ANALYSIS ON PROTEIN-DNA INTERACTION AND GENE EXPRESSION

JIYUN ZHOU

PhD

The Hong Kong Polytechnic University

This programme is jointly offered by The Hong Kong Polytechnic University and Harbin Institute of Technology

2019

The Hong Kong Polytechnic University
Department of Computing

Harbin Institute of Technology
School of Computer Science and Technology

Analysis on Protein-DNA Interaction and Gene Expression

Jiyun Zhou

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

July 2018

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_______________________________ (Signed)

Jiyun Zhou (Name of student)

Abstract

Gene expression is pivotal in genomic biology. As experimental methods for gene expression prediction are costly and labor-consuming, there is an urgent to develop high-performance computational methods for gene expression predictions. As gene expressions are mainly regulated by interactions between DNAs and transcription factors (TFs) which is a type of proteins with special function, analysis on TF-DNA interactions may facilitate the prediction of gene expressions.

This thesis focuses on the analysis of protein-DNA interactions and gene expression. We attempt to address issues in four aspects in gene expression analysis including (1) protein second structure prediction, (2) DNA binding residue prediction, (3) TF binding site (TFBS) prediction and (4) gene expression prediction. Our contribution mainly consists of four parts.

For protein second structure prediction, we present a novel deep learning based prediction method, referred to as CNNH_PSS, which uses a multi-scale CNN with highway to capture both local context and longer-range dependencies. In CNNH_PSS, a specific part of the information is delivered from a current layer to the output of the next one by highways to keep local context and the other parts of information are delivered from current layer to the input of the next one to capture dependencies among residues with longer distance. Therefore, the feature space learned by CNNH_PSS contains both local context and long-range interdependencies.

For DNA-binding residue prediction, the research goal is to learn relationships among residues for the prediction of DNA-binding residues. In this thesis, four prediction methods are proposed to learn relationships among residues. The first method applies PSSM (Position Specific Score Matrix) distance transformation to encode local pairwise relationships between neighboring residues. The second method applies Convolutional Neural Network to learn relationships among several neighboring residues. The third method applies Long Short-Term Memory to learn both local relationships and long-range relationships among residues. The last method makes use of two sliding windows to learn sequence relationships and structure relationships, respectively.

For TF-binding site (TFBS) prediction, three prediction methods are proposed. First, a novel method is proposed to capture higher order relationships among nucleotides by applying two CNNs on histone modifications and DNA sequence, respectively. Second, a multi-task framework. is proposed to particular address data sparseness issue by leveraging on cross-cell-type information available. The method learns common features from multiple cell-types using a shared CNN and individual features by a private CNN for each cell-type. The last method is proposed for for the cross-TF TFBS prediction by learning TFBSs from other TFs in the training set. This method can further address the non-available issue in the current training data.

Current gene expression prediction methods can only be used for cell-types or tissues in which ChIP-seq datasets for most important TFs are labeled. However, for most cell-types or tissues in human beings, the ChIP-seq datasets for most TFs are not available. In this work, a novel prediction method is proposed to first predict TFBSs by our cross-cell-type prediction method and the cross-TF prediction method. They are then combined with histone modifications to learn feature representations for genes. The advantage of this method is that it predict gene expressions for any cell-type regardless of the availability of the TFBS of the considered TFs. Our proposed method can automatically extract combinatorial relationships among histone modifications and TFBSs. These relationships and TFBSs play very important roles in regulating gene expression and facilitate the understanding of gene expression regulation for humans.

List of Publications

Published papers

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. “Prediction of TF-binding site by inclusion of higher order position dependency.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018. (accepted on January 1, 2019)

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. “EL_LSTM: Prediction of DNA-binding residue from Protein sequence by Combining Long Short Term Memory and Ensemble Learning.” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, PP(99):1-1. 2018.

- **Jiyun Zhou**, Hongpeng Wang, Zhishan Zhao, Ruifeng Xu and Qin Lu. “CNNH PSS: Protein 8-class Secondary Structure Prediction by Convolutional Neural Network with Highway.” *BMC Bioinformatics*, 19(Suppl 4): 60. 2018. DOI : 10.1186/s12859-018-2067-8.

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu and Yulan He and Hongpeng Wang. “EL_PSSM-RT: DNA-binding residue prediction by integrating ensemble learning with PSSM Relation Transformation.” *BMC Bioinformatics*, 18(1): 379. 2017. doi:10.1186/s12859-017-1792-8.

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. “Prediction of DNA-binding residues from sequence information using convolutional neural network.” *International Journal of Data Mining and Bioinformatics*,17(2): 132-152. 2017. doi: 10.1504/IJDMB.2017.084265

- **Jiyun Zhou**, Ruifeng Xu, Yulan He, Qin Lu, Hongpeng Wang and Bing Kong. “PDNAsite: Identification of DNAbinding Site from Protein Sequence by Incorporating Spatial and Sequence Context". *Scientific Reports*, 6: 27653. 2016. doi: 10.1038/srep27653.

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. "CNNsite: Prediction of DNA-binding Residues in Proteins Using Convolutional Neural Network with Sequence Features". *In proceedings of the 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017:78-85. Shenzhen, China, Dec 15-18, 2016.
- Ruifeng Xu, **Jiyun Zhou**,Hongpeng Wang, Yulan He, Xiaolong Wang and Bin Liu. "Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation." *BMC Systems Biology*, 9(S1): S10. 2015. doi: 10.1186/1752-0509-9-S1-S10.
- Ruifeng Xu, **Jiyun Zhou**, Bin Liu, Yulan He, Quan Zou, Xiaolong Wang and Kuo-Chen Chou. "Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach." *Journal of Biomolecular Structure and Dynamics*, 33(8): 1720-1730. 2015. doi: 10.1080/07391102.2014.968624.

Papers under review

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. "PDBR_TF: Cross-TF TF Binding Site Prediction by Incorporating Predicted DNA Binding Residues of TFs." *the 27th conference on Intelligent Systems for Molecular Biology and the 18th European Conference on Computational Biology*.

Papers in preparation

- **Jiyun Zhou**, Qin Lu, Ruifeng Xu, Lin Gui and Hongpeng Wang. "MTTFsite: Cross-cell-type TF Binding SitePrediction by Multi-task Learning Framework." *Bioinformatics*, target submission in January, 2019.

Acknowledgements

The two years in PolyU were the most enjoyable and challenging period in my life and I will never forget the wonderful time I spent in PolyU. Many people have kindly helped me during the two years and I would like to express my sincere gratitude to them here.

First of all, I would like to express my utmost thanks to my supervisor, Professor Qin Lu, for her constant encouragement and guidance. Prof. Lu has walked me through all the stages of my study period. Her logic reasoning, serious attitude and rigorous scholarship towards research have a great influence on me. I clearly remember many times that Prof. Lu helped me to polish my papers and thesis until late night and even at weekends. She has helped me to enhance my paper writing ability greatly. In addition, Prof. Lu was very patient with me and gave me enough time to go through my TOEFL test preparations. Her insistence on consistency, coherence, and logic thinking will influence the rest of my life.

Secondly, I would like to thank my co-supervisor Prof. Ruifeng Xu for his valuable comments on my thesis. Prof. Xu has helped me to develop the fundamental and essential academic competence and provided me with necessary materials, advice of great value and inspiration of new ideas with extraordinary patience. I would also like to thank my co-supervisor Prof. Hongpeng Wang for his valuable comments on my researches and thesis.

Thirdly, I am very grateful to Professor Maggie Wenjie Li, who gave me many thoughtful suggestions and valuable comments on my research contents.

Fourthly, I would like to thank all my colleagues and friends. Thanks to Lin Gui, Minglei Li and Yunfei Long for their help with my experiments and valuable discussions. Thanks to Dan Xiong for her elaborate knowledge on daily life, which made my life in HK much easier. Thanks to Chengyao Chen for the inspiring discussions that helped me to polish my papers. Thanks to Yanran Li for her profound knowledge on different re-

search topics, which inspired me a lot. I am very much encouraged by her enthusiasm for research and life in general. Thanks to Dr. Pingping Liu. Discussions between different disciplines really broadened my horizon. Thanks to the friends I made in Poly U including Mohammed Aquil Maud Mirza, Zhaoyan Shen, Muhui Jiang and Zhijian He. Your friendship made my life in PolyU a very pleasant one.

Fifthly, I give my wholehearted thanks to my grandmother Xiuying Huang and my younger sister Xiaojing Zhou for their love, tolerance and unquestioned support.

Last but not least, I would like to thank my wife Ruihua Wang and her family for their unconditional support and sacrifice during my PhD study. Most importantly of all, I am forever grateful to Ruihua for bringing my daughter Muci to this world during my PhD study. Life has never been the same since I saw Muci on my wife's bedside. Thanks to them both for giving me such a wonderful family. They made my work meaningful and my life full of hope and happiness.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A genome is an organism's complete set of DNA, including all the genes (the coding regions) and the non-coding DNA. The genome of an organism contains all the information needed to build and maintain that organism. In humans, each cell contains a copy of the entire genome more than 3 billion DNA base pairs.

Genome analysis has important applications in several fields including medicine, biotechnology and anthropology. Firstly, many deceases have genetic markings. By analyzing the genomic data collected from a large mount of patients with a specific disease, researchers can better understand the genetic basis of the disease [125, 100]. This can help to develop testing methods to identify many types of genetically linked deceases. Secondly, the analysis of genomic data has made applications of synthetic biology become feasible. For example, researchers of the J.Craig Venter Institute created a partially synthetic species of bacterium, Mycoplasma laboratorium according to the genome of Mycoplasma genitalium [10]. The application in synthetic biology will allow researcher to create many new synthetic species with desired properties, such as synthesized drugs with specified efficacy. Thirdly, by analyzing genomic data from a given population, conservationists can help a species without genetic diversity to obtain genetic diversity [6].

In order to understand the genomic formation of bi-organisms of human beings, we must first obtain the sequence of more than 3 billion DNA base pairs in human genes. Scientists proposed the Human Genome Project (HGP) in 1985. HGP was started in 1990 and finished in 2003. With the completion of HGP, these 3 billion base pairs of human genome are sequenced and available for researchers. This means that life science has entered the post genome era [191]. However, data in the scale of 3 billion is too large for any manual study of the genome sequence is not viable. Therefore, developing computational methods for genome data analysis is almost the only way to do genome analysis.

1.1 Biology Basics

Before introducing computational methods for human genome analysis, some basic biology knowledge is introduced to provide background information in the genomics domain. The main research areas of the post genome era includes functional genomics and pharmacogenomics. Functional genomics [191] aims to make use of the vast wealth of data given by genome sequencing projects and RNA sequencing projects to describe gene functions and interactions. RNA is a polymeric molecule essential in various biological roles in coding, decoding, regulation, and expression of genes. There are three main categories of RNA: (1) messenger RNA (mRNA) used to convey genetic information, (2) transfer RNA (tRNA) used to deliver amino acids to the ribosome and (3) ribosomal RNA (rRNA) used to like amino acids together to form proteins. Functional genomics mainly focuses on the dynamic process in aspects such as gene transcription, translation and regulation of gene expressions. Pharmacogenomics [69], on the other hand, studies the role of genomes in drug response, which mainly analyzes how the genetic makeup of an individual may affect his/her response to drugs. More specifically, pharmacogenomics studies the influence of acquired and inherited genetic variation on drug response in patients by correlating gene expression with Pharmacokinetics (drug absorption, distribution, metabolism, and elimination) and Pharmacodynamics (effects mediated through a drug's biological targets). Both functional genomics and pharmacogenomics are related to gene expressions.

This is why research on gene expression is an important research area of both Functional genomics and Pharmacogenomic. Genes are subunits of DNA, which carry the genetic blueprints used to make up for proteins. Figure 1.1 shows the structure for of a eukaryotic protein coding gene. A gene consists of three parts: the regulatory sequence in the 5’ end, open reading frame and the regulatory sequence in the the 3’ end. In the regulatory sequence in the 5’ end, genes contain a promoter sequence. The promoter is recognized and bound by transcription factors and RNA polymerase to initiate transcription. The recognition typically occurs as a consensus sequence like the TATA box. A gene can have more than one promoter, resulting in messenger RNAs (mRNA) that differ in how far they extend in the 5’ end. Highly transcribed genes have ”strong” promoter sequences that form strong associations with transcription factors, thereby initiating transcription at a high rate. Others genes have ”weak” promoters that form weak associations with transcription factors and initiate transcription less frequently. In the regulatory sequence in the 3’ end, genes contain a terminator sequence. The terminator can initiate the transcription. Additionally, genes can have regulatory regions many kilobases upstream or downstream of the open reading frame that alter expression. These act by binding to transcription factors which then cause the DNA to loop so that the regulatory sequence (and bound transcription

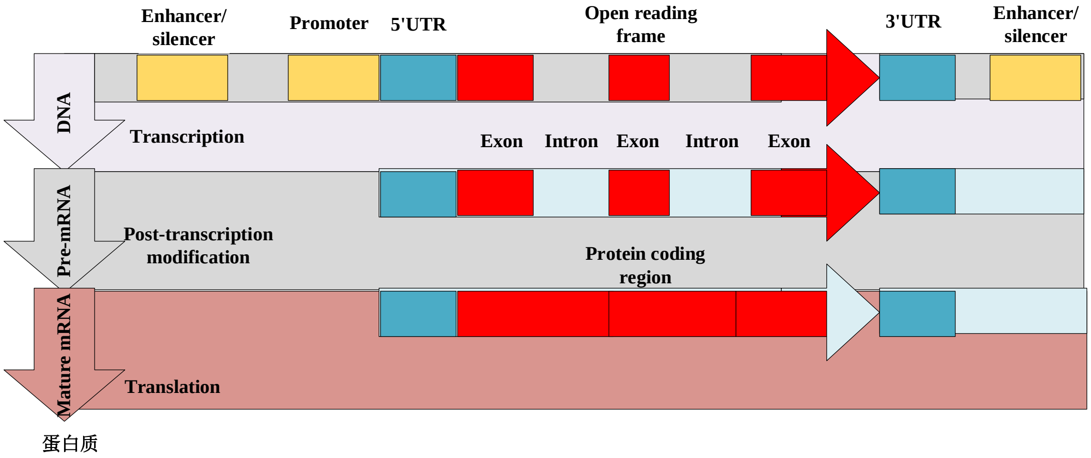


Figure 1.1: The structure of a eukaryotic protein-coding gene.

factor) become close to the RNA polymerase binding site.

Every gene contains a particular set of instructions that encode a specific functional protein. The genetic code stored in a gene is ”interpreted” by the so called gene expression, and the properties of the expression give rise to the organism’s phenotype. The word ”interpreted” implies that gene expression refers to a process. Thus, **gene expression** generally refers to the process by which the information from genes is used to synthesize functional gene products and Figure 1.2 shows the procedure for gene expressions. The products for most gene expressions are proteins, which go on to perform essential functions as enzymes, hormones and receptors. But for non-protein coding genes including tRNA coding genes and small nuclear RNA (snRNA) coding genes, the products are functional RNA. Figure 1.1 shows that gene expression is accomplished by several steps: the transcription, the RNA splicing, the translation, and the post-translational modification of proteins. Transcription is the process by which segments of DNA are copied into RNA (especially mRNA) by the RNA polymerase. RNA splicing is the step to edit the nascent precursor mRNA produced by the transcription into mature mRNA, by which intros are removed and exons are joined together. Translation is the process by which mature mRNAs produced by the RNA splicing process are decoded into amino acid chains or polypeptides in ribosome. Figure 1.3 shows the procedure for the translation of protein encoding genes. In the translation of protein encoding genes, three consecutive bases compose a codon. Every codon can produce a amino acid and the produced amino acids can compose proteins. Finally, protein post-translation modification refers to the covalent and generally enzymatic modification of proteins, by which the polypeptides or the amino acid chains produced by the translation process are transformed into mature proteins. [177]

Gene expression level refers to the level at which a particular gene is expressed within a cell, tissue or organism. Usually, gene expression level are measured by two types of methods including the mRNA quantification method and the protein quantification method. The mRNA quantification method provides the amount of mRNA produced by

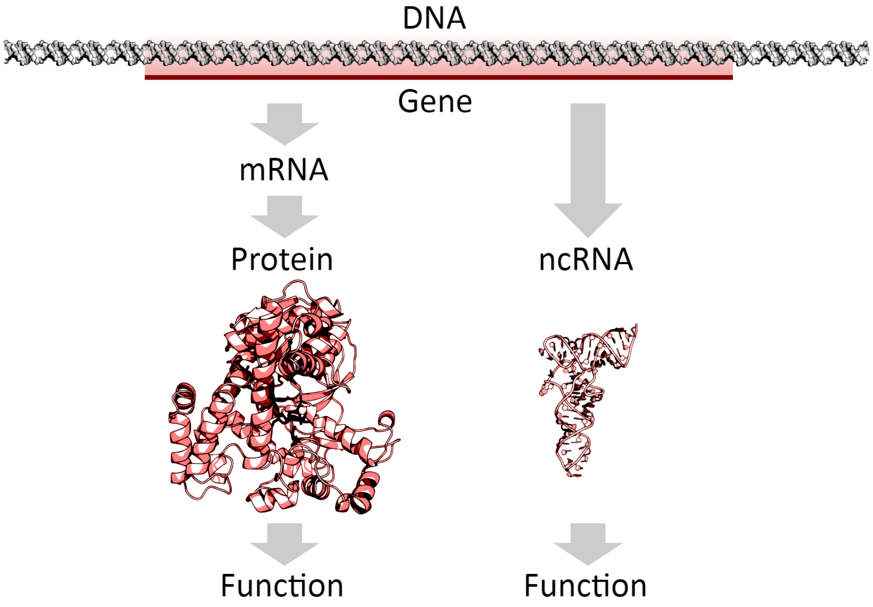


Figure 1.2: The process of gene expressions.

the transcription process in the gene expression of target genes and the amount for mRNA can be measured by several metrics, such as the size of mRNA molecules [77], mRNA abundance and mRNA concentration [119]. The protein quantification method provides the amount of protein products by the whole gene expression process and the amount of proteins can be measured by several metrics, such as the size of protein molecules, protein abundance and protein concentration. In most literature [153, 62, 51, 114, 46, 40, 76], the terms gene expression and gene expression level are used interchangeably because gene expression level is the end result of gene expression as a process. In the remainder text of this thesis, when there is not confusion for the two semantic meanings, the simpler form of ”gene expression” will be used.

Gene expression is mainly determined by gene expression regulation, which is a very complex system of mechanisms to control the amount and timing of appearance of the functional products of gene expression by monitoring cells and their environment. The gene expression regulation process takes internal and external signals, analyzes them, and then decides if a gene product is needed and how much is needed. Gene expression regu-

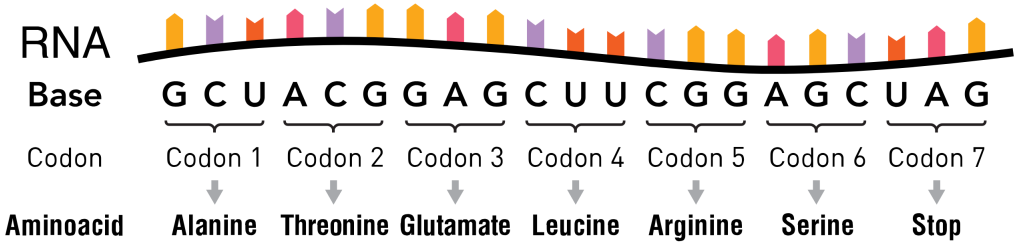


Figure 1.3: The process of translation.

lation involves two step: transcriptional regulation and translational regulation. Transcriptional regulation is the means by which the conversion of DNA to RNA is regulated to orchestrate gene expression [28]. Translational regulation refers to the process by which the synthesis of proteins from its mRNA is controlled to the level of gene expression [48].

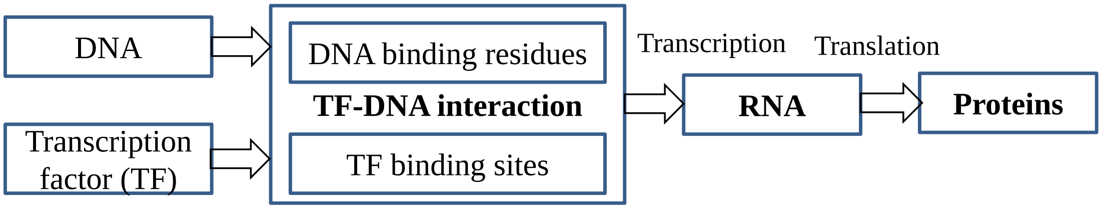


Figure 1.4: The procedure of the transcription regulation.

Transcriptional regulation is completed by sequence-specific interactions between **transcription factor (TF)** and DNA at a point upstream to the target gene [99, 132, 123, 171], which is the **promoter**. TFs are proteins that control the rate of transcription of genetic information from DNA to mRNA by binding to a specific DNA sequence [75]. The functions of TFs are to regulate - turn on and off - genes in order to make sure they are expressed in the right cell at the right time and in the right amount throughout the life of the cell and the organism. The procedure of transcription regulation is shown in Figure 1.4. Figure 1.4 shows that TF-DNA interactions is an important component in the transcription regulation and TF-DNA interactions are mainly made up by DNA binding residues and TF binding sites. Therefore, the Analysis on TF-DNA interaction mainly focuses two components: the residues in TFs which can bind to DNA, referred to DNA binding residues and the sites in DNA which can bind to TFs, referred to as TF binding sites (TFBSs).

DNA binding residues are the residues in TFs which can interact with its specific DNA sequence. More specifically, a DNA binding residue is defined as the residue of which any side chain or backbone atom falls within a cut-off distance of 3.5 angstroms from any atom of its partner DNA molecule. As DNA binding residues are the basic elements in TFs of TF-DNA interactions, the identification of DNA binding residues is very important to understand the recognition mechanism between TF and DNA as well as the mechanism of gene transcriptional regulation. The identification of DNA binding residues also provides some basic knowledge for understanding the pathogenesis of several diseases. For example, the DNA binding residues on the repressor protein P53 can provide knowledge about certain diseases, such as some kinds of tumors. As secondary structure can be used to encode peptides and peptides contain relevance among multiple neighbor residues [89], the formation of DNA binding residues closely correlates with their secondary structures. **Secondary structure** of a TF refers to the three dimensional form of its local segments, which is defined by the pattern of hydrogen bonds between the amine hydrogen and carbonyl oxygen. Secondary structures contain three categories including helix, strand and coil. However, the secondary structures for most TFs are unknown and experimental techniques for TF secondary structure identification are very labor-intensive and costly. Therefore, Secondary structure prediction for TFs is important for DNA binding residue prediction [89, 121, 148].

TF binding sites (TFBSs) are DNA fragments where TFs can bind to. They are different from DNA binding residues because they are a part of DNA sequence and bound by TFs. More specifically, TFBSs are defined as short and often degenerate DNA sequences (typically 4 to 30 base pairs long) that are bound by one or more TFs with various functions. According to their functions, TFBSs can be a part of either the promoter or enhancer region of genes. **Promoters** sit upstream to genes and contain three important regions including regulatory protein binding site, TFBS and RNA polymerase binding site. **Enhancers** are usually even farther upstream of a gene. Binding of TFs to an enhancer will stimulate transcription at a higher rate compared to a bound promoter. So, binding of a TF to an enhancer accelerates the transcription of a target gene. Therefore, TFBSs play an important role in gene expression regulation [204, 147, 34].

There are more than 200 different cell-types or tissues for humans. A **cell-type** is a set of morphologically or phenotypically related cells within a species. A multicellular organism may contain a number of widely differing and specialized cell types, such as muscle cells and skin cells in humans, that differ both in appearance and function yet are genetically identical. A **tissue** is an ensemble of similar cells from the same origin that together carry out a specific function. As both the term cell-type and tissue are used to denote a set of cells with a specific function, the term cell-type is used in the rest of the thesis to refers to both cell-type and tissue. All cell-types in an organism have a same genome, but different cell-types have different gene expressions due to differential gene regulations. Different gene regulations are caused by different binding sites of TFs in different cell-types. Moreover, In each cell-type, the expression of genes are regulated collaboratively by more than 2600 TFs. different TFs usually have different sequences and biology functions. So, they may bind to different positions of the genome to play different roles in regulations of gene expression. Therefore, when developing prediction algorithms, TFBSs for different TFs in a same cell-type are different and the TFBSs of a same TF for different cell-types also are different.

In addition to DNA binding residues and TFBSs, histone modification is another important type of factors for gene expression regulation. A Histone is one kind of proteins which can interact with DNA. Compared with TFs, a histone binds to DNA less specifically. Nucleosomes, as the basic units of chromosome, are formed by two copies of four core histone proteins including H2A, H2B, H3 and H4 and 147 base pair (bp) DNA sequence. These histone proteins are subject to a number of post-translational covalent modifications, such as methylation, acetylation and phosphorylation [7, 18]. These modifications can change the structure and function of chromatin by altering the charge of the nucleosome and/or by recruiting enzymes individually or in combination [81]. These **histone modifications** can form a ‘histone code’, which is read out by proteins to give rise to various downstream effects on gene expression regulation [90].

DNase I hypersensitive sites (DHSs) are regions of chromatin that are sensitive to cleavage by the DNase I enzyme. In these specific regions of the genome, a chromatin has lost its condensed structure, exposing the DNA and making it accessible to DNA degradation by enzymes, such as DNase I. These accessible chromatin zones are functionally related to transcriptional activities since this remodeled state is necessary for the binding of proteins such as TF. Many works in the literature [61, 60, 27] have found that DHSs are closely correlated to regulatory regions including promoters, distal enhancers, insulators and active histone marks. Moreover, DNase-seq were proposed for profiling DHSs in a genome-wide fashion [155]. DNase-seq (DNase I hypersensitive sites sequencing) is a experimental method in molecular biology used to identify the location of regulatory regions which is based on the genome-wide sequencing of regions sensitive to cleavage by DNase I. Therefore, DNase-seq data for DHSs is another important type of features for gene expression prediction.

1.2 Problem Statements and Research Objectives

With the development of technologies, several technologies have been proposed for measuring gene expression in an automated manner, including microarry experiment [165] and RNA-seq technology [119, 44, 186]. Microarry experiment [165] contains GeneChip oligonucleotide probe based arrays and high density bead arrays. RNA-seq technology [119, 44, 186] uses next-generation sequencing methods to quantify RNA in a cell or tissue sample. Next-generation sequencing, also called second-generation sequencing, represents several high-throughput approaches to DNA sequencing by massively parallel processing [176]. However, there exists several fundamental impediments for current profiling technologies. First, the RNA-seq technology and microarry experiment is too costly to adapt in research and clinical applications alike. Second, the requirements for data storage and high computation complexity is a great challenge. Last but not least, the microarray experiment often inevitably miss a larger number of data, which adversely affects downstream analysis. Therefore, it is urgent to propose computational methods for the prediction of gene expression levels.

This thesis focuses on the prediction of gene expression in specific cells or tissues. Generally speaking, to find high quality computational prediction methods for gene expression requires four parts. The first part is to investigate methods for predicting secondary structures of TFs. The second part is to investigate methods for predicting DNA binding residues based on predicted secondary structures of TFs. The third part is to investigate methods for predicting TFBSs from DNA based on predicted DNA binding residues. The last part is to investigate methods for predicting gene expression levels based on predicted TFBSs features and histone modifications. The major problems in current works that motivate this work and the research objectives are given based on the these four parts. Note that the second part and the third part are two important areas of study on TF-DNA interaction. Since most cell-types or tissues lack TF-DNA interaction information measured by experiments, this work put particular emphasis on the study of TF-DNA interaction including DNA residue prediction and TFBS prediction.

Protein secondary structure prediction

Secondary structure prediction of proteins is the inference of secondary structure of protein fragments based on their amino acid sequence. In bioinformatics and theoretical chemistry, secondary structure prediction is very important for medicine and biotechnology such as drug design [8] and the design of novel enzymes. Current computational approaches for secondary structure predictions can be divided into 3 categories. The first category uses statistical models to analyze the probability of secondary structure elements for individual amino acids [13]. The second category is evolutionary information based methods. These methods usually used position-specific scoring matrices (PSSM)[16] from PSI-BLAST for prediction. The last category applies deep learning method to learn feature representations for residues. However, statistical methods and evolutionary information based methods cannot extract both local context and long-range dependencies. The third category of methods currently either has limited ability to extract both local context and long-range dependencies or the models are too complex computationally.

Objective 1: To investigate more efficient methods to extract both local context and long-range dependencies use deep learning models.

DNA binding residue prediction

TFs are the most intensively studied DNA-binding proteins which form interactions with DNA by some specific interaction mechanism[7]. They activate or inhibit the transcription of genes by binding to particular DNA sequences closing to their promoters or enhancers through their DNA binding residues. DNA binding residues are also important for the functions of TFs. The mutations of some DNA binding residues may predispose individuals to disease. Thus, the prediction of DNA binding residues is not only important for understanding the gene regulation process but also helpful for annotating the function of proteins.

Many computational methods have been proposed for the prediction of DNA-binding residues. Commonly used features include three types: sequence features, evolutionary features, and structure features [13–16]. However, most current methods based these features did not consider the relationships among different residues because data sparseness issue. As the function and the structure of a target residue are often closely related to its contextual residues, we hypothesize that relationships among a residue and its context are important for the predictions of it function.

Objective 2: To investigate effective ways to extract relationships among target residues and their contextual residues to learned feature representations for residues and then applies the learned representation for DNA binding site prediction.

TF binding site prediction

Genes often have several TFBSs around their coding region. Profiling of TFBSs is an important yet challenging problem because TFBSs are often short and disperse. Furthermore, the form of TFBSs vary depending on the type of tissues, the stage of development, and the physiological condition. Such condition-dependent characteristic makes the problem of TFBSs prediction even more challenging. Many classical computational methods used PWM to represent TFBS [159, 160]. The basic assumption of PWM based methods is that each nucleotide within a TFBS participates independently in the corresponding DNA-protein interaction. In order to incorporate nucleotide dependency into prediction, a new approach called dinucleotide weight matrix (DWM) was proposed recently [149]. In addition to DWM, TFFM proposed by Mathelier and Wasserman can also capture nucleotide dependency for prediction [110]. Although DWM [149], TFFM [110], and other methods [190, 111] can use nucleotide dependency for prediction, current works cannot capture higher order dependency.

Objective 3: To investigate effective methods to extract higher order dependencies for TFBS prediction.

Cross-cell and cross-TF binding site prediction

Currently, to develop a model for a target TF in a specific cell-type requires collecting TFBSs of the target TF of that cell-type for training. However, for many target TFs in a specific cell-type, the target TFs do not have any training sample in the considered cell-type. Even if the TFBSs of the target TFs in different cell-types are different, they may have a common TF-DNA interaction mechanism.

Objective 4: To explore effective cross-cell TFBS prediction methods for cell-types without any TFBS by using available TFBSs from other cell-types.

The cross-cell TFBS prediction methods developed in Objective 4 can be used to predict TFBSs for TFs in a specific cell-type by using training samples from other cell-types. However, many TFs do not have any TFBS in any cell-type. So, the method developed in Objective 4 cannot be applied. Fortunately, in a specific cell-type, there exist many other TFs which have TFBSs identified by experimental methods. Even though a majority of TFs have different sequences and biology functions, some TFs do have similar sequences and biology function. As these TFs similar in sequences and biology functions tend to bind to similar positions of the genome, the TFBSs of these TFs can be collected to build a model for the target TFs.

Objective 5: To explore effective cross-TF TFBS prediction methods to predict TFBSs for TFs by using the TFBSs of other TFs from the same cell-type.

Gene expression prediction

Gene expression refers to the amount of RNAs or proteins that are produced by genes under specific circumstance or in a specific cell-type or tissue. The measurement of gene expression is an indispensable part in the study of life science. Microarry experiment and RNA-seq technology have been proposed for measuring gene expression. However, RNA-seq technology and microarry experiment are too labor-intensive and costly to be adapted in research and clinical applications alike.

Many computational methods have been proposed for gene expression prediction. Commonly used features in these works include two types: histone modifications and TFBSs. Both histone modifications and TFBSs play important roles in gene expression prediction. However, many methods use only histone modifications in predictions due to the lack of TFBSs in a large number of cell-types or tissues. Although many existing methods do use histone modifications and TFBSs, their TFBSs were identified by experimental techniques. The experimental techniques for TFBS identification often are labor-consuming and costly, so a large number of cell-types or tissues lack TFBSs for most TFs. This situation limits the application of the these methods to only a few of cell-types or tissues which have TFBSs for enough number of TFs.

Objective 6: To explore a new prediction method for gene expression by using the predicted TFBSs by cross-cell prediction method or cross-TF prediction method and histone modifications.

1.3 Thesis Outline

Chapter 2 introduces background knowledge related to this thesis, mainly including literature study of protein secondary structure prediction, DNA binding residue prediction, TF binding site prediction, gene expression prediction and deep learning models.

Chapter 3 introduces a novel convolutional neural network based method with highway for protein secondary structure prediction. Our proposed method can make good use of both local context and long-distance dependencies around target residues for secondary structure prediction [214].

Chapter 4 introduces four methods for DNA binding residue prediction, including (1) EL_PSSM-RT which uses Position Specific Score Matrix (PSSM) relation transformation (PSSM-RT) to encode pair relationships between residues [213]; (2) CNNsite which applies convolutional neural network (CNN) to extract relationships among multiple residues [211, 212]; (3) EL_LSTM which applies Long Short-Term Memory network (LSTM) to extract both local context and long-distance relationships; and (4) PDNAsite which uses sequence sliding window and spatial sliding window to encode both sequence context and spatial context [215].

Chapter 5 introduces three works for TFBS prediction: (1) a CNN based method using sequence features and histone modification features to automatically extract both first order dependency and higher order dependency; (2) a cross-cell TFBS prediction method MTTFsite using TFBSs from multiple cell-types to train a prediction model by multi-task learning framework; and (3) a cross-TF TFBS prediction method PDBR_TF using TFBSs from multiple other TFs to train a prediction model. It combines predicted DNA binding sites predicted by CNNsite, DNA sequence and histone modification features to encode feature representation.

Chapter 6 proposes a novel gene expression prediction method TFChrome by using histone modification features and TFBSs, in which the TFBSs are predicted by our cross-cell TFBS prediction method MTTFsite or cross-TF TFBS prediction method PDBR_TF instead of experimental techniques.

Chapter 7 concludes the thesis by summarizing the main contributions and limitations as well discussions on future works on gene expression analysis.

Chapter 2

Background

This chapter gives an overview of research related in four areas of our research including (1) protein secondary structure prediction, (2) DNA binding residue prediction, (3) TF binding site prediction, and (4) gene expression prediction. The last part also gives an overview of deep learning models which is related to our proposed methods.

2.1 Overview of protein secondary structure prediction

The concept of secondary structure was first introduced by Linderstrm-Lang at Stanford in 1952[93, 145] to represent the three dimensional form of local segments of proteins. Protein secondary structure is defined by the pattern of hydrogen bonds between the amine hydrogen and carbonyl oxygen. There are two ways used for the classification of protein secondary structures: three-category classification(Q3) and eight-category classification(Q8). Q3 classifies target residues into helix(H), strand(E) and coil(C). The more comprehensive Q8 classifies target residues into 3-turn helix(G), 4-turn helix(H), 5-turn helix(I), hydrogen bonded turn(T), extended strand in parallel and/or anti-parallel β-sheet conformation(E), residue in isolated β-bridge (B), bend(S), and coil(C)[73, 209, 197]. Most recent state-of-the-art methods use Q8 classification to evaluate their proposed methods. In order to compare with state-of-the-art methods, the work in this thesis also uses Q8 classification for secondary structure prediction.

Three experimental methods were proposed to identify secondary structures for proteins including far-ultraviolet circular dichroism, infrared spectroscopy, and NMR spectrum. Far-ultraviolet circular dichroism predicts pronounced double minimum at 208 and 222 nm as α-helical structure and single minimum at 204 nm or 217 nm as random-coil or β-sheet structure, respectively[133]. Infrared spectroscopy uses the difference in bond oscillations of amide groups for prediction[115] while NMR spectrum predicts protein secondary structure by using estimated chemical shifts [115]. As experimental methods are costly and proteins with known sequences continue to outnumber the proteins with experimentally identified secondary structures, developing computational approaches for protein secondary structure prediction becomes increasingly urgent.

Generally speaking, computational approaches for protein secondary structure prediction can be divided into 3 categories. The first category is either rule-based or statistic-based which can be dated back to 1970s. The rule-based methods uses empirical rules for predicting the initiation and the termination of helical and β regions in proteins. The empirical rules used are that when six successive residues have four helical formers or five successive residues have three β formers clustered together in any native protein segment, the nucleation of these secondary structures begins and propagates in both directions until terminated by sequence of breakers [42]. Later, statistical models are applied for segments of 9-21 amino acids. For example, the GGBSM method [55] uses amino acid segments to predict the structure of its central residue based on a well-known biological fact that amino acid compositions of different secondary structures are different. In GGBSM, the secondary structure is expressed as a function of the local amino acid composition by using three sets of parameters whose values are calculated from the 62 proteins of the Kabsch and Sander data bank. However, this category of methods use only sequence features to establish rules or to calculate probabilities for residue secondary structures, their performances ($<$ 60% of Q3 accuracy) is not good enough for use in any practical application.

Inspired by the successful use of volutionary features in DNA binding residue prediction [4] and Protein folding recognition [50], etc., the second category of methods uses evolutionary information of proteins from the same structural family [140] extracted by multiple-sequence alignment and position-specific scoring matrices (PSSM) [72] from PSI-BLAST for prediction. An evolutionary information based two-layered feed-forward neural network was developed on a non-redundant database of 130 protein chains to predict the secondary structure of water-soluble proteins, in which evolutionary information in the form of multi-sequence alignment is used as input instead of the use of single sequences [140]. The inclusion of evolutionary information in this method increases the prediction accuracy by 5%-10% of Q3 accuracy. This is because the method can make better use of protein family information for the target protein. Later, Hua and Sun proposed the first support vector machine(SVM) classifier for protein secondary structure prediction[65]. The input evolutionary information is in the form of multiple sequence alignment. Their work uses Q3 for classification by using three binary classifiers and assemble them into a tertiary classifier. The performance of this method reaches 76.2% on segment overlap accuracy (SOV).

Unbalanced data is a common problem for protein secondary structure prediction. Kim and Park [78] proposed a new protein secondary structure prediction method SVMpsi using an improved SVM with a jury decision system. The improved SVM can reduce the influence of noise and outliers by leveraging the theoretical relationships in the soft margin of an SVM. SVMpsi achieves the highest published Q3 accuracy and segment overlap accuracy (SOV) on two common datasets including both RS126 and CB513. Another SVM based method, called PMSVM, uses a dual-layer SVM and evolutionary information in the form of Position Specific Score Matrix(PSSM). [57]. In the dual-layer support vector machine, the first layer takes PSSM as input and outputs a feature matrix and the second layer takes the feature matrix as input. So, the first layer serves to learn feature representations for proteins and the second layer is for prediction.

Inspired by the successful application of deep learning models in natural language processing tasks [198, 205], the third category of methods uses different neural network models including convolutional neural networks (CNN) [86] and recurrent neural networks (RNN) [134, 11]. Many deep learning methods have been proposed for protein secondary structure prediction problems [185]. For example, Wang et al. proposed a Deep Convolutional Neural Fields (DeepCNF) method[185]. DeepCNF mainly contains two components including a deep neural network and a conditional random fields (CRF). In DeepCNF, the deep neural network component extract local context for target residues and the CRF component models dependencies among protein secondary structure of adjacent residues. DeepCNF achieves a high performance of 84% of Q3 accuracy and 72% of Q3 accuracy on the CASP dataset and the CAMEO dataset, respectively. The GSN method proposed by Zhou and Troyanskaya [209] uses a supervised generative stochastic network and convolutional architecture. Supervised generative stochastic network is a recently proposed deep learning technique [15], which can learns a Markov chain to sample from a conditional distribution. So GSN can automatically extract local context for target residues by a Markov chain in GSN. Although CNN can automatically extract local context for a target residue, it lacks the ability to extract long-range dependency.

To extract long-range dependency, several methods were proposed including bidirectional recurrent neural networks (BRNN) [134, 11] and probabilistic graphical models [146, 43]. For example, a novel deep convolutional and recurrent neural network (DCRNN) was proposed by Li and Yu [92] which can extract both local context and long-range dependences.

Even though theses deep learning based methods can extract both local context and long-range dependency, such as GSN and DCRNN, they need to combine two complex models to extract local context and long-range dependency, separately. Thus they are quite memory-intensive and time-consuming.

2.2 Overview of DNA binding residue prediction

The interactions between proteins and DNA are important for transcriptional regulation, which is essential to decipher the genetic pathways of various cellular processes [136]. As DNA binding residues are the basic elements in proteins-DNA interactions [101, 102, 103], the identification of DNA binding residues is important to understand the recognition mechanism between a protein and its DNA as well as the mechanism of transcriptional regulation. Due to the importance of DNA binding residues, many experimental methods were developed to identify DNA binding residues from protein sequences including electrophoretic mobility shift assays (EMSAs) [71, 72], nuclear magnetic resonance (NMR) spectroscopy [135], X-ray crystallography [128], peptide nucleic acid (PNA)-assisted identification of RNA binding proteins (RBPs) (PAIR) [127], MicroChIP [102], Fast ChIP [107], and conventional chromatin immunoprecipitation (ChIP) [80]. However, experimental methods are costly and labor-consuming. They are also not suited for analysis at the genome level. With the development of protein sequencing technology, more protein sequence data are now available. However, their 3D structures and biological functions are still mostly unknown. Development of computational methods for the prediction of DNA binding residues is now possible and can provide more insight to the understanding of interactions between proteins and DNA at the genome level.

Docking is a method which predicts the preferred orientation of one molecule to a second when bound to each other to form a stable complex. Knowledge of the preferred orientation in turn may be used to predict the strength of association or binding affinity between two molecules using, for example, scoring functions. Docking methods have been applied to ligand docking and protein-protein docking. Ligand docking [173] refers to cases where small molecule (“ligand”) is being docked into much larger macromolecule (”target”). Protein-protein docking [157] refers to predict the structure of protein–protein complexes using docking approaches. The ultimate goal of docking is the prediction of the three dimensional structure of the macromolecular complex of interest as it would occur in a living organism. Docking itself only produces plausible candidate structures. These candidates must be ranked using methods such as scoring functions to identify structures that are most likely to occur in nature. Although docking method can be used to identify ligand binding residues and protein-protein interaction residues, they have not been used to identify DNA binding residues. There may be two main reasons. The first one is target proteins need similar proteins which have available co-crystal structure with DNA to help inform of the interaction pocket. The other one is some DNA-binding proteins can bind DNA regardless of DNA sequence while others (for example transcription factors) bind specific DNA sequence. For many target proteins, we do not know their interacting DNA sequence before doing protein-DNA docking.

Many computational methods were proposed for the prediction of DNA binding residues. Commonly used features include three types: sequence features, evolutionary features, and structure features [180, 181, 182, 85]. In principle, both sequence features and evolutionary features can be extracted from protein sequence directly. However, earlier studies using evolutionary features have two limitations. Limited computer power makes it difficult to obtain evolutionary features of large protein sequences. Limited data in protein sequence databases also makes it hard to obtain high quality evolutionary features. Structure features need to be extracted from 3D structures of protein sequences which were unavailable until recently. Thus, earlier DNA binding residue predictions only use sequence features for prediction. Sequence features, as a very important type of features, include amino acid composition, predicted structure features and physiochemical properties. The Naïve Bayes classifier developed by Yan et al.[195] was trained from the identities of a target residue and its contextual residues. The SVM predictor proposed by Wang et al. [180] was based on sequence features including side chain pKa value, hydrophobicity index, and molecular mass.

As computing power is improved at a very rapid speed and dataset on protein sequences are progressively getting larger, high quality evolutionary features can be obtained now for most protein sequences. Thus, evolutionary features are used in many recent prediction methods. Originally, evolutionary features are used as a singular type of features for predictions such as the neural network classifier proposed by Ahmad et al. [4]. The classifier is trained by using Position Specific Score Matrix (PSSM), a commonly used representation for evolutionary features. Later works also make combined use of both sequence features and evolutionary features. For instance, the neural network classifier proposed by Ofran et al. [126] combines PSSM and sequence features including predicted secondary structure and predicted solvent accessibility. The used secondary structure and solvent accessibility are predicted by computational methods based on protein sequence , so they also belong to sequence features. The Random Forest classifier developed by Wang et al. combines PSSM with hydrophobicity index, side chain pKa value and molecular mass [182]. The Random Forest predictor DNABR proposed by Ma et al. [105], combines PSSM with six physicochemical properties which are different from that used in Wang et al.'s work [182]. For these kinds of features, the commonly used machine learning models include SVM, neural network and ensemble learning [129, 66, 148].

Technology advancement in recent years greatly helped the works for protein structure identification. 3D structures of many protein sequences are now available. Thus, using structure features in computational methods for DNA binding residue prediction is becoming feasible. Frequently used structure features include secondary structure, solvent accessible surface area, spatial neighbors, B-factor, protrusion index and depth index. Structure features are often used in machine learning models either as a singular type of features or are used in combination with sequence features and evolutionary features. For example, a SVM classifier developed by Bhardwaj et al. [20, 21] uses only structure features including Solvent accessibility, local composition, net charge, and electrostatic potentials. The neural network classifier developed by Ahmad et al. [3] uses the combination of sequence features and structure features. The neural network classifier DISPLAR proposed by Tjong et al. [166] combines evolutionary features and structure features, such as the 14 closest spatial neighbors and solvent accessibility. The SVM classifier developed by Kuznetsov et al. [85] incorporates evolutionary features, sequence features and structure features. SVM models [21, 3, 91, 98, 175] and neural network models [166] are the two commonly used machine learning models. Since machine learning based methods need intensive computing power and time to train, other works used non machine learning methods as alternatives for prediction. For example, Ozbek et al. [129] proposed the DNABINDPROT method by first selecting candidate residues based on the fluctuations of residues in high-frequency modes and then filtering selected residues with their evolutionary conservation profiles. Chen et al. [39] proposed DR_bind by first calculating geometry features, electrostatics features as well as conservation features and then selecting the three patches with the largest features as binding residues. However, relationships between residues, including pairwise relationships, relationships among multiple residues, are yet to be explored because the relationship information is naturally sparse.

2.3 Overview of TF binding site prediction

Gene expression is mainly regulated by interactions between DNAs and TFs [68]. The prediction of TF binding site (TFBS) is very important for understanding transcriptional regulatory networks and crucial in understanding fundamental cellular processes [52]. Two experimental techniques were developed for TFBS identification: chromatin immunoprecipitation followed by high throughput sequencing (ChIP-Seq) [67, 59, 79] and chromatin immunoprecipitation followed by array hybridization (ChIP-chip) [67, 139]. These technologies have been successfully used to map binding locations in many organisms. But, some properties of these techniques such as tissue and condition specificity, the availability of antibodies for TFs under study, and the expense of the experiments have made them useful only for a limited number of TFs. Therefore, high quality computational methods for TFBS prediction are urgently required for TFBS identification.

TFBSs are normally short and often diverse sequence motifs [36], which makes it computationally difficult to be modeled and predicted at the genomic scale. TFBS can be represented by a consensus sequence and a position weight matrix (PWM) [159, 160]. The consensus sequence representation makes it easy for visual interpretation of TFBS. But, variations in the compositions of nucleotide types in TFBS make consensus sequence an unsuitable approach for TFBS representation [88, 64]. So many classical computational methods use PWMs to represent TFBS [159, 160]. A PWM is often derived from a set of aligned and functionally related sequences. The basic assumption of PWM based methods is that each position within a TFBS participates independently in the corresponding DNA-protein interaction. However, position dependence within TFBS motifs is observed in many studies including crystal structure analyses [102] and biochemical studies [106]. Furthermore, quantitative analyses of protein binding microarray (PBM) data [120, 188, 208] have demonstrated that position dependence between neighboring positions is stronger than that between other positions. In order to incorporate position dependence into prediction, a new approach called a dinucleotide weight matrix (DWM) was proposed recently [149]. A DWM extends the basic PWM by considering the dependence between all pairs of neighbor positions within TFBS [149]. In addition to DWM, TFFM proposed by Mathelier and Wasserman can also capture position dependence for prediction [110] by using state transition probabilities in a hidden Markov model (HMM) [108] to model position dependence within TFBS.

Although DWMs [149] and TFFMs [110] can capture dependency between neighbor positions, they can only model dependency between two positions and cannot be used to capture dependency among multiple positions. Histone modification features are post-translational modification levels of histones on chromatin structure. Also, histone modification features covers multiple positions (at least 25 positions) and distinct DNA fragments may have different histone modification features. Thus, histone modification features by nature contain dependencies among multiple positions. Several studies [167, 83, 190] have shown that TF-DNA interactions are associated with various histone modification levels. According to these observations, several studies [167, 83, 190] have developed methods to improve the accuracy of TFBS predictions by incorporating histone modification features. Talebzadeh and Zare-Mirakabad [164] developed a method to make combined use of two types of histone modification features: the closet distance to a specific histone and the total number of specific histone modification. Won at al. [16] proposed a HMM based method called Chromia, in which both histone modification features and sequence features are used for feature representation learning. Recently, Tsai et al. [167] analyzed the contributions of sequence features, histone modification features, and DNA structure features to predictive models of TFBS by using a random forest model [31] and the conclusion is that all three feature types are useful for TFBS prediction. Recent studies have also suggested that DNA shape is another important type of features for TFBS prediction [111]. DNA shape represents the 3D structure of a DNA and a DNA shape feature covers indefinite number of positions. Thus by nature it contains dependences between multiple positions. Methelier [111] proposed a method by using DNA shape features and demonstrated that DNA shape features indeed play an important role in TFBS prediction. However, current works only contains first order dependency. Moreover, to predict the TFBSs of a TF for a specific cell-type, all current methods need training data from that specific cell-type. If a TF lacks training samples in a considered cell-type, current methods would not be able to predict the TFBSs of the TF for the considered cell-type.

2.4 Overview of gene expression prediction

Based on the three feature types including TFBSs, histone modification features and DHSs measured by DNase-seq data, multiple computational models were proposed for predicting gene expression. Beer and Tavazoie (2004) [14] proposed a systematic genome-wide probabilistic approach for learning complex combinatorial code underlying gene expression by using occupancy states of multiple TFBSs. The method can identify both positional and combinatorial constraints context-dependent gene expressions. The identified regulatory rules composed of positional and combinatorial constraints achieve 73% prediction accuracy for those genes in Saccharomyces cerevisiae. Ruan (2010) [142] later proposed a simple k-nearest-neighbor (KNN) method. Despite of the simplicity of KNN, this method works well in the third DREAM (Dialogue on Reverse Engineering Assessments and Methods project) Challenge on gene expression prediction [158] sharing the "top performers" with another team whose method is much more complicated [58]. To quantify the relationships between histone modification features and gene expression levels and understand interactions among different histone modifications features, both Karlic et al. (2010) [76] and Costa et al. (2011) [46] proposed linear regression method to build prediction model. Karlic et al. (2010) built a linear regression model from histone modification features and applied it to predict gene expression for human T-cell [184]. Result shows a high correlation level between their predictions and the observed gene expressions further proves that histone modification features indeed correlate with gene expression. Costa et al. (2011) proposed a model using the combination of two linear regression models to investigate the relative importance and effect of each histone modification feature for gene expression prediction. The conclusion is that H3K4me3 and H3K27me3 are positively and negatively correlated with gene expression, respectively.

Cheng at al. (2011) [40] attempted to reformulate gene expression prediction as a classification problem such that SVM can be used as the classifier. To incorporate position-specific histone modification features into a SVM model, the regions flanking transcription start site (TSS) and transcription termination site (TTS) are split into 160 bins of 100 bps and a SVM model is built for each bin, resulting in 160 SVM models. To quantify pairwise interactions between different histone modification features, they further proposed a linear regression model with binary combinatorial terms. They also investigated higher order interactions by applying a Bayesian network rather instead of a polynomial regression model. Dong at al. (2012) [51] built a Random Forest model from histone modification features to classify the level of expressions genes as either high or low. Result of the Random Forest model is inputted to a linear regression model to predict their gene expression level. The method uses the histone modification features in the bin with the closest relationship with gene expression to build its model for each bin. Four groups are formed for ll the histone modification features according to their functions to analyze the effects of pairwise interactions between groups on gene expression prediction.

Hu et al. (2015) [62] proposed a rule-learning method on the 20 most discriminative histone modification features in the T-cells datasets and learned 83 combinatorial rules to be used to discriminate high expression genes from low expression genes. The 83 rules provide good indication of the possible roles of histone modification combinations in gene regulation. Kumar et al. (2013) [84] proposed a linear estimation method called EFilter to predict gene expression from histone modification features imputed by ChromImpute, which was proposed by Ernst and Kellis (2015) [53] to impute histone modification signals for a new sample using an ensemble of regression trees.

With the deep learning technologies becoming more successful in many areas of research, many studies in the bioinformatics community have started to use deep learning methods to learn meaningful and hierarchical representation for DNA and proteins. Singh et al. (2016) [153] developed a gene expression prediction model, called DeepChrome, by applying a deep convolutional neural network [87]. In DeepChrome, only the regions flanking transcription start site (TSS) are divided into 100 bp bins and the histone modification signals in all bins are concatenated as input to the method. In contrast to aforementioned methods, DeepChrome can automatically extract complex interactions among different histone modification features and it also allows to visualize the combinatorial interactions among histone modification features by feature pattern maps from the learnt deep model.

Even though many methods use histone modification features. TFBS features and DHS features are much less explored. Natarajan at al. (2012) [122] proposed a method to predict cell-type-specific gene expressions from DHS information obtained by DNase-seq. The result shows that the DHS information for the promoters of three types of genes are substantially different. The gene types include cell-type-specific up-regulated, down-regulated, and constitutively expressed genes. The major limitation of this method is that it can extract DHS information only from DNase-seq data, which limit its application to only those cell-types with DNase-seq data. Schmidt et al. (2017) [147] proposed a novel segmentation-based method, called TEPIC, to predict gene expressions. TEPIC requires two steps in its process. The first step predict TF binding strength by combing position weight matrices and open-chromatin regions (OCRs). OCRs are defined as the DNA regions accessible by proteins such as TFs. In the second step, the TF binding strength of the regions flanking TSS are used to predict gene expression levels using a log regression model. In TEPIC, both DNase-seq and NOMe-seq data can be used to measure open-chromatin regions, so TEPIC has wider applications. The results on several cell-types demonstrated that the open-chromatin signal indeed can improve gene expression prediction further.

Recently, Zhang and Li (2017) [204] proposed a statistical model to estimate the effects of TFBSs and histone modification features on gene expressions by using a support vector regression(SVR) model. In this method, genome-wide TFBSs of 15 TFs, 10 histone modification features and DHSs are used as input for the SVR model to predict gene expression for three cell lines: H1-HESc, Gm12878 and K562, respectively. The study on combinatorial interactions among TFBSs, histone modification features concludes that even though TFBSs and histone modification features have some redundancies in the prediction, likely due to regulation mechanism, their effects on the predictive abilities of the SVR model for prediction are different. Currently, only a limited number of TFs have identified TFBSs using the aforesaid methods. even in common cell-types, current methods can only incorporate the TFBSs of a limited number of TFs.

2.5 Deep learning models

Convolutional Neural Network (CNN) and Long Short-Term Memory Network (LSTM) are two representative methods in deep learning methods and can learn information from sequence data. Thus they are suited for learning local context as well as long-range dependences. As many of the state-of-the-art algorithms use these two methods, more details information is given here.

2.5.1 Convolutional Neural Network

A convolutional Neural Network (CNN) is a type of feed-forward Artificial Neural Network. The individual neurons in a CNN are arranged in such a way that they respond to overlapping regions tiling the visual field. CNNs are applied in many fields including image and video recognition [45], recommender systems [168] and natural language processing. In recent years, CNN has gradually been introduce into bioinformatics to learn protein sequence representation for predictions of protein structures and functions. For example, Alipanahi et al. [5] developed DeepBind for the prediction of sequence specificities of DNA- and RNA-binding proteins. Wang et al. [183] proposed a CNN based method for protein secondary structure prediction.

A CNN comprises four computational layers: a convolution layer, a rectification layer, a pooling layer and a neural network layer. The first three layers can discover important motifs of input instances and the last layer is used to obtain prediction results. The convolution layer, the rectification layer, and the network layer have trainable motif detectors D, thresholds b, and weights W, respectively. For an instance S, a CNN produces a real-valued score $f(S)$ according to the following formula

$$f(S) = net_W(pool(rect_b(conv_D(S)))) \tag{2.1}$$

where $convD()$, $rectb()$, $pool()$ and $netw()$ denote the four layers in CNN. This real-valued score is used for prediction.

Convolution layer: In the convolution layer, several filters, called motif detectors, are used to convolve the raw input. For instance, the convolution of a motif detector can play the same role as a "motif scan" operation in a PWM or a PSAM-based model. For a motif detector of size m, the instance S should be padded by concatenating $(m - 1)$ unuseful residues on either sides. The padded sequence of S is represented as a matrix M defined as follows:

$$\begin{cases} 0.05 & \text{if } i = m \; or \; i > n - m \\ 1 & \text{if } S_{i-m+1} = j^{th} \; base \\ 0 & otherwise \end{cases} \tag{2.2}$$

where n is the length of the input instance. The output of the convolution layer is a matrix X in which an element $X_{i,k}$ is essentially the score of motif detector k aligned to position i of the padded sequence M. Given that the motif detectors are represented as an array D, where element $D_{k,j,l}$ is the coefficient of motif detector k at motif position j and base l, the element $X_{i,k}$ of the output is calculated by following formula

$$X_{i,k} = \sum_{j=1}^{m} \sum_{l=1}^{20} M_{i+j,l} D_{k,j,l} \tag{2.3}$$

So, the column $X_{.,k}$ is the motif scan of motif detector k applied to the padded sequence M and row $X_{i,.}$ is the motif scan of all the motif detectors on position i of the padded sequence M.

