specificity, though it's possible our protocol for generating TF motif enrichment features is suboptimal.

We also compare the performance of NetTIME with that of DeepBind, BindSpace and Catchitt. Given only DNA sequences as the input feature, NetTIME significantly outperforms DeepBind and BindSpace. NetTIME also doubles the AUPR score achieved by Catchitt when all three types of features are used for training (Table 4.1).

The AUPR scores for DeepBind and BindSpace are significantly lower than those reported in the original manuscripts. One possible reason is that Table 4.1 reports model performance under

	Seq	Seq + TF	Seq + CT	Seq + CT + TF
DeepBind	0.025±0.02	NA	NA	NA
BindSpace	0.035±0.02	NA	NA	NA
Catchitt	NA	NA	NA	0.260±0.14
NetTIME	0.384±0.15	0.378±0.15	**0.534±0.16**	0.525±0.15

Table 4.1: Comparing supervised prediction performance for DeepBind, BindSpace, Catchitt and NetTIME evaluated at 1 bp resolution.
NetTIME models are trained separately under different feature settings. Seq: DNA sequence features; CT: cell type-specific features including DNase-seq, and H3K4me1, H3K4me3 and H3K27ac histone ChIP-seq data; TF: TF-specific features containing the HOCOMOCO TF motif enrichment scores for plus and minus strands. We report here the mean±standard deviation of the AUPR scores across all training conditions.

1 bp resolution (Supplementary section A.1.1.2), although none of these competing methods truly achieves base-pair resolution.

To avoid being biased towards NetTIME, we additionally compare method performance after reducing our prediction resolutions to match those of other methods. For each 1000 bp input sequence, we first divide NetTIME predictions into n-bp bins ($1 \leq n \leq 1000$). The binding probability of a particular bin is subsequently obtained by taking the maximum binding probabilities across all positions within that bin. We set the bin width $n = 20, 50, 100, 200, 500, 1000$. NetTIME maintains higher AUPR scores across all bin widths tested (Figure 4.5). We observe a consistent increase in prediction AUPR scores for DeepBind, BindSpace and NetTIME as we reduce the prediction resolutions. However, Catchitt achieves the highest AUPR score at $n = 50$. This is the same bin width Catchitt used for the ENCODE-DREAM challenge, raising the possibility that Catchitt overfits parameters for a particular bin width of interest.

4.3.3 TF-specific and cell-type-specific embeddings are crucial for effective multitask learning strategy.

Although NetTIME outperforms several existing methods, we have yet to dissect the contributions of different components to our predictive accuracy. We use the TF and cell type embedding

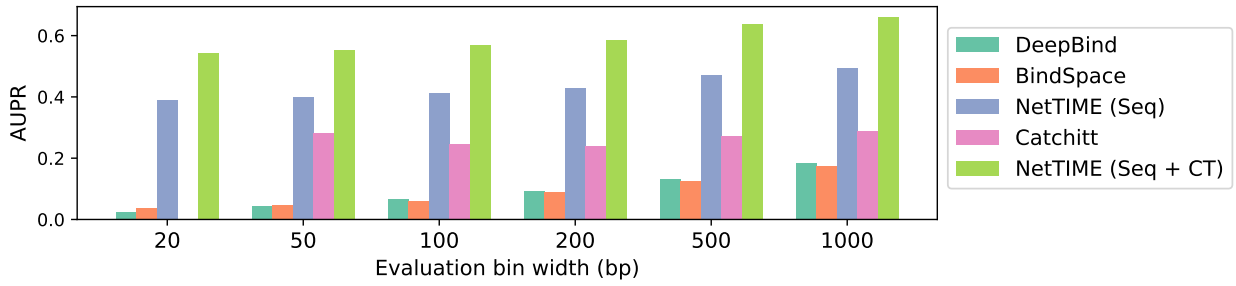


Figure 4.5: Supervised performance comparison of DeepBind, BindSpace, Catchitt and NetTIME evaluated at different bin widths.
The Catchitt AUPR score at 20 bp bin width is set to 0 as the the program exited with error when given a 20bp-long input sequence.

vectors to learn condition-specific features and biases, and a combination of CNNs and RNNs to learn the non-condition-specific TF-DNA interaction patterns. TF and cell type embedding vectors can be replaced with random vectors at prediction time and training time to evaluate the contribution of each component individually.

To evaluate the model's sensitivity to different TF and cell type labels, TF and cell type embedding vectors are replaced with random vectors at prediction time (Table 4.2). Substituting both types of embeddings with random vectors reduces our model performance by 54.4% on average. Although replacing either TF or cell type embeddings with random vectors drastically reduces AUPR scores, the performance drop is more significant for cell type embeddings. This indicates that cell type-specific chromatin landscape features are more important for defining *in vivo* TF binding sites, which explains the redundancy of TF motif features and the lack of correlation between TF ChIP-seq signals and TF motif enrichment mentioned in Section 4.3.2 and Chen et al. [2017].

We additionally swap both types of embedding vectors for random vectors at training time to remove the condition-specific component, which results in a 2% drop in the mean AUPR score across all training conditions (Fig 4.6a). This suggests that, while embedding vectors are important for learning TF and cell type specificity, the network components for learning common binding patterns among TFs and cell types are also crucial for maintaining high prediction accu-

	TF embeddings	
Cell type embeddings	Random	Trained
Random	0.244 ± 0.16	0.409 ± 0.16
Trained	0.310 ± 0.18	0.535 ± 0.15

Table 4.2: Evaluating the contribution of condition-specific network components.
Trained TF and cell type embedding vectors (Trained) are replaced by random vectors (Random) at prediction time.

racy.

Visualizing the trained TF embedding vectors in two dimensions using t-SNE [Van der Maaten and Hinton 2008] reveals that a subset of embedding vectors also reflect the TF functional similarities. Some TFs that are in close proximity in t-SNE space are from the same TF families, including FOXA1 and FOXA2, HNF4A and HNF4G, STAT1 and STAT3, ATF3 and ATF7, and JUN, JUNB and JUND (Figure 4.6b, solid circles). Functionally related TFs including IRF3 and STAT1 [Mogensen 2019] are also adjacent to each other in t-SNE space (Figure 4.6b, dashed circle). However, these TF embedding vectors are explicitly trained to learn the biases introduced by TF labels. Available data for TFs of the same protein family are not necessarily from the same set of cell types. As a result, not all functionally related TFs are close in t-SNE space, such as IRF (IRF3, IRF4 and IRF5) family proteins and TFs associated with c-Myc proteins (MAX and MAZ).

4.3.4 TF and cell type embeddings allow more reliable transfer predictions.

Transfer learning allows models to make cross-TF and cross-cell type predictions beyond training conditions. Most existing methods achieve transfer learning by providing input features from a new cell type to a model trained on a different cell type. If multiple trained cell types are available for the same TF, the final cross-cell type predictions are generated by averaging predictions from all trained cell types (Average Trained). Suppose we wish to make transfer predictions for TF p in cell type q, denoted $[p, q]$. This approach allows us to take advantage of all available data for

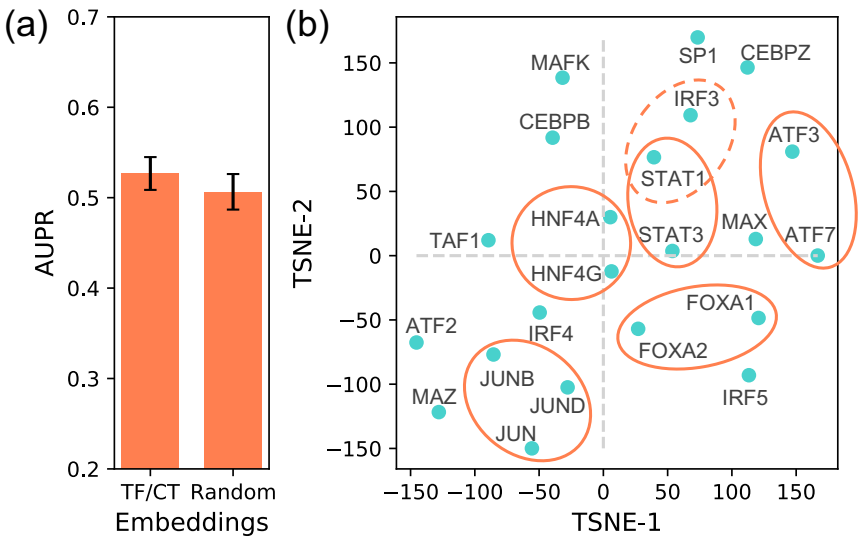


Figure 4.6: Properties of trained embedding vectors.
(a) Evaluating the effect of replacing TF and cell type embeddings (TF/CT) with random vectors (Random) at training time.
(b) t-SNE visualization of the TF embedding vectors. Orange circles indicate related TFs that are in close proximity in t-SNE projection space: solid circles illustrate TFs from the same protein family, and dashed circles illustrate TFs having similar functions.

TF p ($[p, *]$). However, the TF and cell type embedding approach (Embedding Transfer) allows our model to leverage all available data for both TF p and cell type q ($[p, *] \cup [*, q]$). Since the multitask learning paradigm benefits from having more training data (Section 4.3.1) and *in vivo* TF binding sites mostly correlate with cell type-specific features (Section 4.3.3), the latter approach can potentially improve our model's transfer predictive performance.

To evaluate the prediction quality of these two approaches, we pretrain a NetTIME model by leaving out 10 conditions for transfer learning. For each transfer condition $[p, q]$, we use the pretrained model to derive both the Average Trained and the Embedding Transfer predictions. Transfer learning predictions are generally less accurate compared to supervised predictions (Supervised). However, transfer predictions generated by Embedding Transfer still significantly outperform those of the Average Trained (Figure 4.7a). Transfer predictions derived from NetTIME also achieves considerably higher accuracy compared to those from Catchitt (Figure 4.7b). Aver-

age Trained and Embedding Transfer predictions can also be obtained after fine-tuning the pretrained model with data from $[p, *] \setminus [p, q]$ and $[p, *] \cup [*, q] \setminus [p, q]$, respectively. This fine-tuning step additionally introduces marginal improvement measured by average AUPR score across all leave-out conditions (Figure 4.7c).

Using trained TF and cell type embeddings allows models to perform binding predictions beyond the training panels of TFs and cell types. We therefore test our model's robustness when making predictions on unknown conditions using 6 conditions from 6 new TFs in 3 new cell types. Starting from a NetTIME model pretrained on all original training conditions (Section 4.2.1), we fine-tune the pretrained model for each transfer condition $[p', q']$ by collecting available samples from $[p', *] \cup [*, q'] \setminus [p', q']$. Transfer predictions generated from models trained with TF and cell type embeddings significantly outperform those from models trained with random embeddings that cannot distinguish different TF and cell type identities (Figure 4.8a). TF binding motifs derived from predicted binding sites also show a strong resemblance to those derived from conserved ChIP-seq peaks (Figure 4.8b).

