Statistical Methods to Detect Hierarchical Topological Domains in Chromatin

by

Yifan Zhou

A thesis submitted to Johns Hopkins University in conformity with the requirements for the degree of Master of Science.

Baltimore, Maryland

April, 2017

© Yifan Zhou 2017
All rights reserved
Abstract

Knowing the three-dimensional architecture of chromosomes in nuclear space is important in studying gene expression and regulation features. Hi-C, a technology based on chromosome conformation capture, could analyze the spatial organization of chromatin by quantifying the frequency of interactions between genomic loci that are nearby in nucleus in the format of an interaction matrix. There are highly self-interacting regions with rectangle shape in the contact map of interaction matrix which are called topological domains. Theses topological domains are directly related to biological functions. Most existing methods to identify topological domains focus on diagonal region in contact map and do not allow for overlapping hierarchical structure of topological domains. In this study, we introduce an algorithm to identify topological domains in both diagonal and off-diagonal regions with the hierarchical overlapping structure. The algorithm consists of three main parts: (1) generating domain boundaries using a changepoint detection method called Binary Segmentation and identifying potential domains, (2) filtering out non-qualified domains using multi-
ABSTRACT

ple t-tests, (3) reducing the number of domains using optimization and cross-validation. We conducted real data analysis on mouse embryonic stem (ES) cells and four groups of simulation studies. The results show the ability of the algorithm to estimate the Hi-C interaction matrix and identify overlapping domains. The algorithm performs better than other existing methods in detecting hierarchical three-dimensional topological domain structures.

Advisor: Dr. Hongkai Ji

Reader: Dr. Kasper Daniel Hansen
Acknowledgments

Firstly, I would like to thank my thesis advisor Dr. Hongkai Ji who inspired my original thesis idea and encouraged me to work in genomics field. During the last two years, I learned how to become a qualified researcher and think scientifically from him. I would also like to thank my thesis reader Dr. Kasper Daniel Hansen for his valuable comments on my thesis. I am also grateful for all of the faculty in the Department of Biostatistics for their teaching and guidance to the students, and all of the department staff for their service. Special thanks go to Jason Ji and Ding Ding from Hongkai’s lab who provided me with useful suggestions and encouragement when I have difficulties in research or thesis work. Finally, I must express my gratitude to my parents and boyfriend for providing me with unfailing support and continuous encouragement during the last two years. All accomplishments would not have been possible without them.
Contents

Abstract ii

Acknowledgments iv

List of Tables vii

List of Figures viii

1 Introduction 1

2 Method 5

2.1 Statistical Model ................................. 5

2.2 Algorithm ........................................ 6

2.2.1 Raw blocks Generation ...................... 7

2.2.2 Simple-filter .................................. 13

2.2.3 CV-filter .................................... 13
## CONTENTS

3 Result  

3.1 Real Data .......................................................... 17  
3.2 Simulated Data ..................................................... 23  

4 Conclusions and Discussion 27  

A Appendix 29  

Bibliography 33  

Curriculum Vitae 36
List of Tables

3.1 Table 3.1 ......................................................... 19
3.2 Table 3.2 ......................................................... 24
List of Figures

1.1 Figure 1.1 ......................................................... 3
1.2 Figure 1.2 ......................................................... 4

2.1 Figure 2.1 ......................................................... 8
2.2 Figure 2.2 ......................................................... 10

3.1 Figure 3.1 ......................................................... 18
3.2 Figure 3.2 ......................................................... 20
3.3 Figure 3.3 ......................................................... 21
3.4 Figure 3.4 ......................................................... 22
3.5 Figure 3.5 ......................................................... 23
3.6 Figure 3.6 ......................................................... 25
3.7 Figure 3.7 ......................................................... 26

A.1 Figure A.1 ......................................................... 30
A.2 Figure A.2 ......................................................... 31
A.3 Figure A.3 ......................................................... 32
Chapter 1

