Computational Inference of Genome-Wide Protein-DNA Interactions Using High-Throughput Genomic Data

by

Jianling Zhong

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: ____________________

Approved:

__________________________________________
Alexander J. Hartemink, Supervisor

__________________________________________
Gregory E. Crawford

__________________________________________
Sayan Mukherjee

__________________________________________
Raluca M. Gordân

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program of Computational Biology and Bioinformatics in the Graduate School of Duke University

2015
Abstract

Computational Inference of Genome-Wide Protein-DNA Interactions Using High-Throughput Genomic Data

by

Jianling Zhong

Graduate Program in Computational Biology and Bioinformatics
Duke University

Date: __________________

Approved:

_____________________
Alexander J. Hartemink, Supervisor

_____________________
Gregory E. Crawford

_____________________
Sayan Mukherjee

_____________________
Raluca M. Gordân

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Graduate Program in Computational Biology and Bioinformatics in the Graduate School of Duke University

2015
Abstract

Transcriptional regulation has been studied intensively in recent decades. One important aspect of this regulation is the interaction between regulatory proteins, such as transcription factors (TF) and nucleosomes, and the genome. Different high-throughput techniques have been invented to map these interactions genome-wide, including ChIP-based methods (ChIP-chip, ChIP-seq, etc.), nuclease digestion methods (DNase-seq, MNase-seq, etc.), and others. However, a single experimental technique often only provides partial and noisy information about the whole picture of protein-DNA interactions. Therefore, the overarching goal of this dissertation is to provide computational developments for jointly modeling different experimental datasets to achieve a holistic inference on the protein-DNA interaction landscape.

We first present a computational framework that can incorporate the protein binding information in MNase-seq data into a thermodynamic model of protein-DNA interaction. We use a correlation-based objective function to model the MNase-seq data and a Markov chain Monte Carlo method to maximize the function. Our results show that the inferred protein-DNA interaction landscape is concordant with the MNase-seq data and provides a mechanistic explanation for the experimentally collected MNase-seq fragments. Our framework is flexible and can easily incorporate other data sources. To demonstrate this flexibility, we use prior distributions to integrate experimentally measured protein concentrations.

We also study the ability of DNase-seq data to position nucleosomes. Tradition-
ally, DNase-seq has only been widely used to identify DNase hypersensitive sites, which tend to be open chromatin regulatory regions devoid of nucleosomes. We reveal for the first time that DNase-seq datasets also contain substantial information about nucleosome translational positioning, and that existing DNase-seq data can be used to infer nucleosome positions with high accuracy. We develop a Bayes-factor–based nucleosome scoring method to position nucleosomes using DNase-seq data. Our approach utilizes several effective strategies to extract nucleosome positioning signals from the noisy DNase-seq data, including jointly modeling data points across the nucleosome body and explicitly modeling the quadratic and oscillatory DNase I digestion pattern on nucleosomes. We show that our DNase-seq–based nucleosome map is highly consistent with previous high-resolution maps. We also show that the oscillatory DNase I digestion pattern is useful in revealing the nucleosome rotational context around TF binding sites.

Finally, we present a state-space model (SSM) for jointly modeling different kinds of genomic data to provide an accurate view of the protein-DNA interaction landscape. We also provide an efficient expectation-maximization algorithm to learn model parameters from data. We first show in simulation studies that the SSM can effectively recover underlying true protein binding configurations. We then apply the SSM to model real genomic data (both DNase-seq and MNase-seq data). Through incrementally increasing the types of genomic data in the SSM, we show that different data types can contribute complementary information for the inference of protein binding landscape and that the most accurate inference comes from modeling all available datasets.

This dissertation provides a foundation for future research by taking a step toward the genome-wide inference of protein-DNA interaction landscape through data integration.
I dedicate this dissertation to my parents.
Contents

Abstract iv
List of Tables xii
List of Figures xiii
List of Abbreviations and Symbols xxiv
Acknowledgments xxvi

1 Introduction 1

1.1 Eukaryotic gene transcription . . . . . . . . . . . . . . . . . . . . . 5
  1.1.1 Transcription initiation . . . . . . . . . . . . . . . . . . . . . . 6
  1.1.2 Transcription elongation . . . . . . . . . . . . . . . . . . . . . 8
  1.1.3 Transcription termination . . . . . . . . . . . . . . . . . . . . 9

1.2 Transcription factors and their roles in transcriptional regulation . . 10
  1.2.1 General properties of transcription factors . . . . . . . . . . . . 10
  1.2.2 Transcriptional regulation by transcription factors . . . . . . . 13
  1.2.3 The continuous, competitive yet cooperative nature of transcrip-
  tion factors binding . . . . . . . . . . . . . . . . . . . . . . . 15

1.3 Role of chromatin in transcriptional regulation . . . . . . . . . . . . 17

1.4 Experimental methods in studying protein-DNA interactions . . . . 18
  1.4.1 ChIP-based methods . . . . . . . . . . . . . . . . . . . . . . . 18
  1.4.2 Nuclease-digestion–based methods . . . . . . . . . . . . . . . 20
  1.4.3 Other in vivo methods . . . . . . . . . . . . . . . . . . . . . . 20
1.4.4  \textit{In vitro} methods for studying TF binding specificity . . . . . .  21
1.4.5 Large-scale collaborative projects in studying gene transcription  22
1.5 Computational methods in studying protein-DNA interactions . . . .  22
   1.5.1 Overview of a biophysical model for multi-factor binding . . .  25
1.6 Organization of this dissertation .............................................  27

2 Learning a multi-factor binding model from MNase-seq data 30
   2.1 Methods ...........................................................................  31
      2.1.1 Model formulation and model learning objective ...............  31
      2.1.2 Using paired-end MNase-seq data as measures of genomic oc-
            cupancy levels of DNA-binding proteins .........................  34
      2.1.3 Incorporating MNase-seq data through an objective function .  36
      2.1.4 Selecting a subset of TFs and promoter regions ...............  38
      2.1.5 Incorporating the pre-initiation complex .........................  40
      2.1.6 Inference method .........................................................  43
      2.1.7 Framework implementation ............................................  46
   2.2 Results .............................................................................  46
      2.2.1 Overall inference performance evaluated by cross-validation .  46
      2.2.2 A mechanistic explanation for paired-end MNase-seq data ....  47
      2.2.3 Incorporating measurements of protein concentration through
            prior distributions ...........................................................  50
      2.2.4 MNase sequence preference does not bias the MNase-seq cov-
            erage data .....................................................................  52
      2.2.5 Specific framework parameters do not influence the overall in-
            ference results .................................................................  53
   2.3 Discussion .........................................................................  54

3 Mapping genome-wide nucleosomes using DNase-seq data 58
   3.1 Methods ...........................................................................  60
4 A multivariate state-space model for integrating genomic data

4.1 A multi-factor binding model with a proper likelihood function

4.2 Multivariate state-space model (SSM)

4.2.1 Model definition

4.2.2 Sampling distributions in SSM

4.3 General expectation maximization (EM) theory

4.4 Learning model parameters through EM

4.4.1 Updating transition probability parameters through EM

4.4.2 Updating sampling distribution parameters through EM

4.4.3 Updating model parameters with multiple DNA sequences

4.5 Model implementation

4.6 Model inference results

4.6.1 Inference using simulated data

4.6.2 Inference using DNase-seq data

4.7 Incorporating MNase-seq data

4.8 Discussion

5 Future directions, discussions, and conclusions

5.1 Possible improvements to current modeling approach

5.1.1 Handling bias in genomic data

5.1.2 Spatial considerations of DBF binding

5.1.3 Alternative models to PWM

5.1.4 Including other kinds of data

5.2 An alternative modeling framework

5.2.1 Shall we decouple the binding probability calculation and the likelihood calculation?

5.2.2 The convolution process entailed in generating genomic data
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3 Future work and applications</td>
<td>145</td>
</tr>
<tr>
<td>5.3.1 Creating a new resource to describe genome-wide protein-DNA</td>
<td>145</td>
</tr>
<tr>
<td>interaction landscapes</td>
<td></td>
</tr>
<tr>
<td>5.3.2 <em>In silico</em> screening for biologically important single nucleotide</td>
<td>145</td>
</tr>
<tr>
<td>polymorphism</td>
<td></td>
</tr>
<tr>
<td>5.3.3 Predicting transcription rate based on protein-DNA interaction</td>
<td>146</td>
</tr>
<tr>
<td>landscape</td>
<td></td>
</tr>
<tr>
<td>5.4 Conclusions</td>
<td>147</td>
</tr>
<tr>
<td>A Supplemental materials for Chapter 2</td>
<td>148</td>
</tr>
<tr>
<td>Bibliography</td>
<td>155</td>
</tr>
<tr>
<td>Biography</td>
<td>169</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Transcription factors used in the framework .......................... 39
2.2 Promoter regions used in the framework ............................... 41
2.3 Parameter values for the sensitivity analysis ......................... 54
List of Figures

1.1 Typical organization of eukaryotic genes. Figure adapted from Courey (2009). ................................................................. 6

1.2 Different pathways to assemble the PIC in eukaryotic genes. Figure adapted from Sikorski and Buratowski (2009). ............... 7

1.3 Schematic structure of a zinc finger domain. Figure adapted from Latchman (2010). ............................................................ 12

1.4 The crystal structure of bZIP (left) and bHLH domains (right). Figure adapted from Berman et al. (2008) .............................. 12

1.5 An example binding configuration near a promoter region. In this configuration, four nucleosomes (Nuc) and two TFs are included. . 25

1.6 The COMPETE model structure. A blue circular node corresponds to a single state within a DBF. COMPETE also uses silent states that do not emit any nucleotides to facilitate transition (dashed circular nodes). A module of states, such as a TF or a nucleosome, is shown in an orange rectangle. (A) The overall structure of COMPETE. (B) A module of nucleosome. (C) and (D) A module of a TF and an origin recognition complex (ORC), respectively. Figure adapted from Wasson and Hartemink (2009). .................................................. 28

1.7 Calculation of the probability that a DBF \( \pi_k \) binds to a given genomic location is simply a summation of probabilities of all relevant binding configurations in the ensemble. ............................................ 29
2.1 Erroneous predictions of COMPETE when the transition weights are set incorrectly. Shown in the figure are COMPETE predictions on six promoter regions. The transition weights used in the prediction are randomly sampled from the uniform prior distributions mentioned in Section 2.1.6. In almost all cases, TFs (color legends not shown) completely occupied the entire promoter region. Comparing these random profiles with the fitted profiles in Figure 2.5 shows the importance of learning from data.

2.2 Overview of how the objective function is evaluated and iteratively optimized. (A) Predicted probability that each particular DBF binds at a given genomic position, as calculated by COMPETE, given current DBF weights. We then separate these probabilities into two profiles: (B) predicted nucleosome binding profile and (C) predicted composite TF binding profile in which protein identities have been removed; the latter is smoothed to make it comparable to a short fragment coverage profile. Similarly, we separate the observed MNase-seq fragments (F) into long (140–200 bp) and short (0–100 bp) fragments, which are summed to produce measures of coverage. (D) Total long fragment coverage is processed into a large protein binding profile, which is compared to predicted nucleosomal binding, arriving at Pearson correlation $r_1$. (E) Total short fragment coverage is processed into a small protein binding profile, which is compared to predicted composite TF binding, arriving at Pearson correlation $r_2$. For this promoter, the quantity $h$ that appears in our objective function (the pseudo-likelihood) is simply the geometric mean of the two correlations, after they are rescaled to lie in the interval $[0, 1]$:

$$h = \frac{1}{2} \sqrt{(1 + r_1) \times (1 + r_2)}.$$  The complete pseudo-likelihood over all promoters is then optimized with respect to the DBF weights using the inference method described in Section 2.1.6.
2.3 PIC protection shape downstream of TATA elements. (A) We selected the 100 most highly transcribed promoters, ranked by the number of NET-seq (nascent elongation transcription sequencing) reads (Churchman and Weissman, 2011) mapping to 500 bp downstream of the TATA element. From these, we compute and plot the average long (black) and short (blue) fragment coverage. The plot reveals a PIC protection peak in the average short fragment coverage, located approximately 40 bp downstream of TATA element. This is consistent with the findings of Rhee and Pugh (2012). (B) We processed the average short fragment coverage the same way as we processed MNase-seq short fragment coverage data through thresholding and smoothing. The resultant PIC protection shape (0–100 bp downstream of TATA element) was extracted from the smoothed profile. This protection shape (multiplied by the corresponding COMPETE predicted TBP binding probability) is added to predicted small protein binding probability downstream of the TATA element.

2.4 Comparison of cross validated inference performance to various baselines. Data from the 81 promoter regions were split into nine equal parts. A standard nine-fold cross-validation procedure was applied: 72 promoter regions were used as training data to obtain trained DBF weights $\hat{X}$; we then calculated $h(\hat{X})$ values of the held out nine promoter regions (testing results). “CV training result” considers the $h(\hat{X})$ values for each promoter when used as training data. “CV testing result” shows the $h(\hat{X})$ values for each promoter when used as testing data. Uniformly drawn TF transition weights and different multiples of $K_d$ are used as baseline comparisons. Variance is reduced in the random baseline case because each result is the average of 1,000 random samples.

2.5 Predicted binding profiles versus MNase-seq binding profiles. For six promoter regions in our 81 promoter set, we plot the predicted binding profiles when they were evaluated as testing data. We also indicate reported binding sites from ChIP-exo (Rhee and Pugh, 2011) underneath the predicted binding profiles; these have the same color as the corresponding TF’s binding probability. No binding event is reported by MacIsaac et al. (2006, p-value < 0.001 and conservation level 3) for these promoter regions.
2.6 The fitted protein-DNA interaction landscape provides mechanistic explanations for the observed MNase paired-end sequencing fragments. Top panels, the observed MNase-seq fragments, plotted as horizontal bars in a genome coordinate vs. fragment size 2D plot. Bottom panels, the fitted protein-DNA interaction landscapes. The fitted landscapes explain the fragments in terms of the binding proteins and their respective binding probabilities.

2.7 Comparison of cross-validation performance with and without prior information regarding measured cellular protein concentration. Performance for each promoter is measured by the geometric mean \((h(\hat{X}))\) of the two Pearson correlations defined in Figure 2.2. Each boxplot shows the performance summary of the 81 promoter regions across all the cross-validation trials.

2.8 Cut percentage is notably different from (and heavily influenced by) the underlying nucleotide composition. However, the coverage percentage is roughly the same as the underlying nucleotide composition, indicating that coverage is much less sensitive to MNase bias.

2.9 Comparison of inference results using different settings to pre-process MNase-seq data. MNase-seq data were pre-processed using the seven settings described in the main text and Table 2.3. For each setting, TF transition weights were fitted using the same inference method as described in the main text. 50% posterior credible intervals of each fitted TF transition weight in log scale are shown here. In all these settings, the 50% posterior credible intervals overlap significantly for the TFs shown here. We therefore conclude that the specific parameters we used to pre-process the MNase-seq data do not have a significant impact on inference results. Results for all TFs are shown in Appendix Figure A.3, in which we can observe similar conclusions.

3.1 Mean and variance of DNase-seq counts at each position along the nucleosome are proportional to each other (after inverse hyperbolic sine transformation). Each point corresponds to the mean and variance of one of the 147 positions in the nucleosome window.

3.2 DNase I cleavage profiles (average transformed DNase-seq counts) at 2,000 random genomic windows are essentially flat, modulo random sampling noise.
3.3 (A) Strand-specific cleavage profile of DNase I along the nucleosome, computed by averaging DNase-seq counts (transformed by the inverse hyperbolic sine function) within the 2,000 most strongly positioned nucleosomes in the yeast genome. (B) Crystal structure of a nucleosome shown from two angles; which strand faces outward as the minor groove becomes accessible differs on opposite sides of the nucleosome dyad. The two positions labeled with arrows have the same relative distance to the dyad. Left: the blue strand faces outwards and is more exposed to digestion at this position while the red strand faces inwards and is less exposed. Right: the red strand is now the one that faces outwards while the blue strand faces inwards. In other words, on opposite sides of the nucleosome dyad, each strand is exposed differently as the minor groove becomes accessible (more exposed upstream of the dyad and less exposed downstream of the dyad). (C) Posterior density of the period of oscillation, as determined by Bayesian harmonic regression. The most probable period \textit{a posteriori} for each of the two different strands is around 10.3 bp. 71

3.4 DNase I cleavage profiles along the nucleosome for the Watson and Crick strands are almost exactly mirror symmetric with each other. This figure is similar to Figure 3.3A, but with the Crick strand profile flipped around the line \(x = 0\). 72

3.5 Strand-specific cleavage profile of DNase I along the nucleosome, computed by averaging DNase-seq counts in the Hesselberth data (transformed by the inverse hyperbolic sine function) within the 2,000 most strongly positioned nucleosomes in the yeast genome. The same average is calculated on both \textit{in vivo} (green and orange) and \textit{in vitro} (purple and red) data. The distinct DNase I cleavage profile observed in the \textit{in vivo} data is completely absent in the \textit{in vitro} data. We observe that the DNase I digestion pattern near the dyad is slightly noisier in the Hesselberth data (both \textit{in vivo} and \textit{in vitro}) than we observe in our data, perhaps due to slight differences in the protocols. 74

3.6 Posterior density of the period of oscillation in the Hesselberth data, as determined by Bayesian harmonic regression. The most probable period \textit{a posteriori} for each of the two different strands is around 10.4 bp. 75

3.7 Classification ROC for 10-fold cross-validation on both our data (green) and the data of Hesselberth and colleagues (orange). All test cases from the ten folds were combined to draw an overall ROC for each dataset. The areas under the two ROCs are computed and presented as a bar chart (inset). 76
3.8 Example genomic region in which nucleosome positions are mapped using our moving window nucleosome scoring approach. (A) Raw DNase-seq counts in this region. Note that DNase-seq analysis has traditionally focused only on finding and exploring DHS regions, such as the one that corresponds to the strong peak of signal just to the left of coordinate 61,000 (promoter of ESL1). (B) Transforming the raw DNase-seq counts using an inverse hyperbolic sine function allows clearer (but still weak) patterns to be seen in nucleosome-associated DNA. (C) Smoothed moving window nucleosome score on this region (green), in comparison with the NCP score-to-noise ratios from Brogaard et al. (2012) (orange).

3.9 (A) Distribution of center-to-center distances between our DNase-seq–based nucleosome map and the Brogaard reference map (green), and between the consensus MNase-seq–based nucleosome map and the Brogaard reference map (blue). (B) The number of true positives (TP), false negatives (FN), and false positives (FP) of our DNase-seq–based nucleosome map and the consensus MNase-seq–based nucleosome map. The numbers of TP, FN, and FP are defined using the Brogaard nucleosome map as a reference (gold standard).

3.10 Similar to Figure 3.9B. Using the nucleosome map from Brogaard et al. (2012) as a reference, we calculate the number of true positives (TP), false negatives (FN), and false positives (FP) of our DNase-seq–based map, the consensus MNase-seq–based map, and each of the four individual MNase-seq–based maps. The consensus MNase-seq–based map is compiled by Jiang and Pugh (2009) based on the four individual MNase-seq–based maps. Set 1 is from Mavrich et al. (2008). Set 2 is from Field et al. (2008). Set 3 is from Jiang and Pugh (2009). Set 4 is from Shivaswamy et al. (2008).

3.11 Similar to Figure 3.9A. Using the nucleosome map from Brogaard et al. (2012) as a reference, we calculate the distribution of center-to-center distances between our DNase-seq–based nucleosome map and the Brogaard reference map (green), and between the consensus MNase-seq–based nucleosome map and the Brogaard reference map (blue), as well as between each of MNase-seq–based nucleosome maps and the Brogaard reference map (orange, red, black, and purple).
3.12 Median nucleosome score around all mRNA transcription start sites (A), around origin of replication ACS sites (B), and around binding sites of Abf1 (C) and Reb1 (D) across the yeast genome. TSS coordinates are taken from Rhee and Pugh (2012), ACS coordinates are taken from Eaton et al. (2010), and TF binding site coordinates are taken from MacIsaac et al. (2006). The bottom of each panel shows the average NCP score-to-noise ratios from Brogaard et al. (2012) at corresponding positions.

3.13 Additional example regions showing nucleosome scores around TSSs (top panels) and ACSs (bottom panels). The composite score pattern shown in Figure 3.12 is also observed at individual genomic loci.

3.14 Additional example regions showing nucleosome scores around Abf1 (top panels) and Reb1 (bottom panels) binding sites. The composite score pattern shown in Figure 3.12 is also observed at individual genomic loci.

3.15 Classification performance (AUROC) as a function of subsample percentage. Sequencing depths (reads/nt) corresponding to each subsample percentage are displayed next to the curve.

3.16 Comparison of DNase-seq–identified nucleosomes and MNase-seq–identified nucleosomes in human Chromosome 12 (intronic region of CACNA1C). Nucleosome centers are identified by the same greedy algorithm on our nucleosome score curve and the MNase-seq fragment middle point counts, respectively. Nucleosomes identified from the two types of data overlap significantly, especially considering the large noise in the MNase-seq data. The middle panel also shows the DARNs from Winter et al. (2013), which likely represent portions of phased nucleosomes.

3.17 The distribution of DARNs lengths, taken from Winter et al. (2013). Note that the vast majority of DARNs are shorter than 100 bp.

3.18 (A) Distribution of center-to-center distances between our DNase-seq–based nucleosome map and the MNase-seq–based map of Gaffney et al. (2012). (B) The number of nucleosomes identified in both DNase-seq and MNase-seq maps, as well as the number of nucleosomes that only appear in one of the maps.

3.19 Median nucleosome score around human TSSs (A) and CTCF binding sites (B).
3.20 DNase I digestion profiles on human nucleosomal DNA (black) and yeast nucleosomal DNA (red). The digestion profiles shown in the figure are an average of the forward and reverse oscillation profiles.

3.21 Average oscillations around motif matches of Abf1 and Mcm1 (yeast), and CTCF and GABPA (human) (“motif oscillation”, black). “Random oscillations” (blue) are average oscillations around randomly chosen genomic sites. One hundred random oscillations are calculated and shown for each TF. Red dashed lines indicate the boundaries and centers of the TF motifs. For many TFs, including those shown here, their “motif oscillations” have significantly higher amplitude than the corresponding “random oscillations”, indicating TF motif matches locate preferentially with respect to the rotational phasing of the DNA along the nucleosome.

3.22 Most TF motifs are either major- or minor-groove-associated. Twenty-one yeast TFs are shown as blue dots. Five human TFs are shown as red triangles. For each TF, the distance between its motif center to the nearest peak of its motif oscillation is calculated and converted to a “oscillation phase” within $[-\frac{\pi}{2}, \frac{\pi}{2}]$: if the motif center is located at the peak of the composite oscillation and thus tends to be centered on an exposed minor groove, its phase is $-\frac{\pi}{2}$; conversely, if the motif center is located at a trough and thus centered on a major groove, its phase is $\frac{\pi}{2}$. The y-axis shows the ratio between each TF’s motif oscillation amplitude and the amplitude of random oscillations. This is a measure of the significance of the preferential localization. Most TFs have a phase $\geq \frac{\pi}{4}$ or $\leq -\frac{\pi}{4}$, and a ratio $> 3$, indicating they are significantly associated with the major or minor groove, respectively.

3.23 We randomly generated five GC (or AT) rich motifs, each with a length of 13 bp (the length of the Abf1 motif). We pooled together all motif matches for these five motifs, filtering out overlapping matches, and then randomly selected 2,600 motif matches from this pool (2,600 being the number of motif matches for Abf1). We calculated and plotted the average oscillation around these motif matches as one curve above. We then repeated this entire process 50 times, resulting in 50 curves for the GC (green) and AT (orange) rich motifs. The anti-correlated curves are consistent with nucleosome sequence preferences for GC and AT nucleotides, which are known to oscillate out of phase with one another.
3.24 Performance (as measured by AUROC) of different classification approaches on both our data and Hesselberth data. Classifying nucleosomal vs. non-nucleosomal locations on the basis of a simple likelihood ratio has the worst performance. Bayes factor approaches that integrate out both the level parameter $a$ and the curvature parameter $b$ perform similarly to approaches that only integrate out $a$. Approaches that include the oscillation pattern have better performance than approaches without the oscillation pattern. .................................................. 96

3.25 False negative (FN), true positive (TP), and false positive (FP) calculated as in Figure 3.9B using different greedy algorithm parameters. Performance of the consensus MNase-seq–based nucleosome map is shown in the last column, for reference. .................................................. 97

4.1 Possible states in the thermodynamic equilibrium of a system with a three-nucleotide sequence and two DBFs. .................................................. 103

4.2 A schematic description of the multivariate SSM state path. Square box denotes a state in the state path. States that belong to the same DBF are grouped using dashed rectangles. Circles denote the particular data emitted by a state. Arrows denote either the transition between states or the emission of data at each state. (A) COMPETE only models the sequence data (only nucleotide is emitted at each state). (B) In our multivariate SSM, we will add additional data type, such as DNase-seq counts. .................................................. 106

4.3 (A) Comparison between estimated NB distribution and empirical counts distribution at Phd1 binding sites, using quantile-quantile plot. (B) Comparison between estimated NB distribution and empirical counts distribution at digital footprints. .................................................. 109

4.4 Average DNase-seq data at four TF binding sites (MacIsaac et al., 2006). Red dashed lines indicate the boundary of each TF binding site. The number of binding sites, as well as the estimated NB parameters using DNase-seq counts within the binding site, is shown as titles of each plot. .................................................. 110

4.5 Simulation with nucleosome as the only DBF. True binding configuration is shown at the bottom of each panel. Nucleosomes are shown as gray bars. (A) Simulated DNase-seq counts. (B) Recovered binding configuration through training model with DNA sequence only. (C) Recovered binding configuration through training model with both DNA sequence and DNase-seq data. .................................................. 124
4.6 Simulation with DBFs nucleosome, Abf1 and Cbf1. True binding configuration is shown at the bottom of each panel. Nucleosomes are shown as gray bars. Abf1 and Cbf1 are shown as red and blue dots, respectively. (A) Simulated DNase-seq counts. (B) Recovered binding configuration through training model with DNA sequence only. (C) Recovered binding configuration through training model with both DNA sequence and DNase-seq data. 125

4.7 Learning protein-DNA interaction landscape in region ChrXIII:659,000–660,000 (sacCer2) using DNA sequence only (A) and using both DNA sequence and DNase-seq data (C). DNase-seq data is shown in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel. 128

4.8 Learning protein-DNA interaction landscape in region ChrI:60,500–64,000 (sacCer2) using DNA sequence only (A) and using both DNA sequence and DNase-seq data (C). DNase-seq data is shown in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel. 129

4.9 MNase-seq large fragment middle point counts in nucleosomal positions (A) and TF binding sites (B and C). To ensure that the signal we observe is not caused by MNase bias, we also showed the in vitro MNase digestion data on naked DNA from Deniz et al. (2011). Nucleosome positions are from Brogaard et al. (2012). TF binding sites are from MacIsaac et al. (2006). 131

4.10 Learning protein-DNA interaction in region ChrXIII:659,000–660,000 (sacCer2) using DNA + DNase-seq counts (A) and using DNA + DNase-seq counts + MNase-seq large fragments middle point counts (C). MNase-seq fragments are shown in (B). MNase-seq large fragments middle point counts are also shown as a blue curve in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel. 132
4.11 Learning protein-DNA interaction in region ChrI:60,500–64,000 (sacCer2) using DNA + DNase-seq counts (A) and using DNA + DNase-seq counts + MNase-seq large fragments middle point counts (C). MNase-seq fragments are shown in (B). MNase-seq large fragments middle point counts are also shown as a blue curve in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel.

A.1 Predicted binding profiles versus MNase-seq binding profiles. Shown here are the same six promoter regions shown in Figure 2.5. For each promoter, we show the comparison between processed large MNase-seq fragment coverage and predicted nucleosome profile (top left) and the comparison between processed small MNase-seq fragment coverage and predicted composite TF binding profile (top right). We also show these comparisons as scatter plots (bottom left and right, respectively). Since both the data and our predictions describe how a binding profile changes along the genome, the scatter plots should not be interpreted as traditional scatter plots (of independently sampled points). Rather, they are more appropriately interpreted as something akin to phase space plots, swept out as one traces along the genome.

A.2 Comparison of in vivo and in vitro MNase-seq coverage. We also examined the severity of MNase bias by comparing the in vivo MNase-seq data we used with in vitro MNase-seq data on naked DNA from Deniz et al. (2011). Shown here are coverage plots for representative promoter regions from the 81 promoter regions used in this study. Considering the difference in sequencing depth, we do observe several overlapping coverage peaks between in vivo and in vitro data, such as those in YER055C, YDR023W, and YMR229C promoters. However, the majority of the in vivo signals cannot be explained by sequence bias. This is in line with our observation in Figure 2.8.