4.3.5 A CRF classifier post-processing step effectively reduces prediction noise.

Summarizing the binding strength, or probability, along the chromosome at each discrete binding site is an important step for several downstream tasks ranging from visualization to validation. To make binary binding decisions from binding probability scores, we first test the predictive performance of 300 probability thresholds and find that at threshold 0.143, our model achieves the highest IOU score of 35.6% on the validation data (Figure 4.9).

We alternatively train a CRF classifier, as a manually selected probability threshold is poorly generalizable to unknown datasets. These two approaches achieve similar predictive performance as evaluated by IOU scores (Figure 4.10a). However, prediction noises manifested as high prob-

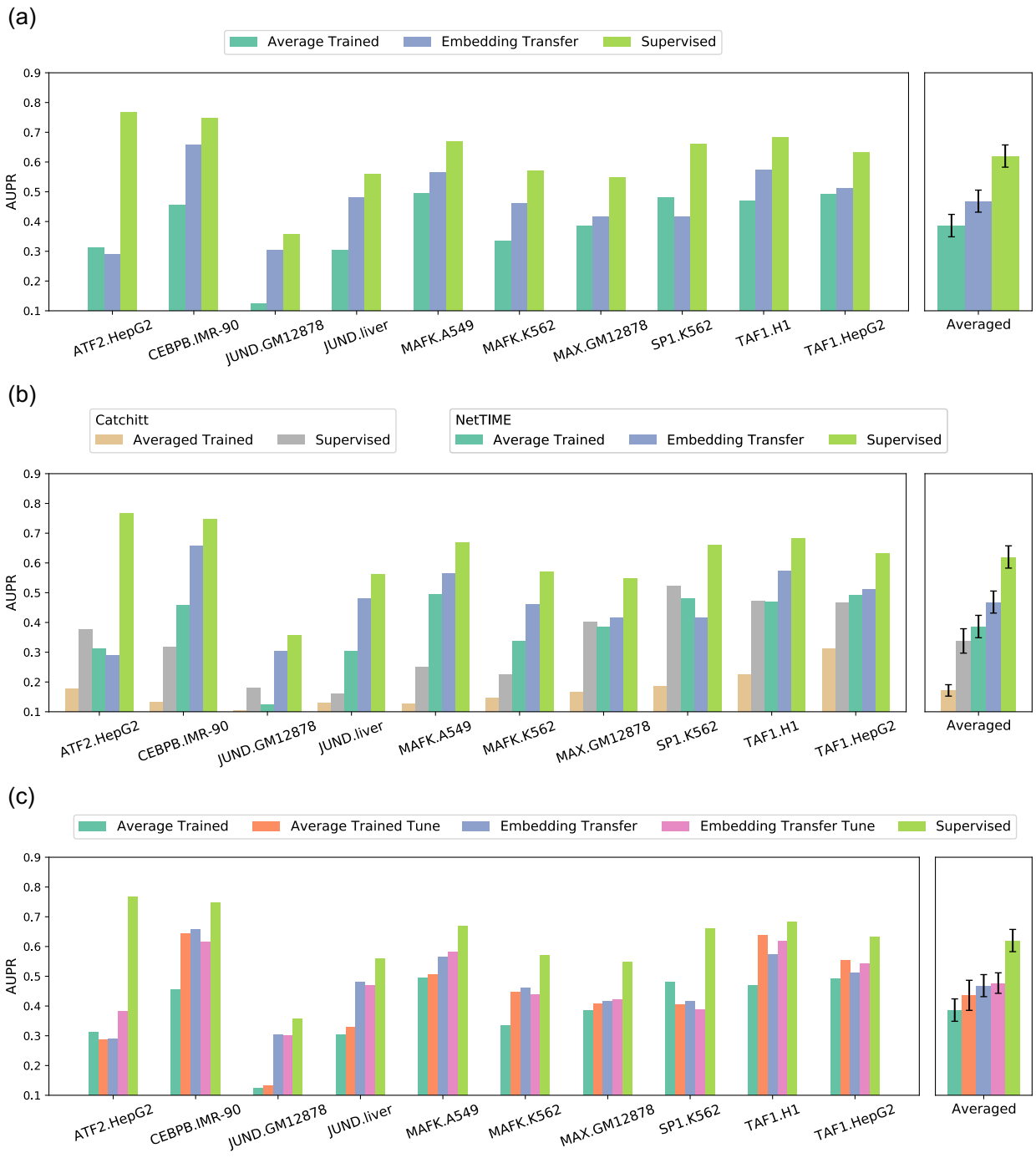


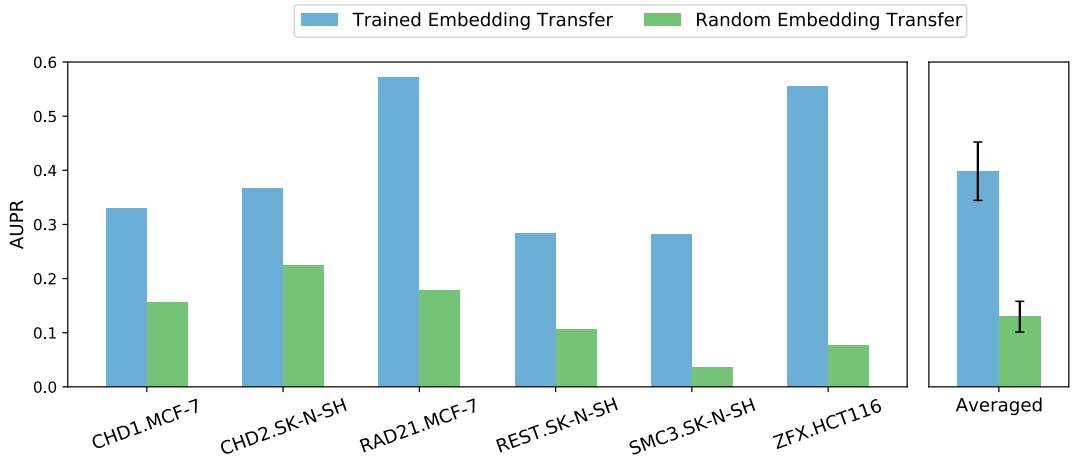
Figure 4.7: Transfer learning with NetTIME using 10 leave-out conditions within the training set of conditions.

(Continue on next page.)

Figure 4.7: (Continued from previous page.)
(a) Comparing the prediction efficiency of two transfer learning strategies. Both types of transfer predictions are derived from a pretrained model trained without the 10 leave-out conditions shown on the x-axis. We denote each leave-out condition as $[p, q]$, where p denotes the TF label and q denotes the cell type label. Average Trained: for each $[p, q]$, we first generate transfer predictions using trained embedding vectors from a different cell type r, where $[p, r]$ refers to any condition containing TF p in the training set of conditions. Multiple such transfer predictions are then averaged to predict the final transfer predictions for condition $[p, q]$. Embedding Transfer predictions are generated using trained embedding vectors of p and q.
(b) Comparing transfer learning accuracy of Catchitt and NetTIME. NetTIME Average Train and Embedding Transfer predictions are generated using procedures described in (a). Transfer learning in Catchitt is achieved using the Average Trained method, where input features for the transfer conditions are given to models trained on different cell types of the same TF. The final prediction probabilities are derived by averaging all models' predictions if multiple trained models exist. Transfer predictions derived from Catchitt are significantly less accurate compared to NetTIME. Catchitt's supervised predictions achieve lower average AUPR scores compared to transfer predictions obtained from both NetTIME transfer learning strategies.
(c) Comparing transfer learning accuracy in pretrained and fine-tuned models. Average Trained and Embedding Transfer predictions are generated using procedures described in (a). Average Trained Tune and Embedding Transfer Tune: transfer predictions derived after fine-tuning the pretrained model with relevant datasets. Supervised: supervised predictions derived from a model where training data from the above 10 conditions are included during training. Marginal improvements are observed for both transfer learning approaches after the fine-tuning step. However, the Embedding Transfer approach consistently outperforms the Average Trained regardless of fine-tuning.

ability spikes are likely to be classified as bound using the probability threshold approach. To evaluate the effectiveness of reducing prediction noises using the probability threshold and the CRF approaches, we calculate the percentage of class label transitions per sequence within the target labels and within each of the predicted labels generated by these two approaches. The transition percentage using CRF is comparable to that of the true target labels, and is also significantly lower than the percentage obtained using the probability threshold approach. This indicates that CRF is more effective at reducing prediction noise, and therefore CRF predictions exhibit a higher degree of resemblance to target labels.

(a)



(b)

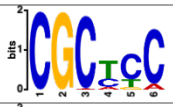
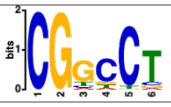
Condition	Target	Prediction	Similarity score
CHD1.MCF-7			2.1e-2
CHD2.SK-N-SH			6.8e-4
SMC3.SK-N-SH			1.5e-16
RAD21.MCF-7			8.5e-28
REST.SK-N-SH			1.5e-14
ZFX.HCT116			1.1e-2

Figure 4.8: Transfer learning with NetTIME using 6 conditions beyond the training set of conditions.
(a) Transfer predictions using models trained with either TF and cell type embedding vectors (Trained Embedding Transfer) or random vectors (Random Embedding Transfer).
(b) Comparison of *de novo* discovered motifs derived from transfer predictions and from target ChIP-seq conserved peaks. Predicted motifs are derived from Trained Embedding Transfer predictions using data from 6 conditions beyond the original training panels of TFs and cell types. *De novo* motif discovery is conducted using STREME [Bailey 2020] software. Motif similarity p-values shown in the top right corner of the Prediction column are derived by comparing predicted and target motifs using TOMTOM [Gupta et al. 2007].

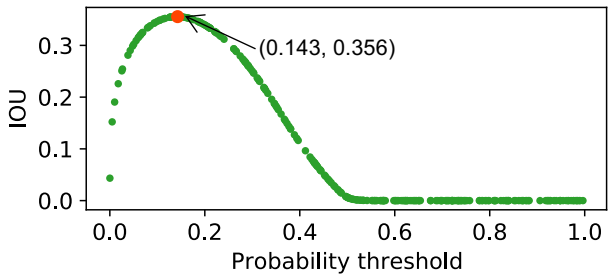


Figure 4.9: Testing binary prediction performance with 300 probability thresholds.
All thresholds are chosen randomly within the interval $[0, 1]$. Performance is evaluated using the IOU score; the predictions achieve the highest IOU score of 0.356 on validation data when we set the threshold at 0.143.