Rectification layer: The rectification layer plays an important role for CNNs as it is used to filter unimportant motif features. Its takes the output from the convolution layer. Its output $Y = rect_b(X)$ is an matrix of the same size as that of X defined as follows:

$$Y_{i,k} = Max(0, X_{i,k} - b_k) \tag{2.4}$$

where b_k is the activation threshold for motif detector k, which is learned in the training process of CNN. The formula means that if score $X_{i,k}eb_k$, the relative score of motif detector k at position i is passed to the next stage; Otherwise motif detector k is deemed irrelevant at position i and so the relative score is zero. Only those motif features with scores larger than a specified threshold will pass through this layer.

Pooling layer: The output Z of the pooling layer is a feature vector, whose dimension depends on the number of motif detectors in the convolution layer. The pooling layer for motif detector $k(1 \leqslant k \leqslant d)$ is formulated as

$$Z_k = Max(Y_{1,k}, \cdots, Y_{n,k}) \tag{2.5}$$

For every instance, we can obtain a vector Z with dimension of d, where d is the number motifs used in the Convolution layer. The features contained in vector Z are motif features captured by d motif detectors in the convolution layer.

Neural network layer: The neural network layer is used for prediction. In order to avoid overfitting, a recently proposed dropout technique are now commonly used before the hidden layer in the neural network layer. For example, DeepChrome proposed by Singh et al.(2016) contains a drop with a chosen probability of 0.5 between the pooling layer and the neural network layer [153]. With the dropout technique, the entries of hidden representations are set to 0 with a dropout rate.

CNN can extract relationships in multiple positions, but only local context in short distance. In proteins and DNA, the long-range dependencies between positions also play an very important role for function and structure prediction. The Long Short-Term Memory Network, on the other hand, is more suited to, extract long-range dependencies.

2.5.2 Long Short-Term Memory Network

Long Short-Term Memory Network (LSTM) [63] is a extended form of recurrent neural networks (RNNs) in deep learning. LSTM have been successfully applied to handwriting recognition [56], speech recognition [143] and a wide range of NLP tasks such as machine translation [161], constituency parsing [175], language modeling [203] and natural language inference including rule-based systems [26]. Simply put, LSTMs use several gate vectors at each position to control the passing of information along a sequence and this helps to model long-range dependences.

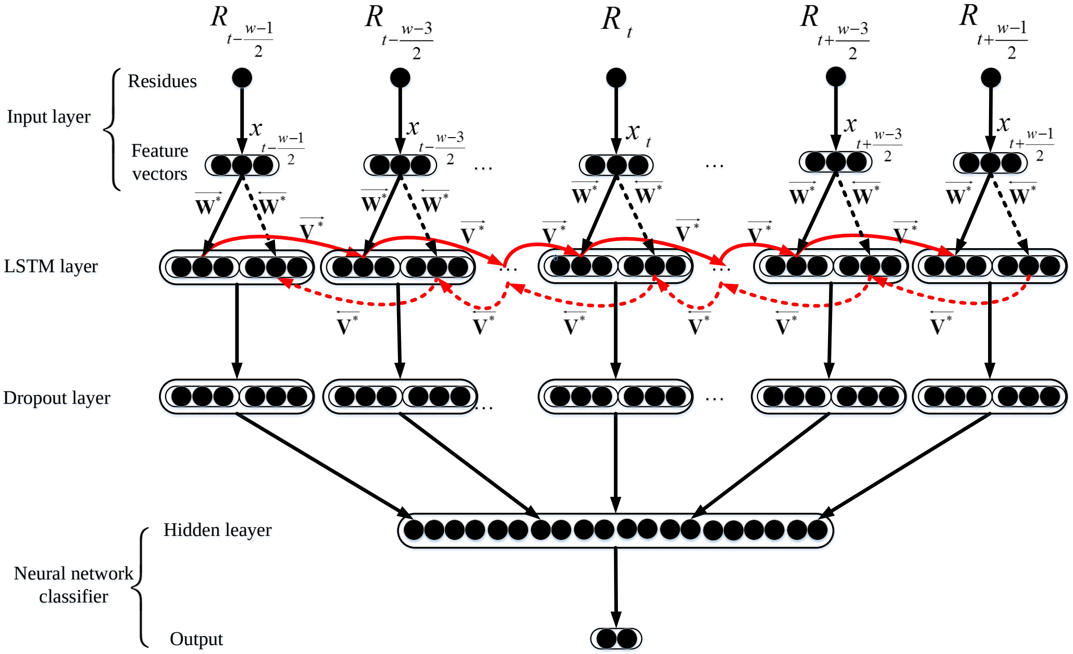


Figure 2.1: The framework diagram of LSTM.

Figure 2.1 shows the framework of a LSTM model with four layers: the input layer, the LSTM layer, the dropout layer and the final neural network classifier.

The input layer: the elements of each instance in this layer are encoded by different features. Given an instance S_t, the input layer generates its representation as a feature vector sequence

$$X_t = x_{t-\frac{w-1}{2}}, x_{t-\frac{w-3}{2}}, \cdots, x_t, \cdots, x_{t+\frac{w-3}{2}}, x_{t+\frac{w-1}{2}} \tag{2.6}$$

where $\mathbf{x}_k \in \mathbb{R}^d (t-(w-1)/2 \leqslant k \leqslant t+(w-1)/2)$ is a feature vector of residue in position k. Each feature vector is a representation of the different features of S_t. The dimension size of the feature vector is denoted by d.

The LSTM layer: This layer is often referred to as the hidden layer in neural networks because it learns information not explicitly coded from the input layer. If the left context and right context for a node in different instances are not symmetric, useful features for a node in the left context and the right context need to be extracted separately. Thus, in the LSTM layer, two separate representations are built for a node k. The forward representation, denoted by $\overrightarrow{\mathbf{h}_k}$, encodes pairwise relationships between node k and its left neighbor. The backward representation, denoted by $\overleftarrow{\mathbf{h}_k}$, encodes pairwise relationships between node k and its right neighbor. In the representation of the node in either direction, a node k has five internal vectors. To avoid duplication, let use explain this using forward direction only. The five internal vectors are (1) an input gate $\overrightarrow{\mathbf{i}_k}$, (2) a forget gate $\overrightarrow{\mathbf{f}_k}$, (3) an output gate $\overrightarrow{\mathbf{o}_k}$, (4) a candidate memory cell $\overrightarrow{\tilde{\mathbf{c}}_k}$, and (5) a memory cell $\overrightarrow{\mathbf{c}_k}$. $\overrightarrow{\mathbf{i}_k}$,$\overrightarrow{\mathbf{f}_k}$ and $\overrightarrow{\mathbf{o}_k}$ are used to indicate which values should be updated, forget or kept in the LSTM model. $\overrightarrow{\tilde{\mathbf{c}}_k}$ and $\overrightarrow{\mathbf{c}_k}$ are used to keep the candidate features and the accepted features, respectively. Two sets of weight matrices $\overrightarrow{\mathbf{W}^*}(\overrightarrow{\mathbf{W}^\mathrm{i}}, \overrightarrow{\mathbf{W}^\mathrm{f}}, \overrightarrow{\mathbf{W}^\mathrm{o}}, \overrightarrow{\mathbf{W}^\mathrm{c}})$ and $\overrightarrow{\mathbf{V}^*}(\overrightarrow{\mathbf{V}^\mathrm{i}}, \overrightarrow{\mathbf{V}^\mathrm{f}}, \overrightarrow{\mathbf{V}^\mathrm{o}}, \overrightarrow{\mathbf{V}^\mathrm{c}})$ are used to connect input feature vectors and the left neighbor to the current node k, respectively. These weight matrices are learned in the training process of a LSTM model. $\overrightarrow{\mathbf{i}_k}$,$\overrightarrow{\mathbf{f}_k}$, $\overrightarrow{\mathbf{o}_k}$ and $\overrightarrow{\tilde{\mathbf{c}}_k}$ are computed from the input feature vector and the forward representation of its left neighbor. Formally, they are computed by the following formulae:

$$\overrightarrow{\mathbf{i}_k} = \sigma(\overrightarrow{\mathbf{W}^{\mathrm{i}}} * \mathbf{x}_k + \overrightarrow{\mathbf{V}^{\mathrm{i}}} * \overrightarrow{\mathbf{h}_{k-1}} + \overrightarrow{\mathbf{b}^{\mathrm{i}}}) \tag{2.7}$$

$$\overrightarrow{\mathbf{f}_k} = \sigma(\overrightarrow{\mathbf{W}^{\mathrm{f}}} * \mathbf{x}_k + \overrightarrow{\mathbf{V}^{\mathrm{f}}} * \overrightarrow{\mathbf{h}_{k-1}} + \overrightarrow{\mathbf{b}^{\mathrm{f}}}) \tag{2.8}$$

$$\overrightarrow{\mathbf{o}_k} = \sigma(\overrightarrow{\mathbf{W}^{\mathrm{o}}} * \mathbf{x}_k + \overrightarrow{\mathbf{V}^{\mathrm{o}}} * \overrightarrow{\mathbf{h}_{k-1}} + \overrightarrow{\mathbf{b}^{\mathrm{o}}}) \tag{2.9}$$

$$\widetilde{\overrightarrow{\mathbf{c}_k}} = \sigma(\overrightarrow{\mathbf{W}^{\mathrm{c}}} * \mathbf{x}_k + \overrightarrow{\mathbf{V}^{\mathrm{c}}} * \overrightarrow{\mathbf{h}_{k-1}} + \overrightarrow{\mathbf{b}^{\mathrm{c}}}) \tag{2.10}$$

where is σ the sigmoid function and $\overrightarrow{\mathbf{b}^{*}}(\overrightarrow{\mathbf{b}^{\mathrm{i}}}, \overrightarrow{\mathbf{b}^{\mathrm{f}}}, \overrightarrow{\mathbf{b}^{\mathrm{o}}}, \overrightarrow{\mathbf{b}^{\mathrm{c}}})$ are the bias vectors to be learned. The memory cell $\overrightarrow{\mathbf{c}_k}$ is the sum of the filtered left memory cell $\overrightarrow{\mathbf{c}_{k-1}}$ by the forget gate $\overrightarrow{\mathbf{f}_k}$, and the updated candidate memory cell $\widetilde{\overrightarrow{\mathbf{c}_k}}$ of the current node by the input gate $\overrightarrow{\mathbf{i}_k}$ as defined below:

$$\overrightarrow{\mathbf{c}_k} = \overrightarrow{\mathbf{f}_k} \odot \overrightarrow{\mathbf{c}_{k-1}} + \overrightarrow{\mathbf{i}_k} \odot \widetilde{\overrightarrow{\mathbf{c}_k}} \tag{2.11}$$

where $\odot$ is the element-wise multiplication of the two vectors. Finally, the forward representation $\overrightarrow{\mathbf{h}_k}$ is calculated by multiplying memory cell $\overrightarrow{\mathbf{c}_k}$ with the output gate $\overrightarrow{\mathbf{o}_k}$.

$$\overrightarrow{\mathbf{h}_k} = \overrightarrow{\mathbf{o}_k} \odot tanh(\overrightarrow{\mathbf{c}_k}) \tag{2.12}$$

Similarly, backward representation, the 5 gates can be defined by the following formulae:

$$\overleftarrow{\mathbf{i}_k} = \sigma(\overleftarrow{\mathbf{W}^{\mathrm{i}}} * \mathbf{x}_k + \overleftarrow{\mathbf{V}^{\mathrm{i}}} * \overleftarrow{\mathbf{h}_{k-1}} + \overleftarrow{\mathbf{b}^{\mathrm{i}}}) \tag{2.13}$$

$$\overleftarrow{\mathbf{f}_k} = \sigma(\overleftarrow{\mathbf{W}^{\mathrm{f}}} * \mathbf{x}_k + \overleftarrow{\mathbf{V}^{\mathrm{f}}} * \overleftarrow{\mathbf{h}_{k-1}} + \overleftarrow{\mathbf{b}^{\mathrm{f}}}) \tag{2.14}$$

$$\overleftarrow{\mathbf{o}_k} = \sigma(\overleftarrow{\mathbf{W}^{\mathrm{o}}} * \mathbf{x}_k + \overleftarrow{\mathbf{V}^{\mathrm{o}}} * \overleftarrow{\mathbf{h}_{k-1}} + \overleftarrow{\mathbf{b}^{\mathrm{o}}}) \tag{2.15}$$

$$\widetilde{\overleftarrow{\mathbf{c}_k}} = \sigma(\overleftarrow{\mathbf{W}^{\mathrm{c}}} * \mathbf{x}_k + \overleftarrow{\mathbf{V}^{\mathrm{c}}} * \overleftarrow{\mathbf{h}_{k-1}} + \overleftarrow{\mathbf{b}^{\mathrm{c}}}) \tag{2.16}$$

$$\overleftarrow{\mathbf{c}_k} = \overleftarrow{\mathbf{f}_k} \odot \overleftarrow{\mathbf{c}_{k-1}} + \overleftarrow{\mathbf{i}_k} \odot \widetilde{\overleftarrow{\mathbf{c}_k}} \tag{2.17}$$

$$\overleftarrow{\mathbf{h}_k} = \overleftarrow{\mathbf{o}_k} \odot tanh(\overleftarrow{\mathbf{c}_k}) \tag{2.18}$$

Note that the learned feature vectors for every residue includes the forward representation and the backward representation symmetrically.

Dropout Layer: This layer is introduced to avoid overfitting in the learning process. In this layer, the entries of the representations for every node are set to 0 with a dropout rate, tuned based on development data. After processed by the dropout layer, feature vectors of all the nodes are combined to form the final feature vector representation. This final feature vector representation is then used by the neural network classifier for prediction. The neural network classifier includes a hidden layer and an output layer where the hidden layer is used to reduce the dimensionality of the feature space and the output layer is used for prediction.

2.6 Chapter Summary

In this chapter, the related background knowledge for the four problems is introduced. For protein secondary structure prediction, current methods cannot extract local context and long-range dependency efficiently. For DNA binding residue prediction, existing methods cannot capture the relationships between residues including relationships between residues with short distance and long distance. For TF binding site prediction, existing methods cannot extract higher order dependencies between positions and also cannot be applied to predict the TFBSs of TFs for cell-types without training data. For gene expression prediction, existing methods either did not use TFBSs of TFs or can only be applied for cell-types with TFBSs for most TFs. The limitations of research works in the above areas motivate the works in this thesis. The technology development in hardware and software and the availability of additional datasets makes solutions using deep learning models possible. Thus, in the following chapters, novel methods will be proposed to overcome these limitations.

Chapter 3

Protein secondary structure prediction

In bioinformatics and theoretical chemistry, protein secondary structure prediction is very important in medicine development and biotechnology, for example in drug design[124] and in the design of novel enzymes. Since secondary structure can be used to find distant related proteins with unalignable primary structures, incorporating both secondary structure information and simple sequence information can improve the accuracy of protein sequences alignment[151]. Protein secondary structure prediction also plays a important role in protein tertiary structure prediction. As protein secondary structure can determine the structure types of protein local fragments, the freedom degree of protein local fragments in the tertiary structure can be reduced. Therefore accurate secondary structure prediction can potentially improve the accuracy of protein tertiary structure prediction[209, 185, 196].

As presented in Chapter 2, the first category and the second category of approaches cannot extract local context and long-range dependency for prediction. Although some works in the 3rd category can extract both local context and long-range dependency using deep learning methods, they need to combine two models to extract local context and long-range dependency, separately. For example, GSN [209] needs to use CNN to extract local context and a supervised generative stochastic network to extract long-range dependency. DCRNN [92] needs to use CNN to extract local context and RNN to extract long-range dependency. As both supervised generative stochastic networks and RNN are very complex models, these methods are very computationally intensive and thus need to use a lot of computing power.

In this chapter, we proposed a novel method, referred to as CNNH_PSS, to extract both local context and long-range dependency. The principle idea of CNNH_PSS is to use a multi-scale CNN with a highway structure to pass some information learned from the previous layer. The highway can bypass the current layer to go into the next layer without any change. More specifically, a mulit-scale CNN uses several sets of kernels with different lengths to extract relationships of different lengths. For example, relationships between two residues has a length of 2 and relationship among 3 residues has a length of 3,etc.. In CNNH_PSS, every layer in the network have a bypass, commonly referred to as a highway, between the input and output. For each layer, a portion of information is delivered from the input to the output directly by the highway while the other portion of the information is processed by the current layer to extract relationships between more remote residues. A weight is learned for each layer to control how much information is retained by the highway and how much information is processed by the kernels in the layer. By means of the additional highway structure in each layer, local context extracted by a layer can passes through directly to upper layers so that CNNH_PSS can learn long-range dependency by higher layers while retaining local context extracted by low layers. Finally, the extracted local context and long-range dependency are inputted to a Multilayer perceptron (MLP) containing a fully connected layer and a softmax layer for prediction.

3.1 The CNNH_PSS Method

As shown by many recently published works[193, 194], a complete prediction model in bioinformatics should contain the following four components: validation benchmark dataset(s), an effective feature extraction procedure, an efficient predicting algorithm and a set of fair evaluation criteria. In this section, we describe each of these four components of CNNH_PSS in details.

3.1.1 Datasets

Two publicly available datasets: CB6133 and CB513 are used to evaluate the performance of our proposed method CNNH_PSS in comparison to other state-of-the-art methods. The state-of-the-art methods on CB6133 include GSN [209] and DCRNN [92]. The state-of-the-art methods on CB513 include SSpro8 [134], CNF [185] and DeepCNF [183].

CB6133: CB6133 was produced by PISCES CullPDB[179] and it is a large non-homologous protein dataset with known secondary structures for every protein. It contains 6,128 proteins, in which 5,600 proteins are training samples. 256 proteins are validation samples and 272 proteins are testing samples. This dataset is publicly available [92].

CB513: CB513 is a testing dataset and can be freely obtained [209, 47]. When CB513 is used for testing, CB6133 is the training dataset. As there are some redundancy between CB513 and CB6133, CB6133 is filtered by removing all sequences having over 25% sequence similarity with sequences in CB513. After filtering, 5,534 cleaned proteins in CB6133 are used as training samples. Since CB513 is used by most of the state-of-the-art methods including GSN [209] and DCRNN [92] as well as other methods [183, 185], CB513 serves as the benchmark dataset to make fair comparison to other methods.

3.1.2 Feature representation

Given a protein with L residues as $X = x_1, x_2, x_3, \cdots, x_L$, where $x_i \in \mathbb{R}^m$ is a m-dimensional feature vector of the i^{th} residue, the secondary structure prediction for this protein is formulated as determining $S = s_1, s_2, s_3, \cdots, s_L$ for X where s_i is an Q8 secondary structure label. In this study, x_i is encoded by both sequence features and evolutionary information. Sequence features are used to specify the category of the target residue. Two methods are used to encode sequence features. The first one is **one hot vector** representation and the second one is **residue embedding**. One hot representation encodes sequence features of each residue by a 21-dimension one-hot vector, in which only one element equals to 1 and the remaining elements are set to 0, where every one-hot vector corresponds a residue type. However, one-hot vector is a sparse representation and unsuitable for measuring relations between different residues. In order to get a dense representation of sequence features, an embedding technique in natural language processing is used to transform 21-dimensional one-hot vector to a 21-dimensional dense representation[117]. The transformation is implemented by a feed forward neural network layer before the multi-scale CNN in CNNH_PSS.

Evolutionary information such as position-specific scoring matrix (PSSM) is considered as informative features for predicting secondary structure in previous works[72]. PSSM is a common representation for evolutionary information and has been used in many bioinformatics studies including protein functionality annotation and protein structure prediction [82, 70, 23, 174, 207]. In this study, PSSM is calculated by PSI-BLAST[144] against the UniRef90 database with E-value threshold 0.001 and 3 iterations. For a protein with length L, PSSM is usually represented as a matrix with $L \times 21$ dimensions where 21 denotes the 20 standard types of residues and one extra residue type which represents all non-standard residue types. Before PSSMs are used as inputs for CNNH_PSS, they need to be transformed to 0-1 range by the sigmoid function. By concatenating sequence features and evolutional information, each residue in protein sequences can be encoded by a feature vector with dimension of 42.

3.1.3 Multi-scale CNN with highway

In this work, the proposed CNNH_PSS uses a multi-scale CNN with highway to extract both local context and long-range dependency. Then, the extracted local context and long-range dependencies are inputted to a fully connected softmax layer for prediction. The framework of one layer in a CNNH_PSS node is shown in Figure 3.1. Figure 3.1 shows that each layer in CNNH_PSS contains three parts: the input section, the

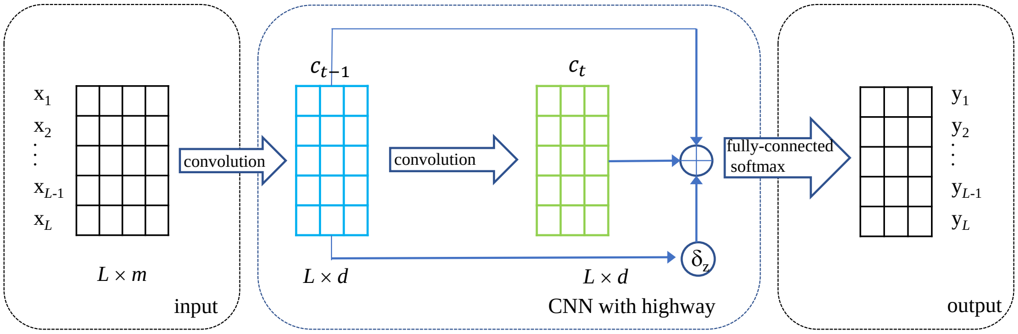


Figure 3.1: The framework of a single layer in CNNH_PSS

CNNH_PSS section, and the output section. In the input section, $x_i \in \mathbb{R}^m$ denotes the feature vector of i^{th} residue in protein, which is the concatenation of sequence features and evolutional information. Thus a protein of length L is encoded as a $L \times m$ matrix $x_{1:L} = [x_1, x_2, x_3, \cdots, x_L]^{\mathrm{T}}$, where L and m denote the length of proteins and the number of features used to encode residues, respectively. In this study, as both sequence features and evolutional information have 21 dimensions, m is thus 42. As the the output of a layer and the information bypassed by the highway need to be summed up by element-wise manner to get the final output of the layer, so the input and output of all layers must have the same sample length. In order to keep the output of this layer to have the same length as that of the input, we need to pad $\lfloor h/2 \rfloor$ and $\lfloor (h-1)/2 \rfloor$ m-dimensional zero vectors to the respective head and the tail of the input $x_{1:L}$, where h is the length of the kernel in this layer. The CNNH_PSS section contains two parts: the multi-scale CNN and the highway. Each multi-scale CNN contains n layers. In the layer $(t-1)^{th}$, the convolution operation of the k^{th} kernel $w_k^{t-1} \in \mathbb{R}^{h \times m}$ executed on DNA fragment $x_{i:i+h-1}$ is expressed as

$$c_{k,i}^{t-1} = f^{t-1}(w_k^{t-1} \cdot x_{i:i+h-1} + b_k^{t-1}) \tag{3.1}$$

where h is the length of the kernel, b_k^{t-1} is the bias of the k^{th} kernel, f is the activation function and $x_{i:i+h-1}$ denotes the DNA fragment $x_i, x_{i+1}, x_{i+2}, \cdots, x_{i+h-1}$. Through executing convolution operation of the k^{th} kernel on the padded input, we get a novel feature vector

$$c_k^{t-1} = [c_{k,1}^{t-1}, c_{k,2}^{t-1}, c_{k,3}^{t-1}, \cdots, c_{k,L}^{t-1}]^{\mathrm{T}} \tag{3.2}$$

Suppose we have d kernels in the layer, thus we can get d novel features vectors as formula (2). By concatenating d novel feature vectors, we can get a novel feature matrix with dimension $L \times d$

$$c^{t-1} = [c_1^{t-1}, c_2^{t-1}, c_3^{t-1}, \cdots, c_d^{t-1}] \tag{3.3}$$

This novel feature matrix is used as the input of the next layer. If there are n layers and θ_t is used to denote the kernels and the bias of the t^{th} layer, the output of the n^{th} layer is

$$c^n = f_{\theta_n}^n(f_{\theta_{n-1}}^{n-1}(\cdots f_{\theta_1}^1(x_{1:L}))) \tag{3.4}$$

Finally, the output of the n^{th} layer is used as the input of the fully connected softmax layer for prediction

$$y_i = argmax(w \cdot c^n + b) \tag{3.5}$$

where w and b is the weight and bias of the fully connected softmax layer, respectively. y_i is the predicted secondary structure for the i^{th}residue in the target protein.

CNN has achieved huge progress in many tasks of image processing filed, one common sense is that the success of CNN is attributed to multiple layers in CNN because CNN with more number of layers can extract dependencies between more remote positions. However, with the augment of number of layers in CNN, the information communication between layers will become more difficult and the gradient will disappear[156]. Srivastava et al. [156] has proposed highway network to resolve these problems. So in CNNH_PSS, highway network and multi-scale CNN are incorporated to predict secondary structures.

In CNNH_PSS, each layer has a highway (shown in Figure 3.1). For each layer, the highway is used to deliver a portion information from the input to the output directly to retain the relationships contained by the previous layer and the kernels in the current layer are used to process the other portion of information to extract relationships between more remote residues. Each highway has a weight z_t to determine how much information is delivered by the highway and how much information are used to extract relationships between remote residues in the current layer. The weight z_t is calculated by the weighted sum of the values in the input c^{t-1} of current layer.

$$z_t = \delta(w_z^t c^{t-1}) \tag{3.6}$$

where $\delta(\cdot)$ is sigmoid function and w_z^t is a weight. The output of the current layer is the weighted sum of the information delivered by the highway and the output of the kernels of the current layer.

$$c^t = (1 - z_t) \times f_{\theta_t}^t(c^{t-1}) + z_t \times c^{t-1} \tag{3.7}$$

where $f_{\theta_t}^t$ is the convolution operation of the current layer. So the output of the t^{th} layer contains two portions: information from the highway and that outputted by the convolution kernels of the current layer.

Table 3.1: Hyper-parameters of multi-scale CNN

Hyper-parameter	Value
Kernel length	[7,9,11]
Number of kernels	80 for each kernel length
Batch size	50
Learning rate	2e-3
Regularizer	5e-5
Decay rate	0.05
Activation function	ReLU

3.2 Performance Evaluation

The purpose of the evaluation is to examine the effectiveness of our proposed CNNH_PSS. Four sets of evaluations are conducted. The first experiment evaluates the performance of multi-scale CNN on CB6133 and CB513. The second experiment evaluates our proposed CNNH_PSS on CB6133 and CB513. The third experiment compares CNNH_PSS with state-of-the-art methods. Finally, based on CB6133, we analyze the local context and long-range dependencies learned by CNNH_PSS. The performance of prediction methods are measured by Q8 accuracy[209, 92]. For multi-scale CNN and CNNH_PSS, the hyper-parameters of multi-scale CNN in this study are listed in Table 3.1.

3.2.1 The performance of multi-scale CNN model

Note that three set of kernels with different lengths are used in the multi-scale CNN model and 80 kernels are used for each kernel length. To conveniently encode and process protein sequences, we normalize the length of all protein sequences to 700 according to literature [209]. When sequences are shorter than 700, they are padded with zero vectors. If sequences are longer than 700, the additional part is truncated. In order to get the best performance, we need to determine how many layers the multi-scale CNN model should contain. We conduct experiments to evaluate the performance of the multi-scale CNNs model with different number of layers on CB513. The performance is shown in Figure

3.2, where the x-axis is the number of epochs used to train the multi-scale CNN model and the y-axis is validation accuracy. Figure 3.2 shows the performances for the models

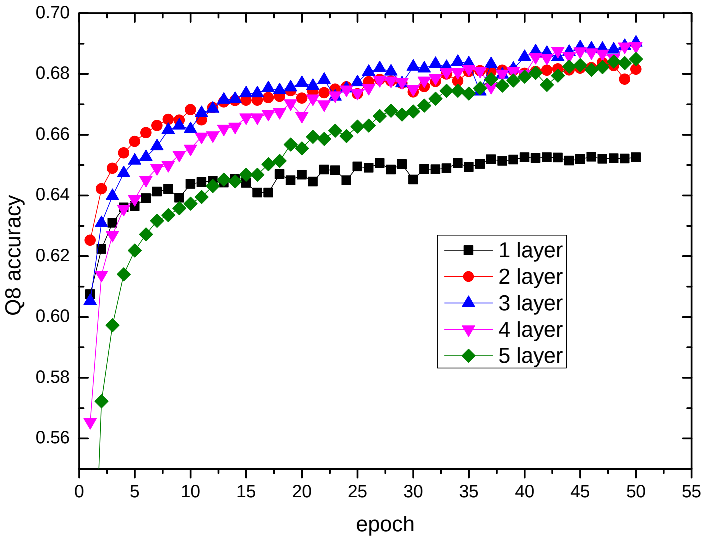


Figure 3.2: The performance of multi-scale CNN with different number of convolutional layers

with the number of layers from 1 to 5. This figure shows that the model with 3 layers gets the best accuracy. When the number of layers is increased to 4 or 5, accuracy decreases obviously. The main reason for this phenomenon may be that extracted local context will lost when more layers are added in CNN. We know that with the augment of number of layers in CNN, CNN can extract relationships between more remote residues, but most local context may be lost. When the number of layers is increased to 3, CNN may extract both local context and long-range dependencies, which is validated by that the CNN with 3 layers gets the best accuracy in our problem. However, when the number of convolution layers is more than 3, most local context extracted by lower layers are used to learn dependencies between more remote residues by higher layers so that relationships outputted by higher layers may contains less local context. Thus, the predicting accuracy starts to decrease when the number of layers is more than 3.

The performances of the multi-scale CNN with 3 layers on CB6133 and CB513 are

Table 3.2: Q8 accuracy of the multi-scale CNN with 3 layers

datasets	CB6133	CB513
Multi-scale CNN(one hot)	0.721	0.689
Multi-scale CNN(embedding)	**0.729**	**0.693**

shown in Table 3.2, where the best performers and the second best performers are marked by bold and underscore, respectively. Two sequence features encoding methods for residues are evaluated: one hot and residue embedding.

Table 3.2 shows that residue embedding outperforms one hot on both CB6133 and CB513 by at least 0.004 Q8 accuracy, indicating that residue embedding is a better encoding method for sequence features. In the next section, we use residue embedding method to encode sequence features in both the multi-scale CNN model and our proposed method CNNH_PSS.

3.2.2 The performance of CNNH_PSS

We evaluate the performance of our proposed method CNNH_PSSs with different number of layers on CB513. The performance is shown in Figure 3.3.

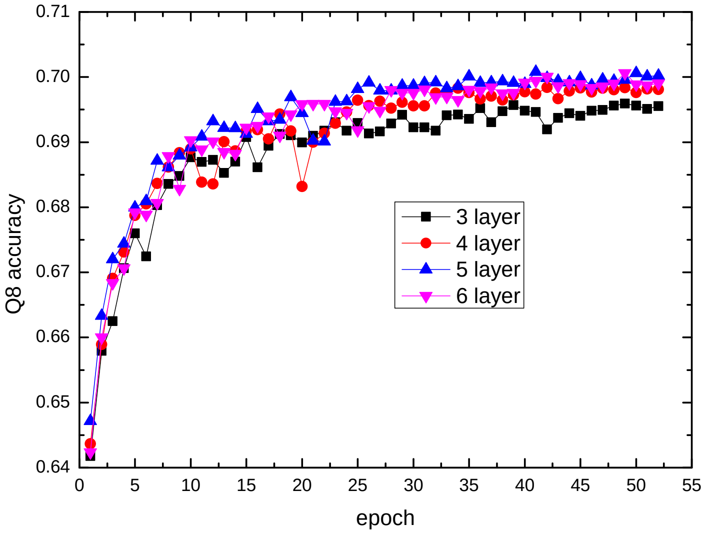


Figure 3.3: The performance of CNNH_PSS with different number of layers

Table 3.3: Q8 accuracy of CNNH_PSS with 5 convolutional layers

Method	CB6133	CB513
Dilated Residual Network	0.710	0.670
Multi-scale CNN	<u>0.729</u>	<u>0.693</u>
CNNH_PSS	**0.740**	**0.703**

Figure 3.3 shows that CNNH_PSS achieves the best performance when the number of layers is 5. When the number of layers is more than 5, the performance of our method starts to decrease. So CNNH_PSS with 5 layers is used in the rest of the evaluation. Comparing with the 3-layer multi-scale CNN model in the previous section CNNH_PSS not only contains a highway for every layer, but also has more number of layers. It means that CNNH_PSS not only can retain more local context, but also extract more long-range dependencies between more remote residues. The performances of CNNH_PSS and the multi-scale CNN model on CB6133 and CB513 are shown in Table 3.3, where the best performers and the second best performers are marked by bold and underscore, respectively. Table 3.3 shows that CNNH_PSS outperforms the multi-scale CNN model by 0.011 Q8 accuracy on CB6133 and 0.010 Q8 accuracy on CB513. The performance improvement by CNNH_PSS in both datasets validates that the highways in CNN indeed can retain both local context as well as extract long-range dependencies between more remote residues.

Dilated Residual Network [201] is a combined method of Deep Residue Network [74] and Dilated Convolution [200], which can increase the resolution of output feature maps without reducing the receptive field of individual neurons. This means that Dilated Residual Network not only can learn long-range dependencies, but also can keep the resolution of output feature maps. Therefore, when compared with CNN with highway, the strength of Dilated Residual Network is that it can increase the resolution of output feature maps. However, the key component for protein secondary structure prediction methods is to learn dependencies among residues with very long distance. It may not be unnecessary to increase the resolution of output feature maps for amino acid residues.

To demonstrate the strength of our proposed CNNH_PSS over Dilated Residual Network, we compared CNNH_PSS with Dilated Residual Network on both CB6133 and CB513. Table 3.3 shows that CNNH_PSS performs better than Dilated Residual Network by 3.0% and 3.3% on CB6133 and CB613, respectively. Although Dilated Residual Network also can extract long-range dependencies between residues through increasing the receptive field of individual neurons, the better performance of CNNH_PSS demonstrates that our proposed CNNH_PSS can extract more effective long-range dependencies for protein secondary structure predictions than Dilated Residual Network.

3.2.3 Comparison with state-of-the-art methods

Protein secondary structure prediction is an important problem in bioinformatics and critical for analyzing protein function and applications like drug design. So many state-of-the-art methods have been proposed for the prediction. SSpro8 is a prediction method proposed by Pollastri et al. [134] by combining bidirectional recurrent neural networks (RNN) and PSI-BLAST-derived profiles. CNF is a Conditional Neural Fields based method which was proposed by Wang et al. [185]. CNF not only can extract relationships between sequence features of residues and their secondary structures, but also capture local context[185]. Later, an extension version of CNF (DeepCNF) was proposed by Wang et al.[183] using deep learning extension of conditional neural fields, which is an integration of conditional neural fields and shallow neural networks. It can extract both complex sequence-structure relationship and dependency between adjacent SS labels. These three methods only make use of local context for prediction. The GSN method proposed by Zhou and Troyanskaya[209] uses a supervised generative stochastic network and convolutional architectures. Using supervised generative stochastic networks is a recently proposed deep learning technique[15] well suited for extracting local context and also has the ability to capture some long-range dependencies. The DCRNN method, recently pro-

Table 3.4: Q8 accuracy of CNNH_PSS and state-of-the-art methods containing only local context

Method	CB513
SSpro8	0.511
CNF	0.633
DeepCNF	0.683
CNNH_PSS	**0.703**

Table 3.5: Q8 accuracy of CNNH_PSS and state-of-the-art methods containing both local context and long-range dependences

Method	CB6133	CB513
GSN	0.721	0.664
DCRNN	0.732	0.694
CNNH_PSS	**0.740**	**0.703**

posed by Li and Yu [92], is the best method up to now. DCRNN uses a multi-scale CNN and three staked bidirectional gate recurrent units (BGRUs)[41]. GSN and DCRNN can extract both local context and long-range dependencies. But, they need to combine two complex models to capture the two types features, separately.

In this evaluation, We first compare our proposed CNNH_PSS with the three state-of-the-art methods which can extract only local context on CB513. The result is listed in Table 3.4, where the best performers and the second best performers are marked by bold and underscore, respectively. Table 3.4 shows that CNNH_PSS outperforms the three methods by at least 0.020 Q8 accuracy. This indicates that the long-range dependencies extracted by CNNH_PSS are indeed useful features for protein secondary structure prediction.

The next evaluation uses both CB6133 and CB513 to compare all the methods that can extract long range dependencies including our CNNH_PSS, GSN, and DCRNN. Results for both CB6133 and CB513 are listed in Table 3.5, where the best performers and the second best performers are marked by bold and underscore, respectively. Result shows that CNNH_PSS performs better than both GSN and DCRNN by at least 0.008 Q8 accuracy on CB6133 and 0.009 Q8 accuracy on CB513. Note that both GSN and DCRNN use CNN for local context extraction and an additional model for long-range dependencies extraction. We conducted a CPU consumption evaluation for these three methods using GTX TITANX GPU. CNNH_PSS tends to converge after only half an hour whereas DCRNN needs more than 24 hours to converge [92]. So CNNH_PSS is almost 50 times faster than DCRNN. Although the exact running time for GSN is not known, GSN does need to be trained for 300 epochs [209] which is much larger than that of CNNH_PSS as CNNH_PSS tends to converge after training for less than 35 epochs as shown in Figure 3.3. It means that CNNH_PSS is almost 9 times faster then GSN. These two evaluations indicate that CNNH_PSS not only can extract both local context and long-range dependency more effectively, but also more efficiently.

3.2.4 Learned local context and long-range dependency

The advantage of our proposed CNNH_PSS over state-of-the-art methods is that it can extract both local context and long-range dependency by using multi-scale CNN with highway. In CNNH_PSS, every layer has a highway to deliver a portion of information from the input to the output. In each layer, the highway and the kernels can extract local context and long-range dependencies, respectively. In this section, we use CNNH_PSS with 5 layers and the kernel length of 11 to introduce the extraction of both local context and long-range dependency as shown in Figure 3.4. First, the target protein is inputted to the first layer and the kernels in this layer can extract local context from the input protein. So the output of the first layer contains local context within 11 consecutive residues. The information from the output of the first layer is delivered to the output of the second layer in two ways: direct delivery by the highway in the second layer and delivery as the result of the processing by the second layer kernels. So, the output of the second layer is the weighted sum of these two pieces of information. As the kernels in the secondary layer can extract relationships between residues with distance of up to 19 residues, the output of the second layer contains both local context within 11 consecutive residues and dependencies

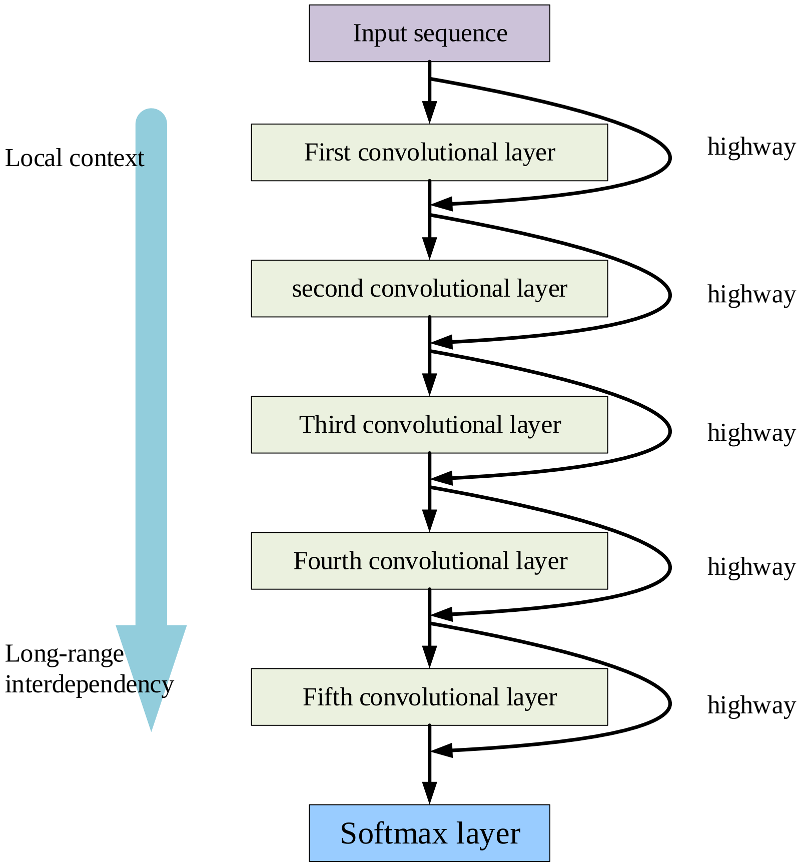

Input sequence
Local context
First convolutional layer
highway
second convolutional layer
highway
Third convolutional layer
highway
Fourth convolutional layer
highway
Long-range interdependency
Fifth convolutional layer
highway
Softmax layer

Figure 3.4: Extraction process for local context and long-range dependences

between residues with distance of 19 residues. Using the same principle, the output of the fifth layer also contains two parts. The first part is the information from the output of the fourth layer delivered by the highway, which contains local context within consecutive 11 and dependencies between residues with distance of 19, 29 and 39 residues. The second part is the information output by the kernels of the 5th layer, which contains dependencies between residues with distance of up to 49 residues. Therefore, CNNH_PSS can output local context within 11 consecutive residues and dependencies between residues with distance of 19, 29, 39 and 49 residues, In contrary, the multi-scale CNN with the same number of convolution layers only outputs dependences between residues with distance of 49 residues without local context within 11 consecutive residues and nor dependences between residues with distance of 19, 29 and 39 residues.

In order to demonstrate the importance of learned local context and long-range dependencies in protein secondary structure prediction, we show learned local context and long-range dependencies in a representative protein PDB 154L[152], obtained from the publicly available protein data bank[17]. The learned local context and long-range dependencies by CNNH_PSS in protein PDB 154L are shown in Figure 3.5, where the protein contains 185 amino acids.

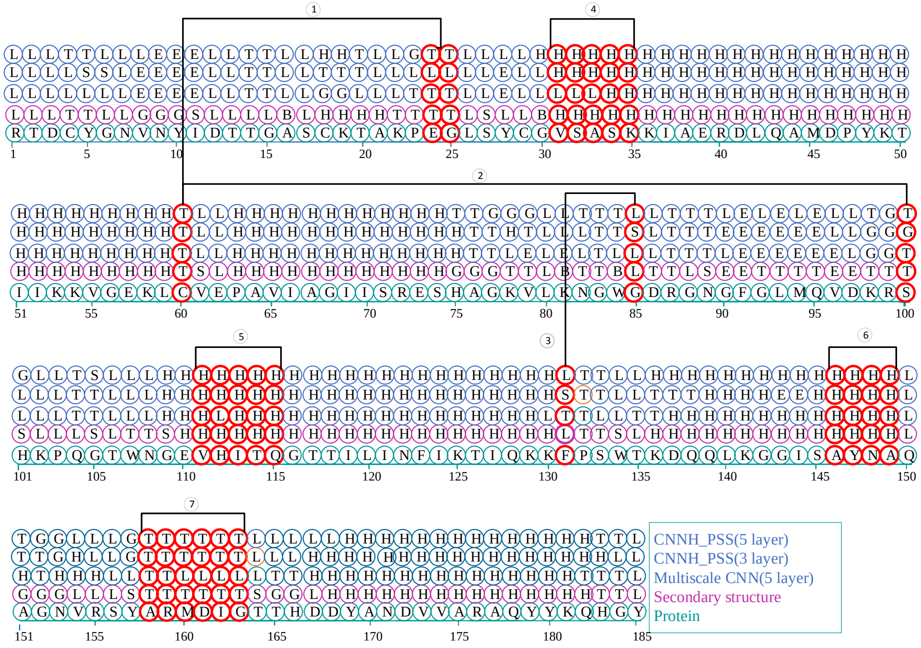


Figure 3.5: Prediction results of 154L by CNNH_PSS and comparing methods

In Figure 3.5, the first three lines correspond to the predicted results by (1) CNNH_PSS with 5 layers, (2) CNNH_PSS with 3 layers, (3) multi-scale CNN with 5 layers, respectively, and the next two lines are (4) the actual secondary structures and (5) the amino acid sequence of the protein PDB 154L, respectively. The comparison between their predicted results is more suitable for demonstrating the importance of local context and long-range dependences in protein secondary structure prediction. Figure 5.5 shows three instances for long-range dependencies: (1) dependency between the 24th and the 60th residue and that between the 25th and the 60th residue; (2) dependency between the 60th and the 100th residue and (3) dependency between the 85th and the 131th residue. As these three learned dependencies are composed by residues with distance of more than 29 and less than 49 residues, both CNNH_PSS with 5 layers and the multi-scale CNN with 5 layers can extract them while CNNH_PSS with 3 layers cannot capture them. So both CNNH_PSS with 5 layers and the multi-scale CNN with 5 layers make correct prediction for the 24th, 25th, 85th, 100th and 131th residues while CNNH_PSS with 3 layers cannot make correct predictions for them. It validates that CNNH_PSS with more layers indeed can extract long-range dependences between more remote residues.

Furthermore, Figure 3.5 also shows 4 instances for learned local context: (1) context from the 31th to the 35th residues; (2) that from the 111th to the 115th residues; (3) that from the 146th to the 149th residues and (4) that from the 158th to the 163th residues. Both CNNH_PSS with 3 layers and that with 5 layers can learn these four context so that the secondary structures of all the residues in the learned context can be correctly predicted. However, the multi-scale CNN with 5 layers cannot learn these four context. So it cannot predict the secondary structures correctly for some of these residues. It validates that the highways in the CNNH_PSS indeed can be used to extract local context for prediction. Our future works will validate these conclusions by experimental methods.

3.3 Chapter Summary

We propose a novel CNNH_PSS method by using a unified model of multi-scale CNN with an additional highway as a mechanism to obtain long-range dependencies. CNNH_PSS is able to extract long-range dependences by higher layers and retain local context extracted in lower layers through the highway of every convolution layers. Contrast to existing methods, which either cannot extract local context or long-range dependency or need to combine two complex models to extract both of them, our proposed CNNH_PSS requires only one model to extract both of them. The advantage of CNNH_PSS is demonstrated in both efficiency and performance. In terms of efficiency, CNNH_PSS is almost 50 times faster than DCRNN and 9 times faster than GSN when trained on GTX TITANX GPU. In terms of performance, CNNH_PSS outperforms the state-of-the-art methods on CB513 by at least 0.008 in Q8 accuracy. When evaluated on CB6133, CNNH_PSS outperforms the state-of-the-art methods by at least 0.009 in Q8 accuracy.

Chapter 4

DNA binding residue prediction

DNA binding residues are the residues which can bind to a corresponding DNA fragment in a protein sequence. For example, TFs have many binding residues in their sequence. Through these binding residues, TFs can bind to corresponding DNA fragments including promoters, enhancers, insulators and silencers to regulate gene expressions. The identification of DNA binding residues is crucial to understand the recognition mechanism between proteins and DNA as well as the mechanism of transcriptional regulation. The identification of DNA binding residues also provides basic knowledge for understanding the pathogenesis of several diseases. For example, DNA binding residues on the repressor protein P53 provides information about certain diseases, such as certain tumors [35].

As presented in Chapter 2, the three basic feature categories of residues do not contain information on the relationships between residues. Current methods of using feature concatenations in the contextual residues are based on a simple hypothesis that features of the target residue and that of its context are independent and thus each feature is used as one dimension in the final feature space. The relationships between residues cannot be extracted by concatenated features.

As functions and structures of residues are often related closely to their contextual residues, we hypothesize that relationships among residues are meaningful for the predictions of their functions and structures and the inclusion of these features should be able to improve prediction performance. Based on this hypothesis, four methods are proposed in this chapter to include extracted relationships between residues for DNA binding residue prediction. The first method uses PSSM Relationship Transformation (PSSM-RT) to encode residues by incorporating pairwise relationships of evolutionary information between residues. The second method use a CNN based methed, CNNsite, to encode residues on sequence features, which can extract relationships among multiple residues. The third method uses a LSTM based method, EL_LSTM, to encode residues on sequence features to capture relationships among residues both at short range (local context) and long range (long-range dependency). The last method, PDNAsite, uses sliding windows for both sequence feature and spatial features, to extract relationships among target residues and their spatial neighbor residues as well as relationships among target residues and their sequence neighbor residues.

4.1 Material and feature representation

4.1.1 Datasets

For evaluation of DNA binding residues, three commonly used benchmarking datasets and two independent datasets are used. The benchmarking datasets used by many works in the literature include PDNA-62, PDNA-224 and DBP-123 [91, 3, 192]. HOLO-83 and TS-61 [192] serves as two independent datasets which are used only as testing sets.

PDNA-62 [3] was firstly constructed by Ahmad et al. to train an ANN classifier to distinguish DNA binding sites from non-binding residues. It was later employed to train different machine learning classifiers by many studies, including ANN, SVM, Random Forest and Naïve Bayes12,14,17,21. PDNA-62 was derived from the structure data of 62 protein-DNA complexes in the Protein Data Bank (PDB)20. The dataset contains 67 sequences and the sequence identity between any two sequences is less than 25%. As in most previous studies, in the structure of the protein-DNA complexes, a residue in protein is regarded as interacting with DNA if the side chain or the backbone atoms of the residue falls within a cutoff distance of 3.5 Å from any atom of the partner DNA molecule in the complex. All the other residues were regarded as non-binding sites. As a result, this data set contains 1,215 DNA binding residues and 6,948 non-binding residues. As this dataset has been used in many studies, it is convenient for comparing the predicting accuracy of PDNAsite with that of other existing methods.

PDNA-224 was collected by Li et al. [91] from the Protein Data Bank (PDB) (released by January 2011) [19], which contains 224 protein chains with pairwise sequence identity less than 0.25. The binding residues and the non-binding residues for this dataset follow the definition of Ahmad et al. [3]: a residue in a protein is regarded as a binding residue if the side chain or the backbone atoms of the residue falls within a cut-off distance of 3.5 angstroms from any atom of the partner DNA molecule in the complex; otherwise, the residue is considered as a non-binding residue.

DBP-123 was collected by Xiong et al. [192] from PDB (September 2009 release) [19] and it is composed of 123 protein sequences of 119 protein-DNA complex. The sequence identity between any two sequences is less than 0.25 and each sequence contains a minimum of 40 residues. The binding residues and the non-binding residues follow the definitions of Tjong and Xiong [166, 192]: a residue is considered as a surface residue if its exposed surface area is larger than 10% of its nominal maximum area [141]; Further, a surface residue is defined as a binding residue if it contains at least one heavy atom that falls within the distance of 4.5 angstroms from any heavy atoms of the DNA molecule.

HOLO-83 is a common independent dataset given by Xiong et al. from 82 protein-DNA complexes [192]. The 83 protein sequences have sequence similarity less than 0.25. The sequence identity between HOLO-83 and DBP-123 is less than 0.25. The protein sequences from HOLO-83 was built from PDB's September 2009 release. So, it did not include some of the newly decoded proteins. To include the newer decoded protein sequences, we build a new independent dataset, referred to as TS-61 from the December

2016 release of PDB.

TS-61 is constructed in this work by applying the follow procedure: (1) retrieving protein-DNA complexes from PDB; (2) screening obtained sequences with the cut-off pairwise sequence similarity of 25% as well as dissimilarities to PDNA-224, DBP-123, and HOLO-83 of at least 75%. This selection process ends up with 61 sequences. The PDB id and the chain id of the 61 protein sequences in TS-61 are listed in addition file 1, which can be obtained from our web site http://hlt.hitsz.edu.cn/EL_LSTM/. Both HOLO-83 and TS-61 serve as testing data. Training for their predictions uses DBP-123. Thus, the binding residues for HOLO-83 and TS-61 use the same definitions as that of DBP-123. The number of binding residues and non-binding residues of these five datasets are summarized in Table 4.1.

Table 4.1: The details of the four datasets

datasets	PDNA-62	PDNA-224	DBP-123	HOLO-83	TS-61
binding residues	1,215	3,778	2,895	2,038	1,491
non-binding residues	6,948	53,570	15,428	12,200	8,385
total number	8,163	57,348	18,323	14,238	9,876

4.1.2 Feature Descriptors

For prediction of DNA binding residues, residues are the prediction targets. As the functions of each residue are closely related to its contextual residues, the fragment including the target residue and its context, referred to as a residue-wise data instance, is used to learn the feature representation for the target residue. A **residue-wise data instance** is observed within a sliding widow of length w, where the target residue is positioned in the center and $(w-1)/2$ contextual residues are located on both sides. For a given protein sequence P with length of L, it is defined by a sequence of residues denoted by

$$P = R_1R_2R_3R_4R_5R_6\dots R_L \tag{4.1}$$

where R_1 denotes the first residue of the chain P, R_2 is the second residue and so forth. For the target residue R_t at position t of the protein sequence P, its residue-wise data instance S_t is defined as a fragment with w (w is an odd number) residues where the target residue R_t is positioned in the middle of the fragment with the other $w-1$ residues located on either sides of R_t. So the residue-wise data instance S_t for the target residue R_t can be represented as

$$S_t = R_{t-\frac{w-1}{2}} R_{t-\frac{w-3}{2}} \dots R_{t-1} R_t R_{t+1} \dots R_{t-\frac{w-1}{2}} \tag{4.2}$$

where R_t in the middle is the target residue and the other $(w-1)$ residues are its contextual residues. For example, in protein 1NG9 [PDB:1NG9] chain A, the residue-wise data instance S_6 for the 6th residue is ”MSAIE**N**FDAHT”, where ”N” is the target residue. The set of residues ”MSAIE” on the left side and ”FDAHT” on the right side are referred to as its left context and its right context, respectively. A residue-wise data instance is defined as a positive sample if the target residue is a DNA binding residue or a negative sample if the target residue is a non-binding residue. Features for a target residue are extracted from its residue-wise data instance.

Residues have three commonly used feature types: sequence features, evolutionary features, and structure features. **Sequence features** include identity property and physiochemical properties. The identity property of a residue is represented as a 20-dimensional one-hot vector with each position corresponds to one residue type. Thus, for a residue, its identity property vector has the corresponding entry being set to 1 and the remainder positions are all 0s. For example, the identity of the residue Arg is represented with a 20-dimensional feature vector with the fourth position as 1 and the remaining positions as 0. Physiochemical properties contain pKa index and hydrophobic index. Sequence features are encoded by concatenating identity property with physiochemical properties.

Evolutionary features are often represented by the Position Specific Score Matrix (PSSM). PSSM is obtained by running the PSI-BLAST program to search against the non-redundant (NR) database through three iterations with 0.001 as the E-value cutoff for multiple sequence alignment. PSSM for a protein sequence with length L is represented as a matrix with dimension of $L \times 20$, in which the score in row i and column j represents the conservation degree of residue type j in position i. PSSM for a residue is a row in PSSM. A score in PSSM needs to be scaled between 0 and 1 using the following equation

$$\text{NPSSM}(i, j) = \frac{1}{1 + e^{-\text{PSSM}(i,j)}} \tag{4.3}$$

Structure features are extracted from the 3D structures of protein sequences. Commonly used structure features include solvent accessibility surface area, B-factor, packing density and secondary structures. The solvent accessibility surface area (SASA) is calculated from protein 3D structure by the DSSP program. SASA, to be used as feature, needs to be normalized by dividing the max SASA value of the residue type. Packing density is defined as the number of non-covalently bounded residues whose Ca position falls within a sphere of 6Å radius from the Ca position of the target residue. Secondary structures include three types and are often represented as a 3 dimensional one-hot vectors.

4.1.3 Evaluation metrics

As the residue-wise data instances in this paper are extracted by a sliding window of size w on protein sequences, the value of w needs to be set properly. Due to the length of this paper, we are showing the experiments for parameter training. Size of 11 is experimentally validated and used in all the following experiments.

Five common metrics are used in performance evaluation: Sensitivity (SN), Specificity (SP), Strength (ST), Accuracy (ACC), and Mathews Correlation Coefficient (MCC). Their respective definitions are listed below:

$$\text{SN} = TP/(TP + FN)) \tag{4.4}$$

$$\text{SP} = TN/(TN + FP)) \tag{4.5}$$

$$\text{ST} = (\text{SN} + \text{SP})/2 \tag{4.6}$$

$$\text{ACC} = (TP + TN)/(TP + FP + TN + FN) \tag{4.7}$$

$$\text{MCC} = \frac{TP * TN - FP * FN}{\sqrt{T * (TP + FP) * N * (TN + FN)}} \tag{4.8}$$

where TP is the number of true positives; TN is the number of true negatives; FP is the number of false positives; FN is the number of false negatives; T is the total number of positives; N is the total number of negatives.

Since the non-binding residues outnumber the binding residues by at least 5.5 times in our datasets, accuracy ACC cannot provide unbiased evaluation because by classifying all test samples as non-binding residues without running any machine learning algorithm will already give a very high ACC value. For skewed datasets, the most important performance measure is ST and MCC. Many works in literature [180, 181, 182, 91] pointed out that ST, the average of SN and SP, can provide a more appropriate evaluation for skewed datasets. We also use the ROC (Receiver Operating Characteristic) curve and the area under the ROC curve(AUC) as unbiased performance measures. ROC curve is a standard representation for the trade-off between false positive rate and sensitivity. The curve is drawn by plotting the true positive rates (i.e. sensitivity) against the false positive rates (i.e. 1-specificity) by varying the classification threshold. So, the area under the ROC curve (AUC) is also an unbiased metric for unbalanced dataset. Therefore, ST, MCC and AUC are used as the main metrics and shaded for easy observation in all the forms of this chapter. The other three metrics, SN, SP, and ACC are used for reference only.

To demonstrate the significance of the performance improvement, we apply the Wilcoxon signed-ranks test[189] to compute the p-value for the comparison. The Wilcoxon signed-ranks test[189] is a non-parametric test method, which ranks the difference in performances of two classifiers for each dataset, ignoring the signs, and compares the ranks for positive and negative differences [8]. Let R^+ be the sum of ranks for the datasets on which the second classifier outperforms the first one, R^- be the sum of ranks for the opposite and T be the smaller of the two sums. For a comparison with N datasets, the statistics

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{4.9}$$

is considered to be distributed approximately normally when N is larger ($\geqslant$ 20). For more details about the Wilcoxon signed-ranks test, [189] and [189] should be helpful references. The p-value for all the performance comparisons in this paper is computed by the Wilcoxon signed-ranks test.