A.3 Similar to Figure 2.9. Comparison of inference results using different settings to pre-process MNase-seq data. MNase-seq data were pre-processed using the seven settings described in Section 2.2.5. 50% posterior credible intervals of each fitted TF transition weight in log scale are shown here.
List of Abbreviations and Symbols

Abbreviations

DNase I deoxyribonuclease I
DNase-seq DNase I digestion followed by high-throughput sequencing
EDA exploratory data analysis
CRM cis-regulatory modules
MNase micrococcal nuclease
MNase-seq MNase digestion followed by high-throughput sequencing
DHS DNase hypersensitive site(s)
DBN dynamic Bayesian network
HMM hidden Markov model
ChIP-seq chromatin immunoprecipitation followed by high-throughput sequencing
NCP nucleosome center position
TSS transcription start site
ENCODE Encyclopedia of DNA Elements
LOWESS locally weighted scatterplot smoothing
DNA deoxyribonucleic acid
TF transcription factor
MLE maximum likelihood estimate
ROC receiver operating characteristic
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFR</td>
<td>nucleosome free region</td>
</tr>
<tr>
<td>ACS</td>
<td>ARS consensus sequence</td>
</tr>
<tr>
<td>ORC</td>
<td>origin replication complex</td>
</tr>
<tr>
<td>tRNA</td>
<td>transfer RNA</td>
</tr>
<tr>
<td>mRNA</td>
<td>messenger RNA</td>
</tr>
<tr>
<td>TBP</td>
<td>TATA-box binding protein</td>
</tr>
<tr>
<td>GTF</td>
<td>general transcription factor</td>
</tr>
<tr>
<td>DBF</td>
<td>DNA binding factor</td>
</tr>
<tr>
<td>Pol II</td>
<td>RNA polymerase II</td>
</tr>
<tr>
<td>PWM</td>
<td>position weight matrix</td>
</tr>
<tr>
<td>SSM</td>
<td>state-space model</td>
</tr>
<tr>
<td>CTD</td>
<td>C-terminal domain</td>
</tr>
<tr>
<td>PMF</td>
<td>probability mass function</td>
</tr>
<tr>
<td>EM</td>
<td>expectation maximization</td>
</tr>
</tbody>
</table>
Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Alex Hartemink, for his guidance during my Ph.D. study. Alex has always been supportive, not only academically but also personally. He always listens to my ideas and provides insightful feedback. His clear thinking and writing influence the way I approach new problems and express my ideas in writing. My stay at Duke would be less fruitful without his guidance.

I’d also like to thank Dr. Ed Iversen. Ed was my advisor for my master’s degree in statistics. His patience and detailed instructions helped me greatly in understanding statistical modeling under uncertainty.

I want to thank my thesis committees, Dr. Raluca Gordân, Dr. Greg Crawford, and Dr. Sayan Mukherjee, for their helpful discussions and feedbacks on my progress. I also want to thank professors in the Duke Statistics Department for their engaging lectures in different areas of statistics. Especially, I’d like to thank Dr. Mike West, Dr. Merlise Clyde, Dr. Li Ma, and Dr. Surya Tokdar.

Thanks also go to current and past members of the Hartemink group. I had many academic and non-academic discussions with Dr. Xin Guo, Dr. Michael Mayhew, Yezhou Huang, Dr. Jason Belsky, and Kaixuan Luo. I appreciate their company during my Ph.D. study. I also want to thank my fellow students in the Duke Computational Biology and Bioinformatics Program, especially Zack Scholl, Dr. Jason Belsky, David Winski, Kaixuan Luo, Dr. Yu Jiang, Dr. Xianrui Cheng, and Dr. Peng Dong.
They are the first friends I met when I came to the United States. My experience at Duke would be much less enjoyable without them.

I’d like to thank my past roommate Wenjian Mo for his emotional support during difficult times, and my current roommate Xiangyu Wang for all the fun time we had together.

I am also grateful for the generous financial support from the Duke Computer Science Department and the Duke Computational Biology and Bioinformatics Program during my Ph.D. study.

Finally, I want to express my deepest gratitude to my parents. I would not have come this far without their unconditional love.
Introduction

Gene expression, the process of expressing genetic information encoded in DNA sequences in the form of functional products (often proteins), consists of two major steps according to the central dogma: transcription, the way by which DNA-encoded information is copied and re-encoded in messenger RNAs (mRNA); and translation, the synthesis of polypeptides (usually as building blocks of more complex protein macromolecules). Through various mechanisms, each step is tightly regulated to ensure proper spatial and temporal expression of genetic information.

This dissertation primarily concerns the regulation of transcription. Specifically, works presented here develop computational methods to study the interactions between regulatory proteins (also called DNA binding factors, or DBFs) and the genome to be transcribed. Related questions, such as what kind of factor is interacting with the genome at a given location, where and how likely these interactions happen on the genome-wide scale, are essential in understanding transcriptional regulation. They have been studied extensively in the past century, both experimentally and computationally. Building on previous work, we seek to establish a genome-wide, holistic view of the protein-DNA interaction landscape — the probabilistic view of
any given DBF interacting at any given location in the genome. A central theme of this dissertation is that we attempt to achieve this goal through exploratory data analysis, statistical model learning and validation (especially the use of Bayesian approaches), along with the integration of many sources of data.

Genome-wide protein-DNA interactions can be mapped by different experimental methods, including chromatin immunoprecipitation- (ChIP-) based methods, such as ChIP-chip (ChIP followed by microarray assay) and ChIP-seq (ChIP followed by high-throughput sequencing), and nuclease-digestion–based methods, such as deoxyribonuclease-seq (DNase-seq) or micrococcal-nuclease–seq (MNase-seq). In addition, in vitro methods, such as PBM (protein binding microarrays) and MITOMI (mechanically induced trapping of molecular interactions), can be used to study sequence specificities of DBFs in an in vitro environment. Although large volumes of data have been accumulated, a specific data source often only reveals one aspect of the protein-DNA interaction landscape: ChIP-based methods can reveal protein-DNA interaction sites with high-resolution but only for a single antibody-targeted factor under a specific condition; nuclease digestion datasets provide binding information for all proteins, but do not reveal the identities of the proteins; and in vitro experiments only look at the binding property of one isolated protein in an in vitro environment. Correspondingly, the computational methods that are based on one data source are only able to provide inferences for the intended, single aspect of protein-DNA interaction landscape. A mere accumulation of data does not necessarily provide additional insights into the protein-DNA interaction landscape. We need sophisticated computational frameworks that can effectively integrate all these data together to generate deeper insights into transcriptional regulation.

To this end, in Chapter 2, we built a general framework based on a biophysical model called COMPETE (Wasson and Hartemink, 2009) that can integrate different data sources. COMPETE treats interactions between proteins and DNA as probabilis-
tic events and can model the binding of different DBFs on the genome simultaneously, including nucleosomes. However, a major limitation of COMPETE is that it is a purely theoretical model of binding, based on thermodynamic first principles, but not guided by data regarding \textit{in vivo} binding events. We therefore integrate COMPETE into a larger framework that can infer the actual thermodynamic interactions between DBFs and the genome using information from different experimental observations. We demonstrate the use of this framework by integrating paired-end micrococcal nuclease sequencing (MNase-seq) data, which reveals information about the binding occupancy of both nucleosomes and smaller (subnucleosomal) factors. Our framework also integrates protein binding specificity information from PBM data and produces a more accurate and realistic protein-DNA interaction landscape than COMPETE alone. The cross-validated performance of our framework is significantly higher when compared to several baselines. Our framework is flexible and can easily incorporate other data sources as well. We further demonstrate its flexibility by incorporating experimentally measured protein concentrations to guide the inference of the model. Our work represents a general modeling framework for integrating multiple sources of information to produce a more precise view of the interaction landscape undergirding transcriptional regulation.

In an effort to incorporate additional data sources into a computational framework, we devote another chapter to carefully study a specific kind of genomic data: the DNase I digestion followed by sequencing (DNase-seq) datasets. DNase I was initially used to probe the structure of an individual nucleosome before its exact details were known. However, in recent years, DNase I has only been widely used to identify DNase hypersensitive sites (DHS), which tend to be open chromatin regulatory regions devoid of nucleosomes like promoters, silencers, and enhancers. In Chapter 3, we show that DNase-seq datasets also contain substantial information about nucleosome translational positioning, and that existing DNase-seq data can be used
to infer nucleosome positions with high accuracy. To achieve this, we develop a Bayes-factor-based nucleosome scoring method that can model a distinct, quadratic and oscillatory pattern created by DNase I when it is cutting nucleosome-associated DNA. We show that this method can achieve high sensitivity and specificity in distinguishing nucleosomal and non-nucleosomal genomic regions. Applying this method, we generate the first genome-wide nucleosome maps based on DNase-seq data for both yeast and human. We show that the resulting maps are highly concordant with previous maps. Canonical nucleosome positioning properties, such as highly phased nucleosome arrays around transcription start sites (TSSs) and replication origins, are clearly reflected in our DNase-based maps. The spatial relationships we observe between nucleosomes and bound transcription factors (TFs) (such as Abf1 and Reb1 in yeast and CTCF in human) are also in strong accordance with previous reports. Additional insights, such as the nucleosome rotational positioning around TF motif matches can also be derived from our methods. The work in this chapter provides a modeling basis for integrating DNase-seq data into a larger framework for inferring the protein-DNA interaction landscape.

Building on the work summarized above, we finally develop a more principled general-purpose statistical model to efficiently learn the protein-DNA interaction landscape. This model is a multivariate state-space model (similar to a multivariate hidden Markov model). In Chapter 4, we present the detailed model formulation and provide efficient expectation-maximization (EM) algorithms for model fitting. We first test the model performance using simulated datasets. We also present results from modeling real genomic data (both DNase-seq and MNase-seq data). Through incrementally adding data into the model, we will show that different datasets can provide complementary information and that the most accurate inference comes from modeling all available datasets. We also release a flexible software tool for the research community to explore modeling possibilities with different kinds of data.
Before detailed presentation on our computational efforts, we first introduce the biological backgrounds that motivated our work. We will also briefly review current experimental and computational techniques that are commonly used in studying protein-DNA interactions. We will then end this chapter with an outline of this dissertation.

1.1 Eukaryotic gene transcription

Eukaryotic genes are highly organized sequential strings of four bases: adenine (A), cytosine (C), guanine (G), and thymine (T). Typical structures of class II genes (Figure 1.1), which are genes transcribed by RNA polymerase II (Pol II) and are the most numerous eukaryotic genes, include a gene body (that encodes the actual gene product), a core promoter region upstream of the gene body and many regulatory sites called cis-regulatory modules (CRMs), such as TF binding sites. Metazoan genes could be very complex: Gene bodies are split into multiple introns and exons and CRMs may include enhancers that are millions of base pair (bp) away from the gene body. However, for simpler model organisms, such as \textit{S. cerevisiae} (budding yeast, the main focus of this dissertation), the gene organization is more compact: Both gene body split and distal enhancers are rare.

Eukaryotic gene transcription is usually a coordination between CRMs and many trans-acting factors, including general and specific TFs, TATA-box binding protein (TBP) and RNA polymerases. The transcription process can be roughly separated into three stages: \textit{initiation}, \textit{elongation} and \textit{termination}, each of which is tightly regulated. Different classes of RNA polymerases (Pol I, Pol II and Pol III) have different mechanisms involved to regulate those stages. We will be focusing on the regulation related to Pol II, which transcribes the majority of protein-coding genes.
1.1.1 Transcription initiation

The core promoter region is the primary location where transcription initiation starts (Smale and Kadonaga, 2003). Many different sequence elements appear in the core promoter region (Figure 1.1), such as BRE (TFIIB recognition element) and DPE (downstream core promoter element, recognized by certain subunits of TFIID), but perhaps the most important element is the TATA box (Kim et al., 1993; Smale and Kadonaga, 2003; Juven-Gershon et al., 2008). Although many eukaryotic genes were identified as “TATA-less” (no TATA box was found) (Pugh and Tjian, 1991; Basehoar et al., 2004), there were “TATA-like” elements (elements with up to two mismatches of the TATA box TATAWAWR consensus) identified to be associated with those genes recently (Rhee and Pugh, 2012).

Both “TATA” and “TATA-less” genes require the arrival of TBP for the assembly of the pre-initiation complex (PIC), a crucial event of transcription initiation. There are multiple pathways that lead to the assembly of PIC (Figure 1.2), but two common pathways in *S. cerevisiae* are the “TFIID” pathway (utilized by “TATA-less” genes) and the “SAGA” pathway (utilized by “TATA” genes). We will focus on these pathways in this section.

![Typical organization of eukaryotic genes. Figure adapted from Courey (2009).](image)
The canonical TFIID pathway is usually described as a step-wise process (as reviewed by Thomas and Chiang (2006)). It starts from the binding of TFIID, a subunit of which is TBP, to the core promoter region. Then two other general transcription factors (GTFs), TFIIA and TFIIB, are recruited to help stabilize the binding of TFIID. The TFIIF-Pol II complex is then recruited. The last two GTFs, TFIIE and TFIIH, are subsequently recruited to form the PIC. It is also possible that a functional complex called the “RNA polymerase holoenzyme” (a complex includes Pol II, TFIIE, TFIIH, and others) is first assembled. The holoenzyme is then recruited to the core promoter region as one entity (Lemon and Tjian, 2000; Lee and Young, 2000; Thomas and Chiang, 2006).

The SAGA pathway of PIC assembly is widely used by the “TATA” genes in *S. cerevisiae*. Unlike in the TFIID pathway, TBP is directed to the core promoter region by the SAGA complex (Spt-Ada-Gcn5-Acetyltransferase) (Lemon and Tjian,
2000; Bhaumik and Green, 2002). In addition, since the gene promoters contain a TATA-box, TBP is directly binding at the TATA-box. It has been reported that SAGA pathway is often utilized by stress response genes while the TFIID pathway is primarily utilized by housekeeping genes (Basehoar et al., 2004; Huisinga and Pugh, 2004).

A common factor that is recruited to the PIC in both pathways is the TFIIH. The presence of TFIIH is important because its kinase activity is required to phosphorylate the C-terminal domain (CTD) of Pol II before Pol II can start transcription elongation (Courey, 2009).

### 1.1.2 Transcription elongation

After the PIC is assembled and Pol II is phosphorylated, transcription elongation can start. The elongation may seem like a dull process of synthesizing mRNAs. However, this process, as with other transcription processes, is highly regulated (for a recent review, see Jonkers and Lis (2015)). One important regulatory event in metazoans is the so-called Pol II pausing: After transcribing to 30-60 bp downstream of TSS, Pol II stops and accumulates near the promoter region (Adelman and Lis, 2012; Kwak and Lis, 2013). Different genome-wide techniques, such as ChIP methods (Muse et al., 2007; Zeitlinger et al., 2007) and global run-on sequencing (GRO-seq) methods (Core et al., 2008), all revealed that Pol II pausing is a genome-wide feature.

Different factors, such as negative elongation factor (NELF), DRB-sensitivity-inducing factor (DSIF) as well as the +1 nucleosome may contribute to the pausing of Pol II that has left the PIC (Adelman and Lis, 2012; Kwak and Lis, 2013). Another key factor, the positive transcription elongation factor-b (P-TEFb), has been shown to be required for the release of paused Pol II. P-TEFb phosphorylated Pol II, as well as NELF and DSIF, to convert paused Pol II into a productive elongation state (Peterlin and Price, 2006; Lis et al., 2000).
There is, however, little evidence observed in *S. cerevisiae* (Adelman and Lis, 2012) for the pause of Pol II at promoter-proximal regions. But there are studies (for example, Churchman and Weissman (2011)) that showed Pol II pausing could occur throughout the gene body in *S. cerevisiae* and particularly at the first four nucleosomes. This Pol II pausing, although can regulate transcription, appears to be mainly induced by nucleosomes (compared to the different factors utilized in regulating Pol II pausing in metazoans).

1.1.3 Transcription termination

After a period of productive elongation, the termination of RNA polymerases is crucial for several reasons (Richard and Manley, 2009): (1) The transcription of downstream genes should not be interfered by the transcription of upstream genes; (2) The Pol II involved should be released and returned to the pool of available Pol II; and (3) The formation of antisense transcripts should be prevented by releasing the Pol II (so that it will not backtrack).

Many different factors can affect the termination of Pol II transcription. It has been shown that termination is associated with the cleavage of the 3’ end of the nascent mRNA transcripts and the phosphorylation of CTD of Pol II. The cleavage and polyadenylation specificity factor (CPSF), a protein complex that has homologs in both human and *S. cerevisiae*, can recognize a canonical signal called the poly(A) site/signal (a conserved AAUAAA site). Termination is also tightly coupled with the phosphorylation of the CTD of Pol II. CPSF may bind at the phosphorylated CTD of Pol II to induce the pausing of Pol II (Gromak et al., 2006; Glover-Cutter et al., 2008), which further facilitates the transcription termination. Paused Pol II gives time for other factors, such as RNA-trafficking protein 1 (Rat1) and its co-factors, to eventually release Pol II and terminate the transcription (Kuehner et al., 2011).

Besides the poly(A) signal dependent termination mechanism, another mecha-
nism called the Sen1-dependent mechanism is found to be involved in the transcription termination of noncoding RNAs, such as snRNAs and snoRNAs (Steinmetz et al., 2006). This mechanism depends on the Sen1 helicase to unwind the DNA-RNA complex in order to terminate the transcription.

1.2 Transcription factors and their roles in transcriptional regulation

The PIC, together with Pol II and related regulatory factors, is sufficient for the basal transcription to happen. However, for greater regulatory control, a host of proteins called specific transcription factors (as oppose to the general transcription factors mentioned above) work in combination with each other to further activate or repress the transcription of certain genes. The interactions of these TFs with the genome (particularly with the promoter regions) are of main interest in this dissertation. We will give a brief review of the general properties of these transcription factors, including their classification, as well as their role in regulating transcription.

1.2.1 General properties of transcription factors

Transcription factors belong to a class of proteins that can bind to specific sequences on the DNA and regulate (either activate or repress) the transcription activity of certain genes. Usually the gene a TF regulates is downstream of, but close to the binding location of the TF in *S. cerevisiae*. However, in higher organisms, this gene could be a long distance away from the TF binding site in the distance of linear DNA (but not necessarily in spatial distance because DNA have a 3D structure that can bring a TF close to the gene it regulates (Duan et al., 2010)).

*DNA binding domain of TFs*  It is apparent that TFs should be able to bind DNA at specific locations. Such property is usually called the sequence preference of a TF and represented by a position weight matrix (PWM).
The DNA binding domain (DBD) of a TF is in direct contact with DNA and therefore is of vital importance to a TF. Because of this, TFs are usually classified by their DBDs. Several major classes of DBDs have been studied to date include:

1. The homeodomain. The homeodomain TFs were identified during genetic studies in *Drosophila melanogaster* as products of homeotic genes (Gehring et al., 1994). The mutation of those genes leads to disruptive development of *Drosophila melanogaster* segmentation. The DNA binding motif within this domain is the helix-turn-helix (HTH) motif. Therefore, this domain sometimes is directly referred as the HTH domain (Berman et al., 2008). Common TFs in this family include Yox1 and Pho2 in *S. cerevisiae*. HTH motif is also found in many prokaryotes and bacteriophage, such as the bacterial Trp repressor (Otwinowski et al., 1988).

2. The zinc finger domain. Zinc finger domain contains a variety of subclasses. A common feature of this domain is that zinc ion(s) is utilized to form the structure of this domain (Figure 1.3). Zinc finger domains contact DNA with two fingers (they are usually homodimers), each on a major groove and the protein spans the minor groove between the two major grooves (MacPherson et al., 2006). Common TFs in this family include Gal4, Leu3, and Abf1 in *S. cerevisiae*.

3. Basic leucine-zipper (bZIP) and basic helix-loop-helix (bHLH) domains. bZIP and bHLH domains are very similar domains (Figure 1.4). They are both basic DNA binding domains. They both have coiled-coil C-terminal domains and can form homo- or hetero-dimers. Unlike bZIP domain, whose C-terminal and N-terminal are the same helices, bHLH domain’s C-terminal and N-terminal are separate structures connected by a loop (Berman et al., 2008). bZIP TFs
TF synthesis and activity regulation  Since TFs are key components in ensuring the transcription of genes are in correct spatial- and temporal-order, their own expressions are also tightly regulated so they become functioning at the right location and time. For example, the FoxO family of transcription factors (FoxO1, FoxO3 and...
FoxO6) has been shown to exhibit differential expression patterns in different parts of the mouse brain (Hoekman et al., 2006). Perhaps the most notable examples are certain transcription factors called “master regulators”: Factors that expressed at a specific cell type and determines the identify of that cell type (Chan and Kyba, 2013). One such example, myoblast determination protein 1 (MyoD1), determines the development of muscle cells and is able to convert embryonic fibroblasts to myoblasts (Davis et al., 1987). The expression of MyoD1 is therefore regulated to ensure proper development of muscle tissue (Asakura et al., 1995). In *S. cerevisiae*, a single-cell organism, spatial regulation is less relevant. However, different mating types of *S. cerevisiae*, including a and α mating types, have different regulatory components expressed from their mating type locus MAT (Herskowitz, 1989).

Besides the expressions of TFs, their activities are also key elements in regulation. One of the most studied examples is the activation and repression of the activities of SBF transcription factor complex (Iyer et al., 2001). SBF is a complex of Swi4 and Swi6 and is activated by Cln/Cdc28 in late G1 phase to initiated transcription of its regulated genes (Koch et al., 1996). Ndd1 (together with Fkh2) is another example that is activated through phosphorylation during late cell cycle (Wittenberg and Reed, 2005). In general, phosphorylation and dephosphorylation are common modifications utilized to activate and de-activate TFs (Bohmann, 1990).

### 1.2.2 Transcriptional regulation by transcription factors

TFs can either activate or repress the transcription of certain genes. In those cases, the TF is usually called an activator or a repressor, respectively. Sometimes the same TF can be an activator for certain genes while acts as a repressor for some other genes.
Transcriptional activation  To activate a gene transcription, TFs can employ different mechanisms:

1. TFs can directly interact with the basal transcription machinery, including PIC and Pol II, to facilitate the assembly of the machinery or to enhance the activity of the machinery after its assembly. One such example is the yeast TF Gal4. Gal4 can interact with TFIID to enhance its ability to bind DNA (Horikoshi et al., 1988);

2. TFs can interact with other regulatory proteins, such as the TAFs and the mediator (Green, 2000; Kornberg, 2005). Those proteins in turn will enhance the ability of the basal transcription machinery;

3. TFs can also alter the structure of the chromatin, by directly inhibiting the formation of nucleosomes or displace existing nucleosomes (Svaren and Hörz, 1997). These changes can open up spaces for the binding of other activators or the formation of basal transcription machinery.

Transcriptional repression  Similarly, a variety of mechanisms can be used by a TF to repress the transcription of certain genes.

1. A repressor can function through interacting with an activator and therefore indirectly repress the gene transcription. It can bind to the DNA sequence that the activator binds or form a heterodimer with the activator. Both cases prevent the binding of the activator to its recognition sequence;

2. Repressors can also directly interact with the basal transcription machinery to inhibit its ability to initiate transcription. The Drosophila TF eve is one such example (Austin and Biggin, 1995);
3. In addition, repressors can also modify the chromatin to repress gene transcription, which is opposite to the effect of activators have (Courey and Jia, 2001).

1.2.3 The continuous, competitive yet cooperative nature of transcription factors binding

Up to this point, we have been discussing the “binding” of TFs without actually quantifying this event. Many studies treat TF binding as binary events amenable to classification: either a protein binds at a particular site on the DNA sequence or it does not. For example, ChIP studies (Harbison et al., 2004; MacIsaac et al., 2006; Foat et al., 2006; Tanay, 2006) often classify peak positions as “binding” sites where proteins “bind” while other positions as “non-protein-binding” sites. However, biological intuition and lots of evidence all suggest that protein binding is a probabilistic event with probabilities range from 0 to 1 continuously (Biggin, 2011). To be more explicit, this statement can be understood from a frequentist’s point of view: Observing a specific genomic site for a given period of time (within which cellular state does not change), the fraction of time that the genomic site is occupied by a given protein is the probability that the given protein binds at the given genomic location. An alternative, equivalent understanding is, observing the same genomic location of a number of cells within a homogeneous cell population, the fraction of locations that are bound by a given protein is the probability that the given protein binds at the given genomic location. TF binding probability sometimes is also called TF binding occupancy. While occupancy, unlike probability, is mostly a measure not bounded between zero and one, we use it with roughly the same physical meaning as binding probability in this dissertation.

Many lines of evidence can support this probabilistic binding view. For example, even though ignored by many studies, ChIP experiments do provide quantitative
information of different levels of binding occupancy for a protein. Li et al. (2008) used control experiments and showed that microarray intensity values correlated with protein occupancy levels. Later the same group showed that 21 Drosophila TFs bind at different target sequences with different occupancy levels and that the high occupancy levels usually meant biologically functional binding while low occupancy binding was usually not functional (MacArthur et al., 2009). Quantitative information from ChIP-seq experiments also proves to be correlated with the expression of the targeted gene of the TF, indicating the effect of the TF binding (and therefore the binding itself) is quantitative rather than binary (Shao et al., 2012). Additional examples come from recently developed ChIP-exo (ChIP followed by exonuclease digestion and sequencing) experiments. These experiments can detect low occupancy (low numbers of mapped sequencing reads) binding at many genomic locations and the occupancy levels correlate with those from ChIP-seq experiments, indicating that the quantitative information is biologically meaningful rather than experiment artifacts (Rhee and Pugh, 2011).

One common phenomenon revealed by experiments mentioned above is that detected binding sites of different TFs overlap significantly with each other (though their occupancy level at overlapping sites may be different). Since many TFs recognize similar sequences (Gordân et al., 2011) and TFs often recognize degenerated sequences, such phenomenon is not surprising. However, space constraints often mean that one genomic location can only be occupied by one TF at a time. This leads to the competitive nature of TF binding. In addition, most of the genome is packed into nucleosomes (Lee et al., 2007; Brogaard et al., 2012). TFs therefore also need to compete with histone cores for genomic binding sites.

However, in many cases, the binding of different TFs can also be cooperative. For example, Miller and Widom (2003) showed that one TF can facilitate the binding of other proteins by displace nearby nucleosomes. They call this collaborative com-
petition. Cooperative binding can also be achieved through direct protein-protein interactions or through interactions with another protein (Merika and Thanos, 2001). The continuous, competitive yet cooperative nature of transcription factors binding means that TFs interact with genomic binding sites in a complex manner. Rarely one single TF regulates the transcription of a gene. Many TFs act in combination of each other to regulate gene transcription (Buchler et al., 2003). For example, TFs POU5F1, SOX2, and NANOG work together to maintain pluripotency in stem cells (Ferraris et al., 2011).

1.3 Role of chromatin in transcriptional regulation

We are interested in the role of chromatin organization on transcriptional regulation. In particular, the role that nucleosomes, the basic packing units, play in transcriptional regulation. Essentially, nucleosomes occupy DNA and prevent other regulatory proteins, such as transcription factors, from accessing the underlying DNA sequence and therefore (mostly) prevent transcription. Many have reported that most of the genome (between 75% and 90%) is covered by nucleosomes (see, for example, van Holde (1989); Lee et al. (2007) and Brogaard et al. (2012)). Workman (2006) reported that nucleosomes are displaced at both promoter regions and gene bodies during transcription. Some transcription-related genome-wide features of nucleosome positioning are reported, such as the common nucleosome depletion at transcription starting sites (TSS), decreased nucleosome occupancy at highly expressed genes, depletion of nucleosome at certain transcription factors binding sites, etc. (Mavrich et al., 2008; Jiang and Pugh, 2009; Brogaard et al., 2012). Nucleosome positions are also dynamic to accommodate changes in transcription scheme, such as those happen during cell cycle progression and environmental perturbations (Hogan et al., 2006; Lee et al., 2004).

Many also reported that the rotational positioning of nucleosomes, which is de-
fined as the orientation of the major and minor grooves of DNA with respect to the histone surface as it wraps around the histone core, could also regulate the binding of transcription factors (Li and Wrange, 1995; Sekiya et al., 2009; Cui and Zhurkin, 2014).

Nucleosomes also affect transcription by carrying epigenetic modifications. However, these are beyond the scope of this dissertation and will not be discussed.

1.4 Experimental methods in studying protein-DNA interactions

Here we will focus on high-throughput methods that are developed in the last decade or so. Low-throughput methods, such as DNA mobility shift assay (Fried, 1989) and DNase I footprinting assay (Galas and Schmitz, 1978), will not be discussed, although some of them are the basis of high-throughput assays.

Roughly, two classes of high-throughput methods are widely used: ChIP-based methods, such as ChIP-chip and ChIP-seq; and nuclease-digestion–based methods, such as DNase-seq and MNase-seq.

1.4.1 ChIP-based methods

ChIP methods are used to study protein-DNA interactions in vivo. An antibody is used to specifically target a protein of interest (POI). After fragmentation of the genome, POIs can be pulled down by the antibody, together with the pieces of DNA they bind. To ensure stable association of the DNA, a formaldehyde treatment is usually used to cross-link the protein with the DNA it binds. Cross-links are reversed later and the released DNA is assayed to determine its sequence. Two methods are usually used to determine the sequence:

1. Microarray methods. Large numbers of pre-defined DNA sequence probes are printed on glass-based chip (Schena et al., 1995). The DNA sequences bound by POI are amplified (through PCR) and labeled with a fluorescent tag such as
Cy5 before they are placed on the surface of a chip. Those DNA sequences can hybridize with their complementary targets on the chip and emit fluorescence upon forming of double helices. The intensity of the fluorescence can be used to quantify the amount of DNA hybridized to a specific probe. The sequence of the DNA can also be determined from the pre-defined probe sequences.