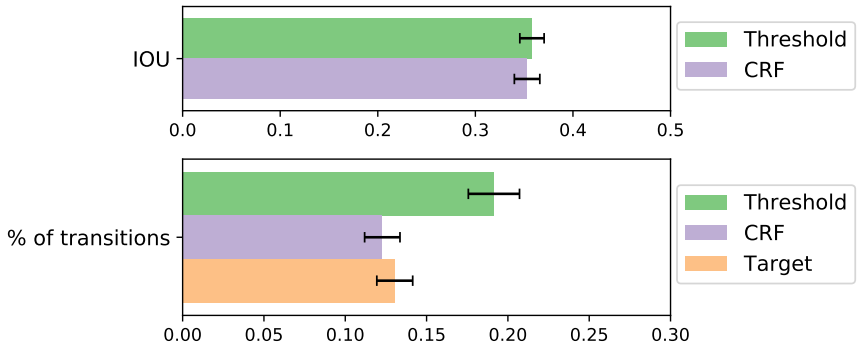


Figure 4.10: Binary classification performance using the probability threshold and CRF.
Performance evaluated by mean IOU score (top) and the percentage of class label transitions per sequence (bottom), both calculated over all training conditions.

4.4 Conclusions

In this work we address several challenges facing many existing methods for TF binding site predictions by introducing a multitask learning framework, called NetTIME, that learns base-pair resolution TF binding sites using embeddings. We first show that the multitask learning approach improves our prediction accuracy through increasing the amount of data available to the model. Both the condition-specific and non-condition-specific components in our multitask framework are important for making accurate condition-specific binding predictions. The use of TF and cell type embedding vectors additionally allows us to make accurate transfer learning

predictions within and beyond the training panels of TFs and cell types. Our method also significantly outperforms previous methods under both supervised and transfer learning settings, including DeepBind, BindSpace and Catchitt.

Although DNA sequencing currently can achieve base-pair resolution, the resolution of ChIP-seq data is still limited by the size of DNA fragments obtained through random clipping. A considerable fraction of the fragments are therefore false positives, whereas many transient and low affinity binding sites are missed [Park 2009]. Additionally, ChIP-seq requires suitable antibodies for proteins of interest, which can be difficult to obtain for rare cell types and TFs. Alternative assays have been proposed to improve data resolution [Rhee and Pugh 2011; He et al. 2015; Rossi et al. 2018] as well as to eliminate the requirement for antibodies [van Steensel and Henikoff 2000; Southall et al. 2013]. However, datasets generated from these techniques are rare or missing in data consortiums such as ENCODE [Moore et al. 2020] and ReMap [Chèneby et al. 2020]. Base-pair resolution methods for predicting binding sites from these assays [Salekin et al. 2018; Avsec et al. 2021b] have largely been limited to characterizing TF sequence specificity. NetTIME can potentially provide base pair resolution solutions to more complex DNA sequence problems as labels generated from these alternative assays become more widely available in the future.

ATAC-seq (Assay for Transposase-Accessible Chromatin using sequencing [Buenrostro et al. 2013]) has overtaken DNase-seq as the preferred assay to profile chromatin accessibility, as it requires fewer steps and input materials. However, these two techniques each offer unique insights into the cell type-specific chromatin states [Calviello et al. 2019], and it is therefore potentially beneficial to incorporate both data types for TF binding predictions. In fact, extensive feature engineering has been the focus of many recent *in vivo* TF binding prediction methods [Chen et al. 2017; Quang and Xie 2019; Keilwagen et al. 2019]. It is also important to note that, without strategies for handling missing features, increasing feature requirements significantly restricts models' scope of application (Figure 4.1). Comprehensive evaluation of data imputation methods [Troyanskaya et al. 2001; Howie et al. 2009; Van Dijk et al. 2018; Amodio et al. 2019] can be difficult

due to the lack of knowledge of the true underlying data distribution. We plan to extend our model's ability to learn from a more diverse set of features, and investigate more efficient ways to handle missing data. We also plan to explore other neural network architectures to improve model performance while reducing model's feature requirement.

5 | Conclusion and Future Directions

5.1 Conclusions

In this thesis, we have discussed methods for integrating data from data-rich problems to improve deep learning solutions for data-limited problems in genomics. Theoretical and technical advances in deep learning (e.g., the invention of more complex neural network architectures for data with complex topology [Goodfellow et al. 2014; Vaswani et al. 2017; Tan and Le 2019; Zhou et al. 2020; Zaheer et al. 2020], and the expansion of cloud computing using GPUs and TPUs [Jouppi et al. 2017]) have allowed more effective representation learning from an increasingly larger amount of genomics data. Computational modeling for genomics problems requires an effective combination of multitask learning and transfer learning to learn from data with multiple modalities. Biological systems are integrative, and omics data measuring different aspects of the same biological phenomenon can provide unique insight to improve the accuracy of machine learning models. Information sharing among multiple related problems also provides the benefit of implicit data augmentation and better model generalization. This is particularly important for a wide range of genomics problems with limited data due to technical and practical difficulties.

Both multitask learning and transfer learning strategies require task-specific parameters, in addition to shared parameters, in order to learn common representation among tasks while maintaining sensitivity to individual tasks. In search of an effective and easily generalizable approach to learning task-specific features, we turn to a family of feature extraction methods, includ-

ing Word2Vec [Mikolov et al. 2013], AEs [Kramer 1991; Hinton and Salakhutdinov 2006], and StarSpace [Wu et al. 2018]. The goal of these methods is uniformly attempting to identify a set of low-dimensional vector representations for entities, although some methods are trained in a self-supervised fashion while others are supervised. Such techniques are readily extendable to new entities and more complex feature extraction problems and can be particularly suited for multi-task learning and transfer learning strategies when the number of tasks and the complexity of the tasks are not fixed. We, therefore, investigate the effect of using entity vector representations as a task-specific model component in multitask learning and transfer learning problems.

In Chapter 3, we use entity vector representations to improve genetic variant calling accuracy using WES data. Genetic variant calling refers to the computational techniques for identifying genetic variants, including SNPs and indels from NGS experiments. NGS can be used to sample the whole genome or the 1-2% of the genome that codes for proteins called the whole exome. Machine learning approaches to variant calling achieve high accuracy in whole genome data, but the significantly fewer training examples cause training with whole exome data alone to achieve lower accuracy. However, building an accurate whole exome variant caller is crucial as WES remains cost-effective for identifying genetic variants. We found that integrating whole genome data improves the exome variant caller performance, either by the multitask learning approach that jointly trains with whole genome and whole exome data or by the transfer learning approach that warmstarts the whole exome model from a trained whole genome model. However, neither of these straightforward data integration strategies includes a task-specific model component that learns sequencing type-specific feature representations. Additional specification of sequencing type when joint training with whole genome and whole exome data further improves exome caller performance, suggesting the ability of models to generalize insights from the greater whole genome data while retaining performance on the specialized whole exome problem. Such techniques may be applied to other problem areas in genomics, where several specialized models would each see only a subset of the genome.

In Chapter 4, we additionally evaluate the improvement of using entity vector representations in a more complex problem that predicts base-pair resolution cell type-specific TF binding sites. Machine learning models for predicting cell type-specific TF binding sites have become increasingly more accurate thanks to the increased availability of NGS data and more standardized model evaluation criteria. However, knowledge transfer from data-rich to data-limited TFs and cell types remains crucial for improving TF binding prediction models because available binding labels are highly skewed towards a small collection of TFs and cell types. Transfer prediction of TF binding sites can potentially benefit from a multitask learning approach; however, existing methods typically use shallow single-task models to generate low-resolution predictions. Here we propose NetTIME, a multitask learning framework for predicting cell type-specific transcription factor binding sites with base-pair resolution. We show that the multitask learning strategy for TF binding prediction is more efficient than the single-task approach due to the increased data availability. NetTIME trains high-dimensional embedding vectors to distinguish TF and cell type identities. We show that this approach is critical for the success of the multitask learning strategy and allows our model to make accurate transfer predictions within and beyond the training panels of TFs and cell types. We also train a CRF to classify binding predictions and show that this CRF eliminates the need to set a probability threshold and reduce classification noise. We compare our method's predictive performance with several state-of-the-art methods, including DeepBind, BindSpace, and Catchitt, and show that our method outperforms previous methods under supervised and transfer learning settings.

Our contributions is threefold. First, we have shown that entity vector representations are effective methods to extract task-specific features in, and subsequently improve the performance of, multitask learning and transfer learning frameworks. Second, our vector representation approach significantly improves DeepVariant exome calling accuracy. This method can be readily applied to other data augmentation problems, such as calling variants from NGS data generated using different DNA sequencing platforms. Last but not least, we developed a multitask learn-

ing framework for predicting base-pair resolution cell type-specific transcription factor binding sites. Our model, called NetTIME, shows a significant improvement in prediction accuracy and prediction resolution compared to existing state-of-the-art methods, and is effective at predicting binding sites for less-studied TFs and cell types.

5.2 Future directions

5.2.1 Improving strategies for learning entity vector representations.

Task-specific vector representations can be incorporated into neural networks in different ways depending on the network architectures. In Chapter 3, we append the sequencing type-specific vector to the output of the InceptionV3 network, whereas in Chapter 4, TF- and cell type-specific vectors are incorporated as the initial hidden state of the recurrent neural network layer.

Depending on the relatedness of the tasks, these vector representations can be incorporated at the early or late stages of the network. However, we believe the concatenation of the vector representations and the output of the network components should only be used when the output of other network components is unstructured (e.g., the output of the InceptionV3 PreLogit layer). For instance, appending vector representations to the end of an input sequence to a recurrent neural network, such as those described in Chapter 4, is likely suboptimal as it would disrupt the structure of the input data. Future work on vector representation incorporation is required for identifying effective incorporation strategies for structured data.

5.2.2 Improving strategies for handling missing modalities.

Biological systems are integrative. Therefore, it is common practice to leverage multimodal data to study biological processes, both experimentally and computationally, as they are often influenced by multitude of factors [Gligorijević and Pržulj 2015]. Integrating genomic as well as

clinical data improve disease diagnosis and phenotyping [Alipanahi et al. 2021]. Integrating large scale genomic datasets in hundreds of cell types can greatly improve the effectiveness of identifying causal variants from GWAS studies [Boix et al. 2021]. However, increasing the data modalities required by machine learning models can greatly decrease the model's scope of application. Take the NetTIME model (Section 4) and cell-type-specific TF binding as an example. Incorporating chromatin accessibility and histone modification sequencing data improve NetTIME TF binding prediction accuracy (Figure 5.1, green). However, as we incorporate more data modalities into the training regimen, the number of cell types for which data modalities are available in ENCODE decreases drastically (Figure 5.1, red).