4.2 PSSM-RT_SVM: PSSM Distance Transformation based method

4.2.1 PSSM Distance Transformation

Evolutionary features is produced by the evolutionary processes and it is important for protein structure and function prediction. PSSM is a common representation for evolutionary features and has been used in many bioinformatics studies including protein functionality annotation and protein structure prediction [82, 70, 23, 174, 207]. For every protein sequence in this study, its PSSM is calculated from multiple sequence alignments produced by running the PSI-BLAST program [144] to search the non-redundant (NR) database through three iterations with the E-value cutoff at 0.001. For a protein with length L, PSSM is usually represented as a matrix with L x 20 dimensions. 20 denote the 20 standard types of residues. For the sequence fragment F_i using representation defined in Formula (2), its PSSM can be represented as a matrix with dimensions w x 20. Thus, the

PSSM of the residue-wise instance F_i for the target residue R_i can be formulated as

$$\text{PSSM}_{F_i} = \begin{bmatrix} S_{i-\frac{w-1}{2},1} & \cdots & S_{i-\frac{w-1}{2},r} & \cdots & S_{i-\frac{w-1}{2},20} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ S_{i,1} & \cdots & S_{i,r} & \cdots & S_{i,20} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ S_{i+\frac{w-1}{2},1} & \cdots & S_{i+\frac{w-1}{2},r} & \cdots & S_{i+\frac{w-1}{2},20} \end{bmatrix} \tag{4.10}$$

where $S_{i,r}$ is the conservative score of residue type r at position i in the sequence fragment.

Before PSSM-RT is calculated, the conservative scores in PSSM should be normalized between 0 and 1. Thus, for a given $S_{i,r}$, its normalized value $S_{i,r}^{(N)}$ can be expressed by a logistic function given below

$$S_{i,r}^{(N)} = \frac{1}{1 + e^{-S_{i,r}}} \tag{4.11}$$

PSSM-RT contains three categories of features: residue conservations, pair-relationships and multi-relationships. The residue conservations contain the PSSM scores of the target residue and its context residues. The pair-relationship is defined as the relationship of evolutionary information between two residues, for example, the pair-relationship between the residue r_1 of position i and the residue r_2 of position j is calculated as

$$\text{PSSM-RT}(i, j, r_1, r_2) = S_{i,r_1}^{(N)} * S_{j,r_2}^{(N)} \tag{4.12}$$

As every position in a residue-wise data instance has conservative scores for the 20 standard residue type, 400 types of relationships can be calculated for any two positions. As the target position in a residue-wise data instance is influenced by all its context positions, the all pair-relationships for the target position and its context positions needs to be included in the prediction. Thus the pair-relationship for a residue-wise data instance is defined as the sum of pair-relationship for the target position and all its context positions. For example, the pair-relationship between residue r_1 and residue r_2 for a residue-wise data instance with i as its target position is formulated as

$$\text{PSSM-RT}(i, r_1, r_2) = \sum_j \text{PSSM-RT}(i, j, r_1, r_2) \tag{4.13}$$

where j is the context position of the target position.

Multi-relationships are the evolutionary information relationships between multiple residues. We consider two kinds of multi-relationships: the left multi-relationships that include the relationships between the target residue and its left context residues and the right multi-relationships that include the relationships between the target residue and its right context residues. For residue r, the left multi-relationship of residue-wise data instance at target position i is formulated as

$$\text{PSSM-RT}_{left}(i, r) = \sum_{k=i-\frac{w-1}{2}}^{i} S_{k,r}^{(N)} \tag{4.14}$$

For residue r, the right multi-relationship of residue-wise data instance at target position i is formulated as

$$\text{PSSM-RT}_{right}(i, r) = \sum_{k=i}^{i+\frac{w-1}{2}} S_{k,r}^{(N)} \tag{4.15}$$

Thus, the dimension of the feature space constructed by PSSM-RT is $(20 * w + 20 * 20 + 2 * 20)$.

In addition to PSSM-RT, there are two other types of features that are used in this work: sequence features and physiochemical features. Sequence features given in the datasets include amino acid composition, predicted secondary structure, predicted solvent accessible area, and identity of the target residue. Physiochemical features include pKa values of amino group, pKa values of carboxyl group, electron-ion interaction potential (EIIP)[25], number of lone electron pairs(LEPs), Wiener index [170], molecular mass [170], side chain pKa value, and hydrophobicity index. The predicted secondary structure and predicted solvent accessible area are obtained by applying PSIPRED [113] and SABLE [1, 2, 178], respectively.

4.2.2 Ensemble learning

Ensemble learning is now an active area of research in machine learning and pattern recognition. Ensemble learning first learns several base predictors from the training dataset and then combines them into an ensemble predictor. Ensemble learning aims to take advantage of the different learning ability of the different base predictors. There are three widely used ensemble strategies to train base predictors: training by different data subsets, training from different feature subsets and training by different classification algorithms.

In DNA binding residue prediction, non-binding residues outnumber binding residues by a large margin. In order to get a balanced dataset for training, many predictors chose to discard a large part of non-binding residues [172]. However, discarded non-binding residues may potentially be useful information to improve prediction performance. In order to better use all the data available, we propose to use ensemble learning by combining all the three ensemble strategies. And then use our proposed method, referred to as EL_PSSM-RT, to combine the ensemble learning model with PSSM-RT. The system architecture of EL_PSSM-RT is shown in Figure 4.1. Note that EL_PSSM-RT contains 4 steps: Dataset Partition, Feature Extraction, Base Classifier Training and Base Classifier Selection. In Step 1 of Dataset Partition, the non-binding residues in the training dataset

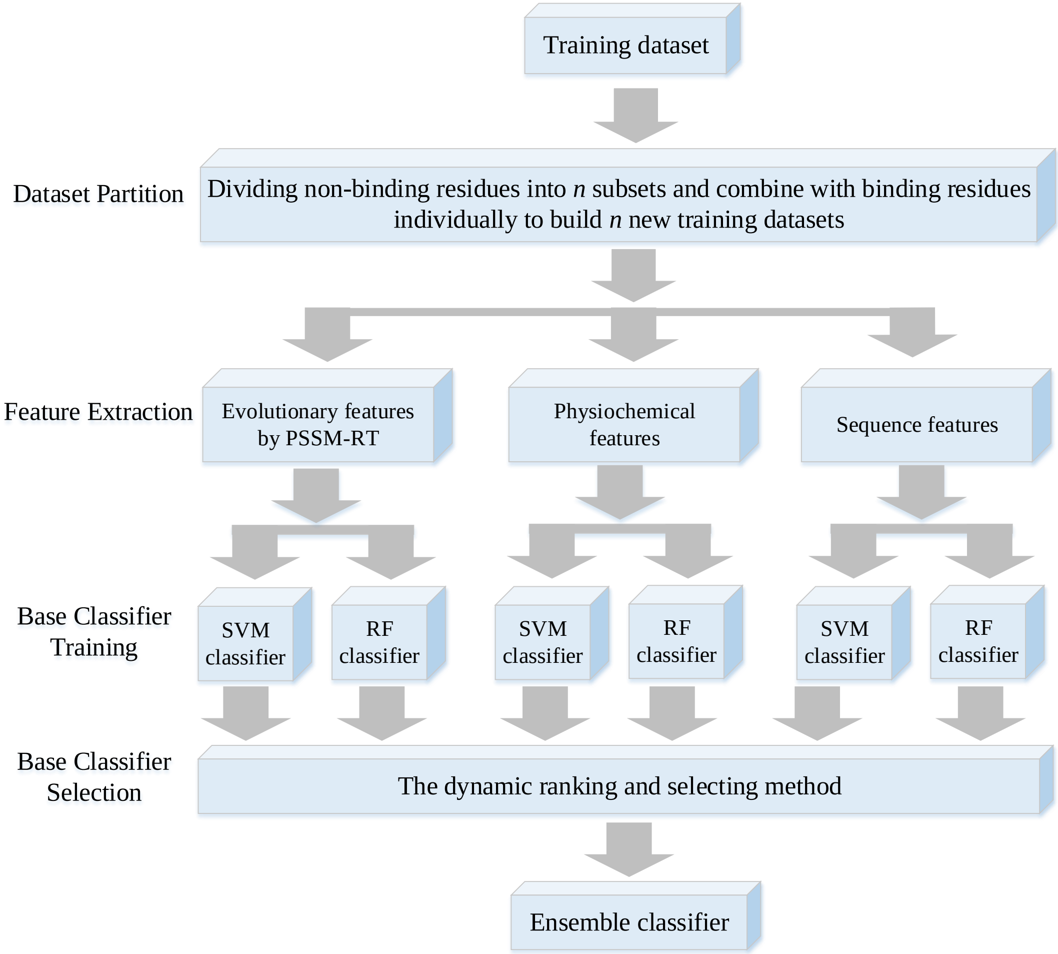

Training dataset
Dataset Partition
Dividing non-binding residues into n subsets and combine with binding residues individually to build n new training datasets
Feature Extraction
Evolutionary features by PSSM-RT
Physiochemical features
Sequence features
Base Classifier Training
SVM classifier
RF classifier
SVM classifier
RF classifier
SVM classifier
RF classifier
Base Classifier Selection
The dynamic ranking and selecting method
Ensemble classifier

Figure 4.1: The framework diagram of EL_PSSM-RT.

are first partitioned into n non-overlapping subsets with the number of samples approximately equal to that of all the binding residues. Then, n new balanced training datasets are formed by adding the binding residues into the n subsets non-binding residues. In Step 2 of Feature Extraction, three categories of features are extracted for residues including sequence features, physiochemical features, and evolutionary features extracted by PSSM-RT. In Step 3 of Base Classifier Training, both the SVM classifier and the Random Forest classifier are used by each category of features on every newly formed training dataset. SVM and Random Forest are used because they are proven to have good predicting performances for DNA binding residue prediction [180, 181, 182]. Thus, $6 * n(2 * 3 * n)$ base predictors are trained in this step. In Step 4 of Base Classifier Selection, a diversity based dynamic ranking and selecting method is designed based on diversity to build the ensemble predictor using an iterative approach. In our dynamic ranking and selecting method, a base predictor is initially selected at random. Then in each iteration, all the unselected base predictors are first ranked based on their diversity with the selected base predictor(s), followed by the selection step in which the one with the largest diversity will be added into the set of selected predictors. The iteration is terminated when the addition of diversity for the set of selected predictors is less than a specified criterion. Finally, the selected base predictors are combined to construct an ensemble predictor by the majority vote strategy.

4.2.3 Experiments and Results

The purpose of the evaluation is to examine the effectiveness of our proposed PSSM-RT over other methods. Four sets of evaluations are conducted here. Experiment 1 compares PSSM-RT with previous encoding methods. Experiment 2 compares the ensemble learning model with the base classifiers. Experiment 3 compares our proposed predictor EL_PSSM-RT with previous predictors, and Experiment 4 evaluates EL_PSSM-RT on two independent datasets. Based on the obtained data, we further analyze the relation-pairs of amino acids followed a case study of two proteins in the identified binding-residues.

Experiment 1: Comparison of PSSM-RT with previous encoding methods

Since PSSM-RT uses a window based approach, the window size needs to be set properly. For the SVM classifier which uses PSSM-RT as features, the performance of the SVM classifier with different window size w is shown in Figure 4.2. It can be seen that the ST value continues to increase and peaks when w reaches 13. So, the window size $w = 13$ is used for all the SVM classifiers.

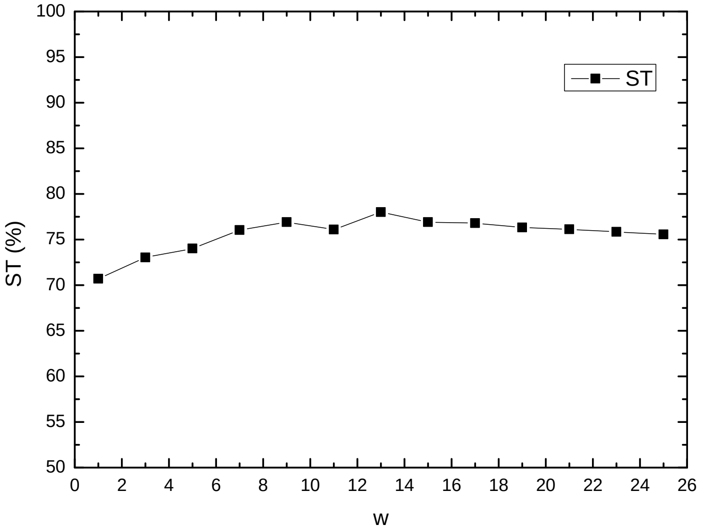


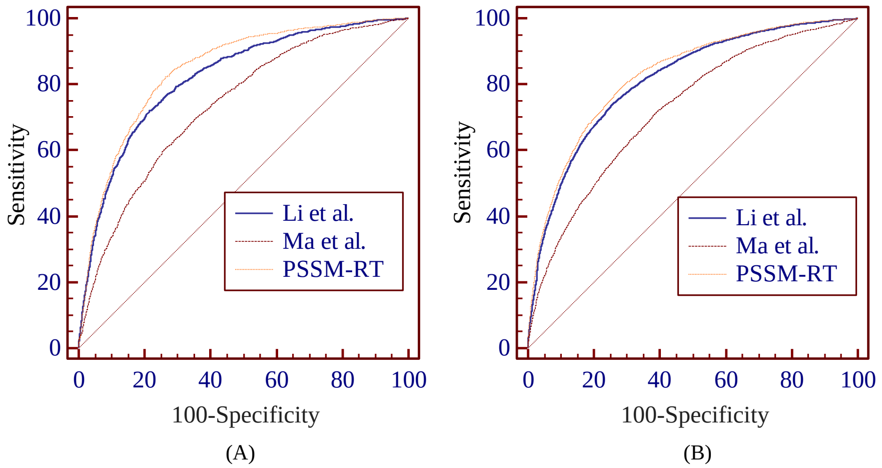
Figure 4.2: The compact of window size w on performance of PSSM-RT.

This set of experiments compares PSSM-RT with two types of existing encoding methods: the combination methods and the concatenation methods. As both of them contain several methods, we only consider the state-of-the-art works for the respective groups. Consequently, Ma et al.'s work using combination method [104] and Li et al.'s work [91] using the concatenation methods are used for comparison. Both Ma et al.'s work [104] and Li et al.'s work [91] used SVM as the classifier, so we also use SVM as the classifier in this experiment. Since both Ma et al.'s work and Li et al.'s work did not provide the performance for evolutionary features and combination with sequence features on PDNA-62 and PDNA-224, their methods are implemented in this study to obtain evaluation data. The performances on both datasets PDNA-62 and PDNA-224 are shown in Table 4.2, where the best performers and the second best performers are marked by bold and underscore, respectively. The corresponding ROC curves are shown in Figure 4.3(A) and 4.3(B), respectively.

From Table 4.2, we can see that PSSM-RT performs better than Ma et al.'s work on both datasets with p-values less than 3.05E-5, which means the improvement is quite significant. More specifically, the increase in the PDNA-62 dataset is 0.17 on MCC, 11.06%

Table 4.2: Performance for evolutionary features on benchmark datasets by SVM.

Dataset	Methods	ACC (%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PDNA-62	Ma et al.	72.23	0.26	59.45	74.48	66.96	0.734
	Li et al.	**77.32**	0.40	72.00	**78.27**	75.14	0.821
	PSSM-RT	76.45	**0.43**	**80.23**	75.80	**78.02**	**0.845**
PDNA-224	Ma et al.	76.88	0.18	50.59	78.87	64.73	0.723
	Li et al.	79.18	0.29	67.21	80.09	73.65	0.813
	PSSM-RT	**80.39**	**0.31**	**68.11**	**81.32**	**74.72**	**0.826**



(A)

(B)

Figure 4.3: Comparison between different encoding methods.

on ST and 0.111 on AUC and 0.13 on MCC, 9.99% on ST and 0.103 on AUC for the PDNA-224 dataset. PSSM-RT outperforms Li et al.'s work quite significantly on both datasets with p-value less than 4.71E-5. More specifically, the increase in the PDNA-62 dataset is 0.03 on MCC, 2.88% on ST and 0.024 on AUC and 0.02 on MCC, 1.07% on ST and 0.013 on AUC on the PDNA-224 dataset. Figure 4.3(A) and 4.3(B) show that PSSM-RT has the best ROC curve on both PDNA-62 and PDNA-224.

When both sequence features and physiochemical features are added, the performances of the three methods on PDNA-62 and PDNA-224 are shown in Table 4.3, where the best performers and the second best performers are marked by bold and underscore, respectively. The corresponding ROC curves are shown in Figure 4.4(A) and 4.4(B). Table 4.3

shows the same performance trends as that in Table 4.2. Figure 4.4(A) and 4.4(B) also show that PSSM-RT has the best ROC curve on both PDNA-62 and PDNA-224 when the three types of features are combined. This clearly indicates that PSSM-RT outperforms both Ma et al.'s work and Li et al.'s work when all three types of features are used. When comparing Table 4.2 and Table 4.3, we observe that the performance of PSSM-RT is improved by 0.05 on MCC, 1.52% on ST and 0.028 on AUC for PDNA-62 and 1.92% on ST and 0.017 on AUC for PDNA-224. This shows that PSSM-RT is complementary to the other two features. This set of experiments indicates that the relationships of evolutionary information between residues perform better than the two previous state-of-the-art encoding methods.

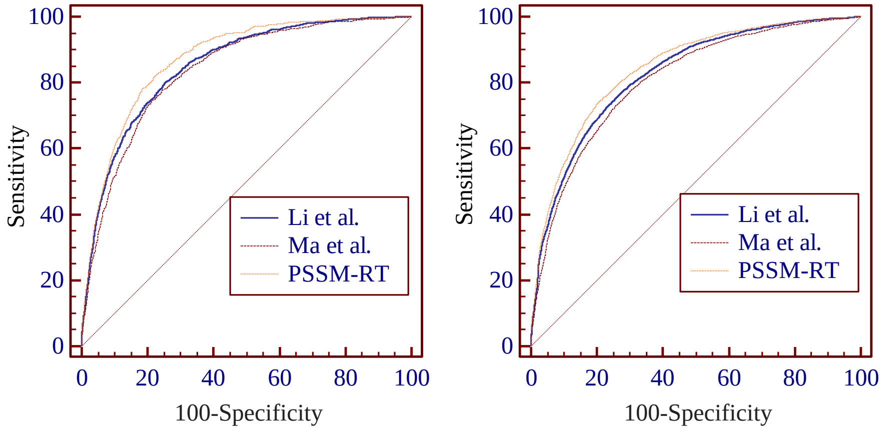
Table 4.3: Performance for all features on benchmark datasets by SVM.

Dataset	Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PDNA-62	Ma et al.	75.11	0.40	**78.22**	74.58	76.40	0.832
	Li et al.	77.81	0.42	75.50	78.24	76.87	0.851
	PSSM-RT	**81.50**	**0.48**	76.74	**82.34**	**79.54**	**0.873**
PDNA-224	Ma et al.	76.66	0.27	68.59	77.25	73.10	0.808
	Li et al.	**78.65**	0.29	69.48	**79.34**	74.41	0.825
	PSSM-RT	78.14	**0.31**	**74.92**	78.38	**76.65**	**0.843**

Experiment 2: Comparison of EL_PSSM-RT with base classifiers

This set of experiments compares EL_PSSM-RT with the base classifiers. The performances of EL_PSSM-RT, the SVM classifier and the Random Forest(RF) classifier are shown in Table 4.4, , where the best performers and the second best performers are marked by bold and underscore, respectively. The corresponding ROC curves are shown in Figure 4.5(A) and 4.5(B).

Table 4.4 shows that compares to both the SVM classifier and the RF classifier, EL_PSSM-RT achieves significant performance improvement on both PDNA-62 with p-value less

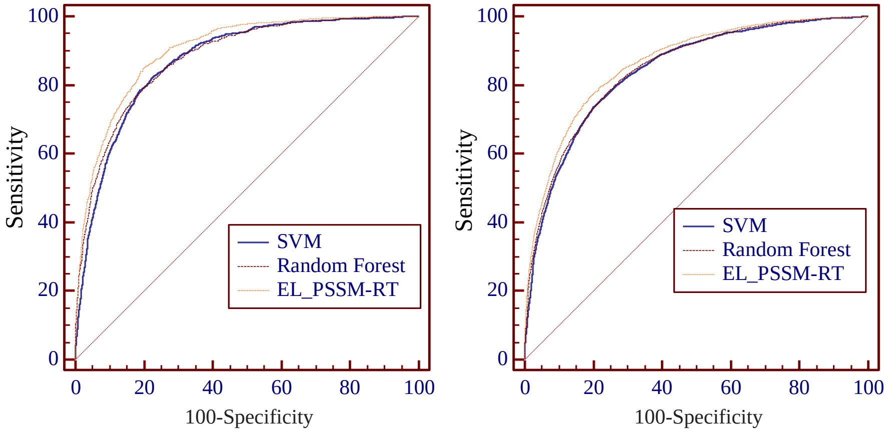


(A) (B)

Figure 4.4: Comparison between different encoding methods when combining with sequence features and physicochemical features.

Table 4.4: Comparison of EL_PSSM-RT with base classifiers on benchmark datasets.

Dataset	Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PDNA-62	SVM	**81.50**	0.48	76.74	**82.34**	79.54	0.873
	RF	80.90	0.47	77.43	81.42	79.43	0.880
	EL_PSSM-RT	80.82	**0.51**	**85.04**	80.10	**82.57**	**0.901**
PDNA-224	SVM	78.14	0.31	74.92	78.38	76.65	0.843
	RF	**80.95**	0.32	71.11	**81.69**	76.40	0.844
	EL_PSSM-RT	78.09	**0.34**	**79.58**	77.98	**78.78**	**0.865**



(A) (B)

Figure 4.5: Comparison between EL_PSSM-RT, SVM classifier and Random Forest classifier.

than 6.52E-5 and PDNA-224 with p-value less than 7.25E-5. More specifically, on the PDNA-62 dataset, the increase to the SVM classifier is 0.03 on MCC, 3.03% on ST and 0.028 on AUC and 0.04 on ACC, 3.14% on ST and 0.021 on AUC to the RF classifier. For the PDNA-224 dataset, the increase to the SVM classifier is 0.03 on MCC, 2.13% on ST and 0.022 on AUC and to the RF classifier is 0.02 on MCC, 2.38% on ST and 0.021 on AUC. Figure 4.5(A) and 4.5(B) also show that EL_PSSM-RT obtains the best ROC curve on both PDNA-62 and PDNA-224. This indicates that ensemble learning makes EL_PSSM-RT more superior than both the SVM classifier and the RF classifier.

Experiment 3: Comparison with previous predictors

This set of experiments evaluates the performance of our proposed ensemble learning based EL-PSSM-RT compared to other state-of-the art methods trained and tested either on PDNA-62 or PDNA-224 including eight algorithms: (1) Dps-pred [3], (2) Dbs-pssm [4], (3) BindN [180], (4) Dp-bind [85], (5) Dp-Bind [66], (6) BindN-RF [182], (7) BindN+ [181], and (8) PreDNA [91]. The first seven methods were trained and tested on PDNA-62. The last method, PreDNA [91], was trained and tested on both datasets. PreDNA was proposed recently and achieved the best performance for DNA binding residue prediction so far. In addition to sequence features and evolutionary information, PreDNA [91] also used structure features. As we have pointed out earlier, structure features of most proteins are unavailable and the experimental 3D structure is very expensive to obtain. Thus, PreDNA [91] cannot be used as a general method at the current time for DNA binding residue prediction on a genomic scale. For this reason, EL_PSSM-RT does not use any structure feature, similar to many other methods. To fairly compare the performance of various methods, the version of PreDNA without using structure features is used in this evaluation. The prediction accuracy of EL_PSSM-RT and other methods on PDNA-62 and PDNA-224 are shown in Table 4.5 and Table 4.6, respectively, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.5: Performance comparison of various prediction methods on PDNA-62 by five-fold cross-validation.

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Dps-pred	79.10	–	40.30	**81.80**	61.10	–
Dbs-pssm	66.40	–	68.20	66.00	67.10	–
BindN	70.30	–	69.40	70.50	69.95	0.752
Dp-bind	78.10	0.49	79.20	77.20	78.20	–
DP-Bind	77.20	–	76.40	76.60	76.50	–
BindN-RF	78.20	–	78.10	78.20	78.15	0.861
BindN+	79.00	0.44	77.30	79.30	78.30	0.859
PreDNA	79.40	0.42	76.80	79.70	78.30	–
EL_PSSM-RT	**80.82**	**0.51**	**85.04**	80.10	**82.57**	**0.901**

Table 4.6: Performance of EL_PSSM-RT Compared with PreDNA on PDNA-224 by five-fold cross-validation.

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PreDNA	**79.10**	0.29	69.50	79.80	74.60	–
EL_PSSM-RT	78.09	**0.34**	**79.58**	77.98	**78.78**	**0.865**

Table 4.5 shows that EL_PSSM-RT achieves the best performance with significant improvement with p-value less than 3.06E-5 for PDNA-62, outperforming other methods by 0.02-0.07 on MCC, 4.27%-21.47% on ST and 0.040-0.149 on AUC. For the PDNA-224 dataset, EL_PSSM-RT performs better than PreDNA by 0.05 on MCC, 4.18% on ST with p-value less than 3.64E-5. The results on both datasets indicate that the effect use of relation information and ensemble learning is superior to other existing methods.

Experiment 4: Independent tests use TS-72 and TS-61

We evaluate the performance of our EL-PSSM-RT on the TS-72 dataset so we can compare it with the previous published DNABR method [172] and the BindN series [180, 181, 182]. DNABR is a sequence based DNA binding residue prediction method proposed by Ma et al. [172]. BindN, BindN-RF and BindN+ are three methods proposed by Wang et al. using only sequence information [180, 181, 182]. the AUC values of the four published methods are 0.866, 0.748, 0.825 and 0.844, respectively according to Ma et al.' work [172] which are trained on TR265. The AUC value for EL_PSSM-RT, is 0.879. Our method increases the performance by 0.013-0.131 on AUC with p-value less than 8.37E-4 for the independent dataset TS-72.

For the second independent dataset TS-61, we compare our proposed method with DB-Bind [66]. DB-Bind [66] is a web server for predicting DNA binding sites in a DNA binding protein from its amino acid sequence. The web server implements three machine learning classifiers: DP-Bind(SVM) that uses support vector machine, DP-Bind(KLR) that use kernel logistic regression and DP-Bind(PLR) that uses penalized logistic regression. DB-Bind [66] also implements two types of consensus methods. One is majority consensus on the results of three machine learning methods by majority vote, referred to as DP-Bind(MAJ). The other is strict consensus obtained by unanimous agreement, referred to as DP-Bind(STR). The performance of EL_PSSM-RT trained by PDNA-224 and the different DB-Bind methods is shown in Table 4.7, where the best performers and the second best performers are marked by bold and underscore, respectively. From the Table 4.7 shows that our method has the best performance outperforming all the different machine learning methods in DB-Bind with 0.02-0.05 on MCC, 2.26-6.48% on ST and 0.038-0.056 on AUC.

Table 4.7: Performance of EL_PSSM-RT Compared with DP-Bind on TS-61.

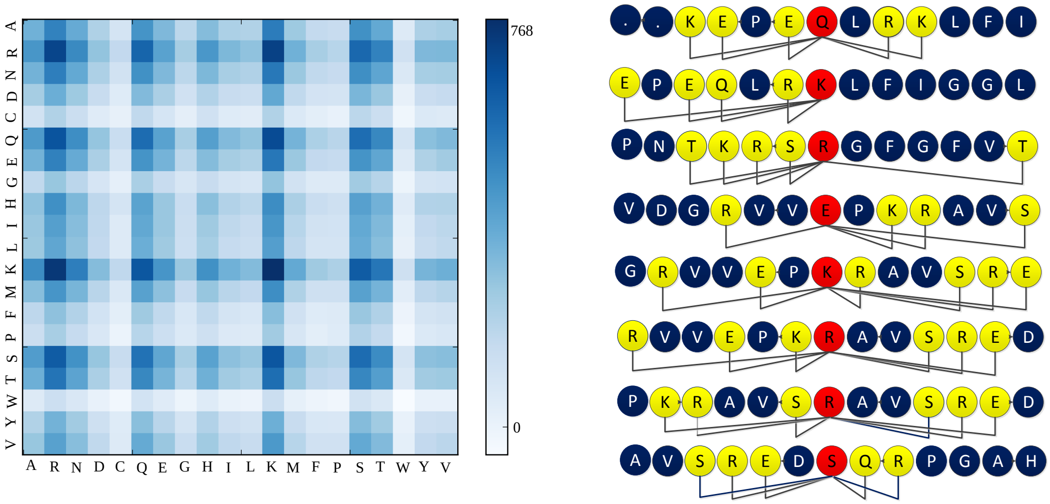
Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
DP-Bind(SVM)	75.90	0.26	65.99	76.70	71.34	0.794
DP-Bind(KLR)	76.45	0.25	64.22	77.45	70.83	0.790
DP-Bind(PLR)	75.46	0.25	65.24	76.29	70.76	0.812
DP-Bind(MAJ)	76.64	0.26	65.24	77.57	71.41	–
DP-Bind(STR)	**80.21**	0.31	**68.74**	**81.11**	74.92	–
EL_PSSM-RT	77.65	**0.33**	76.62	77.74	**77.18**	**0.850**

4.2.4 Analysis of Important Pair-Relationships and case study

To further understand the importance of PSSM-RT for DNA binding residue prediction, we analyze the important pair-relationships found by the learning algorithm. Since the importance of the relations can be reflected by the discriminant weight vector of the pair-relationships extracted by PSSM-RT, the values in the discriminant weight vector indicates the discriminant powers of the features in the feature space. Following the published works in [95, 131, 199], the discriminant weight vector **W** is calculated as follows: first, we obtain the classification weight vector **A** from the ensemble learning classifier during the training process. **W** is calculated by applying the following formulae:

$$\mathbf{W} = \mathbf{A}^{\mathrm{T}} \cdot \mathbf{M} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \cdots & \vdots \\ m_{N1} & m_{N2} & \cdots & m_{Nd} \end{bmatrix} \tag{4.16}$$

where **A** is the classification weight vector of the training dataset by the ensemble learning classifier trained on PDNA-62 and **M** is the feature vectors of all training data instances; d is the dimension of the feature space and N is the number of data instances in the training dataset. The analysis results are shown in Figure 4.6 based on the data given in the part E of the additional file 1 which lists all the discriminant weights of the 400 pair-relationships between the target residue and its neighboring residue. Figure 4.6 includes a heatmap showing the discriminant weight of every pair-relationship and a diagram of binding residues showing the pair-relationships between important residues. Figure 4.6A shows that the relationships between amino acid pairs (K, K), (K, R), (R, R), (Q, K), (Q, R), (S, K), (S, R), (R, Q), (S, S), (S, Q), (T, R), (E, K), (E, R), (E, R),(E, Q) are the fifteen relationships with larger discriminant weights. This means that the amino acids K, R, Q, S, T and E are important in the interaction between proteins and its corresponding



(A)

(B)

Figure 4.6: The feature analysis results of PSSM-RT on PDNA-62.

DNA molecular. This feature analysis result is consistent with many other works for DNA binding proteins research which stated that R, K, E and S are important for the interaction between DNA binding proteins and its responsible DNA molecules [150, 163]. Figure 4.6B shows eight DNA binding residues and its context residues extracted from the structure of a protein-DNA complex (PDB id: 1u1q). As we can see from Figure 4.6B, the relationship between R and K has the highest occurrence frequency among the eight DNA binding residues and is the most important feature for DNA binding residue prediction for this protein. The second most important feature is the relationship between R and K. The relationships between E and Q and between E and R are the third most important features. The analysis result validates the usefulness of PSSM-RT for the representation of DNA binding residues.

In order to further validate the usefulness of EL_PSSM-RT for DNA binding residue prediction, we apply EL_PSSM-RT trained on PDNA-62 to distinguish the binding residues from non-binding residues for two protein-DNA complexes which are not in the training set, namely, 1s40 and 1b3t. The proteins in these two complexes are two typical DNA binding proteins and the sequences have sequence similarity of less than 25% for all the sequences in the training set. On 1s40, EL_PSSM-RT achieves 96.71% on ACC, 0.74 on MCC, 92.06% on SN, 96.96% on SP and 94.51% on ST, respectively. This means that 34 residues out of a total of 39 actual binding residues are correctly predicted by EL_PSSM-RT and only 24 residues in the 264 non-binding residues are incorrectly predicted as binding residues. The actual residues and predicted residues in 1s40 are shown in Figure 4.7A and Figure 4.7B, respectively. The two figures show that most of the real binding residues overlap with the predicted binding residues. This provides a visual indication that most of binding residues are correctly predicted.. In the case of 1b3t, EL_PSSM-RT achieves 90.02% on ACC, 0.60 on MCC, 79.17% on SN, 91.35% and 85.25% on ST, respectively. In other words, 40 residues out of 48 binding residues are correctly predicted and only 32 residues out of 244 non-binding residues are incorrectly predicted as binding residues. Figure 4.7C and Figure 4.7D depict the actual binding regions and predicted binding regions on 1b3t, respectively. We can see that most of the actual binding residues overlap with the predicted binding residues and only very few non-binding residues are wrongly identified as the binding residues.

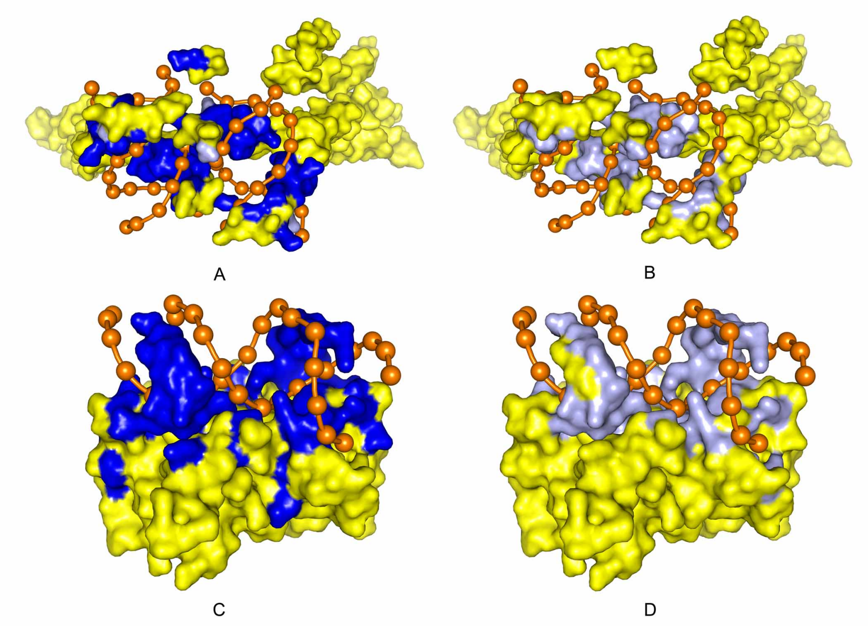


Figure 4.7: Actual residues and predicted residues of proteins in 1s40 and 1b3t.

In summary, our proposed EL_PSSM-RT can extract pairwise relationships between residues from sequence for prediction. Evaluation on the three datasets shows that EL_PSSM-RT outperforms state-of-the-art methods significantly. This shows that pairwise relationships of residues indeed play an important role for DNA binding residue prediction. However, EL_PSSM-RT is limited to extract pair-wise relationships only. As the function and structure of a residues affect multiple neighboring residues, relationships among several residues may also play a role in the prediction.

4.3 CNNsite: Convolutional Neural Network based method

In order to extract relationships of multiple residues, a Convolutional Neural Network (CNN) based method, referred to as CNNsite, is proposed to extract relationships among multiple nucleotides.

4.3.1 CNNsite

In this section, we propose a novel method to identify important motif features from the sequences around the binding residues for DNA binding residue prediction based on CNN and then develop a neural network classifier, referred to as CNNsite, by combining the important motif features (MOT), the sequence features (SEQ) and the evolutionary features (EVO). The frame diagram of CNNsite is shown in Figure 4.8. CNNsite comprises four computational layers: the convolution layer, the rectification layer, the pooling layer and the neural network layer. In our prediction task, the first three layers can discover important motifs of the inputting residue-wise data instances and the last layer is used to get the prediction results. The convolution, rectification, and network layers have trainable motif detectors D, thresholds b, and weights W, respectively. For a residue-wise data instance S, CNNsite produces a real-valued score $f(S)$ for prediction by the following formula

$$f(S) = net_w(pool(rect_b(conv_D(S)))) \tag{4.17}$$

where $conv_D()$, $rect_b()$, $pool()$ and $net_W()$ denote the four layer in CNNsite, respectively. Note that in the last layer, three kinds of features are used as input: the motif features outputted by the pooling layer, the sequence features, and the evolutionary features.

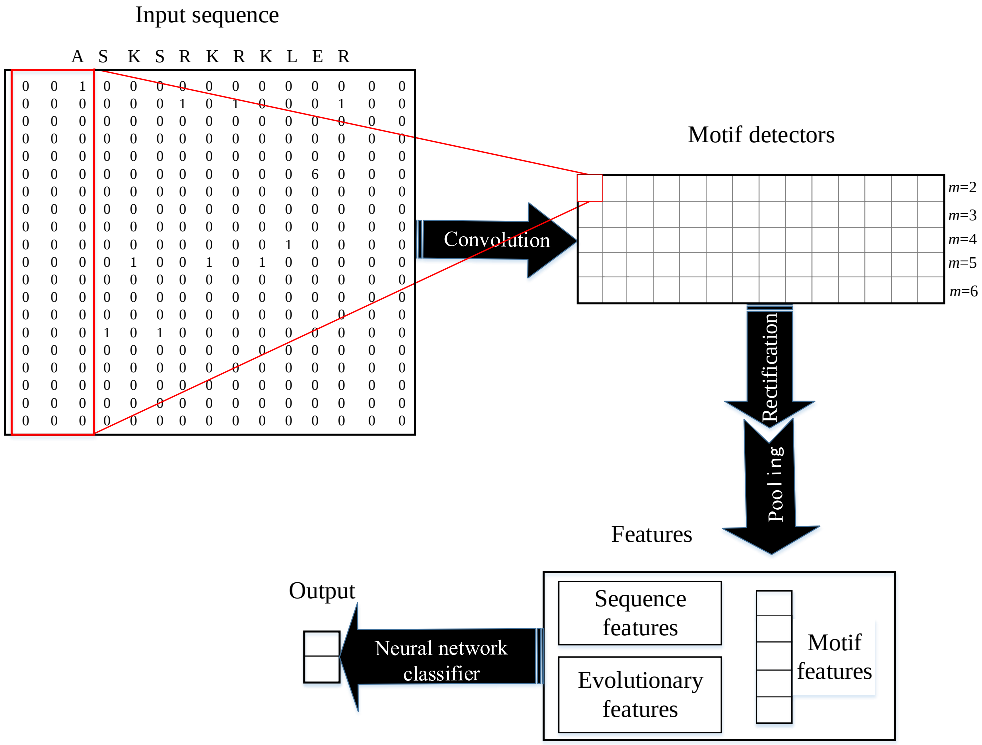


Figure 4.8: The frame diagram of CNNsite.

4.3.2 Experiments and Results

The purpose of the evaluation is to examine the effectiveness of the CNNsite for the prediction of DNA binding residue. Since CNNsite uses a window based approach, the window size needs to be set properly. Due to the length of this paper, we skipped the parameter tuning and all the results shown in this section use the window size $w = 11$ that is the context size is 5 on both the left and right side of the window. Four sets of evaluations are conducted. The first set evaluates the performance of CNNsite with different combinations of the three kinds of features on PDNA-62. The second set evaluates the performance of CNNsite with different combinations of the three kinds of features on PDNA-224. The third one uses the datasets PDNA-62 and PDNA-224 to compare our CNNsite with previous published predictors. And the last one evaluates CNNsite on an independent test TS-72 compared with previous published methods.

The predicted results on PDNA-62

This set of experiments examines the contributions of the three different kinds of features in CNNSite for the DNA binding residue prediction on PDNA-62. The performance is shown in Table 4.8, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.8: The prediction performance on PDNA-62 for various features by ten-fold cross-validation

Method	ACC(%)	MCC	SN(%)	SP(%)	ST (%)	AUC
SEQ	73.78	0.345	70.94	74.29	72.61	0.770
EVO	75.27	0.362	70.74	76.04	73.39	0.802
MOT	77.48	0.459	83.89	76.36	80.12	0.871
MOT+SEQ	78.15	0.473	85.25	76.92	81.09	0.889
MOT+EVO	78.57	0.476	84.81	77.48	81.15	0.897
ALL	**80.63**	**0.509**	**85.87**	**79.78**	**82.67**	**0.911**

As mentioned earlier, MCC, ST and AUC are the main metrics. Thus we shade the best performers of these three metrics for easy observation. It can be seen that the motif features achieve 0.459 for MCC, 80.12% for ST and 0.871 for AUC, outperforming the sequence features by 0.114 for MCC, 7.51% for ST and 0.101 for AUC and performs better than the evolutionary features with 0.097 for MCC, 6.73% for ST, 0.069 for AUC. It indicates that the motif features are more useful than the sequence features and the evolutionary features. When the motif features are combined with the sequence features, its performance is improved on all metrics with 0.014 for MCC, 0.97% for ST and 0.018 for AUC. When the motif features are combined with the evolutionary features, its performance is improved with 0.017 for MCC, 1.03% for ST and 0.026 for AUC. When the three kinds of features are combined, CNNsite achieves 0.509 for MCC, 82.67% for ST and 0.911 for AUC, outperforming other combinations with 0.033-0.164 for MCC, 1.52-10.06% for ST and 0.014-0.141 for AUC. Figure 4.9 also shows that the motif features gets better ROC curve than the sequence features and the evolutionary features and the combination of them gets the best ROC curve. It indicates that the motif features, the sequence features and the evolutionary features are complementary for each other.

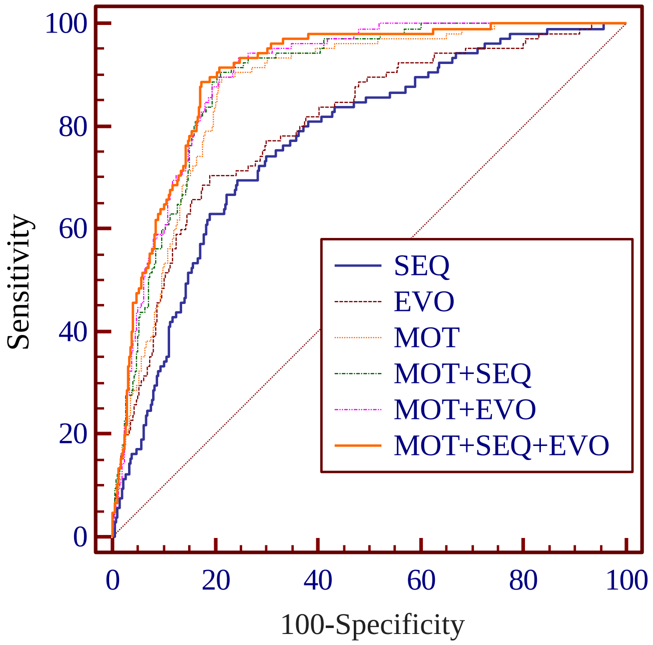


Figure 4.9: ROC curves of CNNsite with different combination of features on PDNA-62.

The predicted results on PDNA-224

This set of experiments examines the contributions of the three different kinds of features in CNNSite for the DNA- binding residue prediction on PDNA-224. To further evaluate the performance of our proposed method CNNsite in predicting DNA binding residues, we evaluate it on a recently proposed dataset PDNA-224. The results of CNNsite using various features are listed in Table 4.9, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.9: The predicting performance on PDNA-224 for various features by ten-fold cross-validation

Method	ACC(%)	MCC	SN(%)	SP(%)	ST (%)	AUC
SEQ	87.58	0.222	33.85	91.80	62.83	0.756
EVO	89.16	0.251	33.23	**93.35**	63.39	0.780
MOT	83.09	0.367	72.85	83.91	78.38	0.858
MOT+SEQ	82.85	0.382	76.63	83.34	79.99	0.869
MOT+EVO	82.40	0.381	**77.35**	82.79	80.07	0.872
ALL	**83.68**	**0.397**	77.12	84.19	**80.66**	**0.892**

The results show that the motif features achieve 0.367 for MCC, 78.38% for ST and 0.858 for AUC, performing better than the sequence features with 0.145 for MCC, 15.55% for ST and 0.102 for AUC and the evolutionary features with 0.116 for MCC, 14.99% for ST and 0.078 for AUC. When the motif features are combined with the sequence features, the performance increases by 0.02 for MCC, 1.61% for ST and 0.011 for AUC. When the motif features are combined with the evolutionary features, the performance increases by 0.014 for MCC, 1.69% for ST and 0.014 for AUC. On this dataset, the best result (MCC of 0.397, ST of 80.66% and MCC of 0.397) is obtained when the three kinds of features are combined. It performs better than other combinations with 0.016-0.175 MCC, 0.59-17.83% ST and 0.02-0.136 AUC. Although the combination of the motif features and the evolutionary features achieves higher value than the combination of the three kinds of features for SN, its SP value is lower than the latter. Figure 4.10 also shows that the motif features get better ROC curve than the sequence features and the evolutionary features and the combination of the three features get the best ROC curve. The results on this dataset also indicate that the motif features are more useful than the sequence features and the evolutionary features for the prediction of DNA binding residue and that these three kinds of features are complementary to each other in CNNsite.

Comparison with previous computational methods

This set of experiments evaluates the performance of our proposed CNNsite compared

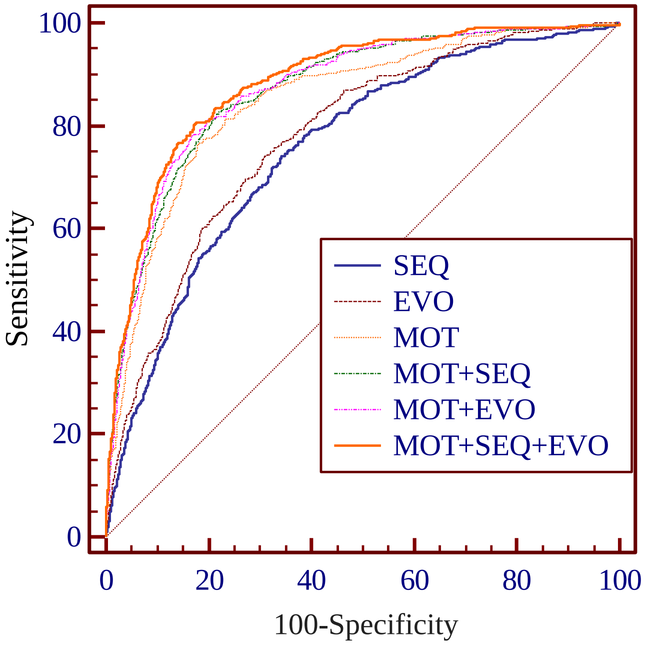


Figure 4.10: ROC curves of CNNsite with different combinations of features on PDNA-224.

with previous published methods which have been trained and tested either on PDNA-62 or PDNA-224. Many predicting algorithms including Dps-pred [3], Dbs-pssm [4], BindN [180], Dp-bind [85], Dp-Bind [66], BindN-RF [182], BindN+ [181] and PreDNA [91] have been proposed for the prediction of DNA binding residue, in which the former seven methods were trained and tested on PDNA-62 and the last one, PreDNA, trained and tested on both data sets. PreDNA [91] was developed by integrating a SVM classifier and a template-based prediction protocol. The SVM classifier was trained by sequence information, evolutionary information and structure information. The template-based prediction protocol is completed by aligning the structure of the current protein-DNA complex and that in template library. Since CNNsite do not use any structure features for prediction, to fairly compare the prediction performance of various methods, we only consider the PreDNA without using any structure features. The prediction performance of CNNsite and other methods on PDNA-62 and PDNA-224 are shown in Table 4.10 and Table 4.11, respectively, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.10: The predicting performance compared with other computational methods on PDNA-62

Method	ACC(%)	MCC	SN(%)	SP(%)	ST (%)	AUC
Dps-pred	79.10	–	40.30	81.80	61.10	–
Dbs-pssm	66.40	–	68.20	66.00	67.10	–
BindN	70.30	–	69.40	70.50	69.95	0.752
Dp-bind	78.10	0.490	79.20	77.20	78.20	–
DP-Bind	77.20	–	76.40	76.60	76.50	–
BindN-RF	78.20	–	78.10	78.20	78.15	0.861
BindN+	79.00	0.440	77.30	79.30	78.30	0.859
PreDNA	79.40	0.420	76.80	79.70	78.30	–
CNNsite	**80.63**	**0.509**	**85.87**	**79.78**	**82.67**	**0.911**

Table 4.11: The predicting performance compared with other computational methods on PDNA-224

Method	ACC(%)	MCC	SN(%)	SP(%)	ST (%)	AUC
PreDNA	79.10	0.290	69.50	79.80	74.60	–
CNNsite	**83.68**	**0.397**	**77.12**	**84.19**	**80.66**	**0.892**

As the performance of the existing methods is cited from the published papers, the values of some metrics are not known. Table 4.10 shows that BindN+ achieved the best performance (MCC of 0.440, ST of 78.30% and AUC of 0.859) on PDNA-62 among the previous published methods. Among all the prediction methods, CNNsite achieves the best performance (MCC of 0.509, ST of 82.67% and AUC of 0.911) outperforming the BindN+ on all the metrics with 0.069 on MCC, 4.37% on ST and 0.040 on AUC for PDNA-62. Table 4.11 shows that when testing on PDNA-224, CNNsite also achieves the best performance (MCC of 0.397, ST of 80.66% and AUC of 0.892) and performs better than PreDNA with 0.107 on MCC, 6.06% on ST for PDNA-224. By comparing the improvement of our proposed CNNsite over previous methods on PDNA-62 and that on PDNA-224, we observe that the improvement on PDNA-224 is higher than the improvement on PDNA-62. This phenomenon may be explained by the fact that the instances in PDNA-224 is more than that in PDNA-62 and CNN can make good use of the large number of training instances to improve its performance.

Independent test

This set of experiments evaluates the performance of CNNsite on an independent test TS-72. Since the performance on PDNA-62 and PDNA-224 are obtained by applying the ten-fold cross-validation and the test set and training set in the cross validation are drawn from the same population, the evaluating performances are not very persuasive. Moreover, there also exists some other predicting methods to be compared with our proposed CNNsite, which have not been evaluated on PDNA-62 and PDNA-224. Therefore, to evaluate CNNsite more objectively and compare it with the methods that have not been evaluated on PDNA-62 and PDNA-224, we conduct an experiment on an independent dataset TS-72. TS-72 is an independent dataset that was proposed by Ma et al. [104] to compare the performance of DNABR with that of three other predictors including BindN [180], BindN-RF [182] and BindN+ [181]. DNABR is a sequence based DNA binding residue prediction method and BindN, BindN-RF and BindN+ are three methods proposed by using only sequence information. In this work, we use TS-72 to compare the performance of CNNsite with these four predictors. The AUC values of CNNsite, DNABR, BindN, BindN-RF, and BindN+ for TS-72 are 0.878, 0.866, 0.748, 825 and 0.844, respectively, where the AUC values of those four previous methods are reported in Ma et al.'work [104]. In summary, our method increases the performance by 0.012-0.130 on AUC for TS-72.

4.3.3 Further analysis of the important motif features

The evaluation on PDNA-62 and PDNA-224 shows that the motif features captured by CNNsite perform better than the sequence features and the evolutionary features, indicating that the motif features are more effective for DNA binding residue prediction than the sequence features and the evolutionary features. In this section, we analyze discriminant powers of the motif features in CNNsite and give an explanation for the effectiveness of motif features for the prediction of DNA binding residue. In the convolution layer of CNNsite, the raw input is convolved with many motif detectors. In CNNsite, 5 sets of motif detectors of length from 2 to 6 are used and every set contains 500 motif detectors. After CNNsite is trained by PDNA-62, the discriminant power of a motif t in CNNsite is calculated by following formula

$$DP(t) = \sum_{i=1}^{p} \sum_{j=1}^{d} f_{i,j}(t) \tag{4.18}$$

$$f_{i,j}(t) = \begin{cases} Z_j & \text{if } argmax(Y_{1,j}, \cdots, Y_{n,j}) = the\, position\, of\, t\, in I_i \\ 0 & others \end{cases} \tag{4.19}$$

where p is the number of positive instances in PDNA-62, d is the number of motif detectors of the same length as motif t, I_i is a positive instance in the PDNA-62, and Z_j is the feature value of motif m in instance (for more entails about Z_j, refer to formula 4.23). The 15 top motif features with the largest discriminant power are shown in Table 4.12. For the motif features of 2 residues, TALBE VI shows that KR, GR, GN, GK, NR, EK, KT, RN, RT and KG are the top ten motif features. We find that the residues R, K, G are the important compositions of these motifs. This finding is consistent with Szilágyi and Skolnick's study [52], in which they found that R, A, G, K and D are important for the formation of protein-DNA interactions. The importance of R for the formation of protein-DNA interactions is further confirmed by Sieber and Allemann's work [150] which states that R can indirectly interact with DNA by interacting with both the phosphate backbone and the carboxylate of E(345). Since these residues are important for the formation of protein-DNA interactions, we speculate that they often occur in the context of DNA binding residues

Table 4.12: The top 15 motif features of various length with the largest discriminant power

Length	2	3	4	5	6
1	KR	RNR	KNWV	NRRRK	SNRRRK
2	GR	RMR	WVSN	KGNRS	KGRRGR
3	GN	RGR	CKGF	TRGRV	VSNRRR
4	GK	RLP	KGFF	GRRGR	VSRGRT
5	NR	RKR	GHRF	TRKRK	TTRKRK
6	EK	KTR	HSPA	RGHRF	KKRRKT
7	KT	HSP	VSNR	KRVRG	GIGNIT
8	RN	LKG	YRPG	VSNRR	YKGNRS
9	RT	TRK	KTRK	SNRRR	KSIGRI
10	KG	ALR	IKNW	RGRVK	MKRVRG
11	GT	IQI	FGKM	KGRRG	RKSIGR
12	IS	DSL	SIGR	KTRGR	GSGNTT
13	DK	RKT	FMKR	RVRGS	NKRMRS
14	TR	MRN	KRMR	KRMRS	SKTRKT
15	SR	RKE	RGHR	SRGRT	KTRGRV

and their occurrences are important features for prediction. In the 15 top motif features of more than 2 residues with the largest discriminant power, most of them also contain these residues with high proportions. Motif features of 3 residues contain RNR, RMR, RGR, RKR and KTR, motif features of 4 residues contain CKGF, GHRF, FMKR, KRMR and RGHR, motif features of 5 residues contain NRRRK, KGNRS, GRRGR, TRKRK and SNRRRK, and motif features of 6 contain SNRRRK, KGRRGR, VSNRRR, VSRGRT and KKRRKT. It can be seen that the proportions of R, K and G in all these motif features are very high. The discriminant powers of all motif features of number residues from 2 to 6 is listed in the support information S1, which is an attached support information file of this paper and can downloaded from our website. The proportions of R, A, G, K and D in the top 15 motif features with the largest discriminant power are shown in Table 4.13. It can be seen that motif features of 5 residues get the highest proportion (78.67%) of the important residues, indicating that the motif features of 5 residues are more useful for DNA binding residue prediction than other motif features. By observing the proportions of the five important residues separately, we found that the proportion of R is higher than that of

Table 4.13: The proposition of R,A,G,K and D in the top 15 motif features of various residues with the largest discriminant power

length	2(%)	3(%)	4(%)	5(%)	6(%)	sum(%)
R	23.33	33.33	16.67	42.67	28.89	144.89
K	20.00	13.33	15.00	12.00	16.67	77.00
G	16.67	4.44	11.67	16.00	13.33	62.11
D	3.33	2.22	0.00	0.00	0.00	5.55
A	0.00	2.22	1.67	0.00	0.00	3.89
Others	36.67	44.44	55.00	29.33	41.11	206.56

other four important residues in all motifs features, it indicates that R is important for the formation of DNA binding residues in protein chains, which is consistent with the findings in Sieber and Allemann's work [150].

In summary, our proposed CNNsite can extract relationships of multiple nucleotides by applying CNN on a DNA sequence. The evaluation on three datasets shows that CNNsite outperforms state-of-the-art methods. This shows that relationships of multiple nucleotides indeed play an important role in prediction. Both EL_PSSM-RT and CNNsite can extract relationships of nucleotides in short range.

4.4 EL_LSTM: Long Short-Term Memory Based Method

In order to extract relationships between residues that have long-range dependency, we propose a method, referred to as EL_LSTM, to use the LSTM method to extract relationships of both short range and long range through the gate mechanism to capture long distance relationships.