2. High-throughput sequencing methods. The ability of microarray methods to detect DNA sequence is limited by the pre-defined DNA probes. High-throughput sequencing methods, however, can determine DNA sequences de novo. After amplification of the POI bound DNA sequences, high-throughput sequencing platforms, such as Illumina Genome Analyzer and Applied Biosystems’ SOLiD, are used to determine the sequences. The raw reads from those platforms needs to be mapped back to reference genomes to determine their origin using tools like Bowtie (Langmead et al., 2009).

ChIP-based methods have many advantages compared to the nuclease-based methods in the next section. For example, the identity of the protein assayed is explicitly defined. The resolution of the binding events can be determined potentially in very high resolution, especially in recent methods like ChIP-exo (Rhee and Pugh, 2011). However, ChIP methods is also labor intensive, considering one experiment can only determine the binding locations of one protein under one specific condition and there are hundreds of possible TFs in even small genomes like the one of *S. cerevisiae*. High quality antibodies are also not always available for the POI. Nevertheless, there are currently high volumes of ChIP data available, such as those in Harbison et al. (2004).
1.4.2 Nuclease-digestion–based methods

Nuclease digestion methods are initially developed to map the accessible regions of the genome, because inaccessible regions that are packed into nucleosomes are not digested by nuclease. Two kinds of nucleases are commonly used.

The first is deoxyribonuclease I (DNase I). DNase I is an endonuclease that digests the genomic locations that are not bound by proteins, especially at the so-called DNase hypersensitive regions (DHS). Those are usually open chromatin regions devoid of nucleosomes. Traditionally, DNase I is only used to study DHSs. However, in this dissertation, we will develop effective methods to map nucleosomes using DNase I.

Another nuclease is micrococcal nuclease (MNase). MNase is similar to DNase I, but it is both an endonuclease and an exonuclease. In addition, MNase is smaller than DNase I and therefore can digest linker regions between nucleosomes effectively. MNase can be used to fragment the genome in ChIP experiments to isolate nucleosomes (Schones et al., 2008). Paired-end MNase-seq data were also shown to be able to map nucleosome and smaller protein binding events (Henikoff et al., 2011).

One clear advantage of nuclease digestion method is that they can profile the whole genome simultaneously, regardless of the identities of binding proteins. However, since binding proteins are profiled anonymously, this could also be a disadvantage. In addition, there is an ongoing debate as to whether the results of nuclease digestion methods are trustworthy or not because of their sequence bias (Hörz and Altenburger, 1981; Chung et al., 2010; Allan et al., 2012).

1.4.3 Other in vivo methods

In addition to the methods mentioned above, other genome-wide high-throughput methods have been developed. Examples include:
1. FAIRE-seq (formaldehyde-assisted isolation of regulatory elements) (Giresi et al., 2007). FAIRE-seq relies on the differential cross-linking by formaldehyde between nucleosome associate regions and open, non-nucleosome associated regions to isolate open chromatins;

2. ATAC-seq (assay for transposase-accessible chromatin using sequencing) (Buenrostro et al., 2013). ATAC-seq uses transposase Tn5 to incorporate a tag into open chromatins and subsequently use such tags to identify open chromatins. Such experiments rely on the fact that transposase Tn5 can more effectively integrate tags to open regions compared to nucleosomal regions because steric hindrance prevents the access of the enzyme.

1.4.4 In vitro methods for studying TF binding specificity

The methods mentioned above are in vivo methods that study protein-DNA binding in their native environments. One difficulty for such studies is that proteins may bind DNA through a co-factor indirectly, therefore the binding events discovered by those methods are actually the binding of another protein (Gordân et al., 2009). Many high-throughput in vitro methods are therefore developed to study the DNA binding property of a given protein directly in an isolated environment.

Two notable methods include PBM (protein binding microarray) and MITOMI (mechanically induced trapping of molecular interactions):

• PBM. First introduced by Mukherjee et al. (2004), PBM is similar to the microarray technology used in a ChIP-chip experiment. However, purified proteins, instead of DNA sequences are placed on the microarray surface, on which double strand DNA probes are printed. Fluorescence intensities are correlated with, therefore can be used to measure, the binding affinity of the protein to different DNA probes.
• MITOMI. Maerkl and Quake (2007) developed a method that used microfluidic devices to measure the binding affinity of different DNA sequences to TFs. The method claims to be able to detect transient protein-DNA interactions that other methods, like PBM, are not able to.

1.4.5 Large-scale collaborative projects in studying gene transcription

Despite the development of high-throughput methods, the task of mapping all protein-DNA interactions in different organisms remains daunting for a single lab. Therefore, large-scale collaborative projects have been developed on a global scale to tackle these challenging problems. A couple notable examples are:

• Encyclopedia of DNA Elements (ENCODE) project (The ENCODE Project Consortium, 2004). ENCODE project aims to map the regulatory elements on the human genome. ENCODE has data production centers in both U.S. and Europe. Both the binding of transcription factors and the chromatin accessibility across the genome are essential subjects of the ENCODE project (The ENCODE Project Consortium, 2007, 2012; Spivakov et al., 2012).

• modENCODE. Similar to the ENCODE project, modENCODE project aims to map the comprehensive encyclopedia of genomic functional elements in the model organisms C. elegans and D. melanogaster (Celniker et al., 2009).

1.5 Computational methods in studying protein-DNA interactions

Different computational methods have been developed to make inferences on protein-DNA interactions. Those methods are generally developed to specifically model a certain kind of genomic data. Besides the simple method of using ratios between treatment and background data to find biological signals, a few other, more sophisticated examples include:
1. Motif discovery methods. Methods of this class often take a set of DNA sequences that are (believed to be) bound by a given protein (those sequences could be results of ChIP-chip, ChIP-seq, or PBM experiments) and attempts to output a consensus motif that is overrepresented in this set. Notable examples include AlignACE (Roth et al., 1998), MDScan (Liu et al., 2002), MatrixREDUCE (Foat et al., 2006), RankMotif++ (Chen et al., 2007), and MEME (Bailey et al., 2015). Additional information, such as nucleosome occupancy and sequence conservation, can be used to aid the motif discovery. Such approaches are used in PRIORITY (Narlikar et al., 2006, 2007).

2. Peak calling methods. Determining the peaks (i.e., binding sites) in ChIP-chip or ChIP-seq data is also a challenging problem because of statistical noise and bias in the experiment protocols. Different tools have been developed to tackle this task. Examples include QuEST (Valouev et al., 2008) and MACS (Zhang et al., 2008).

3. Footprint finding methods. To determine the read-depleted regions, which often are the protein footprint regions, different methods were developed for DNase-seq data. For example, Hesselberth et al. (2009) used a simple scoring scheme to rank short segments of genome regions (8–30 bp) based on how much more enriched that segment is compared to its surrounding region (the probability to observe the number of reads in that segment or less based on a background distribution estimated using data from surrounding region). The significance of the score is calculated based on a permutation test. Later, Chen et al. (2010) improved the modeling on the same dataset. They used dynamic Bayesian networks (DBN) to detect short regions that are depleted of sequencing reads. Their DBN is similar to an HMM that includes three hidden states: footprint, background, and hypersensitive. However, the duration of
each state can be controlled in a DBN. Similarly to the work of Hesselberth et al. (2009), Chen et al. (2010) also ranked segments based on calculated test statistics of each identified segments in a permutation test.

Besides those methods that model a single datasets, there are also attempts to jointly model different kinds of genomic data. For example, Lanckriet et al. (2004) used kernel functions to represent different kinds of genome-wide measures, which are then incorporated into a support vector machine (SVM) to distinguish particular classes of proteins from others. Segway (Hoffman et al., 2012) and chromHMM (Ernst and Kellis, 2012) are two similar models that are able to simultaneously model multiple tracks of chromatin data, such as DNase-seq data and histone modification measurements (ChIP-seq) data, to segment the chromatin into several different functional states. Segway used a DBN while chromHMM used a HMM, both of which belong to a more general class of models called graphical models (Koller and Friedman, 2009).

Many models also attempt to model the binding of TFs from thermodynamics point of view. For example, Zhao et al. (2009) used an energy model to learn the affinities of TF-DNA interactions. Both Raveh-Sadka et al. (2009) and Wasson and Hartemink (2009) used a biophysical model to model the binding of multiple proteins, including TFs and nucleosomes, to genomic sequences. These models used solely protein binding specificity and DNA sequences to calculate the binding probability of given proteins to genomic locations, without considering experimental measured binding data, such as ChIP or nuclease digestion data. Those measurements are important: Kaplan et al. (2011) used a similar model and showed that including experimental measurements of genome-wide DNA accessibility could improve the prediction of binding dramatically.
Figure 1.5: An example binding configuration near a promoter region. In this configuration, four nucleosomes (Nuc) and two TFs are included.

1.5.1 Overview of a biophysical model for multi-factor binding

Part of the work in this dissertation is based on the biophysical model called COMPETE introduced by Wasson and Hartemink (2009). Therefore, a brief overview of the model is included here to establish notations used in this model.

We first define the concept of a binding configuration. We call a specific arrangement of DBFs along a DNA sequence as a binding configuration. A particular location on the sequence could also be unbound (not bound by any DBFs). A valid binding configuration must not have overlapping DBFs. Figure 1.5 shows an example of a binding configuration near a promoter region. A binding configuration can also be viewed as a snapshot of protein-DNA interaction in the cell in a thermodynamic equilibrium. As a biophysical model, COMPETE models an ensemble of binding configurations and calculates the probability of observing one specific binding configuration \( i \) according to the Boltzmann distribution:

\[
p_i = \frac{1}{Z} \times e^{-E_i}
\]

where \( E_i \) is the binding free energy of the configuration in the units of \( k_B T \) (\( k_B \) is the Boltzmann constant and \( T \) is the temperature in degrees Kelvin). \( Z \) is the partition function:

\[
Z = \sum_{\text{all possible binding configuration } j} e^{-E_j}
\]
To proceed further on the calculation of probabilities, COMPETE makes certain assumptions. First, we define the following notations:

1. Denote a DBF using $\pi$ and assume in binding configuration $i$ there are $K$ DBFs: $\pi_1, \ldots, \pi_K$;
2. Each DBF has length $L_k$. In other words, DBF $\pi_k$ will occupy $L_k$ positions on the DNA sequence once it binds to the sequence. Position $l$ of $\pi_k$ can therefore be denoted as $\pi_{k,l}$;
3. The concentration of $\pi_k$ is $c(\pi_k)$;
4. The nucleotide bound by $\pi_k$ at its $l^{th}$ position is $S_{k,l}$, $S_{k,l} \in \{A, C, G, T\}$.

For a given binding configuration $i$, COMPETE assumes an additive energy model:

$$E_i = -\sum_k \log (c(\pi_k)) - \sum_k EN_k$$

where $EN_k$ is:

$$EN_k = \sum_{l=1}^{L_k} \log (e(S_{k,l}|\pi_{k,l}))$$

$EN_k$ can be viewed as the free energy of $\pi_k$ binding to the sequence $S_{k,1}, \ldots, S_{k,L_k}$. $EN_k$ can be calculated from complex models. However, COMPETE assumes a simple position weight matrix (PWM) model where each of $S_{k,l}$ contribute independently. This contribution is represented by $e(S_{k,l}|\pi_{k,l})$ and can be understood as the probability of observing nucleotide $S_{k,l}$ given that it is bound by $\pi_{k,l}$.

Therefore, the probability $p_i$ can be shown to be:

$$p_i = \frac{\prod_{k=1}^K c(\pi_k) \prod_{l=1}^{L_k} e(S_{k,l}|\pi_{k,l})}{Z}$$

To calculate $p_i$ for all possible binding configuration, a naive enumeration is not feasible because the total number of possible configurations grows exponentially as
a function of the DNA sequence length. However, the above problem can be formulated as a HMM: The states of the HMM correspond to \( \pi_{k,l} \)'s; each state emits the nucleotide it binds with probability \( e(S_{k,l} | \pi_{k,l}) \); the transition probability for consecutive states within the same DBF is always 1 and the transition probability for transitioning into the first state of a DBF \( \pi_k \) is \( c(\pi_k) \) (independent of the previous state). In this formulation, each binding configuration corresponds to a state path in the HMM and \( p_i \) (as well as the partition function) can be calculated using the efficient forward-backward algorithm. Note that in COMPETE, \( c(\pi_k)'s \) are not constrained as probabilities. Therefore the model is more similar to a Boltzmann chain (Saul and Jordan, 1995) and \( c(\pi_k)'s \) are called transition weights in COMPETE. Figure 1.6 shows the specific model structure.

Once we know \( p_i \), the probability of DBF \( \pi_k \) binds to a given sequence location is simply the sum of all \( p_i \) such that DBF \( \pi_k \) binds to the location in binding configuration \( i \) (Figure 1.7).

1.6 Organization of this dissertation

The rest of the dissertation is organized as follows. In Chapter 2 we introduce a framework that incorporates MNase-seq data into COMPETE. We use MNase-seq data to learn a set of biological plausible parameters for the biophysical model of COMPETE and show that the learned model improve markedly in terms of predicting a protein-DNA binding landscape that is consistent with data.

The overall effort of the dissertation is to develop a statistical modeling framework that is able to integrate different genomic data to learn protein-DNA interaction landscape. To that end, we devote Chapter 3 to carefully study a particular kind of data, the DNase-seq data, and show that it contains substantial signal for positioning nucleosomes, in addition to its ability to locate open chromatinis. We use a Bayes-factor–based method to model the quadratic and oscillatory DNase I digestion
Figure 1.6: The COMPETE model structure. A blue circular node corresponds to a single state within a DBF. COMPETE also uses silent states that do not emit any nucleotides to facilitate transition (dashed circular nodes). A module of states, such as a TF or a nucleosome, is shown in an orange rectangle. (A) The overall structure of COMPETE. (B) A module of nucleosome. (C) and (D) A module of a TF and an origin recognition complex (ORC), respectively. Figure adapted from Wasson and Hartemink (2009).

pattern on nucleosomes for effective extraction of nucleosome positioning signal.

Based on our work in Chapters 2 and 3, we develop a multivariate state-space model in Chapter 4 that is able to model multiple datasets simultaneously to learn the protein-DNA interaction landscape. The model formulation and theoretic derivations of a training algorithm are presented. Applications to synthetic data and real datasets are also presented. These results show that our model can efficiently learn a holistic view of protein-DNA interaction landscape that is concordant with data.

In the last chapter, we discuss future improvements to the modeling approach,
Figure 1.7: Calculation of the probability that a DBF $\pi_k$ binds to a given genomic location is simply a summation of probabilities of all relevant binding configurations in the ensemble.

possible applications of the work and conclude the dissertation.
Learning a multi-factor binding model from MNase-seq data

The biophysical model of COMPETE that we introduced in Chapter 1 is able to model multiple factors binding to the genome simultaneously, including the nucleosome. This is a significant improvement over other models that model the protein-DNA interactions. However, as a purely theoretical model of binding, its major limitation is that its predictions can be superfluous if the parameters are not guided by data. It is therefore necessary to develop a new computational framework for jointly interpreting experimentally derived data regarding genomic occupancy within a model built upon the thermodynamic foundation of COMPETE.

In this chapter, we develop just such a method: a general framework for combining both a thermodynamic model for protein-DNA interactions (along the lines of COMPETE) and a new statistical model for learning from experimental observations regarding those interactions. Information from different experimental observations can be integrated to infer the actual thermodynamic interactions between DBFs and a genome. In this particular study, we demonstrate the use of this framework
by integrating paired-end micrococcal nuclease sequencing (MNase-seq) data, which reveals information about the binding occupancy of both nucleosomes and smaller (subnucleosomal) factors. Our framework also integrates protein binding specificity information from PBM data and produces a more accurate and realistic protein-DNA interaction landscape than COMPETE alone, along with a mechanistic explanation of MNase-digested fragments of different sizes. The cross-validated performance of our framework is significantly higher than several baselines to which we compared it. Our framework is flexible and can easily incorporate other data sources as well. We demonstrate this by integrating experimentally measured cellular protein concentrations through prior distributions. The framework thus represents a general modeling framework for integrating multiple sources of information to produce a more precise view of the interaction landscape undergirding transcriptional regulation.

The work presented in this chapter is based on our work that appeared in Zhong et al. (2014).

2.1 Methods

2.1.1 Model formulation and model learning objective

As mentioned above, the core model based on which we build the framework is the biophysics model of COMPETE. A more detailed model formulation of COMPETE can be found in Section 1.5.1. Briefly, the protein-DNA interaction landscape (probabilities of proteins binding along the DNA sequence) is calculated by calculating the statistical weights of binding configurations in an ensemble. Following the Boltzmann distribution, the statistical weight $w_i$ of configuration $i$ can be shown to be

$$w_i = \prod_{t=1}^{N_i} X_t \times P(S_t, E_t|DBF_t)$$
where $t$ is an index over the $N_t$ DBF binding sites in configuration $i$, $X_t$ denotes a weight associated with DBF $t$, while $S_t$ and $E_t$ denote the start and end position of the DBF binding site, respectively. $P(S_t, E_t|DBF_t)$ is the probability of observing the DNA sequence between $S_t$ and $E_t$, given that DBF $t$ is bound there. To simplify notation, we have treated each unbound nucleotide as being bound by a special kind of ‘empty’ DBF. If we use $p_i$ to denote the probability of configuration $i$ after normalization by the partition function, we can write the probability that DBF $t$ binds at a specific position $j$ as:

$$
\sum_{i \in I(t, j)} p_i
$$

where $I(t, j)$ is the subset of binding configurations in the ensemble that have DBF $t$ bound at sequence position $j$.

This model can be formulated analogously to a hidden Markov model (HMM) (Rabiner, 1989), in which the states correspond to the binding of different DBFs and the observations are the DNA sequence. The various probabilities, along with the partition function, can then be calculated efficiently using the forward-backward algorithm. For transcription factors, we have chosen to represent $P(S_t, E_t|DBF_t)$ using a position weight matrix (PWM), but more sophisticated models can also be used (e.g., relaxing positional independence, or based on energies rather than probabilities (Weirauch et al., 2013)). Regardless, binding models from different sources and of different forms can be easily incorporated into our model, generating the appropriate states and sequence emission probabilities. We use the curated PWMs from Gordân et al. (2011), derived from in vitro PBM experiments, as the input protein binding specificities and consider them fixed (though our framework also could allow them to be updated).

The analogues of HMM transition probabilities in our model are the DBF weights, but these are not constrained to be probabilities. To allow this flexibility, we adopt
a more general statistical framework called a Boltzmann chain (Saul and Jordan, 1995) which can be understood as an HMM that allows the use of any positive real numbers for these weights. Because of the analogy with an HMM, we henceforth refer to these DBF weights as ‘transition weights’ and denote them collectively as a vector $X = (X_1, X_2, \ldots, X_D)$, where $D$ is the number of different kinds of DBFs.

Setting appropriate transition weights is challenging. Indeed, incorrectly set parameters can lead to erroneous predictions that do not make biological sense (Figure 2.1). The overall goal of the general framework we are developing here therefore is to learn the transition weights from data. Later in this chapter we will describe the experimentally derived genomic data and the inference algorithm we employ to learn the $D$ free parameters in the vector $X$. We should note that the DBF transition weights in a Boltzmann chain are sometimes called “concentrations”. However, it is important to point out that these transition weights are not the same as bulk cellular protein concentrations, of the kind that can sometimes be measured experimentally (Ghaemmaghami et al., 2003). Bulk cellular protein concentrations are not necessarily indicative of the availability of a DBF to bind DNA, because they do not account for phenomena like subcellular localization or extranuclear sequestration, protein activation through post-translational modification or ligand or co-factor binding, or the number of DBFs already bound to DNA. In contrast, our transition weights correspond to nuclear concentrations of active proteins that are free and available to bind DNA. In this sense, our weight parameters are more reasonably interpreted not as cellular concentrations but rather as the chemical potentials of the DBFs for interacting with the genome.
Figure 2.1: Erroneous predictions of COMPETE when the transition weights are set incorrectly. Shown in the figure are COMPETE predictions on six promoter regions. The transition weights used in the prediction are randomly sampled from the uniform prior distributions mentioned in Section 2.1.6. In almost all cases, TFs (color legends not shown) completely occupied the entire promoter region. Comparing these random profiles with the fitted profiles in Figure 2.5 shows the importance of learning from data.

2.1.2 Using paired-end MNase-seq data as measures of genomic occupancy levels of DNA-binding proteins

We used paired-end MNase-seq data from Henikoff et al. (2011). Based on their protocol, the length of the sequencing fragments correspond roughly to the size of the protein protecting that part of the DNA; the number of fragments mapping to the location correlates with the binding strength or occupancy. Therefore, to measure the level of occupancy of different DNA binding proteins, we separate the fragments into long (140–200 bp) and short (0–100 bp) fragment groups and count the number
of fragments in each group that cover a specific genomic location (called long and short fragment coverage, respectively). The long fragment coverage is used as a measure of the occupancy of large protein complexes, which are mainly nucleosomes, while the short fragment coverage is used as a measure of the occupancy of smaller proteins, which are mainly transcription factors.

Given the long and short coverage data (each in the form of a series counts data, one count for each genomic coordinate), we apply the following transformation to them:

1. we define a bottom threshold $T_b$ and a top threshold $T_t$;

2. coverage values that are below $T_b$ are converted to 0, while those above $T_t$ are converted to 1;

3. coverage values between the two thresholds are normalized linearly to $[0,1]$;

4. the resultant data series are then smoothed using a Gaussian kernel of bandwidth $B_m$.

After applying the above transformation separately to long and short fragment coverage data, we get the large and small protein binding profiles, respectively (Figure 2.2 D and E). The reason for this transformation is twofold: we want to convert the measure of coverage (in the unit of counts) to the measure of probability of protein binding; and through thresholding and smoothing, we wish to reduce the noise in the coverage data.

In the results that followed, we used

$$T_b = 200, \quad T_t = 500$$

with $B_m = 10$ for short fragment coverage

and $B_m = 30$ for long fragment coverage.
These values give satisfying results in terms of reducing noise while retaining clear peaks. A sensitivity analysis was performed later to ensure that our results are unaffected across a broad range of these parameters (Section 2.2.5).

We also note that MNase is known to prefer to cut A/T compared to G/C. We also assessed the severity of this well-known bias and observed that it does not affect our final results (Section 2.2.4).

2.1.3 Incorporating MNase-seq data through an objective function

We model MNase-seq data through a correlation-based objective function, which we call a “pseudo-likelihood” function. To calculate the function, we apply the following transformation to the COMPETE calculated TF binding probabilities:

1. each TF binding event is expanded to a flanking region of $C_e$ bp, and then dropped linearly to 0 for another $C_r$ bp;

2. we then sum the probabilities of all TFs (truncating values larger than 1);

3. the resultant track is smoothed using a Gaussian kernel of bandwidth $B_c$.

We call the transformed series the composite TF binding profile (Figure 2.2C). We process the occupancy profile in such a way for two reasons: (a) the resolution of the short fragment coverage does not distinguish protection from adjacent proteins, and (b) MNase does not completely digest all unprotected DNA, leaving some additional nucleotides flanking any TF’s actual binding site. We choose

$$C_e = C_r = B_c = 10$$

As with the threshold and bandwidth parameters discussed above, varying the specific values tends to have only small effects on the model predictions. We do not process the nucleosome profile predicted by COMPETE since the model already takes nucleosome padding into consideration.
Figure 2.2: Overview of how the objective function is evaluated and iteratively optimized.

(A) Predicted probability that each particular DBF binds at a given genomic position, as calculated by COMPETE, given current DBF weights. We then separate these probabilities into two profiles: (B) predicted nucleosome binding profile and (C) predicted composite TF binding profile in which protein identities have been removed; the latter is smoothed to make it comparable to a short fragment coverage profile. Similarly, we separate the observed MNase-seq fragments (F) into long (140–200 bp) and short (0–100 bp) fragments, which are summed to produce measures of coverage. (D) Total long fragment coverage is processed into a large protein binding profile, which is compared to predicted nucleosomal binding, arriving at Pearson correlation $r_1$. (E) Total short fragment coverage is processed into a small protein binding profile, which is compared to predicted composite TF binding, arriving at Pearson correlation $r_2$. For this promoter, the quantity $h$ that appears in our objective function (the pseudo-likelihood) is simply the geometric mean of the two correlations, after they are rescaled to lie in the interval $[0, 1]$: $h = \frac{1}{2} \sqrt{(1 + r_1) \times (1 + r_2)}$.

The complete pseudo-likelihood over all promoters is then optimized with respect to the DBF weights using the inference method described in Section 2.1.6.
We also capture the effect of the pre-initiation complex (PIC) on MNase-seq profile by adding empirical PIC protection shape to predicted binding profile (see Section 2.1.5 for details).

For promoter region \( m \), we calculate two correlations: the Pearson correlation \( r_{1,m} \) between the nucleosome binding profile and the MNase-seq long fragment coverage profile, and the Pearson correlation \( r_{2,m} \) between the composite TF binding profile and the MNase-seq small protein coverage profile. The complete pseudo-likelihood function we seek to maximize is defined as

\[
L(X) = \prod_{m=1}^{M} h_m(X)
\]

where

\[
h_m(X) = \frac{1}{2} \sqrt{(1 + r_{1,m}) \times (1 + r_{2,m})}
\]

the geometric mean of the two rescaled correlations for promoter region \( m \) (an example is shown in Figure 2.2). Note that \( h_m(X) \) depends on the vector of DBF weights \( X \). In this study, \( M = 81 \).

2.1.4 Selecting a subset of TFs and promoter regions

Our framework has the capability to include all \( S. \text{cerevisiae} \) transcription factors. However, our choice of transcription factors is limited by available high quality binding preference data. In addition, adding more TFs increases the dimensionality of the parameter space and therefore the computation time required to explore the space. In this study, we chose a set of 42 TFs with available high quality binding preference data. These TFs cover a wide range of cellular functions including:

- chromatin remodeling activity (Reb1, Rap1, and Abf1);
- pheromone response (Ste12 and Tec1);
• stress response (Msn4);

• cell cycle regulation (Fkh1, Mbp1, and so forth).

We also included some TFs, like Pho2 and Phd1, that regulate a large number of genes according to MacIsaac et al. (2006). While these 42 TFs do not represent all yeast TFs, they are collectively responsible for 66% of the genome-wide protein-DNA interactions reported by MacIsaac et al. (2006) (at p-value < 0.005 and conservation level 3). The full list of the TFs can be found at Table 2.1. Having selected our 42 TFs, we next chose a set of promoter regions that, according to MacIsaac et al. (2006) (at p-value < 0.005 and conservation level 3), are bound exclusively by those TFs. For this study, we focus on 81 such promoter regions (see full list in Table 2.2),

Table 2.1: Transcription factors used in the framework

<table>
<thead>
<tr>
<th>TF</th>
<th>function</th>
<th>TF</th>
<th>function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abf1</td>
<td>chromatin remodeling</td>
<td>Aft2</td>
<td>iron-regulated TF</td>
</tr>
<tr>
<td>Adr1</td>
<td>carbon source-responsive TF</td>
<td>Cbf1</td>
<td>chromosome segregation</td>
</tr>
<tr>
<td>Cin5</td>
<td>drug resistance; salt tolerance</td>
<td>Fkh1</td>
<td>cell cycle regulation</td>
</tr>
<tr>
<td>Cst6</td>
<td>carbon sources utilization, etc.</td>
<td>Ace2</td>
<td>cellular polarity, etc.</td>
</tr>
<tr>
<td>Dal80</td>
<td>nitrogen degradation</td>
<td>Dal82</td>
<td>allantoin degradation</td>
</tr>
<tr>
<td>Fhl1</td>
<td>ribosomal protein transcription</td>
<td>Gcn4</td>
<td>amino acid biosynthetic</td>
</tr>
<tr>
<td>Gat1</td>
<td>nitrogen catabolite repression</td>
<td>Hap1</td>
<td>heme activator protein</td>
</tr>
<tr>
<td>Gcr1</td>
<td>glycolysis</td>
<td>Gln3</td>
<td>nitrogen catabolite repression</td>
</tr>
<tr>
<td>Hcm1</td>
<td>cell cycle regulation</td>
<td>Rap1</td>
<td>chromatin silencing, etc.</td>
</tr>
<tr>
<td>Mbp1</td>
<td>cell cycle regulation</td>
<td>Pho2</td>
<td>biosynthesis</td>
</tr>
<tr>
<td>Mcm1</td>
<td>pheromone response, etc.</td>
<td>Phd1</td>
<td>enhances pseudo hyphal growth</td>
</tr>
<tr>
<td>Met31</td>
<td>sulfur metabolism</td>
<td>Nrg1</td>
<td>filamentous growth, etc.</td>
</tr>
<tr>
<td>Mot3</td>
<td>repressor of aerobic growth, etc.</td>
<td>Msn4</td>
<td>stress response</td>
</tr>
<tr>
<td>Rox1</td>
<td>hypoxic gene repressor</td>
<td>Sfp1</td>
<td>regulator of ribosomal protein, etc.</td>
</tr>
<tr>
<td>Skn7</td>
<td>oxidative stress response</td>
<td>Smo1</td>
<td>osmotic stress response;</td>
</tr>
<tr>
<td>Sok2</td>
<td>pseudohyphal differentiation</td>
<td>Reb1</td>
<td>Pol I enhancer binding protein</td>
</tr>
<tr>
<td>Ste12</td>
<td>pheromone response</td>
<td>Sut1</td>
<td>sterol uptake, etc.</td>
</tr>
<tr>
<td>Swi4</td>
<td>cell cycle regulation</td>
<td>Swi5</td>
<td>cell cycle regulation</td>
</tr>
<tr>
<td>Tec1</td>
<td>pheromone response</td>
<td>Ume6</td>
<td>chromatin remodeling</td>
</tr>
<tr>
<td>Xbp1</td>
<td>stress or starvation response</td>
<td>Yap6</td>
<td>carbohydrate metabolism</td>
</tr>
<tr>
<td>Yap7</td>
<td>Putative TF</td>
<td>Yox1</td>
<td>cell cycle regulation</td>
</tr>
</tbody>
</table>
and extracted MNase-seq data for these loci as follows:

- if the promoter is divergently transcribed, we extracted the MNase-seq data between the two TATA elements, plus 200 bp downstream of each TATA element;

- for the other (non-divergent) promoters, we extracted MNase-seq data 500 bp upstream of the TATA element (or 100 bp upstream of the end of the upstream gene, whichever is smaller), and 200 bp downstream of the TATA element.