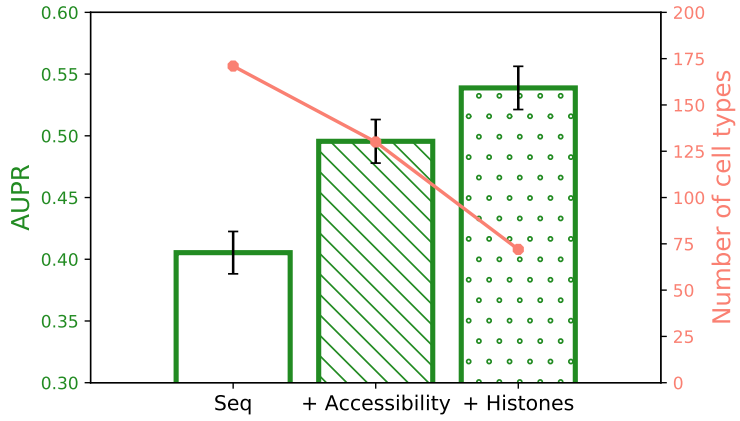


Figure 5.1: Leveraging multple data modalities for improving cell-type-specific TF binding predictions. NetTIME model performance when progressively adding more data modalities is shown in green. The number of cell types for which the ENCODE Consortium hosts sequencing data (i.e., DNase-seq, and H3K4me1, H3K4me3 and H3K27ac histone ChIP-seq experiments) as we increase the number of data modalities considered is shown in red.

Several approaches have been proposed to handle the missing modality problem. A straightforward approach is to randomly drop data modalities when training the model [Jaques et al. 2017a]. Missing modalities can also be imputed using autoencoders [Wang et al. 2018]. However, data imputation may introduce undesired noise when the true underlying data distribution is unknown [Wang et al. 2018], which is common for many genomic problems. The quality of genomic data imputation methods are commonly evaluated based on the similarity between imputed and

real data [Hou et al. 2020]. However, high-throughput sequencing data, especially single-cell sequencing data, only capture snapshots of the dynamic cell states, which partially explains the significance of batch effect when handling biological data [Leek et al. 2010]. Being able to simulate data that resemble any particular real dataset is therefore not an accurate measurement of computational models' ability to more comprehensively capture dynamic cell states.

Missing modality methods that focus on incorporating both complete and incomplete data modalities is therefore particularly suited for handling genomics data as large scale data consortiums have collected large amount of high-throughput sequencing data; but most of them are highly concentrated towards a small collection of well-studied factors and conditions (Figure 1.1 and Boix et al. [2021]). Semi-supervised learning can be deployed to allow different objectives for datapoints having complete and incomplete data modalities [Yang et al. 2018]. Using the notion of knowledge distillation, one can first train separate models for each data modality to generate a set of soft labels that impute labels for missing modalities. The soft labels, as well as the true target labels from multiple data modalities can then be used to train multimodal model for a specific objectives of interest [Wang et al. 2020]. Such methods, although not designed for genomic problems, can provide useful insights from which specialized missing modality methods for genomics can be developed.

5.2.3 Biophysically motivated modeling of biological systems.

It has always been my goal to build machine learning models for biological systems motivated by their biophysical properties. For instance, in Chapter 4, we trained a CRF to convert base-pair resolution binding probabilities to binary binding decisions. This work is inspired by the observation that positions immediately adjacent to a binding site are also likely bound by the same TF. CRF, therefore, is advantageous compared to logistic-regression-based classifiers because the class label at the previous position affects the classification at the current position (Figure 4.10).

Constructing TRNs (see Section 2.4.3) using TF motifs can be problematic, as the source of the

motif library can have a significant effect on predicted network structure [Gibbs et al. 2021]. More importantly, motifs that specify TF sequence specificity only explain a fraction of the *in vivo* TF binding landscape. TF motifs indicate strong *in vitro* binding sites. However, *in vivo* TF binding sites are highly influenced by chromatin accessibility and show a low correlation to TF motif enrichment [Chen et al. 2017]. Many functionally important low-affinity and transient binding sites do not show motif enrichment. These observations have important functional implications. In fact, TF-regulated tissue- and cell type-specific gene expressions are often indicative of tissue- and cell type-specific functions. Methods for predicting base-pair resolution *in vivo* TF binding sites, such as NetTIME proposed in Chapter 4, can serve as a more flexible approach to generating prior network structure as it bypasses the aforementioned unnecessary TF motif constraints. In future work, we hope to adapt the NetTIME framework to explore more efficient approaches for generating prior knowledge for more biophysically motivated TRN inference.

A | Appendix

A.1 Supplementary information for Chapter 4

A.1.1 Supplementary method

A.1.1.1 Data retrieval and preprocessing

The list of 71 TF-focused ChIP-seq experiments we use to generate target labels are provided in Supplementary Data 1. Combined peak set are first generated by merging the conserved and relaxed peak sets from the above experiments. Peaks that are longer than 1000 bp are removed. Two overlapping peaks are merged when 1) they overlap for more than 200 bp, and 2) the resulting merged peak is shorter than 600 bp. Each interval in the merged peak set is used to create one 1000 bp example sequence where the midpoints of the example and the interval are the same. For TF p in cell type q, a nucleotide n is classified as bound if n is within any ChIP-seq peaks for condition $[p, q]$, and unbound otherwise.

The ENCODE experiments used to generate cell type-specific features are provided in Supplementary Data 2. We download the narrowPeak BED files for all DNase-seq and histone ChIP-seq experiments, read-depth normalized signal bigWig files for all DNase-seq experiments, and the signal p-value bigWig files for all histone ChIP-seq experiment from the ENCODE Consortium [1]. The genomic interval of each example sequence is intersected with DNase-seq and Histone ChIP-

[1] https://www.encodeproject.org/

seq bigWig files to retrieve the corresponding example cell-type-specific feature signals. To reduce noise, only positive signals that fall within peak regions, defined in the DNase-seq and histone ChIP-seq narrowPeak BED files, are retrieved to generate the example feature signal tracks. Each cell-type-specific feature signal track across all examples are further zscore normalized before being used as input to the NetTIME model.

To test our model's ability to make transfer predictions beyond the training panels of TFs and cell types, we additionally collect target labels from 40 ENCODE TF-focused ChIP-seq experiments (Supplementary Data 3) and cell type-specific features from 10 cell types (Supplementary Data 4). These additional datasets are processed using the same procedures described above to generate additional fine-tuning examples.

To compare NetTIME performance with that of Catchitt, and to evaluate whether precomputed TF motif PWMs can improve *in vivo* TF binding predictions, we additionally obtain the v11 TF motif PWMs [2] from the HOCOMOCO [Kulakovskiy et al. 2018] motif database. If multiple motif PWMs exist for a particular TF in the database, we select the motif PWM with the highest quality rating. To generate TF motif features for the NetTIME model, we run FIMO [Grant et al. 2011] for each example sequence using the collected TF motif PWMs to find regions in the example sequence that exhibit TF motif enrichment. FIMO motif search is conducted by setting the p-value threshold to 1e-2 while leaving all other parameters as default. The motif enrichment scores for each DNA strand are mapped to the corresponding regions in the example sequence. Regions in the example sequence that are not enriched by the TF of interest is set to zero. As a result, two additional TF-specific features–TF motif enrichment for plus and minus DNA strands–can be added along the feature dimension of the NetTIME input. For each strand, raw motif enrichment scores are zscore normalized across all examples before being used as NetTIME input.

[2] https://hocomoco11.autosome.ru/downloads_v11

A.1.1.2 Method comparison

We divide each 1000 bp example sequence in our test set into n-bp bins ($1 \leq n \leq 1000$). To test model performance under different resolutions, we set the bin width $n = 20, 50, 100, 200, 500, 1000$. These binned sequences, along with other necessary input features, are provided to DeepBind, BindSpace and Catchitt to generate predictions under different resolutions. We describe below in detail how predictions are generated from these three methods.

DeepBind We directly use pretrained DeepBind ChIP-seq models[3] to generate predictions on our test data. Among all DeepBind models pretrained on ChIP-seq data, we found 11 conditions that overlap with our training set of conditions. DeepBind performance is therefore evaluated against our test data using the 11 pretrained models.

BindSpace BindSpace provides pretrained models for 243 TFs, 6 of which overlap with our training set of TFs. We first generate test data predictions for these 6 TFs. As BindSpace's predictions are not cell-type-specific, model performance is assessed by comparing predictions made for each TF with the target labels of the same TF across all available cell types. Among our training set of conditions, target labels are available for 22 conditions containing the above 6 TFs. BindSpace performance is therefore evaluated using test data generated for these 22 conditions.

Catchitt We separately train 71 Cachitt models for all conditions we included in our training set, as pretrained Catchitt models are not publicly available. Both DNase-seq data and the HOCOMOCO TF motifs are used as features and these models are trained by following the Catchitt documentation[4]. Training, validation, and test data are split the same way as our model (Table A.1). Catchitt performance achieved on test dataset are reported here.

For all of the above methods, we use predictions generated using the lowest applicable bin

[3] http://tools.genes.toronto.edu/deepbind/
[4] http://www.jstacs.de/index.php/Catchitt

width n ($n = 20$ for DeepBind and BindSpace, and $n = 50$ for Catchitt) to compare models' base-pair level performance. For any method F, if $F(\mathbf{x}) = c$ for input sequence x of n-bp, then we set $F_{bp}(x_i) = c, \forall i \in [1, n]$, where F_{bp} is the base-pair level prediction derived from original method prediction.

A.1.2 Supplementary Tables

Dataset	Chromosomes	Peak set(s)	Number of conditions	Number of samples	
				Per condition	Total
Training	3-7, 10-20	Conserved + Relaxed	71	937, 676	66, 574, 996
Validation	2, 9, 22	Conserved	71	101, 644	7, 216, 724
Test	1, 8, 21	Conserved	71	102, 178	7, 254, 638

Table A.1: The number of samples in training, validation and test datasets used to train and evaluate NetTIME supervised performance. Data splits are performed according to chromosomes. Both the ENCODE TF ChIP-seq conserved and relaxed peak sets are used for training, whereas only the conserved peak set is used to construct the validation and test datasets. A condition specifies a single TF-focused ChIP-seq experiment conducted in a particular cell type. All training, validation and test data are generated from the same set of 71 ChIP-seq conditions spanning 22 TFs and 7 cell types.