4.4.1 EL_LSTM

Generally speaking, every residue is encoded by sequence features, evolutionary features and structure features. The pairwise relationship between two residues can be represented by bi-grams of these three basic types of features, referred to as feature bi-grams. There are 9 different types of bi-grams formed by the combination of the three basic types of features as shown in Figure 4.11. Let S, E, and T denote sequence feature, evolutionary

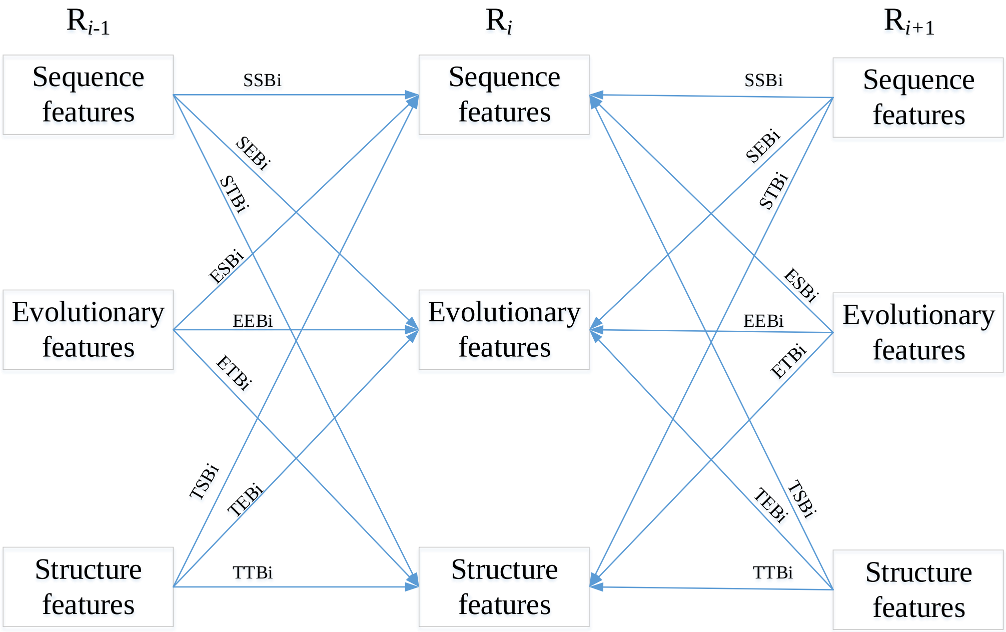


Figure 4.11: The diagram of bi-grams formed by the three basic types of features.

feature, and structure feature, respectively. The nine different bigram types are denoted by: (1) SSBi, (2) SEBi, (3) STBi, (4) EEBi, (5) ESBi, (6) ETBi, (7) TTBi, (8) TSBi and (9) TEBi. Since each target residue also one residue on its left as well as on its right, the set of bi-gram features include 18 pairs. The features used on the left side of the target residue is call left feature bi-grams and the ones on the right side are called right feature bi-grams. It is obvious from Figure 1 that extraction of bi-gram information requires large amount of computation time, which is only possible now.

4.4.2 Ensemble learning

Ensemble learning is now an active area of research in machine learning and pattern recognition. Ensemble learning first learns several base predictors from a training dataset and then combines the predictors into an ensemble predictor. Ensemble learning aims to take the advantage of learning ability of different base predictors suited for different data. There are three widely used ensemble strategies to train base predictors: trained by different data subsets, trained from different feature subsets and trained by different algorithms.

In DNA binding residue predictions, non-binding residues outnumber binding residues by a very large margin. For example, the ratios of non-binding residues to binding residues for our four datasets shown in Table 1 ranges from 5.50 to 15.18. In order to get a balanced dataset for training, many predictors remove the surplus non-binding residues [172]. However, the surplus non-binding residues do contain information and they do have the potential to improve prediction performance.

In order to make good use of the surplus non-binding residues, we develop a predictor, referred to as EL_LSTM, by incorporating ensemble learning into our base LSTM learning method. In EL_LSTM, a variant of the bootstrap aggregating (bagging) strategy [30] is used as the ensemble strategy to train multiple base LSTM classifiers. Since we have sufficient non-binding residues, the binding residues (let the size to be denoted by m) are first taken out from the dataset. The non-binding residues are divided into n subsets with each subset having roughly m number of residues sampled randomly without replacement. Finally the n new non-binding training datasets are formed by using the binding residues as positive samples as n training sets for n LSTM base classifiers.

The system architecture of EL_LSTM is shown in Figure 4.12. The Dataset Partition module in EL_LSTM first split the dataset as described above. Then, each dataset is fed into the Base Classifier Training module using the base LSTM described in Figure 2.1. For ensemble learning in the Base Classifier Selection module, a diversity based dynamic ranking and selecting method is used to select predictors with the largest diversity between each other to build the ensemble predictor.

Our dynamic ranking and selecting method, initially select a base LSTM at random. Then, in each iteration, all the unselected base predictors are ranked based on their diversity with the selected base predictor(s), and the one with the largest diversity will be added into the set of selected predictors. Diversity, as an indication of the difference between

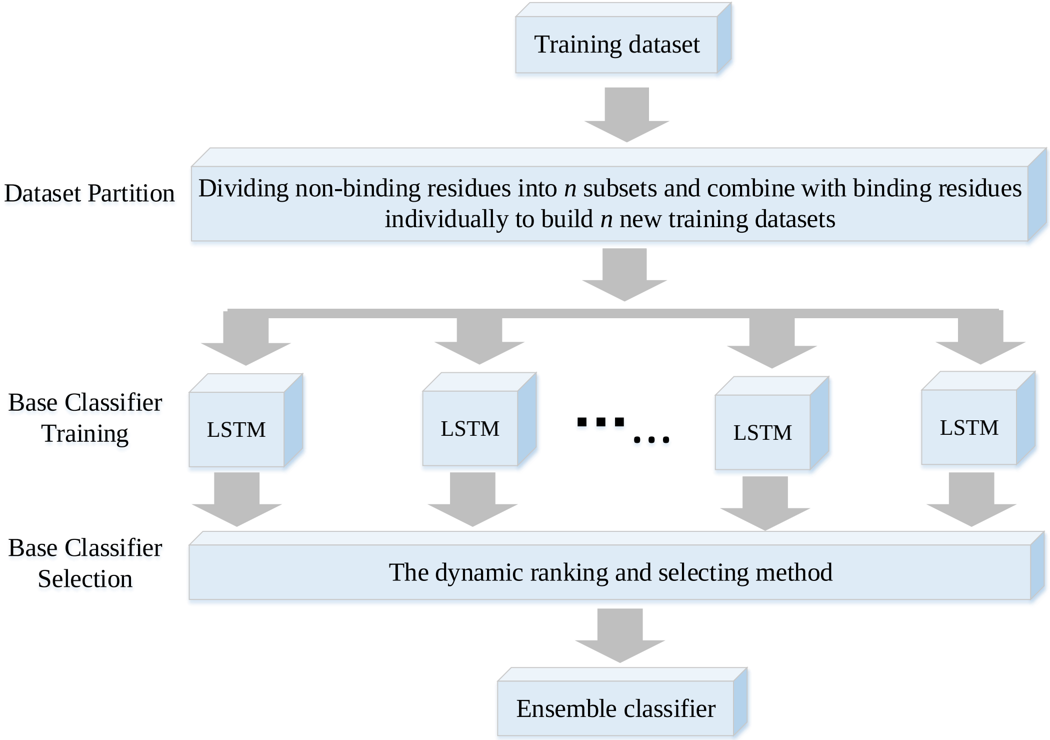

Training dataset
Dataset Partition
Dividing non-binding residues into n subsets and combine with binding residues individually to build n new training datasets
Base Classifier Training
LSTM
LSTM
LSTM
LSTM
Base Classifier Selection
The dynamic ranking and selecting method
Ensemble classifier

Figure 4.12: The framework diagram of EL_LSTM.

two base classifiers, is measured by the proportion of the number of samples with different labels from the two classifiers to the total number of samples in the validation dataset. Given a dataset with n samples and two classifiers f_1 and f_2, the diversity d between the two classifiers on the dataset is calculated as

$$d = \frac{\sum_{i=1}^{n} \mathbf{1}(f_1(x_i) = f_2(x_i))}{n} \tag{4.20}$$

where x_i is one of the sample in the dataset and $\mathbf{1}()$ is an indicator function. If $f_1(x_i)$ equals to $f_2(x_i)$, the value of the indicator function is 1; Otherwise, the value of the function is 0. The iteration is terminated when the addition of diversity no more than a specified criterion. Finally, the selected base predictors are combined to construct an ensemble predictor using a simple majority vote strategy.

4.4.3 Experiments and Results

Performance evaluations serve for two purposes. The first purpose is to measure the effectiveness of LSTM in terms of pairwise relationships extraction reflected in the ability of LSTM to improve performance compared to other methods where pairwise relationships are not used. The second purpose is to examine the effectiveness of EL_LSTM for the prediction of DNA binding residues. Four sets of evaluations are conducted here. The first set compares LSTM with traditional classifiers. The second set compares the ensemble classifier with base classifiers. The third set compares our proposed EL_LSTM with previous predictors on the two benchmarking datasets, and the fourth set evaluates EL_LSTM on the two independent datasets.

Comparison to classifiers not using pairwise relations

The first set of experiments aims to compare the performance of LSTM to other classifiers not using pairwise relationships. In this experiment, we consider three predictors including neural network (NN) classifier, random forest (RF) classifier and SVM classifier. These three classifiers are fed with the same set of residue-wise instances. Note that the difference between LSTM and NN is only because LSTM has an additional hidden layer to capture the pairwise relationships. Also, RF and SVM classifiers have no ability to capture pairwise relationships. The performances of the three classifiers and LSTM on the two benchmark datasets PDNA-224 and DBP-123 are shown in Table 4.14 and Table 4.15, respectively, where the best performers and the second best performers are marked by bold and underscore, respectively.

The performance on PDNA-224 given in Table 4.14 shows that LSTM performs better than the other three classifiers by 0.025-0.061 on MCC, 1.72-4.27% on ST and 0.024-0.11 on AUC in PDNA-224. Table 4.15 shows that LSTM outperforms the other three classifiers by 0.028-0.075 on MCC, 1.18-4.37% on ST and 0.012-0.098 on AUC in DBP-123.The improvements on both PDNA-224 and DBP-123 are very significant with p-value

Table 4.14: Performance of classifiers on PDNA-224 by ten-fold cross-validation

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
NN	72.34	0.261	74.24	72.19	73.22	0.751
RF	75.27	0.299	77.01	75.14	76.08	0.835
SVM	74.98	0.295	76.61	74.86	75.73	0.837
LSTM	**78.36**	**0.356**	**81.29**	**78.48**	**79.89**	**0.861**

Table 4.15: Performance of classifiers on DBP-123 by ten-fold cross-validation

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
NN	76.36	0.352	74.74	76.55	75.64	0.770
RF	77.29	0.386	79.28	77.06	78.17	0.844
SVM	78.34	0.399	**79.43**	78.22	78.83	0.856
LSTM	**80.51**	**0.427**	79.34	**80.67**	**80.01**	**0.868**

less than 1.51e-9. Note that LSTM and the other three classifiers are fed with the same features, but LSTM can learn feature bi-grams between residues and then learn the feature vectors for all residues in each residue-wise data instance using the learned feature bi-grams. The superiority of LSTM indicates that the feature bi-grams between residues are important and indeed useful for the prediction of DNA binding residues.

Comparison between LSTM and EL_LSTM

The second set of experiments compares the performance of LSTM and EL_LSTM on PDNA-224 and DBP-123. The performance of LSTM and EL_LSTM on PDNA-224 and DBP-123 are shown in Table 4.16 which shows that EL_LSTM outperforms LSTM by 0.045 on MCC, 1.89% on ST and 0.03 on AUC in PDNA-224. When evaluated on DBP-123, EL_LSTM outperforms LSTM by 0.021 on MCC, 1.32% on ST and 0.018 on AUC. Note that the improvements on both PDNA-224 and DBP-123 are very significant with p-value less than 5.23e-7. The superiority of EL_LSTM to LSTM indicates that ensemble learning can significantly improve performance by making better use non-binding residues. Figure 4.13(A) and Figure 4.13(B) show the ROC of EL_LSTM and LSTM on PDNA-224 and DBP-123, respectively. EL_LSTM obviously obtains better ROC than

LSTM on both PDNA-224 and DBP-123.

Table 4.16: Performances of LSTM and EL_LSTM by ten-fold cross-validation

Datasets	Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PDNA-224	LSTM	78.36	0.356	81.29	78.48	79.89	0.861
	EL_LSTM	82.59	0.401	80.26	83.18	81.72	0.891
DBP-123	LSTM	80.51	0.427	79.34	80.67	80.01	0.868
	EL_LSTM	81.44	0.448	81.18	81.48	81.33	0.886

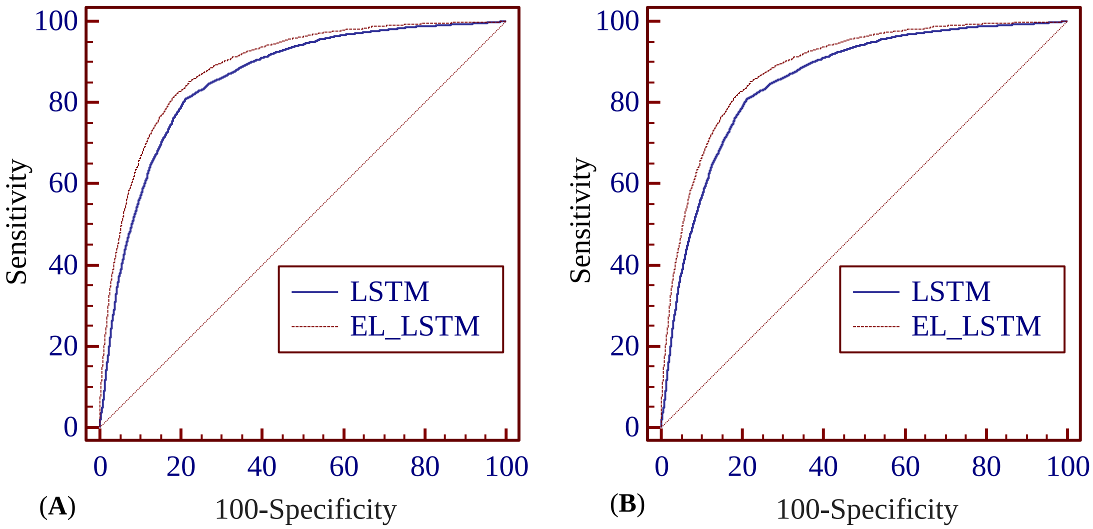


Figure 4.13: The ROC of LSTM and EL_LSTM on PDNA-224 and DBP-123 by ten-fold cross-validation.

Note that the smallest increment on AUC for PDNA-224 and DBP-123 are 0.03 and 0.018, respectively. In these two datasets, the ratios between the non-binding residues in PDNA-224 and that in DBP-123 is 1:3.58, which means that the non-binding residues in PDNA-224 are much more than that in DBP-123. The difference in performance improvement of EL_LSTM to LSTM on PDNA-224 and DBP-123 may be attributed to the difference in the number of non-binding residues in the two datasets. The additional non-binding residues in PDNA-224 can provide more diversity for the base classifiers, which is the key for prediction performance.

Comparison with previous prediction methods

The third set of experiments compares our proposed EL_LSTM with state-of-the-art predictors. We first compare EL_LSTM with PreDNA [28] on PDNA-224 by ten-fold cross-validation. PreDNA was proposed recently and it achieved the state-of-the-art performance for DNA binding residue prediction on PDNA-224 so far. PreDNA was developed by using all three types of common features. The performances of EL_LSTM and PreDNA on PDNA-224 given in Table 4.17 shows that EL_LSTM outperforms PreDNA by 0.051 on MCC and 2.52% on ST in PDNA-224, which is a significant improvement with p-value of 2.31e-9. Table 4.18 shows the comparison of EL_LSTM with two state-of-the-art predictors on DBP-123 by ten-fold cross-validation, where the best performers and the second best performers are marked by bold and underscore, respectively. The first predictor is the SVM classifier develop by Xiong et al. [192] which combines B-factor, packing density and several conventional features including PSSM, Relative solvent accessibility and side chain pKa values. The second predictor is a SVM classifier, called DNABind [98], trained by PSSM, relative solvent accessibility, depth index and protrusion index and topological features including degree, closeness, between-ness, and clustering coefficient. Table 4.18 shows that EL_LSTM outperforms Xiong et al.'s method [192] by 0.07 on MCC, 8.68% on ST and 0.08 on AUC and DNAbind by 0.016 on MCC, 5.05% on ST and 0.041 on AUC on DBP-123 with p-value of 6.14e-8. Although Xiong et al.'s method and DNABind both have higher specificity SP which measures the true negatives. However, their sensitivity SN which measures the true positives are much lower than ours. Since binding residue is what we are looking for, higher true spositives is more meaning full. In short, the superiority of EL_LSTM to the state-of-the-art predictors indicates that the EL_LSTM is a useful method for DNA binding residue predictions.

Performance on independent datasets: HOLO-83 and TS-61

The last set of experiments compares EL_LSTM with state-of-the-art methods on the two independent datasets HOLO-83 and TS-61. We first compare EL_LSTM with three predic-

Table 4.17: Comparison with previous prediction methods on PDNA-224 by ten-fold cross-validation

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PreDNA	81.80	0.350	76.10	82.20	79.20	-
EL_LSTM	**82.59**	**0.401**	**80.26**	**83.18**	**81.72**	**0.891**

Table 4.18: Comparison with previous prediction methods on DBP-123 by ten-fold cross-validation

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Xiong et al.	79.69	0.378	62.50	**82.81**	72.65	0.806
DNABind	80.76	0.432	69.80	82.76	76.28	0.845
EL_LSTM	**81.44**	**0.448**	**81.18**	81.48	**81.33**	**0.886**

tors including Xiong et al.' method, DNABind and DISPLAR on HOLO-83. In addition to the works of Xiong et al. and DNABind, introduced in the the last section, previous experiment, DISPLAR [166] is a neural network classifier using PSI-blast sequence profiles, solvent accessibilities and 14 closest neighboring residues. The performance of EL_LSTM and the other three predictors on HOLO-83 is shown in Table 4.19, where the best performers and the second best performers are marked by bold and underscore, respectively. Among the four predictors, EL_LSTM achieves the best performance. It outperforms the other three predictors by 0.009-0.062 on MCC, 3.37-7.23% on ST, and 0.012-0.051 on AUC. Again, the method by Xiong et al., DNABind and DISPLAR all have higher SP than EL_LSTM. But, their SN are much lower than EL_LSTM.

To compare EL_LSTM with DP-Bind [66] and DNABind [98] on TS-61, we use their web server to get their prediction results for TS-61. DP-Bind [66] is a web server for predicting DNA binding residues in a DNA binding protein from its amino acid sequence. The web server implements three machine learning methods: DP-Bind(SVM) that uses SVM, DP-Bind(KLR) that uses kernel logistic regression, and DP-Bind(PLR) that uses penalized logistic regression. DB-Bind [66] also implements two types of consensus methods. The first one uses majority vote on the results of three machine learning methods, referred to as DP-Bind(MAJ). The second one uses strict consensus that forces unanimous agreement. Since strict consensus is hard to achieve, DP-Bind using this consensus mode cannot provide prediction results for many of the residues. Therefore, in the performance evaluation on TS-61, we only show the result of the majority vote method. The performance of EL_LSTM and other methods on TS-61 is listed in Table 4.20, where the best performers and the second best performers are marked by bold and underscore, respectively. It shows that EL_LSTM outperforms DP-Bind by 0.15-0.16 on MCC, 4.86-5.51% on ST and 0.03-0.052 on AUC and outperforms DNABind by 0.12 on MCC, 8.48% on ST and 0.085 on AUC. This comparison indicates that EL_LSTM performs better than the predictors in DP-Bind [66] and DNABind [98] with a large margin on TS-61.

Because the samples in repetitive training datasets and testing datasets may be related, this may result in a biased estimation[8]. Therefore, the p-value for a comparison between two methods on a single dataset by ten-fold cross-validations may be biased. In order to validate the unbiasedness of our evaluation, we further use the independent test dataset TS-61 to calculate the p-values for comparisons by the Wilcoxon signed-ranks test[189]. As the Wilcoxon signed-ranks test requires a larger number of data sets to ensure test statistic (equation 23) to be distributed approximately normally [8], we divided TS-61 into 20 independent datasets, where every independent datasets contains 3 protein sequences except that the last one contains 4 sequences. Since the pairwise sequence similarity between the sequences in TS-61 is less than 25%s, the 20 independent datasets from TS-61 are unrelated from each other, which may provide unbiased estimation for p-value. Since DNABind outperforms Xiong et al.'s method on both the DBP-123 and HOLO-83, and its prediction results TS-61 can be easily obtained by its web sever while the results of Xiong et al.'s method can't be obtained, we analyze DNABind in the comparison. Additionally, we also analyzed DP-Bind which prediction results for TS-61 can be obtain by using its web sever. As DP-Bind(PLR) can obtain better performance other two settings on TS-61(as shown in Table 4.20), we use DP-Bind(PLR) to represent DP-Bind in this analysis. The AUCs of our method EL_LSTM, DP-Bind and DNABind on the 20 independent datasets are listed in Table 4.21, where the best performers and the second best performers are marked by bold and underscore, respectively. Table 4.21 shows that for the 20 independent datasets except the third and the 7th datasets, our method EL_LSTM can perform better then them. Both the p-values for the comparison with Dp-Bind and that with DNABind are 8.85e-5, which indicates that the outperformance for our method EL_LSTM over DP-Bind and DNABind are significant.

Table 4.19: Comparison with previous prediction methods on HOLO-83 by independent test

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Xiong et al.	80.27	0.358	58.60	83.89	71.25	0.800
DNABind	83.25	0.411	59.00	87.09	73.05	0.839
DISPLAR	**85.66**	0.396	46.10	**92.27**	69.19	-
EL_LSTM	79.19	**0.420**	**72.48**	80.35	**76.42**	**0.851**

Table 4.20: Comparison with previous prediction methods on TS-61 by independent test

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
DP-Bind(SVM)	75.90	0.26	65.99	76.70	71.34	0.794
DP-Bind(KLR)	76.45	0.25	64.22	77.45	70.83	0.790
DP-Bind(PLR)	75.46	0.25	65.24	76.29	70.76	0.812
DP-Bind(MAJ)	76.64	0.26	65.24	77.57	71.41	-
DNABind	**76.99**	0.29	54.79	**80.79**	67.79	0.757
EL_LSTM	76.09	**0.41**	**76.52**	76.02	**76.27**	**0.842**

4.4.4 Feature Analysis

The advantage of LSTM for DNA binding residue prediction is that it can extract the pairwise relationships between neighboring residues. As residues are encoded by sequence features, evolutionary features and structure features, the pairwise relationships between neighboring residues can be represented by 9 types of feature bi-grams formed by these three basic features. As weight matrices $\overrightarrow{\mathbf{V}^{*}}(\overrightarrow{\mathbf{V}^{\mathrm{i}}}, \overrightarrow{\mathbf{V}^{\mathrm{f}}}, \overrightarrow{\mathbf{V}^{\mathrm{o}}}, \overrightarrow{\mathbf{V}^{\mathrm{c}}})$ are four weight matrices in

Table 4.21: AUCs of different prediction methods on 20 independent datasets from TS-61

datasets	0	1	2	3	4	5	6	7	8	9
DP-Bind	0.775	0.750	0.753	0.728	0.808	0.793	0.842	0.785	0.725	0.761
DNABind	0.825	0.796	0.781	**0.784**	0.815	0.791	0.842	0.810	0.797	0.692
EL_LSTM	**0.918**	**0.914**	**0.837**	0.768	**0.878**	**0.847**	**0.810**	**0.879**	**0.856**	**0.862**
datasets	10	11	12	13	14	15	16	17	18	19
DP-Bind	0.661	0.748	0.713	0.645	0.756	0.786	**0.784**	0.787	0.751	0.746
DNABind	0.618	0.815	0.753	0.672	0.740	0.789	0.709	0.767	0.751	0.757
EL_LSTM	**0.824**	**0.847**	**0.788**	**0.711**	**0.806**	**0.873**	0.761	**0.856**	**0.783**	**0.835**

LSTM used for measuring left feature bi-grams, the weight matrix for left feature bi-grams , denoted by $\overrightarrow{\mathbf{V}}$, can be is calculated as follows:

$$\overrightarrow{\mathbf{V}} = sqr(\overrightarrow{\mathbf{V}^{\mathrm{i}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{i}}} + \overrightarrow{\mathbf{V}^{\mathrm{f}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{f}}} + \overrightarrow{\mathbf{V}^{\mathrm{o}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{o}}} + \overrightarrow{\mathbf{V}^{\mathrm{c}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{c}}}) \tag{4.21}$$

where $\odot$ is the element-wise multiplication of the two matrices and $sqr()$ is the root operation of every element in matrix. Similarly, the weight matrix for right feature bi-grams, denoted by $\overleftarrow{\mathbf{V}}$, can be calculated from the four weight matrices $\overleftarrow{\mathbf{V}^{*}}(\overleftarrow{\mathbf{V}^{\mathrm{i}}}, \overleftarrow{\mathbf{V}^{\mathrm{f}}}, \overleftarrow{\mathbf{V}^{\mathrm{o}}}, \overleftarrow{\mathbf{V}^{\mathrm{c}}})$.

$$\overrightarrow{\mathbf{V}} = sqr(\overrightarrow{\mathbf{V}^{\mathrm{i}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{i}}} + \overrightarrow{\mathbf{V}^{\mathrm{f}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{f}}} + \overrightarrow{\mathbf{V}^{\mathrm{o}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{o}}} + \overrightarrow{\mathbf{V}^{\mathrm{c}}} \odot \overrightarrow{\mathbf{V}^{\mathrm{c}}}) \tag{4.22}$$

where $\odot$ is the element-wise multiplication of the two matrices and $sqr()$ is the root operation of every elements in matrix. By observing $\overleftarrow{\mathbf{V}}$ and $\overrightarrow{\mathbf{V}}$, we can identify the feature bi-grams that are important for the DNA binding residue prediction.

Since we cannot find works from literatures to validate the feature bi-grams involved by structure features, we only analyze SSBi, EEBi, SEbi and ESbi. As sequence features are represented by the identities of the 20 residue types and evolutionary features are encoded by conservative scores of the 20 residue types, SSBi, EEBi, SEBi and ESBi can be represented as 400 types of bi-grams formed by the 20 residue types. The heat map for the weights of SSBi, EEBi, SEBi, ESBi are shown in Figure 4.14, Figure 4.15, Figure 4.16 and Figure 4.17, respectively.

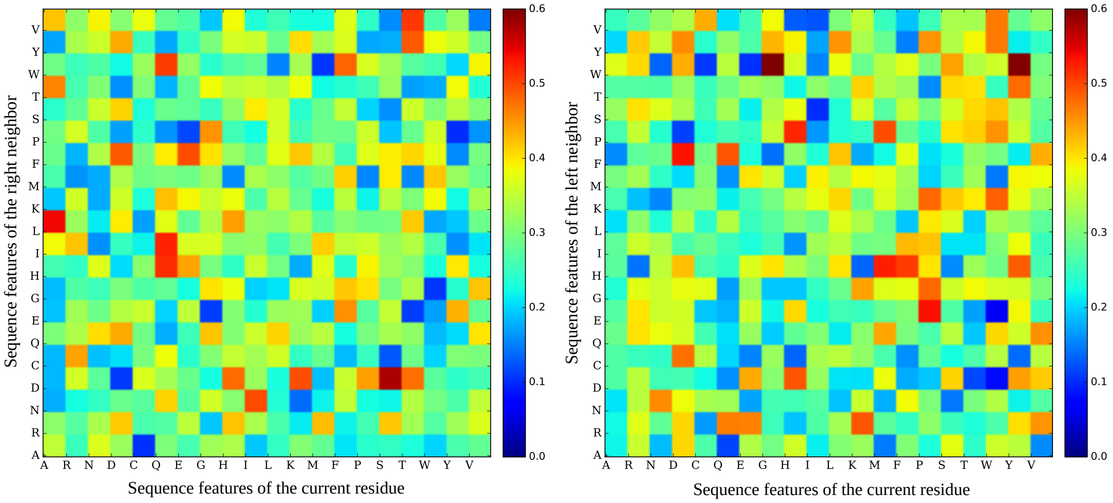


Figure 4.14: The heat map of SSBi.

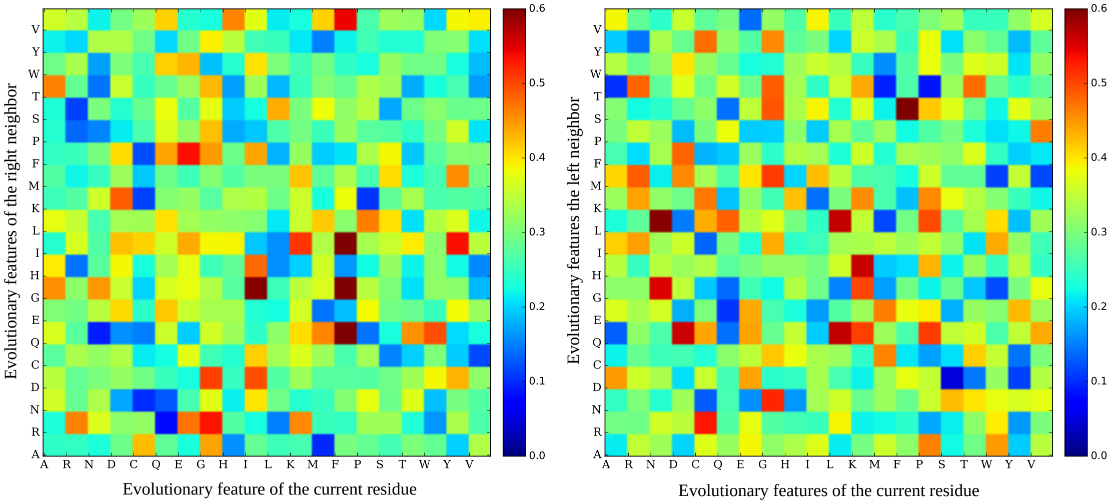


Figure 4.15: The heat map of EEBi.

In these four figures, the left subgraph and the right subgraph show the left feature bi-grams and the right feature bi-grams, respectively. By observing the colors of for the left feature bi-grams and for the corresponding right feature bi-grams, it is obvious that they are not symmetric. This validates the purpose of having separate feature representations for left context and right context. The bi-grams with larger weights for SSBi, EEBi, SEBi, ESBi are listed in Table 4.22.

By analyzing the bi-grams with large weights listed in Table 4.22, we can see that the

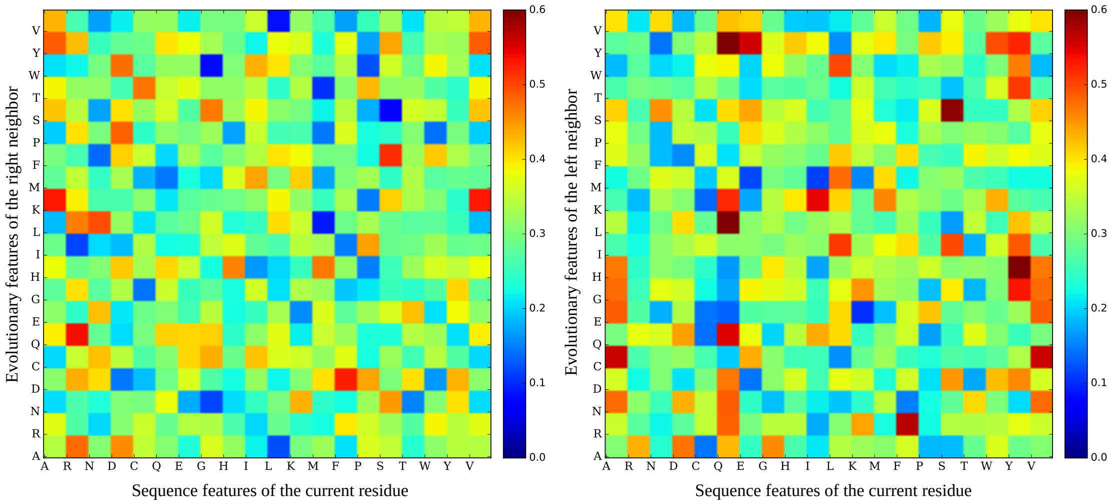


Figure 4.16: The heat map of SEBi.

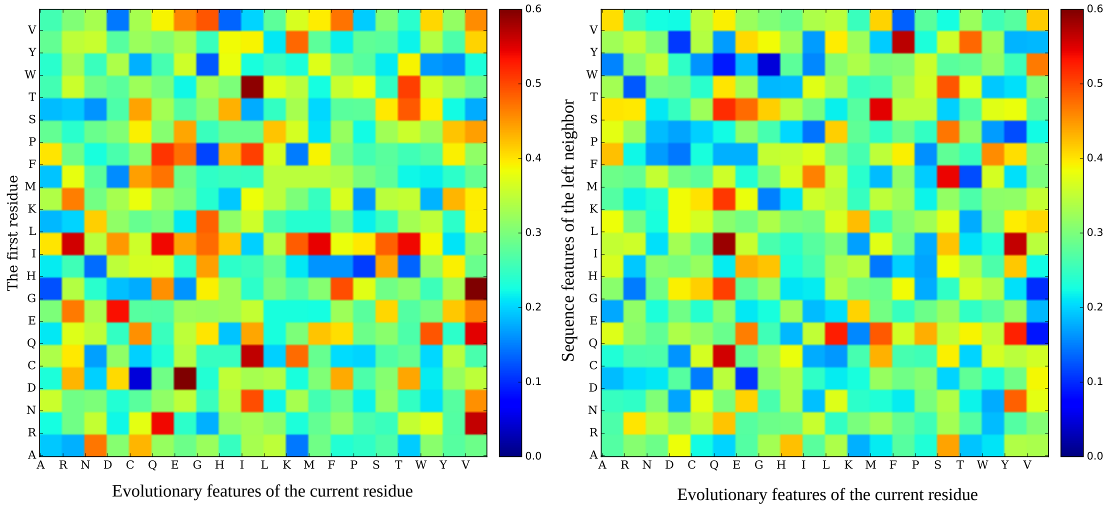


Figure 4.17: The heat map of ESBi.

interactions formed by residues R, D, G, A and K are important in the prediction of DNA binding residue. This finding is consistent with the study of Szilágyi and Skolnick [163], in which they found that R, D, G, A and K are important for the recognition between protein chains and DNA. The importance of R for the prediction of DNA binding residue is further confirmed by the work of Sieber and Allemann [150] which states that R can indirectly interact with DNA by interacting with both the phosphate backbone and the carboxylate of E(345). Table 4.22 also shows that KR, GK and KG are three of the feature bi-grams

Table 4.22: The bi-grams with larger weight for all types of bi-grams

Bi-grams types	Bi-grams with larger weight
Left SSBi	AL, AT, RC, MR, DF, HD, KD, SD, TD, PD
Right SSBi	GR, ER, KR, DC, DF, DY, HD, KG, PG, VR
Left EEBi	AG, AT, RR, ER, GR, KR, KI, IG, FG, DK
Right EEBi	PA, RM, DQ, DF, GN, GM, NG, KG, KH, KQ
Left SEBi	AK, AY, RA, RQ, RL, DP, DW, FD, GS, KN
Right SEBi	AC, AE, AG, AH, QR, FR, GA, YG, QK, IK
Left ESBi	NA, QR, VR, RE, RI, DE, ED, FG, VG, KI
Right ESBi	AF, HA, SA, QR, GH, GS, CG, QG, QK, CK

with the higher weights. It means that the combinations between K and R and between K and G are very important for the interaction between protein and DNA. This conclusion is consistent with the conclusion conducted by study of Ahmad et al. [3], in which they concluded that K and R can enhance R's ability to bind DNA and that the K residues within binding regions seem to favor G as their immediate neighbor on both sides. In addition to R, D, G, A and K, the four figures also show that the feature bi-grams involving Q also has very high weights. For example, for SEBi and ESBi, both the left bi-grams and the right bi-grams involving Q have very hight weights. This indicates that Q is useful for the prediction of DNA binding residues. Consequently, we hypothesize that residue type Q is also important for the interaction between proteins and DNAs. Our feature works will further investigate the importance of residue type Q for the interaction between proteins and DNAs through computational methods and experimental methods.

In summary, EL_LSTM can extract both local context and long-distance dependency for prediction. Evaluation on four datasets indicates that both local context and long-distance dependency play important roles in the prediction. As long-distance dependency is usually composed of residues in short spatial distance, it is not surprising to see that some long-distance dependencies extracted by EL_LSTM are in neighboring residues spatially as long as they are also sequential neighbors. However, EL_LSTM only extract long distance relationships with the maximum sequential distance of 11. How to extract relationships with very short spatial distance yet with very long sequence distance is the focus of our next study.

4.5 PDNAsite: Spatial and Sequence Context based method

In order to extract the relationships between residues with very short spatial distance yet with very long sequence distance, we propose a novel method referred to as PDNAsite. In PDNAsite, two sliding windows are used to capture relationships of sequence neighbors and spatial neighbors, respectively. these two types of neighbor residues are then used jointly to learn both sequence context and spatial context for prediction.

4.5.1 Contextual feature extraction

In the study of DNA binding site prediction, the residue-wise data instances derived from sequence were used as samples to train and evaluate classifiers. In order to make the full use of the sequence context for a target residue, a sequence sliding window of size w(being an odd number) is used. Then, a residue-wise data instance was commonly defined as a fragment with w consecutive amino acids with the target residue positioned in the middle and $(w-1)/2$ neighboring residues on either side. The residues contained in the sequence sliding window provide the sequence context information for the target residue. However, research results in many literatures [166, 20, 21] have indicated that the spatial context can also contribute to the identification of DNA binding site from non-binding sites. In order to extract the spatial context of a target site for its prediction, we propose a spatial sliding window with size m. The spatial sliding window is defined as a sphere with the target site positioned at the center and $(m-1)$ sites with the shortest spatial distance to the target site contained in it. The distance between sites is calculated based on the coordinates of their C α atoms.

For a target site, the sites contained in the spatial sliding window are referred to as the spatial context, while the sites contained in the sequence sliding window are referred to as the sequence context. As some residues in the sequence context may also be within the cutoff spatial distance from the target site, there may be sites contained simultaneously by both the sequence context and the spatial context, referred to as the overlapping sites. Since these sites are closed to the target site within both the sequence distance and the spatial distance, they can have greater effect on the function of the target site. So when the sequence context and the spatial context are combined to extract features, the overlapping sites should be used twice.

Therefore, in this method, a residue-wise data instance is defined as the combination of the sequence context and the spatial context. As a result, a residue-wise data instance should contain $(m + w - 1)$ residues. A residue-wise data instance is labeled with 1 (positive) if the target residue is binding or -1 (negative) if the target residue is non-binding. As SVM classifiers only take numerical values for classification, the residue-wise data instances need to be encoded into feature vectors. In this method, the feature space of residue-wise data instances is constructed by extracting the sequence information and structure information from the spatial context and the sequence context, including local amino acid composition, evolutionary information in terms of PSSM, Solvent accessible surface area, secondary structure, net charge and B-factor, where the entails for Solvent accessible surface area, secondary structure, net charge and B-factor are described in the following text.

Solvent accessible surface area (ASA): the ASA of every residue in protein is calculated from DSSP36. Before encoding the ASAs of the target residue and its neighboring residues, the ASA is divided by the maximum ASA of the corresponding residue type to calculate its relative ASA (RASA). Then, for a data instance, the RASA values of the residues in the spatial context and the sequence context are encoded and added into feature vector. **Secondary structure**: Secondary structure assignments of all residues in the proteins are made with DSSP [73], which classify every residue as one of the nine types:

alpha helix (H), residue in isolated beta-bridge (B), extended strand participates in beta ladder (E), 3-helix (or 310 helix) (G), 5-helix (or pi-helix) (I), hydrogen-bonded turn (T), bend (S), loop (L) and irregular (no designation). In this paper, the 9 types of secondary structure are approximately combined into 3 types: helix (H), β-strand (E) and coil (C). The secondary structure of the target residue is encoded using mutually orthogonal binary vectors: (1,0,0) for helix, (0,1,0) for β-strand and (0,0,1) for coil. Additionally, the secondary structure compositions for the residues in the left sequence sliding window, the right sequence sliding window, the whole sequence sliding window and the spatial sliding window are added into the feature vector, respectively. The values in the structure composition denote the proportion of the number of residues with the corresponding secondary structure type over the total number. **Net charge of a residue**: Due to the negative ambience around the DNA, the charge reciprocality of a residue may play an important role in its binding to the partner DNA. Therefore, the net charge of a residue is used as a feature for classification. A charge of +1 is ascribed to Arg and Lys and -1 to Asp and Glu. His is specified a charge of +0.5 and all other residues are taken as neutral. The net charge of the sites in the sequence and spatial sliding windows are calculated. **B-factor of a residue**: The B-factor of protein crystal structure reflects the fluctuation of atoms about their average positions and provides important information about protein dynamics. The thermal motion is useful for analyzing the dynamic properties of proteins. Therefore, in this work, the B-factor of the Cα and that of the Cβ of the residues in the sequence and spatial windows were encoded. In addition, the sum of the B-factor of the Cα over the residues in spatial sliding window was also calculated.

4.5.2 Latent Semantic Analysis (LSA) and Support Vector Machine

LSA is a method for extracting and representing the contextual meaning of words by statistical computations. Latent Semantic Analysis (LSA) is suitable to remove redundancies in feature space. Recently, LSA has been successfully applied to many bioinformatics problems. For example, Dong and his coworkers [49] developed SVM classifiers for protein remote homology detection by applying the LSA operation. For this problem, the starting point of the LSA operation is the construction of a triplet-sequence matrix W with dimension $(M * N)$ which denotes the co-occurrences between triplets and protein sequences. Triplets denote the combinations of three amino acid types. In the triplet-sequence matrix W, each sequence is expressed as a column vector. However, this representation does not recognize the triplets with similar function in the sequence and the dimension is too large. To resolve these problems, singular value decomposition is used to process the triplet-sequence matrix W. Let K be the rank of W, W can be decomposed into three matrices:

$$W = USV^T \tag{4.23}$$

Where U is the left singular matrix with dimensions $(M * K)$, V is the right singular matrix with dimensions $(N * K)$, S is the $(K * K)$ diagonal matrix with singular values where . One can reduce the dimensions by deleting the smaller singular values in the diagonal matrix and ignore the corresponding columns of matrix U and rows of matrix V. Additionally, Liu et al. [94] further improved the prediction accuracy for protein remote homology detection by applying LSA on the top-n-gram-sequence matrix which denotes the co-occurrences between top-n-grams and protein sequences. Through the analysis of the three matrices (word-document matrix, the triplet-sequence matrix, and the top-n-gram-sequence matrix), we discovered that all these three matrices are constructed by features of the same type, for example, words or biological sequences. We speculate that LSA could only be suitable for processing the feature space constructed by features of the same type, such as words in text, triplet or top-n-gram in protein sequence. In this paper, we construct a feature-instance matrix W which denotes the co-occurrences between features and protein sequences. Since there is much redundant information between the

PSSM features, we need to apply LSA to decrease the redundant information. However, the features used to construct the matrix W do not belong to the same type. So matrix W cannot be processed by LSA directly. In this work, we take the sub space of W with dimension of $(20 * (w + m - 1))$ spanned by only the PSSM features, denoted by W'. Since W' contains features of the same, we can then use LSA.

SVM can be used to resolve both binary-labeled and multi-labeled classification problems. For a binary-labeled classification problem, SVM first maps the input feature space into a higher-dimensional space and then seeks an optimal hyperplane, which maximizes the separation margin between the two classes of training instances, to separate the positive instances from negative instances. As SVM can transform the input features of the instances from a low dimensional space to a higher dimensional space, it has superior generalization power for most classification problems. In this study, the LIBSVM software package available at https://www.csie.ntu.edu.tw/ cjlin/libsvm/ is used. The radial basis function (RBF) is taken as the kernel function. RBF is defined as

$$K(X_i, X_j) = exp(-\gamma \|X_i - X_j\|) \tag{4.24}$$

where γ is a training parameter. A smaller value makes the decision boundary smoother. Another parameter for SVM training is the regularization factor C, which controls the trade-off between low training error and large margin. The optimal value of the parameters and C are obtained by five-fold cross-validation in this work. The data sets used in this study have many more negative instances than positive instances, which will have a great impact on the prediction performance of classifiers. In order to deal with imbalanced data sets, ensemble learning is used. Ensemble learning first divides the negative instances into n folds with non-overlapping instances, where the number of instances in each fold is approximately equal to that of the positive instances. Then the negative instances in each fold and the positive instances are combined to form a new data set. Thus n new data sets are constructed. Finally, the n new data sets are used as training data sets to train n base classifiers, which are subsequently combined as an ensemble classifier for prediction. Five-fold cross-validation is a widely used validation method, where the data set is first divided into five folds with no overlapping instances, and each time one fold is used as the test set and the remaining four folds are taken as the training set. This process is repeated five times until all the instances in the original set are tested once. The average performance over five such runs is used as the final prediction performance. In this study, the performances of our method on the two data sets are evaluated by applying five-fold cross-validation.

4.5.3 Experiments and Results

Performance evaluations serve for two purposes. The first purpose is to measure the effectiveness of spatial sliding window in extracting spatial relationships between residues. The second purpose is to examine the effectiveness of PDNAsite for the prediction of DNA binding residues. Three sets of evaluations are conducted here. The first set compares the performance of sequence sliding window with spatial sliding window. The second set evaluates the application of LSA on feature-instance matrix. The third set compares our proposed PDNAsite with previous predictors on the three benchmarking datasets and two independent datasets.

Performance comparison between sequence sliding window with spatial sliding window

To evaluate the performance of PDNAsite and compare it with other existing predictors, we first analyze the impacts of the sequential sliding window size w and the spatial sliding window size m on the prediction performance of PDNAsite. The impacts of w and m on the prediction performance of PDNAsite on PDNA-62 by five-fold cross-validation are shown in Figure 4.18A and Figure 4.18B, respectively. As can be seen from Figure 4.18A, MCC value and ST value are initially on the rise until they reach their maximum value at around $w = 13$ and then slightly go down with the increasing value of w. Thus we choose $w = 13$ for PDNAsite. This value is used for w in subsequent analysis. From Figure 4.18B, we can see that both MCC and ST values go up as m increases and achieves their best values when $m = 15$. So in all the subsequent experiments, m is set to 15.

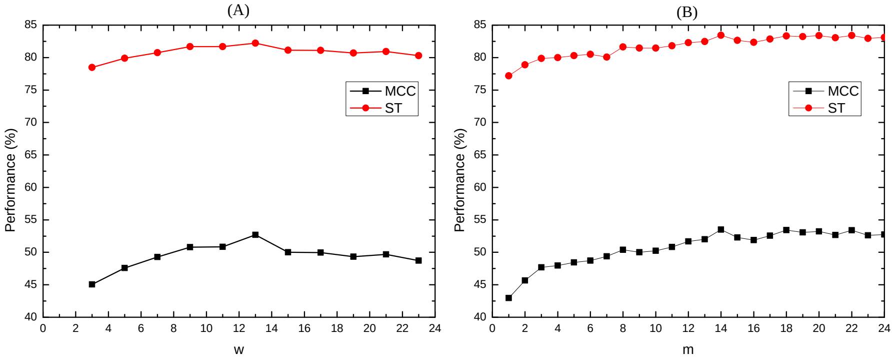


Figure 4.18: Impacts of window size w and m on prediction performance.

In this study, the features from the sequence context and the spatial context are used to construct the feature vector for each target site. In order to find out the contributions of the spatial context and the sequence context to the identification of DNA binding residue, we conduct performance evaluations using three sets of features: sequence context, spatial context and combined use of both. The performances of the predictors using different context on PDNA-62 and PDNA-224 are shown in Table 4.23 and Table 4.24, respectively, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.23: Performance comparison of predictors with different context on PDNA-62.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Sequence_context	83.48	0.527	80.40	84.03	82.22	0.893
Spatial_context	83.78	0.540	82.31	84.04	83.18	0.900
Both	**84.40**	**0.563**	**84.94**	**84.32**	**84.63**	**0.917**

As can be seen from Table 4.23 on PDNA-62, the predictor using spatial context achieved better performance than that using sequence context by 0.013 in terms of MCC,

Table 4.24: Performance comparison of predictors with different context on PDNA-224.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Sequence_context	79.19	0.346	78.52	79.24	78.88	0.868
Spatial_context	79.61	0.358	81.01	79.50	80.26	0.880
Both	**80.81**	**0.387**	**83.04**	**80.63**	**81.84**	**0.894**

0.96% in terms of ST and 0.007 in terms of AUC. The predictor using both of them achieved 0.563 MCC, 84.63% ST and 0.917 AUC, outperforming the one using sequence context alone by 0.036 MCC, 2.41% ST and 0.024 AUC with p-value less than 0.001 indicating the improvement is quite significant. As can be seen from Table 4.24 on PDNA-224, the predictor using spatial context achieved better performance than that using sequence context by 0.012 MCC, 1.38% ST and 0.012 AUC. The predictor using both of them outperformed the one using sequence context alone by 0.041 MCC, 2.96% ST and 0.026 AUC with p-value less than 0.001 indicating the improvement is quite significant. The ROC curves of the predictors using different context on PDNA-62 and PDNA-224 are shown in Figure 4.19 and Figure 4.20, respectively. The ROC curves of the predictors with different context also indicate that the spatial context gives more performance gain than the sequence context and the combination of them can further improve the performance.

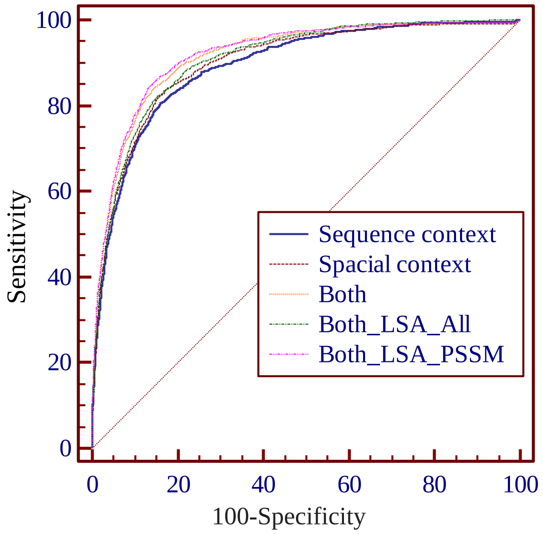


Figure 4.19: The ROC of PDNAsite with different sittings on PDNA-62.

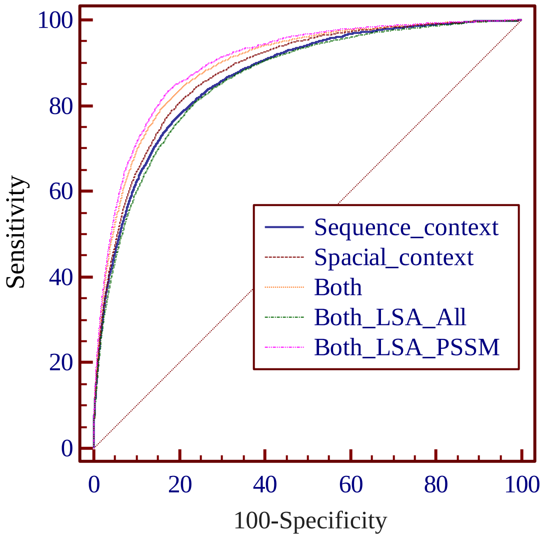


Figure 4.20: The ROC of PDNAsite with different sittings on PDNA-224.

Application of LSA on feature-instance matrix W

LSA is an efficient feature extraction technique widely used to remove noise information for a feature space. In this paper, we applied LSA in two different ways: one is applying LSA on the whole feature space, and the other is employing LSA on the sub feature space spanned by PSSM features. The prediction performances of the two ways on PDNA-62 and PDNA-224 by five-fold cross-validation are shown in Table 4.25 and Table 4.26, respectively, where the best performers and the second best performers are marked by bold and underscore, respectively.

Table 4.25: Performance comparison of predictor with different LSA on PDNA-62.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Both	84.40	0.563	84.94	84.32	84.63	0.917
Both_LSA_All	84.28	0.550	82.63	84.58	83.61	0.908
Both_LSA_PSSM	**85.11**	**0.582**	**86.27**	**84.91**	**85.59**	**0.928**

It can be observed that, on PDNA-62, the prediction performance decreased by 0.013 MCC, 1.02% ST and 0.009 AUC when LSA was applied on the whole feature space, while the prediction performance increased by 0.019 MCC, 0.96% ST and 0.005 AUC when

Table 4.26: Performance comparison of predictor with different LSA on PDNA-224.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Both	80.81	0.387	83.04	80.63	81.84	0.894
Both_LSA_ALL	78.54	0.338	78.24	78.57	78.41	0.860
Both_LSA_PSSM	**82.25**	**0.405**	**83.17**	**82.34**	**82.67**	**0.902**

LSA was applied on the sub feature space spanned by PSSM features. On PDNA-224, the prediction performance decreased by 0.049 MCC, 3.43% ST and 0.034 AUC when LSA was applied on the whole feature space, while the prediction performance increased by 0.018 MCC, 0.83% ST and 0.008 AUC when LSA was applied on the sub feature space spanned by PSSM features. The ROC curves of the two ways on PDNA-62 and PDNA-224 are shown in Figure 4.21 and Figure 4.22, respectively. The results shown in Figure 4.21 and Figure 4.22 indicate that LSA is not suitable to deal with the feature space constructed by features of different types and the application of LSA on the sub feature space spanned by PSSM is capable of improving the performance of PDNAsite.

Comparison with existing methods

DNA binding sites have been predicted successfully by many predictors. To demonstrate the discriminating power of our proposed PDNAsite, its prediction performance is compared with other existing state-of-the-art methods. As a meaningful comparison must be made on the same data sets, the following predictors which used the either of the two datasets are used as comparison including Dps-pred [3], Dbs-pssm [4], BindN [180], Dp-bind [85], Dp-Bind [66], BindN-RF [182], BindN+ [181] which used the first dataset, and PreDNA [91] which used both datasets. PreDNA [91] is the best-performing predictor reported so far. It integrated a machine learning model and a structural alignment model for prediction where the structural alignment model used the amino acid-nucleotide pairs with distance less than 16 Å as the alignment units. In each alignment unit, the distance between the amino acid and the nucleotide is calculated based on their coordinates in the 3D structure of the protein-DNA complex. However, in most cases, the binding sites and the non-binding sites in the training dataset and the test dataset are defined based on the distances between the sites and its neighboring nucleotides. As such, the binding site can be distinguished from the non-binding site based on the distance information directly. We argue that for training classifiers for DNA binding sites, the distance information between amino acid and nucleotide should not be used as features. Therefore, in order to fairly compare the performance of our proposed PDNAsite with the existing methods, we only consider the PreDNA without using the structural alignment model. The prediction accuracies of our method and the existing methods by five-fold cross-validation on PDNA-62 are shown in Table 4.27, where the best performers and the second best performers are marked by bold and underscore, respectively. As can be seen from the table, our method performs better than PreDNA by 0.082 MCC, 3.39% ST with p-value less than 0.001, indicating that not only PDNAsite is the best performer, the improvement is significant on PDNA-62. The comparison between our predictor and PreDNA on PDNA-224 by five-fold cross-validation is shown in Table 4.28, where the best performers and the second best performers are marked by bold and underscore, respectively. Our method outperforms PreDNA by 0.045 in MCC, 3.47% in ST and 0.010 in AUC with p-value less than 0.001, indicating significant performance improvement. DNABind [98] is a recently proposed predictor for DNA binding site prediction, which also used some spatial context as

Table 4.27: Comparison of PDNAsite with other existing methods on PDNA-62.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
Dps-pred	79.10	–	40.30	81.80	61.10	–
Dbs-pssm	66.40	–	68.20	66.00	67.10	–
BindN	70.30	–	69.40	70.50	69.95	0.752
Dp-bind	78.10	0.490	79.20	77.20	78.20	–
DP-Bind	77.20	–	76.40	76.60	76.50	–
BindN-RF	78.20	–	78.10	78.20	78.15	0.861
BindN+	79.00	0.440	77.30	79.30	78.30	0.859
PreDNA	83.06	0.500	80.20	84.10	82.20	–
PDNAsite	**85.11**	**0.582**	**86.27**	**84.91**	**85.59**	**0.928**

Table 4.28: Comparison of PDNAsite with PreDNA on PDNA224.

methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
PreDNA	81.80	0.350	76.10	82.20	79.20	0.892
PDNAsite	**82.25**	**0.405**	**83.17**	**82.34**	**82.67**	**0.902**

classification features, including degree, closeness and betweenness [97]. These features are calculated from the graph structure formed by the target site and its spatial neighboring sites. The features used in this paper include the amino acid composition, secondary structure, evolutionary information and physiochemical information contained in spatial context. In order to demonstrate the effectiveness of the spatial context proposed in this paper for the prediction of DNA binding site, we compared our predictor with DNABind [98] on DBP-123 and HOLO-83. As our predictor is only trained by DBP-123 without using any information in the template library used by DNABind [98], we just compared our predictor with the machine learning-based protocol in DNABind (DNABindML). The results of the two methods are shown in Table 4.29, where the best performers and the second best performers are marked by bold and underscore, respectively. It can be observed that our method outperforms DNABindMLwith 2.66% ST and 0.044 AUC with p-value less than 0.001 on DBP-123 and with 3.98% ST and 0.009 AUC on HOLO-83.

Table 4.29: Comparison of PDNAsite with DNABind on DBP-123 and HOLO-83.

Datasets	methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
DBP-123	DNABind	80.76	0.432	69.80	82.76	76.28	0.845
	PDNAsite	**84.56**	**0.506**	**71.02**	**86.86**	**78.94**	**0.889**
HOLO-83	DNABind	**83.25**	0.411	59.00	**87.09**	73.05	0.839
	PDNAsite	78.66	**0.439**	**74.59**	79.47	**77.03**	**0.848**

DNABR [104] is a sequence based DNA binding site prediction method, which performs better than the three methods proposed by Wang et al., including BindN [180], BindN-RF [182] and BindN+ [181]. To compare with DNABR, an independent test dataset TS-61 with 61 protein chains is applied. TS-61 was first proposed for evaluating the performance of DNABR [104] by extracting proteins-DNA complexes from PDB [17]. On the dataset with 3.5 Å as the distance threshold, results show that the AUC values are 0.8783, 0.8669, 0.7488, 8257, and 0.8445 for our method, DNABR, BindN, BindN-RF, and BindN+ method, respectively. For this evaluation our method, BindN, BindN-RF, and BindN+ are trained on the PDNA62 whereas DNABR is trained on a much larger dataset TR265 with 265 protein chains and the AUC values for the other four methods are referenced from Ma et al.'s work [104]. It indicates that our method performs better than DNABR and other three methods on TS-61.