Locations of TATA elements were taken from Rhee and Pugh (2012).

2.1.5 Incorporating the pre-initiation complex

The pre-initiation complex (PIC) assembles at nucleosome-free promoter regions and facilitates transcription initiation and regulation. PICs compete with other DBFs for binding sites when they are assembled around TATA or TATA-like elements (henceforth referred to as TATA boxes, for simplicity). To account for this competition, we calculate the TATA-binding protein (TBP) binding probability in our model using the DNA binding specificity derived from Rhee and Pugh (2012).

Because of the degenerate nature of the TBP binding motif, we amend our model to allow this competition to occur only at TATA boxes by setting the transition weight for TBP to be zero at all sequence locations except TATA boxes. We modified the COMPETE model implementation to allow this position specific transition weights.

Rhee and Pugh (2012) report that core PICs (TBP-associated factors and general transcription factors) assemble approximately 40 bp downstream of TATA boxes. The MNase digestion data used here also show an enrichment of short fragments coverage at the same location (Figure 2.3A). Therefore, we approximate the PIC protection by adding the same MNase short fragment coverage shape (Figure 2.3B; scaled by the probability of TBP binding calculated by COMPETE) to the predicted
Table 2.2: Promoter regions used in the framework

<table>
<thead>
<tr>
<th>chr.</th>
<th>coordinate (sacCer2)</th>
<th>name</th>
<th>chr.</th>
<th>coordinate (sacCer2)</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIII</td>
<td>187545-188041</td>
<td>YHR039C-A</td>
<td>VII</td>
<td>541475-542175</td>
<td>YGR028W</td>
</tr>
<tr>
<td>IV</td>
<td>217026-217726</td>
<td>YDL136W</td>
<td>V</td>
<td>147492-148192</td>
<td>YEL003W</td>
</tr>
<tr>
<td>VII</td>
<td>729585-730328</td>
<td>YGR119C</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>X</td>
<td>413270-413970</td>
<td>YJR012C</td>
<td>XVII</td>
<td>528545-529245</td>
<td>YPR012C</td>
</tr>
<tr>
<td>IV</td>
<td>716407-716988</td>
<td>YDR130C</td>
<td>III</td>
<td>528545-529245</td>
<td>YPR012C</td>
</tr>
<tr>
<td>IV</td>
<td>1447251-1447951</td>
<td>YDR499W</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>IV</td>
<td>979094-979794</td>
<td>YDL136W</td>
<td>IV</td>
<td>147492-148192</td>
<td>YEL003W</td>
</tr>
<tr>
<td>V</td>
<td>147492-148192</td>
<td>YER012C</td>
<td>V</td>
<td>147492-148192</td>
<td>YEL003W</td>
</tr>
<tr>
<td>VII</td>
<td>729585-730328</td>
<td>YGR119C</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>X</td>
<td>413270-413970</td>
<td>YJR012C</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>IV</td>
<td>716407-716988</td>
<td>YDR130C</td>
<td>III</td>
<td>528545-529245</td>
<td>YPR012C</td>
</tr>
<tr>
<td>X</td>
<td>413270-413970</td>
<td>YJR012C</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>IV</td>
<td>716407-716988</td>
<td>YDR130C</td>
<td>III</td>
<td>528545-529245</td>
<td>YPR012C</td>
</tr>
<tr>
<td>X</td>
<td>413270-413970</td>
<td>YJR012C</td>
<td>V</td>
<td>265699-266399</td>
<td>YER055C</td>
</tr>
<tr>
<td>IV</td>
<td>716407-716988</td>
<td>YDR130C</td>
<td>III</td>
<td>528545-529245</td>
<td>YPR012C</td>
</tr>
</tbody>
</table>
small protein binding probability downstream of the TATA box. We observed that adding PICs allowed the nucleosome free regions to agree better with the MNase fragment data, because the PICs both enhance the exclusion of nucleosomes and explain some of the small MNase fragments downstream of the TATA-like element (so that TFs are not needed to provide that explanation).

**Figure 2.3:** PIC protection shape downstream of TATA elements. (A) We selected the 100 most highly transcribed promoters, ranked by the number of NET-seq (nascent elongation transcription sequencing) reads (Churchman and Weissman, 2011) mapping to 500 bp downstream of the TATA element. From these, we compute and plot the average long (black) and short (blue) fragment coverage. The plot reveals a PIC protection peak in the average short fragment coverage, located approximately 40 bp downstream of TATA element. This is consistent with the findings of Rhee and Pugh (2012). (B) We processed the average short fragment coverage the same way as we processed MNase-seq short fragment coverage data through thresholding and smoothing. The resultant PIC protection shape (0–100 bp downstream of TATA element) was extracted from the smoothed profile. This protection shape (multiplied by the corresponding COMPETE predicted TBP binding probability) is added to predicted small protein binding probability downstream of the TATA element.
2.1.6 Inference method

We use Markov chain Monte Carlo (MCMC) to explore a posterior distribution based on the pseudo-likelihood function. However, since correlation measures the overall goodness of fit for many genomic locations at once, our pseudo-likelihood function is much flatter than typical likelihood functions. This property can be useful in preventing overfitting, but it also imposes some difficulty for parameter inference. To alleviate this concern, and allow for more efficient MCMC exploration, we apply a temperature parameter $\tau$ to each dimension of the search space in order to concentrate the mass of $L(X)$ around its modes. We apply a possibly different temperature to each dimension (i.e., each element of the vector $X$) because the pseudo-likelihood in one dimension may be more or less flat than in others. We base our choice of temperature parameter on the MCMC acceptance rate, and empirically set $\tau$ for each dimension to be one of $\{0.1, 0.05, 0.01, 0.002\}$. Note that none of these choices change the local maxima of our objective function in any way; they simply may make convergence more efficient.

As for the prior over $X$, a nice feature of our framework is that we can use non-uniform priors if there is reason to do so; later, we explore the possibility of including mildly informative priors for certain TFs where measurements of cellular concentrations in *Saccharomyces cerevisiae* are available (Ghaemmaghami et al., 2003). However, when no relevant information is available, a uniform prior distribution is a natural choice. In what follows, we use a uniform prior over $[-10, 2]$ for log transition weights of TFs, a uniform prior over $[-1, 3]$ for the log transition weight of nucleosomes and uniform prior over $[-5, 5]$ for the log transition weight of TBP. Such values are chosen based on the range of TF dissociation constants at their respective optimal binding sites ($K_d$, as defined and computed by Granek and Clarke (2005)): Sig1 has the highest log $K_d$ value of $-2.5$ and Asg1 has the lowest log $K_d$ value of $-7.6$. 

43
To summarize, our MCMC inference algorithm includes the following steps:

1. Draw a random value for each element of $X$ from its corresponding prior distribution as its initial value. Use this initial vector $X_0$ to calculate the pseudo-likelihood $L(X_0)$;

2. Define a proposal distribution for each DBF. We use a truncated log-normal distribution. The mean of the distribution is the current value of the DBF transition weight (initially its $X_{d,0}$ for DBF $d$). The standard deviation of the distribution is initially set to 3.0. The truncation region for a TF is $[-10, 2.0]$, for the TBP is $[-5.0, 5.0]$ and for the nucleosome is $[-1.0, 3]$ (again those values are determined as above).

3. At iteration $i$ ($i$ starts from 1), loop through each elements of $X$, in the $d^{th}$ loop ($1 \leq d \leq D$):
   
   (a) Draw a random value from the corresponding proposal distribution of $X_{d,i}$, which is a truncated log-normal distribution described above (with mean $X_{d,i-1}$). Call this new value $X_{d,i}^*$. Together with other values, we have a vector:
   
   $$X_{d,i}^* = \{X_{1,i}, \ldots, X_{d-1,i}, X_{d,i}^*, X_{d+1,i-1}, \ldots, X_{D,i-1}\}$$
   
   (b) Calculate $L^*(X_{d,i}^*)$. Since $L(X)$ is a correlation based pseudo-likelihood, it can not be calculated when the output of COMPETE has no TF binding or nucleosome binding (the standard deviation for the composite TF binding profile or the nucleosome profile is 0 in this case). When this happens, we reject the value of $X_{d,i}^*$ and set $X_{d,i}$ to $X_{d,i-1}$. When this does not happen, we proceed to the following step. Also, at this step, we should have already calculated $L(X_{d-1,i})$;
(c) Calculate the following quantity:

\[
a = \left( \frac{L^*(X^*_{d,i})}{L(X_{d-1,i})} \right)^{\frac{1}{\tau_d}} \times \frac{P(X^*_{d,i})}{P(X_{d,i-1})}
\]

where \( P \) denotes the prior for \( X_d \) and \( \tau_d \) is the temperature parameter mentioned above;

(d) If \( a \geq 1 \), we accept \( X^*_{d,i} \), otherwise we accept \( X^*_{d,i} \) with probability \( a \). If we accept \( X^*_{d,i} \), we set \( X_{d,i} = X^*_{d,i} \), otherwise we set \( X_{d,i} = X_{d,i-1} \).

4. Increment \( i \) to \( i + 1 \) and start the loop again. At each loop of each iteration, we record whether a proposed value is accepted or not. We calculate the acceptance rate of each DBF every 50 iterations. We can then scale the standard deviation of each DBF’s proposal distribution according to the acceptance rate, as follows:

\[
\text{if acceptance rate} > 0.46, \text{ set } new\_sd = current\_sd \times e^{adp}
\]

\[
\text{if acceptance rate} < 0.42, \text{ set } new\_sd = \frac{current\_sd}{e^{adp}}
\]

where \( adp \) is the adaptation rate:

\[
adp = \min(0.05, \frac{1}{i})
\]

that is we adapt the proposal standard deviation faster at the beginning of the iterations and slower as iteration number grows.

In our Gibbs-style MCMC, each iteration consists of a loop for each of the transition weight parameters in the model. On a commodity computer cluster, we could compute roughly 25 such iterations per hour.
2.1.7 Framework implementation

To implement our framework, we need to three major components:

1. The core COMPETE model implementation to predict the protein-DNA interaction landscape given a set of parameters. The COMPETE implementation should also allow position specific transition weights so that we can incorporate the PIC only at TATA boxes. Our COMPETE implementation is based on the implementation of Wasson and Hartemink (2009);

2. A set of tools of to calculate the pseudo-likelihood function and carry out the MCMC inference algorithm;

3. Tools to carry out parallel computing in a computer cluster environment, since we need to run COMPETE on each of the $M$ promoter regions to calculate the pseudo-likelihood function at each iteration of the MCMC algorithm. These $M$ COMPETE runs are time-consuming, but can be embarrassingly parallelized.

Implementations of all three components can be found at:

https://jianlingzhong.github.io/

2.2 Results

2.2.1 Overall inference performance evaluated by cross-validation

We randomly split our 81 promoter regions into nine equal sets and performed a standard nine-fold cross-validation: parameters were trained on 72 promoter regions using MCMC and we used the mean of MCMC samples as trained DBF weights $\hat{X}$; we then calculated $h(\hat{X})$ values for the held out nine promoter regions. Figure 2.4 shows boxplots of $h(\hat{X})$ values of all the training and testing promoter regions from all the folds of cross-validation. We compare the performance to five baselines:
- Average performance when log transition weights are drawn 1000 times uniformly under the prior;
- Setting TF transition weights to $8 \, K_d$, nucleosome transition weight to 35;
- Setting TF transition weights to $16 \, K_d$, nucleosome transition weight to 35;
- Setting TF transition weights to $32 \, K_d$, nucleosome transition weight to 35;
- Setting TF transition weights to $64 \, K_d$, nucleosome transition weight to 35;

As Figure 2.4 shows, our learned model outperforms all five baselines significantly. Note that $h(X) = 0.5$ indicates no correlation on average between the model predictions and observed data. Median performance for the random baseline is still larger than 0.5 even with uninformed TF transition weights; this is because the model’s emission parameters (derived from in vitro experimental data regarding TF and nucleosome binding specificity) are highly informed.

2.2.2 A mechanistic explanation for paired-end MNase-seq data

Owing to in vitro experiments, our model has knowledge about inherent DBF sequence specificities. The thermodynamic interaction and competition between these DBFs are accounted for by COMPETE. By adding information about in vivo DBF binding occupancy levels present in MNase-seq data, our framework can now infer a DBF binding landscape that provides a mechanistic explanation for the observed data.

Figure 2.5 illustrates examples of predicted binding profiles for each DBF in six promoter regions in the test sets of the nine-fold cross-validation, in comparison with the corresponding MNase-seq binding profile tracks. These examples span the full spectrum of our framework performance, from strong performance to weak performance. In all cases, our predictions for the TF binding profiles provide a good or
fair explanation for the MNase-seq data and are much more consistent with the data compared to random baseline predictions (Figure 2.1). Appendix Figure A.1 provides additional comparisons between composite predicted profiles and processed MNase-seq fragment coverages. In Appendix Figure A.2, we provide the raw coverage for 21 regions, with comparisons with in vitro MNase-seq coverages.

One difficulty in interpreting high-throughput nuclease digestion data is identifying DBFs at read-enriched regions. Traditional motif matching is not satisfactory when there are multiple potentially overlapping motifs, nor can it assess the strength of protein binding. In contrast, our framework provides a principled interpretation for the data in terms of distinct binding events, each with its own probability of occurrence.

**Figure 2.4:** Comparison of cross validated inference performance to various baselines. Data from the 81 promoter regions were split into nine equal parts. A standard nine-fold cross-validation procedure was applied: 72 promoter regions were used as training data to obtain trained DBF weights $\hat{X}$; we then calculated $h(\hat{X})$ values of the held out nine promoter regions (testing results). “CV training result” considers the $h(\hat{X})$ values for each promoter when used as training data. “CV testing result” shows the $h(\hat{X})$ values for each promoter when used as testing data. Uniformly drawn TF transition weights and different multiples of $K_d$ are used as baseline comparisons. Variance is reduced in the random baseline case because each result is the average of 1,000 random samples.
Figure 2.5: Predicted binding profiles versus MNase-seq binding profiles. For six promoter regions in our 81 promoter set, we plot the predicted binding profiles when they were evaluated as testing data. We also indicate reported binding sites from ChIP-exo (Rhee and Pugh, 2011) underneath the predicted binding profiles; these have the same color as the corresponding TF’s binding probability. No binding event is reported by MacIsaac et al. (2006, p-value < 0.001 and conservation level 3) for these promoter regions.

currence based on evaluating the probability of every possible binding configuration in the ensemble. This is demonstrated, for example, in the YHR039C-A, YBL014C, and YPR016C promoter regions (Figure 2.6).

Our approach can also capture weak binding events, such as the Reb1 binding events in the YPR016C and YNL157W promoter regions, which are missed in ChIP-chip experiments (MacIsaac et al., 2006) but are captured in ChIP-exo experiments (Rhee and Pugh, 2011) (Figure 2.5).

Our predictions of nucleosome binding profiles match the data well in spite of
Figure 2.6: The fitted protein-DNA interaction landscape provides mechanistic explanations for the observed MNase paired-end sequencing fragments. Top panels, the observed MNase-seq fragments, plotted as horizontal bars in a genome coordinate vs. fragment size 2D plot. Bottom panels, the fitted protein-DNA interaction landscapes. The fitted landscapes explain the fragments in terms of the binding proteins and their respective binding probabilities.

the fact that nucleosome positioning is less precise than TF positioning. The predictions reflect the intrinsic uncertainty about nucleosome positioning related to their mobility and only mildly sequence preferences, especially when the MNase-seq large protein binding profile is more noisy, as in the promoter regions of YBL014C and YNL157W (Figure 2.5; see Appendix Figure A.2 for raw coverage).

2.2.3 Incorporating measurements of protein concentration through prior distributions

We have demonstrated that our framework can achieve good performance using non-informative priors. However, the framework could potentially perform better by incorporating prior information when it is available. For instance, Ghaemmaghami
et al. (2003) measured cellular protein concentrations using Western blots in *S. cerevisiae* during log phase growth. As discussed above, although cellular protein concentrations are not precisely equivalent to the transition weights we are estimating, the two still might be expected to loosely correlate with one another. We can therefore use these measurements to construct weak prior distributions for the corresponding DBF transition weights. To account for the loose correlation between the two, as well as experimental measurement error, we use a truncated normal prior for log transition weights with a large standard deviation of 2 (so a standard deviation in each direction corresponds to multiplying the weight by 1/100 or 100, respectively). We calculate the mean for this normal prior by converting measurements from Ghaemmaghami et al. (2003) to molar concentration using a yeast cell volume of $5 \times 10^{-14}$L (Bryan et al., 2010). The resulting prior means are in the range of $-8$ to $-6$ in log scale. Note that nine of the 42 TFs in our model do not have measurements available, and thus their priors remain uniform, as in Section 2.1.6.

When we utilize this prior information, we observe no change in training performance and a marginal increase in testing performance (median $h(\hat{X})$ increases by 0.013; Figure 2.7). Such an insignificant result could arise for multiple reasons: (a) the aforementioned difference between cellular concentration and the model’s transition weights means that the information provided by the measured concentrations might not even be relevant; (b) the noisy physiological measurements of both cellular concentration and cell volume means that the measurements we used might not be quite accurate; or (c) the weak prior we utilized in the model because the measured concentrations are not trusted to be very precise means that the objective function landscape might change only slightly.
Figure 2.7: Comparison of cross-validation performance with and without prior information regarding measured cellular protein concentration. Performance for each promoter is measured by the geometric mean ($h(\hat{X})$) of the two Pearson correlations defined in Figure 2.2. Each boxplot shows the performance summary of the 81 promoter regions across all the cross-validation trials.

2.2.4 MNase sequence preference does not bias the MNase-seq coverage data

Several groups have previously reported that MNase preferentially cleaves at AT dinucleotides. Thus, a concern is that this bias may introduce spurious signals. We examined the severity of this bias using in vitro MNase-seq data. We randomly selected ten 5kb regions in the yeast genome. For each region, we calculated the underlying frequency of each nucleotide (nucleotide composition). We also counted the total MNase-seq fragment coverage and total number of MNase cuts for each nucleotide in each region using in vitro data (on naked DNA) from Deniz et al. (2011). The distribution of total coverage and total cuts across different nucleotides were then calculated (coverage percentage and cut percentage, respectively). The ten sets of values were used to create the boxplot in Figure 2.8. As we can see from the figure, the cut percentage is notably different from (and heavily influenced by) the underlying nucleotide composition, in agreement with previous observations. However, the coverage percentage is roughly the same as the underlying nucleotide composi-
Figure 2.8: Cut percentage is notably different from (and heavily influenced by) the underlying nucleotide composition. However, the coverage percentage is roughly the same as the underlying nucleotide composition, indicating that coverage is much less sensitive to MNase bias. In Appendix Figure A.2, we also showed raw coverage data comparisons between in vivo and in vitro datasets. We can see from that figure that the majority of the in vivo signals cannot be explained by sequence bias. We therefore conclude that in vivo MNase fragment coverage derived from paired-end sequencing (in contrast to MNase cuts derived from paired- or single-end sequencing) is largely unaffected by MNase bias, and thus contains important information that is not simply explainable as MNase bias.

2.2.5 Specific framework parameters do not influence the overall inference results

To ensure that model inference results are not sensitive to the particular model parameters, we tried different value settings for the model parameters, as shown in Table 2.3. In Figure 2.9, we compare the inference results in terms of the inferred
transition weights posterior distribution for several TFs (in Appendix Figure A.3 we showed the same results for all TFs). We found that the inferred DBF transition weights are not influenced by specific values.

2.3 Discussion

We show that integrating information from experimental data within a general framework built on a thermodynamic ensemble model of competitive factor binding can improve the accuracy of inferred protein-DNA interactions, providing a more biologically plausible view of the protein-DNA interaction landscape. Such a landscape gives a mechanistic explanation for observed paired-end MNase-seq fragments through various protein binding events, each with its own probability of occurrence. Many of those binding events are weak binding events that are typically missed in other modeling methods, but are captured in our framework; these weaker binding events are also supported by higher resolution experimental data where available (Rhee and Pugh, 2012). These weak binding events are important: It has been reported that low affinity protein-DNA interactions may be involved in fine-tuning transcriptional regulation and are common along the genome (Tanay, 2006; Biggin, 2011; Segal et al., 2008). Our framework’s predictions agree with this viewpoint: 72% of the binding events in our predicted profiles have a probability lower than 0.5.
Figure 2.9: Comparison of inference results using different settings to pre-process MNase-seq data. MNase-seq data were pre-processed using the seven settings described in the main text and Table 2.3. For each setting, TF transition weights were fitted using the same inference method as described in the main text. 50% posterior credible intervals of each fitted TF transition weight in log scale are shown here. In all these settings, the 50% posterior credible intervals overlap significantly for the TFs shown here. We therefore conclude that the specific parameters we used to pre-process the MNase-seq data do not have a significant impact on inference results. Results for all TFs are shown in Appendix Figure A.3, in which we can observe similar conclusions.

Our framework could thus form an important basis for future computational work that connects transcriptional activity with the protein-DNA interaction landscape.

Our framework does not successfully predict a few TF binding events reported by high resolution ChIP-exo experiments (Rhee and Pugh, 2011), most notably some of the binding sites for Phd1 and Reb1. We believe the primary reason is occasional mismatches between our input TF PWMs and these proteins' actual in vivo DNA-binding specificities. For Phd1, Rhee and Pugh (2012) report several distinct in vivo motifs. However, the Phd1 PWM we used in our framework comes from in vitro data (Zhu et al., 2009) and does not match the in vivo DNA-binding specificity.
of Phd1 reported by Rhee and Pugh (2012). Similarly, for Reb1, Rhee and Pugh (2012) report that 40% of Reb1 binding sites are so-called ‘secondary binding sites’, with motifs that deviate from the TTAGGC consensus of the in vitro PWM we are using. This mismatch in DNA binding specificity may account for much of the discrepancy between our predicted profiles and reported binding sites. However, some caution should be taken when interpreting in vivo ChIP data, since the assay cannot distinguish between direct protein-DNA interaction and indirect interaction (Gordân et al., 2009). We also note that our current framework only includes a subset of all yeast TFs. Some unexplained short fragment coverage peaks, such as those in the YBL014C promoter region, could indicate the binding of DBFs that are not in our set. These and other discrepancies may have an impact on our overall inference, resulting in missing binding events (or possibly even superfluous binding events, because of the competition that is inherent in our model).

In the promoters of YNL157W and YDL012C, our predictions do not include Rap1 binding events even though they are reported in ChIP-exo experiments. However, we believe this results from the nature of Rap1 binding: Lickwar et al. (2012) report that Rap1 binding on non-ribosomal protein promoters, like the two mentioned above, is highly dynamic and involves fast turnover. Such binding events are possibly captured in ChIP experiments because of cross-linking, but may be difficult to observe in an MNase-based digestion experiment if the latter does not involve a cross-linking step. Incidentally, the two ChIP-determined Rap1 binding events are not close to MNase-seq small fragment coverage peaks. One possible use of our framework for extending the results shown here would be to incorporate data from ChIP-based experiments and use the framework to estimate parameters that reflect information from both kinds of data.

We demonstrate the use of prior information in our framework through incorporating measured bulk cellular protein concentration. The model performance im-
proved marginally, which can be interpreted two ways. On the one hand, it is reassuring that one need not have measured cellular protein concentrations in order to perform effective inference. The fact that our uniform priors work as well as having priors informed by measured concentrations means that the measured concentrations available currently are not critical for good performance. However, that said, it is also reassuring that our framework has the ability to incorporate this sort of prior information when available because we anticipate such data will only improve. As measurement technologies enable us to move from bulk cellular concentrations toward nuclear concentrations of active TFs, we anticipate that the ability to incorporate prior information will become more useful, if not for achieving better results then perhaps at least for more rapid convergence toward optima when we move to higher-dimensional inference (e.g., more TFs).

With adequately fitted parameters, our framework has the potential to perform in silico simulation for various environmental conditions by changing the protein concentrations. For example, we could simulate in silico heat shock by increasing the concentration of heat shock response factors in our model. We could also investigate how certain single nucleotide polymorphisms (SNP) affect the overall protein-DNA interaction landscape, not just at the site of the SNP but propagating to the surrounding region due to altered competition.

This work represents a first step toward a more general framework. By specifying probabilistic distributions appropriate for other kinds of experiments—like ChIP-seq, FAIRE-seq, or DNase-seq—the framework can integrate other sources of data through a joint likelihood. As more and larger-scale sequencing projects are carried out, such a framework will prove extremely valuable for integrating different pieces of information to infer a more precise view of the protein-DNA interactions that govern transcriptional regulation.
Mapping genome-wide nucleosomes using DNase-seq data

In the previous chapter, we demonstrated a framework that is able to learn a protein-DNA interaction landscape through fitting a pseudo-likelihood function of MNase-seq data. Another class of nuclease digestion data that exists in high volume is DNase-seq data. But to date, it is mainly used in studying DNase hypersensitive sites (DHS), which are open chromatin regions devoid of nucleosomes. In an effort to incorporate more data, we reveal that DNase-seq is also very informative in mapping nucleosomes genome-wide. We devote this chapter to develop a Bayes-factor–based method to map nucleosomes genome-wide using DNase-seq data.

To develop this method, we first describe and characterize the distinctive DNase I cleavage profile on nucleosome-associated DNA. The features of this profile include an overall quadratic shape, a periodic cleavage rate, and a surprising asymmetry of the cleavage of each strand as it winds along the nucleosome. We show that these features can be built into a Bayes-factor–based nucleosome scoring method to achieve high sensitivity and specificity in distinguishing nucleosomal and non-
nucleosomal genomic regions. Applying this method, we generate the first genome-wide nucleosome maps based on DNase-seq data for both yeast and human. We show that the resulting maps are highly concordant with previous maps (Brogaard et al., 2012; Gaffney et al., 2012). Canonical nucleosome positioning properties, such as highly phased nucleosome arrays around transcription start sites (TSSs) and replication origins, are clearly reflected in our DNase-based maps. The spatial relationships we observe between nucleosomes and bound transcription factors (TFs) (such as Abf1 and Reb1 in yeast and CTCF in human) are also in strong accordance with previous reports. Our method therefore adds nucleosome mapping capabilities to the DNase-seq protocol, enabling it to map both DHSs and nucleosome positions simultaneously, and greatly improving the time- and cost-efficiency of this widely used protocol.

An important feature of our approach is that we exploit the DNase I cuts both within and outside nucleosomes, leveraging all available information, while MNase-based methods primarily rely on MNase cleaving the genome at linker regions to identify nucleosome positions. This enables us to identify the well-known 10.3 bp nucleosome translational position offsets (Gaffney et al., 2012) that MNase is not able to identify. Peaks and troughs in the oscillatory cleavage pattern of DNase I correspond to accessibility of the DNA minor and major groove along the nucleosome. They therefore reveal the rotational positioning of nucleosomes, i.e., the orientation of DNA major and minor grooves on the histone surface. The nucleosome rotational setting at potential TF binding sites has been shown to modulate the binding of TFs (Li and Wrange, 1995; Sekiya et al., 2009; Cui and Zhurkin, 2014). Using this pattern, we systematically studied the rotational context of TF motif matches for 21 yeast TFs and five human TFs. We observe that TF motif matches within nucleosome-associated DNA are often more likely to be located in a consistent manner that aligns with either the minor or major groove. This preferential localization may exist to
regulate the ability of TFs to search for their binding sites along the genome.