A.1.3 Supplementary figures

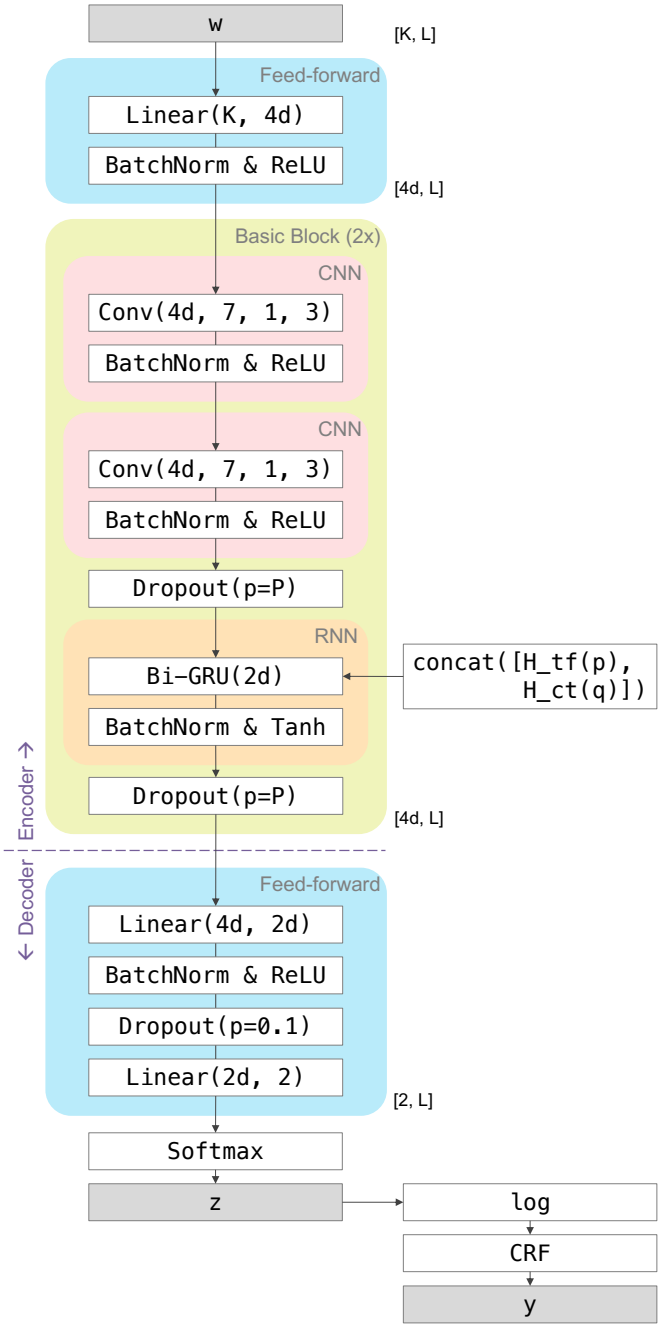


Figure A.1: NetTIME architecture. Detailed view of the NetTIME architecture shown in Figure 4.2b. Three main parameters of the model are the number of input features K, example sequence length L, and embedding dimension d.
(Continue on next page.)

Figure A.1: (Continued from the previous page.)
Output dimensions for different model layers are shown on the right side of the architecture blocks. Linear(a, b) denotes the linear transformation of input size a and output size b; Conv(e, f, g, h) denotes the 1D convolution operation of e output channels, kernel size f, stride g and padding h. Bi-GRU(i) denotes bi-directional GRU layer of hidden size i. Dropout probability $P = 0.1$ for the first pass of the Basic Block, and $P = 0.0$ for the second pass.

A.1.4 Supplementary data

Supplementary Data 1 A list of ENCODE replicated TF ChIP-seq experiments used to generate target labels and train the NetTIME model.

Supplementary Data 2 A list of ENCODE DNase-seq, and H3K4me1, H3K4me3 and H3K27ac ChIP-seq experiments used to generate cell type-specific features to train NetTIME model.

Supplementary Data 3 An additional list of ENCODE replicated TF ChIP-seq experiments used to fine tune pretrained NetTIME model for transfer learning.

Supplementary Data 4 An additional list of ENCODE DNase-seq, and H3K4me1, H3K4me3 and H3K27ac ChIP-seq experiments used to fine tune pretrained NetTIME model for transfer learning.

Bibliography

Ruslan Abasov, Varvara E Tvorogova, Andrey S Glotov, Alexander V Predeus, and Yury A Barbitoff. Systematic benchmark of state-of-the-art variant calling pipelines identifies major factors affecting accuracy of coding sequence variant discovery. *bioRxiv*, 2021.

B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, 33(8):831–838, Aug 2015.

Babak Alipanahi, Farhad Hormozdiari, Babak Behsaz, Justin Cosentino, Zachary R McCaw, Emanuel Schorsch, D Sculley, Elizabeth H Dorfman, Paul J Foster, Lily H Peng, et al. Large-scale machine learning-based phenotyping significantly improves genomic discovery for optic nerve head morphology. *The American Journal of Human Genetics*, 2021.

Shanika L Amarasinghe, Shian Su, Xueyi Dong, Luke Zappia, Matthew E Ritchie, and Quentin Gouil. Opportunities and challenges in long-read sequencing data analysis. *Genome biology*, 21(1):1–16, 2020.

Joanna S Amberger, Carol A Bocchini, Alan F Scott, and Ada Hamosh. Omim. org: leveraging knowledge across phenotype–gene relationships. *Nucleic acids research*, 47(D1):D1038–D1043, 2019.

Matthew Amodio, David Van Dijk, Krishnan Srinivasan, William S Chen, Hussein Mohsen, Kevin R Moon, Allison Campbell, Yujiao Zhao, Xiaomei Wang, Manjunatha Venkataswamy,

et al. Exploring single-cell data with deep multitasking neural networks. *Nature methods*, 16 (11):1139–1145, 2019.

Florent E Angly, Dana Willner, Forest Rohwer, Philip Hugenholtz, and Gene W Tyson. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research*, 40(12):e94–e94, 2012.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017a.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017b.

Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015a.

Ehsaneddin Asgari and Mohammad RK Mofrad. Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, 10(11):e0141287, 2015b.

Ziga 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. *bioRxiv*, 2021a.

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*, pages 1–13, 2021b.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

T. L. Bailey, N. Williams, C. Misleh, and W. W. Li. MEME: discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Res.*, 34(Web Server issue):W369–373, Jul 2006.

Timothy L. Bailey. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36. AAAI Press, 1994.

Timothy L Bailey. Streme: Accurate and versatile sequence motif discovery. *bioRxiv*, 2020.

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

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.

Michael J Bamshad, Sarah B Ng, Abigail W Bigham, Holly K Tabor, Mary J Emond, Deborah A Nickerson, and Jay Shendure. Exome sequencing as a tool for mendelian disease gene discovery. *Nature Reviews Genetics*, 12(11):745, 2011.

Andrew J Bannister and Tony Kouzarides. Regulation of chromatin by histone modifications. *Cell research*, 21(3):381–395, 2011.

Yury A Barbitoff, Dmitrii E Polev, Andrey S Glotov, Elena A Serebryakova, Irina V Shcherbakova, Artem M Kiselev, Anna A Kostareva, Oleg S Glotov, and Alexander V Predeus. Systematic dissection of biases in whole-exome and whole-genome sequencing reveals major determinants of coding sequence coverage. *Scientific reports*, 10(1):1–13, 2020.

Bérénice A Benayoun, Elizabeth A Pollina, Duygu Ucar, Salah Mahmoudi, Kalpana Karra, Edith D Wong, Keerthana Devarajan, Aaron C Daugherty, Anshul B Kundaje, Elena Mancini, et al. H3k4me3 breadth is linked to cell identity and transcriptional consistency. *Cell*, 158(3):673–688, 2014.

Michael F Berger, Anthony A Philippakis, Aaron M Qureshi, Fangxue S He, Preston W Estep, and Martha L Bulyk. Compact, universal dna microarrays to comprehensively determine transcription-factor binding site specificities. *Nature biotechnology*, 24(11):1429–1435, 2006.

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.

Carles A Boix, Benjamin T James, Yongjin P Park, Wouter Meuleman, and Manolis Kellis. Regulatory genomic circuitry of human disease loci by integrative epigenomics. *Nature*, 590(7845): 300–307, 2021.

Alan P Boyle, Sean Davis, Hennady P Shulha, Paul Meltzer, Elliott H Margulies, Zhiping Weng, Terrence S Furey, and Gregory E Crawford. High-resolution mapping and characterization of open chromatin across the genome. *Cell*, 132(2):311–322, 2008.

Michael Breed and Leticia Sanchez. Both environment and genetic makeup influence behavior. *Nature Education Knowledge*, 3(10):68, 2010.

Arie B Brinkman, Thijs Roelofsen, Sebastiaan WC Pennings, Joost HA Martens, Thomas Jenuwein, and Hendrik G Stunnenberg. Histone modification patterns associated with the human x chromosome. *EMBO reports*, 7(6):628–634, 2006.

Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, 10(12):1213, 2013.

Carol J Bult, Judith A Blake, Cynthia L Smith, James A Kadin, and Joel E Richardson. Mouse genome database (mgd) 2019. *Nucleic acids research*, 47(D1):D801–D806, 2019.

Annalisa Buniello, Jacqueline A L MacArthur, Maria Cerezo, Laura W Harris, James Hayhurst, Cinzia Malangone, Aoife McMahon, Joannella Morales, Edward Mountjoy, Elliot Sollis, et al. The nhgri-ebi gwas catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic acids research*, 47(D1):D1005–D1012, 2019.

Diego Calderon, Michelle LT Nguyen, Anja Mezger, Arwa Kathiria, Fabian Müller, Vinh Nguyen, Ninnia Lescano, Beijing Wu, John Trombetta, Jessica V Ribado, et al. Landscape of stimulation-responsive chromatin across diverse human immune cells. *Nature genetics*, pages 1–12, 2019.

Aslıhan Karabacak Calviello, Antje Hirsekorn, Ricardo Wurmus, Dilmurat Yusuf, and Uwe Ohler. Reproducible inference of transcription factor footprints in atac-seq and dnase-seq datasets using protocol-specific bias modeling. *Genome biology*, 20(1):1–13, 2019.

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

Dayanne M Castro, Nicholas R De Veaux, Emily R Miraldi, and Richard Bonneau. Multi-study inference of regulatory networks for more accurate models of gene regulation. *PLoS computational biology*, 15(1):e1006591, 2019.

Huidong Chen, Caleb Lareau, Tommaso Andreani, Michael E Vinyard, Sara P Garcia, Kendell Clement, Miguel A Andrade-Navarro, Jason D Buenrostro, and Luca Pinello. Assessment of computational methods for the analysis of single-cell atac-seq data. *Genome biology*, 20(1): 1–25, 2019.

Xi Chen, Bowen Yu, Nicholas Carriero, Claudio Silva, and Richard Bonneau. Mocap: large-scale inference of transcription factor binding sites from chromatin accessibility. *Nucleic acids research*, 45(8):4315–4329, 2017.

Jeanne Chèneby, Marius Gheorghe, Marie Artufel, Anthony Mathelier, and Benoit Ballester. Remap 2018: an updated atlas of regulatory regions from an integrative analysis of dna-binding chip-seq experiments. *Nucleic acids research*, 46(D1):D267–D275, 2017.

Jeanne Chèneby, Zacharie Ménétrier, Martin Mestdagh, Thomas Rosnet, Allyssa Douida, Wassim Rhalloussi, Aurélie Bergon, Fabrice Lopez, and Benoit Ballester. Remap 2020: a database of regulatory regions from an integrative analysis of human and arabidopsis dna-binding sequencing experiments. *Nucleic acids research*, 48(D1):D180–D188, 2020.

J Michael Cherry, Eurie L Hong, Craig Amundsen, Rama Balakrishnan, Gail Binkley, Esther T Chan, Karen R Christie, Maria C Costanzo, Selina S Dwight, Stacia R Engel, et al. Saccharomyces genome database: the genomics resource of budding yeast. *Nucleic acids research*, 40 (D1):D700–D705, 2012.

Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014a.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014b.