4.5.4 Analysis spatial context and Case study

Different target sites generally have different spatial context. For example, some sites may only contain either non-binding sites or binding sites in their spatial context while other sites may have both of them. Figure 4.21A and Figure 4.21B show the sensitivity and specificity of the predictions for sites with different number of binding sites in their spatial context, respectively. In Figure 4.21A, the x-axis represents the number of binding sites contained in the spatial context and the y-axis represents the predicting sensitivity for the sites with certain number of binding sites in their spatial context. In Figure 4.21A, the x-axis has the same meaning as the one for Figure 4.21A and the y-axis denotes the predicting specificity for the sites with certain number of binding sites in their spatial context. From Figure 4.21A, we can see that the predicting sensitivity increases as the

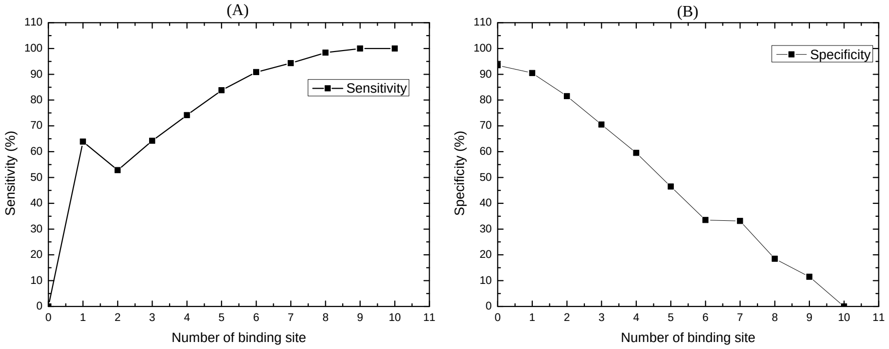


Figure 4.21: Analysis of number of binding sites in the spatial context.

number of binding sites in the spatial context increases to 10; and PDNAsite gets the maximal sensitivity when the number of binding sites in the spatial context equals to or greater than 10. From Figure 4.21B, we can see that the predicting specificity decreases as the number of binding sites in the spatial context increases to 10; and PDNAsite gets the maximal specificity when the number of binding sites in the spatial context equals to 0. This phenomenon indicates that, as the number of binding sites in the spatial context increases, the target site has more capacity to bind to its corresponding DNA molecule, meaning that the number of binding sites in the spatial context has a great impact on the prediction. Therefore, the spatial context extracted from the spatial sliding window can act as a very important discriminant feature for DNA binding site identification. We can also conclude that the interactions between neighboring binding sites in their spatial structure are important for protein-DNA recognition and their binding ability.

Epstein-Barr nuclear antigen 1 (PDB 1B3T) activates the initiation of DNA replication once every cell cycle from the Epstein-Barr virus (EBV) latent origin of DNA replication, oriP [24]. Nucleosome Core Particle (PDB 1KX5) is the greater part of nucleosome and comprises an octamer, containing a single histone H3-H4 tetramer and two histone H2A-H2B dimer, and 147 bp of DNA 45. 1B3T and 1KX5 are two typical protein-DNA complexes and they are not contained by the data sets PDNA-62 and PDNA-224. Moreover, the protein chains in these two complexes show low similarity with that in PDNA-62. So these two complexes are used as study cases for PDNAsite trained on PDNA-62. On complex 1B3T, PDNAsite achieves 86.16% ACC, 0.599 MCC, 96.00% SN, 84.91% SP and 90.45% ST. And on complex 1KX5, PDNAsite achieves 89.12% ACC, 0.600 MCC, 89.71% SN, 89.06% SP and 89.39% ST. The real DNA binding sites and predicted sites by PDNAsite for complex 1B3T and 1KX5 are shown in Figure 4.22. Figure 4.22A and Figure 4.22B denote the real sites and predicted sites of 1B3T, respectively. And Figure 4.22C and Figure 4.22D denote the real sites and predicted sites of 1KX5, respectively. From the figure, we can see that most of the real binding sites are covered by the predicted binding sites, indicating that most real binding sites were successfully predicted by PDNAsite. As there are much more non-binding sites than binding sites in a protein sequence, there are some false predicted non-binding sites shown in Figure 4.22B and Figure 4.22D.

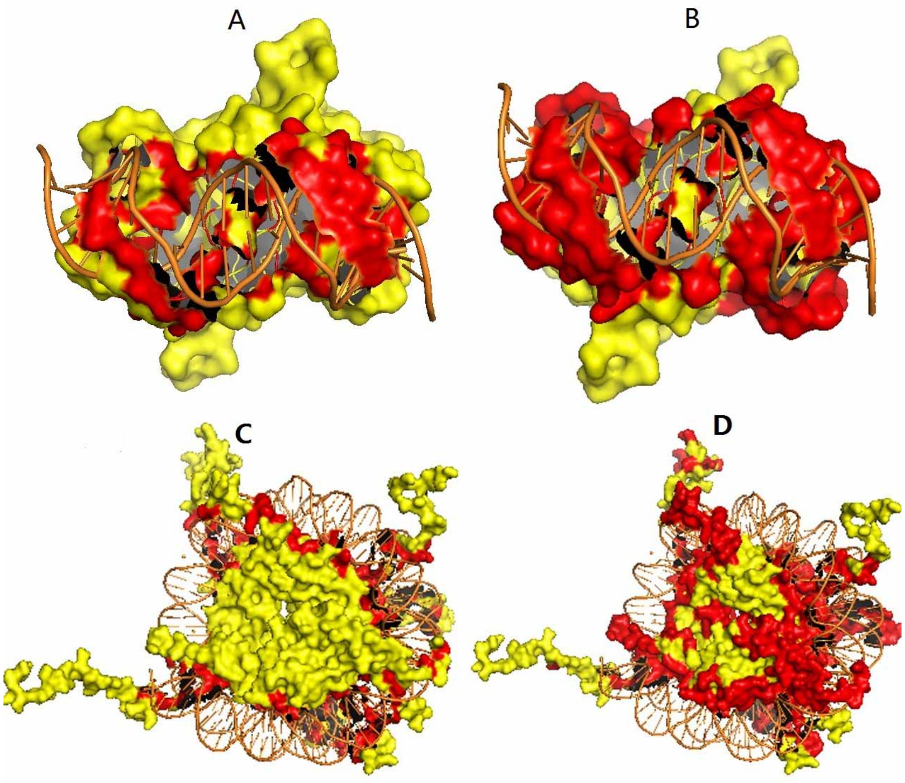


Figure 4.22: The Real sites and the predicted sites of 1B3T and 1KX5.

4.6 Comparison among our four methods

Four methods are proposed in this chapter for binding residue predictions. EL_PSSM-RT is designed to extract relationships between two residues. CNNsite aims to extract relationships of residues in more than two positions. EL_LSTM aims to extract both local context and long-range relationships. PDNAsite can extract both sequence relationships and spatial relationships between two residues. Comparison of the performance of the four methods are tabulated in Table 4.30, Table 4.31 and Table 4.32 for dataset PDNA-62, PDNA-224, and TS-61, respectively. From these three tables, we can see that PDNAsite and EL_LSTM are the top two performers compared to CNNsite and EL_PSSM-RT. This is because PDNAsite and EL_LSTM contain both sequence features and structure features whereas CNNsite and EL_PSS-RT only use sequence features. Thus we can conclude that structure features are indeed useful for DNA binding residue prediction.

Note that in the two top performers, PDNAsite outperforms EL_LSTM with a large margin. This indicates that spatial context captured by the spatial sliding window in PDNAsite is a more salient feature for DNA binding residue prediction. Even though structure features are important for DNA binding residue prediction, structure features are unavailable for most proteins. So PDNAsite and EL_LSTM can be applied to only a limited number of proteins. Between the two methods that use only sequential features, CNNsite outperforms EL_PSSM-RT on all the three datasets. This indicates that multi-residue relationships are more useful for DNA binding prediction than pairwise relationships. However, CNNsite is a deep learning model based method which requires a large number of samples to train while EL_PSSM-RT is much less demanding on the number of sample needed. Therefore, CNNsite is applicable to applications where the datasets are relatively large whereas EL_PSSM-RT is more useful if the training data is relatively small.

Table 4.30: Comparison of performances on PDNA-62.

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
EL_PSSM-RT	78.23	0.490	87.08	76.69	81.87	0.901
CNNsite	80.63	0.509	85.87	79.78	82.67	0.911
PDNAsite	**85.11**	**0.582**	**86.27**	**84.91**	**85.59**	**0.928**

Table 4.31: Comparison of performances on PDNA-224.

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
EL_PSSM-RT	78.93	0.330	75.57	79.20	77.39	0.853
CNNsite	83.68	0.397	77.12	**84.19**	80.66	0.892
EL_LSTM	82.59	0.401	80.26	83.18	81.72	0.891
PDNAsite	**82.25**	**0.405**	**83.17**	82.34	**82.67**	**0.902**

Table 4.32: Comparison of performances on TS-61.

Methods	ACC(%)	MCC	SN(%)	SP(%)	ST(%)	AUC
EL_PSSM-RT	77.33	0.310	73.64	77.73	75.19	0.839
CNNsite	77.37	0.320	74.79	77.61	76.05	0.840
EL_LSTM	76.09	0.410	76.52	76.02	76.27	0.842
PDNAsite	**78.72**	**0.430**	**78.65**	**78.87**	**78.76**	**0.878**

4.7 Chapter Summary

This chapter introduce four novel methods to include relationships of either pair-wise or multiple residues in residue prediction. Each method is focused on a different type of relationship. Both EL_PSSM-RT and CNNsite can extract relationships among residues in different positions, these relationships contain only local context with close distance. EL_LSTM not only can extract local context but also long-range dependency PDNAsite goes one step further to extract structure to include spatial relationships between neighbor residues in residue prediction. The four methods uses different machine learned methods suited for their respective purposes and they also have different application scopes. PDNAsite and EL_LSTM are applicable to proteins with available 3D structures. CNNsite are suitable for applications with large training samples. EL_PSSM-RT, as the simpliest method in this group is suitable for all known proteins as it only need to use sequential features and do not require large training set.

Chapter 5

TF binding site prediction

TF binding sites (TFBSs) is DNA fragments that can be bound by TFs. As TF-DNA interaction plays an important role in gene expression regulations, TFBS prediction is very useful for understanding transcriptional regulatory networks and fundamental cellular processes, such as growth control, cell-cycle progression and development, as well as differentiated cellular function [187, 52, 202].

Dependency between nucleotides are used by several methods, such as DWM [149], TFFM [110], Chromia [190], and DNA shape based methods [111]. However,current works only make use of first order dependencies. The term **first order dependency** refers to relationships between individual nucleotides. **Higher order dependency** refers to relationship between elements containing first order dependency. For example, the relationships between DNA fragments is a higher order dependency because a DNA fragment contains multiple first order dependencies.

Relationships between histone modification features contain higher order dependencies [190]. This is because Histone modification features are properties over DNA fragments (at least 25 bps). In this chapter, we propose to make use of relationships between histone modification features to extract higher order dependencies. Since several widely used classifiers for bioinformatics including support vector machine (SVM) [169], neural network (NN) [22], and random forest (RF) [31] intrinsically lack the ability to capture dependency between input features, we investigate the use of a convolutional neural network(CNN) on histone modification features to extract needed relationships as CNN is proven to be an efficient method for extracting context dependencies contained in a sequence [211, 212]. We first propose a novel method, referred to as **CNN_TF**, by applying CNNs [211, 212] on DNA sequence and histone modification features, respectively. CNN_TF incorporates both first order and higher order dependencies for prediction.

One issue in TF binding site prediction is that there are only limited TF training samples for a large number of cell-types. This is because TFBSs for TFs can only be identified by ChIP-Seq [67, 59, 79] or ChIP-chip [67, 59, 79, 139] which are experimental techniques, too time-consuming and too expensive to scale up.

Recent studies [190, 167, 83] have shown that the TFBSs of a TF are associated with histone modification types of several cell-types. Also, a TF often shares a common binding motif in multiple cell-types [112, 33].

Based on the results of these biological studies, we propose a novel TFBS prediction method, referred to as MTTFsite. MTTFsite uses a multi-task framework to learn common features from multiple cell-types with TF training samples using a so called **common CNN** as well as features of individual cell-types using a group of **private CNNs** for individual cell-types which have TF training samples. In MTTFsite, our proposed CNN_TF is used to build both the common CNN and a private CNN for each cell-type. MTTFsite is designed to predict TFBS for cell-types with insufficient training samples as it can leverage on training samples from other cell-types. Thus, it is referred to as a **cross-cell-type TFBS prediction method**.

As many target TFs do not have any training sample in any of the cell-types, our proposed MTTFsite cannot be applied to predict TFBSs for these TFs. Fortunately, we know that in a specific cell-type, there exist other TFs which have TFBSs identified by experimental methods. Even though a majority of TFs have different sequences and biology functions, some TFs do have similar sequences and biology functions. As these TFs are similar in sequences and biology functions and tend to bind to similar positions of the genome, we propose a novel method, referred to as PDBR_TF, to obtain features for those TFs without training data by using experimentally identified TFBSs of other TFs from the same cell-type. Thus, this method is also referred to as the **cross-TF TFBS prediction method**. In PDBR_TF, the predicted DNA binding residues by our proposed CNNsite are combined with DNA sequence and histone modifications to learn features by the network topology in CNN_TF.

5.1 Feature representation and evaluation Metrics

Most recent studies used the ChIP-seq experiments [61, 13] to identify TFBS. First, every nucleotide in a genome is provided with a signal value by ChIP-seq experiments. Then, a so called peak calling method [37, 206, 137] is used to identify peaks from the genome according to the provided signal values. The obtained peaks are usually provided in one of two formats. One is called the **narrow peak** and the other is called the **broad peak**. Both types of peaks provide chromosome, start position, end position and signal value. The narrow peak data, which requires technically more sophisticated equipment to get, can provide more accurate position for TFBS than the broad peak. However, some datasets are provide with only the broad peak format. In this work, the narrow peak format is used to define TFBSs whenever available. Otherwise, the broad peak format is used. As both the peak and its context are important for their function prediction, we define a **TFBS** as a DNA segment with 101 base pairs, where the midpoint of the peak is positioned in the middle and the equal number of neighbor nucleotides are positioned on either sides. For example, given a genome sequence G of length L ($L >> 101$) denoted as

$$G = N_1N_2N_3N_4N_5N_6 \cdots N_{i-1}N_iN_{i+1} \cdots N_L \tag{5.1}$$

where N_1 represents the first nucleotide of the genome sequence G, N_2 represents the second nucleotide and so forth. For a ChIP-seq peak with the midpoint at nucleotide i in

G, the TFBS can be represented as

$$T_i = N_{i-50}N_{i-49}\cdots N_{i-1}N_iN_{i+1}\cdots N_{i+49}N_{i+50} \tag{5.2}$$

ChIP-seq can be used to map global binding sites precisely for any protein of interest on the genome scale. Therefore, the labeled data in our study and the predicted TF-binding sites by our proposed methods already contain the TF-binding sites located in enhancers and that in other specific genetic regions, including promoters and insulators. Enhancers are short (50–1500 bp) regions of DNA that can be bound by proteins (activators) to increase the likelihood that transcription of particular genes will occur. These proteins are usually referred to as transcription factors. Enhancers can be located up to 1 Mbp away from the gene, upstream or downstream from the start site. There are hundreds of thousands of enhancers in the human genome. Promoters are regions of DNA that initiate transcription of a particular gene. Promoters are located near the transcription start sites of genes and can be about 100–1000 base pairs long. Insulators is a type of cis-regulatory element known as a long-range regulatory element and working over distances from the promoter. Insulators are typically 300 bp to 2000 bp in length and function either as an enhancer-blocker or a barrier, or both.

Both sequence features and histone modification features are very important features for TFBS. **Sequence features** of a TFBS are defined by all the nucleotides within the TFBS and they can be represented by concatenating one-hot vectors of these nucleotides. Sequence features are used to capture first order dependencies by CNN_TF. **Histone modification features** refer to the post-translational modification levels of histones in chromatin structure. Histone modification features are used to capture higher order dependencies by CNN_TF. With the advancement of technology, the mapping of histone modification features can be completed by ChIP-seq technology on a genome scale. However, as the ChIP-seq experiments for histone modification features mapping are costly and labor-intensive, many histone modification features are absent for mES cell. In this study, eight types of histone modification features are used for mES cell:*H3, H3K4me1, H3K4me2, H3K4me3, H3K9me3, H3K36me3, H3K20me3*, and *H3K27me3*. The ChIP-seq data for these histone modification features can be obtained from literature [118, 116]. For the 5 cell-types of humans, seven types of histone modification features are: *H3K4me2, H3K4me3, H4K20me1, H3K9ac, H3K27ac, H3K27me3* and *H3K36me3*. The ChIP-seq data for these histone modification features can be obtained from the work of [83]. In study of [190], 25-bp bin is used as a unit to measure histone modification features becasue the resolution for the ChIP-seq experiments is 25-bp. The histone modification features for each 25-bp bin are obtained by estimating the number of end-sequenced ChIP reads of corresponding histone modification marks that overlap the bin in a reference genome. And then, the histone modification features for each 100-bp bin are computed by averaging the histone modification features over the four 25-bp bins within a 100-bp bin. According to this method, we use the following scheme to apply histone modification features in CNN_TF: We first estimate the histone modification features for every 25-bp bin without overlap and then calculate the histone modification features for every 100-bp bin by averaging them over the four 25-bp bins within the 100-bp bin. Finally, the histone modification features for each TFBS are calculated by concatenating the histone modification features of the twenty 100-bp bins within the TFBS.

We evaluate our proposed method using the Area under Receiver Operating Characteristic(ROC) [162], curve (AUC) [29], and positive predictive values (PPV)[190]. ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier. It is drawn by plotting the true positive rates (i.e. sensitivity) against the false positive rates (i.e. 1-specificity) calculated by changing the classification threshold for predictors. AUC [29] is the area under the ROC curve and is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. An AUC of 1.0 and 0.5 indicate the best performance and a random performance, respectively. PPV [190] is another useful evaluation metric for TFBS prediction problem, which can be calculated by following the following formula

$$PPV = TP/(TP + FP), \tag{5.3}$$

where TP denotes the number of true positive samples and FP denotes the number of false positive samples.

5.2 CNN_TF: Higher order dependency based method

5.2.1 CNN based TF site prediction method (CNN_TF)

In this chapter, we propose a novel method, referred to as CNN_TF, to extract both first order dependency and higher order dependency by applying CNN on sequence features and histone modification features, respectively. The extracted first order dependencies andhigher order dependencies are incorporated into a neural network classifier.

The general framework of CNN_TF is shown in **Figure 5.1**. The CNN_TF model consists of two CNNs: one is used to extracting first order dependency from sequence features and the other is used to extracting higher order dependency from histone modification features. Both CNNs contain three computational layers: the convolution layer, the rectification layer, and the pooling layer. Feature vectors learned by the two CNNs are then concatenated and fed into a softmax classifier for prediction. CNN_TF includes three sets of parameters: (1) motif detectors F_S and thresholds b_S for sequence features S, (2) motif detectors F_C and thresholds b_C for histone modification features C, and (3) the weights W for the neural network classifier. For a TFBS T, CNN_TF provides a real-valued score $f(T)$ according to the following formula

$$\begin{aligned} f(T) = softmax_W(pool(rect_{b_S}(conv_{F_S}(S))) \\ \bigoplus pool(rect_{b_C}(conv_{F_C}(C)))), \end{aligned} \tag{5.4}$$

where $f(T)$ is defined by the softmax classifier through concatenation $\bigoplus$ of the two elements. The first element is the output from the CNN for sequence features, denoted by

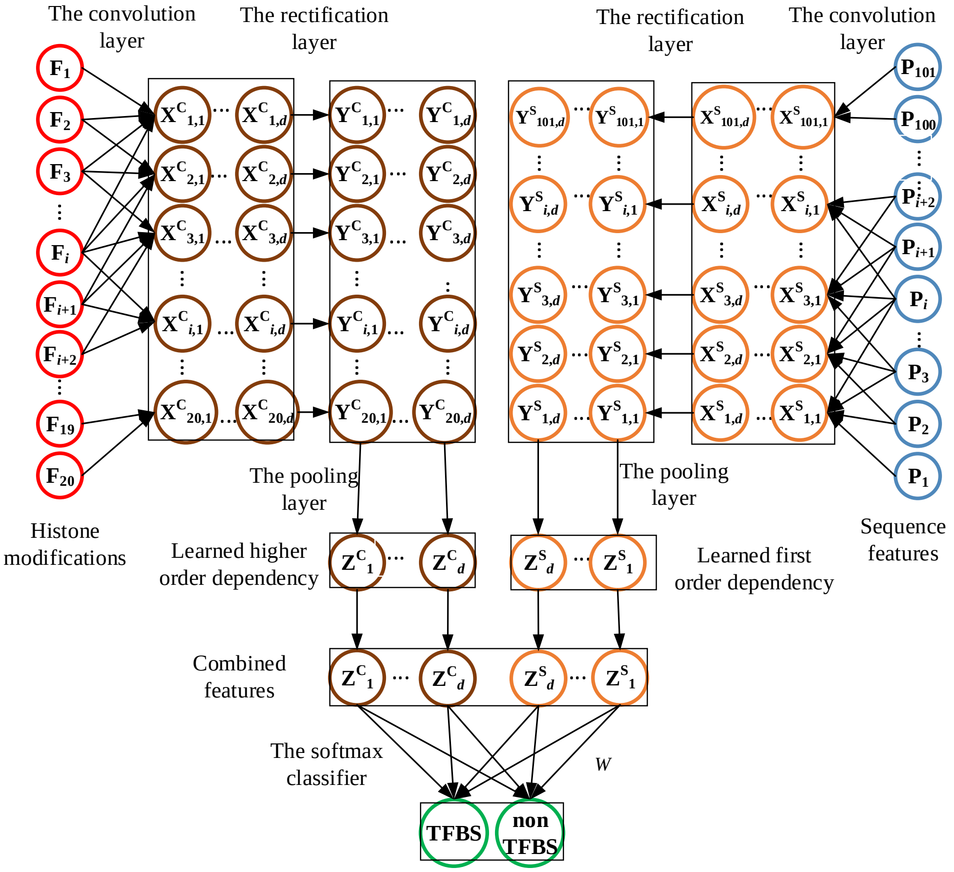


Figure 5.1: The schematic graph of the architecture of CNN_TF.

S with $conv_{F_S}()$, $rect_{b_S}$ and $pool()$ representing the three layers in CNN. Similarly the second element is the output from the CNN for histone modification features. $conv_{F_C}()$, $rect_{b_C}$ and $pool()$ denote the three layers in the CNN for histone modification features C. This real-valued softmax score is used for prediction.

5.2.2 Experiments and Results

Performance evaluations serve for two purposes. The first purpose is to measure the effectiveness of higher order dependency TFBS prediction. The second purpose is to examine the complementary between first order dependency and higher order dependency for TFBS prediction. Four sets of evaluations are conducted in the evaluation. The first set compares the performance between first order dependency and higher order dependency. The second set compares the performance of CNN_TF and that of typical bio-classifiers which cannot extract dependency. The third set evaluates CNN_TF by 5 TFs in five cell-types of humans. The fourth set compares our proposed CNN_TF with the state-of-the-art predictors.

Datasets

Two datasets are used in this work to evaluate the performance of our proposed CNN_TF: 13 TFs in the mouse embryonic stem cell and 5 TFs in 5 cell-types of humans.

13 TFs in mES cell: 13 TFs in mouse embryonic stem (mES) cell have been widely used by multiple TFBS prediction methods: *CTCF, E2F1, Esrrb, Klf4, c-Myc, n-Myc, Nanog, Oct4, Sox2, Smad1, STAT3, Tcfcp2l1*, and *Zfx*. These 13 TFs are used to evaluate the performance of our proposed CNN_TF and compare with state-of-the-art methods. For all the 13 TFs in the mES cell, TFBSs are obtained from ChIP-seq experiments and the ChIP-seq data of these 13 TFs can be accessed freely [38]. Leave-one-chromosome-out cross-validation method is applied for evaluation where one chromosome is left out for test; one is used for validation and the remainder chromosomes are used for training. As all the 101 bp DNA fragments except the TFBSs are non TFBSs, the number of non TFBS is significantly larger than that of TFBSs. The imbalance between the number of TFBSs and that of non TFBSs can badly affect the performance of machine learning models. So in the training set, an approximately equal number of non TFBSs matching the approximate same composition distribution of all dinucleotide types with TFBSs are selected. In the validation phase, the TFBSs and all non TFBSs in the validation chromosome are used to determine the best hyperparameters for CNN_TF. In the testing phase, the TFBSs and all non TFBSs in the test chromosome are used for evaluation. The above test process is repeated for multiple times until all the chromosomes are tested one time and the performance is averaged over all the chromosomes. Note that there is no overlapping of non TFBSs in the training set, the validation set and the test set.

5 TFs in 5 cell-types: In order to evaluate the influence of different cell-types on the predicting performance of our method, we evaluate CNN_TF by a recent dataset collected by the Gene Expression Omnibus (GEO) [12]. In this dataset, five dissimilar TFs are selected: *CTCF, JunD, REST, GABP* and *USF2* and five dissimilar cell-types are selected: *GM12878, H1-hESC, HeLa-S3, HepG2* and *K562*. These five cell-types are chosen because they represent diverse classes of cell-types. Besides, ChIP-seq data of all the five TFs in these cell-types are available. The ChIP-seq signal peak lists of the five TFs in these five cell-types can be downloaded freely from literature [83]. As this dataset contains too many cell-type TF pairs, ten-fold cross-validation method is used to evaluate CNN_TF on this dataset instead of leave-one-chromosome-out cross-validation method to decrease time cost in evaluation. First, for each cell-type TF pair, approximately equal number of non TFBSs as that of TFBSs are selected, in which the composition distribution of all dinucleotide types for the selected TFBSs is approximately same with TFBSs. second, the combination of TFBSs and the selected non-TFBSs are divided into ten folds randomly. Then one fold of samples are left out for test, one is used for validation and the remainder eight folds are used for training set. Finally, the above test process is repeated 10 times as ten-folds and the performance is averaged over the 10 repeats.

Performance of first order dependency and higher order dependency

To demonstrate the superiority of higher order dependency over first order dependency for TFBS prediction, we compare their predicting performance on the 13 TFs in mES cell by leave-one-chromosome-out cross-validation.

Table 5.1 shows the AUCs of first order dependency and higher order dependency as well as their combined use on the 13 TFs in mES cell, where the best performers and the second best performers are marked by bold and underscore, respectively. Among the 13 TFs in mES cell, higher order dependency outperforms first order dependency significantly in 10 of the 13 TFs, first order dependency outperforms higher order dependency

Table 5.1: AUCs of first dependency and higher order dependency on the 13 TFs in mES cell

TF	first order	higher order	combine	p-value[a]	p-value[b]
Zfx	0.967	0.991	**0.995**	7.32e-04	6.67e-13
CTCF	0.985	0.945	**0.991**	3.69e-05	1.81e-24
c-Myc	0.973	0.986	**0.992**	6.76e-03	1.01e-07
n-Myc	0.968	0.980	**0.983**	3.87e-02	2.41e-06
E2f1	0.913	0.986	**0.989**	2.26e-02	8.33e-22
Esrrb	0.977	0.972	**0.994**	3.39e-19	4.81e-04
Klf4	0.970	0.986	**0.994**	2.97e-11	1.53e-17
Tcfcp	0.964	0.964	**0.987**	8.36e-01	1.12e-21
Nanog	0.909	0.928	**0.964**	4.62e-10	6.11e-05
Oct4	0.908	0.967	**0.981**	1.70e-06	3.14e-18
Smad	0.809	0.942	**0.944**	8.78e-03	1.81e-15
Sox2	0.939	0.966	**0.985**	1.39e-12	2.67e-10
STAT	0.904	0.958	**0.969**	6.59e-03	6.79e-14

[a] denotes the comparison between first order and higher order dependency, [b] denotes the maximum p-value of the comparisons between the combine and the two individual feature types.

only in 2 of the 13 TFs. one TF as similar performance for both first order dependency and higher order dependency. These results indicate that higher order dependency contain more useful information than first order dependency on most TFs. When first order dependency and higher order dependency are used in combination, it outperforms both first order dependency and higher order dependency significantly in all the 13 TFs. This is a clear indication that first order dependency and higher order dependency are two complimentary features for TFBS prediction. Two sets of p-values are given in this experiment by Wilcoxon rank sum test The two p-values for each TF show that improvements by higher order dependency and combined use are significant with p-value at no more than 3.87e-02.

Comparison with typical bio-classifiers

The main advantage of our proposed method CNN_TF is that it can extract both first order dependency and higher order dependency from sequence features and histone modification features, respectively. To demonstrate the learned features are indeed useful, this evaluation compares the performance of CNN_TF with typical classifiers used for bioinformatics classifications including support vector machine (SVM) [169], neural network (NN) [22] and random forest (RF) [31]. Neural networks can learn higher dependencies by their hidden layers. In SVM, radial kernel is used. So SVM can learn higher order dependencies by the radial kernel. Evaluation is done on the 13 TFs in mES cell by leave-one-chromosome-out cross-validation. The AUCs of CNN_TF and the three traditional classifiers are shown in Table 5.2, where the best performers and the second best performers are marked by bold and by underscore, respectively. Note that the input features for CNN_TF and the three traditional classifiers are same.

We first compared CNN_TF containing only first order dependencies (called FOD_CNN) to SVM and NN. The main difference between them is that FOD_CNN consists of only a CNN to learn first order dependencies from sequence features while SVM and NN can learn both first order dependencies by relationships among sequence features and higher order dependencies by relationships between among histone modification features. Table 5.2 shows that SVM outperforms FOD_CNN in 11 TFs out of the 13 TFs in the mES cell. For the 11 TFs, the average improvement and the maximum improvement are 2.0% and 6.4%. When comparing FOD_CNN with NN, NN outperforms FOD_CNN for 11 TFs out of the 13 TFs in the mES cell. For Zfx, E2f1, Klf4, Oct4 and Sox2, the improvements are 2.1%, 6.7%, 1.6%, 4.4% and 1.6%, respectively, which are very significant improvements. As SVM and NN can learn long-range depedencies while FOD_CNN cannot learn them, addedbetter performances achieved by SVM and NN are contributed to the learned long-range dependencies. It concludes that long-range dependencies play an important role in TFBS predictions, irrespective of which method is applied to learn these dependencies.

We then compare CNN_TF containing both first order dependencies and higher order dependencies to SVM, RF and NN. In 7 out of the 13 TFs, CNN_TF outperforms all the other classifiers in larger margins ($> 1\%$) with the highest p-value at 2.33e-2 indicating that the improvements are very significant. More impressively, the improvements on CTCF, Nanog, Oct4, Smad, Sox2 and STAT3, are larger than 2%. For the remaining 5 TFs, although the improvements by CNN_TF are marginal ($< 1\%$), the improvements are significant as indicated by the p-value of no more than 4.95e-05. As SVM and NN also can extract higher order dependencies, the improvements achieved by CNN_TF indicates that higher order dependencies extracted by CNN_TF can be more useful than that extracted by SVM and neural networks. It indicates that CNN_TF is more effective for learning long-range dependencies than NN. Moreover, SVM and NN cannot learn higher order dependencies for predictions. On the contrary, our proposed CNN_TF can provide the learned higher order dependencies and their contribution in the predictions. It can provide us with deep understanding of the TF-DNA interactions.

Table 5.2: AUCs of CNN_TF and three state-of-the-art traditional classifiers on the 13 TFs in the mES cell

TF	CNN_TF	FOD_CNN	SVM	RF	NN	p-value[a]
Zfx	**0.995**	0.967	0.990	0.988	0.988	4.95e-05
CTCF	**0.991**	0.985	0.961	0.947	0.963	1.48e-21
c-Myc	**0.992**	0.973	0.979	0.984	0.976	8.78e-06
n-Myc	**0.982**	0.968	0.963	0.976	0.977	2.15e-07
E2f1	**0.989**	0.913	0.977	0.981	0.980	5.71e-09
Esrrb	**0.994**	0.977	0.980	0.971	0.979	4.37e-15
Klf4	**0.994**	0.970	0.988	0.985	0.986	4.08e-10
Tcfcp211	**0.987**	0.964	0.975	0.966	0.976	1.56e-12
Nanog	**0.964**	0.909	0.920	0.907	0.917	8.01e-12
Oct4	**0.981**	0.908	0.952	0.956	0.952	1.37e-08
Smad1	**0.944**	0.809	0.877	0.924	0.816	2.33e-02
Sox2	**0.985**	0.939	0.961	0.958	0.955	1.19e-10
STAT3	**0.969**	0.904	0.915	0.939	0.893	3.67e-10

[a] denotes the maximum p value of the comparisons between CNN_TF and the three state-of-the-art traditional classifiers.

Performance of CNN_TF on five TFs in five cell-types

Several recent studies have reported that TF binding is influenced by chromatin contextual features such as DNA accessibility, nucleosome occupancy, or the presence of some specific histone post-translational modifications. These chromatin contextual features are different for different cell-types. So in this evaluation, CNN_TF is applied to predict TFBSs for TFs in multiple different cell-types to analyze their influence on prediction performance. We use five diverse TFs as examples: (1) the insulator protein **CTCF** featuring 11 zinc finger domains, (2) the transcriptional activator **GABPA**, (3) **JunD**, a leucine zipper protein and member of the activator protein 1 (AP1) family, (4) the transcriptional repressor **REST**, and (5) **USF2** a member of the evolutionary conserved basic helix-loophelix leucine zipper TF family. These factors are chosen because they represent diverse classes of transcription factors. For the five TFs, five human cell-types are considered: GM12878, H1-hESC, HeLa-S3, HepG2 and K562. The five cell-types are selected because they represent diverse classes of cell-types in humans [83].

We first evaluate the influence of different cell-types on the contributions of first order dependency and higher order dependency for TFBS prediction. The AUCs of first order dependency, higher order dependency and their combined use are shown in Table 5.4, where the best performers and the second best performers are marked by bold and underscore, respectively. Results show that for each TF, the predicting performance of the first dependency for different cell-types are different. In the case of GABPA, higher order dependency outperforms first order dependency significantly in all the five cell-types. When the two features are combined, the predicting performance is improved significantly in all the five cell-types. On the other hand, first order dependency perform significantly better than higher order dependency for CTCF in all the five cell types. In the CTCF case too, the combined method still gains significant improvement in all the five cell types. For the remaining three TFs, no single dependency type plays a dominant role. However, the performance for all the cell-types are improved significantly when the two features are combined. This experiment clearly shows that first order dependency and higher order dependency have different contributions in TFBS prediction for different cell-types. Fur-

Table 5.3: AUC of first order dependency and higher order dependency on the five TFs in the five cell-types

TF	CELL	first order	higher order	Combine	p-value[a]	p-value[b]
CTCF	GM12878	0.949	0.751	**0.941**	2.27e-03	5.01e-14
	H1-hESC	0.898	0.700	**0.929**	3.29e-02	2.52e-16
	HeLa-S3	0.930	0.738	**0.895**	2.38e-05	1.03e-06
	HepG2	0.928	0.792	**0.938**	2.81e-01	8.46e-12
	K562	0.888	0.744	**0.884**	9.19e-01	1.12e-07
GABPA	GM12878	0.908	0.955	**0.982**	4.74e-10	2.32e-04
	H1-hESC	0.851	0.876	**0.913**	2.37e-03	1.26e-03
	HeLa-S3	0.926	0.906	**0.955**	6.31e-02	2.64e-02
	HepG2	0.931	0.946	**0.983**	9.29e-07	2.84e-03
	K562	0.872	0.954	**0.966**	7.37e-07	4.46e-01
JunD	GM12878	0.772	0.993	**0.994**	1.21e-12	6.09e-01
	H1-hESC	0.932	0.846	**0.942**	3.51e-01	8.07e-04
	HeLa-S3	0.954	0.964	**0.991**	8.04e-09	3.21e-02
	HepG2	0.977	0.746	**0.971**	1.31e-01	2.64e-08
	K562	0.879	0.874	**0.948**	1.43e-03	7.63e-03
REST	GM12878	0.816	0.790	**0.935**	4.91e-08	2.11e-04
	H1-hESC	0.832	0.714	**0.917**	1.64e-09	5.63e-09
	HeLa-S3	0.877	0.770	**0.942**	1.32e-03	2.26e-05
	HepG2	0.816	0.768	**0.945**	5.41e-11	1.67e-09
	K562	0.839	0.878	**0.951**	1.31e-11	5.41e-06
USF2	GM12878	0.947	0.902	**0.955**	7.96e-01	2.23e-02
	H1-hESC	0.897	0.817	**0.920**	4.41e-01	5.68e-04
	HeLa-S3	0.932	0.877	**0.940**	8.47e-01	1.82e-02
	HepG2	0.855	0.840	**0.930**	4.35e-02	1.45e-03
	K562	0.806	0.903	**0.950**	1.41e-05	1.21e-09

[a] denotes the comparison between firs order and higher order dependency, [b] denotes the maximum p-value of the comparisons between the combine and the two individual features.

thermore, the two types of dependency are complementary to each other and thus their combined use outperforms any single use irrespective of their dominance as a single dependency for different cell-types.

Then, we evaluate the roles of another type of features, DNA methylations, in TFBS predictions. DNA methylation is a process by which methyl groups are added to the DNA molecule. Methylation can change the activity of a DNA segment without changing the

Table 5.4: AUC of CNN_TF with DNA methylation and that without DNA methylation on the five TFs in the five cell-types

TF	CELL	histone	histone	combine	combine	p-value[a]	p-value[b]
			+methy		+methy		
CTCF	GM12878	0.751	**0.752**	**0.941**	**0.941**	9.29e-01	9.84e-01
	H1-hESC	**0.700**	0.677	**0.929**	0.926	9.87e-03	5.17e-01
	HeLa-S3	**0.738**	**0.738**	0.895	**0.898**	9.99e-01	7.59e-01
	HepG2	**0.792**	**0.792**	**0.938**	**0.938**	9.97e-01	9.20e-01
	K562	**0.744**	0.736	0.884	**0.885**	6.70e-01	9.69e-01
GABPA	GM12878	0.955	**0.962**	**0.982**	**0.982**	2.10e-01	9.03e-01
	H1-hESC	**0.876**	0.874	0.913	**0.914**	8.97e-01	9.33e-01
	HeLa-S3	0.906	**0.928**	0.955	**0.961**	1.70e-01	4.72e-01
	HepG2	0.946	**0.951**	**0.983**	**0.983**	6.60e-01	9.64e-01
	K562	**0.954**	0.951	**0.966**	0.965	6.89e-01	8.49e-01
JunD	GM12878	**0.993**	0.992	**0.994**	**0.994**	8.36e-01	8.30e-01
	H1-hESC	**0.846**	0.832	**0.942**	0.939	6.75e-01	8.05e-01
	HeLa-S3	0.964	**0.966**	**0.991**	**0.991**	1.70e-01	4.72e-01
	HepG2	0.746	**0.748**	**0.971**	**0.971**	9.71e-01	9.16e-01
	K562	**0.874**	0.839	**0.948**	0.941	3.06e-01	5.64e-01
REST	GM12878	0.790	**0.815**	0.935	**0.936**	4.03e-01	9.37e-01
	H1-hESC	0.714	**0.728**	**0.917**	**0.917**	4.75e-01	5.59e-01
	HeLa-S3	0.770	**0.781**	**0.942**	0.941	7.48e-01	9.34e-01
	HepG2	0.768	**0.774**	**0.945**	0.943	7.55e-01	8.12e-01
	K562	**0.878**	0.877	**0.951**	0.948	9.15e-01	4.38e-01
USF2	GM12878	**0.902**	**0.902**	**0.955**	0.954	9.84e-01	9.90e-01
	H1-hESC	**0.817**	0.810	**0.920**	0.919	8.06e-01	9.51e-01
	HeLa-S3	**0.877**	0.875	0.940	**0.941**	9.50e-01	9.45e-01
	HepG2	**0.840**	0.839	**0.930**	0.929	9.62e-01	9.43e-01
	K562	**0.903**	0.899	**0.950**	**0.950**	8.89e-01	9.71e-01

[a] denotes the comparison between histone modification with DNA methylation and that without DNA methylation, [b] denotes the maximum p-value of the comparisons between the combination with DNA methylation and that without DNA methylation the two individual features.

sequence. When located in a gene promoter, DNA methylations typically act to repress gene transcription. DNA methylations are essential for normal development and is associated with a number of key processes including genomic imprinting, X-chromosome inactivation, repression of transposable elements, aging, and carcinogenesis. This means that DNA methylations may also be important features for TFBS predictions.

We conduct two experiments to evaluate the roles of DNA methylations in TFBS predictions: the first one is combining with only histone modification features and the second one combines both histone modification features and sequence features. In these two experiments, DNA methylations are calculated by the same method as histone modifications and are added into histone modification features as an additional dimension. Therefore, the combination of DNA methylations and histone modification features for an instance can be encoded by an feature matrix with dimension of 8×20. The results of these two experiments are shown in Table 5.3. Table 5.3 shows that among the 25 cell-type TF pairs, there is only one pair, CTCF in H1-hESC, the performance is decreased significantly when DNA methylations are combined with histone modification features. The differences for all the remaining pairs are not significant. It indicates that DNA methylations cannot provide additional contributions for TFBS predictions. The reason may be that the roles of DNA methylations in TFBS predictions are redundant with that of histone modification features. Our future work will explore which modification features have redundant roles with DNA methylations.

Comparison between CNN_TF and state-of-the-art methods on TFs in mES cell

In this experiment, we compare our method with several state-of-the-art methods including Chromia [190], Cluster-Buster (CB) [54], MCAST [9], EEL [130] and Stubb [154] on the TFs in mES cell. Stubb has two versions: one is called the Stubb-Single (SS) and the other is called Stubb-Multiple (SM). Chromia was proposed by Won at al. [16] based on a HMM model, in which both histone modification features and sequence features are used for learning feature representation. In Chromia, three HMM models including promoter model, enhancer model and background model are trained and the log-odd score of the promoter model or the enhancer model to the background model is used for prediction. Cluster-Buster [54] uses motifs documented in databases including JASPAR [33] and TRANSFAC [112] or predicted by de novo motif finding algorithms to search for TF-

Table 5.5: PPVs of CNN_TF and three state-of-the-art methods on the 13 TFs in mES cell

TF	CNN_TF	Chromia	CB[a]	MCAST	EEL	SS[b]	SM[c]
Zfx	**81.5%**	51.7%	5.6%	0.2%	24.8%	46.9%	26.0%
CTCF	**98.6%**	13.2%	51.3%	37.9%	44.0%	13.4%	3.9%
Myc	**82.8%**	57.8%	7.1%	0.4%	3.3%	20.2%	17.8%
E2f1	**98.1%**	85.3%	0.0%	1.3%	0.5%	12.0%	8.2%
Esrrb	**66.8%**	23.5%	9.7%	4.9%	16.2%	13.9%	5.1%
Klf4	**60.0%**	34.2%	5.7%	0.3%	12.5%	28.6%	9.5%
Tcfcp211	**77.3%**	33.8%	5.0%	11.5%	27.2%	12.7%	5.3%
Nanog	**47.3%**	7.8%	0.0%	0.4%	0.7%	1.4%	0.1%
Oct4	**25.0%**	15.0%	0.0%	2.8%	3.5%	0.5%	0.0%
Smad1	**10.6%**	1.0%	0.0%	0.4%	0.2%	0.0%	0.0%
Sox2	**35.8%**	4.2%	0.0%	2.4%	2.8%	0.2%	0.8%
STAT3	**17.1%**	1.0%	0.0%	0.2%	1.6%	2.9%	0.8%

[a] denotes Cluster-Buster, [b] denotes Stubb-Single and [c] denotes Stubb-Multiple.

BSs from test sequence. MCAST [9] uses a motif-based HMM model with several novel features to model TFBSs, for which a DNA database and a collection of known binding motifs are used as inputs. In MCAST, motif-specific p-value is used to identify motif occurrence by a user-specified threshold. EEL [130] uses motif conservation information and TFBS clustering in the prediction model, which locates the enhancer elements according to a simplified biochemical and physical model of TF binding [130]. In EEL, the binding score of a putative TFBS is calculated by aligning the putative TFBS to the orthologous sequences and used for prediction. Stubb [154] includes motif conservation information and TFBS clustering in the prediction model and uses a HMM framework to model enhancer. In Stubb, the calculated free energy is used for prediction. The free energy in Stubb-Single is based on correlations between binding sites while that in Stubb-Multiple incorporates phylogenetic comparisons among sequences from multiple species.

All the six state-of-the-art methods were run using their default setup and parameters. For Chromia, the bins containing both strong histone modification signals and large PSSM scores were selected for training. More specifically, at first, H3K4me3 and H3K4me1 or

H3K4me2 were used to select promoters and enhancers by a cutoff value, respectively. Next, all the selected promoters or enhancers are ranked by PSSM score in descending order. The top 100 promoters and 100 enhancers were selected to train a promoter and a enhancer model, respectively, and a background model was trained by the entire chromosome 1. The log-odd score of the promoter or the enhancer model to the background model is used for prediction. Cluster-Buster was run with the option '-p0 -m0 -c0' to get the output. For MCAST, the option '-e-thresh 0' was selected to turn off thresholding. To run EEL and Stubb, we used human and mouse orthologous sequences obtained from the UCSC genome browser. For Stubb-Multiple, we used LAGAN [32] to align human and mouse orthologous sequences and used 'window size' = 500 and 'shiftsize' = 100.

As Stubb and EEL both require pairwise alignment with other genomes and it is too time-consuming to evaluate their performance on the entire genome, only 20 chunks of the genomic sequences (total513,846,568 bp) [16] that had pairwise alignment with the human genome were selected from the UCSC genome browser for test. The remainder genomic sequences are used for training. As the TFBSs for c-Myc and n-Myc have similar properties and Chromia combined them into a dataset labeled as Myc, we also incorporated them into a dataset. In the evaluation for the 6 state-of-the-art methods and our CNN_TF method, the top 600 sites with larger prediction weight are used for evaluation. The PPV score of the top 600 predicted sites for each method is calculated. The PPVs of CNN_FT and the state-of-the-art methods are shown in **TABLE 5.5**, where the best performers and the second best performers are marked by bold and underscore, respectively. Results show that CNN_TF achieves obvious improvements for all the 12 TF sets. For some TFs, the improvement by CNN_FT is very promising. For example, the improvement for CTCF is over 60%PPV and the improvements for Esrrb, Tcfcp and Nanog are over or near 40%PPV. Note that some of the state-of-the-art methods cannot even provide any true TFBSs for some TFs. For example, Stubb-Single and Stubb-Multiple cannot identify any true TFBSs for Smad and EEL; Cluster-Buster cannot identify any true TFBSs for

E2f1, Nanog, Oct4, Smad1, Sox2 and STAT3. This comparison validates the usefulness of higher order dependency for TFBS prediction.

Comparison of CNN_TF with state-of-the-art methods on TFs in cell-types of humans

DNA shape represents the 3D structure of a DNA. Recently, Mathelier at al. [111] proposed several DNA shape based methods for TFBS prediction in vivo. Four DNA shape features including helix twist (HelT), minor groove width (MGW), propeller twist (ProT), and Roll were used to represent putative TFBSs, which were computed by the DNA shape method [216]. In Mathelier's work, four prediction methods were proposed: (1) 4-bits+shape, which combines one-hot encoding with DNA shape features; (2) PSSM+shape, which combines PSSM encoding with DNA shape features; (3) TFFM_d+shape, which combines detailed TFFM encoding and DNA shape features, and (4) TFFM_f+shape, which combines 1st-order TFFM encoding and DNA shape features. The one-hot encoding, PSSM encoding and TFFM encoding used in these DNA shape based methods are representations of DNA sequence features, so the inputs of our proposed CNN_TF and the four DNA shape based methods are same except that the additional features in the four methods are DNA shape features whereas the additional features in our method are higher order dependency.

To make the comparison fair, the evaluation is conducted on the five TFs in the five cell-types of humans by ten-fold cross-validation. The results are listed in **TABLE 5.6**, where the best performers and the second best performers are marked by bold and underscore, respectively. **TABLE 5.6** shows that CNN_TF outperforms the four DNA shape based methods significantly on all the five TFs in all the five cell-types. CNN_TF performs better than the four DNA shape based methods by at least 0.073 AUC. The outperformance of our method over the four DNA shape based methods indicates that higher order dependency is more useful than DNA shape features.

Table 5.6: AUCs of the four DNA shape based methods and CNN_TF on TFs in cell-types of humans

TF	CELL	4-bits	PSSM	TFFM_d	TFFM_f	CNN_TF	p-value
CTCF	GM12878	0.763	0.762	0.748	0.750	**0.935**	9.94e-04
	H1-hESC	0.762	0.758	0.740	0.744	**0.929**	7.23e-04
	HeLa-S3	0.739	0.736	0.724	0.726	**0.900**	3.95e-03
	HepG2	0.759	0.757	0.741	0.746	**0.943**	9.18e-04
	K562	0.747	0.745	0.731	0.733	**0.886**	9.88e-03
GABP	GM12878	0.830	0.828	0.830	0.830	**0.981**	6.84e-03
	H1-hESC	0.832	0.828	0.824	0.824	**0.905**	1.96e-01
	HeLa-S3	0.796	0.796	0.792	0.787	**0.955**	5.11e-03
	HepG2	0.842	0.838	0.830	0.837	**0.978**	7.91e-03
	K562	0.822	0.817	0.812	0.815	**0.966**	8.59e-03
JunD	GM12878	0.752	0.751	0.742	0.753	**0.993**	1.17e-13
	H1-hESC	0.762	0.760	0.750	0.753	**0.947**	8.44e-14
	HeLa-S3	0.800	0.797	0.773	0.777	**0.989**	2.55e-14
	HepG2	0.774	0.771	0.754	0.757	**0.970**	1.18e-19
	K562	0.763	0.760	0.742	0.746	**0.945**	8.31e-14
REST	GM12878	0.782	0.780	0.764	0.774	**0.927**	1.45e-02
	H1-hESC	0.768	0.768	0.751	0.753	**0.917**	9.79e-03
	HeLa-S3	0.620	0.621	0.605	0.594	**0.936**	1.75e-19
	HepG2	0.781	0.780	0.771	0.770	**0.938**	2.27e-02
	K562	0.774	0.771	0.758	0.763	**0.948**	8.95e-03
USF2	GM12878	0.773	0.771	0.754	0.758	**0.952**	2.11e-12
	H1-hESC	0.784	0.780	0.765	0.770	**0.916**	1.01e-08
	HeLa-S3	0.750	0.746	0.731	0.735	**0.937**	5.96e-12
	HepG2	0.788	0.784	0.762	0.766	**0.924**	1.14e-07
	K562	0.773	0.773	0.752	0.753	**0.945**	8.53e-09

In addition to the DNA shape based methods, several deep learning methods have been proposed. DeepSea [210] and DanQ [138] are two representative methods. DeepSea [210] was proposed by Zhou and Troyanskaya (2015) by applying CNN on DNA sequence and DanQ [138] was proposed by Quang and Xie (2016) by combining CNN and Recurrent neural network (RNN) on sequence features to learn features. Both DeepSea and DanQ used multi-task learning to learn representations for putative TFBSs and contain 919 tasks including 690 TFBS prediction tasks for 160 TFs, modification value prediction tasks for

104 histone marks, prediction tasks for 125 DNase I–hypersensitive sites (DHSs). As the comparison among CNN_TF and them is conducted on 5 TFs in 5 cell-types of humans, the tasks in DeepSea [210] and DanQ [138] contain 25 TFBS prediction tasks (the 5 TFs in the 5 cell-types of humans). For DanQ [138], Quang and Xie (2016) also have proposed an alternative model, called DanQ-JASPAR, by initializing half of the kernels in CNN with motifs from the JASPAR database [109]. For DeepSea [210], we also consider its alternative model, abbreviated as DeepSea-JASPAR, by using the same kernel initializing method. We downloaded the torch (https://github.com/torch/torch7) implementation of DeepSea [210] from the software's webpage (http://deepsea.princeton.edu/) and the Keras (https://github.com/fchollet/keras) implementation of DanQ [138] from the software's webpage (http://github.com/uci-cbcl/DanQ). All these four state-of-the-art methods were run using their default setup and parameters. The AUCs of our method CNN_TF and the four state-of-the-art methods are listed in TABLE 5.9. TABLE 5.9 shows that CNN_TF performs better than the other four methods for 24 out of the 25 cell-type-TF pairs. On the 24 cell-type-TF pairs, the minimum improvement and the maximum improvement achieved by our method are 0.051 on GABP in HepG2 and 0.258 on REST in HeLa-S3, respectively. The average improvement on the 24 cell-type-TF pairs is 0.145, which is a prominent improvement. As the dominated difference between our proposed CNN_TF and the four state-of-the-art methods is that CNN_TF can extract higher order dependency while the four state-of-the-art methods cannot extract it, the improvements indicates that the higher order dependency learned by CNN_TF indeed plays an important role for TFBS prediction.

5.2.3 Analysis of learned features

Distinct histone modification features have been observed at various genomic loci including promoters and enhancers. Won [190] investigated the ChIP-seq signals of the eight types of histone modification features for the TFBSs of the 13 TFs in the mES cell. They

Table 5.7: AUCs of CNN_TF and four state-of-the-art methods on TFs in cell-types of humans

TF	CELL	DanQ	DanQ-J	DeepSea	DeepSea-J	CNN_TF
CTCF	GM12878	0.780	0.703	0.745	0.617	**0.935**
	H1-hESC	0.824	0.723	0.767	0.656	**0.929**
	HeLa-S3	0.754	0.670	0.699	0.605	**0.900**
	HepG2	0.826	0.724	0.772	0.644	**0.943**
	K562	0.772	0.687	0.720	0.618	**0.886**
GABP	GM12878	0.929	0.907	0.906	0.895	**0.981**
	H1-hESC	**0.922**	0.906	0.907	0.894	0.905
	HeLa-S3	0.808	0.772	0.766	0.752	**0.955**
	HepG2	0.927	0.914	0.913	0.906	**0.978**
	K562	0.911	0.898	0.900	0.892	**0.966**
JunD	GM12878	0.835	0.779	0.789	0.729	**0.993**
	H1-hESC	0.771	0.718	0.726	0.699	**0.947**
	HeLa-S3	0.850	0.721	0.766	0.671	**0.989**
	HepG2	0.842	0.725	0.765	0.697	**0.970**
	K562	0.717	0.652	0.664	0.624	**0.945**
REST	GM12878	0.750	0.651	0.657	0.630	**0.927**
	H1-hESC	0.699	0.604	0.603	0.580	**0.917**
	HeLa-S3	0.678	0.580	0.584	0.560	**0.936**
	HepG2	0.758	0.673	0.682	0.664	**0.938**
	K562	0.756	0.711	0.715	0.695	**0.948**
USF2	GM12878	0.789	0.706	0.710	0.689	**0.952**
	H1-hESC	0.849	0.775	0.780	0.758	**0.916**
	HeLa-S3	0.723	0.637	0.644	0.609	**0.937**
	HepG2	0.811	0.691	0.693	0.668	**0.924**
	K562	0.809	0.704	0.717	0.691	**0.945**

found that H3K4m1, H3K4m2 and H3K4m3 show strong signals for all the TFBSs of the 13 TFs. In contrast, the signals of H3K27m3 are much weaker. Their study on the association of any histone modification feature patterns with a specific TF shows that H3K4me1 and H3K4me2 present a distinct bimodal profile in all TFBSs; H3K4me3 shows a strong peak for the TFBSs of E2F1, c-Myc, n-Myc and Zfx, intermediate peaks for Esrrb, Klf4, STAT3 and Tcfcp211, and weak signals for CTCF, Nanog, Oct4, Smad1 and Sox2. H3K36me3 shows relatively strong signals for E2f1, c-Myc, n-Myc and Zfx. The repressive features H3K9me3, H3K20me3 and H3K27me3 show an overall low signal. The advantage of our proposed CNN_TF is that it can capture higher order dependency and then combine with first order dependency for prediction. To demonstrate the competence of CNN_TF for higher order dependency extraction, we analyze the higher order dependencies extracted by CNN_TF for all the 13 TFs in mES cell. In CNN_TF, d filters are used to calculate higher order dependency.

For each histone modification feature, the higher order dependency is calculated as the weighted sum of all the learned filters from histone modification features. In order to show the learned higher order dependency learned by CNN_TF directly, we sum up the d learned filters by the following formula:

$$F = \sum_{k=1}^{d} W_{0,(d+k)} F_C^k, \tag{5.5}$$

where $W_{0,*}$ is the weight vector in the neural network classifier of CNN_TF and used for classifying input sequences as TFBSs. The weights are the contributions of the corresponding learned filters in classifying input sequences as TFBSs. F_C represents the learned filters from histone modification features by CNN_TF. d denotes the number of filters histone modification features. For more details about W_0, please refer to Formula (5.4). In fact, F in Formula (5.5) shows how higher order dependencies is learned from histone modification features. Due to the length limit of this paper, we only show the learned higher order dependencies for c-Myc and Oct4. The learned higher order dependencies for other TFs are listed in Figure S1 to S11 in the Additional file 1 in our website (http://hlt.hitsz.edu.cn/CNN_TF/). The learned higher order dependencies for c-Myc and Oct4 are shown in Figure 5.2 and Figure 5.3, respectively.

Figure 5.2 shows that H3K4me1 and H3K4me2 indeed present a bimodal profile in the TFBSs of c-Myc. H3K4me3 also shows a strong signal in the TFBSs. The learned higher order dependencies for H3K9me3, H3K20me3 and H3K27me3 show an overall

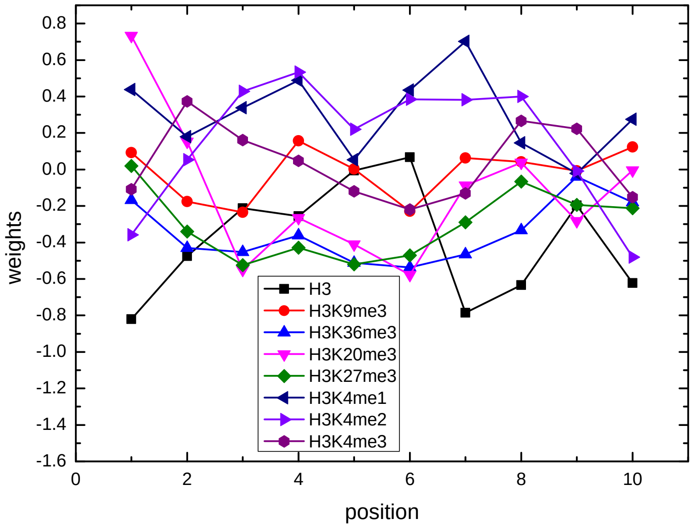


Figure 5.2: The learned higher order dependency for Oct4.