The work presented in this chapter is based on our work in Zhong et al. (2015).

3.1 Methods

3.1.1 DNase-seq data

The DNase-seq data generated in this study were derived from a W303 strain of yeast, grown asynchronously in rich medium. Protocols for nucleus isolation, DNase I digestion, and sequencing library preparation are adapted from Henikoff et al. (2011) and Song and Crawford (2010), with minor changes:

1. Intact nuclei were prepared as before (Henikoff et al., 2011) with some alterations:
   - Unfixed yeast cells (asynchronous W303 cells grown in YPD to OD_{600} 0.7) were centrifuged at 2000 rpm for 5 min, washed with sterile water, and resuspended in 20 mL buffer Z (0.56 M sorbitol, 7.4 pH 50 mM Tris, autoclaved);
   - After resuspension, 14 µL β-mercaptoethanol and 500 µL of 10 mg/mL zymolyase dissolved in buffer Z were added. Samples were incubated on the benchtop for 30 min and inverted every few minutes;
   - Cells were then centrifuged at 1500 rpm for 6 min at 4°C and resuspended in 2.5 mL modified NP buffer (1 M sorbitol, 50 mM NaCl, 7.4 pH 10 mM Tris, 5 mM MgCl$_2$) supplemented with 0.5 mM spermidine, 0.007% β-mercaptoethanol, and 0.075% NP-40.

2. Dilutions of DNase I were prepared on ice to determine the best digestion conditions:
- In 1.5 mL tubes, 400 µL of cell mixture from above was added to 12 µL of the following DNase I solutions: 0.03, 0.1, 0.3, and 1 U/µL. Samples were inverted once and incubated at 37°C for 16 min;
- After incubation, 100 µL of stop buffer (5% SDS, 50 mM EDTA) was added to terminate the reaction. Proteinase K (0.2 mg/mL) was added to each tube, and the tubes were then inverted to mix and placed at 65°C overnight;
- The next day, DNA was recovered by phenol extraction and isopropanol precipitation and run on a 0.8% agarose gel for 3 hours at 85 V to check sample digestion;
- The samples that have the optimal amount of DNase I digestion were then prepared for sequencing following the Crawford DNase-seq protocol (Song and Crawford, 2010).

Reads generated by the sequencer are 50 bp each, but only the first 20 bp of each read is genomic DNA because of the MmeI digestion step in the protocol. So only the first 20 bp are used when aligning reads to the genome.

The DNase-seq raw reads from Hesselberth et al. (2009) were obtained from the Sequence Read Archive (SRA, http://www.ncbi.nlm.nih.gov/sra), with accession numbers SRX002990 (in vivo) and SRX003233 (in vitro).

Raw reads from all yeast datasets were mapped to the June 2008 build of the *S. cerevisiae* genome using Bowtie (Langmead et al., 2009). Only uniquely mapped reads are reported. We count the number of reads that are mapped to each genomic location (only the 5’ end of a read is counted, not its full length). Human DNase-seq data were obtained from Degner et al. (2012). We used their mapped reads at http://eqtl.uchicago.edu/dsQTL_data/MAPPED_READS/. All of their DNase-seq data were pooled to increase sequencing depth. The read count at each genomic...
coordinate was then transformed using the inverse hyperbolic sine function ($\text{asinh}$):

$$\text{asinh}(\theta) = \ln(\theta + \sqrt{1 + \theta^2})$$

This transformation has been used by others in modeling genomic read counts (Hoffman et al., 2012): It is quite similar to a log transformation; like a log transformation, it reduces both the variance in the data and the influence of large outliers, but unlike a log transformation, it handles zero values gracefully.

### 3.1.2 Nucleosomal positions and non-nucleosomal positions

The 2,000 nucleosome centers with the highest NCP score-to-noise ratios, as reported by Brogaard et al. (2012), were selected as *bona fide* nucleosome centers. Genomic windows around those nucleosome centers were used for training our models. They were also used as positive examples in the binary classification task.

To create negative examples for the classification task, we uniformly randomly selected 2,000 genomic windows as non-nucleosomal positions. Note that whenever a genomic window of size 147 bp is called nucleosomal in this chapter, it means a nucleosome dyad is positioned at the exact center of that window. So, although nucleosomes cover a large portion of the genome, most genomic positions will not be nucleosome centers. Therefore, our random genomic windows are reasonable approximations of non-nucleosomal windows (error rate $\frac{1}{147}$).

Note that the nucleosomal and non-nucleosomal positions defined here are for yeast. We don’t require *bona fide* nucleosome centers in human as we used parameters trained using yeast data when applying our model to human data.

### 3.1.3 DNase I digestion profile on nucleosomal DNA

To calculate the digestion profile of DNase I on nucleosomal DNA, we extracted the $\text{asinh}$ transformed counts of the 147 bp windows around each of the 2,000 nucleosome centers identified above. The count vectors were stacked to form two matrices of size
2000 × 147, one for each strand. The column averages of the two matrices are visualized in Figure 3.3A.

To isolate the oscillatory pattern, we computed a detrended pattern by smoothing the digestion profile shown in Figure 3.3A using LOWESS (Cleveland, 1979). We then subtracted the smoothed version from the original digestion profile. We call the resultant 147 bp series the detrended oscillation pattern.

3.1.4 Bayesian harmonic regression analysis

We used harmonic regression to analyze the oscillatory cleavage pattern of DNase I on nucleosomal DNA. We describe the method briefly here; for more details, see Prado and West (2010).

For a given strand, we denote its detrended oscillation pattern as:

\[ z_{-73}, z_{-72}, \ldots, z_0, \ldots, z_{72}, z_{73} \]

This series is then modeled using linear regression:

\[ z_t = A \times \cos(\omega t) + B \times \sin(\omega t) + \epsilon_t \quad \epsilon_t \sim N(0, \sigma^2) \quad \forall t \in [-73, -72, \ldots, 0, \ldots, 72, 73] \]

Note that for mathematical convenience, we parameterize the model using \( \omega \), the oscillation frequency of the series, even though the parameter we are interested in estimating is the period, which is \( \frac{2\pi}{\omega} \). We can rewrite the linear regression in matrix form:

\[ Z = F \beta + \epsilon \]

where \( F \) is the design matrix and \( \beta = (A \ B)^T \). For a given \( \omega \) value, we can calculate the maximum likelihood estimate for \( \beta \) as:

\[ \hat{\beta} = (F^T F)^{-1} F^T Z \]
By assuming a reference prior, \( p(\beta, \sigma|\omega) \propto \sigma^{-1} \), we can calculate the (unnormalized) posterior of \( \omega \) as:

\[
p(\omega|Z) \propto |F^T F|^{-1/2} \{1 - \hat{\beta}^T F^T F \hat{\beta}/(Z^T Z)\}^{(2-T)/2} p(\omega)
\]

where \( T \) is the total number of data points in \( Z \). If we assume a uniform prior for the period parameter, we can then calculate the unnormalized posterior density of the period (as shown in Figure 3.3C) using the above through a change of variables, and can use that density to identify the posterior mode of the period.

3.1.5 Bayes-factor–based nucleosome score

Likelihoods and Bayes factor computation

Given a genomic window of size 147 bp, we wish to assess how likely a nucleosome is centered at this window using statistical models. We therefore define two models: a nucleosome model and a background (non-nucleosome) model. We describe the likelihood functions for the two models in the following paragraphs.

Denote the associated \textit{asinh} transformed DNase-seq counts of a 147 bp window as:

Forward strand \( X = x_{-73}, x_{-72}, \ldots, x_0, \ldots, x_{72}, x_{73} \)

Reverse strand \( Y = y_{-73}, y_{-72}, \ldots, y_0, \ldots, y_{72}, y_{73} \)

We model the data using normal distributions, with variance proportional to the mean:

\[
x_t \sim N(\mu_t^{(x)}, k \times \mu_t^{(x)}) \quad y_t \sim N(\mu_t^{(y)}, k \times \mu_t^{(y)}) \quad k > 0
\]

The mean-variance relationship is intrinsic to the data (see Figure 3.1 for evidence).

For the nucleosome model, we parameterize the mean curve as:

\[
\mu_t^{(x)} = \mu_t^{(y)} = c^a + c^b \times t^2 + z_t
\]
Figure 3.1: Mean and variance of DNase-seq counts at each position along the nucleosome are proportional to each other (after inverse hyperbolic sine transformation). Each point corresponds to the mean and variance of one of the 147 positions in the nucleosome window.

where \( t \) is the nucleosome base pair index, \( z_t \) is the observed, detrended oscillation pattern described in Section 3.1.3, and \( a \) and \( b \) are parameters that describe the baseline level of cleavage and the curvature of the curve, respectively. This parameterization captures the main features shown in Figure 3.3A. The likelihood for this nucleosome model is therefore:

\[
L_n(a, b, k|X, Y) = \prod_{t=-73}^{73} \left(2\pi k(e^a + e^b \times t^2 + z_t)\right)^{-1/2} \exp \left(-\frac{(x_t - e^a - e^b \times t^2 - z_t)^2}{2k(e^a + e^b \times t^2 + z_t)}\right) \times \\
\left(2\pi k(e^a + e^b \times t^2 + z_{-t})\right)^{-1/2} \exp \left(-\frac{(y_t - e^a - e^b \times t^2 - z_{-t})^2}{2k(e^a + e^b \times t^2 + z_{-t})}\right)
\]

(3.1)

For the background model, we parameterize the mean curve as a flat line:

\[
\mu_t^{(x)} = \mu_{-t}^{(y)} = e^a
\]
Figure 3.2: DNase I cleavage profiles (average transformed DNase-seq counts) at 2,000 random genomic windows are essentially flat, modulo random sampling noise. This follows from the observation that, when averaged over a large number of randomly selected 147 bp genomic windows, DNase-seq cleavage profiles are essentially flat, modulo random sampling noise (see Figure 3.2). The likelihood for the background model is thus:

\[
L_r(a, k| \mathbf{X}, \mathbf{Y}) = \prod_{t=-73}^{73} \left( 2\pi ke^a \right)^{-\frac{1}{2}} \exp \left( -\frac{(x_t - e^a)^2}{2ke^a} \right) \left( 2\pi ke^a \right)^{-\frac{1}{2}} \exp \left( -\frac{(y_t - e^a)^2}{2ke^a} \right)
\]

(3.2)

The likelihood ratio between the nucleosome model and the background model is a measure of how likely the genomic window is to be a nucleosomal window, and could therefore be used as a nucleosome score. However, individual windows exhibit a significant amount of variation in the DNase-seq data, as with any genomic data. So we instead calculate a Bayes factor that is able to integrate out the uncertainties of the model parameters:

\[
BF = \frac{\int_a L_n(a, \hat{b}, \hat{k}| \mathbf{X}, \mathbf{Y}) p_n(a) da}{\int_a L_r(a, \hat{k}| \mathbf{X}, \mathbf{Y}) p_r(a) da}
\]
Here, we take the uncertainty in the baseline cleavage level into account by integrating out \( a \) in both likelihood functions. The prior distribution for \( a \) in both likelihood functions is normal, with mean and variance hyperparameters determined using an empirical Bayes approach (see next section); other parameters are kept fixed at their maximum likelihood estimates (MLE, see next section). We evaluated our approach in comparison with several alternative approaches and chose it because of its high accuracy and low computational cost (see Section 3.2.8 and Figure 3.24).

Prior distributions and MLE of model parameters

To determine the prior distributions used in calculating Bayes factors, we used an empirical Bayes approach (Carlin and Louis, 1997). For each of the 2,000 nucleosomal positions, we computed the MLE of \( a, b, \) and \( k \) in our likelihood function (3.1):

\[
\hat{a}_i, \hat{b}_i, \hat{k}_i \quad \forall i \in \{1, 2, \ldots, 2000\}
\]

To determine the prior distribution for \( a \) in the nucleosomal model, all 2,000 \( \hat{a}_i \)s were used to fit a normal distribution through maximum likelihood estimation. The resultant normal distribution was used as the prior distribution for \( a \). The prior distribution for \( a \) in the background model was determined similarly, but using background data windows. Any remaining parameters that were not integrated out were fixed at their MLE values, these being calculated by pooling the 2,000 nucleosomal (or background) data windows together.

3.1.6 Mapping nucleosome positions with a greedy algorithm

We wanted to produce a genome-wide map of nucleosome positions, given the moving window nucleosome scores across the genome. To identify the nucleosome centers that would comprise our map, we used the following greedy algorithm, which is quite similar to the one used by Brogaard et al. (2012):
1. Rank the series of nucleosome scores in descending order.

2. The genomic position corresponding to the highest nucleosome score is called a nucleosome center.

3. The nucleosome scores in the 117 bp window centered on the position identified in the previous step are removed from the series. We chose 117 bp instead of 147 bp to allow some overlap between two selected nucleosomes; this can partially mitigate the greediness of the algorithm. We confirmed that the result shown in Section 3.2.3 is largely unchanged for parameter values between 97 bp and 127 bp (see Section 3.2.9).

4. Repeat all above steps until no nucleosome score greater than zero remains in the series.

Applying this algorithm to the whole genome or separately for each chromosome gives identical results, so we did the latter when computing our genome-wide nucleosome position map (to reduce memory usage).

3.1.7 Comparing pairs of nucleosome maps

To compare a given nucleosome map with a reference nucleosome map, we calculated three quantities: the number of nucleosomes shared between the two maps (true positives), the number of nucleosomes that only appear on the reference map (false negatives) and the number of nucleosomes that only appear in the given nucleosome map (false positives). If two nucleosome centers on the two different maps were less than 73 bp away from each other, they were said to be shared between the two maps; otherwise they were either false negatives or false positives. The number of true positives, false negatives, and false positives were reported. In addition, for all nucleosomes shared between two maps, we calculated all the center-to-center
distances between corresponding shared nucleosomes. We used kernel density estimation (with a kernel bandwidth of 1 bp) to visualize the distribution of these distances, as shown in Figure 3.9A.

3.1.8 Identifying TF motif matches with motif scanning

We used TF motifs from MacIsaac et al. (2006) (for yeast TFs) and JASPAR database (Mathelier et al., 2013) (for human TFs). The yeast TFs used in this chapter also have similar motifs from in vitro PBM experiments (Gordân et al., 2011). We defined candidate binding sites by scanning a position weight matrix (PWM) across the genome, using a permissive threshold with PWM score (log-likelihood ratio of seeing a motif-width DNA sequence under the PWM model versus under a fourth-order Markov background sequence model) greater than four.

3.1.9 Calculating nucleosome-associated oscillation around TF motif matches and random genomic sites

In our identified genome-wide map of nucleosome centers, each center is associated with an oscillatory DNase I digestion pattern. For a given TF with \( N \) motif matches, we extracted the nucleosome associated oscillatory patterns occurred within \( \pm 73 \) bp of each motif match center (a 147 bp window). We then calculated the average oscillation across the \( N \) matches to obtain the “motif oscillation”. As a control, we randomly selected \( N \) genomic sites and carried out the same calculation to obtain the “random oscillation”. To assess the significance of the motif oscillation amplitude (the amplitude at the motif match, i.e., the amplitude inside the red dashed lines of Figure 3.21), we computed 100 random oscillations and calculated their average amplitude. The ratio between the motif oscillation amplitude and the average random oscillation amplitude is calculated as a measure of how significant the motif oscillation deviates from the random oscillation.
3.2 Results

In the sections that follow, we first use DNase-seq data in yeast to develop our model, and then apply our model to produce a genome-wide nucleosome map for the yeast genome. After validating the quality of our DNase-based map, we consider the required sequencing depth needed to achieve similar quality in larger genomes like human. We then pool data from existing DNase-seq datasets (see Methods) and use the pooled data to produce a nucleosome map for the human genome.

3.2.1 DNase I cleavage shows a distinctive oscillatory profile along the nucleosome

Figure 3.3 shows the cleavage profile calculated in Section 3.1.3. In the cleavage profile, we see an overall quadratic shape, indicating that DNA nearer the nucleosome dyad is better protected from DNase I cleavage than DNA nearer the edge of the nucleosome. This is most likely due to dynamic nucleosome wrapping and un-wrapping (DNA breathing) (Li and Widom, 2004; Li et al., 2005). Overlaid on this quadratic shape, we also observe a signature oscillatory cleavage pattern, which has been previously well-established (Noll, 1974; Boyle et al., 2008; Winter et al., 2013). Since DNase I binds within the minor groove, it can more easily nick nucleosomal DNA when the minor groove is exposed. A harmonic regression analysis shows that the period of this oscillation is roughly 10.3 bp for our data (Figure 3.3C), a value that agree with earlier studies and with the periodic exposure of the minor groove along the nucleosome. Comparing the cleavage profiles for the two different strands, we see a 2–3 bp offset in the periodic patterns. This is likely due to the fact that DNase I nicks one strand at a time in the presence of Mg$^{2+}$, and the active site of this nicking activity is not quite centered in the enzyme (Suck et al., 1988), resulting in a 2–3 bp offset between nicks on opposite strands (Cousins et al., 2004; Boyle et al., 2008).
Figure 3.3: (A) Strand-specific cleavage profile of DNase I along the nucleosome, computed by averaging DNase-seq counts (transformed by the inverse hyperbolic sine function) within the 2,000 most strongly positioned nucleosomes in the yeast genome. (B) Crystal structure of a nucleosome shown from two angles; which strand faces outward as the minor groove becomes accessible differs on opposite sides of the nucleosome dyad. The two positions labeled with arrows have the same relative distance to the dyad. Left: the blue strand faces outwards and is more exposed to digestion at this position while the red strand faces inwards and is less exposed. Right: the red strand is now the one that faces outwards while the blue strand faces inwards. In other words, on opposite sides of the nucleosome dyad, each strand is exposed differently as the minor groove becomes accessible (more exposed upstream of the dyad and less exposed downstream of the dyad). (C) Posterior density of the period of oscillation, as determined by Bayesian harmonic regression. The most probable period a posteriori for each of the two different strands is around 10.3 bp.

We note two other interesting and important features in the cleavage profile. First, for each strand, the rate of cleavage is asymmetric across the nucleosome dyad. Specifically, the oscillatory pattern is strong upstream of the dyad (on the 5′ side) but is markedly dampened downstream (on the 3′ side). Second, however, the cleavage profiles of the two strands are almost exact mirror images of each other, as would be expected (Figure 3.4). We believe the within-strand asymmetry across the dyad arises from the way DNase I can interact with each strand given the specific
Figure 3.4: DNase I cleavage profiles along the nucleosome for the Watson and Crick strands are almost exactly mirror symmetric with each other. This figure is similar to Figure 3.3A, but with the Crick strand profile flipped around the line $x = 0$.

3-dimensional structure of nucleosome-associated DNA. From the crystal structure of the nucleosome (Figure 3.3B) (Luger et al., 1997), we see that upstream ($5'$) of the dyad, the minor groove of each strand faces outward from the histone octamer, but downstream ($3'$) of the dyad, access to the minor groove is somewhat impeded by the previous wrap of the DNA around the histone octamer. This asymmetry was first observed and explained by Lutter (1978), who reasoned (before the structure of the nucleosome was determined) that the asymmetry along each strand is the direct consequence of DNA wrapping around the histone core in a left-hand manner.

To ensure the robustness of our results, we repeated this same analysis using DNase-seq data from yeast nuclei published by Hesselberth et al. (2009); the Hesselberth DNase-seq data exhibit the same profile we observe in our own data (Figure 3.5). The period is 10.4 bp for the Hesselberth data (Figure 3.6), which is similar to our data.

It is known that nucleosomes exhibit weak periodic sequence preferences that enable them to wrap DNA more effectively (Satchwell et al., 1986; Segal et al., 2006).
These weak sequence preferences are thus correlated with the periodic exposure of the minor groove along the DNA, leading an alternative explanation of the oscillatory cleavage pattern: that it instead arises from periodic sequence patterns across large stretches the genome coupled with sequence bias in the cleavage preferences of DNase I. To explore this possibility, we took DNase-seq data generated from naked yeast DNA, entirely devoid of nucleosomes, and repeated our analysis with this dataset (collected by Hesselberth et al. (2009); results shown in Figure 3.5). The distinctive features of the DNase I cleavage profiles are completely absent in this dataset, so we rule out the possibility that they arise from DNase I sequence bias acting on periodic sequence patterns. We thus conclude they are best explained by the periodic exposure of the minor groove that occurs in nucleosome-associated DNA.

### 3.2.2 Distinctive DNase I cleavage profile allows nucleosome positions to be distinguished from non-nucleosome positions

We hypothesized that the quadratic, oscillatory, and within-strand–asymmetric DNase I cleavage profile would be very informative and specific for identifying nucleosome positions along the genome. To test this hypothesis, we first explored the possibility of utilizing the profile to distinguish nucleosomal from non-nucleosomal genomic positions in a classification setting. We built major features of the profile into a Bayes-factor–based nucleosome scoring method (see Methods). Using the 2,000 true positive nucleosome center and 2,000 true negative non-nucleosome centers defined in Section 3.1.2, we carried out a 10-fold cross-validation in which both the nucleosomal and non-nucleosomal windows were randomly split into 10 equal partitions. Model parameters were trained on nine of the partitions using an empirical Bayes approach, and each model so trained was then used to classify windows from the remaining partition as being nucleosomal or non-nucleosomal. All of the test cases
Figure 3.5: Strand-specific cleavage profile of DNase I along the nucleosome, computed by averaging DNase-seq counts in the Hesselberth data (transformed by the inverse hyperbolic sine function) within the 2,000 most strongly positioned nucleosomes in the yeast genome. The same average is calculated on both in vivo (green and orange) and in vitro (purple and red) data. The distinct DNase I cleavage profile observed in the in vivo data is completely absent in the in vitro data. We observe that the DNase I digestion pattern near the dyad is slightly noisier in the Hesselberth data (both in vivo and in vitro) than we observe in our data, perhaps due to slight differences in the protocols.

from across the ten folds were combined to compute a receiver operating characteristic (ROC) curve; area under the ROC (AUROC) was used to assess classification performance (Figure 3.7). Our classifier achieves a good combination of sensitivity and specificity in distinguishing nucleosomal from non-nucleosomal windows (out-of-sample AUROC is 0.89 on our data, and for comparison, 0.85 on Hesselberth data). The data we publish here provide mildly better discriminatory power compared to the Hesselberth data, particularly on less well-positioned nucleosomes (the two ROCs in Figure 3.7 overlap at first, but then diverge beyond a false positive rate around 5%). This difference might be because the Hesselberth data have a lower sequencing depth and/or because our data exhibit a less noisy nucleosomal cleavage profile (Figure 3.3A vs. Figure 3.5). In the following sections, we only present results using
Figure 3.6: Posterior density of the period of oscillation in the Hesselberth data, as determined by Bayesian harmonic regression. The most probable period \( a \text{ posteriori} \) for each of the two different strands is around 10.4 bp.

our data (parallel analyses conducted with Hesselberth data exhibit the same general properties).

3.2.3 Distinctive DNase I cleavage profile allows nucleosome positions to be mapped genome-wide

Encouraged by the performance of the DNase I cleavage profile when used as a binary classifier, we investigated whether we could map nucleosome positions genome-wide using such a profile. We trained model parameters using all of the nucleosomes in the top 2,000 set above, and then calculated a nucleosome score at every position genome-wide, using a moving 147 bp window. Figure 3.8 shows an example region from Chromosome IX in yeast, where we plot the raw and transformed DNase-seq counts, and then the smoothed moving window nucleosome score computed from these data. The figure also compares our nucleosome score with the NCP score-to-noise ratios reported by Brogaard et al. (2012). Our Bayes-factor–based nucleosome score peaks accord well with NCP score-to-noise ratio peaks, even when the DNase-
Figure 3.7: Classification ROC for 10-fold cross-validation on both our data (green) and the data of Hesselberth and colleagues (orange). All test cases from the ten folds were combined to draw an overall ROC for each dataset. The areas under the two ROCs are computed and presented as a bar chart (inset).

seq data seem visually to exhibit only a weak positioning signal. This shows the power of using a Bayesian method that not only integrates a weak signal across multiple genomic positions, but also integrates out uncertainty in model parameters.

To further explore the validity of these scores, we used a greedy algorithm (see Methods) to select nucleosome center positions across the genome, and thereby compute a genome-wide nucleosome map, derived entirely from DNase-seq data. Using the chemical-cleavage map from Brogaard et al. (2012) as a reference, we calculated the nucleosome center-to-center distances from our nucleosomes to those of the reference (see Methods). We also counted the number of consensus nucleosomes between our map and the reference map (true positives), as well as the number of nucleosomes in the reference that do not appear in our map (false negatives) and the number of nucleosomes that appear in our map but not in the reference (false positives). As a
Figure 3.8: Example genomic region in which nucleosome positions are mapped using our moving window nucleosome scoring approach. (A) Raw DNase-seq counts in this region. Note that DNase-seq analysis has traditionally focused only on finding and exploring DHS regions, such as the one that corresponds to the strong peak of signal just to the left of coordinate 61,000 (promoter of ESL1). (B) Transforming the raw DNase-seq counts using an inverse hyperbolic sine function allows clearer (but still weak) patterns to be seen in nucleosome-associated DNA. (C) Smoothed moving window nucleosome score on this region (green), in comparison with the NCP score-to-noise ratios from Brogaard et al. (2012) (orange).

In comparison, we performed the same calculation for an MNase-seq–based nucleosome map from Jiang and Pugh (2009). Note that we used their “consensus set”, which itself is compiled as a consensus of four separate MNase-seq datasets (Jiang and Pugh, 2009; Mavrich et al., 2008; Field et al., 2008; Shivaswamy et al., 2008). Figure 3.9A shows a density estimation of the center-to-center distance of the different maps.
FIGURE 3.9: (A) Distribution of center-to-center distances between our DNase-seq–based nucleosome map and the Brogaard reference map (green), and between the consensus MNase-seq–based nucleosome map and the Brogaard reference map (blue). (B) The number of true positives (TP), false negatives (FN), and false positives (FP) of our DNase-seq–based nucleosome map and the consensus MNase-seq–based nucleosome map. The numbers of TP, FN, and FP are defined using the Brogaard nucleosome map as a reference (gold standard).

Overall, our DNase-seq–based map achieved a precision similar to MNase-seq–based maps. However, a notable 10.3 bp fluctuation is apparent in the density estimate of only the map based on DNase-seq. This indicates that if a DNase-based nucleosome center does not overlap with a reference nucleosome center, it is far more likely to be multiples of 10.3 bp away from it. Many have reported that nucleosomes exhibit this translational positioning offset property, in which a nucleosome is likely
to position itself at “rotationally in-phase” positions that are multiples of 10.3 bp away from each other (Gaffney et al., 2012; Albert et al., 2007; Brogaard et al., 2012; Winter et al., 2013). We believe that our DNase-seq-based approach identifies genuine nucleosome centers, and where the map does not perfectly coincide with the reference, the differences appear to represent translational offsets with respect to reference nucleosome centers. MNase-seq-based approaches, on the other hand, are not able to identify the precise translational offsets associated with alternative nucleosome positions. Furthermore, our DNase-seq-based nucleosome map has both a higher sensitivity and specificity than the MNase-seq-based map (Figure 3.9B), even when the latter is compiled as the consensus of multiple datasets (and each individual dataset performs worse than the consensus: see Figures 3.10 and 3.11). The superior ability of our approach to produce an accurate nucleosome map is most likely due to the fact that it uses DNase I cut information both within and outside the nucleosome. The oscillatory cut information can be maintained even at fuzzy nucleosomes because of the stability of nucleosome rotational positioning (Gaffney et al., 2012; Winter et al., 2013). However, MNase-seq methods rely primarily on the digestion signals from within nucleosome linkers, which are weaker when nucleosomes are not well positioned.

3.2.4 DNase-seq–based map recapitulates known features of nucleosome positioning

We confirmed that our DNase-seq–based nucleosome map recapitulates well-known features of nucleosome positioning genome-wide. Specifically, the DNase-seq–derived composite nucleosome positioning patterns around transcription start sites (TSS), around the ARS consensus sequences (ACS) that mark origins of replication, and around TF binding sites are essentially identical to previous reports using other methods (Figure 3.12).