Kyunghyun Cho et al. Foundations and advances in deep learning. 2014c.

1000 Genomes Project Consortium et al. A global reference for human genetic variation. *Nature*, 526(7571):68, 2015.

ENCODE Project Consortium et al. The encode (encyclopedia of dna elements) project. *Science*, 306(5696):636–640, 2004.

International Human Genome Sequencing Consortium et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

Menno P Creyghton, Albert W Cheng, G Grant Welstead, Tristan Kooistra, Bryce W Carey, Eveline J Steine, Jacob Hanna, Michael A Lodato, Garrett M Frampton, Phillip A Sharp, et al. Histone h3k27ac separates active from poised enhancers and predicts developmental state. *Proceedings of the National Academy of Sciences*, 107(50):21931–21936, 2010.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Kathleen Danna and Daniel Nathans. Specific cleavage of simian virus 40 dna by restriction endonuclease of hemophilus influenzae. *Proceedings of the National Academy of Sciences*, 68 (12):2913–2917, 1971.

Vishaka Datta, Sridhar Hannenhalli, and Rahul Siddharthan. Chipulate: A comprehensive chip-seq simulation pipeline. *PLoS computational biology*, 15(3):e1006921, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo Del Angel, Manuel A Rivas, Matt Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491, 2011a.

Mark A DePristo, Eric Banks, Ryan Poplin, Kiran V Garimella, Jared R Maguire, Christopher Hartl, Anthony A Philippakis, Guillermo Del Angel, Manuel A Rivas, Matt Hanna, et al. A framework for variation discovery and genotyping using next-generation dna sequencing data. *Nature genetics*, 43(5):491, 2011b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018a.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018b.

DREAM. Encode-dream in vivo transcription factor binding site prediction challenge, 2017.

Dent Earl, Keith Bradnam, John St John, Aaron Darling, Dawei Lin, Joseph Fass, Hung On Ken Yu, Vince Buffalo, Daniel R Zerbino, Mark Diekhans, et al. Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome research*, 21(12):2224–2241, 2011.

Albert O Edwards, Robert Ritter, Kenneth J Abel, Alisa Manning, Carolien Panhuysen, and Lindsay A Farrer. Complement factor h polymorphism and age-related macular degeneration. *Science*, 308(5720):421–424, 2005.

Hilary M Ellis and H Robert Horvitz. Genetic control of programmed cell death in the nematode c. elegans. *Cell*, 44(6):817–829, 1986.

ENCODE. Encode experiment guidelines, 2020. URL https://www.encodeproject.org/about/experiment-guidelines.

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.

Merly Escalona, Sara Rocha, and David Posada. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature Reviews Genetics*, 17(8):459, 2016.

Benjamin Jung Fair, Lauren E Blake, Abhishek Sarkar, Bryan J Pavlovic, Claudia Cuevas, and Yoav Gilad. Gene expression variability in human and chimpanzee populations share common determinants. *Elife*, 9:e59929, 2020.

Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS biol*, 5(1):e8, 2007.

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.

Alyssa C Frazee, Andrew E Jaffe, Ben Langmead, and Jeffrey T Leek. Polyester: simulating rna-seq datasets with differential transcript expression. *Bioinformatics*, 31(17):2778–2784, 2015.

Kelly A Frazer, Sarah S Murray, Nicholas J Schork, and Eric J Topol. Human genetic variation and its contribution to complex traits. *Nature Reviews Genetics*, 10(4):241–251, 2009.

Seth Frietze and Peggy J Farnham. Transcription factor effector domains. In *A handbook of transcription factors*, pages 261–277. Springer, 2011.

Shuhua Fu, Anqi Wang, and Kin Fai Au. A comparative evaluation of hybrid error correction methods for error-prone long reads. *Genome biology*, 20(1):1–17, 2019.

Erik Garrison and Gabor Marth. Haplotype-based variant detection from short-read sequencing. *arXiv preprint arXiv:1207.3907*, 2012.

Dongliang Ge, Jacques Fellay, Alexander J Thompson, Jason S Simon, Kevin V Shianna, Thomas J Urban, Erin L Heinzen, Ping Qiu, Arthur H Bertelsen, Andrew J Muir, et al. Genetic variation in il28b predicts hepatitis c treatment-induced viral clearance. *Nature*, 461(7262):399–401, 2009.

Pablo V Gejman, Alan R Sanders, and Jubao Duan. The role of genetics in the etiology of schizophrenia. *Psychiatric Clinics*, 33(1):35–66, 2010.

David Gerard. Data-based rna-seq simulations by binomial thinning. *BMC bioinformatics*, 21: 1–14, 2020.

David Gfeller, Frank Butty, Marta Wierzbicka, Erik Verschueren, Peter Vanhee, Haiming Huang, Andreas Ernst, Nisa Dar, Igor Stagljar, Luis Serrano, et al. The multiple-specificity landscape of modular peptide recognition domains. *Molecular systems biology*, 7(1):484, 2011.

Soumita Ghosh, Abhik Datta, and Hyungwon Choi. multislide is a web server for exploring connected elements of biological pathways in multi-omics data. *Nature Communications*, 12 (1):1–11, 2021.

Claudia Skok Gibbs, Christopher A Jackson, Giuseppe-Antonio Saldi, Aashna Shah, Andreas Tjärnberg, Aaron Watters, Nicholas De Veaux, Konstantine Tchourine, Ren Yi, Tymor Hamamsy, et al. Single-cell gene regulatory network inference at scale: The inferelator 3.0. *bioRxiv*, 2021.

Paul G Giresi, Jonghwan Kim, Ryan M McDaniell, Vishwanath R Iyer, and Jason D Lieb. Faire (formaldehyde-assisted isolation of regulatory elements) isolates active regulatory elements from human chromatin. *Genome research*, 17(6):877–885, 2007.

Vladimir Gligorijević and Nataša Pržulj. Methods for biological data integration: perspectives and challenges. *Journal of the Royal Society Interface*, 12(112):20150571, 2015.

Vladimir Gligorijević, Meet Barot, and Richard Bonneau. deepnf: deep network fusion for protein function prediction. *Bioinformatics*, 34(22):3873–3881, 2018.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

David Gomez-Cabrero, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merkenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, and Jesper Tegnér. Data integration in the era of omics: current and future challenges. *BMC systems biology*, 8(2):1–10, 2014.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

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

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

AJF Griffiths, WM Gelbart, JH Miller, and RC Lewontin. Chromosomal rearrangements. *Modern genetic analysis*, 1999.

Elin Grundberg, Kerrin S Small, Åsa K Hedman, Alexandra C Nica, Alfonso Buil, Sarah Keildson, Jordana T Bell, Tsun-Po Yang, Eshwar Meduri, Amy Barrett, et al. Mapping cis-and trans-regulatory effects across multiple tissues in twins. *Nature genetics*, 44(10):1084–1089, 2012.

Shobhit Gupta, John A Stamatoyannopoulos, Timothy L Bailey, and William Stafford Noble. Quantifying similarity between motifs. *Genome biology*, 8(2):R24, 2007.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R Curran. Evaluating entity linking with wikipedia. *Artificial intelligence*, 194:130–150, 2013.

William L Hamilton, Payal Bajaj, Marinka Zitnik, Dan Jurafsky, and Jure Leskovec. Embedding logical queries on knowledge graphs. *arXiv preprint arXiv:1806.01445*, 2018.

Heonjong Han, Hongseok Shim, Donghyun Shin, Jung Eun Shim, Yunhee Ko, Junha Shin, Hanhae Kim, Ara Cho, Eiru Kim, Tak Lee, et al. Trrust: a reference database of human transcriptional regulatory interactions. *Scientific reports*, 5(1):1–11, 2015.

Heonjong Han, Jae-Won Cho, Sangyoung Lee, Ayoung Yun, Hyojin Kim, Dasom Bae, Sunmo Yang, Chan Yeong Kim, Muyoung Lee, Eunbeen Kim, et al. Trrust v2: an expanded reference database of human and mouse transcriptional regulatory interactions. *Nucleic acids research*, 46(D1):D380–D386, 2018.

Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, et al. Gencode: the reference human genome annotation for the encode project. *Genome research*, 22(9): 1760–1774, 2012.

Yehudit Hasin-Brumshtein, Farhad Hormozdiari, Lisa Martin, Atila Van Nas, Eleazar Eskin, Aldons J Lusis, and Thomas A Drake. Allele-specific expression and eqtl analysis in mouse adipose tissue. *BMC genomics*, 15(1):1–13, 2014.

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.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Qiye He, Jeff Johnston, and Julia Zeitlinger. Chip-nexus enables improved detection of in vivo transcription factor binding footprints. *Nature biotechnology*, 33(4):395–401, 2015.

Michael O Hengartner and H Robert Horvitz. C. elegans cell survival gene ced-9 encodes a functional homolog of the mammalian proto-oncogene bcl-2. *Cell*, 76(4):665–676, 1994.

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.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

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

Manuel Holtgrewe. Mason: a read simulator for second generation sequencing data. 2010.

Wenpin Hou, Zhicheng Ji, Hongkai Ji, and Stephanie C Hicks. A systematic evaluation of single-cell rna-sequencing imputation methods. *Genome biology*, 21(1):1–30, 2020.

Bryan N Howie, Peter Donnelly, and Jonathan Marchini. A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS Genet*, 5(6): e1000529, 2009.

Hsin-Lung Hsieh, Jen-Tzung Chien, Koichi Shinoda, and Sadaoki Furui. Independent component analysis for noisy speech recognition. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4369–4372. IEEE, 2009.

Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 16–23. IEEE, 2017.

Jessica Xin Hu, Cecilia Engel Thomas, and Søren Brunak. Network biology concepts in complex disease comorbidities. *Nature Reviews Genetics*, 17(10):615, 2016.

Xuesong Hu, Jianying Yuan, Yujian Shi, Jianliang Lu, Binghang Liu, Zhenyu Li, Yanxiang Chen, Desheng Mu, Hao Zhang, Nan Li, et al. pirs: Profile-based illumina pair-end reads simulator. *Bioinformatics*, 28(11):1533–1535, 2012.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

Mingtao Huang, Jichen Bao, Björn M Hallström, Dina Petranovic, and Jens Nielsen. Efficient protein production by yeast requires global tuning of metabolism. *Nature communications*, 8 (1):1–12, 2017.

Maxwell A Hume, Luis A Barrera, Stephen S Gisselbrecht, and Martha L Bulyk. Uniprobe, update 2015: new tools and content for the online database of protein-binding microarray data on protein–dna interactions. *Nucleic acids research*, 43(D1):D117–D122, 2015.

Kwangbeom Hyun, Jongcheol Jeon, Kihyun Park, and Jaehoon Kim. Writing, erasing and reading histone lysine methylations. *Experimental & molecular medicine*, 49(4):e324–e324, 2017.

Aapo Hyvärinen. Independent component analysis: recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110534, 2013.

Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 202–208. IEEE, 2017a.

Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multimodal autoencoder: A deep learning approach to filling in missing sensor data and enabling better mood prediction. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 202–208. IEEE, 2017b.

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, 2007a.

W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007b.

Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

Arttu Jolma, Yimeng Yin, Kazuhiro R Nitta, Kashyap Dave, Alexander Popov, Minna Taipale, Martin Enge, Teemu Kivioja, Ekaterina Morgunova, and Jussi Taipale. Dna-dependent formation of transcription factor pairs alters their binding specificity. *Nature*, 527(7578):384–388, 2015.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*, pages 1–12. IEEE, 2017.

Andrew R Joyce and Bernhard Ø Palsson. The model organism as a system: integrating'omics' data sets. *Nature reviews Molecular cell biology*, 7(3):198–210, 2006.

Kenji Kamimoto, Christy M Hoffmann, and Samantha A Morris. Celloracle: Dissecting cell identity via network inference and in silico gene perturbation. *bioRxiv*, 2020.

Krishanpal Karmodiya, Arnaud R Krebs, Mustapha Oulad-Abdelghani, Hiroshi Kimura, and Laszlo Tora. H3k9 and h3k14 acetylation co-occur at many gene regulatory elements, while

h3k14ac marks a subset of inactive inducible promoters in mouse embryonic stem cells. *BMC genomics*, 13(1):1–18, 2012.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

Jens Keilwagen, Stefan Posch, and Jan Grau. Accurate prediction of cell type-specific transcription factor binding. *Genome biology*, 20(1):9, 2019.

David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018a.

David R Kelley, Yakir A Reshef, Maxwell Bileschi, David Belanger, Cory Y McLean, and Jasper Snoek. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome research*, 28(5):739–750, 2018b.

Joost JB Keurentjes, Jingyuan Fu, Inez R Terpstra, Juan M Garcia, Guido van den Ackerveken, L Basten Snoek, Anton JM Peeters, Dick Vreugdenhil, Maarten Koornneef, and Ritsert C Jansen. Regulatory network construction in arabidopsis by using genome-wide gene expression quantitative trait loci. *Proceedings of the National Academy of Sciences*, 104(5):1708–1713, 2007.

Sangtae Kim, Konrad Scheffler, Aaron L Halpern, Mitchell A Bekritsky, Eunho Noh, Morten Källberg, Xiaoyu Chen, Yeonbin Kim, Doruk Beyter, Peter Krusche, et al. Strelka2: fast and accurate calling of germline and somatic variants. *Nature methods*, 15(8):591, 2018.

Motoo Kimura et al. Evolutionary rate at the molecular level. *Nature*, 217(5129):624–626, 1968.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

David C Klein and Sarah J Hainer. Genomic methods in profiling dna accessibility and factor localization. *Chromosome Research*, 28(1):69–85, 2020.

Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. *arXiv preprint arXiv:1805.06201*, 2018.

Alexander Kohlmaier, Fabio Savarese, Monika Lachner, Joost Martens, Thomas Jenuwein, and Anton Wutz. A chromosomal memory triggered by xist regulates histone methylation in x inactivation. *PLoS Biol*, 2(7):e171, 2004.

Alexey Kolesnikov, Sidharth Goel, Maria Nattestad, Taedong Yun, Gunjan Baid, Howard Yang, Cory McLean, Pi-Chuan Chang, and Andrew Carroll. Deeptrio: Variant calling in families using deep learning. *bioRxiv*, 2021.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019.

Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Peter Krusche, Len Trigg, Paul C Boutros, Christopher E Mason, M Francisco, Benjamin L Moore, Mar Gonzalez-Porta, Michael A Eberle, Zivana Tezak, Samir Lababidi, et al. Best practices for benchmarking germline small-variant calls in human genomes. *Nature biotechnology*, 37(5): 555–560, 2019.

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.

Samuel A Lambert, Arttu Jolma, Laura F Campitelli, Pratyush K Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R Hughes, and Matthew T Weirauch. The human transcription factors. *Cell*, 172(4):650–665, 2018.

Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357, 2012.

Aoife Larkin, Steven J Marygold, Giulia Antonazzo, Helen Attrill, Gilberto Dos Santos, Phani V Garapati, Joshua L Goodman, L Sian Gramates, Gillian Millburn, Victor B Strelets, et al. Flybase: updates to the drosophila melanogaster knowledge base. *Nucleic Acids Research*, 49(D1):D899–D907, 2021.

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

Jeffrey T Leek, Robert B Scharpf, Héctor Corrada Bravo, David Simcha, Benjamin Langmead, W Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A Irizarry. Tackling the

widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.

Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.

Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

Yifeng Li, Fang-Xiang Wu, and Alioune Ngom. A review on machine learning principles for multi-view biological data integration. *Briefings in bioinformatics*, 19(2):325–340, 2018.

Gangning Liang, Joy CY Lin, Vivian Wei, Christine Yoo, Jonathan C Cheng, Carvell T Nguyen, Daniel J Weisenberger, Gerda Egger, Daiya Takai, Felicidad A Gonzales, et al. Distinct localization of histone h3 acetylation and h3-k4 methylation to the transcription start sites in the human genome. *Proceedings of the National Academy of Sciences*, 101(19):7357–7362, 2004.

James C Liao, Riccardo Boscolo, Young-Lyeol Yang, Linh My Tran, Chiara Sabatti, and Vwani P Roychowdhury. Network component analysis: reconstruction of regulatory signals in biological systems. *Proceedings of the National Academy of Sciences*, 100(26):15522–15527, 2003.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*, 2018.

John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580–585, 2013.

Karolin Luger, Armin W Mäder, Robin K Richmond, David F Sargent, and Timothy J Richmond. Crystal structure of the nucleosome core particle at 2.8 å resolution. *Nature*, 389(6648):251–260, 1997.

Ruibang Luo, Fritz J Sedlazeck, Tak-Wah Lam, and Michael C Schatz. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nature communications*, 10(1):998, 2019.

Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. Smil: Multimodal learning with severely missing modality. *arXiv preprint arXiv:2103.05677*, 2021.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, and Gustavo Stolovitzky. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

Mark I McCarthy, Gonçalo R Abecasis, Lon R Cardon, David B Goldstein, Julian Little, John PA Ioannidis, and Joel N Hirschhorn. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature reviews genetics*, 9(5):356–369, 2008.

Kerensa E McElroy, Fabio Luciani, and Torsten Thomas. Gemsim: general, error-model based simulator of next-generation sequencing data. *BMC genomics*, 13(1):1–9, 2012.

Martin J McKeown, Lars Kai Hansen, and Terrence J Sejnowsk. Independent component analysis of functional mri: what is signal and what is noise? *Current opinion in neurobiology*, 13(5): 620–629, 2003.

Fatima Mechta-Grigoriou, Damien Gerald, and Moshe Yaniv. The mammalian jun proteins: redundancy and specificity. *Oncogene*, 20(19):2378–2389, 2001.

Janine Meienberg, Rémy Bruggmann, Konrad Oexle, and Gabor Matyas. Clinical sequencing: is wgs the better wes? *Human genetics*, 135(3):359–362, 2016.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Emily R Miraldi, Maria Pokrovskii, Aaron Watters, Dayanne M Castro, Nicholas De Veaux, Jason A Hall, June-Yong Lee, Maria Ciofani, Aviv Madar, Nick Carriero, et al. Leveraging chromatin accessibility for transcriptional regulatory network inference in t helper 17 cells. *Genome research*, 29(3):449–463, 2019.

Trine H Mogensen. Irf and stat transcription factors-from basic biology to roles in infection, protective immunity, and primary immunodeficiencies. *Frontiers in immunology*, 9:3047, 2019.

Jill E Moore, Michael J Purcaro, Henry E Pratt, Charles B Epstein, Noam Shoresh, Jessika Adrian, Trupti Kawli, Carrie A Davis, Alexander Dobin, Rajinder Kaul, et al. Expanded encyclopaedias of dna elements in the human and mouse genomes. *Nature*, 583(7818):699–710, 2020.

Mohammad Amin Morid, Alireza Borjali, and Guilherme Del Fiol. A scoping review of transfer learning research on medical image analysis using imagenet. *Computers in Biology and Medicine*, page 104115, 2020.

AJ Muir, L Gong, SG Johnson, MT Michael Lee, MS Williams, TE Klein, KE Caudle, and DR Nelson. Clinical pharmacogenetics implementation consortium (cpic) guidelines for ifnl3 (il28b) genotype and peg interferon-α–based regimens. *Clinical Pharmacology & Therapeutics*, 95(2): 141–146, 2014.

Cancer Genome Atlas Network et al. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61, 2012.

Daniel E Newburger and Martha L Bulyk. Uniprobe: an online database of protein binding microarray data on protein–dna interactions. *Nucleic acids research*, 37(suppl_1):D77–D82, 2009.

Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

Clare Pacini, Joshua M Dempster, Isabella Boyle, Emanuel Gonçalves, Hanna Najgebauer, Emre Karakoc, Dieudonne van der Meer, Andrew Barthorpe, Howard Lightfoot, Patricia Jaaks, et al. Integrated cross-study datasets of genetic dependencies in cancer. *Nature communications*, 12 (1):1–14, 2021.

Peter J Park. Chip–seq: advantages and challenges of a maturing technology. *Nature reviews genetics*, 10(10):669–680, 2009.

Isabelle S Peter and Eric H Davidson. Evolution of gene regulatory networks controlling body plan development. *Cell*, 144(6):970–985, 2011.

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, 2018a.

Ryan Poplin, Valentin Ruano-Rubio, Mark A DePristo, Tim J Fennell, Mauricio O Carneiro, Geraldine A Van der Auwera, David E Kling, Laura D Gauthier, Ami Levy-Moonshine, David Roazen, et al. Scaling accurate genetic variant discovery to tens of thousands of samples. *BioRxiv*, page 201178, 2018b.

Aditya Pratapa, Amogh P Jalihal, Jeffrey N Law, Aditya Bharadwaj, and TM Murali. Benchmark-

ing algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154, 2020.

Qian Qin and Jianxing Feng. Imputation for transcription factor binding predictions based on deep learning. *PLoS computational biology*, 13(2):e1005403, 2017.

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.

Alvaro Rada-Iglesias. Is h3k4me1 at enhancers correlative or causative? *Nature genetics*, 50(1): 4–5, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

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

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Icml*, 2011.

ER Riggs, DM Church, K Hanson, VL Horner, EB Kaminsky, RM Kuhn, KE Wain, ES Williams, S Aradhya, HM Kearney, et al. Towards an evidence-based process for the clinical interpretation of copy number variation. *Clinical genetics*, 81(5):403–412, 2012.

Matthew J Rossi, William KM Lai, and B Franklin Pugh. Simplified chip-exo assays. *Nature communications*, 9(1):1–13, 2018.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

Leyla Ruzicka, Douglas G Howe, Sridhar Ramachandran, Sabrina Toro, Ceri E Van Slyke, Yvonne M Bradford, Anne Eagle, David Fashena, Ken Frazer, Patrick Kalita, et al. The zebrafish information network: new support for non-coding genes, richer gene ontology annotations and the alliance of genome resources. *Nucleic acids research*, 47(D1):D867–D873, 2019.