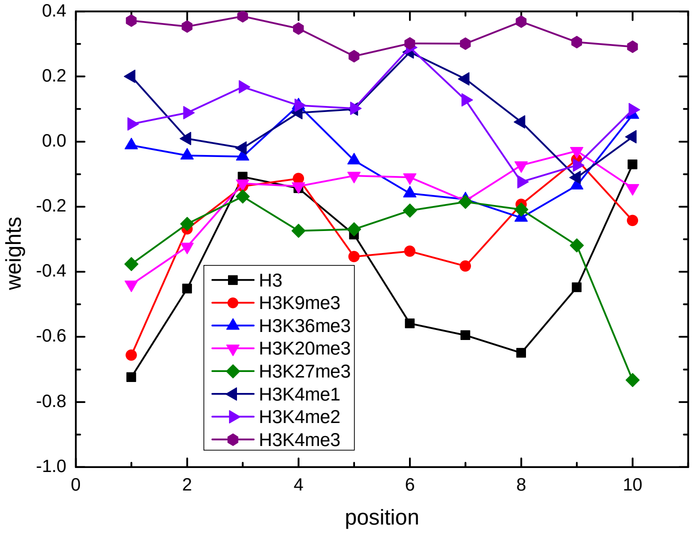


Figure 5.3: The learned higher order dependency for c-Myc.

low signal for c-Myc. This indicates that the learned higher order dependencies are consistent with the dependencies from ChIP-seq signals by a previous study [190]. **Figure 5.3** also shows that H3K4me1 and H3K4me2 present bimodal profile. H3K4me3 and the three repressive features including H3K9me3, H3K20me3 and H3K27me3 show weak signals for Oct4, which are also consistent with the conclusions of a previous study [190].

These results show that CNN_TF can indeed capture useful higher order dependencies for prediction. By observing **Figure 5.2** and **Figure 5.3**, we find that, except for H3K4me1 and H3K4me2 described above, other histone modification features including H3K4me3, H3K9me3 and H3K20me3 also show bimodal profile for both c-Myc and Oct4. Furthermore, the learned dependencies for the remaining 11 TFs, which are listed in Addition file 1, also show bimodal profile for the 5 histone modification features. This set of experiments indicates that among the eight histone modification features used in CNN_TF, five histone modification features show bimodal profile in the TFBSs for all the 13 TFs.

In summary, CNN_TF extracts both first order dependency and higher order dependency by applying CNN on sequence features and histone modification features to predict TFBSs for TFs. However, to predict the TFBSs of target TFs for specific cell-types, CNN_TF need sufficient training samples of the target TFs from the specific cell-types. However, many target TFs do not have training samples in the specific cell-types. Thus CNN_TF cannot be applied to predict TFBSs of the target TFs in these cell-types.

5.3 MTTFsite:Multi-task learning based method

5.3.1 Multi-Task Learning for TFBS Prediction (MTTFsite)

Multi-task learning is an effective approach for improving the performance of a single task with the help of other related tasks [96]. Multi-task learning attempts to divide the features of multiple tasks into private and common spaces based on whether parameters of some components should be shared. In multi-task learning framework, each task contains two feature spaces, private feature space and common feature space. For TFBS prediction, the learned private feature space contains the interaction mechanism specific to the target cell-type, which is referred to as the private interaction mechanism. The common feature space is referred to as the common interaction mechanism, which are shared by multiple cell-types. Assuming for each cell-type (task) m, we have a dataset D_m with N_m samples, each sample is a pair of a DNA fragment x_i^m and its corresponding label y_i^m, that is:

$$D_m = \{(x_i^m, y_i^m)\}_{i=1}^{N_m} \tag{5.6}$$

In multi-task learning, there could be two types of learning methods, fully-shared method and shared-private method [96]. The fully-shared method uses a single common CNN to extract features for multiple cell-types, which hypothesizes that all cell-types share the same feature space (as shown in the left panel of Figure 5.4). The shared-private method contains two feature spaces for each cell-types, private feature space and common feature space. The private feature space contains features that are specific to the target cell-type while the common feature space contains features that are common for all cell-types. The shared-private scheme is illustrated in the right panel of Figure 5.4. In the TFBS prediction problem investigated here, each cell-type is considered as a task.

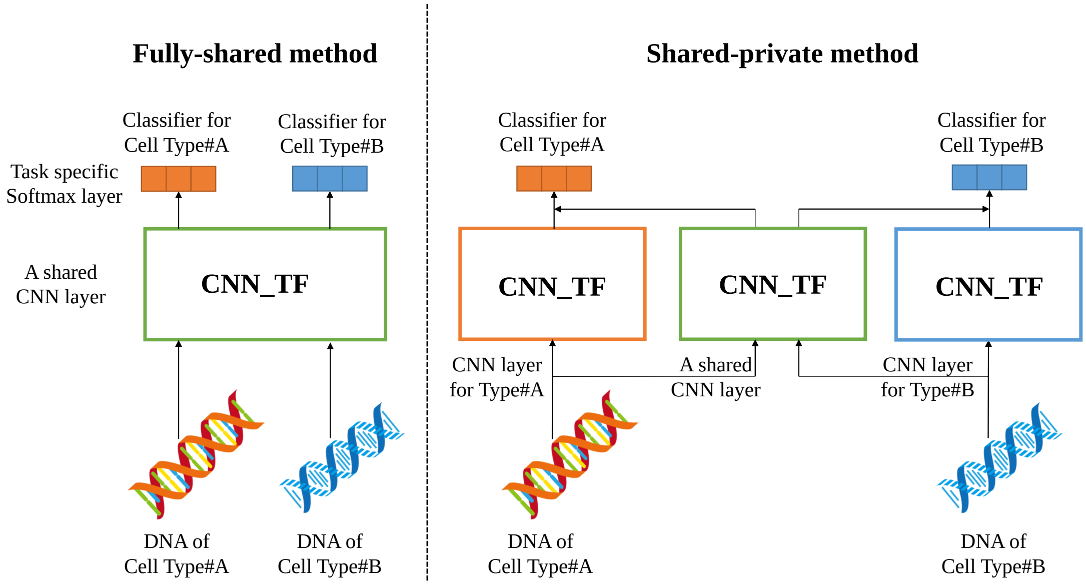


Figure 5.4: Architecture of multi-task learning for TFBS Prediction.

The fully-shared method assumes the multiple cell-types have a common CNN for feature learning while the shared-private method has a private CNN for each cell-type to learn private features apart from the common CNN across multiple cell-types to learn common features. As CNN is used to learn representations for all TFBSs, the private interaction mechanism h^m and the common interaction mechanism s^m for TFBSs in cell-type m are formally formulated as:

$$h^m = \mathrm{CNN}(x, \theta_m) \tag{5.7}$$

$$s^m = \mathrm{CNN}(x, \theta_s) \tag{5.8}$$

where x is the input for a TFBS; θ_m and θ_s are the parameters for the private interaction mechanism for cell-type m and the common interaction mechanism, respectively. Our proposed MTTFsite follows the shared-private method and it therefore has the ability to take into account both private and common interaction mechanisms for TFBS prediction.

For both the fully-shared method and shared-private method, the network topology in our proposed CNN_TF is applied to extract features. The network topology contains two convolution layers: one convolution layer for sequence features and one convolution layer for histone modification features. The convolution layer for sequence features consists of 200 convolution kernels of length 10, followed by a maxpooling with size of 92. The convolution layer for histone modification features consists of 200 convolution kernels of length 10, followed a maxpooling with size of 11. Then, the features learned by the two convolution layers are concatenated into feature vectors. The classifiers in both the fully-shared method and shared-private method are multilayer perceptron, in which two fully connected layers of 200 neurons were used to dense feature vectors for TFBSs.

5.3.2 Experiments and Results

Four sets of evaluations are conducted here. The first set compares the performance between the fully-shared method and the baseline method without applying multi-task learning framework. The second set compares the performance of the shared-private method and that of the fully-shared method. The third set compares our proposed MTTFsite with state-of-the-art predictors and the last one applies MTTFsite for cross-cell TFBS prediction. In this section, the datasets used to evaluate our proposed method will be introduced first. Then, details of the four components of will be described in sequence.

Datasets

Five cell-types including GM12878, H1-hESC, HeLa-S3, HepG2 and K562 are used to evaluate our proposed method. As MTTFsite is required to be evaluated by the TFs with labeled data in at least two cell-types, where one is used for testing and the others are used for training, a total of 72 TFs are used to evaluate MTTFsite, where 17, 14, 18, 23 TFs have labeled data in all the five cell-types, four cell-types, three cell-types and two cell-types, respectively. The TFBSs of these TFs have been identified by ChIP-seq experiments and their peak lists can be downloaded from ENCODE freely. The obtained peaks are usually provided in one of two formats: **narrow peak** and **broad peak**. The narrow peak format, which requires technically more sophisticated equipments to get, can provide more accurate the positions for TFBSs than the broad peak format. So the narrow peak format is used if available, otherwise the broad peak format is used. Based on works [5], the TFBS at each peak is defined as a 101 bp sequence by taking the midpoint of the peak as the center. Contrast to TFBSs, the non-TFBSs of a TF are defined as 101 bp DNA regions which can not be bound by the target TF. Many literatures [190, 83] used a shuffle method to construct non-TFBSs. In the shuffle method, a non-TFBS is constructed for each TFBS by shuffling the dinucleotides in the TFBS to keep the distribution of dinucleotides unchanged. However, in this study, as both histone modification features and DNA sequences are needed to encode TFBSs and histone modification features need to be extracted from actual DNA sequences, we require to extract actual DNA fragments to construct non-TFBSs. So we construct a non-TFBS for each TFBS by selecting a 101 bp DNA fragment having more than 98% dinucleotide similarity with the TFBS and nonoverlapping with all the TFBSs. Thus, we can construct the same number of non-TFBSs as TFBSs for each TF. For each

TF in each cell-type, the labeled data are divided into 3 separate, but equal size folds: one fold for training, one fold for validation and one fold for test.

Results of fully-shared (FS) method

We first evaluate the performance of the fully-shared method on the TFs in the five cell-types and compare with a baseline method. The baseline method is similar to the fully-shared scheme except that the baseline method is trained by the training samples from only the target cell-type while the fully-shared method is trained by combining the training sample from the multiple cell-types. In this experiment, the experimentally identified TFBSs of each cell-type are divided into 3 separate, but equal size folds: training set, validation set and test set. We train the baseline method with the training set, select model with the best performance on the validation test and test it by the test set. For the the fully-shared method, the model is trained by combining the the training set from the target cell-type and the training samples from other multiple cell-types.

The comparison among the fully-share model and the baseline model is shown in Figure 5.5. Figure 5.5A shows that the fully-shared model performs better than the baseline method for most (cell-type,TF) pairs, where a (cell-type,TF) pair refers to a prediction task for a TF in a cell-type. Figure 5.5D shows that the first quartile, the median and the third quartile of AUC for the fully-shared model are higher than that of the baseline method. More specifically, the AUC of the fully-shared model and the baseline method for TFs in GM12878, H1-hESC, HeLa-S3, HepG2 and K562 are are listed in Table A.1, A.2, A.3, A.4 and A.5, respectively, where bold represents the best performance and underline represents the second best performance. Table A.1 shows that the fully-shared method outperforms the baseline method for 49 TFs out of the 58 TFs in GM12878. Table A.2 shows that the fully-shared scheme performs better than the baseline method for 31 out of the 41 TFs in H1-hESC. Table A.3 shows that there are 37 TFs out of the 42 TFs in HeLa-S3 on which the fully-shared method outperforms than the baseline method. Table A.4

shows that there are 42 TFs out of the 43 TFs on which the fully-shared method performs better than the baseline method. Table A.5 shows that there are 60 TFs out of the 65 TFs in K562 on which the fully-shared method outperforms the baseline method. It indicates that the learned common feature space indeed plays an important role for TFBS prediction and impies that the interaction between a TF and DNA indeed have common mechanisms among multiple cell-types.

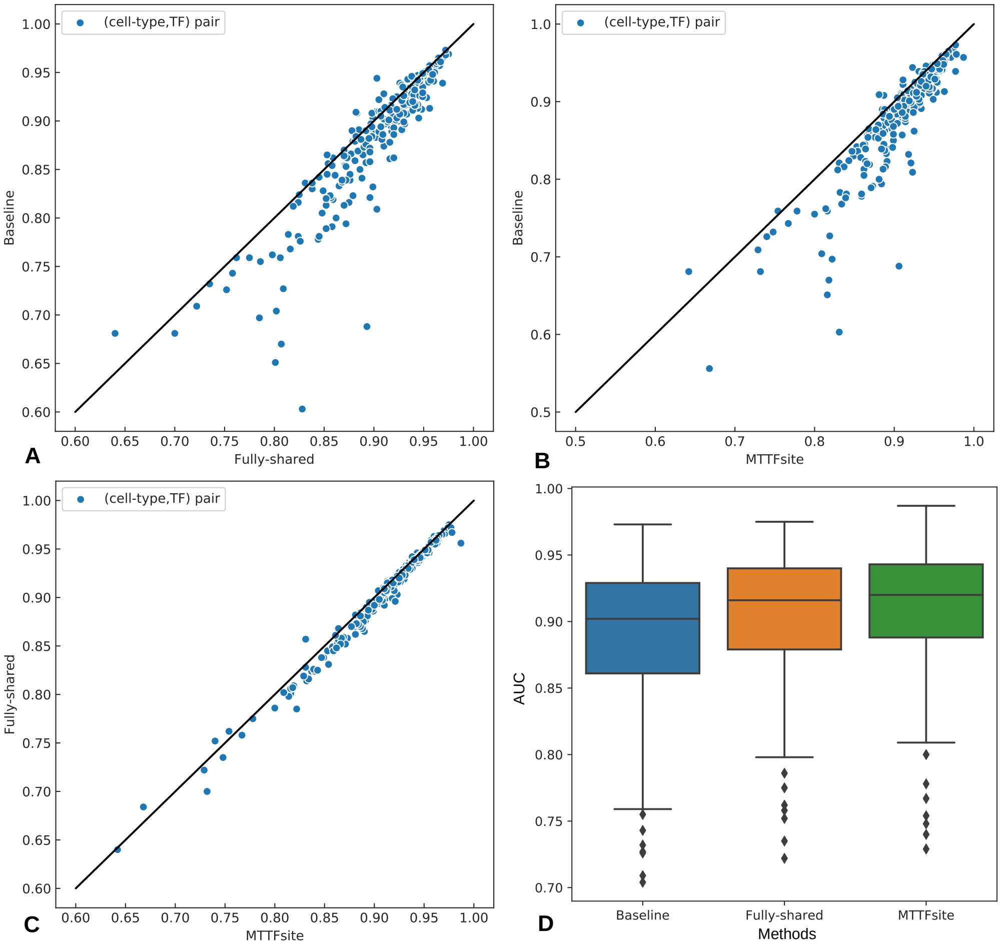


Figure 5.5: The AUCs for the cell-types with sparse labeled data.

Results of shared-private method (MTTFsite)

Our proposed MTTFsite contains a common CNN for multiple cell-types and a private CNN for each cell-type. The difference between the fully-shared method and our proposed

MTTFsite is that the fully-shared method has only a common CNN for multiple cell-types while MTTFsite also has a private CNN for each cell-type except the common CNN among multiple cell-types.

The comparison among the fully-share model, the baseline method and our proposed MTTFsite is shown in Figure 5.5. Figure 5.5B and Figure 5.5C shows that MTTFsite performs better than both the baseline method and the fully-shared model for most (cell-type,TF) pairs. Figure 5.5D shows that the first quartile, the median and the third quartile of the AUC for MTTFsite are higher than that of the baseline method and the fully-shared model. More specifically, the AUC for the baseline method, the fully-share model, and our proposed MTTFsite for TFs in GM12878, H1-hESC, HeLa-S3, HepG2 and K562 are listed in Table A.1, A.2, A.3, A.4 and A.5, respectively, where bold represents the best performance and underline represents the second best performance, respectively. Table A.1 shows that MTTFsite performs better than both the fully-shared method and the baseline method for 43 TFs out of the 58 TFs in GM12878. There are four TFs on which the baseline method outperforms both the fully-shared method and MTTFsite, it may be due to the TF-DNA interaction in multiple cell-types have dissimilar mechanisms. On these four TFs, even if MTTFsite achieves lower performance than the baseline method, it performs better than the fully-shared method, which validates that MTTFsite can indeed extract both common features and private features. Table A.1 also shows that there are 16 TFs on which our proposed MTTFsite performs better than the baseline method by 0.01 AUC. The improvements on some TFs are more than 0.05 AUC, such as ATFs, CTCF, SMC3, FOS, RAD21 and ZNF143, and the improvements on NRSF and TR4 are even more than 0.15 AUC. Note that there are 7 TFs on which MTTFsite performs better than the fully-shared method by more than 0.01 AUC. Table A.2 shows that there are 36 out of the 41 TFs in H1-hESC on which MTTFsite outperforms both the fully-shared method and the baseline method. There are six TFs on which the fully-shared method performs worse than the baseline method, but MTTFsite performs better than the baseline method. It validates that the fully-shared method loses the private features of the target cell-type when combining the training samples from the multiple cell-types while MTTFsite can extract both the common features among the multiple cell-types and the private of the target cell-type. Table A.2 also shows that there are 26 TFs out of the 41 TFs in H1-hESC on which our proposed MTTFsite performs better than the fully-shared method by more than 0.01 AUC. The improvements on some TFs are more than 0.05 AUC, such as ATF3, BRCA1, CTCF, MAFK, RAD21, RFX5 and SRF, and the improvement on NRSF is more 0.09 AUC. Note that there are 20 TFs on which MTTFsite performs better than the fully-shared method by more than 0.01 AUC.

Table A.3 shows that there are 41 TFs out of the 42 TFs in HeLa-S3 on which MTTFsite outperforms both the fully-shared method and the baseline method. There are five TFs on which the fully-shared method performs worse than the baseline method while MTTFsite performs better than the baseline method. It also validates that MTTFsite can extract both the common features among the multiple cell-types and the private of the target cell-type. Table A.3 also shows that there are 17 TFs on which the improvements achieved by MTTFsite are more than 0.01 AUC. The improvements on some TFs are more than 0.05 AUC, such as BRF2, CTCF, MAFK, RAD21 TR4, SMC3 and ZNF143, and the improvements on BDP1 and NRSF are more than 0.1 AUC. There are also 9 TFs on which MTTFsite performs better than the fully-shared method by more than 0.01 AUC, and the improvements on BRF2, RAD21, SMC3 and ZNF274 are more than 0.02 AUC, which is a very large improvement. Table A.4 shows that there are 41 TFs out of the 43 TFs in HepG2 on which MTTFsite performs better than both the fully-shared method and the baseline method. Table A.4 also shows that there are 18 TFs on which MTTFsite performs better than the baseline method by more than 0.01 AUC. The improvements on some TFs are more than 0.05 AUC, such as SMC3 and SP2, and the improvements on CTCF and RAD21 are more than 0.07 AUC. There are also 8 TFs on which MTTFsite performs better than the fully-shared method by at least 0.01 AUC. Table A.5 shows that MTTFsite outperforms both the fully-shared method and the baseline method for 61 TFs out of the 65 TFs in K562. There are four TFs on which the fully-shared method performs worse than the baseline method while MTTFsite performs better than the baseline method. Table A.5 also shows that there are 36 TFs on which MTTFsite performs better the baseline method by more than 0.01 AUC. The improvements on some TFs are more than 0.05 AUC, such as CTCF, SMC3 and YY1, and the improvements on RAD21 and TR4 are more than 0.1 AUC. Note that there are 11 TFs on which MTTFsite performs better than the fully-shared method by more than 0.01 AUC and the improvements on RAD21 and TR4 are more than 0.02 AUC.

In sumarry, by carefully analyzing the comparisons among the basline method, the fully-shared method and MTTFsite on the TFs in the five cell-types, we can draw two conclusions: (1) the training samples of target TFs from multiple cell-types can indeed improve the prediction accuracy for them in each cell-type, which indicates that the TF-DNA interaction in multiple different cell-types indeed have common mechanisms; (2) the fully-shared method performs worse than the baseline method while MTTFsite outperforms the baseline method for some TFs while MTTFsite performs better than it on most TFs in the five cell-types, which indicates that each cell-type also has its private mechanism except the common interaction mechanism among the multiple cell-types.

Comparison between MTTFsite and state-of-the-art methods

The AUCs of MTTFsite and the four existing methods using DNA shape features on five TFs in four cell-types are shown in Table 5.8. It can be observed that MTTFsite outperforms the four existing methods on all the five TFs in the four cell-types except GABP in H1-hESC. The minimum improvement and maximum improvement are 0.022 AUC on GABP in HepG2 and 0.233 AUC on JunD in GM12878, respectively. The average improvement is 0.115 AUC, which is a very big improvement for TFBS prediction.

The AUCs of our proposed MTTFsite and the four existing deep learning methods on

Table 5.8: AUCs of the four DNA shape based methods and MTTFsite on four TFs in five cell-types.

TF	CELL	4-bits	PSSM	TFFM_d	TFFM_f	MTTFsite
CTCF	GM12878	0.763	0.762	0.748	0.750	**0.859**
	H1-hESC	0.762	0.758	0.740	0.744	**0.816**
	HeLa-S3	0.739	0.736	0.724	0.726	**0.834**
	HepG2	0.759	0.757	0.741	0.746	**0.871**
	K562	0.747	0.745	0.731	0.733	**0.839**
GABP	GM12878	0.830	0.828	0.830	0.830	**0.934**
	H1-hESC	**0.832**	0.828	0.824	0.824	0.729
	HeLa-S3	0.796	0.796	0.792	0.787	**0.946**
	HepG2	0.842	0.838	0.830	0.837	**0.864**
	K562	0.822	0.817	0.812	0.815	**0.913**
JunD	GM12878	0.752	0.751	0.742	0.753	**0.975**
	H1-hESC	0.762	0.760	0.750	0.753	**0.876**
	HeLa-S3	0.800	0.797	0.773	0.777	**0.942**
	HepG2	0.774	0.771	0.754	0.757	**0.829**
	K562	0.763	0.760	0.742	0.746	**0.912**
USF2	GM12878	0.773	0.771	0.754	0.758	**0.938**
	H1-hESC	0.784	0.780	0.765	0.770	**0.887**
	HeLa-S3	0.750	0.746	0.731	0.735	**0.938**
	HepG2	0.788	0.784	0.762	0.766	**0.904**

five TFs in four cell-types are listed in TABLE 5.9. TABLE 5.9 shows that MTTFsite performs better than the four methods for 15 out of the 19 pairs. On the 15 pairs, the maximum performance improvement 0.215 AUC on USF2 in HeLa-S3. The average improvement on the 15 pairs is 0.095, which is a prominent improvement and indicates that MTTFsite achieves better performance than state-of-the-art methods.

Prediction for cell-types without training samples

Due to the high cost of the ChIP-seq experiment, most TFs do not have labeled data for certain cell-types. So it is urgent to predict the TFBSs of TFs in the cell-types without labeled data. As MTTFsite can use a common CNN to learn common features by using the available labeled data from multiple cell-types, it can predict the TFBSs for TFs in

Table 5.9: AUCs of MTTFsite and four state-of-the-art methods on TFs in cell-types of humans

TF	CELL	DanQ	DanQ-J	DeepSea	DeepSea-J	MTTFsite
CTCF	GM12878	0.780	0.703	0.745	0.617	**0.895**
	H1-hESC	**0.824**	0.723	0.767	0.656	0.816
	HeLa-S3	0.754	0.670	0.699	0.605	**0.834**
	HepG2	0.826	0.724	0.772	0.644	**0.871**
	K562	0.772	0.687	0.720	0.618	**0.839**
GABP	GM12878	0.929	0.907	0.906	0.895	**0.934**
	H1-hESC	**0.922**	0.906	0.907	0.894	0.729
	HeLa-S3	0.808	0.772	0.766	0.752	0.946
	HepG2	**0.927**	0.914	0.913	0.906	**0.864**
	K562	0.911	0.898	0.900	0.892	**0.913**
JunD	GM12878	0.835	0.779	0.789	0.729	**0.975**
	H1-hESC	0.771	0.718	0.726	0.699	**0.876**
	HeLa-S3	0.850	0.721	0.766	0.671	**0.942**
	HepG2	**0.842**	0.725	0.765	0.697	0.829
	K562	0.717	0.652	0.664	0.624	**0.912**
USF2	GM12878	0.789	0.706	0.710	0.689	**0.938**
	H1-hESC	0.849	0.775	0.780	0.758	**0.887**
	HeLa-S3	0.723	0.637	0.644	0.609	**0.938**
	HepG2	0.811	0.691	0.693	0.668	**0.904**

the cell-types without labeled data, which is referred to as cross-cell-type predictions. In MTTFsite, the private CNN for each cell-type need to be trained by the labeled data in the cell-type. As the private CNNs for the cell-types without labeled data do not have training data, MTTFsite trains them by leveraging on the training data from cell-types with available labeled data. Thus, by comparing MTTFsite and the full-shared model, we find that the private CNNs for the cell-types without labeled data are similar to the common CNN in the fully-shared model, because they both are trained by the combined training data from the cell-types with available labeled data. The only difference is that MTTFsite contains both the features learned by private CNNs and that learned by the common CNN while the fully-shared model contains only the features learned by the common CNN. We have noted that if some cell-types contains too much training data, the feature space learned by the fully-shared model is overoccupied by private features such that many common features are lost. As MTTFsite can separate private features from common features, the lost common features in the private CNNs can be complemented by the common features learned by the common CNN. Therefore, the features learned by MTTFsite for each cell-type contains more common features than that learned by the fully-shared model.

In order to evaluate the cross-cell-type prediction performance of MTTFsite on a cell-type, we suppose that only the test set for the cell-type is available and both the training set and the validation set are unavailable. So the fully-shared model and MTTFsite need to be trained and validated by the combined training data and the combined validation data from the other cell-types with available labeled data, respectively. In addition, we also compare MTTFsite with the baseline method in cross-cell-type TFBS prediction. The baseline method is similar to the fully-shared model except that the baseline method is trained by the training data of only the target cell-type. So the baseline method is a supervised method, in which the training set and validation set come from the same cell-type with the test set.

The comparison among the baseline method, the fully-share model and our proposed MTTFsite is shown in Figure 5.6. Figure 5.6A shows that the fully-shared model performs better than the baseline method for most (cell-type,TF) pairs. Figure 5.6D shows that the first quartile, the median and the third quartile of the AUC for the fully-shared model are higher than that of the baseline model. It shows that the fully-shared model trained by cross-cell-type can achieve better performance the baseline method trained by the target cell-type. It indicates that the fully-shared model can achieve good performance for cross-cell-type TFBS predictions. Figure 5.6B and Figure 5.6C show that MTTFsite performs better than both the baseline method and the fully-shared model for most (cell-type,TF) pairs. Figure 5.6D shows that the first quartile, the median and the third quartile of the AUC for MTTFsite are higher than that of both the baseline method and the full-shared

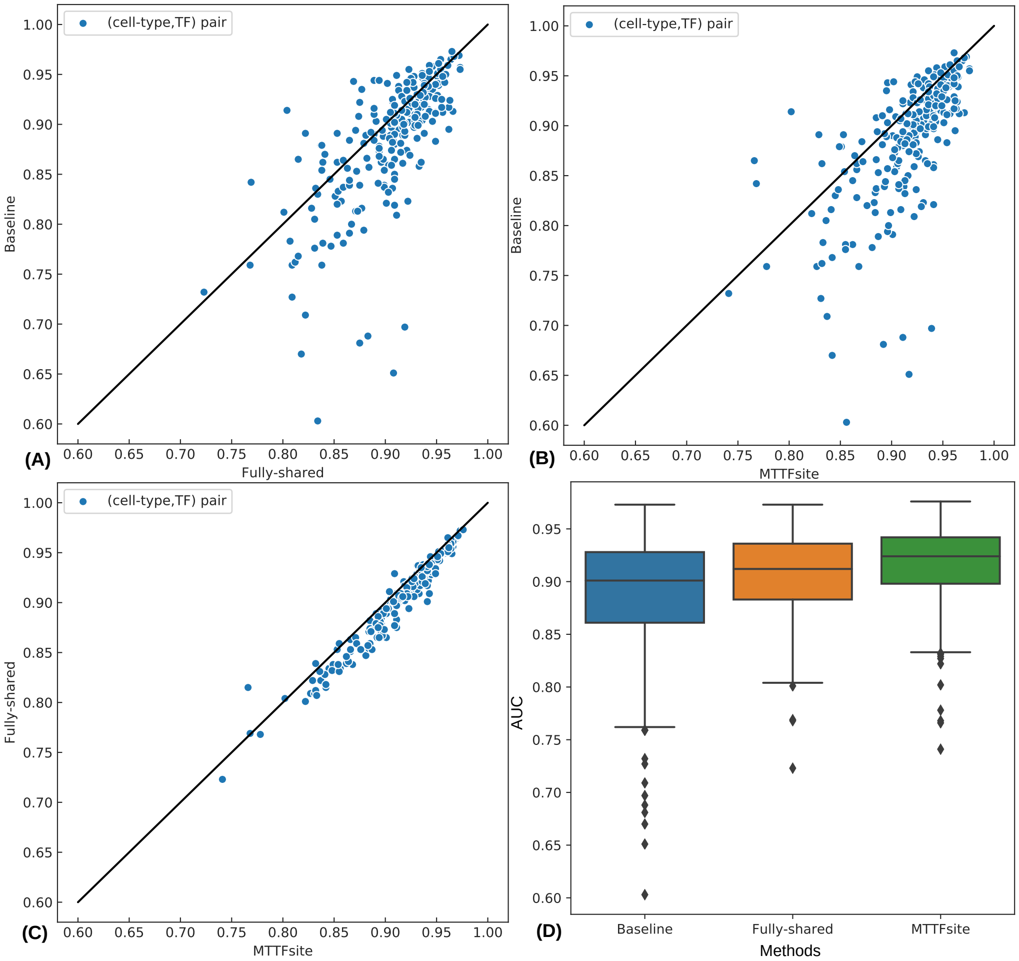


Figure 5.6: The AUCs for the cell-types without labeled data.

model. More specifically, the comparison among the baseline method, the fully-shared model and MTTFsite for GM12878, H1-hESC, HeLa-S3, HepG2 and K562 are listed in Table A.6, A.7, A.8, A.9 and A.10, respectively, where bold represents the best performance. We first compare MTTFsite with the baseline method in detail. For the 56 TFs in GM12878, Table A.6 shows that there are 46 TFs on which MTTFsite performs better than the baseline method. The maximum and the average improvement are 0.266 and 0.042 AUC, respectively. For the 42 TFs in H1-hESC, Table A.7 shows that there are 31 TFs on which MTTFsite performs better than the baseline method. The maximum and the average improvement are 0.211 and 0.063 AUC, respectively. For the 37 TFs in HeLa-

S3, Table A.8 shows that there are 29 TFs on which MTTFsite performs better than the baseline method. The maximum and the average improvement are 0.172 and 0.034 AUC, respectively. For the 43 TFs in HepG2, Table A.9 shows that there are 35 TFs on which MTTFsite performs better than the baseline method. The maximum and the average improvement are 0.253 and 0.042 AUC, respectively. For the 63 TFs in K562, Table A.10 shows that there 54 TFs on which MTTFsite performs better than the baseline method. The maximum and the average improvement are 0.242 and 0.034 AUC, respectively. It indicates that MTTFiste trained by cross-cell-type can achieve better performance than the baseline method trained by the target cell-type.

Then we compare MTTFsite with the fully-shared model in detail. For the 56 TFs in GM12878, Table A.6 shows that there are 54 TFs on which MTTFsite performs better than the fully-shared model. The improvements on NRSF, SMC3, RAD21, YY1, NFE2 and BCL11A are more than 0.02 AUC, and the improvements on CTCF and EZH2 are even more than 0.04 AUC. For the 42 TFs in H1hesc, Table A.7 shows that there are 36 TFs on which MTTFsite outperforms the fully-shared model. The improvements on RFX5, NRSF, RXRA, TCF12 and ZNF143 are more than 0.02 AUC, and the improvements on RAD21 and MAFK are even more than 0.03 AUC. For the 37 TFs in HeLa-S3, Table A.8 shows that there are 33 TFs on which MTTFsite performs better than the fully-shared model. The improvements on SMC3, RAD21, MAFK and NRSF are more than 0.02 AUC, and the improvements on CTCF is more than 0.04 AUC. For the 43 TFs in HepG2, Table A.9 shows that there are 41 TFs on which MTTFsite performs better than the fully-shared model. The improvements on CTCF, JUND, MYC, CEBPB, GABP, SMC3, ZBTB33, MAFF, RAD21 and MAFK are more than 0.02 AUC, and the improvement on NRSF is even more than 0.04 AUC. For the 63 TFs in K562, Table A.10 shows that there are 62 TFs on which MTTFsite performs better than the fully-shared model. The improvements on NFE2, RFX5, ZBTB33, MAFK, NRSF and ZNF143 are more than 0.02 AUC, and improvements on CTCF, RAD21 and SMC3 are more than 0.03 AUC. It indicates that MTTFsite can also performs better than the fully-shared model for cross-cell-type predictions.

In summary, MTTFsite can predict TFBSs of target TFs for a large number of cell-type without training samples by using training samples from multiple other cell-types. However, many target TFs do not have any training sample in any of the cell-types, thus MTTFsite cannot be applied to predict TFBSs of target TFs in any of the cell-types.

5.4 PDBR_TF:Predicted DNA binding residue based method

In above section, we presented a cross-cell-type TFBS prediction method MTTFsite based on multi-task learning. MTTFsite can predict TFBSs of target TFs in specific cell-types without training samples by using the training samples from multiple other cell-types. However, many TFs do not have any training sample in any of the cell-types. Thus, MTTFsite cannot be applied to predict TFBSs for these TFs. Fortunately, we know that in a specific cell-type, there exist other TFs which have TFBSs identified by experimental methods. Even though a majority of TFs have different sequences and biology functions, some TFs do have similar sequences and biology functions. As these TFs are similar in sequences and biology functions and tend to bind to similar positions of the genome, we propose PDBR_TF to obtain features for TFs without training data by using experimentally identified TFBSs of other TFs from the same cell-type.

5.4.1 Cross-TF TFBS prediction

A simple method for cross-TF TFBS prediction is to train a model for a target TF in a specific cell-type by assembling the training samples from multiple other TFs with training samples in the sample cell-type. Since CNN_TF have achieved good performance for TFBS prediction, we can train a CNN_TF model for a target TF in a specific cell-type by assembling the training samples from multiple TFs in the same cell-type, which is considered as the baseline method for this problem. By carefully analyzing the baseline method, we found that the baseline method treats the training samples from different TFs equally in training model. They deem that the training samples from different TFs will provide same contributions in the training, which is an inappropriate hypothesis. In factor, different TFs have different similarity to the target TF in sequences and biology functions and TFs with higher similarity in sequences and biology functions tend to bind to same positions in a genome. Thus, the TFs with higher similarity with the target TF tend to have larger contribution for the training than the TFs with lower similarity with the target TF.

In order to incorporate information of TFs into prediction model, we present a sequence based method in which the sequence features of TFs and The putative TFBS are combined as input features. In the sequence based method, the network topology in CNN_TF is used to learn sequence features from DNA sequence and a LSTM network is used to learn sequence features from the sequence of the corresponding TFs, and then the features of a DNA sequence and the features of the corresponding TF are concatenated into a feature vector. The reason for why LSTM network is used to learn sequence features for TFs is that LSTM network is capable of automatically capturing both local context and long-range dependency among residues in a sequence. The sequence of TFs contain hundreds or thousands of residues and multiple residues often participate combinatorially in the TF-DNA interaction, thus there may exist many local context and long-range relationships in the sequence of TFs. Finally, the concatenated feature vectors are fed into a fully connected network to learn condensed feature vectors. In the fully connected network, the features from a TF and that from a putative TFBS are blended together. In factor, the features learned from TFs by LSTM can determine the contributions of the training samples from different TFs in the training process.

The sequence of TFs usually contains a larger number of residues, which makes the LSTM network be very time consuming. For example, BDP1 contains as many as 2624 residues, which will consume much time. Furthermore, a TF may also contains several other domains except a DNA binding domain. Other domains include many types, such as

Death effector domain (DED), Immunoglobulin-like domains, Phosphotyrosine-binding domain (PTB), Pleckstrin homology domain (PH), Src homology 2 domain (SH2), etc.. These domains play other roles like allowing protein–protein binding, playing roles in the immune system, binding to phosphorylated tyrosine residues, binding phosphoinositides with high affinity, binding to phosphorylated tyrosine, etc.. Since the residues in other domains are not involved in TF-DNA interaction, they may affect TFBS prediction. DNA binding domain is an independently folded protein domain that contains at least one structural motif that recognizes double- or single-stranded DNA. A DNA binding domain can recognize a specific DNA sequence (TF binding site) or have a general affinity to DNA. DNA binding domains include many DNA binding residues in their folded structure and these residues play important roles in TF-DNA interaction. Therefore, we aim to make good use of DNA binding residues in TF for cross-TF TFBS prediction.

As experimental methods for identification of DNA binding residues require known tertiary structures for both TF and DNA and only a limited number of TFs have known tertiary structures, we aim to use DNA binding residues predicted by computational methods to complete for cross-TF TFBS prediction. We have proposed four computational methods for DNA binding residue prediction, but the lacking of structure features for most TFs make only EL_PSSM-RT and CNNsite be suitable for our TFBS prediction. Moreover, we have collected enough number of training samples for DNA binding residue prediction, so the binding residues predicted by our proposed CNNsite are used to build our proposed cross-TF TFBS prediction method PDBR_TF.

5.4.2 Predicted DNA binding residue based method (PBDR_TF)

The framework of PDBR_TF is shown in Figure 5.7. PDBR_TF contains three main parts. The first part is a LSTM network followed by a layer normalization layer. The input of this part is the feature matrix concatenated by the one-hot vectors of all predicted binding residues within the corresponding TF. The second part is a CNN_TF model, which is used to learn feature vectors for putative TFBSs. The CNN_TF model contains two convolution layers followed by a pooling layer and a layer normalization layer. One convolution layer inputs the feature vector concatenated by the one-hot vectors of the nucleotides within the target putative TFBS while the other convolution layer inputs the feature vector concatenated by the histone modification features of the target putative TFBS. The last part is a fully connected network followed by a softmax classier. The fully connected network contains three hidden layer and a dropout layer used to avoid over-fitting before the first hidden layer. The fully connected network inputs the feature vector concatenated by the features outputted by the LSTM network and that outputted by the CNN_TF and outputs a label for indicating whether the inputting putative TFBS is an actual TFBS or not.

5.4.3 Experiments and results

The performance of PDBR_TF is measured by AUC. We compare PDBR_TF with two the methods including the baseline method and the sequence method. The difference between PDBR_TF and the baseline method is that PDBR_TF has a LSTM applied on the amino acid sequence of corresponding TFs except the network topology in CNN_TF. The different between PDBR_TF and the sequence method is that PDBR_TF applies a LSTM on the amino acid sequence composed by only predicted DNA binding residues within corresponding TFs while the sequence method applies a LSTM on the amino acid sequence composed by all the residues of corresponding TFs. Four sets of evaluations are conducted here. The first set compares the performance between the baseline method and the sequence method. The second set compares the performance between the sequence method and our proposed PDBR_TF. The third set compares the performance between our proposed cross-TF TFBS prediction method PDBR_TF with our proposed cross-cell-type TFBS prediction method. In this section, the datasets used to evaluate our proposed method will be introduced first. Then, details of the three components of our proposed PDBR_TF will be described in sequence.

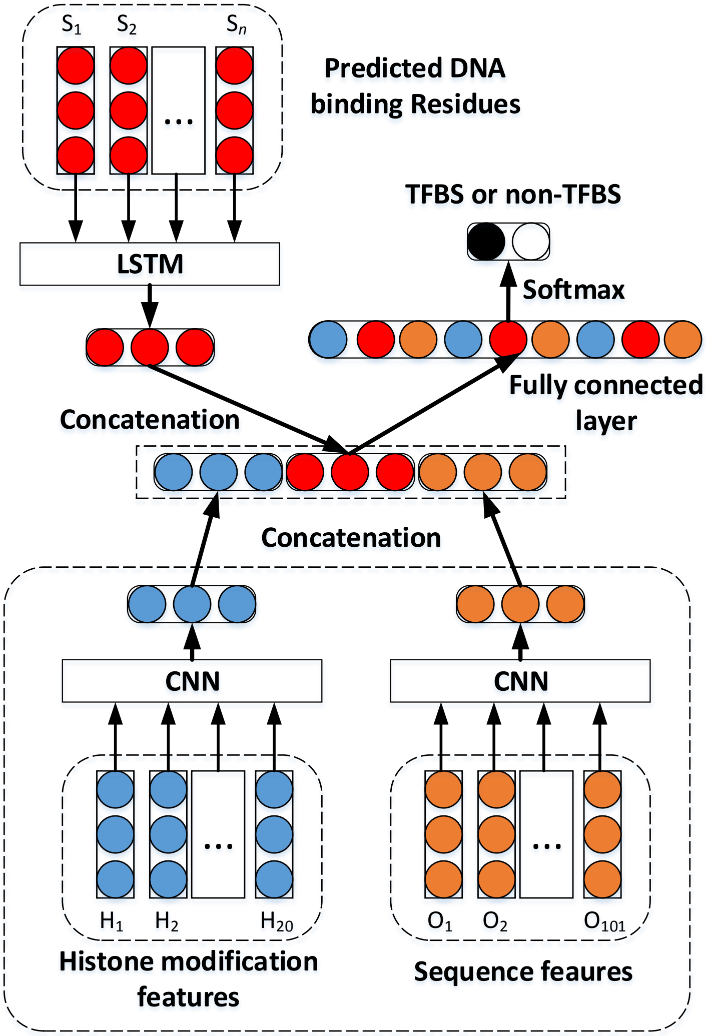


Figure 5.7: The framework of PDBR_TF.

Datasets

Five cell-types including GM12878, H1-hESC, HeLa-S3, HepG2 and K562 are used to evaluate our proposed method. A total of 132 TFs are used to evaluate PDBR_TF, where 55, 30, 17, 14, 16 TFs have labeled data in one cell-types, two cell-types, three cell-types and four cell-types and all the five cell-types, respectively. The TFBSs of these TFs have been identified by ChIP-seq experiments and their peak lists can be downloaded from ENCODE freely. The obtained peaks are usually provided in one of two formats: **narrow peak** and **broad peak**. The narrow peak format, which requires technically more sophisticated equipments to get, can provide more accurate positions for TFBSs than the broad peak format. So the narrow peak format is used if available, otherwise the broad peak format is used. Based on works [5], the TFBS at each peak is defined as a 101 bp sequence by taking the midpoint of the peak as the center. Contrast to TFBSs, the non-TFBSs of a TF are defined as 101 bp DNA regions unbound by the target TF. In this study, as TFBSs are encoded by both histone modification features and DNA sequences, we construct non-TFBSs by extracting actual DNA fragments. A non-TFBS is constructed for each TFBS by selecting a 101 bp DNA fragment with more than 98% dinucleotide similarity to the TFBS and nonoverlapping with any TFBS. Thus, we can construct the same number of non-TFBSs as TFBSs for each TF. For each TF in each cell-type, the labeled data are divided into 3 separate, but equal size folds: one fold for training, one fold for validation and one fold for test.

The performance of the sequence method

We first evaluate the performance of the sequence method on the TFs in the five cell-types and compare it with the baseline method. For each TF in each cell-type, the sequence method and the baseline method are trained by the training set of other TFs, validated and tested by the validation set and the test set of the target TF, respectively.

The comparison between the baseline method and the sequence method is shown in Figure 5.8. Figure 5.8A shows that the sequenc method achieves lower performance than the baseline method for most (cell-type,TF) pairs, where a (cell-type,TF) pair refers to a prediction task for a TF in a cell-type. Figure 5.8D shows that the first quartile, the median and the third quartile of AUC for the sequence method are higher than that of the baseline method. More specifically, the comparison between the baseline method and the sequence method on the TFs in GM12878, H1-hESC, HeLa-S3, HepG2 and K562 is shown in table B.1, B.2, B.3, B.4 and B.5, respectively. Table B.1 shows that there are 42 TFs out of the 69 TFs in GM12878 on which the baseline method performs better than the sequence method and the baseline method performs worse than the sequence method on the other 25 TFs. Table B.2 shows that the baseline method performs better than the sequence method on 37 TFs out of the 46 TFs and performs worse than the sequence method on the other 9 TFs. Table B.3 shows that the baseline method performs better than the sequence method on 27 TFs of the 46 TFs in HeLa-S3 and performs worse than the sequence method on the other 19 TFs. Table B.4 shows that the baseline method outperforms the sequence on 33 TFs out of the 52 TFs in HepG2 and performs worse than the sequence method on the other 19 TFs. Table B.5 shows that the baseline method outperforms that sequence method on 49 TFs out of the 88 TFs in K562 and performs worse than the sequence method on the other 39 TFs.

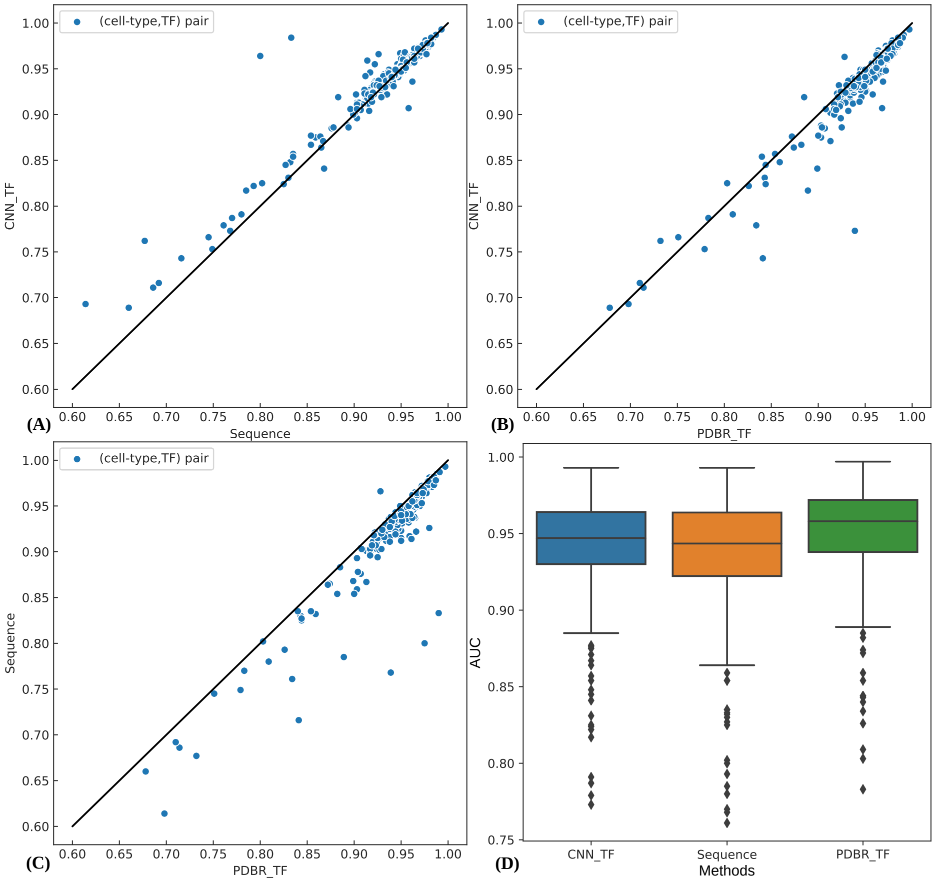


Figure 5.8: The comparisons between CNN_TF, the sequence method and PDBR_TF.

By analyzing the above five tables, we found that when the sequence features of corresponding TFs are incorporated into prediction, the performance is declined. The reason may be that a TF may contain multiple different protein domains. Different types of protein domains often involved in different biology functions, for example, Basic Leucine zipper domain (bZIP domain) and Zinc finger DNA binding domain (ZnF_GATA) are two examples of the domains involved in DNA binding contains and Death effector domain (DED) allows protein–protein binding by homotypic interactions (DED-DED), and Phosphotyrosine-binding domain usually binds to phosphorylated tyrosine residues. When incorporating the sequence features from the whole amino acid sequence of corresponding TFs into prediction, the extracted features not only contain the information from domains involved in DNA binding, but also contain the information from domains involved in other functions. With the incorporating of the information involved in other functions, the incorporated features will affect the performance of the sequence method. Moreover, as the amino acid sequence of TFs usually contains a large number of residues, applying a LSTM network on the amino acid sequence of TFs to extract features costs plenty of time and much computer resources.

The performance of PDBR_TF

The comparison among our proposed PDBR_TF, the sequence method and the baseline method are shown in Figure 5.8. Figure 5.8B and Figure 5.8C shows that PDBR_TF performs better than both the baseline method and the sequence method for most (cell-type,TF) pairs. Figure 5.8D shows that the first quartile, the median and the third quartile of the AUC for PDBR_TF are higher than that of the baseline method and the sequence method. More specifically, the comparison among our proposed PDBR_TF, the baseline method and the sequence method on the TFs in GM12878 is shown in table B.1. Table B.1 shows that PDBR_TF performs better than both the baseline method and the sequence method on 65 TFs out of the 69 TFs in GM12878. TFs with improvement more than 0.01

Table 5.10: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 23 TFs in GM12878.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
GABP	0.948	0.945	**0.962**	TCF3	0.949	0.946	**0.961**
ELK1	0.937	0.962	**0.971**	RUNX3	0.959	0.963	**0.975**
ELF1	0.943	0.940	**0.953**	SIX5	0.944	0.949	**0.965**
RFX5	0.950	0.947	**0.961**	USF2	0.959	0.958	**0.970**
ATF2	0.965	0.980	**0.987**	POU2F	0.959	0.962	**0.972**
BCL3	0.917	0.917	**0.928**	NFE2	0.938	0.937	**0.954**
SMC3	0.969	0.968	**0.979**	ZZZ3	0.848	0.833	**0.860**
NFATC1	0.907	0.959	**0.970**	BATF	0.965	0.970	**0.979**
BHLHE40	0.957	0.959	**0.972**	SRF	0.936	0.929	**0.943**
SP1	0.963	0.964	**0.976**	EGR1	0.939	0.937	**0.950**
USF1	0.926	0.930	**0.953**	FOS	0.961	0.966	**0.974**
NFYB	0.875	0.860	**0.904**				

AUC are listed Table 5.10, which shows that there 23 TFs on which PDBR_TF performs better the baseline method by more than 0.01 AUC. The table also shows that the improvements for some TFs are more than 0.02 AUC, such as ELK1, ATF2, USF1, NFYB and SIX5, and the improvement on NFATC1 is more than 0.06 AUC. There are 18 TFs on which PDBR_TF performs better the sequence method by more than 0.01 AUC. The improvements on USF1 and ZZZ3 are more than 0.02 AUC and the improvement on NFYB is more than 0.04 AUC. The comparison between our proposed PDBR_TF, the baseline method and the sequence method on the TFs in H1-hESC is shown in table B.2, which shows that PDBR_TF performs better than both the baseline method and the sequence method on 45 TFs out of the 46 TFs in H1-hESC. The TFs with improvement more than 0.01 AUC are listed in table 5.11, which shows that improvements on 26 TFs out of the 46 TFs in H1-hESC are more than 0.01 AUC. This table also shows that the improvements for some TFs are more than 0.02 AUC, such as TEAD4, EZH2, FOSL1, EGR1, POU2F, SIX5, RXRA and USF1. On the 26 TFs, PDBR_TF also performs better than the sequence method by 0.01 AUC and the improvement on 19 TFs are more than 0.02 AUC.

Table 5.11: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 26 TFs in H1-hESC.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.935	0.924	**0.951**	MAFK	0.867	0.855	**0.883**
SP4	0.936	0.931	**0.950**	JUND	0.937	0.930	**0.953**
SUZ12	0.832	0.831	**0.845**	NANOG	0.945	0.932	**0.958**
MXI1	0.937	0.934	**0.950**	NRF1	0.887	0.879	**0.905**
TEAD4	0.913	0.915	**0.938**	CEBPB	0.791	0.781	**0.810**
EZH2	0.888	0.894	**0.905**	POU2F	0.948	0.949	**0.970**
FOSL1	0.878	0.855	**0.902**	SIX5	0.925	0.916	**0.952**
RFX5	0.865	0.865	**0.876**	GABP	0.922	0.912	**0.933**
ATF3	0.949	0.938	**0.966**	RXRA	0.928	0.925	**0.949**
BACH1	0.919	0.913	**0.936**	MAX	0.930	0.925	**0.947**
MYC	0.932	0.926	**0.949**	HDAC2	0.946	0.940	**0.959**
RAD21	0.954	0.951	**0.971**	P300	0.942	0.936	**0.955**
EGR1	0.885	0.877	**0.908**	USF1	0.897	0.903	**0.920**

The improvement on FOSL1, EGR1, SIX5 are more than 0.04 AUC. The comparison between our proposed PDBR_TF, the baseline method and the sequence method on the TFs in HeLa-S3 is shown in table B.3. Table B.3 shows that there are 43 TFs out of the 46 TFs in HeLa-S3 on which the PDBR_TF performs better than both the baseline method and the sequence method. The TFs with improvement more than 0.01 AUC are listed in table 5.12, which shows than there are 25 TFs out of the 46 TFs in HeLa-S3 on which the improvement of our proposed PDBR_TF over the baseline method are more than 0.01 AUC. This table also shows that the improvements on NFYA, CEBPB and BRF1 are more than 0.02 AUC, which are very larger improvements. There are 22 TFs on which PDBR_TF performs better than the sequence method by more than 0.01 AUC. The improvement on CTCF, ZNF143, USF2 and CEBPB are more than 0.02 AUC and the improvement on BRF1 is more than 0.03 AUC.

The comparison between our proposed PDBR_TF, the baseline method and the sequence method on the TFs in HepG2 is shown in table B.4. This table shows that there are 49 TFs out of the 52 TFs in HepG2 on which PDBR_TF performs better than both the

Table 5.12: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 25 TFs in HeLa-S3.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.954	0.951	**0.967**	E2F1	0.902	0.902	**0.915**
CTCF	0.913	0.903	**0.930**	ZNF143	0.925	0.917	**0.939**
AP2A	0.949	0.947	**0.963**	MAZ	0.929	0.926	**0.940**
NRF1	0.911	0.908	**0.921**	MXI1	0.959	0.955	**0.967**
ELK4	0.957	0.954	**0.968**	USF2	0.945	0.938	**0.965**
AP2G	0.945	0.941	**0.955**	CEBPB	0.922	0.935	**0.959**
NFYA	0.927	0.929	**0.947**	BRF1	0.860	0.868	**0.900**
BAF170	0.964	0.970	**0.976**	IRF3	0.954	0.954	**0.968**
E2F6	0.926	0.922	**0.936**	SMC3	0.961	0.965	**0.975**
TBP	0.971	0.974	**0.981**	INI1	0.931	0.927	**0.941**
MAX	0.952	0.950	**0.963**	MYC	0.952	0.955	**0.966**
RAD21	0.959	0.961	**0.973**	BRCA1	0.960	0.961	**0.972**
NFYB	0.901	0.900	**0.918**				

Table 5.13: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 20 TFs in HepG2.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
ZBTB33	0.955	0.923	**0.966**	HNF4G	0.968	0.973	**0.981**
FOXA2	0.967	0.973	**0.982**	BHLHE40	0.952	0.950	**0.964**
CTCF	0.941	0.947	**0.953**	HNF4A	0.975	0.979	**0.986**
MAZ	0.942	0.932	**0.952**	ELF1	0.947	0.937	**0.957**
NRF1	0.927	0.911	**0.940**	MXI1	0.946	0.918	**0.960**
CEBPB	0.951	0.954	**0.961**	NRSF	0.780	0.761	**0.836**
FOXA1	0.966	0.927	**0.981**	USF2	0.950	0.952	**0.969**
GABP	0.911	0.909	**0.926**	RXRA	0.969	0.974	**0.979**
SP2	0.773	0.768	**0.941**	MAX	0.943	0.938	**0.963**
MYC	0.967	0.968	**0.978**	USF1	0.920	0.929	**0.948**

Table 5.14: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 35 TFs in K562.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.930	0.933	**0.946**	ELK1	0.957	0.958	**0.968**
JUND	0.962	0.964	**0.972**	SIN3A	0.925	0.926	**0.944**
CTCF	0.935	0.936	**0.949**	JUN	0.973	0.973	**0.985**
ZBTB33	0.906	0.908	**0.920**	MAFK	0.967	0.977	**0.980**
TEAD4	0.958	0.964	**0.968**	BHLH40	0.940	0.942	**0.952**
KAP1	0.817	0.786	**0.891**	MAZ	0.943	0.946	**0.954**
TAL1	0.951	0.955	**0.964**	SIX5	0.937	0.941	**0.956**
ATF3	0.933	0.925	**0.945**	CEBPB	0.824	0.825	**0.845**
ZNF274	0.754	0.749	**0.780**	USF2	0.962	0.965	**0.974**
RAD21	0.976	0.974	**0.986**	P300	0.886	0.895	**0.926**
NFYB	0.904	0.917	**0.933**	NFYA	0.940	0.941	**0.961**
NFE2	0.988	0.987	**0.992**	NRSF	0.872	0.867	**0.915**
SMC3	0.965	0.964	**0.978**	E2F4	0.930	0.929	**0.940**
MAFF	0.948	0.953	**0.973**	TR4	0.906	0.897	**0.918**
SP2	0.938	0.935	**0.952**	MAX	0.949	0.952	**0.960**
SRF	0.930	0.934	**0.941**	SP1	0.948	0.950	**0.959**
FOS	0.964	0.968	**0.976**	GABP	0.933	0.936	**0.948**
USF1	0.932	0.943	**0.952**				

baseline method and the sequence method. The TFs with improvements of more than 0.01 AUC are listed in table 5.13, which shows that the improvement on 20 TFs out of the 52 TFs in HepG2 are more than 0.02 AUC. The table also shows that the improvement on MAX, USF1 and NRSF are more than 0.02 AUC and improvement on SP2 is even more than 0.16 AUC, which are very large improvements. There are 14 TFs on which PDBR_TF performs better than the sequence method by more than 0.01 AUC and improvement on 9 TFs are more than 0.02 AUC. The improvement on ZBTB33, FOXA1 and NRSF are more than 0.04 AUC and the improvement on SP2 is more than 0.17 AUC. The comparison between our proposed PDBR_TF, the baseline method and the sequence method on the TFs in K562 is shown in table B.5. This table shows that there are 81 TFs out of the 88 TFs in K562 on which our proposed PDBR_TF performs better than both the baseline method and the sequence method. The TFs with more than 0.01 AUC improvements are listed in table 5.14, which shows that there are 35 TFs out of the 88 TFs in K562 on which the PDBR_TF performs better than the baseline method by more than 0.01 AUC and improvement on 9 TFs are more than 0.02 AUC. The improvements on P300 and NRSF are more than 0.04 AUC, and the improvement on KAP1 is even more than 0.07 AUC, which are very large improvements. There are 22 TFs on which PDBR_TF performs better than the sequence method by more than 0.01 AUC, and the improvement on 9 TFs are more than 0.02 AUC. The improvement on P300 and NRSF are more than 0.03 AUC, and the improvement on KAP1 are even more than 0.1 AUC. The above analysis for the TFS in the five cell-types shows that PDBR_TF performs better than both the baseline method and the sequence method with large margins for most TFs in all the five cell-types. It indicates that the features learned by LSTM from the amino acid sequence composed by predicted DNA binding residues indeed play a important role in TFBS prediction.