Around the TSS (Figure 3.12A), we see the +1 nucleosome is the most strongly
Figure 3.10: Similar to Figure 3.9B. Using the nucleosome map from Brogaard et al. (2012) as a reference, we calculate the number of true positives (TP), false negatives (FN), and false positives (FP) of our DNase-seq-based map, the consensus MNase-seq-based map, and each of the four individual MNase-seq-based maps. The consensus MNase-seq-based map is compiled by Jiang and Pugh (2009) based on the four individual MNase-seq-based maps. Set 1 is from Mavrich et al. (2008). Set 2 is from Field et al. (2008). Set 3 is from Jiang and Pugh (2009). Set 4 is from Shivaswamy et al. (2008).

positioned, while subsequent downstream nucleosomes become progressively weaker as one moves further into the gene body (consistent with the barrier model). We also observe a strong depletion of nucleosomes immediately upstream of the TSS, in a location called the nucleosome free region (NFR). The spacing of nucleosomes upstream of the NFR is somewhat less regular, and again becomes progressively weaker as one moves further upstream. These patterns all agree with previous reports (for example, Brogaard et al. (2012) and Mavrich et al. (2008)). Similarly, we observe
Figure 3.11: Similar to Figure 3.9A. Using the nucleosome map from Brogaard et al. (2012) as a reference, we calculate the distribution of center-to-center distances between our DNase-seq–based nucleosome map and the Brogaard reference map (green), and between the consensus MNase-seq–based nucleosome map and the Brogaard reference map (blue), as well as between each of MNase-seq–based nucleosome maps and the Brogaard reference map (orange, red, black, and purple).

strong and regular nucleosome positioning around replication origins (Figure 3.12B). The ACS site is known to be closer to the upstream nucleosome, with the upstream and downstream nucleosome centers at –90 and 160 bp relative to the origin, respectively. Our observation accords well with these previous reports that use MNase-seq data (Eaton et al., 2010). The composite nucleosome score patterns can also be seen at individual genomic locus (Figure 3.13).

Figure 3.12C and Figure 3.12D shows median nucleosome scores around Abf1 and Reb1 binding sites, along with the mean NCP score-to-noise ratios from Brogaard et al. (2012) for comparison. Abf1 and Reb1 can both act as barriers for nucleosome
Figure 3.12: Median nucleosome score around all mRNA transcription start sites (A), around origin of replication ACS sites (B), and around binding sites of Abf1 (C) and Reb1 (D) across the yeast genome. TSS coordinates are taken from Rhee and Pugh (2012), ACS coordinates are taken from Eaton et al. (2010), and TF binding site coordinates are taken from MacIsaac et al. (2006). The bottom of each panel shows the average NCP score-to-noise ratios from Brogaard et al. (2012) at corresponding positions.
positioning, and we see regularly positioned nucleosome arrays both upstream and downstream of their binding sites, in close correspondence with the scores from Brogaard et al. (2012). Note that in both our DNase-seq–based map and the Brogaard map, we observe weak nucleosome center signals very close to the TF binding sites. Although it is possible that some of these TFs are bound to nucleosome-associated DNA, it seems more likely that the binding of a TF and the binding of a nucleosome at overlapping genomic locations are mutually exclusive events in each individual cell, but that the data represent a mixture of different binding configurations across

**Figure 3.13:** Additional example regions showing nucleosome scores around TSSs (top panels) and ACSs (bottom panels). The composite score pattern shown in Figure 3.12 is also observed at individual genomic loci.
the population of cells in the assay. Again, the composite score patterns can also be seen at individual genomic locus (Figure 3.14)

3.2.5 Sequencing depth influences DNase-seq–based mapping of nucleosomes

We have demonstrated that we can use DNase-seq data to accurately map nucleosome positions genome-wide in yeast. Yeast has a relatively small genome that can be easily sequenced to high depth. We were interested to explore how well our approach might scale to larger genomes, so we started by evaluating how sequencing depth influenced the performance of our method in the classification setting. To this end, we
uniformly subsampled 80%, 60%, 40%, 20%, 5%, and 1% of our DNase-seq data and performed the same classification as in Section 3.2.2 on those subsampled datasets. Figure 3.15 shows the classification performance as a function of the subsample percentage. Since our DNase-seq counts scale with the number of reads rather than the number of sequenced nucleotides, the ratio of the number of reads to the overall size of the genome (reads/nt) is the appropriate measure of sequencing depth for our purposes. The full dataset we report here consists of 83,053,784 reads across the yeast genome, which corresponds to 6.9 reads per genomic nucleotide. As Figure 3.15 indicates, a sequencing depth of 1.4 reads/nt (corresponding to a 20% subsample) still achieves a good classification performance (AUROC > 0.85). However, further subsampling of the data starts to decrease the performance more dramatically.

Most currently available human DNase-seq datasets have sequencing depth around 0.05 reads/nt (Boyle et al., 2008; Degner et al., 2012; Winter et al., 2013). However,
certain datasets, for example Degner et al. (2012), have multiple replicates from the same cell type that can be pooled together to increase the overall sequencing depth. In the following section, we use the data from Degner et al. (2012) to demonstrate the application of our method across the human genome.

3.2.6 Mapping nucleosome positions on human genome

Degner et al. (2012) performed DNase I sequencing in 70 Yoruba lymphoblastoid cell lines. We pooled all their DNase-seq datasets together (overall sequencing depth ~0.9 reads/nt) and applied our method in this combined human dataset. In this application, we used the model parameters trained on our yeast DNase-seq data, so did not require any prior knowledge about human nucleosome positioning for training. We compare our results to those derived from the MNase-seq data reported by Gaffney et al. (2012) (paired-end MNase-seq on lymphoblastoid cell lines derived from seven Yoruba individuals). Figure 3.16 shows the comparison in one example region from Chromosome 12. As before, we used a greedy algorithm to identify nucleosome centers from both our nucleosome score and the paired-end MNase-seq fragment middle point counts of Gaffney et al. (2012). More than 70% of the nucleosomes identified from DNase-seq data overlap significantly with the nucleosomes identified from MNase-seq data (i.e., two nucleosome centers are less than 73 bp away from each other, see Methods; Figure 3.18B).

Figure 3.16 also compares DARNS from Winter et al. (2013) with nucleosomes identified from DNase-seq and MNase-seq. DARNS are regions over which nucleosomes maintain their rotational positioning, and by design, usually represent only part of a nucleosome (Figure 3.17 shows the length distribution of all DARNS across the genome, as reported by Winter et al. (2013)).

To compare the MNase-seq and DNase-seq nucleosome maps more globally, we calculated the center-to-center distances between nucleosome centers identified by
Figure 3.16: Comparison of DNase-seq–identified nucleosomes and MNase-seq–identified
nucleosomes in human Chromosome 12 (intronic region of CACNA1C). Nucleosome centers
are identified by the same greedy algorithm on our nucleosome score curve and the MNase-
seq fragment middle point counts, respectively. Nucleosomes identified from the two types
of data overlap significantly, especially considering the large noise in the MNase-seq data.
The middle panel also shows the DARNS from Winter et al. (2013), which likely represent
portions of phased nucleosomes.

these two methods. The low sequencing depth of the human DNase-seq data meant
that large stretches of the genome had almost no cuts. For efficiency, these regions
were not processed, and all comparisons below are on those regions where sufficient
DNase-seq depth was present to identify nucleosomes. Figure 3.18A shows the dis-
tribution of these center-to-center distances, and we observe that when predicted
center positions do not agree, they are more likely to be translational offsets of each
other that are rotationally in phase (multiples of 10.3 bp away from each other). In
Figure 3.18B, we display numbers of overlapping and non-overlapping nucleosomes
Figure 3.17: The distribution of DARNS lengths, taken from Winter et al. (2013). Note that the vast majority of DARNS are shorter than 100 bp.

between the two maps, calculated the same way as the results shown in Figure 3.9. We see that we are able to identify nucleosome positions with good precision (but as mentioned above, with low recall; we expect the recall will improve markedly with greater sequencing depth, as was the case in the yeast genome).

Additional genome-wide nucleosome positioning features, including canonically positioned nucleosomes around TSSs and TF binding sites, are also recapitulated by our DNase-seq–derived nucleosome map (Figure 3.19).

Using MNase-seq–identified nucleosome centers as an approximate ground truth, we were able to compute a DNase I cleavage profile in human and see an oscillation profile largely similar to the one we observed in yeast using the nucleosome centers of Brogaard et al. (2012) (Figure 3.20).
Figure 3.18: (A) Distribution of center-to-center distances between our DNase-seq–based nucleosome map and the MNase-seq–based map of Gaffney et al. (2012). (B) The number of nucleosomes identified in both DNase-seq and MNase-seq maps, as well as the number of nucleosomes that only appear in one of the maps.

Figure 3.19: Median nucleosome score around human TSSs (A) and CTCF binding sites (B).
Figure 3.20: DNase I digestion profiles on human nucleosomal DNA (black) and yeast nucleosomal DNA (red). The digestion profiles shown in the figure are an average of the forward and reverse oscillation profiles.

3.2.7 TF motif matches within nucleosomal DNA are often located preferentially on major or minor groove

The oscillatory DNase I cleavage pattern provides information about nucleosome rotational positioning (the orientation of the major and minor grooves of DNA with respect to the histone surface as it wraps around the histone core). Nucleosome rotational positioning has been shown to be able to regulate the binding of TFs to sites along the DNA (Li and Wrangel, 1995; Sekiya et al., 2009; Cui and Zhurkin, 2014). Here, we used this oscillatory pattern (detrended, see Methods) to examine the nucleosome rotational positioning around TF motif matches of 21 yeast TFs that bind DNA directly in rich medium (Gordân et al., 2009) and 5 human TFs with well-defined motifs that have been shown to be pioneer factors (Sherwood et al., 2014).

For a TF with $N$ motif matches, we calculated the average nucleosome-associated oscillation around these $N$ motif matches; as a control, we calculated the average os-
cillation around \( N \) randomly selected genomic sites (we call these “motif oscillation” and “random oscillation”, respectively; see Methods for more details). The peaks and troughs of the oscillation correspond to the minor and major grooves of nucleosomal DNA, respectively. Since randomly selected genomic sites are equally likely to lie anywhere along a helical twist of DNA, their oscillatory peaks and troughs should largely cancel, resulting in a low-amplitude oscillation. However, the motif oscillations of many TFs have significantly higher amplitude than random oscillations (Figure 3.21 shows four examples: Abf1 and Reb1 in yeast, and CTCF and GABPA in human). This indicates that TF motif matches seem to locate preferentially with respect to the rotational phasing of the DNA along the nucleosome.

We then placed these 26 TF motifs along a scale from minor-groove–associated to major-groove–associated according to the distance between motif center and the nearest peak or trough (Figure 3.22). Notably, most TF motif matches are strongly enriched to be centered close to either the major or the minor groove. A similar calculation on randomly generated GC or AT rich motifs suggests that such preferential localization may at least partly be due to the coincidence between motif sequence composition and nucleosome sequence preference at DNA major and minor grooves (Figure 3.23) (Satchwell et al., 1986; Segal et al., 2006). Regardless of the reason, this result indicates that nucleosome rotational positioning is tightly coupled with the sequence preferences of many TFs.

3.2.8 Alternative approaches for computing nucleosome scores

Our current approach for calculating nucleosome scores (Approach 4 below; see Section 3.1.5) was evaluated and selected from the following alternative approaches for computing nucleosome scores, reflecting different levels of model complexity:

1. The mean curve for the nucleosome model is parameterized as \( e^a + e^b \times t^2 \) (without the oscillatory pattern) and nucleosome score is simply the likelihood
Figure 3.21: Average oscillations around motif matches of Abf1 and Mcm1 (yeast), and CTCF and GABPA (human) (“motif oscillation”, black). “Random oscillations” (blue) are average oscillations around randomly chosen genomic sites. One hundred random oscillations are calculated and shown for each TF. Red dashed lines indicate the boundaries and centers of the TF motifs. For many TFs, including those shown here, their “motif oscillations” have significantly higher amplitude than the corresponding “random oscillations”, indicating TF motif matches locate preferentially with respect to the rotational phasing of the DNA along the nucleosome.
Figure 3.22: Most TF motifs are either major- or minor-groove–associated. Twenty-one yeast TFs are shown as blue dots. Five human TFs are shown as red triangles. For each TF, the distance between its motif center to the nearest peak of its motif oscillation is calculated and converted to a “oscillation phase” within $[-\frac{\pi}{4}, \frac{\pi}{4}]$: if the motif center is located at the peak of the composite oscillation and thus tends to be centered on an exposed minor groove, its phase is $-\frac{\pi}{4}$; conversely, if the motif center is located at a trough and thus centered on a major groove, its phase is $\frac{\pi}{4}$. The y-axis shows the ratio between each TF’s motif oscillation amplitude and the amplitude of random oscillations. This is a measure of the significance of the preferential localization. Most TFs have a phase $\geq \frac{\pi}{4}$ or $\leq -\frac{\pi}{4}$, and a ratio $> 3$, indicating they are significantly associated with the major or minor groove, respectively.

1. Ratio between the nucleosome model and the background model (no Bayes factor).

2. Similar to Approach 1, but the nucleosome score is a Bayes factor in which the level parameters $a$ of both likelihood functions are integrated out.

3. Similar to Approach 2, but the curvature parameter $b$ of the nucleosome model
Figure 3.23: We randomly generated five GC (or AT) rich motifs, each with a length of 13 bp (the length of the Abf1 motif). We pooled together all motif matches for these five motifs, filtering out overlapping matches, and then randomly selected 2,600 motif matches from this pool (2,600 being the number of motif matches for Abf1). We calculated and plotted the average oscillation around these motif matches as one curve above. We then repeated this entire process 50 times, resulting in 50 curves for the GC (green) and AT (orange) rich motifs. The anti-correlated curves are consistent with nucleosome sequence preferences for GC and AT nucleotides, which are known to oscillate out of phase with one another.

is also integrated out.

4. Similar to Approach 2, but the oscillatory pattern series ($z_t$) is added to the mean curve of the nucleosome model.

5. Similar to Approach 3, but the oscillatory pattern series ($z_t$) is added to the mean curve of the nucleosome model.

Approach 1 has the lowest computational cost because it does not require numerical integration; conversely, Approaches 3 and 5 have the highest computational cost.
because they require numerical integration in two dimensions. The performance of different approaches in distinguishing nucleosomal from non-nucleosomal windows (as measured by AUROC) is shown in Figure 3.24. Based on these results, we conclude that the simple likelihood ratio (Approach 1) performs significantly worse than all of the Bayes-factor–based approaches (Approaches 2–5). Among the Bayes-factor–based approaches, those that integrate out both $a$ and $b$ do not perform markedly better than those that integrate out only $a$, despite requiring significantly more computation time. Therefore, we decided to use an approach that only integrates out $a$. Finally, adding the oscillation pattern $z_t$ provides better performance, and this improved performance arises without incurring additional computation time, so we included it in our final approach: Approach 4.

3.2.9 Results are insensitive to greedy algorithm parameters

To explore how sensitive our greedy algorithm introduced in Section 3.1.6 might be to various settings of the overlap parameter, we tested parameter values different from 117 bp. The same false negative (FN), true positive (TP), and false positive (FP) values as in Figure 3.9B were calculated and are shown in Figure 3.25. We can see that the results were largely insensitive for a range of reasonable parameter values between 97 bp and 127 bp.

3.3 Data and code access

The DNase-seq data generated in this study have been deposited at Gene Expression Omnibus (GEO) under accession number GSE69651. Our computational tools have been released as a Python package called NucID (Nucleosome Identification using DNase), available on GitHub at:

https://jianlingzhong.github.io/NucID/
Figure 3.24: Performance (as measured by AUROC) of different classification approaches on both our data and Hesselberth data. Classifying nucleosomal vs. non-nucleosomal locations on the basis of a simple likelihood ratio has the worst performance. Bayes factor approaches that integrate out both the level parameter $a$ and the curvature parameter $b$ perform similarly to approaches that only integrate out $a$. Approaches that include the oscillation pattern have better performance than approaches without the oscillation pattern.

Pre-computed tracks of genome-wide nucleosome scores are available at:

http://trackhub.genome.duke.edu/harteminklab/NucID/

Links to all of these resources is also made available from a single location on the web at:

http://www.cs.duke.edu/~amink/software/
Figure 3.25: False negative (FN), true positive (TP), and false positive (FP) calculated as in Figure 3.9B using different greedy algorithm parameters. Performance of the consensus MNase-seq–based nucleosome map is shown in the last column, for reference.

3.4 Discussion

DNase-seq data have primarily been used to date to discover DHS sites, but here we develop a Bayes-factor–based method that uses DNase-seq data to map nucleosome positions, and can do so along the whole genome. DNase I cleavage of nucleosomal DNA produces a distinctive within-strand–asymmetric oscillatory profile reflective of the nucleosome structure rather than the sequence bias of DNase I. We were able to model this profile and use a Bayes factor as a nucleosome score to build a highly sensitive and specific nucleosome position classifier. This same nucleosome score allowed us to derive a genome-wide nucleosome map in yeast, with similar precision to maps based on MNase digestion but greater accuracy, exhibiting fewer false positives and false negatives. Our method explicitly models the oscillatory DNase I cutting pattern within nucleosomal DNA, which is often maintained even at
fuzzy nucleosomes because of their rotational positioning stability. This enables us to identify alternative nucleosome translational positioning offsets that other methods cannot. Our maps recapitulate canonical associations between nucleosome positions and other genomic features, such as TSS, ACS, and TF binding sites.

Genomic data are in general quite noisy. This is evident in the visually weak nucleosome positioning signals shown in Figure 3.8. Still, our method is able to effectively extract a nucleosome positioning signal from this noisy data by combining the following modeling strategies: (1) data for all base pairs across the entire nucleosome window are jointly modeled to leverage all relevant information; (2) the nucleosome-structure-specific oscillatory pattern is incorporated in our model, increasing its specificity; (3) the two genomic strands are modeled to share a mirror-symmetric profile, reducing the number of parameters in the model, and then the use of strand-specific data further increases the specificity of the model; (4) data are transformed using a log-like \texttt{asinh} function to decrease their variance; and (5) we adopt a Bayesian approach to integrate out remaining uncertainty in the model parameters. The last point is particularly important when modeling noisy data.

Our method is also applicable to other organisms, and we applied it to human DNase-seq data to produce a genome-wide map of nucleosome positions in human. Owing to the lower sequencing depth, large stretches of the genome have insufficient reads to make any determination of nucleosome position, so we filtered these regions out, resulting in lower recall. However, in regions where the number of reads is sufficient, we have good accuracy, based on the analysis of Sections 3.2.5 and 3.2.6. We observe reasonable alignment with nucleosome positions determined from the MNase-seq data of Gaffney et al. (2012), considering the low DNase-seq sequencing depth, the variability of nucleosome positions across human cell lines (Radman-Livaja and Rando, 2010), the use of a model trained on yeast data, and the fact that Gaffney et al. (2012) studied seven lymphoblastoid cell lines while the DNase-seq data from
Degner et al. (2012) are derived from 70 Yoruba lymphoblastoid cell lines.

The oscillatory pattern utilized by our method reveals important insights about minor and major groove accessibility of nucleosome-associated DNA. Many studies have shown that nucleosome rotational positioning can regulate the binding of TFs to sites along the DNA (Li and Wrange, 1995; Sekiya et al., 2009; Cui and Zhurkin, 2014), and we observed that for many TFs, their motif matches within nucleosomes tend to center near either the major or minor groove of DNA. We expect several factors are coupled, including the sequence preferences of TFs, the sequence preferences of nucleosomes as they contact major and minor grooves of DNA, as well as the structural exposure of major and minor grooves along the nucleosome. Such coupling may contribute to the complex regulation of TF binding when TFs and nucleosomes compete with each other for binding sites. For example, certain TFs that are able to bind their target sites inside a nucleosome may act as pioneer factors to open chromatin and facilitate the binding of other TFs.

The method we develop here adds nucleosome mapping capabilities to the already widely used DNase-seq protocol, a standard tool for studying genomic regulatory elements in many organisms, including those in the ENCODE and modENCODE projects. Since nucleosome structure is well conserved across eukaryotic species, our method is readily applicable to any DNase-seq dataset with sufficient sequencing depth, as we demonstrated using human DNase-seq data. The previous work of Winter et al. (2013) showed that DNase-seq has the ability to identify nucleosome rotational positioning, and numerous other studies have shown that DNase-seq can be used to identify and study DHSs and TF binding sites (Luo and Hartemink, 2013). Those results, together with our results here, show that DNase-seq is a highly time- and cost- efficient protocol that is able to map nucleosomes and DHSs simultaneously. Our method provides a basis for mining additional insights about nucleosome binding from already available DNase-seq datasets, as well as future ones. The method can
also be incorporated into a more general framework for inferring a whole-genome protein-DNA interaction landscape (Zhong et al., 2014), which includes the binding of both nucleosomes and transcription factors.
A multivariate state-space model for integrating genomic data

In Chapter 2, we introduced a modeling framework to learn protein-DNA interaction landscapes through optimizing a pseudo-likelihood function of MNase-seq data. We showed that we were able to learn a protein-DNA interaction landscape that reasonably explained the experimental observed MNase-seq data. In Chapter 3, we developed a Bayes-factor–based method to map nucleosomes genome-wide using DNase-seq data. Both chapters are modeling one specific kind of data. The framework in Chapter 2 does not model MNase-seq data in a direct, principled way and Chapter 3 only considers nucleosomes. In this chapter, we will develop a principled modeling framework that is able to model different kinds of genomic data. We will also provide an efficient expectation-maximization (EM) model learning algorithm so that the model can potentially infer protein-DNA interaction landscape on a genome-wide scale with all known DBFs.

We will use COMPETE as a starting point for building our model. We first describe the connection between COMPETE and a state-space model (SSM) with a proper
likelihood function. We will then present detailed model formulations and derive the EM learning algorithm. We test our model performance using simulated data from known underlying DBF binding configurations. Finally, we present results of applying our model to real genomic data, including both DNase-seq data and MNase-seq data. Those results will show that we can infer an accurate view of protein-DNA interaction landscape when jointly modeling all available datasets.

4.1 A multi-factor binding model with a proper likelihood function

COMPETE is developed as a Boltzmann chain (Wasson and Hartemink, 2009). Its transition weights are not constrained to be probabilities but can be any positive real number. Its emission weights are not constrained either, although they always are since PWM models are used to set the emission weights. Since the learning algorithms, as well as other theories, are much better developed and easier to work with if all COMPETE parameters are constrained to be probabilities and a proper likelihood function can be calculated from COMPETE. Therefore, in this section, we will show that for a set of unconstrained positive transition weights, we can always find a corresponding set of transition probabilities that will give the same inference results in posterior decoding. In other words, we can work with a set of proper transition probabilities and a proper likelihood function in a similar model of COMPETE and if desired, the set of probabilities can be transformed into the transition weights that are usually used in COMPETE.

Using the notation established in Section 1.5.1, we consider the following simple case in which we have a sequence of length 3 and two DBFs \( \pi_1 \) and \( \pi_2 \), and the unbound “factor” \( \pi_0 \). \( \pi_2 \) has a length 2 motif (so we have \( \pi_{2,1}, \pi_{2,2} \)) and \( \pi_1 \) has a length 1 motif. At thermodynamic equilibrium, we have 12 possible states (Figure 4.1).

Suppose we need to calculate the probability of state \( C_{11} \) under the Boltzmann
distribution. At equilibrium, we have:

\[ \pi_1 + \pi_2 + S \Rightarrow C_{11} \]

The statistical weight of a particular state, like \( C_{11} \), calculated by COMPETE is:

\[ w(C_{11}) = c(\pi_1)c(\pi_2)e(S_1|\pi_1)e(S_2|\pi_{2,1})e(S_3|\pi_{2,2}) \]

And the posterior probability of \( C_{11} \) is the normalized weight:

\[
P(C_{11}|S_{1:G}) = \frac{w(C_{11})}{\sum_{i=0}^{11} w(C_i)} = \frac{c(\pi_1)c(\pi_2)e(S_1|\pi_1)e(S_2|\pi_{2,1})e(S_3|\pi_{2,2})}{c(\pi_0)c(\pi_0)e(S_1|\pi_0)e(S_2|\pi_0)e(S_3|\pi_0) + \sum_{i=1}^{11} w(C_i)}
\] (4.1)

It can be recognized that the first term in the denominator is the weight for \( C_0 \).

Let’s denote a set of transition probabilities in the SSM as \( \alpha(\pi_j|\pi_i) \), the transition probability of transiting from state \( \pi_i \) to state \( \pi_j \). Note that in the setting of our model, if \( \pi_i \) and \( \pi_j \) are consecutive states belonging to the same DBF, then \( \alpha(\pi_j|\pi_i) \) is always 1. This means a DBF has to bind continuously for its length until next
DBF can bind. We therefore only need to consider the case where \( \pi_i \) is the last state of a DBF and \( \pi_j \) is the beginning state of a DBF (they could be the same DBF), i.e., the probability \( \alpha(\pi_{i,1}|\pi_{j,L_i}) \). In addition, our model assumes that the probability of transitioning into a DBF does not depend on last state:

\[
\alpha(\pi_{i,1}|\cdot) = \alpha(\pi_{i,1})
\]

and we have the constraint:

\[
\sum_i \alpha(\pi_{i,1}) = 1 \quad (4.2)
\]

We can see from Equation (4.1) that if:

\[
\frac{c(\pi_i)}{c(\pi_0)^{L_i}} = \frac{\alpha(\pi_i)}{\alpha(\pi_0)^{L_i}} \quad \forall \ i \quad (4.3)
\]

then the posterior probability calculated by Equation (4.1) does not change. Equation (4.3) also suggests that, since the transition weights \( c(\pi_i) \) in COMPETE is unconstrained positive real numbers, it is necessary to set the unbound factor transition weight \( c(\pi_0) \) to be a fixed number so the parameters in COMPETE are identifiable. If we fix \( c(\pi_0) \) to be 1, then the above can be written as:

\[
\frac{\alpha(\pi_i)}{\alpha(\pi_0)^{L_i}} = \frac{c(\pi_i)}{c(\pi_0)^{L_i}} = c(\pi_i) \quad \forall \ i \quad (4.4)
\]

Therefore, given a set of transition weights, we can get the probabilities that give identical posterior probabilities by solving the equations defined by Equations (4.2) and (4.4) and vice versa. Using transition probabilities, COMPETE can be used to calculate a proper likelihood function of the data. Given the equivalence between the transition weights and transition probabilities, we will work with transition probabilities only in the rest of this chapter.
4.2 Multivariate state-space model (SSM)

4.2.1 Model definition

The COMPETE model structure introduced in Section 1.5.1 illustrates how the state transition structures. To define a multivariate SSM, we will look at a specific state path. We first establish a few notations:

1. Denote the DNA sequence in the model as $S$, and the sequence length is $G$. The nucleotide at position $g$ is $s_g$ ($1 \leq g \leq G$ and $s_g \in \{A, C, G, T\}$);

2. Suppose we consider $K + 1$ DBFs $\pi_0, \ldots, \pi_K$ in the model. DBF $\pi_k$ has a length $L_k$ motif. We further designate $\pi_0$ to be a special “unbound” state with length 1 and $\pi_K$ to be the nucleosome state with length 157 (147 nucleosome states plus 5 padding states on each side of the nucleosome, see Figure 1.6B);

3. Use variable $z_g$ to denote the state at position $g$ of the DNA sequence, $z_g \in \{\pi_0, \pi_{1,1}, \ldots, \pi_{1,L_1}, \pi_{2,1}, \ldots, \pi_{2,L_2}, \ldots, \pi_{K,1}, \ldots, \pi_{K,L_K}\}$;

4. Suppose for a particular state path, there are $N$ DBFs binding at the DNA sequence. Denote the $n^{th}$ DBF as $t_n$, $t_n \in \{\pi_0, \ldots, \pi_K\}$.

Figure 4.2 shows a schematic view of the state path and the defined notations.

Figure 4.2A shows the state path in COMPETE: COMPETE only models the DNA sequence. Each state in the path only emits a nucleotide. In our multivariate SSM, we will add additional data types. As a starting point, we will add DNase-seq counts first (Figure 4.2B):

Denote DNase-seq count at position $g$ as $d_g$, and $D = \{d_1, \ldots, d_G\}$

We will later expand the model to add MNase-seq data.

In the following section, we will further present some distributional assumptions in the model to complete the model definition.
Figure 4.2: A schematic description of the multivariate SSM state path. Square box denotes a state in the state path. States that belong to the same DBF are grouped using dashed rectangles. Circles denote the particular data emitted by a state. Arrows denote either the transition between states or the emission of data at each state. (A) COMPETE only models the sequence data (only nucleotide is emitted at each state). (B) In our multivariate SSM, we will add additional data type, such as DNase-seq counts.

4.2.2 Sampling distributions in SSM

Transition probabilities

Similar to COMPETE, we define transition probabilities with a structure similar to Figure 1.6:

- \( P(z_{g+1} = \pi_{k,l} | z_g = \pi_{k,l-1}) = 1.0 \ \forall l > 1 \), i.e., transition probability between consecutive states within a DBF is always 1.0;

- \( P(z_{g+1} = \pi_{k,1} | z_g = \pi_{j,L_s}) = \alpha_k \), i.e., probability of transitioning into a new DBF is independent of the previous DBF;

- \( P(z_{g+1} | z_g) = 0 \) in any other cases;
Let \( \alpha = (\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_K) \). The transition probabilities should sum to 1:
\[
\sum_{0}^{K} \alpha_k = 1.0.
\]

The initial probabilities are defined similarly:

- \( P(z_1 = \pi_{k,1}) = \alpha_k, \quad \forall k \in \{0, 1, 2, \ldots, K\} \)
- \( P(z_1 = \pi_{k,i}) = 0, \quad \forall k \in \{0, 1, 2, \ldots, K\} \) and \( i > 1 \)

**Emission distribution for nucleotides**

We use similar assumptions for the emission distribution for nucleotides as in COMPETE:
\[
P(s_g | z_g = \pi_{k,l}) = \text{Categorical}(s_g | \pi_{k,l}) \quad \forall k \in \{0, 1, 2, \ldots, K - 1\}
\]

\( \text{Categorical}(s_g | \pi_{k,l}) \) is a categorical distribution defined by the PWM of DBF \( \pi_k \).