Sirajul Salekin, Jianqiu Michelle Zhang, and Yufei Huang. Base-pair resolution detection of transcription factor binding site by deep deconvolutional network. *Bioinformatics*, 34(20):3446–3453, 2018.

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.

Roman Schulte-Sasse, Stefan Budach, Denes Hnisz, and Annalisa Marsico. Integration of multiomics data with graph convolutional networks to identify new cancer genes and their associated molecular mechanisms. *Nature Machine Intelligence*, pages 1–14, 2021.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Trevor Siggers and Raluca Gordân. Protein–dna binding: complexities and multi-protein codes. *Nucleic acids research*, 42(4):2099–2111, 2014.

Paja Sijacic, Marko Bajic, Elizabeth C McKinney, Richard B Meagher, and Roger B Deal. Changes in chromatin accessibility between arabidopsis stem cells and mesophyll cells illuminate cell type-specific transcription factor networks. *The Plant Journal*, 94(2):215–231, 2018.

Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

Montgomery Slatkin. Linkage disequilibrium—understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics*, 9(6):477–485, 2008.

Hamilton O Smith and KW Welcox. A restriction enzyme from hemophilus influenzae: I. purification and general properties. *Journal of molecular biology*, 51(2):379–391, 1970.

Gordon K Smyth and Terry Speed. Normalization of cdna microarray data. *Methods*, 31(4):265–273, 2003.

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.

Oksana Sorokina, Colin Mclean, Mike DR Croning, Katharina F Heil, Emilia Wysocka, Xin He, David Sterratt, Seth GN Grant, Thomas I Simpson, and J Douglas Armstrong. A unified resource and configurable model of the synapse proteome and its role in disease. *Scientific Reports*, 11 (1):1–9, 2021.

Tony D Southall, Katrina S Gold, Boris Egger, Catherine M Davidson, Elizabeth E Caygill, Owen J Marshall, and Andrea H Brand. Cell-type-specific profiling of gene expression and chromatin binding without cell isolation: assaying rna pol ii occupancy in neural stem cells. *Developmental cell*, 26(1):101–112, 2013.

Dominique Stehelin, Harold E Varmus, J Michael Bishop, and Peter K Vogt. Dna related to the transforming gene (s) of avian sarcoma viruses is present in normal avian dna. *Nature*, 260 (5547):170–173, 1976.

A. J. Stewart, S. Hannenhalli, and J. B. Plotkin. Why transcription factor binding sites are ten nucleotides long. *Genetics*, 192(3):973–985, Nov 2012.

Tatiana Subkhankulova, Fedor Naumenko, Oleg E Tolmachov, and Yuriy L Orlov. Novel chip-seq simulating program with superior versatility: ischip. *Briefings in Bioinformatics*, 2020.

John E Sulston and H Robert Horvitz. Post-embryonic cell lineages of the nematode, caenorhabditis elegans. *Developmental biology*, 56(1):110–156, 1977.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Charles Sutton and Andrew McCallum. An introduction to conditional random fields, 2010.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016a.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016b.

Vivian Tam, Nikunj Patel, Michelle Turcotte, Yohan Bossé, Guillaume Paré, and David Meyre. Benefits and limitations of genome-wide association studies. *Nature Reviews Genetics*, 20(8): 467–484, 2019.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

Peter Tessarz and Tony Kouzarides. Histone core modifications regulating nucleosome structure and dynamics. *Nature reviews Molecular cell biology*, 15(11):703–708, 2014.

Remi Torracinta, Laurent Mesnard, Susan Levine, Rita Shaknovich, Maureen Hanson, and Fabien Campagne. Adaptive somatic mutations calls with deep learning and semi-simulated data. *BioRxiv*, page 079087, 2016.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Gosia Trynka, Karen A Hunt, Nicholas A Bockett, Jihane Romanos, Vanisha Mistry, Agata Szperl, Sjoerd F Bakker, Maria Teresa Bardella, Leena Bhaw-Rosun, Gemma Castillejo, et al. Dense genotyping identifies and localizes multiple common and rare variant association signals in celiac disease. *Nature genetics*, 43(12):1193, 2011.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al. Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729, 2018.

David A Van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.

Bas van Steensel and Steven Henikoff. Identification of in vivo dna targets of chromatin proteins using tethered dam methyltransferase. *Nature biotechnology*, 18(4):424–428, 2000.

Juan M Vaquerizas, Sarah K Kummerfeld, Sarah A Teichmann, and Nicholas M Luscombe. A census of human transcription factors: function, expression and evolution. *Nature Reviews Genetics*, 10(4):252–263, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

Ana Viñuela, L Basten Snoek, Joost AG Riksen, and Jan E Kammenga. Aging uncouples heritability and expression-qtl in caenorhabditis elegans. *G3: Genes| Genomes| Genetics*, 2(5):597–605, 2012.

Cheng Wang, Mathias Niepert, and Hui Li. Lrmm: learning to recommend with missing modalities. *arXiv preprint arXiv:1808.06791*, 2018.

Haohan Wang, Aaksha Meghawat, Louis-Philippe Morency, and Eric P Xing. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 949–954. IEEE, 2017.

Qi Wang, Liang Zhan, Paul Thompson, and Jiayu Zhou. Multimodal learning with incomplete modalities by knowledge distillation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1828–1838, 2020.

Yaolai Wang, Feng Liu, and Wei Wang. Dynamic mechanism for the transcription apparatus orchestrating reliable responses to activators. *Scientific reports*, 2(1):1–6, 2012.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

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.

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.

Joshua D Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z Macosko. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887, 2019.

Harm-Jan Westra and Lude Franke. From genome to function by studying eqtls. *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease*, 1842(10):1896–1902, 2014.

Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*, 2017.

Edgar Wingender, Peter Dietze, Holger Karas, and Rainer Knüppel. Transfac: a database on transcription factors and their dna binding sites. *Nucleic acids research*, 24(1):238–241, 1996.

Edgar Wingender, Xin Chen, Reinhard Hehl, Holger Karas, Ines Liebich, Volker Matys, T Meinhardt, M Prüß, Ingmar Reuter, and Frank Schacherer. Transfac: an integrated system for gene expression regulation. *Nucleic acids research*, 28(1):316–319, 2000.

Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic classification of artifactual ica-components for artifact removal in eeg signals. *Behavioral and Brain Functions*, 7(1):1–15, 2011.

Ledell Yu Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. Starspace: Embed all the things! In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Yuan Xue, Arunkanth Ankala, William R Wilcox, and Madhuri R Hegde. Solving the molecular diagnostic testing conundrum for mendelian disorders in the era of next-generation sequencing: single-gene, gene panel, or exome/genome sequencing. *Genetics in Medicine*, 17(6):444–451, 2015.

Yang Yang, De-Chuan Zhan, Xiang-Rong Sheng, and Yuan Jiang. Semi-supervised multi-modal learning with incomplete modalities. In *IJCAI*, pages 2998–3004, 2018.

Yaping Yang, Donna M Muzny, Fan Xia, Zhiyv Niu, Richard Person, Yan Ding, Patricia Ward, Alicia Braxton, Min Wang, Christian Buhay, et al. Molecular findings among patients referred for clinical whole-exome sequencing. *Jama*, 312(18):1870–1879, 2014.

Ren Yi, Pi-Chuan Chang, Gunjan Baid, and Andrew Carroll. Learning from data-rich problems: A case study on genetic variant calling. *arXiv preprint arXiv:1911.05151*, 2019a.

Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, page 101552, 2019b.

Han Yuan, Meghana Kshirsagar, Lee Zamparo, Yuheng Lu, and Christina S Leslie. Bindspace decodes transcription factor binding signals by large-scale sequence embedding. *Nature methods*, 16(9):858–861, 2019.

Junying Yuan, Shai Shaham, Stephane Ledoux, Hilary M Ellis, and H Robert Horvitz. The c. elegans cell death gene ced-3 encodes a protein similar to mammalian interleukin-1β-converting enzyme. *Cell*, 75(4):641–652, 1993.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33, 2020.

Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):1–15, 2017.

Mehdi Zarrei, Jeffrey R MacDonald, Daniele Merico, and Stephen W Scherer. A copy number variation map of the human genome. *Nature reviews genetics*, 16(3):172–183, 2015.

Haowen Zhang, Chirag Jain, and Srinivas Aluru. A comprehensive evaluation of long read error correction methods. *BMC genomics*, 21(6):1–15, 2020.

Xiuwei Zhang, Chenling Xu, and Nir Yosef. Simulating multiple faceted variability in single cell rna sequencing. *Nature communications*, 10(1):1–16, 2019.

Yue Zhao and Gary D Stormo. Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature biotechnology*, 29(6):480–483, 2011.

An Zheng, Michael Lamkin, Yutong Qiu, Kevin Ren, Alon Goren, and Melissa Gymrek. A flexible chip-sequencing simulation toolkit. *BMC bioinformatics*, 22(1):1–10, 2021.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

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

Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Fangjie Zhu, Lucas Farnung, Eevi Kaasinen, Biswajyoti Sahu, Yimeng Yin, Bei Wei, Svetlana O Dodonova, Kazuhiro R Nitta, Ekaterina Morgunova, Minna Taipale, et al. The interaction landscape between transcription factors and the nucleosome. *Nature*, 562(7725):76–81, 2018.

Zhihong Zhu, Futao Zhang, Han Hu, Andrew Bakshi, Matthew R Robinson, Joseph E Powell, Grant W Montgomery, Michael E Goddard, Naomi R Wray, Peter M Visscher, et al. Integration of summary data from gwas and eqtl studies predicts complex trait gene targets. *Nature genetics*, 48(5):481–487, 2016.

Marinka Zitnik, Francis Nguyen, Bo Wang, Jure Leskovec, Anna Goldenberg, and Michael M Hoffman. Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91, 2019.

Justin Zook, Jennifer McDaniel, Hemang Parikh, Haynes Heaton, Sean A Irvine, Len Trigg, Rebecca Truty, Cory Y McLean, Francisco M De La Vega, Marc Salit, et al. Reproducible integration of multiple sequencing datasets to form high-confidence snp, indel, and reference calls for five human genome reference materials. *BioRxiv*, page 281006, 2018.

Justin M Zook, David Catoe, Jennifer McDaniel, Lindsay Vang, Noah Spies, Arend Sidow, Ziming Weng, Yuling Liu, Christopher E Mason, Noah Alexander, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific data*, 3:160025, 2016.

Justin M Zook, Nancy F Hansen, Nathan D Olson, Lesley Chapman, James C Mullikin, Chunlin Xiao, Stephen Sherry, Sergey Koren, Adam M Phillippy, Paul C Boutros, et al. A robust benchmark for detection of germline large deletions and insertions. *Nature biotechnology*, pages 1–9, 2020.