As the difference between our proposed PDBR_TF and the baseline method is that PDBR_TF incorporates features learned by LSTM from the amino acid sequence composed by predicted DNA binding residues, the outperformance of PDBR_TF over the baseline method is attributed by the features learned from predicted DNA binding residues. Moreover, the difference between PDBR_TF and the sequence method is that the features of TFs in PDBR_TF are learned from predicted DNA binding residues while that in the sequence method are learned from all the residues. So the outperformance of PDBR_TF over the sequence method is attributed by the predicted DNA binding residues instead of a entire TF. Contrast to a entire TF containing residues involved in different functions, the predicted DNA binding residues contain only residues involved in DNA binding function. So the predicted DNA binding residues are more suitable to measure the similarity between two TFs in terms of the DNA binding function than a entire TF. As the cross-TF TFBS prediction model including PDBR_TF, the baseline model and the sequence model for a target TF are trained by the experimentally identified TFBSs from multiple other TFs and different TFs have different structure and function similarity with the target TF, the training samples of different TFs have different contributions for the model training. However, the baseline method treats the training samples from different TFs equally and deem the training samples from different TFs to have equal contribution for training. Even if the sequence method treats the training samples in different TFs unequally, it just uses the features learned by LSTM from a entire TF to determine the contributions of their TFBSs in training. Contrast to the sequence method, our proposed PDBR_TF applies the features learned by LSTM from only predicted DNA binding residues. Therefore, PDBR_TF can determine the contribution of the training samples from a TF more accurately by using only features involved in DNA binding function.

Comparison between PDBR_TF and MTTFsite

MTTFsite is our proposed cross-cell-type TFBS prediction method, which can predict TFBSs of target TFs for specific cell-types without training samples by using the training samples of the target TFs from multiple other cell-types. In order to compare the performance between our proposed cross-TF TFBS prediction method PDBR_TF and cross-cell-type TFBS prediction method MTTFsite, we apply PDBR_TF and MTTFsite on the TFs in the five cell-types. For a target TF in a specific cell-type, PDBR_TF uses the experimentally identified TFBSs of the target TF in the specific cell-type in test set and the TFBSs of other TFs in the same cell-type in training set. MTTFsite uses the experimentally identified TFBSs of the target TF in the specific cell-type in test set and the TFBSs of the target TF in other cell-types in training set. For PDBR_TF and MTTFsite, 10% samples of the training set are left out as validation set. As there are 55 out of the 69 TFs in GM12878, 42 out of the 46 TFs in H1-hESC, 37 out of the 46 TFs in HeLa-S3, 42 out of the 52 TFs in HepG2 and 60 out of the 88 TFs in k562, on which both PDBR_TF and MTTFsite can be evaluated, we compare them on the these TFs for the five cell-types.

The comparisons between PDBR_TF and MTTFsite for the TFs in the five cell-types are listed in Figure 5.9. Figure 5.9A shows that PDBR_TF performs better than MTTF-

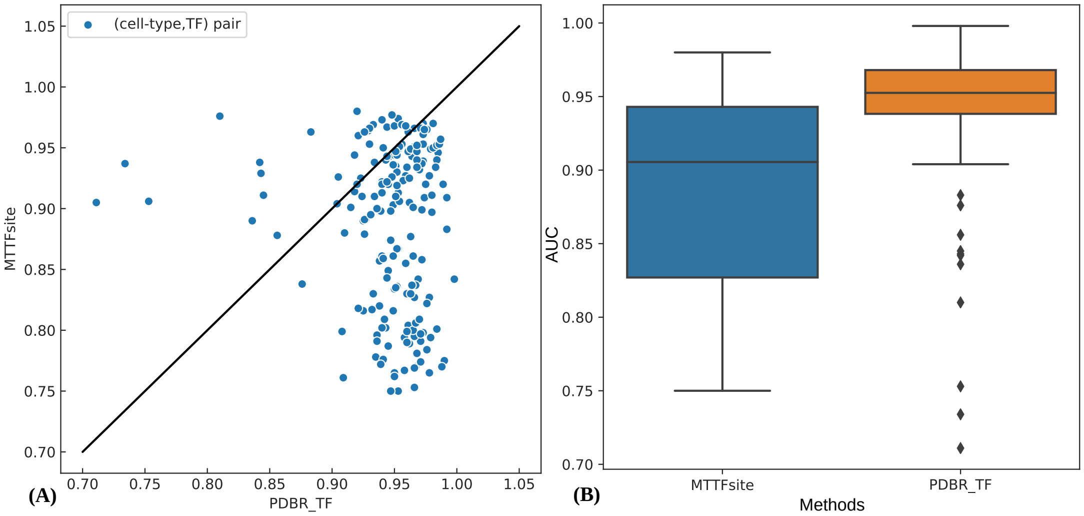


Figure 5.9: The comparisons between our proposed MTTFsite and PDBR_TF.

site for most (cell-type,TF) pairs. Figure 5.9B shows that the first quartile, the median and the third quartile of the AUC for PDBR_TF are higher than that of MTTFsite. More specifically, PDBR_TF achieves higher AUC than MTTFsite for 206 pairs out of the 237 common (cell-type,TF) pairs between them. The largest improvement and average improvement are 43.9% and 14.7%, respectively, which is a very large improvement. It indicates that PDBR_TF is more useful for the TFBS prediction of TFs without labeled data than MTTFsite.

The comparisons on the 55 TFs in GM12878 shows that PDBR_TF outperforms MTTFsite on 52 out of the 55 TFs in GM12878. The improvement on 49 TFs are more than 0.02 AUC. The comparisons on the 42 TFs in H1-hESC shows that PDBR_TF outperforms MTTFsite on 35 out of the 42 TFs in H1-hESC and the improvements on 33 TFs are mores than 0.02 AUC. The comparisons on the 37 TFs in HeLa-S3 shows that PDBR_TF outperforms MTTFsite on 33 out of the 37 TFs in HeLa-S3 and the improvement on 31 TFs are more than 0.02 AUC. The comparisons on the 42 TFs in HepG2 shows that PDBR_TF outperforms MTTFsite on 35 out of the 42 TFs in HepG2 and the improvement on 33 TFs are more than 0.02 AUC. The comparisons on the 60 TFs in K562 shows that PDBR_TF outperforms MTTFsite on 48 out of the 60 TFs in K562 and the improvement on 41 TFs are more than 0.02 AUC.

There are totally 31 TFs, on which MTTFsite outperforms PDBR_TF, for all the five cell-types. Out of the 31 TFs, there are 16 TFs with training samples in all the five cells, 11 TFs with training samples in four cells, 4 TFs with training samples in 3 cells. It indicates that the 31 TFs have training samples for at least 3 cell-types and 27 TFs out of them have training samples for at four cell-types. So the higher performance of MTTFsite over PDBR_TF is attributed to the enough number of cell-types with training samples for target TFs. Therefore, we conclude that the performance of MTTFsite closely relies on the number of cell-types with training samples for target TFs. Moreover, in the five cell-types, there are totally 84 TFs with training samples for only one cell-types. MTTFsite cannot be applied to predict TFBSs for these TFs because these TFs have only test cell-types and do not have training cell-types from which the training samples are achieved. By contrary, for target TFs in a specific cell-types, there are many other TFs with training samples in the same cell-type, our our proposed cross-TF prediction method PDBR_TF is suitable to be applied to predict TFBSs for these target TFs. Therefore, our proposed cross-TF prediction method PDBR_TF and cross-cell prediction method MTTFsite can applied for different cases. MTTFsite is suitable to be applied for target TFs with training samples in multiple other cell-types while PDBR_TF is suitable to be applied for target TFs in specific cell-types with training sample for multiple other TFs.

5.5 Chapter Summary

In this chapter, three novel methods are proposed for TFBS prediction. Firstly, the proposed novel CNN_TF method aims to capture both first order and higher order dependencies for those TFs with sufficiently large number of training samples for a specific cell-type. CNN_TF uses both sequence features and histone modification features using

CNN for prediction. Performance evaluation shows that higher order dependencies is a very useful feature for TF binding site prediction, and CNN_TF outperforms the current state-of-the-art methods with 0.051 AUC improvement. Secondly, the proposed MTTFsite is a novel cross-cell-type TFBS prediction method to address data sparseness issue for a specific cell-type. MTTFsite leverages on training samples of TFs available in other cell types. Lastly, the proposed novel PDBR_TF method is the first attempt to address the non-available data issue through a cross-TF TFBS prediction method such that TF prediction can be done even if there is no TF training sample in any of the cell-types. PDBR_TF leverages on training samples available for other TFs of the same cell-type. Evaluation on TFs in five cell-types demonstrates that both MTTFsite and PDBR_TF can predict TFBSs of target TFs for cell-types without training samples.

The main strength of our proposed MTTFsite and PDBR_TF is that they can be applied to real population. However, TF-binding site data and histone modification data for real population are unavailable, so our proposed MTTFsite and PDBR_TF cannot be applied to real population to evaluate their performance. With the advance of biotechnologies, many data for real population may be available to researchers in the future. By then, we will be able to evaluate the performance for our proposed methods using data for real population.

Chapter 6

Gene expression prediction

Many computational methods have been proposed for gene expression level prediction. DeepChrome, TEPIC and Zhang and Li's method are three state-of-the-art methods for gene expression prediction. DeepChrome [153] is a unified end-to-end architecture for gene expression prediction using a convolutional neural network. the big advantage of DeepChrome is that it can capture both pairwise interactions between neighboring bins and combinatorial relationships between different histone modification marks. However, DeepChrom does not incorporate TFBSs of any TF in its prediction. TEPIC is a segmentation-based method which predict TFBSs by combining sets of open-chromatin regions with position weight matrices [147]. In TEPIC, position weight matrices are used to identify TFBSs for TFs. However, only a small portion of TFs have known position weight matrix. So TEPIC can only incorporate the TFBSs of a small number of TFs into prediction while the TFBSs for large number of TFs cannot be used. Moreover, the predicted TFBSs by position weight matrices usually have very high false positive rate. Another method by Zhang and Li uses a combination of 10 histone modification marks, TFBSs of 15 TFs and DNase-I hypersensitivity data for prediction [204]. The TFBSs of the 15 TFs are identified by experimental methods. As only a small number of cell-types have enough number of TFs with experimentally identified TFBSs, this method is limited to only a very small number of cell-types.

In this chapter, we propose a novel method, referred to as TFChrome, to predict gene expression for two groups of cell-types to which TEPIC and Zhang and Li's method cannot be applied. In the first group of cell-types, only a very small number of TFs have experimentally identified TFBSs. For these cell-types, we apply our proposed PDBR_TF to predict TFBSs for TFs without experimentally identified TFBSs by using training samples of other TFs from the cell-types. In the second group of cell-types, all the TFs do not have experimentally identified TFBSs. For these cell-types, our proposed MTTFsite is applied to predict TFBSs of TFs by collecting training samples of the same TFs from other cell-types. Then, the predicted TFBS are used to predict gene expressions for these target cell-types by combining with histone modification features. Therefore, TFChrome is not limited to only a few cell-types with known information for training.

6.1 Datasets

Two sets of datasets are used to evaluate our proposed TFChrome: one set comes from Encyclopedia of DNA Elements (ENCODE) database and the other one comes from Roadmap Epigenomics Consortium (RMEC) database. The ENCODE Consortium is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI) and the REMC is a public resource of human epigenomic data produced from hundreds of cell-types.

The datasets from ENCODE contain four cells: GM12878, HeLa-S3, HepG2 and K562. The reason for selecting these four cell-types for evaluating our method is that they contains experimentally identified TFBSs for enough number of TFs and gene expressions measured by RNA-seq experiment. Therefore, for these four cell-types, there exists enough training samples to predict TFBSs for a large number of other TFs without experimentally identified TFBSs. As these four cell-types have both many TFs with experimentally identified TFBSs and also a larger number of TFs for which the TFBSs have not been identified by experimental methods, these four cell-types are suited to evaluate the usefulness of the TFBSs predicted by PDBR_TF in gene expression prediction.

The datasets from RMEC contain 20 different cell-types. These 20 cell-types have three common characteristics: having at least five core histone modification marks, having gene expression measured by RNA-seq experiment and having open-chromatin data measured by DNase-seq experiment. As these 20 cell-types do not have any TF with experimentally identified TFBSs, the TFBSs for TFs in these cell-types can only be predicted by our proposed MTTFsite, so these datasets is suitable to evaluate the usefulness of the TFBSs predicted by MTTFsite in gene expression prediction. For TFs in these 20 cell-types, the TFBSs from GM12878, HeLa-S3, HepG2, K562 and H1-hESC are combined as training samples, because these five cell-types have known TFBSs for many TFs. In order to make sure that the training samples are enough to train MTTFsite, 71 TFs with experimentally identified TFBSs in at least 3 cell-types are used. So we will predict the TFBSs for these 71 in all the 20 cell-types and use them as features.

In this thesis, following previous works, we formulates gene expression prediction problem as a binary classification problem. More specifically, we use labels +1 and -1 to represent gene expression as high or low, respectively. Referring to Cheng et al. (2011) and Singh et al. (2016), the genes in a particular cell-type are discretized into high (+1) or low (-1) by using the median gene expression across all the genes in the target cell-type as a threshold.

6.2 Feature representation

For our proposed TFChrome, two types of features are used to encode genes: histone modification features and TFBSs. For the 4 cell-types from ENCODE, the TFBSs are predicted by our proposed PDBR_TF and the TFBSs in the 20 cell-types from RMEC are predicted by our proposed MTTFsite.

Following Cheng et al.(2011), Dong et al. (2012) and Singh et al. (2016), we uses the 10,000 base-pair (bp) DNA region (±5000 bp) around the transcription start site (TSS) of a gene to represent the gene. As both TFBSs and histone modification features are DNA region features, we divide the 10,000 bp DNA region around TSS into bins of length 100 bp. For histone modification features, the cell-types from RMEC use seven core histone modification features including H3K27ac, H3K37me3, H3K36me3, H3K9ac, H3K9me3, H3K4me1 and H3K4me3 and the cell-types from ENCODE use 8 core histone modification features including H3K27ac, H3K37me3, H3K36me3, H3K9ac, H3K9me3, H3K4me1, H3K4me2 and H3K4me3. In each bin, the histone modification features are concatenated to represent the bin. Then the concatenated features for all bins are concatenated to represent the target gene. Thus, the input for every gene can be represented as a $n \times 100$ matrix, where the 100 columns denote the different bins and the n lows denote the histone modification features. For predicted TFBSs, for each TF, a bin is encoded as 1 if it contains a predicted TFBSs of the target TF; otherwise, the bin is encoded as 0. This makes the predicted TFBSs for each gene a $m \times 100$ matrix, where the 100 columns denote the different bins and the m lows denote the different TFs. The cell-types from ENCODE use 151 TFs while the cell-types from RMEC use 71 TFs. Finally, the $n \times 100$ matrix representing histone modification features and the $m \times 100$ matrix representing predicted TFBSs features are concatenated into a $(m + n) \times 100$ to represent a target gene.

6.3 TFChrome: A deep learning method for gene expression prediction

Due to the high performance in learning meaningful and hierarchical representations from larger datasets for biological sequence including protein, RNA and DNA, deep learning models have been widely used in bioinformatics community. Lin et al.(2016) presented a method for protein property prediction by using deep CNN and outperformed most state-of-the-art method. Leung et al.(2014) presented a method for predicting alternative splicing patterns in individual tissues and their difference across different tissues by using deep learning model. Aliphanahi et al. (2015) proposed a prediction method for sequence specificities of DNA- and RNA-binding proteins by applying CNN to learn motifs from inputting DNA or RNA sequence. Zhou and Troyanskaya (2015) proposed a prediction method for chromatin features and TFBSs for multiple TFs by combining CNN with multi-task learning method. All these works demonstrates that deep learning models are suitable for learning representations for biological sequences. So Singh et al. (2016) proposed a method, called DeepChrome, for gene expression prediction by using CNN, which demonstrates that DeepChrome can automatically capture the combinatorial relationships among different histone modifications for learning representations for genes. However, the regulation of gene expression is completed by the combination of histone modification and TF-DNA interaction. Histone modification is not enough to capture information for gene expression prediction.

In this chapter, we propose a method, called TFChrome, for gene expression prediction by applying CNN on the combination of histone modification and TFBSs of multiple TFs to learn representations for genes, which can capture three kinds of relationships: combinatorial relationships among different histone modifications, combinatorial relationships among TFBSs of different TFs and that among histone modifications and TFBSs of multiple TFs. The framework of TFChrome is shown in Figure 6.1. Figure 6.1 shows that in TFChrome, for the cell-types with labeled data for some TFs, PDBR_TF is applied to the TFs without labeled data. For the certain cell-type without labeled data for any TF, MT-TFsite is applied to the TFs without labeled data. Then the predicted TFBSs for the TFs without labeled data and the known TFBSs for other TFs are combined to be TFBS features. For each gene, TFBS features and histone modification features are fed into a CNN model for feature extraction. The features learned by the CNN model for each gene are fed into a MLP model to predict the expression level. As experimental methods for TFBS identification including ChIP-seq and Chip-chip are very time-consuming and costly, the TFBSs for most TFs in human beings are unknown. So TFChrome cannot be directly applied for cell-types in human beings. In our thesis, there are two set of cell-types are used evaluate our method TFChrome. The first set of cell-types contains four cell-types extracted from ENCODE and the four cell-types have a number of TFs with known TFBSs identified by ChIP-seq. For each one of the four cell-type, we combine the known TFBSs available for any TF as training set to train a TFBS prediction model by applying our proposed PDBR_TF. Then we apply the trained model to predict TFBSs for all the 151 considered TFs. As the the whole genome of human genes has more than 3 billion DNA base pair, it is very time-consuming to apply PDBR_TF on the whole genome. Furthermore, TFs often bind to DNA in open-chromatin regions, because open-chromatin regions are accessible for TFs to form TF-DNA interaction. Therefore, we apply PDBR_TF on only open-chromatin regions. In this thesis, we use DNase I hypersensitive sites (DHSs) measured by DNase-seq method to locate open-chromatin regions.

For the 20 cell-types from RMEC, the TFBSs for all TFs are unknown. So our proposed PDBR_TF can not be applied to predict TFBSs for TFs in these cell-types duo to lacking training samples. Even if these 20 cell-types do not have experimentally identified TFBSs for all TFs, there also exist some other cell-types with experimentally identified TFBSs for a number of TFs. So we use our proposed MTTFsite to predict TFBSs for TFs in these 20 cell-types. MTTFsite is trained by the experimentally identified TFBSs from five cell-types from ENCODE. The five cell-types are GM12878, H1-hESC, HeLa-S3, HepG2 and K562, because these five cell-types have many TFs with experimentally identified TFBSs. As there are 71 TFs with experimentally identified TFBSs in at least three cell-types of these five cell-types, we can predict the TFBSs for these 71 TFs in all the 20 cell-types from RMEC by applying MTTFsite. For each TF of the 71 TFs, We combine the labeled data avalaible in the five cell-types and train a prediction model by applying MTTFsite. The trained modle for each TF can be applied to predict its TFBSs in any cell-

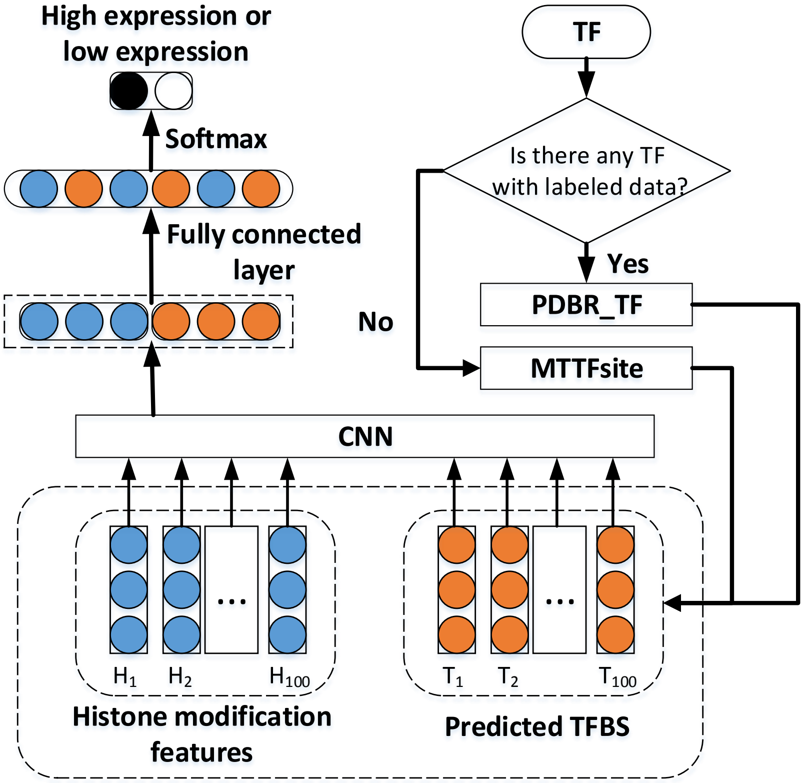

High expression or low expression
Softmax
Fully connected layer
CNN
H1
H2
H100
Histone modification features
T1
T2
T100
Predicted TFBS
TF
Is there any TF with labeled data?
Yes
No
PDBR_TF
MTTFsite

Figure 6.1: The framework of TFChrome.

type. In order to decrease the time cost of applying MTTFsite in these 20 cell-types, we also applies MTTFsite on only open-chromatin regions identified by DNase-seq method.

After obtaining predicted TFBSs for the considered TFs by our proposed methods, we concatenate histone modification features and the predicted TFBSs of the considered TFs to encode genes following section 6.2. Thus, each gene can be encoded as a $(m+n)\times 100$ matrix, where m,n represent the number of considered histone modification features and that of considered TFs. Then, the $(m+n)\times 100$ matrix for every gene is fed into TFChrome to predict its expression. TFChrome mainly contains four parts: the convolution layer, the pooling layer, the dropout layer and the fully connected neural network. In the two convolution layers, 200 convolution filters of size 10 have be used. In the pooling layer, maxpooling and pool size of 82 are used. In the dropout layer, the probability of 0.5 is chosen. In the fully connected neural network, two fully connected layer and a softmax layer are used. The number of hidden units chosen for the two fully connected layer are 100 and 50, respectively.

6.4 Results on the 4 cell-types from ENCODE

We first evaluate our method by the 4 cell-types from ENCODE. In this evaluation, three methods are compared. The first one is the method using only predicted TFBSs, the second one is the method using only histone modification features and the last one is our proposed TFChrome which uses both the predicted TFBSs and histone modification features. The hyper-parameters for these three methods are same. The only difference among these three methods is that the inputting features are different. The performance of these three methods on the 4 cell-types from ENCODE is listed in table 6.1, where TFBS, Histone and Combine denote the three methods, respectively, and the best performers are marked by bold and underscore, respectively. The p value denotes the difference between the method used only histone modification features and our proposed TFChrome.

Table 6.1: Results on the 4 cell-types from ENCODE.

Cells	TFBS	Histone	Combine	p value
GM12878	0.603	0.712	**0.733**	2.38e-2
HeLa-S3	0.569	0.607	**0.657**	1.29e-3
HepG2	0.560	0.742	**0.751**	6.82e-3
K562	0.595	0.763	**0.773**	1.57e-3

Table 6.1 shows that the method using only predicted TFBS achieves good performance on all the four cell-types, which are far better than random guessing. So we can indicates that the predicted TFBSs by our proposed PDBR_TF indeed play important roles for gene expression prediction.

Table 6.1 also shows that our prosed TFChrome performs better than the method using only histone modification features on all the four cell-types. More specifically, TFChrome performs better than the method using only histone modification features by 0.021 AUC on GM12878, 0.05 AUC on HeLa-S3, 0.009 AUC on HepG2 and 0.01 AUC on K562, which indicates that the outperformances of TFChrome on all the four cell-types are very prominent improvements. And the p values listed in table 6.1 indicate that the outperformances of TFChrome over the method using only histone modification features on all the four cell-types are signifficant. As the only difference between TFChrome and the method using only histone modification features is that TFChrome incorporates predicted TFBSs for prediction, we conclude that the TFBSs predicted by our proposed PDBR_TF and histone modifications are complementary for gene expression prediction.

Table 6.2: Results of TFChrome with and without predicted TFBSs on the 4 cells from ENCODE.

Cells	Partial TFChrome	TFChrome	p value
GM12878	0.714	**0.733**	3.45e-2
HeLa-S3	0.646	**0.657**	7.72e-3
HepG2	0.734	**0.751**	2.04e-2
K562	0.770	**0.773**	5.17e-2

Furthermore, the predicted TFBSs used by TFChrome also contains the predicted TFBSs of the TFs with experimentally identified TFBSs. However, the TFBSs of these TFs already have been identified by experimental methods and do not need to be predicted by computational methods. So, the performance listed in table 6.1 may overestimate the contribution of predicted TFBSs for gene expression prediction. To evaluate the effect of the TFs without experimentally identified TFBSs, we execute another experiment on the four cell-types. In the new experiment, two methods are compared. The first method is our method TFChrome and the other method is a variant of TFChrome in which only the predicted TFBSs of the TFs with experimentally identified TFBSs are incorporated into prediction. Their performance of these two methods on the four cell-types are listed in table 6.2, where the best performers are marked by bold. The Partial TFChrome denote TFChrome using only the predicted TFBSs of the TFs with experimentally identified TFBSs. Table 6.2 shows that TFChrome outperforms the partial TFChrome by 0.019 AUC on

GM12878, 0.011 AUC on HeLa-S3, 0.017 AUC on HepG2, 0.003 AUC on K562. The p values shows that the improvements achieved by TFChrome on three cell-types are significant. As the difference between TFChrome and the Partial TFChrome is that TFChrome incorporates predicted TFBSs of the TFs without experimentally identified TFBSs into prediction, it indicates the predicted TFBSs of the TFs without experimentally identified TFBSs indeed play important roles in prediction. However, the improvement on K562 is not significant. The reason may be that the number of TFs having experimentally TFBSs are larger while the number of TFs without identified TFBSs are small.

6.5 Results on the 20 cell-types from RMEC

In this section, we evaluate our method TFChrome by the 20 cell-types from RMEC. On these 20 cell-types, we also compare three methods: method using only predicted TFBSs, method using only histone modification features and our proposed TFChrome. Their performances on the 20 cell-types are listed in Table 6.3, where the best performers are marked by bold and underscore, respectively. The p value denotes the difference between the method using only histone modification features and TFChrome. Table 6.3 shows that for the 20 cell-types, the maximum, minimum and average AUC of the method using only the predicted TFBSs are 0.815, 0.744 and 0.769, which shows that the method using only the predicted TFBss performs far better than random guessing. It indicates that the predicted TFBSs by our proposed MTTFsite indeed play important roles in prediction. Note that the used TFBSs are predicted by MTTFsite trained by the TFBSs of target TFs from multiple other cell-types, which means that we do not use any information about histone modifications and TF-DNA interactions of target cell-types. It implies that we can predict gene expression by our proposed TFChrome for the cell-types without histone modification features and TF-DNA interactions.

Table 6.3 also shows the performance of TFChrome and the method using only his-

Table 6.3: Results on the 20 cell-types from RMEC.

Cells	TFBS	Histone	Combine	p value
Breast_vHMEC	0.779	0.859	**0.864**	1.20e-2
Fetal_Brain	0.764	0.848	**0.855**	4.68e-2
Fetal_Muscle_Leg	0.773	0.854	**0.858**	3.62e-2
Fetal_Muscle_Trunk	0.759	0.802	**0.849**	2.93e-2
Gastric	0.752	0.813	**0.819**	1.48e-2
H1_BMP4_Derived_Mesendoderm_Cultured_Cells	0.746	0.787	**0.827**	2.74e-2
H1_BMP4_Derived_Trophoblast_Cultured_Cells	0.751	0.831	**0.840**	1.66e-3
H1_Cell_Line	0.754	0.837	**0.844**	9.98e-3
H1_Derived_Mesenchymal_Stem_Cells	0.782	0.833	**0.839**	4.17e-2
H1_Derived_Neuronal_Progenitor_Cultured_Cells	0.752	0.833	**0.839**	2.85e-2
IMR90_Cell_Line	0.789	0.852	**0.860**	2.13e-2
iPS_DF_19.11_Cell_Line	0.744	0.808	**0.813**	1.25e-2
iPS_DF_6.9_Cell_Line	0.746	0.823	**0.826**	4.50e-2
Mobilized_CD34_Primary_Cells	0.797	0.872	**0.878**	2.00e-2
Pancreas	0.754	0.824	**0.832**	1.75e-2
Penis_Foreskin_Fibroblast_Primary_Cells	0.815	0.885	**0.891**	2.44e-2
Penis_Foreskin_Keratinocyte_Primary_Cells	0.794	0.872	**0.880**	3.83e-2
Penis_Foreskin_Melanocyte_Primary_Cells	0.801	0.875	**0.881**	6.58e-3
Psoas_Muscle	0.767	0.801	**0.858**	2.80e-2
Small_Intestine	0.767	0.835	**0.840**	4.25e-2

tone modification features, which indicates that TFchrome performs better than the method using only histone modifications on all the 20 cell-types. The p values show that all the improvements achieved by using predicted TFBSs are significant. For some cell-types, the performance improvement achieved by predicted TFBSs are very larger. For example, the performance improvement for Fetal_Muscle_Trunk, H1_BMP4_Derived_Trophoblast_Cultured_Cells and Psoas_Muscle are 0.033, 0.040 and 0.057, respectively. These improvements achieved by predicted TFBSs on all the 20 cell-types indicates that the TFBSs predicted by our proposed MTTFsite indeed play important roles in gene expression prediction.

6.6 Comparison with state-of-the-art methods

So far, many computational methods have been proposed for gene expression prediction. DeepChrome, TEPIC and Zhang and Li's method are three state-of-the-art methods. As the datasets used in our proposed TFChrome is different from TEPIC and Zhang and Li's method formulates gene expression prediction as an linear regression problem, the performance comparison among TFChrome, TEPIC and Zhang and Li's method cannot be conducted directly. Therefore, in this section, we only compare TFChrome and DeepChrome. DeepChrome is proposed by using CNN and 5 core histone modification features, which outperforms the most previous methods on 56 cell-types from RMEC. So, in this section, we compare our proposed TFChrome with DeepChrome on 15 cell-types from RMEC which are the common cell-types between the 20 cell-types used in our thesis and the 56 cell-types used in DeepChrome. The comparisons between TFChrome and DeepChrome on the 15 common cell-types are listed in table 6.4,where the best performers are marked by bold. Note that the AUCs of DeepChrome on the 15 common cell-types are referred from the work proposed by Singh et al.(2016). The table shows that our proposed TFChrome performs far better than DeepChrome on 14 cell-types of the 15 common cell-types. The maximum improvement, minimum improvement and average improvement is 0.120, 0.017 and 0.062, which are larger improvements for all cell-types. As the main

Table 6.4: Comparison with DeepChrome on the 20 cell-types from RMEC.

Cells	DeepChrome	TFChrome
Breast_vHMEC	0.810	**0.864**
Fetal_Brain	0.780	**0.855**
Gastric	0.730	**0.819**
H1_BMP4_Derived_Mesendoderm_Cultured_Cells	0.810	**0.827**
H1_BMP4_Derived_Trophoblast_Cultured_Cells	0.800	**0.840**
H1_Cell_Line	0.770	**0.844**
H1_Derived_Mesenchymal_Stem_Cells	0.820	**0.839**
H1_Derived_Neuronal_Progenitor_Cultured_Cells	0.770	**0.839**
Mobilized_CD34_Primary_Cells	0.800	**0.878**
Pancreas	0.740	**0.832**
Penis_Foreskin_Fibroblast_Primary_Cells	0.830	**0.891**
Penis_Foreskin_Keratinocyte_Primary_Cells	0.840	**0.880**
Penis_Foreskin_Melanocyte_Primary_Cells	0.840	**0.881**
Psoas_Muscle	**0.900**	0.858
Small_Intestine	0.720	**0.840**

difference between TFchrome and DeepChrome is that TFChrome uses both the histone modifications and the predicted TFBSs in features while DeepChrome uses only the histone modifications in features, the larger outperformance of TFChrome over DeepChrome further demonstrates the usefulness of predicted TFBSs for gene expression prediction.

6.7 Chapter Summary

For most cell-types of humans, only a limited number of TFs have experimentally identified TFBSs. For a larger number of cell-types, no TFBSs for their TFs are known. For these cell-types, either PDBR_TF or MTTFsite proposed in this thesis can be used to predict their TFBSs. In this chapter, a novel method, referred to as TFChrome, is proposed for gene expression prediction. The main idea is to leverage on histone modification features and TFBSs of the considered TFs. There is no requirement that gene expression must have available TFBSs of the considered TFs as they can be predicted by either our proposed PDBR_TF or MTTFsite to learn the representations for genes. The performance evaluation on two groups of cell-types shows that the predicted TFBSs of considered TFs can achieve far better performance than random guessing and the combined use of the predicted TFBSs and histone modification features performs better than histone modification features alone. This indicates that the use of predicted TFBSs using our proposed methods are indeed important in gene expression prediction. As our proposed MTTFsite and PDBR_TF can be used predict TFBSs of considered TFs for all cell-types regardless of availability of training samples for a particular cell-type, our proposed TFChrome can be used to predict gene expressions for all cell-types.

Chapter 7

Conclusions and Future Work

Gene expression prediction is a very important research area in bioinformatics. It can help with disease diagnosis as well as improve drug design for patients. One of the key elements in gene expression is TF-DNA interaction. This thesis studies TF-DNA interactions and its application in gene expression prediction. The focus is on predicting DNA binding residues, TF binding sites and then use them for predicting gene expression.

7.1 Contributions

This thesis covers a comprehensive range of studies on gene expression prediction from protein second structure prediction, DNA binding residue prediction, TF binding site prediction to gene expression prediction. The main conclusions and contributions are summarized as follows:

1. **Protein second structure prediction.**

   A general framework, called CNNH_PSS, is proposed to automatically learn feature representations for residues in protein sequence. The main contribution of CNNH_PSS is to use Position Specific Score Matrix by a multi-scale CNN framework with a highway as a mechanism to capture both local context and long-range dependencies. It is one of the first attempts to predict secondary structures of residues for TFs with high-quality performance.

2. **DNA binding residue prediction.** Four effective methods for DNA binding residue prediction are proposed to learn various relationships among residues to overcome the limitations of the current state-of-the-art methods. (1) EL_PSSM-RT can effectively learn pairwise relationships between residues in a short range; (2) CNNsite can learn relationships among multiple residues in a short range; (3) EL_LSTM can learn residue representations by extracting both local context and long-range dependencies; and (4) PDNAsite can learn sequence relationships between sequence neighbor residues as well as spatial relationships between spatial neighbor residues.

3. **TF binding site prediction.** Three effective methods for TF binding site prediction are proposed. The CNN_TF method uses a deep learning method to effectively include higher order dependencies as an additional feature in TF binding site prediction for a particular cell-type which has sufficient training samples. The MTTFsite method addresses data sparseness within a particular cell-type by leveraging on TF training samples available in other cell-types using multi-task learning. MTTFsite learns common features from multiple cell-types with training TFBS samples using a common CNN as well as features of individual cell-types using a group of private CNNs for individual cell-types which have training TFBS samples. The PDBR_TF further addresses the non-availability issue by using the training TFBS samples of other TFs for a target TF.

4. **Gene expression prediction.** Finally, the TFChrome method is proposed to effectively predict gene expressions making using of all the new methods we have developed so as to include more relationships as well as effective methods to address both data sparseness issue and data unavailability issue. In TFChrome, predicted TFBSs and histone modifications are combined to learn feature representations for genes. For cell-types with only a small number TFs having experimentally identified TFBS, TFChrome can use our proposed PDBR_TF to predict TFBSs for the TFs

without experimentally identified TFBSs. For cell-types which do not have TFs with experimentally identified TFBSs, TFChrome can use MTTFsite to predict TFBSs of for all considered TFs. Evaluations on two sets of cell-types and comparisons with state-of-the-art methods indicates that predicted TFBSs indeed play important roles for gene expression prediction. TFChrome can be widely used to predict gene expressions for any cell-types regardless of the availability of training data available for that cell type.

7.2 Limitations and Future Work

We have shown the roles of long-range dependencies learned by CNNH_PSS in protein secondary structure prediction. However, validation on whether the learned long-range dependencies really exist is yet to be done. Because only a limited number of proteins has known 3D structures, the DNA binding residue prediction methods using both sequence features and structure features are applicable to only a limited number of proteins. For many specific cell-types of other species(except humans and mice), many of the target TFs do not have any training sample for any cell-type and some specific cell-types do not have any training sample for any TF. Thus, both MTTFsite and PDBR_TF are currently applicable to human and mice mostly. Sequence information is only one type of features for genes. Its combination with histone modification features and TFBSs may have potential effect for gene expression prediction. But, sequence features is yet to be incorporated into TFChrome for gene expression prediction.

Recent studies have shown that X-ray diffraction crystallography and NMR can identify the 3D structures for proteins [135, 128], so these methods can be used to calculate the spatial distance between any two residues in a protein sequence. Long-range dependencies between residues are usually formed by the short spatial distance between them because neighboring residues located spatially tend to have similar structures and biology functions. Therefore, the spatial distance between two residues can validate whether the learned long-range dependency between them is really existed. Thus, one possible future direction is to validate the learned long-range dependencies learned by CNNH_PSS by 3D structures of proteins. Another direction of research is to develop high-performance computational methods to predict 3D structures for proteins based on sequence features.

With regards to TFBS predictions, another direction of work can be finding effective prediction methods for TFBS of target TFs in a specific cell-type by using training TFBS samples of other TFs in other cell-types. This can help to predict the TFBSs of a target TF for all cell-types.

The curretn work on gene expression prediction can be further improved by combining sequence features, histone modification features and TFBSs to in the representation of genes. As these three types of features contains both the features of genes and cell-types, the developed methods may be applicable to predict the expressions of genes for cell-types which are different from the training cell-types.

Appendices

Appendix A

Performance for cross-cell-type TFBS prediction.

Table A.1: The AUC of the baseline method (Base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in GM12878.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ATF2	0.949	0.949	**0.951**	JUND	0.969	0.975	**0.975**
ATF3	0.823	0.879	**0.891**	MAX	0.920	**0.940**	0.939
BCL11A	**0.943**	0.934	0.939	MAZ	0.907	**0.933**	0.932
BCL3	0.891	0.895	**0.900**	MEF2A	0.948	0.949	**0.951**
BCLAF1	0.923	**0.926**	0.925	MXI1	0.928	0.942	**0.943**
BHLHE40	0.925	0.938	**0.941**	MYC	0.912	0.919	**0.919**
BRCA1	0.939	0.969	**0.977**	NFE2	**0.890**	0.877	0.885
CEBPB	0.965	0.969	**0.970**	NFIC	0.951	0.952	**0.956**
CHD1	0.958	0.956	**0.960**	NFYA	0.912	0.948	**0.953**
CHD2	0.929	0.939	**0.941**	NFYB	0.902	0.922	**0.927**
CTCF	0.778	0.844	**0.859**	NRF1	0.904	**0.917**	0.913
E2F4	0.911	0.927	**0.930**	NRSF	0.688	0.893	**0.906**
EGR1	0.900	0.916	**0.920**	P300	0.957	0.966	**0.971**
ELF1	0.880	**0.910**	0.909	PML	0.935	0.945	**0.949**
ELK1	0.929	**0.936**	0.935	RAD21	0.791	0.858	**0.873**
ETS1	0.912	0.915	**0.921**	REST	0.947	**0.959**	0.957
EZH2	**0.922**	0.905	0.911	RFX5	0.916	0.931	**0.939**
FOS	0.913	0.956	**0.963**	RXRA	0.923	0.919	**0.930**
GABP	0.891	0.928	**0.934**	SIN3A	0.930	**0.942**	0.939
SIX5	0.912	0.940	**0.946**	SMC3	0.794	0.872	**0.884**
SP1	0.939	0.949	**0.953**	SRF	0.893	0.906	**0.912**
STAT1	**0.910**	0.903	0.908	STAT3	0.948	0.950	**0.955**

STAT5A	0.947	0.944	**0.948**	TAF1	0.941	0.960	**0.960**
TBP	0.949	0.958	**0.961**	TBLR1	0.937	0.942	**0.943**
TCF12	0.925	0.923	**0.926**	GTF2F1	0.859	0.881	**0.886**
TR4	0.651	0.801	**0.816**	USF1	0.874	0.910	**0.914**
USF2	0.915	0.938	**0.938**	YY1	0.926	**0.946**	0.943
ZBTB33	0.897	0.931	**0.931**	ZNF143	0.800	0.862	**0.881**
ZZZ3	0.726	**0.752**	0.740	ZNF274	**0.908**	0.883	0.887

Table A.2: The AUC of the baseline method (Base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in H1-hESC.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ATF2	0.891	0.885	**0.895**	JUND	0.854	0.871	**0.876**
ATF3	0.839	0.876	**0.891**	JUN	0.883	0.909	**0.912**
BACH1	0.842	0.845	**0.853**	MAFK	0.781	0.845	**0.859**
BCL11A	**0.944**	0.903	0.923	MAX	0.890	0.894	**0.905**
BRCA1	0.809	0.903	**0.923**	MXI1	0.862	0.873	**0.888**
BRG1	0.955	0.959	**0.962**	SIN3A	0.862	0.859	**0.868**
CEBPB	0.813	0.852	**0.862**	MYC	**0.681**	0.640	0.642
CHD1	0.939	0.926	**0.931**	NRF1	0.869	0.884	**0.886**
CHD2	0.850	0.886	**0.899**	NRSF	0.727	0.809	**0.819**
CTCF	0.759	0.806	**0.816**	TEAD4	0.870	0.870	**0.880**
EGR1	0.830	0.838	**0.849**	P300	0.890	0.878	**0.888**
EZH2	0.959	**0.963**	0.960	RAD21	0.762	0.798	**0.814**
FOSL1	0.759	**0.762**	0.754	RBBP5	0.879	0.879	**0.887**
GABP	0.709	0.722	**0.729**	RFX5	0.819	0.859	**0.869**
RXRA	0.836	0.831	**0.854**	ZNF143	0.783	0.814	**0.832**
SIX5	0.865	0.889	**0.897**	SP1	0.903	0.899	**0.911**
SP2	0.878	**0.916**	0.914	SRF	0.837	0.869	**0.887**
TAF1	0.866	0.877	**0.885**	TAF7	0.865	0.853	**0.868**
TBP	0.856	0.854	**0.866**	TCF12	0.879	0.881	**0.893**
USF1	0.828	0.849	**0.858**	USF2	0.845	0.876	**0.887**
YY1	0.845	0.853	**0.863**				

Table A.3: The AUC of the baseline method (Base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in HeLa-S3.

TF	Base	FS	MTTF	TF	FS	Base	MTTF
BDP1	0.704	0.802	**0.809**	BRF1	0.743	0.758	**0.767**
BRCA1	0.938	0.940	**0.941**	JUND	0.919	0.939	**0.942**
JUN	0.951	0.955	**0.958**	REST	0.946	0.938	**0.940**
CEBPB	0.888	0.908	**0.914**	MAFK	0.857	0.892	**0.910**
CHD2	0.930	0.934	**0.940**	MAX	0.926	0.939	**0.943**
CTCF	0.768	0.816	**0.834**	MAZ	0.889	0.918	**0.920**
E2F4	0.921	0.933	**0.934**	MXI1	0.935	0.949	**0.951**
MYC	0.955	0.960	**0.964**	TAF1	0.940	0.955	**0.957**
ELK1	0.935	0.940	**0.945**	NFYA	0.889	0.903	**0.910**
EZH2	0.937	0.928	**0.936**	NFYB	0.911	0.939	**0.944**
FOS	0.968	0.972	**0.974**	NRF1	0.917	0.942	**0.948**
GABP	0.913	0.944	**0.946**	NRSF	0.670	0.807	**0.818**
GTF2F1	0.941	0.941	**0.943**	P300	0.955	0.960	**0.961**
HDAC2	0.908	0.898	**0.908**	ZNF274	0.957	0.956	**0.987**
IRF3	0.941	0.952	**0.955**	ZZZ3	0.821	**0.857**	0.831
BRF2	0.681	0.700	**0.732**	RAD21	0.833	0.865	**0.890**
RFX5	0.910	0.905	**0.912**	SMC3	0.837	0.868	**0.888**
STAT1	0.916	0.917	**0.921**	STAT3	0.965	0.965	**0.968**
TCF12	0.917	0.944	**0.949**	TBP	0.959	0.963	**0.964**
TCF7L2	0.938	0.943	**0.947**	TR4	0.861	0.916	**0.926**
USF2	0.917	0.930	**0.938**	ZNF143	0.832	0.899	**0.918**

Table A.4: The AUC of the baseline method (Base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in HepG2.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ARID3A	0.927	0.941	**0.942**	MAFK	0.732	0.735	**0.748**
ATF3	0.903	0.945	**0.949**	MAX	0.901	0.922	**0.923**
BHLHE40	0.896	0.917	**0.920**	MAZ	0.893	0.920	**0.925**
BRCA1	0.858	0.896	**0.899**	MXI1	0.905	0.918	**0.918**
CEBPB	0.816	0.824	**0.837**	MYC	0.934	0.936	**0.938**
CHD2	0.900	0.925	**0.929**	NFIC	0.961	0.964	**0.965**
CTCF	0.789	0.852	**0.871**	NRF1	0.924	0.953	**0.955**
ELF1	0.916	0.941	**0.944**	NRSF	0.603	0.828	**0.831**
EZH2	0.897	0.897	**0.908**	P300	0.950	0.961	**0.963**
GABP	0.823	0.856	**0.864**	RAD21	0.816	0.875	**0.889**
HDAC2	0.942	0.947	**0.949**	REST	**0.928**	0.910	0.911

IRF3	0.895	0.917	**0.924**	RFX5	0.883	0.902	**0.907**
JUND	0.812	0.819	**0.829**	RXRA	0.944	0.946	**0.952**
JUN	0.836	0.838	**0.847**	SIN3A	0.923	0.929	**0.929**
MAFF	0.759	0.775	**0.778**	SMC3	0.841	0.888	**0.898**
ZNF274	0.941	0.946	**0.955**	SP1	0.953	0.955	**0.961**
SP2	0.781	0.824	**0.840**	SRF	0.919	0.936	**0.944**
TAF1	0.873	0.890	**0.892**	TBP	0.942	0.957	**0.957**
TCF7L2	0.939	**0.942**	0.938	TEAD4	0.948	0.959	**0.962**
TR4	0.900	0.920	**0.923**	USF2	0.872	0.896	**0.904**
YY1	0.876	0.892	**0.897**	ZBTB33	0.897	0.911	**0.916**
ZBTB7A	0.908	0.917	**0.919**				

Table A.5: The AUC of the baseline method (Base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in K562.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ARID3A	0.930	0.938	**0.945**	GTF2F1	0.890	0.892	**0.900**
ATF3	0.889	0.900	**0.906**	HDAC2	0.911	0.916	**0.924**
BACH1	0.902	0.914	**0.919**	JUND	0.894	0.907	**0.912**
BCL3	**0.909**	0.882	0.881	JUN	0.928	0.938	**0.943**
BCLAF1	0.934	0.939	**0.941**	MAFF	0.805	0.848	**0.862**
BDP1	0.556	**0.684**	0.668	BRF1	0.961	0.967	**0.978**
BHLHE40	0.889	0.905	**0.910**	MAFK	0.839	0.872	**0.884**
BRG1	0.973	0.972	**0.977**	MAX	0.901	0.915	**0.919**
CEBPB	0.853	0.872	**0.880**	MAZ	0.881	0.887	**0.894**
CHD1	0.914	0.913	**0.916**	MEF2A	0.921	0.928	**0.933**
CHD2	0.881	0.908	**0.914**	MXI1	0.927	0.940	**0.943**
CTCF	0.776	0.826	**0.839**	MYC	0.942	0.944	**0.947**
E2F4	0.884	0.898	**0.903**	NFE2	0.935	0.929	**0.936**
E2F6	0.892	0.895	**0.897**	NFYA	0.908	0.930	**0.934**
EGR1	0.854	0.858	**0.867**	NFYB	0.926	0.946	**0.953**
ELF1	0.868	0.896	**0.899**	NRF1	0.903	0.923	**0.931**
ELK1	0.899	0.914	**0.921**	NRSF	0.862	0.873	**0.882**
ETS1	0.884	0.882	**0.886**	P300	0.932	0.941	**0.946**
EZH2	0.910	0.907	**0.920**	PML	0.926	0.930	**0.934**
FOSL1	0.914	0.911	**0.917**	RAD21	0.821	0.896	**0.921**
FOS	0.929	0.949	**0.953**	RBBP5	0.886	0.909	**0.910**
GABP	0.901	**0.915**	0.913	REST	0.894	0.898	**0.905**
ZNF274	0.824	0.825	**0.843**	RFX5	0.859	0.881	**0.886**
SIN3A	0.902	0.910	**0.913**	SIX5	0.886	0.920	**0.924**
SMC3	0.813	0.870	**0.886**	SP1	0.925	0.949	**0.954**

BRF2	0.755	0.786	**0.800**	SP2	0.839	0.868	**0.864**
SRF	0.899	0.930	**0.932**	STAT1	0.929	0.949	**0.951**
STAT5A	0.949	0.949	**0.954**	TAF1	0.920	0.939	**0.940**
TAF7	0.905	**0.907**	0.904	TBP	0.907	0.926	**0.928**
TBLR1	0.882	0.895	**0.895**	TEAD4	0.912	0.923	**0.928**
TR4	0.697	0.785	**0.822**	YY1	0.862	0.920	**0.925**
ZBTB33	0.844	0.861	**0.861**	ZBTB7A	0.864	0.870	**0.877**
ZNF143	0.820	0.852	**0.866**				

Table A.6: The AUC of the baseline method (base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in GM12878 without using training samples.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ATF2	**0.949**	0.911	0.925	ATF3	0.823	0.922	**0.931**
BCL11A	**0.943**	0.869	0.896	BCL3	**0.891**	0.822	0.829
BCLAF1	**0.923**	0.907	0.918	BHLHE40	0.925	0.934	**0.939**
BRCA1	0.939	0.949	**0.965**	CEBPB	0.965	0.964	**0.969**
CHD1	**0.958**	0.946	0.944	CHD2	0.929	0.938	**0.938**
CTCF	0.778	0.847	**0.881**	E2F4	0.911	0.923	**0.934**
EGR1	0.900	0.911	**0.918**	ELF1	0.880	0.909	**0.911**
ELK1	0.929	0.938	**0.949**	ETS1	0.912	0.912	**0.924**
EZH2	**0.922**	0.875	0.911	FOS	0.913	0.966	**0.971**
GABP	0.891	**0.938**	0.933	GTF2F1	0.859	0.903	**0.905**
JUND	0.969	0.972	**0.973**	MAX	0.920	0.939	**0.942**
MAZ	0.907	0.929	**0.931**	MEF2A	**0.948**	0.928	0.943
MXI1	0.928	0.939	**0.942**	MYC	0.912	0.909	**0.914**
NFE2	0.890	0.908	**0.923**	NFIC	**0.951**	0.936	0.942
NFYA	0.912	0.956	**0.966**	NFYB	0.902	0.923	**0.933**
NRF1	0.904	0.936	**0.943**	NRSF	0.688	0.883	**0.911**
P300	0.957	0.973	**0.976**	PML	0.935	0.947	**0.951**
RAD21	0.791	0.865	**0.901**	REST	0.947	0.936	**0.948**
RFX5	0.916	0.923	**0.941**	RXRA	0.923	0.922	**0.942**
SIN3A	0.930	0.931	**0.933**	SIX5	0.912	0.949	**0.957**
SMC3	0.794	0.879	**0.896**	SP1	0.939	0.952	**0.956**
SRF	0.893	0.911	**0.924**	STAT1	0.910	0.929	**0.934**
STAT3	0.948	0.951	**0.958**	STAT5A	0.947	0.942	**0.951**
TAF1	0.941	0.951	**0.952**	TBP	0.949	0.959	**0.965**
TBLR1	**0.937**	0.929	0.937	TCF12	**0.925**	0.907	0.911
TR4	0.651	0.908	**0.917**	USF1	0.874	0.921	**0.927**
USF2	0.915	0.942	**0.948**	YY1	0.926	0.933	**0.947**
ZBTB33	0.897	0.928	**0.941**	ZNF143	0.800	0.867	**0.897**

Table A.7: The AUC of the baseline method (base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in H1-hESC without using training samples.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ATF2	**0.891**	0.853	0.853	ATF3	0.839	0.899	**0.912**
BACH1	**0.842**	0.769	0.768	BCL11A	**0.944**	0.889	0.902
BRCA1	0.809	0.911	**0.922**	BRG1	0.955	0.973	**0.976**
CEBPB	0.813	0.871	**0.884**	CHD1	**0.939**	0.937	0.931
CHD2	0.850	0.909	**0.916**	CTCF	0.759	0.809	**0.827**
E2F6	0.912	**0.921**	0.918	EGR1	0.830	0.834	**0.845**
EZH2	**0.959**	0.943	0.952	FOSL1	0.759	0.838	**0.868**
GABP	0.709	0.822	**0.837**	JUND	0.854	0.899	**0.903**
JUN	0.883	0.949	**0.954**	MAFK	0.781	0.839	**0.862**
MAX	0.890	0.889	**0.901**	MXI1	0.862	0.896	**0.906**
MYC	0.681	0.875	**0.892**	NRF1	0.869	0.924	**0.926**
NRSF	0.727	0.809	**0.831**	P300	0.890	0.898	**0.912**
RAD21	0.762	0.812	**0.832**	RBBP5	**0.879**	0.837	0.851
RFX5	0.819	0.909	**0.929**	RXRA	0.836	0.907	**0.924**
SIN3A	**0.862**	0.839	0.832	SIX5	0.865	0.909	**0.933**
SP1	**0.903**	0.891	0.896	SP2	0.878	0.894	**0.906**
SRF	0.837	0.899	**0.907**	TAF1	0.866	0.882	**0.885**
TAF7	**0.865**	0.815	0.766	TBP	0.856	0.863	**0.866**
TCF12	**0.879**	0.838	0.849	TEAD4	**0.870**	0.841	0.864
USF1	0.828	0.851	**0.866**	USF2	0.845	0.909	**0.913**
YY1	0.845	0.846	**0.862**	ZNF143	0.783	0.807	**0.833**

Table A.8: The AUC of the baseline method (base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in HeLa-S3 without using training samples.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
BRCA1	**0.938**	0.913	0.931	CEBPB	0.888	0.898	**0.905**
CHD2	0.930	0.931	**0.941**	CTCF	0.768	0.815	**0.842**
E2F4	0.921	0.936	**0.945**	ELK1	0.935	0.941	**0.946**
EZH2	0.937	0.928	**0.945**	FOS	0.968	0.967	**0.971**
GABP	0.913	0.943	**0.951**	GTF2F1	**0.941**	0.902	0.919
HDAC2	**0.908**	0.874	0.885	IRF3	0.941	0.938	**0.948**
JUND	0.919	0.935	**0.942**	JUN	0.951	0.951	**0.955**
MAFK	0.857	0.884	**0.896**	MAX	0.926	**0.937**	0.932
MAZ	0.889	0.917	**0.919**	MXI1	0.935	0.946	**0.951**
MYC	**0.955**	0.923	0.936	NFYA	0.889	0.919	**0.923**
NFYB	0.911	0.949	**0.954**	NRF1	0.917	0.962	**0.963**

NRSF	0.670	0.818	**0.842**	P300	0.955	0.954	**0.961**
RAD21	0.833	0.854	**0.884**	REST	**0.946**	0.925	0.932
RFX5	**0.910**	0.889	0.891	SMC3	0.837	0.859	**0.885**
STAT1	**0.916**	0.889	0.898	STAT3	**0.965**	0.954	0.964
TAF1	0.940	0.949	**0.955**	TBP	0.959	0.961	**0.966**
TCF12	0.917	0.955	**0.961**	TCF7L2	0.938	0.942	**0.953**
TR4	0.861	0.917	**0.941**	USF2	0.917	0.935	**0.938**
ZNF143	0.832	0.903	**0.913**				

Table A.9: The AUC of the baseline method (base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in HepG2 without using training samples.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ARID3A	0.927	0.924	**0.936**	ATF3	0.903	0.929	**0.939**
BHLHE40	0.896	0.913	**0.925**	BRCA1	0.858	0.933	**0.941**
CEBPB	0.816	0.828	**0.841**	CHD2	0.901	0.933	**0.938**
CTCF	0.789	0.853	**0.887**	ELF1	0.916	0.932	**0.934**
EZH2	0.897	0.917	**0.934**	GABP	0.823	0.857	**0.883**
HDAC2	**0.942**	0.927	0.936	IRF3	0.895	0.962	**0.962**
JUND	0.812	0.801	**0.822**	JUN	0.836	0.832	**0.848**
MAFF	0.759	0.768	**0.778**	MAFK	0.732	0.723	**0.741**
MAX	0.901	0.919	**0.929**	MAZ	0.893	0.919	**0.922**
MXI1	0.905	0.916	**0.924**	MYC	**0.934**	0.913	0.921
NFIC	**0.961**	0.952	0.957	NRF1	0.924	0.963	**0.964**
NRSF	0.603	0.834	**0.856**	P300	0.950	0.959	**0.964**
RAD21	0.816	0.877	**0.909**	REST	**0.928**	0.916	0.921
RFX5	0.883	0.908	**0.924**	RXRA	**0.944**	0.894	0.923
SIN3A	0.923	0.919	**0.925**	SMC3	0.841	0.893	**0.907**
SP1	**0.953**	0.943	0.948	SP2	0.781	**0.859**	0.855
SRF	0.919	0.909	**0.943**	TAF1	0.873	0.894	**0.903**
TBP	0.942	0.958	**0.961**	TCF7L2	**0.939**	0.938	0.937
TEAD4	0.948	0.956	**0.961**	TR4	0.900	0.918	**0.924**
USF1	0.865	0.898	**0.907**	USF2	0.872	0.915	**0.924**
YY1	0.876	0.894	**0.903**	ZBTB33	0.897	0.906	**0.928**
ZBTB7A	**0.908**	0.907	0.908				

Table A.10: The AUC of the baseline method (base), the fully-shared (FS) method and MTTFsite (MTTF) on TFs in K562 without using training samples.