Since we have high quality estimations of PWMs, we currently consider these to be fixed parameters obtained from other data sources, such as PBM data, although we could relax this restriction to allow PWMs to be learned through model fitting as well. For \( \pi_K \), the nucleosome state, the model uses a slightly more complicated dinucleotide emission model. However, since we will keep that fixed as well, it suffices to understand it as similarly with a PWM model. For details, see Wasson and Hartemink (2009).

**Emission distribution for DNase-seq counts**

Here we use the DNase-seq data in Hesselberth et al. (2009). To determine an appropriate sampling distribution for the DNase-seq data, we usually look at the empirical distribution of the data and try to find a parametric distribution that describes the data well. We found that a negative binomial (NB) distribution can describe the data sufficiently well, based on the following observations:
1. We pooled all DNase-seq data at a specific position of all the Phd1 binding sites (identified by Rhee and Pugh (2011)) and estimated a NB distribution using MLE. We then used quantile-quantile plot to compare the estimated NB distribution and empirical distribution from data (Figure 4.3A). We found that the estimated NB distribution can describe the empirical counts distribution sufficiently well;

2. We pooled DNase-seq data from all the digital footprints (identified by Hesselberth et al. (2009)) and estimated a NB distribution using MLE. We then visualized the estimated NB distribution along side the empirical counts distribution from data (Figure 4.3B). We found that these two distributions were very similar to each other;

3. For each TF whose binding sites were profiled in Rhee and Pugh (2011), we randomly selected a few positions within its binding sites; for each position, we pooled the corresponding DNase-seq data and estimated a NB distribution as before; we then used $\chi^2$ test to test whether the estimated NB distribution was significantly different from the empirical counts distribution. For all the positions we tested (except a few positions within the Rap1 binding site), the p-values were larger than 0.05 (not significantly different), which meant the estimated NB distributions were sufficient fit for the data.

Based on the observations above, we will use negative binomial sampling distribution to model the DNase-seq counts. Formally, the negative binomial distribution is parameterized by a size parameter $\phi$ and a mean parameter $\mu$. Its probability mass function (PMF) for a count $d$ is:

$$P(d|\phi, \mu) = \frac{\Gamma(\phi + d)}{\Gamma(d)\Gamma(\phi)} \left(\frac{\mu}{\mu + \phi}\right)^d \left(\frac{\phi}{\phi + \mu}\right)^\phi$$
In addition to our observations in Figure 4.3, we also observe that for some TFs, the mean DNase-seq counts is the same across its binding sites (Figure 4.4). However, we do observe that in many cases, there is no visible difference between the DNase-seq counts within the TF binding site and its flanking region (for example, Cst6 in Figure 4.4). We will discuss these cases further in later parts of this chapter.

Based on the observations above, as well as the observation we made on the DNase-seq data at nucleosome positions in Chapter 3, we parameterize the NB emission distribution at each DBF as the following:

- if the DBF is a TF (or the “unbound” factor), i.e., for \( k \in \{0, 1, \ldots, K - 1\} \), we used the same size parameter \( \phi_k \) and the same mean parameter \( \mu_k \) for the DNase-seq counts within the DBF binding site;

- if the DBF is a nucleosome (\( \pi_k \)), we use the same size parameter \( \phi_K \) for counts within nucleosome states. We use one parameter \( \mu_K \) for the mean counts at
Figure 4.4: Average DNase-seq data at four TF binding sites (MacIsaac et al., 2006). Red dashed lines indicate the boundary of each TF binding site. The number of binding sites, as well as the estimated NB parameters using DNase-seq counts within the binding site, is shown as titles of each plot.

the center of the nucleosome (i.e., for state \( \pi_{K,79} \)). We then use scaling factors \( h_t, l \in \{1, 2, \ldots, 157\} \) to scale \( \mu_K \) to get the mean parameters for other states within the nucleosome. By definition \( h_{79} = 1 \). We also constrain the scaling factors to be symmetric across the nucleosome states: \( h_l = h_{157-l+1} \). Scaling factors are calculated using data at nucleosome positions (defined by Brogaard et al. (2012)) and are kept fixed.

This concludes our model definition. Our model therefore includes the following parameters that need to be estimated from data:

Transition probabilities: \( \alpha = \{\alpha_0, \alpha_1, \ldots, \alpha_K\} \)

Negative binomial size parameters: \( \phi = \{\phi_0, \phi_1, \ldots, \phi_K\} \)

Negative binomial mean parameters: \( \mu = \{\mu_0, \mu_1, \ldots, \mu_K\} \)

Use \( \theta = \{\alpha, \phi, \mu\} \) to denote the parameter set. We will use an EM algorithm introduced in the following sections to estimate those parameters.
4.3 General expectation maximization (EM) theory

Expectation maximization is a data augmentation technique for maximizing the marginal likelihood (marginalized over hidden, unobserved data) of the parameter \( \theta \) given the observed data, or the posterior probability of parameter \( \theta \) given the observed data. We denote the observed data as \( Y \) and unobserved data as \( Z \). We want to find \( \theta \) that maximizes the following:

\[
\log[P(\theta|Y)]
\]

Using data augmentation, we can have the following:

\[
\log[P(\theta|Y)] = \log[P(\theta|Z,Y)] + \log[P(Z|Y)] - \log[P(Z|\theta,Y)]
\]

\[
= \int_Z \log[P(\theta|Z,Y)]P(Z|\theta^*,Y)dZ + \int_Z \log[P(Z|Y)]P(Z|\theta^*,Y)dZ
\]

\[
- \int_Z \log[P(Z|\theta,Y)]P(Z|\theta^*,Y)dZ
\]

where \( \theta^* \) is some known value of the parameter \( \theta \). Define

- \( Q(\theta, \theta^*) = \int_Z \log[P(\theta|Z,Y)]P(Z|\theta^*,Y)dZ \)
- \( K(\theta, \theta^*) = \int_Z \log[P(Z|Y)]P(Z|\theta^*,Y)dZ \)
- \( H(\theta, \theta^*) = \int_Z \log[P(Z|\theta,Y)]P(Z|\theta^*,Y)dZ \)

It can be shown that \( H(\theta, \theta^*) \leq H(\theta^*, \theta^*) \) (i.e., as a function of \( \theta \), \( H(\theta, \theta^*) \) is maximized at \( \theta^* \)). Since \( K(\theta, \theta^*) \) does not depend on \( \theta \), therefore if we can find a new \( \theta \) that maximize \( Q \), the posterior of \( \theta \) will increase at this new value of \( \theta \) (compared to the posterior at \( \theta^* \)). The EM algorithm therefore works as the following:

1. Initialize the value of \( \theta \) to some value \( \theta_0 \) (could be a random value) and set \( \theta^* \) to \( \theta_0 \);
2. Find the new value of $\theta$ that maximizes $Q$. Call this new value $\theta^*$. The specific form of $\theta^*$ depends on the functional form of $Q$. Usually it can be found by differentiating $Q$;

3. Update $\theta^*$ to $\theta^*$ and repeat the step above.

It can be shown that through the iterative updates, $\theta$ will converge to a (local) mode of the posterior probability.

Following this general guideline, we will derive the specific form of the $Q$ function in our model, as well as the specific expression of $\theta^*$ for each parameter in the model.

4.4 Learning model parameters through EM

In the case of our SSM, the unobserved data is the discrete state sequence $Z$ and the observed data $Y = \{S, D\}$, therefore:

$$Q(\theta, \theta^*) = \sum_z \log[P(\theta|Z, Y)]P(Z|\theta^*, Y)$$

$$= \sum_z P(Z|\theta^*, Y)\log \left( \frac{P(Z, Y|\theta)P(\theta)}{P(Z, Y)} \right)$$

Maximizing $Q(\theta, \theta^*)$ is equivalent to maximizing

$$\sum_z P(Z|\theta^*, Y)\log[P(Z, Y|\theta)P(\theta)]$$

If using uniform prior for all the parameters, $P(\theta) = P(\alpha, \phi, \mu) \propto 1$, then it is also equivalent to maximizing

$$\sum_z P(Z|\theta^*, Y)\log[P(Z|Y|\theta)]$$
In our SSM, we can expand $\log[P(Z,Y|\theta)]$ to get:

$$Q'(\theta, \theta^*) = \sum_Z P(Z|\theta^*, Y) \log[P(Z,Y|\theta)]$$

$$= \sum_Z P(Z|\theta^*, Y)$$

$$\times \left( \log P(z_1|\alpha) + \sum_{g=2}^G \log P(z_g|z_{g-1}, \alpha) + \sum_{g=1}^G \log P(s_g|z_g) + \sum_{g=1}^G \log P(d_g|z_g) \right)$$

We can find the parameter values that maximize $Q'$ through differentiate $Q'$. We'll derive the form of each parameter in the following sections.

4.4.1 Updating transition probability parameters through EM

Without loss of generality, we consider one of the transition probabilities, $\alpha_1$. Since $\alpha_k$ is subjected to the constraint that $\sum \alpha_k = 1$, we need to do maximization using Lagrange multiplier $\delta$:

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \alpha_1} - \frac{\partial \delta(\sum \alpha_k - 1)}{\partial \alpha_1} = \sum_Z P(Z|\theta^*, Y) \times \frac{\partial}{\partial \alpha_1} \left( \log P(z_1|\alpha) + \sum_{g=2}^G \log P(z_g|z_{g-1}, \alpha) \right) - \delta$$

$$= \sum_{g=1}^G P(z_g = \pi_{1,1}|\theta^*, Y) - \delta$$

$$= 0$$

Solve for $\alpha_1$, we have:

$$\alpha_1 = \frac{\sum_{g=1}^G P(z_g = \pi_{1,1}|\theta^*, Y)}{\delta}$$

Also $\sum \alpha_k = 1$. So we can update each $\alpha_k$ as:

$$\alpha_k = \frac{\sum_{g=1}^G P(z_g = \pi_{k,1}|\theta^*, Y)}{\sum_{k=0}^K \sum_{g=1}^G P(z_g = \pi_{k,1}|\theta^*, Y)}$$

$P(z_g = \pi_{k,1}|\theta^*, Y)$ can be calculated through forward-backward algorithm.
4.4.2 Updating sampling distribution parameters through EM

The \( \{ \phi_k, \mu_k \} \) pairs are independent of each other, so we can consider one such pair at a time. Here we will abuse notation (again) to let \( \pi_k = \{ \pi_{k,1}, \ldots, \pi_{k,L_K} \} \). We derive the partial derivatives with respect to \( \phi_k \) and \( \mu_k \) of the NB PMF here, since we will need them to derive the updating equation for each NB parameter:

\[
P(d_g | \phi_k, \mu_k, z_g \in \pi_k) = \frac{\Gamma(\phi_k + d_g)}{\Gamma(\phi_k)d_g!} \left( \frac{\mu_k}{\phi_k + \mu_k} \right)^{d_g} \left( \frac{\phi_k}{\mu_k + \phi_k} \right)^{\phi_k}
\]

\[
\frac{\partial \log P(d_g | \phi_k, \mu_k, z_g \in \pi_k)}{\partial \phi_k} = \frac{\partial \ln \Gamma(\phi_k + d_g)}{\partial \phi_k} - \frac{\partial \ln \Gamma(\phi_k)}{\partial \phi_k} - \frac{d_g}{\phi_k + \mu_k} + \log \phi_k - \log(\mu_k + \phi_k) + 1 - \frac{\phi_k}{\phi_k + \mu_k}
\]

\[
\frac{\partial \log P(d_g | \phi_k, \mu_k, z_g \in \pi_k)}{\partial \mu_k} = \frac{d_g}{\mu_k} - \frac{d_g}{\mu_k + \phi_k} - \frac{\phi_k}{\mu_k + \phi_k}
\]

where \( \ln \Gamma(\phi) \) is the log Gamma function. If \( \mu \) is parameterized using a scaling factor \( h_l \) with respect to \( \mu_K \), such as the parameterization used in nucleosome states. Then

\[
P(d_g | \phi_K, \mu_K, h_l, z_g = \pi_{K,l}) = \frac{\Gamma(\phi_K + d_g)}{\Gamma(\phi_K)d_g!} \left( \frac{h_l \mu_K}{\phi_K + h_l \mu_K} \right)^{d_g} \left( \frac{\phi_K}{h_l \mu_K + \phi_K} \right)^{\phi_K}
\]

\[
\frac{\partial \log P(d_g | \phi_K, \mu_K, h_l, z_g = \pi_{K,l})}{\partial \mu_K} = \frac{d_g}{\mu_K} - \frac{h_l \times d_g}{h_l \mu_K + \phi_K} - \frac{h_l \phi_K}{h_l \mu_K + \phi_K}
\]

\[
\frac{\partial \log P(d_g | \phi_K, \mu_K, h_l, z_g = \pi_{K,l})}{\partial h_l} = \frac{d_g}{h_l} - \frac{\mu_K \times d_g}{h_l \mu_K + \phi_K} - \frac{\mu_K \phi_K}{h_l \mu_K + \phi_K}
\]

The partial derivative with respect to \( \phi_K \) should remain similarly as before.
Updating size parameter $\phi$

Considering updating $\phi$ for a single state first:

$$
\frac{\partial Q'(\theta, \theta^*)}{\partial \phi_{k,l}}
= \sum_{g=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y) \frac{\partial \log P(d_g|\phi_{k,l}, \mu_{k,l}, z_g = \pi_{k,l})}{\partial \phi_{k,l}}
= \sum_{G=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y)
\times \left( \frac{\partial \Gamma(\phi_{k,l} + d_g) - \partial \Gamma(\phi_{k,l})}{\partial \phi_{k,l}} - \frac{d_g}{\phi_{k,l} + \mu_{k,l}} + \log \phi_{k,l} - \log(\mu_{k,l} + \phi_{k,l}) + 1 - \frac{\phi_{k,l}}{\phi_{k,l} + \mu_{k,l}} \right)
= \left( \log \phi_{k,l} - \log(\mu_{k,l} + \phi_{k,l}) + 1 - \frac{\phi_{k,l}}{\phi_{k,l} + \mu_{k,l}} - \frac{\partial \Gamma(\phi_{k,l})}{\partial \phi_{k,l}} \right) \times \sum_{g=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y)
+ \sum_{g=1}^{G} \left( \left( \frac{\partial \Gamma(\phi_{k,l} + d_g)}{\partial \phi_{k,l}} - \frac{d_g}{\phi_{k,l} + \mu_{k,l}} \right) \times P(z_g = \pi_{k,l}|\theta^*, Y) \right)

All the states that have the same $\phi_k$ should be pooled together to estimate that $\phi_k$.

These states should belong to the same $\pi_k$

$$
\frac{\partial Q'(\theta, \theta^*)}{\partial \phi_k}
= \sum_{\pi \in \pi_k} \left( \sum_{g=1}^{G} P(z_g = \pi|\theta^*, Y) \frac{\partial \log P(d_g|\phi_{k,l}, \mu_{k,l}, z_g = \pi)}{\partial \phi_k} \right)
= \sum_{\pi \in \pi_k} \left( \left( \log \phi_k - \log(\mu_k + \phi_k) + 1 - \frac{\phi_k}{\phi_k + \mu_{k,l}} - \frac{\partial \Gamma(\phi_k)}{\partial \phi_k} \right) \times \sum_{g=1}^{G} P(z_g = \pi|\theta^*, Y) \right)
+ \sum_{\pi \in \pi_k} \left( \sum_{g=1}^{G} \left( \left( \frac{\partial \Gamma(\phi_k + d_g)}{\partial \phi_k} - \frac{d_g}{\phi_k + \mu_{k,l}} \right) \times P(z_g = \pi|\theta^*, Y) \right) \right)
$$

115
If all the states in $\pi_k$ have the same mean $\mu_k$, then

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \phi_k}$$

$$= \left( \log\phi_k - \log(\mu_k + \phi_k) + 1 - \frac{\phi_k}{\phi_k + \mu_k} - \frac{\partial L \Gamma(\phi_k)}{\partial \phi_k} \right) \times \sum_{g=1}^{G} \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y)$$

$$+ \sum_{g=1}^{G} \left( \frac{\partial L \Gamma(\phi_k + d_g)}{\partial \phi_k} - \frac{d_g}{\phi_k + \mu_k} \right) \times \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y) = 0$$

If the $\mu$ is different for the states in the group (for example, when scaling $\mu$ with a scaling factor in nucleosome states), then

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \phi_k}$$

$$= \left( \log\phi_k + 1 - \frac{\partial L \Gamma(\phi_k)}{\partial \phi_k} \right) \times \sum_{g=1}^{G} \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y)$$

$$- \sum_{\pi \in \pi_k} \left( \log(\mu_{k,l} + \phi_k) + \frac{\phi_k}{\phi_k + \mu_{k,l}} \right) \sum_{g=1}^{G} P(z_g = \pi|\theta^*, Y)$$

$$+ \sum_{g=1}^{G} \left( \frac{\partial L \Gamma(\phi_k + d_g)}{\partial \phi_k} \times \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y) \right) - \sum_{\pi \in \pi_k} \left( \sum_{g=1}^{G} \frac{d_g}{\phi_k + \mu_k} P(z_g = \pi|\theta^*, Y) \right)$$

$$= 0$$

The above equations needs to be solved numerically to update $\phi_k$. 

116
Updating mean parameter $\mu$

Again, consider updating $\mu$ for a single state first:

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \mu_{k,l}}$$

$$= \sum_{g=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y) \frac{\partial \log P(d_g|\phi_{k,l}, \mu_{k,l}, z_g = \pi_{k,l})}{\partial \mu_{k,l}}$$

$$= \sum_{g=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y) \left( \frac{d_g}{\mu_{k,l}} - \frac{d_g}{\mu_{k,l} + \phi_{k,l}} - \frac{\phi_{k,l}}{\mu_{k,l} + \phi_{k,l}} \right)$$

$$= \sum_{g=1}^{G} \left[ \left( \frac{1}{\mu_{k,l}} - \frac{1}{\mu_{k,l} + \phi_{k,l}} \right) d_g P(z_g = \pi_{k,l}|\theta^*, Y) - \frac{\phi_{k,l}}{\mu_{k,l} + \phi_{k,l}} P(z_g = \pi_{k,l}|\theta^*, Y) \right]$$

$$= \frac{\phi_{k,l}}{\mu_{k,l} + \phi_{k,l}} \left( \frac{1}{\mu_{k,l}} \sum_{g=1}^{G} d_g P(z_g = \pi_{k,l}|\theta^*, Y) - \sum_{g=1}^{G} P(z_g = \pi_{k,l}|\theta^*, Y) \right)$$

Also, the states that have the same $\mu_k$ should be pooled together to estimate $\mu_k$.

Those states should also have the same $\phi_k$ in our model setup.

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \mu_k}$$

$$= \sum_{\pi \in \pi_k} \left( \frac{\phi_k}{\mu_k + \phi_k} \left( \frac{1}{\mu_k} \sum_{g=1}^{G} d_g P(z_g = \pi|\theta^*, Y) - \sum_{g=1}^{G} P(z_g = \pi|\theta^*, Y) \right) \right)$$

$$= \frac{\phi_k}{\mu_k + \phi_k} \left( \frac{1}{\mu_k} \sum_{g=1}^{G} \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y) - \sum_{g=1}^{G} \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y) \right)$$

$$= 0$$

Solve for $\mu$, we get:

$$\mu_k = \frac{\sum_{g=1}^{G} d_g \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y)}{\sum_{g=1}^{G} \sum_{\pi \in \pi_k} P(z_g = \pi|\theta^*, Y)}$$

So the update of $\mu$ is a weighted average of the data, weighted by the posterior binding probabilities of each state.
Updating \( \mu \) with scaling factor \( h_l \)  

In the case of nucleosome states, the DNase-seq mean is parameterized using a single mean \( \mu_K \) for the nucleosome center and mean scaling factors \( h_l \) \((l \in \{1, 2, \ldots, 157\})\), including 5 bp linker region on each side. We could impose symmetry constraints, i.e., \( h_l = h_{157-l+1} \).

\[
\frac{\partial Q'(\theta, \theta^*)}{\partial \mu_{K,l}}
= \sum_{g=1}^{G} P(z_g = \pi_{K,l}|\theta^*, Y) \left( \frac{d_g}{\mu_K} - \frac{h_l \times d_g}{h_l \mu_K + \phi_{K,l}} - \frac{h_l \phi_{K,l}}{h_l \mu_K + \phi_{K,l}} \right)
\]

\[
= \sum_{g=1}^{G} P(z_g = \pi_{K,l}|\theta^*, Y)d_g\left( \frac{1}{\mu_K} - \frac{h_l}{h_l \mu_K + \phi_{K,l}} \right) - \frac{h_l \phi_{K,l}}{h_l \mu_K + \phi_{K,l}} \sum_{g=1}^{G} P(z_g = \pi_{K,l}|\theta^*, Y)
\]

\[
= \frac{\phi_{K,l}}{h_l \mu_K + \phi_{K,l}} \left( \frac{1}{\mu_K} \sum_{g=1}^{G} d_g P(z_g = \pi_{K,l}|\theta^*, Y) - h_l \sum_{g=1}^{G} P(z_g = \pi_{K,l}|\theta^*, Y) \right)
\]

Pool all the states that share the same \( \mu_K \) together. Again, those states should have the same \( \phi_K \), but they may have different scaling factor \( h_l \). So to update \( \mu_K \), we need to solve the following equation numerically:

\[
\sum_{l=1}^{157} \frac{\phi_K}{h_l \mu_K + \phi_K} \left( \frac{1}{\mu_K} \sum_{g=1}^{G} d_g P(z_g = \pi_{K,l}|\theta^*, Y) - h_l \sum_{g=1}^{G} P(z_g = \pi_{K,l}|\theta^*, Y) \right) = 0
\]

4.4.3 Updating model parameters with multiple DNA sequences

We usually will need to train the model on multiple sequences. Suppose we have \( J \) sequence of observations (\( J \) promoter regions, for example), each of them has length \( G_j \). Conditional on model parameters, the \( J \) sequences of observations are
independent with each other. The $Q$ function can be rewritten as the following:

$$Q(\theta, \theta^*) = \sum_Z P(Z|\theta^*, Y) \log[P(Z, Y|\theta)P(\theta)]$$

$$= \sum_Z \left( \prod_{j=1}^{J} P(Z_j|\theta^*, Y_j) \right) \left( \log(P(\theta)) + \sum_{j=1}^{J} \log(P(Z_j, Y_j|\theta)) \right)$$

$$= \sum_Z \left( \prod_{j=1}^{J} P(Z_j|\theta^*, Y_j) \sum_{j=1}^{J} \log(P(Z_j, Y_j|\theta)) \right) + \log(P(\theta))$$

For one of the term:

$$\sum_{j=1}^{J} P(Z_j|\theta^*, Y_j) \log(P(Z_1, Y_1|\theta)) = \sum_{Z_1} P(Z_1|\theta^*, Y_1) \log(P(Z_1, Y_1|\theta))$$

To see why the above is true, consider the state path in one of the sequences, $Z_2$ for example. Fix the state path on all other sequences at some path $\tilde{Z}_j$, then the above becomes:

$$\sum_{j=1}^{J} P(Z_j|\theta^*, Y_j) \left( \prod_{j=1}^{J} P(Z_j|\theta^*, Y_j) \right) P(Z_1|\theta^*, Y_1) \log(P(Z_1, Y_1|\theta)) $$

$$= \left( \prod_{j=3}^{J} P(Z_j|\theta^*, Y_j) \right) P(Z_1|\theta^*, Y_1) \log(P(Z_1, Y_1|\theta))$$

This relationship repeats for another $\tilde{Z}_j$ so all $P(Z_j|\theta^*, Y_j)$ can sum to 1 (and disappear from the equation) except $P(Z_1|\theta^*, Y_1)$ So

$$Q(\theta, \theta^*) = \sum_{j=1}^{J} \sum_{Z_j} P(Z_j|\theta^*, Y_j) \log(P(Z_j, Y_j|\theta)) + \log(P(\theta))$$

Therefore,

$$Q'(\theta, \theta^*) = \sum_{j=1}^{J} Q'_j(\theta, \theta^*)$$
Updating transition probabilities

\[
\frac{\partial Q'(\theta, \theta^*)}{\partial \alpha_1} - \frac{\partial \delta (\sum \alpha_k - 1)}{\partial \alpha_1} = \sum_{j=1}^J \frac{\partial Q_j'(\theta, \theta^*)}{\partial \alpha_1} - \frac{\partial \delta (\sum \alpha_k - 1)}{\partial \alpha_1} = \sum_{j=1}^J \sum_{g=1}^{G_j} P(z_g^j = \pi_{1,1}^{\theta^*}, Y) - \delta = 0
\]

Again, \( \sum_{k=0}^K \alpha_k = 1 \), solve for \( \alpha_k \), we have

\[
\alpha_k = \frac{\sum_{j=1}^J \sum_{g=1}^{G_j} P(z_g^j = \pi_{1,1}^{\theta^*}, Y)}{\sum_{k=1}^K \sum_{j=1}^J \sum_{g=1}^{G_j} P(z_g^j = \pi_{1,1}^{\theta^*}, Y)}
\]

Updating \( \phi \)

\[
\frac{\partial Q'(\theta, \theta^*)}{\partial \phi_k} = \sum_{j=1}^J \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z_g^j = \pi_{l,1}^{\theta^*}, Y) \frac{\partial \log P(d_g^j|\phi_k, \mu_k, z_g^j \in \pi_k)}{\partial \phi_k} = \sum_{j=1}^J \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z_g^j = \pi_{l,1}^{\theta^*}, Y) \times \left( \frac{\partial L \Gamma(\phi_k + d_g^j)}{\phi_k} - \frac{\partial L \Gamma(\phi_k)}{\phi_k} - \frac{d_g^j}{\phi_k + \mu_k} + log \phi_k - log (\mu_k + \phi_k) + 1 - \frac{\phi_k}{\phi_k + \mu_k} \right)
\]

\[
= \left( log \phi_k - log (\mu_k + \phi_k) + 1 - \frac{\phi_k}{\phi_k + \mu_k} - \frac{\partial L \Gamma(\phi_k)}{\phi_k} \right) \times \sum_{j=1}^J \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z_g^j = \pi_{l,1}^{\theta^*}, Y) + \sum_{j=1}^J \sum_{g=1}^{G_j} \left( \frac{\partial L \Gamma(\phi_k + d_g^j)}{\phi_k} - \frac{d_g^j}{\phi_k + \mu_k} \right) \times \sum_{l=1}^{L_k} P(z_g^j = \pi_{l,1}^{\theta^*}, Y)
\]

The equation needs to be solved numerically.
Updating $\mu$

$$\frac{\partial Q'(\theta, \theta^*)}{\partial \mu_k} = \sum_{j=1}^{J} \frac{\partial Q'_j(\theta, \theta^*)}{\partial \mu_k}$$

$$= \sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y) \frac{\partial \log P(d^j_g|\phi_k, \mu_k, z^j_g \in \pi_k)}{\partial \mu_k}$$

$$= \sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y) \left( \frac{d^j_g}{\mu_k} - \frac{d^j_g}{\mu_k + \phi_k} - \frac{\phi_k}{\mu_k + \phi_k} \right)$$

$$= \sum_{j=1}^{J} \sum_{g=1}^{G_j} \left( \frac{1}{\mu_k} - \frac{1}{\mu_k + \phi_k} \right) \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y) - \frac{\phi_k}{\mu_k + \phi_k} \sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y)$$

$$= \frac{\phi_k}{\mu_k(\mu_k + \phi_k)} \sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y) - \frac{\phi_k \mu_k}{\mu_k(\mu_k + \phi_k)} \sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y)$$

$$= 0$$

Solve for $\mu$, we get:

$$\mu_k = \frac{\sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y)}{\sum_{j=1}^{J} \sum_{g=1}^{G_j} \sum_{l=1}^{L_k} P(z^j_g = \pi_{k,l|\theta^*}, Y)}$$

### 4.5 Model implementation

The model is implemented as a Python package and includes the following components:

1. Core forward-backward algorithm. This component is crucial for both model inference and model learning. For computational efficiency, it is implemented in C and interfaces with other components through Cython (Behnel et al., 2011);
2. Model construction. This component is responsible for constructing appropriate model skeleton structure given a set of DBFs;

3. Data IO. This component is responsible for reading genomic data, including sequence and DNase-seq data (as well as MNase-seq data mentioned later);

4. Inference output and visualization. This component is responsible for outputting model inference results (in the form of posterior probabilities) and visualizing the results.

5. Simulation. This component is responsible for generating known binding configurations given a set of DBFs and their associated transition probabilities; and for generating synthetic data given a binding configuration and a set of emission distribution parameters. This component is useful for debugging the code and for testing model performance.

The model implementation can be found at:

https://jianlingzhong.github.io/

4.6 Model inference results

4.6.1 Inference using simulated data

We first test our model performance using simulated data. The simulated data is synthesized through the following:

1. Given a set of DBFs and their associated transition probabilities, simulate a binding configuration;

2. Based on the binding configuration, simulate DNase-seq read counts using negative binomial parameters calculated through MLE (by pooling DNase-seq data on a given DBF’s binding sites, see Figure 4.4);
3. Also based on the binding configuration, simulate a DNA sequence using the PWMs from Gordân et al. (2011) and the nucleosome dinucleotide model from Wasson and Hartemink (2009)

Using simulated data, we test our model performance by recovering the underlying binding configuration. In this simulation study, we set the model’s DNase-seq negative binomial emission distribution parameters to the true parameters. However, the transition probabilities are learned from the simulated data using EM.

Here we present results from two simulation settings:

1. The true binding configuration is generated with only nucleosome and the “unbound” states. A binding configuration path of 5,000 bp is simulated. To recover the true binding configuration, a SSM with the 42 TFs in Section 2.1.4, “unbound” and nucleosome in the model is trained using only the DNA sequence (Figure 4.5B) or using both DNA sequence and DNase-seq counts (Figure 4.5C);

2. The true binding configuration is generated with nucleosome, Abf1, Cbf1 and the “unbound” states. A binding configuration path of 5,000 bp is simulated. To recover the true binding configuration, a SSM with the 42 TFs in Section 2.1.4, “unbound” and nucleosome in the model is trained using only the DNA sequence (Figure 4.6B) or using both DNA sequence and DNase-seq counts (Figure 4.6C);

We can see from Figures 4.5 and 4.6 that:

1. Learning from DNA sequence only seems to poorly recover the true binding configuration. This makes sense because sequence preferences of many DBF are degenerated and DNA sequence is only mildly informative about the binding DBF. In addition, in real in vivo environments, other forces, such as nucleosome
Figure 4.5: Simulation with nucleosome as the only DBF. True binding configuration is shown at the bottom of each panel. Nucleosomes are shown as gray bars. (A) Simulated DNase-seq counts. (B) Recovered binding configuration through training model with DNA sequence only. (C) Recovered binding configuration through training model with both DNA sequence and DNase-seq data.

remodels and RNA Pol II, will also shape the binding landscape, although we do not have those additional influencing factors in our simulation study here;

2. However, DNA sequences do provide useful information about DBF binding. For example, in both Figure 4.5B and Figure 4.6B, we see that true nucleosome
Figure 4.6: Simulation with DBFs nucleosome, Abf1 and Cbf1. True binding configuration is shown at the bottom of each panel. Nucleosomes are shown as gray bars. Abf1 and Cbf1 are shown as red and blue dots, respectively. (A) Simulated DNase-seq counts. (B) Recovered binding configuration through training model with DNA sequence only. (C) Recovered binding configuration through training model with both DNA sequence and DNase-seq data.
binding positions are generally covered by nucleosomes in the predicted landscape. And in Figure 4.6B, all the true TF binding events are also recovered (although there are lots false positives). These observations confirm that our model setup is consistent with previous views that sequence is a factor that influences nucleosome positioning (Struhl and Segal, 2013);

3. Adding DNase-seq data into the model dramatically improved the model prediction. Specifically, all the nucleosome binding events are successfully recovered (with 0 false positive and 0 false negative). All the TF binding events are also successfully recovered. In addition, in both Figure 4.5C and Figure 4.6C, the false positive TF binding events are reduced significantly. These results show that DNase-seq is very informative in predicting protein-DNA interaction landscape.

Figures 4.5 and 4.6 show encouraging results. In the following sections, we will apply our model to real data.

4.6.2 Inference using DNase-seq data

Here we will present results of applying our SSM to real DNase-seq data. Again, we used the DNase-seq data in Hesselberth et al. (2009). Specifically, we applied our model to two regions: ChrXIII:659,000–660,000 (sacCer2, promoter region of VTII and CIK1) and ChrI:60,500–64,000 (sacCer2, promoter regions of PTA1, ERV46 and CDC24). We used a model with 42 TFs (Section 2.1.4), “unbound” and nucleosome to learn the protein-DNA interaction landscape from data.

Figures 4.7 and 4.8 show the results of learned landscapes. From the figures, we can see that:

1. Although the model includes a number of DBFs, learning from DNA sequence only can still provide a reasonable estimate of the binding landscape (comparing
to using randomly guessed parameters in Figure 2.1). This is similar to our observation in simulation studies (Figures 4.5 and 4.6).

2. However, a few mispredictions occur when only using DNA sequence: false positive TF binding events (such as the Adr1 binding in both Figures 4.5 and 4.6); false negative TF binding events and misplaced nucleosomes in Figure 4.6 (near the experimentally identified Abf1 binding site on the left); Nucleosome arrays do not agree with canonical nucleosome positioning near TSS in Figure 4.5.

3. Using both DNA sequence and DNase-seq greatly improved the binding landscape predictions: The previously mispredicted nucleosome free region, as well as the Abf1 binding events, in Figure 4.6 are now correctly predicted; nucleosomes in both figures are correctly phased; many spuriously predicted TF binding events are cleared out by adding DNase-seq data to the model.

However, the predictions using both DNase-seq data and DNA sequence are not perfect. For example, there are still false positive TF binding events predicted. And in the next section, where we further introduce MNase-seq data into the model, we will also see that there are false negative predictions on nucleosome positions as well.

4.7 Incorporating MNase-seq data

Henikoff et al. (2011) collected paired-end sequencing data from MNase digestion experiments. Their protocol allows capturing fragments of a wide range of sizes, including nucleosomal fragments (140–160 bp) and subnucleosomal fragments (20–100 bp). This dataset is very informative about the protein-DNA interaction landscape.

Figure 4.10B and Figure 4.11B show the MNase-seq fragments in the two regions shown in last section (with comparisons with prediction binding landscape using DNase-seq data in panel (A) of each figure). We can see that our predictions made
in the last section largely agree with the MNase-seq data. However, there are some discrepancies, for example, the nucleosome free region in Figure 4.10 actually has an additional nucleosome shown in the MNase-seq data. We therefore hypothesize that adding MNase-seq data into the model could further improve the model predictions.

However, modeling the fragments in paired-end MNase-seq is fundamentally different than modeling the cuts in single-end DNase-seq data. In SSM, we assume

**Figure 4.7:** Learning protein-DNA interaction landscape in region ChrXIII:659,000–660,000 (sacCer2) using DNA sequence only (A) and using both DNA sequence and DNase-seq data (C). DNase-seq data is shown in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel.
that conditioning on the underlying hidden states, current observations are independent with the previous observations. This assumption is reasonable for the cuts data. However, if we represent fragment data using the coverage introduced in Chapter 2, then the assumption breaks because fragments will cover a segment of DNA sequence and the current coverage is not independent with previous coverages, even conditioning on the underlying states.
We therefore use another statistic to represent MNase-seq fragments: the middle point counts, i.e., count the number of fragment middle points that are mapped to a particular genomic location. Each fragment is therefore reduced to one summary statistic (instead of a series of counts in counting the coverage) and we can reasonably assume that each fragment is independent of each other.

In this section, we will focus on the large fragment (or nucleosomal fragments, 140 bp–160 bp fragments) middle point counts. Those fragments are informative in distinguishing nucleosome and nucleosome-free regions (Figure 4.9).

We also model the MNase-seq large fragment middle point counts using negative binomial distribution. Therefore, the learning algorithm is the same as those introduced in Section 4.4. Figures 4.10 and 4.11 show results of incorporating both DNase-seq and MNase-seq data into the model to predict protein binding landscapes. We can see from the figures that MNase-seq data further improves the predictions by shifting nucleosomes to the correct positions and eliminating false positive TF binding events.

4.8 Discussion

In this chapter, we presented a state-space model (SSM) to jointly model different kinds of genomics. We described details of the model formulation as well as the model learning algorithms. To test the performance of the model, we carried out simulation studies to recover known binding configurations. We further applied model to real genomic data. By incrementally increasing the kinds of data in the model and comparing the model performance of models with different levels of complexity, we showed that different data types (DNA sequence, DNase-seq counts and MNase-seq fragments) possess useful and complementary information in predicting protein-DNA interaction landscape. The best performance came from modeling all available data jointly.
Figure 4.9: MNase-seq large fragment middle point counts in nucleosomal positions (A) and TF binding sites (B and C). To ensure that the signal we observe is not caused by MNase bias, we also showed the *in vitro* MNase digestion data on naked DNA from Deniz et al. (2011). Nucleosome positions are from Brogaard et al. (2012). TF binding sites are from MacIsaac et al. (2006).

The EM learning algorithm is critical to ensure an optimal predictive power. One impediment for the learning framework introduced in Chapter 2 is that the MCMC-based learning algorithm is too time-consuming and is not feasible to scale to larger models that include more comprehensive DBFs and more (even genome-wide) data. EM algorithms, on the other hand, converge to an optimal parameter estimation in a reasonable amount of time, enabling the inference on a genome-wide scale. Although EM algorithms typically converges to a local optimal, it should still be a preferred
Figure 4.10: Learning protein-DNA interaction in region ChrXIII:659,000–660,000 (sac-Cer2) using DNA + DNase-seq counts (A) and using DNA + DNase-seq counts + MNase-seq large fragments middle point counts (C). MNase-seq fragments are shown in (B). MNase-seq large fragments middle point counts are also shown as a blue curve in (B). Experimentally identified TF binding sites (MacIsaac et al., 2006) as well as TATA box positions (Rhee and Pugh, 2012) are shown as dots beneath each panel.

choice over MCMC-based approach given the sheer amount of data required, even in the small genome of S. cerevisiae.

We observe that many TFs do not have visible footprints in the DNase-seq data of Hesselberth et al. (2009) (Figure 4.4 includes an example). Possibly those TFs didn’t form stable binding with DNA during the digestion of DNase I. This DNase-seq dataset is therefore not informative for inferring the binding of those TFs. We
can still apply the model to this data since model parameters will adapt through model fitting. However, to infer accurate binding events for those TFs, we will need additional data, such as the MNase-seq data.

This is the first step in building a unified and principled modeling framework to infer protein-DNA interaction landscape through data integration. The model can
be further improved in at least the following aspects:

1. Here we model counts data using negative binomial distributions. In Chapter 3 we mentioned that a better way to model noisy genomic data is to model the \texttt{asinh} transformed version of the data. \texttt{asinh} transformation reduces the variance in data and alleviates the influences of outliers. A different emission distribution should be used to model the transformed data, such as normal distribution. Therefore the specific forms of EM algorithm introduced here needs to be updated;

2. For simplicity, we only used large fragment middle points counts in the paired-end MNase-seq data. Additional information in the data can be modeled as well, such as the short fragments middle points counts. Fragments data provide additional information because they reflect the length of the protecting DBF. Therefore we may consider modeling the fragments directly (see further discussion in Chapter 5).

3. In the current results, we can observe that nucleosomes are often predicted to be binding at certain positions with very high confidence (probability close to 1). However, it is known that nucleosomes that are far away from TSS are fuzzier in their positioning. Such high confidence predictions may be an artifact of our modeling approach: Model parameters may be too sensitive to noise in the data and our approach didn’t capture the variance in parameters. One way to alleviate this issue is to down-weight the DNase-seq or MNase-seq data in the model likelihood by applying an exponentiating factor $\lambda (< 1)$ to those parts of the likelihood function.

As a preliminary study, we only studied a few particular regions of the genome. Further studies with larger, genome-wide studies should be carried out: both in
training model parameters and in making model inferences.
The overarching goal of this dissertation is to infer the interaction between regulatory proteins and the DNA sequence: where and how likely those interaction events happen; what kind of properties do these interaction possess, genome-wide and individually, etc. To achieve this goal, a number of techniques, combined with a few different kinds of data, were used:

- MNase-seq fragments were used to fit (and therefore to guide the inference of) a multi-factor binding biophysical model through maximizing a correlation-based pseudo-likelihood function. We showed that MNase-seq fragments were very informative about DBF binding landscape. When the biophysical principles, the DBF binding preferences and the MNase-seq data were combined in a framework, they were able to predict a protein binding landscape that is consistent with the MNase-seq fragments.

- A Bayes-factor–based method was developed to find genome-wide nucleosome positioning signals in DNase-seq data. Different strategies were used to effectively extract positioning signals from the noisy data, including transforming
the data to reduce variance, exploiting the unique DNase I digestion pattern on nucleosomes and integrating out the uncertainty in model parameters, etc. Our results show that DNase-seq data are very informative about nucleosome positioning and that data points in non-DHS regions should be fully utilized to infer protein-DNA interaction.

- The results above, as well as the discussions therein, showed that different kinds of data often contain noisy information about certain aspects of the whole picture of protein-DNA interaction landscape. Therefore it is necessary to develop a unified, principled and flexible framework to integrate different kinds of data. We presented one such attempt under the framework of state-space model. We also derived an efficient EM algorithm to learn model parameters. By incrementally adding more data into our model, our preliminary results showed that the best predictive performance came from integrating complementary information from all available datasets in the model.

Although the work of this dissertation concludes at the development of the unified modeling framework, the research in this direction merely just started. In this final chapter, we will discuss some possible improvements that we can make in different parts of the modeling approach. We will further discuss an alternative modeling framework to achieve the same goals through data integration. Finally, we will discuss some immediate and important applications of the modeling approach and the protein binding landscape it learned.

5.1 Possible improvements to current modeling approach

5.1.1 Handling bias in genomic data

There have been debates on whether bias in genomic data will overwhelm the true signal or not (Hörz and Altenburger, 1981; Chung et al., 2010; Allan et al., 2012).
However, the existence of bias is a consensus knowledge. Bias can be introduced in
different stages of the experimental protocol, such as enzyme digestion, PCR am-
plification and high-throughput sequencing (see a review by Meyer and Liu (2014)).
In our work described in Chapters 2 and 3, bias does not distort the signal of the
specific data type we used (see detailed discussion in those chapters). However, when
developing a flexible framework for data integration, we need to explicitly handle the
bias of data. Regardless of the source of the bias, its common reflection is that the
distribution of data among different k-mers is not uniform (where it should be, such
as in the data from digestion on naked DNA sequences). There are two possible ways
to handle the bias:

1. Create a “bias-corrected” version of the data outside the model and feed the
“bias-corrected” data into the model instead. Since the bias-correction is in-
dependent of the model, this is the most general-purpose approach. The “cor-
rected” data can be used in any downstream modeling framework. One possible
approach is to model the data $d_g$ at genomic position $g$ as a sum of two terms:

$$d_g = x_g + y_g$$

where $x_g$ is the contribution from protocol bias and $y_g$ is the contribution of
the “real” biological signal. The distribution of $x_g$, which should be a function
of the specific k-mer, can be modeled using data produced by the same protocol
but using naked DNA sequences:

$$x_g \sim f^{bias}(k\text{-}mer)$$

The distribution of $y_g$ can therefore be calculated by subtracting $x_g$ from $d_g$.
Further development of this direction will need more assumptions about the
distributional forms.
2. Model the bias inside a specific modeling framework. For example, a bias correction term $b_g$ can be added into the negative binomial distribution used in Chapter 4:

$$d_g|\mu_k, \phi_k \sim NB(b_g \times \mu_k, \phi_k)$$

$b_g$ could be a function of the specific k-mer in position $g$ and can be calculated beforehand.

5.1.2 Spatial considerations of DBF binding

In our current modeling approach, we assume that a DBF only occupies the number of nucleotides that equal to its motif length. However, in reality, a protein has a 3D structure and its interaction with DNA may protect spaces larger than its motif. For example, in Figure 4.4 we can see that the binding of certain TFs protects DNA from being digested by DNase I even at positions outside the motif. Therefore, in our modeling assumptions we may need to add extra padding positions to each DBF’s motif. Of course, there will be considerations on how many padding positions to add. And the specific size of spatial protection will differ for different proteins. It is possible to describe the padding size using additional parameters in the model and learn those parameters from data. However, this approach complicates the model learning algorithms. Additional biological knowledge on the protection of DBFs will be extremely useful in such cases.

5.1.3 Alternative models to PWM

In the modeling approaches present in this dissertation, we use PWM to model the distribution of nucleotides given a specific underlying state. PWM model assumes each position contribute independently to the binding of a DBF, therefore it can be very easily incorporated in our Markovian-like models. However, the sufficiency of a simple PWM model in representing the binding a DBF has been the subject of an
ongoing debate (Badis et al., 2009; Zhao and Stormo, 2011). More complex model
has been proposed, such as allowing inter-dependent positions with motifs (Bulyk
et al., 2002) and using Bayesian networks to represent motif binding (Agarwal and
Bafna, 1998). While we believe that a simple PWM model suffices most of the time,
more complex models could be considered. However, the overall modeling framework
will need to be revised to allow this flexibility, such as using a generalized-hidden-

5.1.4 Including other kinds of data

The genomic data we explicitly considered in this dissertation are mainly nuclease
digestion data, including DNase-seq data and MNase-seq data. But other kinds of
data, especially ChIP data, can be model using similar approaches.

ChIP data provides binding information for one specific DBF. Different ChIP
data, such as ChIP-seq and ChIP-exo, provides different resolution in locating the
the binding events. Certain DBFs also bind DNA indirectly through a co-factor
(Gordán et al., 2009). It is necessary to take all those aspects into account so that
the model will not be misled.

In addition, one lesson we learned during the course of this dissertation is that ge-
nomic data are extremely noisy. The noise come from different aspects: the intrinsic
noise in biological systems; the low sequencing depth; the difference from different
sequencing platforms; the variations in different labs and personnels that performed
the specific experiments, etc. We, as modeling practitioners, must be aware of these
noises and take appropriate approach to account for these noises.
5.2 An alternative modeling framework

5.2.1 Shall we decouple the binding probability calculation and the likelihood calculation?

Both Chapter 2 and Chapter 4 presented model approaches to infer the protein-DNA interaction landscape. Chapter 4 integrates additional data and therefore is a preferred approach in this dissertation. However, one important difference between the two approaches, besides the data types used, is that Chapter 2 separates the calculation of the data (pseudo-)likelihood from the calculation of the protein binding landscape, while Chapter 4 ties the two together. In other words, in Chapter 2, once we have learned the parameters by maximizing the (pseudo-)likelihood, we can calculate the binding landscape without using the MNase-seq data again. However, in Chapter 4 we still need to include the data to reproduce the learned binding landscape even after the model parameters were estimated.

The approach of Chapter 2 leads to another, alternative framework for data integration. Denote the binding landscape calculated based a given set of parameters \( \theta \) and sequence \( S \) as \( C = \{C_{g,k}\} \) and \( C_{g,k} \) is the posterior binding probability of state \( \pi_k \) at position \( g \). We further use \( C_g \) to denote the vector of posterior binding probability of all states at position \( g \). The likelihood of the genomic data is calculated based on the binding landscape, for example, the likelihood of DNase-seq counts \( d_{1:G} \) could be:

\[
P(d_{1:G}|C) = \prod_g P(d_g|C_g)
\]

In other words, we calculate the binding landscape first (through posterior decoding, for example). Then, conditioning on the binding landscape, we calculate the likelihood of the data. To complete the likelihood function, we need further assumption on the specific sampling distributions. But the general idea is in the equation above.

Using this approach has several advantages:
1. The calculation of binding landscape and the model fitting process can be decoupled. A similar MCMC-based approach can be used to maximize the likelihood of the genomic data. Once the model parameters are learned, calculating the binding landscape is independent of the genomic data. This complies better with our understanding that the generation of genomic data depends on the underlying DBF binding status but the later should not depend on the former. This also allows us to perform in silico simulation of single nucleotide polymorphisms to study the influence of nucleotide mutations on the binding landscape. If we don’t decouple the two process, we will need genomic data under the mutation condition to allow such studies, which is not feasible for systematic genome-wide screening.

2. Complex likelihood functions can be flexibly incorporated. For example, for paired-end fragment data, we can simply use the following form:

\[
P(m_{g_1:g_2}|C) = P(m_{g_1:g_2}|C_{g_1}, \ldots, C_{g_2})
\]

for a fragment \(m\) that spans from position \(g_1\) to position \(g_2\). In particular, we can write the likelihood of this fragment as a function of the bound state probabilities within the spanning region of this fragment. This flexibility is hard to achieve if we don’t decouple the two processes.

3. By decoupling the two processes, and writing the likelihood of the data as \(P(d_g|C_g)\) (the likelihood of a data point is dependent on a probability vector, not a particular state), the model is closer to the data generation process and can model the generation process correctly. See next section for more discussions.

However, a major disadvantage of using this approach is that a simple and efficient EM-based learning algorithm is not obvious. An efficient algorithm is crucial when
performing genome-wide scale inference. Therefore, further work needs to be carried out on the theoretic basis of this direction.

5.2.2 The convolution process entailed in generating genomic data

In our HMM-like model formulation, an underlying assumption is that data at a particular position is generated by a single unknown state. However, our knowledge about genomic data generation process (which we present below) suggests that this assumption may not be accurate. For illustration purpose, we focus only on the DNase-seq counts in this section.

In a population of homogeneous cells, DBF binding at the same genomic position of different cells is dynamic. A TF in one cell may have overlapping binding position with another TF in another cell because of this dynamics. When mapping the sequencing reads back to the reference genome, reads from different kinds of DBF binding at the same genomic location are indistinguishable from each other. Therefore, the final read counts represent convolutions of different binding status.

In the context of the DNase-seq data, a given count $d_g$ at genomic location $g$ is the summation of 0s and 1s from all cells:

$$d_g = 0 + 1 + 1 + 0 + 0 + 0 + 1 + \cdots$$

where the total number of 0s and 1s is the number of cells in the sequenced cell population (assuming there is no amplification from PCR), because for each cell it either generated a read at position $g$ or not. The specific probability of being 0 or 1 depends on the state of the genome at position $g$ in that cell. For simplicity, we only consider the nucleosome states and assume there are only 148 possible states (one state for each nucleosome position, plus one state for background). So each 0 or 1 came from one of the following Bernoulli distributions:

$$\text{Bernoulli}(\omega_l) \quad l \in \{0, 1, 2, \ldots, 147\}$$
in which \( o_l = P(1|\text{state} = l) \). The sum of 0s and 1s within each state therefore follows a binomial distribution:

\[
\mathcal{B}(n_l, o_l) \quad l \in \{0, 1, 2, \ldots, 147\}
\]

where \( n_l \) is the number of cells in state \( l \). Because usually there are millions of cells in a sequenced population, it is reasonable to approximate each of the binomial distributions using a Poisson distribution:

\[
\mathcal{Poi}(n_l \times o_l) \quad l \in \{0, 1, 2, \ldots, 147\}
\]

Use \( N \) to denote the total number of cells and \( p_l \) to denote the percentage of cells in state \( l \), then the Poisson distribution mean can be rewritten as:

\[
n_l \times o_l = p_l \times N \times o_l = p_l \times \mu_l \quad \mu_l = N \times o_l
\]

\( \mu_l \) here can be understood as the mean counts if all cells in the population are in state \( l \) at genomic position \( g \). Since all the cells are independent with each other, the sum of Poisson distribution is again a Poisson distribution. So

\[
d_g \sim \mathcal{Poi}(\sum_{l=0}^{147} p_l \times \mu_l)
\]

This means read counts at each genomic location are convolutions of a known number of Poisson random variables. In other words, the counts are generated by convolving counts from different states instead of being generated by a single unknown state.

This is a simplified version of data generation mechanism. In reality, there are more than 148 states because of the existence of TFs and other DNA binding proteins. There could also be variability in the Bernoulli probability even if different cells are in the same state. There will also be influences from PCR amplification and biases. In addition, the probability of different states that belong to the same binding protein
(such as the different nucleosome states) should be related as well. However, this simplified version represents the core idea of the genomic data generation process. This generation process also points out that we may need to adopt the alternative modeling approach presented in the last section.

5.3 Future work and applications

5.3.1 Creating a new resource to describe genome-wide protein-DNA interaction landscapes

The work of Harbison et al. (2004) (and later MacIsaac et al. (2006)) has been the de facto standard resource on genome-wide TF binding events in S. cerevisiae. However, this resource is based on out-dated ChIP-chip datasets and TF binding was treated as a binary event. Binding information on nucleosomes is also missing. Our estimations of genome-wide protein-DNA interaction landscape are based on recent nuclease digestion data (and also incorporates information from PBM datasets). These genome-wide probabilistic view of protein binding can potentially released as an alternative resource of Harbison et al. (2004).

5.3.2 In silico screening for biologically important single nucleotide polymorphisms

Single nucleotide polymorphisms (SNP) are single nucleotide variations in certain genomic locations. SNPs have been shown to be able to influence phenotypes through changing the promoter activities (Shastry, 2009). However, there are also many SNPs that are neutral (Doniger et al., 2008). Therefore screening the genome for biologically important SNPs is important in understanding transcriptional variation among individuals or different strain variants.

Our model provides a powerful tool to study the influence of DNA sequence on the binding landscape, once appropriate model parameters are learned. We can therefore perform in silico mutations at each genomic location and calculate its influence on
protein binding landscape by comparing the model inference results before and after the mutation.

Such simulation studies will be extremely efficient and provide principled starting points for downstream experimental validations. Strope et al. (2015) sequenced and annotated genomes of 100 *S. cerevisiae* strains, providing an ideal venue for such a study. A ranked list of nucleotide mutations that change the protein binding landscape the most can be derived by running the model on all available strain genomes. Such change of binding landscape could provide insight on the phenotypic differences between different strains. Additional ChIP and nuclease digestion experiments can be performed on those strains to validate the model predictions. Based on model predictions and experimental validations, we could further collect transcription data, such RNA-seq data, to further study the influence of protein binding landscape on downstream transcription.

5.3.3 *Predicting transcription rate based on protein-DNA interaction landscape*

Transcription is regulated by the binding of transcription factors and nucleosomes, as well as by the basal transcription machinery. Given the predicted binding landscape, a number of features can be extracted or derived, such as the identify of binding proteins and their respective binding probability, the binding site distances to TSS, the length of the nucleosome free regions, etc. (see reviews of Weingarten-Gabbay and Segal (2014) and Levo and Segal (2014)). Those features can be built into predictive models to predict the transcription rate of downstream genes. Examples of past work include Segal et al. (2008), He et al. (2010), Raveh-Sadka et al. (2012) and Rajkumar et al. (2013). Our own lab is also actively pursuing this line of research. An accurate and holistic view of genome-wide protein-DNA interaction landscape will greatly advance this research direction.
5.4 Conclusions

Our work in this dissertation builds computational inference tools to infer genome-wide protein-DNA interaction landscape through jointly modeling high-throughput genomic data. Our results showed that our approaches are powerful in providing inferences that are consistent with data. This dissertation primarily studies the *S. cerevisiae* model organism. However, these tools and principles can be easily extended to study human cells. High-throughput techniques are taking increasingly important roles in understanding major human diseases, many of which are caused by malfunctions in transcriptional regulation. Our work in this dissertation will provide useful tools to integrate and interpret those high-throughput datasets.
Appendix A

Supplemental materials for Chapter 2
Appendix Figure A.1: Predicted binding profiles versus MNase-seq binding profiles. Shown here are the same six promoter regions shown in Figure 2.5. For each promoter, we show the comparison between processed large MNase-seq fragment coverage and predicted nucleosome profile (top left) and the comparison between processed small MNase-seq fragment coverage and predicted composite TF binding profile (top right). We also show these comparisons as scatter plots (bottom left and right, respectively). Since both the data and our predictions describe how a binding profile changes along the genome, the scatter plots should not be interpreted as traditional scatter plots (of independently sampled points). Rather, they are more appropriately interpreted as something akin to phase space plots, swept out as one traces along the genome.
Appendix Figure A.2: Comparison of in vivo and in vitro MNase-seq coverage. We also examined the severity of MNase bias by comparing the in vivo MNase-seq data we used with in vitro MNase-seq data on naked DNA from Deniz et al. (2011). Shown here are coverage plots for representative promoter regions from the 81 promoter regions used in this study. Considering the difference in sequencing depth, we do observe several overlapping coverage peaks between in vivo and in vitro data, such as those in YER055C, YDR023W, and YMR229C promoters. However, the majority of the in vivo signals cannot be explained by sequence bias. This is in line with our observation in Figure 2.8.
APPENDIX FIGURE A.3: Similar to Figure 2.9. Comparison of inference results using different settings to pre-process MNase-seq data. MNase-seq data were pre-processed using the seven settings described in Section 2.2.5. 50% posterior credible intervals of each fitted TF transition weight in log scale are shown here.
Bibliography


Courey, A. J. (2009), Mechanisms in transcriptional regulation, John Wiley and Sons, Malden, MA, USA.


Biography

Jianling Zhong was born in 1986 in the town of Tangjiang in Southern China. In 2009, he received his bachelor's degree in both Computer Science and Biotechnology from Huazhong University of Science and Technology in Wuhan, China. He completed a master’s degree in statistics in the Statistical Science Department of Duke University in May, 2015.

During his Ph.D. work at Duke, he contributed to the following publications:


He also earned a N.S.F. travel award for RECOMB 2014 and the Duke University Graduate School Fellowships (2010, 2011).