TF	Base	FS	MTTF	TF	Base	FS	MTTF
ARID3A	0.930	0.938	**0.941**	ATF3	0.889	0.881	**0.895**
BACH1	0.902	0.909	**0.914**	BCL3	0.909	**0.911**	0.904
BCLAF1	0.934	0.921	**0.936**	BHLHE40	0.889	0.905	**0.912**
BRG1	**0.973**	0.965	0.961	CEBPB	0.853	0.872	**0.887**
CHD1	0.914	0.916	**0.927**	CHD2	0.881	0.915	**0.921**
CTCF	0.776	0.831	**0.855**	E2F4	0.884	0.889	**0.909**
E2F6	0.892	0.886	**0.893**	EGR1	**0.854**	0.838	0.854
ELF1	0.868	0.894	**0.901**	ELK1	0.899	0.929	**0.936**
ETS1	**0.884**	0.865	0.871	EZH2	0.910	0.931	**0.941**
FOSL1	**0.914**	0.804	0.802	FOS	0.929	0.951	**0.961**
GABP	0.901	0.905	**0.916**	GTF2F1	0.890	0.909	**0.917**
HDAC2	0.911	0.909	**0.921**	JUND	0.894	0.897	**0.911**
JUN	0.928	0.943	**0.951**	MAFF	0.805	0.831	**0.836**
MAFK	0.839	0.865	**0.892**	MAX	0.901	0.909	**0.921**
MAZ	0.881	0.879	**0.891**	MEF2A	0.921	0.926	**0.936**
MXI1	0.927	0.939	**0.945**	MYC	**0.942**	0.921	0.926
NFE2	**0.935**	0.877	0.895	NFYA	0.908	0.939	**0.942**
NFYB	0.926	0.949	**0.954**	NRF1	0.903	0.946	**0.951**
NRSF	0.862	0.853	**0.866**	P300	0.932	0.937	**0.944**
PML	0.926	0.924	**0.928**	RAD21	0.821	0.901	**0.941**
RBBP5	0.886	0.904	**0.915**	REST	**0.894**	0.871	0.886
RFX5	0.859	0.906	**0.922**	SIN3A	0.902	0.909	**0.915**
SIX5	0.886	0.931	**0.945**	SMC3	0.813	0.873	**0.899**
SP1	0.925	0.955	**0.962**	SP2	0.839	0.875	**0.893**
SRF	0.899	0.932	**0.941**	STAT1	0.929	0.934	**0.949**
STAT5A	**0.949**	0.936	0.941	TAF1	0.920	0.938	**0.943**
TAF7	0.905	0.901	**0.908**	TBP	0.907	**0.929**	0.909
TBLR1	0.882	0.907	**0.913**	TEAD4	0.912	0.905	**0.916**
TR4	0.697	0.919	**0.939**	USF1	0.874	0.906	**0.917**
USF2	0.920	0.929	**0.949**	YY1	0.862	0.935	**0.935**
ZBTB33	0.844	0.865	**0.894**	ZBTB7A	0.864	0.859	**0.872**
ZNF143	0.820	0.853	**0.876**				

Appendix B

The performance for cross-TF TFBS prediction.

Table B.1: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 69 TFs in GM12878.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.964	0.965	**0.973**	EBF1	0.935	0.932	**0.943**
JUND	0.993	0.994	**0.998**	ETS1	0.945	0.942	**0.953**
PAX5	0.931	0.929	**0.936**	TCF3	0.949	0.946	**0.961**
ELK1	0.937	0.962	**0.971**	RUNX3	0.959	0.963	**0.975**
CTCF	0.942	0.934	**0.944**	ZNF143	**0.967**	0.949	0.964
RFX3	0.953	0.951	**0.960**	MAZ	0.952	0.947	**0.960**
NRF1	0.929	0.919	**0.934**	MXI1	0.959	0.957	**0.966**
TAF1	0.960	0.964	**0.966**	ELF1	0.943	0.940	**0.953**
EZH2	0.787	0.770	**0.787**	STAT3	0.980	0.978	**0.982**
YY1	0.957	0.952	**0.959**	CEBPB	0.985	0.986	**0.990**
SIX5	0.944	0.949	**0.965**	USF2	0.959	0.958	**0.970**
RFX5	0.950	0.947	**0.961**	ATF2	0.965	0.980	**0.987**
ATF3	0.929	0.920	**0.938**	POU2F	0.959	0.962	**0.972**
BCL3	0.917	0.917	**0.928**	NFE2	0.938	0.937	**0.954**
MEF2C	0.983	0.981	**0.989**	NRSF	**0.767**	0.746	0.753
MTA3	0.978	0.979	**0.981**	SMC3	0.969	0.968	**0.979**
SIN3A	0.959	0.957	**0.962**	ZEB1	0.932	0.928	**0.936**
IKZF1	0.981	0.981	**0.982**	ZZZ3	0.848	0.833	**0.860**
NFATC1	0.907	0.959	**0.970**	RXRA	0.957	0.953	**0.966**
ZBTB33	0.943	0.940	**0.950**	TR4	0.922	0.918	**0.927**
PBX3	0.939	0.936	**0.948**	BATF	0.965	0.970	**0.979**
BHLHE40	0.957	0.959	**0.972**	BCL11A	0.976	0.977	**0.985**

CHD1	0.973	0.973	**0.974**	TBP	0.974	0.970	**0.979**
BCLAF1	0.971	0.970	**0.974**	E2F4	0.956	0.953	**0.958**
IRF4	0.982	0.983	**0.988**	MAX	0.964	0.962	**0.969**
SRF	0.936	0.929	**0.943**	MYC	0.952	0.955	**0.958**
SP1	0.963	0.964	**0.976**	TCF12F	0.961	0.959	**0.967**
RAD21	0.968	0.968	**0.973**	NFIC	0.981	0.981	**0.988**
PML	0.967	0.964	**0.970**	EGR1	0.939	0.937	**0.950**
USF1	0.926	0.930	**0.953**	FOS	0.961	0.966	**0.974**
P300	0.976	0.975	**0.978**	BRCA1	0.971	0.966	0.966
FOXM1	0.981	0.980	**0.986**	GABP	0.948	0.945	**0.962**
STAT1	0.940	**0.945**	0.939	NFYB	0.875	0.860	**0.904**
STAT5A	0.973	0.973	**0.980**	MEF2A	0.977	0.977	**0.984**
NFYA	0.956	0.946	**0.965**				

Table B.2: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 46 TFs in H1-hESC.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.935	0.924	**0.951**	MAFK	0.867	0.855	**0.883**
GTF2F	0.936	0.922	**0.937**	SP4	0.936	0.931	**0.950**
JUND	0.937	0.930	**0.953**	ZNF143	0.922	0.902	**0.925**
SUZ12	0.832	0.831	**0.845**	NANOG	0.945	0.932	**0.958**
CTCF	0.943	0.938	**0.950**	MXI1	0.937	0.934	**0.950**
TEAD4	0.913	0.915	**0.938**	CEBPB	0.791	0.781	**0.810**
NRF1	0.887	0.879	**0.905**	USF2	0.951	**0.956**	0.944
TAF1	0.931	0.923	**0.932**	ATF2	0.950	0.944	**0.958**
EZH2	0.888	0.894	**0.905**	POU2F	0.948	0.949	**0.970**
YY1	0.918	0.914	**0.927**	FOSL1	0.878	0.855	**0.902**
SIX5	0.925	0.916	**0.952**	RBBP5	0.922	0.922	**0.924**
RFX5	0.865	0.865	**0.876**	NRSF	0.744	0.716	**0.843**
ATF3	0.949	0.938	**0.966**	RXRA	0.928	0.925	**0.949**
SIN3A	0.939	0.934	**0.945**	BCL11A	0.966	0.966	**0.966**
JUN	0.950	0.952	**0.972**	TBP	0.906	0.904	**0.909**
CHD1	0.962	0.962	**0.965**	BACH1	0.919	0.913	**0.936**
SP2	0.927	0.917	**0.935**	MAX	0.930	0.925	**0.947**
SRF	0.858	0.836	**0.856**	MYC	0.932	0.926	**0.949**
HDAC2	0.946	0.940	**0.959**	SP1	0.934	0.921	**0.936**
TCF12	0.938	0.931	**0.942**	RAD21	0.954	0.951	**0.971**
EGR1	0.885	0.877	**0.908**	USF1	0.897	0.903	**0.920**
TAF7	0.924	0.922	**0.929**	P300	0.942	0.936	**0.955**
BRCA1	0.936	0.925	**0.941**	GABP	0.922	0.912	**0.933**

Table B.3: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 46 TFs in HeLa-S3.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.954	0.951	**0.967**	E2F1	0.902	0.902	**0.915**
GTF2F	0.973	0.974	**0.976**	JUN	0.982	0.981	**0.989**
JUND	0.976	0.971	**0.985**	TCF7	0.960	0.958	**0.966**
ELK1	0.943	0.938	**0.945**	MAFK	0.854	0.836	**0.854**
CTCF	0.913	0.903	**0.930**	ZNF143	0.925	0.917	**0.939**
AP2A	0.949	0.947	**0.963**	MAZ	0.929	0.926	**0.940**
NRF1	0.911	0.908	**0.921**	MXI1	0.959	0.955	**0.967**
ELK4	0.957	0.954	**0.968**	STAT3	0.985	0.985	**0.990**
AP2G	0.945	0.941	**0.955**	CEBPB	0.922	0.935	**0.959**
TAF1	0.962	0.959	**0.970**	USF2	0.945	0.938	**0.965**
NFYA	0.927	0.929	**0.947**	BRF1	0.860	0.868	**0.900**
BAF170	0.964	0.970	**0.976**	NRSF	**0.717**	0.700	0.711
E2F6	0.926	0.922	**0.936**	SMC3	0.961	0.965	**0.975**
BAF155	0.969	0.968	**0.978**	IRF3	0.954	0.954	**0.968**
RFX5	0.962	0.956	**0.970**	ZZZ3	**0.919**	0.884	0.886
BRF2	0.712	0.686	**0.716**	BRG1	0.966	0.963	**0.974**
TR4	0.941	0.932	**0.944**	TBP	0.971	0.974	**0.981**
E2F4	0.936	0.930	**0.940**	INI1	0.931	0.927	**0.941**
MAX	0.952	0.950	**0.963**	MYC	0.952	0.955	**0.966**
RAD21	0.959	0.961	**0.973**	BDP1	**0.949**	0.938	0.946
FOS	0.986	0.987	**0.991**	P300	0.984	0.984	**0.988**
NFYB	0.901	0.900	**0.918**	BRCA1	0.960	0.961	**0.972**
GABP	0.932	0.930	**0.940**	STAT1	0.914	0.906	**0.920**

Table B.4: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 52 TFs in HepG2.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.936	0.932	**0.944**	ZBTB33	0.955	0.923	**0.966**
FOXA2	0.967	0.973	**0.982**	BHLHE40	0.952	0.950	**0.964**
JUND	0.963	0.964	**0.971**	MAFK	0.960	**0.966**	0.929
CTCF	0.941	0.947	**0.953**	HNF4A	0.975	0.979	**0.986**
HNF4G	0.968	0.973	**0.981**	PGC1A	0.977	0.972	**0.978**
TEAD4	0.976	0.976	**0.984**	MAZ	0.942	0.932	**0.952**
NRF1	0.927	0.911	**0.940**	MXI1	0.946	0.918	**0.960**
TAF1	0.923	0.921	**0.925**	ELF1	0.947	0.937	**0.957**
EZH2	**0.690**	0.660	0.679	ZBTB7A	0.930	0.929	**0.939**

YY1	0.945	0.937	**0.949**	CEBPB	0.951	0.954	**0.961**
FOXA1	0.966	0.927	**0.981**	USF2	0.950	0.952	**0.969**
RFX5	0.924	0.911	0.921	HSF1	0.949	0.946	**0.957**
ATF3	0.962	0.950	**0.971**	NRSF	0.780	0.761	**0.836**
GABP	0.911	0.909	**0.926**	SMC3	0.975	0.972	**0.984**
SIN3A	0.953	0.953	**0.960**	IRF3	0.952	0.952	0.952
JUN	0.978	0.981	**0.981**	MAFF	0.968	0.954	**0.972**
TCF7	0.973	0.964	**0.976**	MBD4	0.976	0.968	**0.979**
FOSL2	0.962	0.957	**0.970**	RXRA	0.969	0.974	**0.979**
TR4	0.954	0.951	**0.963**	TBP	0.964	0.961	**0.971**
SP2	0.773	0.768	**0.941**	MAX	0.943	0.938	**0.963**
SRF	0.936	0.928	**0.941**	MYBL2	0.975	0.977	**0.983**
MYC	0.967	0.968	**0.978**	HDAC2	0.974	0.974	**0.983**
SP1	0.973	0.971	**0.979**	TCF12	0.977	0.975	**0.983**
RAD21	0.961	0.952	**0.966**	NFIC	0.980	0.980	**0.985**
ARID3A	0.980	0.983	**0.987**	USF1	0.920	0.929	**0.948**
P300	0.977	0.979	**0.984**	BRCA1	0.965	0.953	**0.968**

Table B.5: The AUC of the baseline method (base), the sequence method (seq) and PDBR_TF (PDBR) on 88 TFs in K562.

TF	Base	Seq	PDBR	TF	Base	Seq	PDBR
CHD2	0.930	0.933	**0.946**	BRF2	**0.826**	0.803	0.804
JUND	0.962	0.964	**0.972**	SIN3A	0.925	0.926	**0.944**
ELK1	0.957	0.958	**0.968**	CBX3	0.934	**0.935**	0.925
PLU1	0.945	0.943	**0.945**	STAT2	0.973	0.968	0.975
CTCF	0.935	0.936	**0.949**	JUN	0.973	0.973	**0.985**
JUNB	0.979	0.978	**0.980**	ZBTB33	0.906	0.908	**0.920**
TEAD4	0.958	0.964	**0.968**	BHLH40	0.940	0.942	**0.952**
UBTF	**0.924**	0.921	0.923	CHD1	0.949	0.947	**0.949**
NRF1	0.942	**0.951**	0.930	CCNT2	0.922	0.921	**0.930**
TAF1	0.944	0.945	**0.950**	GTF2F1	0.958	0.957	**0.962**
EZH2	0.762	0.677	**0.735**	MAFK	0.967	0.977	**0.980**
GATA1	0.978	0.982	**0.987**	ETS1	0.939	0.941	**0.947**
KAP1	0.817	0.786	**0.891**	HDAC1	0.942	0.943	**0.945**
NR2F2	0.957	0.956	**0.965**	SAP30	0.946	**0.951**	0.950
TAL1	0.951	0.955	**0.964**	GATA2	0.968	0.971	**0.975**
YY1	0.950	0.944	**0.951**	ZNF143	0.942	0.940	**0.951**
E2F6	0.930	0.932	**0.936**	MAZ	0.943	0.946	**0.954**
SIX5	0.937	0.941	**0.956**	MXI1	0.946	0.947	**0.954**
RFX5	0.931	0.930	**0.934**	ELF1	0.922	0.915	**0.926**

ATF3	0.933	0.925	**0.945**	ZBTB7A	0.907	0.904	**0.910**
BCL3	0.932	0.921	**0.924**	CEBPB	0.824	0.825	**0.845**
ZNF274	0.754	0.749	**0.780**	USF2	0.962	0.965	**0.974**
ZNF263	0.845	0.827	**0.845**	HDAC2	0.914	0.920	**0.923**
NELFE	0.925	0.920	**0.925**	RAD21	0.976	0.974	**0.986**
ARID3A	0.982	0.981	**0.984**	PML	0.962	0.963	**0.968**
BDP1	**0.837**	0.827	0.933	TAF7	0.932	0.924	**0.931**
P300	0.886	0.895	**0.926**	HMGN3	0.911	0.904	**0.918**
PHF8	0.941	0.943	**0.944**	BACH1	0.978	0.975	**0.983**
NFYB	0.904	0.917	**0.933**	STAT5A	0.978	0.979	**0.987**
NFYA	0.940	0.941	**0.961**	FOSL1	0.962	0.967	**0.967**
SIRT6	0.985	0.986	**0.986**	HDAC8	**0.975**	0.973	0.973
RBBP5	0.939	0.940	**0.940**	BRF1	0.974	0.971	**0.976**
NFE2	0.988	0.987	**0.992**	NRSF	0.872	0.867	**0.915**
SMC3	0.965	0.964	**0.978**	GTF2B	0.959	0.955	**0.962**
MAFF	0.948	0.953	**0.973**	TR4	0.906	0.897	**0.918**
BRG1	0.981	0.977	**0.981**	TBP	0.940	0.939	**0.944**
THAP1	0.931	0.927	**0.939**	BCLAF1	0.943	0.940	**0.951**
E2F4	0.930	0.929	**0.940**	INI1	0.973	0.970	**0.970**
SP2	0.938	0.935	**0.952**	MAX	0.949	0.952	**0.960**
SRF	0.930	0.934	**0.941**	SETDB1	0.822	0.794	**0.827**
MYC	0.963	0.962	**0.968**	SP1	0.948	0.950	**0.959**
HDAC6	0.877	0.864	**0.873**	USF1	0.932	0.943	**0.952**
FOS	0.964	0.968	**0.976**	GABP	0.933	0.936	**0.948**
STAT1	0.985	0.983	**0.986**	MEF2A	0.966	0.962	**0.963**

Bibliography

[1] Rafał Adamczak, Aleksey Porollo, and Jarosław Meller. Accurate prediction of solvent accessibility using neural networks–based regression. *Proteins: Structure, Function, and Bioinformatics*, 56(4):753–767, 2004.

[2] Rafał Adamczak, Aleksey Porollo, and Jarosław Meller. Combining prediction of secondary structure and solvent accessibility in proteins. *Proteins: Structure, Function, and Bioinformatics*, 59(3):467–475, 2005.

[3] Shandar Ahmad, M Michael Gromiha, and Akinori Sarai. Analysis and prediction of DNA-binding proteins and their binding residues based on composition, sequence and structural information. *Bioinformatics*, 20(4):477–486, 2004.

[4] Shandar Ahmad and Akinori Sarai. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinformatics*, 6(1):1, 2005.

[5] Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature Biotechnology*, 2015.

[6] Fred W. Allendorf, Paul A. Hohenlohe, and Gordon Luikart. Genomics and the future of conservation genetics. *Nature Reviews Genetics*, 11(10):697–709, 2010.

[7] V. G. Allfrey, R. Faulkner, and A. E. Mirsky. Acetylation and methylation of histones and their possible role in the regulation of RNA synthesis. *Proceedings of the National Academy of Sciences*, 51(5):786, 1964.

[8] Janez Ar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(1):1–30, 2006.

[9] Timothy L. Bailey and William Stafford Noble. Searching for statistically significant regulatory modules. *Bioinformatics*, 19 Suppl 2(suppl_2):ii16, 2003.

[10] Monya Baker. Synthetic genomes: The next step for the synthetic genome. *Nature*, 473(7347):405–8, 2011.

[11] Pierre Baldi, Søren Brunak, Paolo Frasconi, Giovanni Soda, and Gianluca Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

[12] Tanya Barrett, Stephen E Wilhite, Pierre Ledoux, Carlos Evangelista, Irene F Kim, Maxim Tomashevsky, Kimberly A Marshall, Katherine H Phillippy, Patti M Sherman, Michelle Holko, et al. NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Research*, 41(D1):D991–D995, 2012.

[13] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E. Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. High-resolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, 2016.

[14] Michael A. Beer and Saeed Tavazoie. Predicting gene expression from sequence. *Cell*, 117(2):185–98, 2004.

[15] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *Proceedings of the 31st International Conference on Machine learning*, pages 226–234, 2014.

[16] 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.

[17] Helen M Berman, Tammy Battistuz, TN Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

[18] Bradley E. Bernstein, Alexander Meissner, and Eric S. Lander. The mammalian epigenome. *Cell*, 128(4):669–681, 2007.

[19] Frances C Bernstein, Thomas F Koetzle, Graheme JB Williams, Edgar F Meyer, Michael D Brice, John R Rodgers, Olga Kennard, Takehiko Shimanouchi, and Mitsuo Tasumi. The protein data bank. *The FEBS Journal*, 80(2):319–324, 1977.

[20] Nitin Bhardwaj, Robert E Langlois, Guijun Zhao, and Hui Lu. Structure based prediction of binding residues on DNA-binding proteins. In *Proceeding of the 27th Annual International Conference On the Engineering in Medicine and Biology Society.*, pages 2611–2614. IEEE, 2005.

[21] Nitin Bhardwaj and Hui Lu. Residue-level prediction of DNA-binding sites and its application on DNA-binding protein predictions. *FEBS Letters*, 581(5):1058–1066, 2007.

[22] Christopher M. Bishop. Neural networks for pattern recognition. *Agricultural Engineering International the Cigr Journal of Scientific Research & Development Manuscript Pm*, 12(5):1235 – 1242, 2001.

[23] Ashis Kumer Biswas, Nasimul Noman, and Abdur Rahman Sikder. Machine learning approach to predict protein phosphorylation sites by incorporating evolutionary information. *BMC bioinformatics*, 11(1):273, 2010.

[24] Alexey Bochkarev, Elena Bochkareva, Lori Frappier, and Aled M. Edwards. The 2.2 A structure of a permanganate-sensitive DNA site bound by the Epstein-Barr virus origin binding protein, EBNA1. *Journal of Molecular Biology*, 284(5):1273–1278, 1998.

[25] Danail Bonchev. The overall wiener index A new tool for characterization of molecular topology. *Journal of Chemical Information and Computer Sciences*, 41(3):582–592, 2001.

[26] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.

[27] Sa Boyadijiev and Ew Jabs. Online mendelian inheritance in man (omim) as a knowledgebase for human developmental disorders. *Clinical Genetics*, 57(4):253–266, 2000.

[28] Laurie A Boyer, Tong Ihn Lee, Megan F Cole, Sarah E Johnstone, Stuart S Levine, Jacob P Zucker, Matthew G Guenther, Roshan M Kumar, Heather L Murray, Richard G Jenner, et al. Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, 122(6):947–956, 2005.

[29] Andrew P Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[30] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[31] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[32] Michael Brudno, Chuong B. Do, Gregory M. Cooper, Michael F. Kim, Eugene Davydov, Eric D. Green, Arend Sidow, and Serafim Batzoglou. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Research*, 13(4):721–731, 2003.

[33] Jan Christian Bryne, Eivind Valen, Man-Hung Eric Tang, Troels Marstrand, Ole Winther, Isabelle da Piedade, Anders Krogh, Boris Lenhard, and Albin Sandelin. JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucleic Acids Research*, 36(suppl_1):D102–D106, 2007.

[34] David M. Budden, Daniel G. Hurley, Joseph Cursons, John F. Markham, Melissa J. Davis, and Edmund J. Crampin. Predicting expression: the complementary power

of histone modification and transcription factor binding data. *Epigenetics and Chromatin*, 7(1):1–12, 2014.

[35] Alex N Bullock and Alan R Fersht. Rescuing the function of mutant p53. *Nature Reviews Cancer*, 1(1):68–76, 2001.

[36] Martha L Bulyk. Computational prediction of transcription-factor binding site locations. *Genome Biology*, 5(1):201, 2003.

[37] Ting Wen Chen, Hsin Pai Li, Chi Ching Lee, Ruei Chi Gan, Po Jung Huang, Timothy H Wu, Cheng Yang Lee, Yi Feng Chang, and Petrus Tang. ChIPseek, a web-based analysis tool for ChIP data. *BMC Genomics*, 15(1):539, 2014.

[38] Xi Chen, Han Xu, Ping Yuan, Fang Fang, Mikael Huss, Vinsensius B Vega, Eleanor Wong, Yuriy L Orlov, Weiwei Zhang, Jianming Jiang, et al. Integration of external signaling pathways with the core transcriptional network in embryonic stem cells. *Cell*, 133(6):1106–1117, 2008.

[39] Yao Chi Chen, Jon D Wright, and Carmay Lim. DR_bind: a web server for predicting DNA-binding residues from the protein structure based on electrostatics, evolution and geometry. *Nucleic Acids Research*, page gks481, 2012.

[40] Chao Cheng, Koon-Kiu Yan, Kevin Y. Yip, Joel Rozowsky, Roger Alexander, Chong Shou, and Mark Gerstein. A statistical framework for modeling gene expression using chromatin features and application to modENCODE datasets. *Genome Biology*, 12(2):R15, 2011.

[41] 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*, 2014.

[42] Peter Y Chou and Gerald D Fasman. Prediction of protein conformation. *Biochemistry*, 13(2):222–245, 1974.

[43] Wei Chu, Zoubin Ghahramani, and David L Wild. A graphical model for protein secondary structure prediction. In *Proceedings of the 21st International Conference on Machine learning*, page 21. ACM, 2004.

[44] Yongjun Chu and David R. Corey. RNA Sequencing: Platform Selection, Experimental Design, and Data Interpretation. *Nucleic Acid Therapeutics*, 22(4):271, 2012.

[45] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Proceeding of the 2012 IEEE Conference on Computer Vision and Pattern Recognition.*, pages 3642–3649. IEEE, 2012.

[46] Ivan G Costa, Helge G Roider, Thais G Do Rego, and Francisco De At De Carvalho. Predicting gene expression in T cell differentiation from histone modifications and transcription factor binding affinities by linear mixture models. *BMC Bioinformatics*, 12 Suppl 1(1):S29, 2011.

[47] James A Cuff and Geoffrey J Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[48] Daniel Curtis, Ruth Lehmann, and Phillip D Zamore. Translational regulation in development. *Cell*, 81(2):171, 1995.

[49] Qi Wen Dong, Xiao Long Wang, and Lei Lin. Application of latent semantic analysis to protein remote homology detection. *Bioinformatics (Oxford, England)*, 22(3):285, 2006.

[50] Qiwen Dong, Shuigeng Zhou, and Jihong Guan. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, 25(20):2655, 2009.

[51] Xianjun Dong, Melissa C Greven, Anshul Kundaje, Sarah Djebali, James B Brown, Chao Cheng, Thomas R Gingeras, Mark Gerstein, Roderic Guigó, and Ewan Birney. Modeling gene expression using chromatin features in various cellular contexts. *Genome Biology*, 13(9):R53, 2012.

[52] Iris Dror, Remo Rohs, and Yael Mandel-Gutfreund. How motif environment influences transcription factor search dynamics: Finding a needle in a haystack. *BioEssays*, 38(7):605–612, 2016.

[53] Jason Ernst and Manolis Kellis. Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. *Nature Biotechnology*, 33(4):364, 2015.

[54] Martin C. Frith, Michael C. Li, and Zhiping Weng. Cluster-Buster: Finding dense clusters of motifs in DNA sequences. *Nucleic Acids Research*, 31(13):3666, 2003.

[55] Olivier Gascuel and JL Golmard. A simple method for predicting the secondary structure of globular proteins: implications and accuracy. *Bioinformatics*, 4(3):357–365, 1988.

[56] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2009.

[57] Jian Guo, Hu Chen, Zhirong Sun, and Yuanlie Lin. A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *Proteins: Structure, Function, and Bioinformatics*, 54(4):738–743, 2004.

[58] Mika Gustafsson and Michael Hörnquist. Gene expression prediction by soft integration and the elastic net—best performance of the DREAM3 gene expression challenge. *PoS One*, 5(2):e9134, 2010.

[59] Christopher T Harbison, D Benjamin Gordon, Tong Ihn Lee, Nicola J Rinaldi, Kenzie D Macisaac, Timothy W Danford, Nancy M Hannett, Jean-Bosco Tagne, David B Reynolds, Jane Yoo, et al. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.

[60] Nathaniel D. Heintzman, Gary C. Hon, R. David Hawkins, Pouya Kheradpour, Alexander Stark, Lindsey F. Harp, Zhen Ye, Leonard K. Lee, Rhona K. Stuart, and Christina W. Ching. Histone modifications at human enhancers reflect global cell type-specific gene expression. *Nature*, 459(7243):108–12, 2009.

[61] Nathaniel D Heintzman, Rhona K Stuart, Gary Hon, Yutao Fu, Christina W Ching, R David Hawkins, Leah O Barrera, Sara Van Calcar, Chunxu Qu, Keith A Ching, et al. Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nature Genetics*, 39(3):311–318, 2007.

[62] Bich Hai Ho, Rania Mohammed Kotb Hassen, and Ngoc Tu Le. *Combinatorial Roles of DNA Methylation and Histone Modifications on Gene Expression*. Springer International Publishing, 2015.

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

[64] Dustin T Holloway, Mark Kon, and Charles De Lisi. Integrating genomic data to predict transcription factor binding. *Genome Informatics*, 16(1):83–94, 2005.

[65] Sujun Hua and Zhirong Sun. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *Journal of molecular biology*, 308(2):397–407, 2001.

[66] Seungwoo Hwang, Zhenkun Gou, and Igor B Kuznetsov. DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins. *Bioinformatics*, 23(5):634–636, 2007.

[67] Vishwanath R Iyer, Christine E Horak, Charles S Scafe, David Botstein, Michael Snyder, and Patrick O Brown. Genomic binding sites of the yeast cell-cycle transcription factors SBF and MBF. *Nature*, 409(6819):533–538, 2001.

[68] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3(3):318–356, 1961.

[69] Julie A Johnson. Pharmacogenetics: potential for individualized drug therapy through genetics. *Trends in Genetics*, 19(11):660–666, 2003.

[70] David T Jones. Improving the accuracy of transmembrane protein topology prediction using evolutionary information. *Bioinformatics*, 23(5):538–544, 2007.

[71] Susan Jones, Jonathan A Barker, Irene Nobeli, and Janet M Thornton. Using structural motif templates to identify proteins with DNA binding function. *Nucleic Acids Research*, 31(11):2811–2823, 2003.

[72] Susan Jones, Paul van Heyningen, Helen M Berman, and Janet M Thornton. Protein-DNA interactions: a structural analysis. *Journal of Molecular Biology*, 287(5):877–896, 1999.

[73] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[74] He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. Deep residual learning for image recognition. In *In Proceeding of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[75] M Karin. Too many transcription factors: positive and negative interactions. *New Biol*, 2(2):126–131, 1990.

[76] Rosa Karlić, Ho Ryun Chung, Julia Lasserre, Kristian Vlahoviček, and Martin Vingron. Histone modification levels are predictive for gene expression. *Proceedings of the National Academy of Sciences*, 107(7):2926–31, 2010.

[77] Christopher G. Kevil, Loren Walsh, F. Stephen. Laroux, Theodore Kalogeris, Matthew B. Grisham, and J. S. Alexander. An improved, rapid Northern protocol. *Biochemical and Biophysical Research Communications*, 238(2):277–9, 1997.

[78] Hyunsoo Kim and Haesun Park. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, 16(8):553–560, 2003.

[79] Tae Hoon Kim, Leah O Barrera, Ming Zheng, Chunxu Qu, Michael A Singer, Todd A Richmond, Yingnian Wu, Roland D Green, and Bing Ren. A high-resolution map of active promoters in the human genome. *Nature*, 436(7052):876–880, 2005.

[80] Hidetoshi Kono and Akinori Sarai. Structure-based prediction of DNA target sites by regulatory proteins. *Proteins: Structure, Function, and Bioinformatics*, 35(1):114–131, 1999.

[81] Tony Kouzarides. Chromatin modifications and their function. *Cell*, 128(4):693–705, 2007.

[82] Manish Kumar, Michael M Gromiha, and Gajendra PS Raghava. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC Bioinformatics*, 8(1):463, 2007.

[83] Sunil Kumar and Philipp Bucher. Predicting transcription factor site occupancy using DNA sequence intrinsic and cell-type specific chromatin features. *BMC Bioinformatics*, 17(1):S4, 2016.

[84] Vibhor Kumar, Masafumi Muratani, Nirmala Arul Rayan, Petra Kraus, Thomas Lufkin, Huck Hui Ng, and Shyam Prabhakar. Uniform, optimal signal processing of mapped deep-sequencing data. *Nature Biotechnology*, 31(7):615, 2013.

[85] Igor B Kuznetsov, Zhenkun Gou, Run Li, and Seungwoo Hwang. Using evolutionary and structural information to predict DNA-binding sites on DNA-binding proteins. *Proteins: Structure, Function, and Bioinformatics*, 64(1):19–27, 2006.

[86] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[87] Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[88] Boris Lenhard, Albin Sandelin, Luis Mendoza, Pär Engström, Niclas Jareborg, and Wyeth W Wasserman. Identification of conserved regulatory elements by comparative genome analysis. *Journal of Biology*, 2(2):13, 2003.

[89] Bi-Qing Li, Kai-Yan Feng, Juan Ding, and Yu-Dong Cai. Predicting DNA-binding sites of proteins based on sequential and 3D structural information. *Molecular Genetics Genomics*, 289(3):489–499, 2014.

[90] Bing Li, Michael Carey, and Jerry L. Workman. The role of chromatin during transcription. *Cell*, 128(4):707–719, 2007.

[91] Tao Li, Qian-Zhong Li, Shuai Liu, Guo-Liang Fan, Yong-Chun Zuo, and Yong Peng. PreDNA: accurate prediction of DNA-binding sites in proteins by integrating sequence and geometric structure information. *Bioinformatics*, 29(6):678–685, 2013.

[92] Zhen Li and Yizhou Yu. Protein secondary structure prediction using cascaded convolutional and recurrent neural networks. *arXiv preprint arXiv:1604.07176*, 2016.

[93] Kaj Ulrik Linderstrøm-Lang. *Lane medical lectures: proteins and enzymes*, volume 6. Stanford University Press, 1952.

[94] Bin Liu, Xiaolong Wang, Lin Lei, Qiwen Dong, and Wang Xuan. A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. *BMC Bioinformatics*, 9(1):510, 2008.

[95] Bin Liu, Jinghao Xu, Quan Zou, Ruifeng Xu, Xiaolong Wang, and Qingcai Chen. Using distances between top-n-gram and residue pairs for protein remote homology detection. *BMC Bioinformatics*, 15(2):S3, 2014.

[96] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017.

[97] Rong Liu and Jianjun Hu. Computational prediction of heme-binding residues by exploiting residue interaction network. *PLoS One*, 6(10):e25560, 2011.

[98] Rong Liu and Jianjun Hu. DNABind: A hybrid algorithm for structure-based prediction of DNA-binding residues by combining machine learning-and template-based approaches. *Proteins: Structure, Function, and Bioinformatics*, 81(11):1885–1899, 2013.

[99] Marek Los, Subbareddy Maddika, Bettina Erb, and Klaus Schulze-Osthoff. Switching Akt: from survival signaling to deadly response. *Bioessays News and Reviews in Molecular Cellular and Developmental Biology*, 31(5):492–495, 2010.

[100] Yi-Fan Lu, David B. Goldstein, Misha Angrist, and Gianpiero Cavalleri. Personalized medicine and human genetic diversity. *Cold Spring Harbor Perspectives in Medicine*, 4(9):a008581, 2014.

[101] Nicholas M Luscombe, Susan E Austin, Helen M Berman, and Janet M Thornton. An overview of the structures of protein-DNA complexes. *Genome Biology*, 1(1):1, 2000.

[102] Nicholas M Luscombe, Roman A Laskowski, and Janet M Thornton. Amino acid–base interactions: a three-dimensional analysis of protein–DNA interactions at an atomic level. *Nucleic Acids Research*, 29(13):2860–2874, 2001.

[103] Nicholas M Luscombe and Janet M Thornton. Protein–DNA interactions: amino acid conservation and the effects of mutations on binding specificity. *Journal of Molecular Biology*, 320(5):991–1009, 2002.

[104] Xin Ma, Jing Guo, Hong-De Liu, Jian-Ming Xie, and Xiao Sun. Sequence-based prediction of DNA-binding residues in proteins with conservation and correlation information. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(6):1766–1775, 2012.

[105] Xin Ma, Jian-Sheng Wu, Hong-De Liu, Xi-Nan Yang, Jian-Ming Xie, and Xiao Sun. SVM-based approach for predicting DNA-binding residues in proteins from amino acid sequences. In *Proceeding of the 2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing.*, pages 225–229. IEEE, 2009.

[106] Tsz-Kwong Man and Gary D Stormo. Non-independence of Mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Research*, 29(12):2471–2478, 2001.

[107] Yael Mandel-Gutfreund and Hanah Margalit. Quantitative parameters for amino acid-base interaction: implications for prediction of protein-DNA binding sites. *Nucleic Acids Research*, 26(10):2306–2312, 1998.

[108] Voichita D Marinescu, Isaac S Kohane, and Alberto Riva. MAPPER: a search engine for the computational identification of putative transcription factor binding sites in multiple genomes. *BMC Bioinformatics*, 6(1):79, 2005.

[109] Anthony Mathelier, Oriol Fornes, David J. Arenillas, Chih-yu Chen, Grégoire Denay, Jessica Lee, Wenqiang Shi, Casper Shyr, Ge Tan, Rebecca Worsley-Hunt, et al. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Research*, 44(Database issue):D110–D115, 2016.

[110] Anthony Mathelier and Wyeth W Wasserman. The next generation of transcription factor binding site prediction. *PLoS Computational Biology*, 9(9):e1003214, 2013.

[111] Anthony Mathelier, Beibei Xin, Tsu-Pei Chiu, Lin Yang, Remo Rohs, and Wyeth W Wasserman. DNA shape features improve transcription factor binding site predictions in vivo. *Cell Systems*, 3(3):278–286, 2016.

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

[113] Liam J McGuffin, Kevin Bryson, and David T Jones. The PSIPRED protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[114] Robert C Mcleay, Tom Lesluyes, Gabriel Cuellar Partida, and Timothy L Bailey. Genome-wide in silico prediction of gene expression. *Bioinformatics*, 28(21):2789–2796, 2012.

[115] Jens Meiler and David Baker. Rapid protein fold determination using unassigned NMR data. *Proceedings of the National Academy of Sciences*, 100(26):15404–15409, 2003.

[116] Alexander Meissner, Tarjei S Mikkelsen, Hongcang Gu, Marius Wernig, Jacob Hanna, Andrey Sivachenko, Xiaolan Zhang, Bradley E Bernstein, Chad Nusbaum, David B Jaffe, et al. Genome-scale DNA methylation maps of pluripotent and differentiated cells. *Nature*, 454(7205):766–770, 2008.

[117] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(3):530–539, 2015.

[118] Tarjei S Mikkelsen, Manching Ku, David B Jaffe, Biju Issac, Erez Lieberman, Georgia Giannoukos, Pablo Alvarez, William Brockman, Tae-Kyung Kim, Richard P Koche, et al. Genome-wide maps of chromatin state in pluripotent and lineage-committed cells. *Nature*, 448(7153):553–560, 2007.

[119] Ryan D. Morin, Matthew Bainbridge, Anthony Fejes, Martin Hirst, Martin Krzywinski, Trevor J. Pugh, Helen Mcdonald, Richard Varhol, Steven J. M. Jones, and Marco A. Marra. Profiling the HeLa-S3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *Biotechniques*, 45(1):81–94, 2008.

[120] Sonali Mukherjee, Michael F Berger, Ghil Jona, Xun S Wang, Dale Muzzey, Michael Snyder, Richard A Young, and Martha L Bulyk. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nature Genetics*, 36(12):1331–1339, 2004.

[121] R. Nagarajan, Shandar Ahmad, and M. Michael Gromiha. Novel approach for selecting the best predictor for identifying the binding sites in DNA binding proteins. *Nucleic Acids Research*, 41(16):7606–7614, 2013.

[122] Anirudh Natarajan, Galip Gürkan Yardımcı, Nathan C. Sheffield, Gregory E. Crawford, and Uwe Ohler. Predicting cell-type-specific gene expression from regions of open chromatin. *Genome Research*, 22(9):1711–1722, 2012.

[123] Truyen Nguyen, Paul Nioi, and Cecil B. Pickett. The Nrf2-Antioxidant Response Element Signaling Pathway and Its Activation by Oxidative Stress. *Journal of Biological Chemistry*, 284(20):13291, 2009.

[124] Martin EM Noble, Jane A Endicott, and Louise N Johnson. Protein kinase inhibitors: insights into drug design from structure. *Science*, 303(5665):1800–1805, 2004.

[125] Christopher J. O'Donnell and Elizabeth G. Nabel. Genomics of cardiovascular disease. *New England Journal of Medicine*, 365(22):2098–109, 2011.

[126] Yanay Ofran, Venkatesh Mysore, and Burkhard Rost. Prediction of DNA-binding residues from sequence. *Bioinformatics*, 23(13):i347–i353, 2007.

[127] Wilma K Olson, Andrey A Gorin, Xiang-Jun Lu, Lynette M Hock, and Victor B Zhurkin. DNA sequence-dependent deformability deduced from protein–DNA crystal complexes. *Proceedings of the National Academy of Sciences*, 95(19):11163–11168, 1998.

[128] Christine A Orengo, AD Michie, S Jones, David T Jones, MB Swindells, and Janet M Thornton. CATH–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[129] Pemra Ozbek, Seren Soner, Burak Erman, and Turkan Haliloglu. PDNABIND-PROT: fluctuation-based predictor of DNA-binding residues within a network of interacting residues. *Nucleic Acids Research*, page gkq396, 2010.

[130] Kimmo Palin, Jussi Taipale, and Esko Ukkonen. Locating potential enhancer elements by comparative genomics using the EEL software. *Nature Protocols*, 1(1):368–74, 2006.

[131] Keun-Joon Park and Minoru Kanehisa. Prediction of protein subcellular locations by support vector machines using compositions of amino acids and amino acid pairs. *Bioinformatics*, 19(13):1656–1663, 2003.

[132] Subhankar Paul. Dysfunction of the ubiquitin-proteasome system in multiple disease conditions: therapeutic approaches. *Bioessays*, 30(11-12):1172–1184, 2010.

[133] John T Pelton and Larry R McLean. Spectroscopic methods for analysis of protein secondary structure. *Analytical biochemistry*, 277(2):167–176, 2000.

[134] Gianluca Pollastri, Darisz Przybylski, Burkhard Rost, and Pierre Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins: Structure, Function, and Bioinformatics*, 47(2):228–235, 2002.

[135] Chris P Ponting, Jörg Schultz, Frank Milpetz, and Peer Bork. SMART: identification and annotation of domains from signalling and extracellular protein sequences. *Nucleic Acids Research*, 27(1):229–232, 1999.

[136] Mark Ptashne. Regulation of transcription: from lambda to eukaryotes. *Trends in Biochemical Sciences*, 30(6):275–279, 2005.

[137] Zhaohui S Qin, Jianjun Yu, Jincheng Shen, Christopher A Maher, Hu Ming, Shanker Kalyana-Sundaram, Jindan Yu, and Arul M Chinnaiyan. HPeak: an HMM-based algorithm for defining read-enriched regions in ChIP-seq data. *BMC Bioinformatics*, 11(1):369, 2010.

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

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

[140] Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993.

[141] Burkhard Rost and Chris Sander. Conservation and prediction of solvent accessibility in protein families. *Proteins: Structure, Function, and Bioinformatics*, 20(3):216–226, 1994.

[142] Jianhua Ruan. A top-performing algorithm for the DREAM3 gene expression prediction challenge. *PLoS One*, 5(2):e8944, 2010.

[143] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[144] Alejandro A Schäffer, L Aravind, Thomas L Madden, Sergei Shavirin, John L Spouge, Yuri I Wolf, Eugene V Koonin, and Stephen F Altschul. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Research*, 29(14):2994–3005, 2001.

[145] John A Schellman and Charlotte G Schellman. Kaj ulrik linderstrøm-lang (1896–1959). *Protein science*, 6(5):1092–1100, 1997.

[146] Scott C Schmidler, Jun S Liu, and Douglas L Brutlag. Bayesian segmentation of protein secondary structure. *Journal of computational biology*, 7(1-2):233–248, 2000.

[147] Florian Schmidt, Nina Gasparoni, Gilles Gasparoni, Kathrin Gianmoena, Cristina Cadenas, Julia K. Polansky, Peter Ebert, Karl Nordström, Matthias Barann, and Anupam Sinha. Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction. *Nucleic Acids Research*, 45(1):54–66, 2016.

[148] Jingna Si, Zengming Zhang, Biaoyang Lin, Michael Schroeder, and Bingding Huang. MetaDBSite: a meta approach to improve protein DNA-binding sites prediction. *BMC Systems Biology*, 5(1):S7, 2011.

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

[150] M Sieber and Rudolf Konrad Allemann. Arginine (348) is a major determinant of the dna binding specificity of transcription factor E12. *Biological Chemistry*, 379(6):731–735, 1998.

[151] VA Simossis and J Heringa. Integrating protein secondary structure prediction and multiple sequence alignment. *Current Protein and Peptide Science*, 5(4):249–266, 2004.

[152] Richard J Simpson and Francis J Morgan. Complete amino acid sequence of Embden goose (anser anser) egg-white lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, 744(3):349–351, 1983.

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

[154] Saurabh Sinha, Yupu Liang, and Eric Siggia. Stubb: a program for discovery and analysis of cis-regulatory modules. *Nucleic Acids Research*, 34(Web Server issue):555–9, 2006.

[155] Lingyun Song, Zhancheng Zhang, Linda L. Grasfeder, Alan P. Boyle, Paul G. Giresi, Bum Kyu Lee, Nathan C. Sheffield, Stefan Gräf, Mikael Huss, and Damian Keefe. Open chromatin defined by DNaseIa and FAIRE identifies regulatory elements that shape cell-type identity. *Genome Research*, 21(10):1757–1767, 2011.

[156] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Proceeding of the 32nd Conference on Advances in Neural Information Processing Systems*, pages 2377–2385, 2015.

[157] Michael J. E. Sternberg, Henry A. Gabb, Richard M. Jackson, and Gidon Moont. Protein-protein docking. *Methods in Molecular Biology*, pages 399–415, 2000.

[158] Gustavo Stolovitzky, Don Monroe, and Andrea Califano. Dialogue on reverse-engineering assessment and methods: the DREAM of high-throughput pathway inference. *Annals of the New York Academy of Sciences*, 1115(1):1, 2007.

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

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

[161] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proceeding of the 31st Conference on Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[162] John A Swets et al. Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293, 1988.

[163] András Szilágyi and Jeffrey Skolnick. Efficient prediction of nucleic acid binding function from low-resolution protein structures. *Journal of Molecular Biology*, 358(3):922–933, 2006.

[164] Mohammad Talebzadeh and Fatemeh Zare-Mirakabad. Transcription factor binding sites prediction based on modified nucleosomes. *PLoS One*, 9(2):e89226, 2014.

[165] Floyd E. Taub, James M. Deleo, and E. Brad Thompson. Sequential comparative hybridizations analyzed by computerized image processing can identify and quantitate regulated RNAs. *DNA*, 2(4):309–27, 1983.

[166] Harianto Tjong and Huan-Xiang Zhou. DISPLAR: an accurate method for predicting DNA-binding sites on protein surfaces. *Nucleic Acids Research*, 35(5):1465–1477, 2007.

[167] Zing Tsung-Yeh Tsai, Shin-Han Shiu, and Huai-Kuang Tsai. Contribution of sequence motif, chromatin state, and DNA structure features to predictive models of transcription factor binding in yeast. *PLoS Computational Biology*, 11(8):e1004418, 2015.

[168] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Proceeding of the 27th Annual Conference on Advances in neural information processing systems*, pages 2643–2651, 2013.

[169] Vladimir N. Vapnik. The nature of statistical learning theory. *IEEE Transactions on Neural Networks*, 8(6):1564–1564, 1997.

[170] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[171] Reiner A. Veitia. One thousand and one ways of making functionally similar transcriptional enhancers. *Bioessays*, 30(11-12):1052–1057, 2010.

[172] V Veljkovic, N Veljkovic, JA Este, A Huther, and U Dietrich. Application of the EIIP/ISM bioinformatics concept in development of new drugs. *Current Medicinal Chemistry*, 14(4):441–453, 2007.

[173] Marcel L. Verdonk, Jason C. Cole, Michael J. Hartshorn, Christopher W. Murray, and Richard D. Taylor. Improved protein–ligand docking using gold. *Proteins-structure Function Bioinformatics*, 52(4):609–623, 2010.

[174] Ruchi Verma, Grish C Varshney, and GPS Raghava. Prediction of mitochondrial proteins of malaria parasite using split amino acid composition and PSSM profile. *Amino Acids*, 39(1):101–110, 2010.

[175] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Proceeding of the 32nd Conference on Advances in Neural Information Processing Systems*, pages 2773–2781, 2015.

[176] Karl V. Voelkerding, Shale A. Dames, and Jacob D. Durtschi. Next-Generation Sequencing: From Basic Research to Diagnostics. *Clinical Chemistry*, 55(4):641–658, 2009.

[177] Donald Voet, Judith G Voet, and Charlotte W Pratt. Fundamentals of biochemistry : life at the molecular level. *Fundamentals of Biochemistry Life at the Molecular Level*, (2):484, 2012.

[178] Michael Wagner, Rafal Adamczak, Aleksey Porollo, and Jaroslaw Meller. Linear regression models for solvent accessibility prediction in proteins. *Journal of Computational Biology*, 12(3):355–369, 2005.

[179] Guoli Wang and Roland L Dunbrack Jr. PISCES: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[180] Liangjiang Wang and Susan J Brown. BindN: a web-based tool for efficient prediction of DNA and RNA binding sites in amino acid sequences. *Nucleic Acids Research*, 34(suppl 2):W243–W248, 2006.

[181] Liangjiang Wang, Caiyan Huang, Mary Qu Yang, and Jack Y Yang. BindN+ for accurate prediction of DNA and RNA-binding residues from protein sequence features. *BMC Systems Biology*, 4(1):1, 2010.

[182] Liangjiang Wang, Mary Qu Yang, and Jack Y Yang. Prediction of DNA-binding residues from protein sequence information using random forests. *BMC Genomics*, 10(1):1, 2009.

[183] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6, 2016.

[184] Zhibin Wang, Chongzhi Zang, Jeffrey A. Rosenfeld, Dustin E. Schones, Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Weiqun Peng, Michael Q., and Keji Zhao. Combinatorial patterns of histone acetylations and methylations in the human genome. *Nature Genetics*, 40(7):897–903, 2008.

[185] Zhiyong Wang, Feng Zhao, Jian Peng, and Jinbo Xu. Protein 8-class secondary structure prediction using conditional neural fields. *Proteomics*, 11(19):3786–3792, 2011.

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

[187] Wyeth W Wasserman and Albin Sandelin. Applied bioinformatics for the identification of regulatory elements. *Nature Reviews Genetics*, 5(4):276–287, 2004.

[188] 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.

[189] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.

[190] Kyoung-Jae Won, Bing Ren, and Wei Wang. Genome-wide prediction of transcription factor binding sites using an integrated model. *Genome Biology*, 11(1):R7, 2010.

[191] R. P. Woychik, M. L. Klebig, M. J. Justice, T. R. Magnuson, E. D. Avner, and E. D. Avrer. Functional genomics in the post-genome era. *Mutat Res*, 400(1-2):3–14, 1998.

[192] Yi Xiong, Juan Liu, and Dong-Qing Wei. An accurate feature-based method for identifying DNA-binding residues on protein surfaces. *Proteins: Structure, Function, and Bioinformatics*, 79(2):509–517, 2011.

[193] Ruifeng Xu, Jiyun Zhou, Bin Liu, Yulan He, Quan Zou, Xiaolong Wang, and Kuo-Chen Chou. Identification of DNA-binding proteins by incorporating evolutionary information into pseudo amino acid composition via the top-n-gram approach. *Journal of Biomolecular Structure and Dynamics*, 33(8):1720–1730, 2015.

[194] Ruifeng Xu, Jiyun Zhou, Hongpeng Wang, Yulan He, Xiaolong Wang, and Bin Liu. Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. *BMC Systems Biology*, 9(1):S10, 2015.

[195] Changhui Yan, Michael Terribilini, Feihong Wu, Robert L Jernigan, Drena Dobbs, and Vasant Honavar. Predicting DNA-binding sites of proteins from amino acid sequence. *BMC Bioinformatics*, 7(1):262, 2006.

[196] Ashraf Yaseen and Yaohang Li. Context-based features enhance protein secondary structure prediction accuracy. *Journal of chemical information and modeling*, 54(3):992–1002, 2014.

[197] Ashraf Yaseen and Yaohang Li. Template-based C8-SCORPION: a protein 8-state secondary structure prediction method using structural information and context-based features. *BMC Bioinformatics*, 15(8):S3, 2014.

[198] Wen-tau Yih, Kristina Toutanova, John C Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In *Proceedings of the 15th*

Conference on Computational Natural Language Learning, pages 247–256. Association for Computational Linguistics, 2011.

[199] Chin-Sheng Yu, Yu-Ching Chen, Chih-Hao Lu, and Jenn-Kang Hwang. Prediction of protein subcellular localization. *Proteins: Structure, Function, and Bioinformatics*, 64(3):643–651, 2006.

[200] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *In Proceeding of the 3rd International Conference on Learning Representations*, pages 770–778, 2015.

[201] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *In Proceeding of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 636–644, 2017.

[202] Federico Zambelli, Graziano Pesole, and Giulio Pavesi. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. *Briefings in Bioinformatics*, 14(2):225–237, 2012.

[203] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[204] Lu-Qiang. Zhang and Qian-Zhong Li. Estimating the effects of transcription factors binding and histone modifications on gene expression levels in human cells. *Oncotarget*, 8(25):40090–40103, 2017.

[205] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceeding of the 32nd Conference on Advances in Neural Information Processing Systems*, pages 649–657, 2015.

[206] Yong Zhang, Tao Liu, Clifford A. Meyer, Jérǒme Eeckhoute, David S. Johnson, Bradley E. Bernstein, Chad Nusbaum, Richard M. Myers, Myles Brown, Wei Li, and Liu X Shirley. Model-based analysis of ChIP-Seq (MACS). *Genome Biology*, 9(9):R137, 2008.

[207] Xiaowei Zhao, Xiangtao Li, Zhiqiang Ma, and Minghao Yin. Prediction of lysine ubiquitylation with ensemble classifier and feature selection. *International journal of molecular sciences*, 12(12):8347–8361, 2011.

[208] Yue Zhao, Shuxiang Ruan, Manishi Pandey, and Gary D Stormo. Improved models for transcription factor binding site identification using nonindependent interactions. *Genetics*, 191(3):781–790, 2012.

[209] Jian Zhou and Olga Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In *Proceeding of the 31st International Conference on Machine Learning*, pages 745–753, 2014.

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

[211] Jiyun Zhou, Qin Lu, Ruifeng Xu, Lin Gui, and Hongpeng Wang. CNNsite: Prediction of DNA-binding residues in proteins using convolutional neural network with sequence features. In *Proceeding of the 2016 IEEE International Conference on Bioinformatics and Biomedicine*, pages 78–85. IEEE, 2016.

[212] Jiyun Zhou, Qin Lu, Ruifeng Xu, Lin Gui, and Hongpeng Wang. Prediction of DNA-binding residues from sequence information using convolutional neural network. *International Journal of Data Mining and Bioinformatics*, 17(2):132–152, 2017.

[213] Jiyun Zhou, Qin Lu, Ruifeng Xu, Yulan He, and Hongpeng Wang. EL_PSSM-RT: DNA-binding residue prediction by integrating ensemble learning with PSSM Relation Transformation. *BMC Bioinformatics*, 18(1):379, 2017.

[214] Jiyun Zhou, Hongpeng Wang, Zhishan Zhao, Ruifeng Xu, and Qin Lu. CNNH_PSS: protein 8-class secondary structure prediction by convolutional neural network with highway. *BMC Bioinformatics*, 19(4):60, May 2018.

[215] Jiyun Zhou, Ruifeng Xu, Yulan He, Qin Lu, Hongpeng Wang, and Bing Kong. PDNAsite: identification of DNA-binding site from protein sequence by incorporating spatial and sequence context. *Scientific Reports*, 6, 2016.

[216] Tianyin Zhou, Ning Shen, Lin Yang, Namiko Abe, John Horton, Richard S Mann, Harmen J Bussemaker, Raluca Gordân, and Remo Rohs. Quantitative modeling of transcription factor binding specificities using DNA shape. *Proceedings of the National Academy of Sciences*, 112(15):4654–4659, 2015.