Introduction

In the cell nucleus, chromosomes form a complex three-dimensional (3D) spatial structure by bending, twisting and folding to allow for specific biological function. For example, many expressed genes will assemble in the area around transcription factors while other silent genes may exist at the nuclear periphery. The non-coding DNA regions such as cis-regulatory elements could regulate transcription of genes located far away from via DNA looping [14]. To better understand the genome function during gene expression, it is essential to study the spatial structure of chromosomes. Chromosome Conformation Capture (3C) is used to probe the frequency of interaction between two genomic loci in the nuclear space [1]. It has been shown by 3C experiments that transcription factors modulate the chromatin loops between genes and cis-regulatory element regions, and remote enhancers also loop to the genes regulated by them during expression [11]. Hi-C is a new chromosome conformation cap-
ture method to detect the 3D spatial organization of whole genomes by ligating fragments which are close in the cell nucleus and then using high-throughput sequencing to find the ligated sequence of fragments. The result of a Hi-C experiment is a genome-wide interaction matrix indicating the frequency of interaction for each fragment in the nucleus [10]. Figure 1.1 is an example of the contact map generated by a Hi-C interaction matrix for chromosome 2 in mouse embryonic stem (ES) cells. There are strong patterns of regions with high interaction frequency in the contact map (see rectangular blocks in bright color in Figure 1.1) which are termed as topological domains. The boundaries of these topological domains are formed by short segments in the contact map where interaction frequency changes rapidly (color changes rapidly in Figure 1.1) [2]. Since the spatial architecture structure of chromatin is highly related with genome function, identifying topological domains using a genome-wide Hi-C matrix became an interesting topic.

A variety of methods have been developed in the literature to identify topological domains. Dixon et al. use an indicator, Directionality Index (DI), to detect the change of frequency between the upstream and downstream regions of topological domains. Then a Hidden Markov Model (HMM) is used to find whether the changes are from upstream, downstream or none of them. The topological domains are identified as the regions between downstream bias locations and upstream bias locations [2]. Sexton et al. use a distance-scaling
CHAPTER 1. INTRODUCTION

Figure 1.1

factor to detect the regions with rapidly change of interaction frequency across the diagonal of contact matrix which are potential boundary regions. They then determine the location of topological domain boundaries using the maximum value in distance-scaling factors [13]. Filippova et al. develop a method called “Armatus” based on dynamic programming. Here the optimal and near-optimal domains are first constructed by a fixed resolution and then aggregated together with different resolution [3]. Levy-Leduc et al. develop another dynamic programming method called “HiCseg” to maximize the likelihood of block boundaries with the estimated number of domains [9]. However, most of the existing methods for identifying topological domains are focused on analyzing the diagonal regions of Hi-C contact map, and many of them do not allow for overlapping domains or multi-scale domains and ignore the hierarchical structure of the potentially overlapping topological domains. The hierarchical
overlapping patterns with domains in multiple scales are common in contact maps. In addition, off-diagonal regions could also contain topological domains (See Figure 1.2). For example, in Figure 1.1, there are topological domains with relatively strong block patterns located in the off-diagonal regions.

The hierarchical structure of the topological domains contains vital information for discovering unknown gene expression features [4], and the off-diagonal domains may reveal interactions between distant regulatory elements and genes. The purpose of this study is to develop a method to better identify multi-scale topological domains with hierarchical overlapping structure and capable of handling off-diagonal domains.
Chapter 2

Method

Given a Hi-C interaction matrix, our purpose is to identify all potential topological domains. In this chapter, we first introduce the statistical model used to estimate the interaction matrix. We then describe the algorithm to identify all potential domains. All Hi-C interaction matrices are log2 transformed.

2.1 Statistical Model

We assume the interaction matrix $M$ with dimensions $n \times n$ is formed by $K$ potential overlapping domains and noises. $M$ is a symmetric matrix ($M' = M$). Domains have rectangular shapes. Let \{D\} indicate a set including all topological domains to be identified, and the number of topological domains is $K$. Each domain is denoted by $D_k$ where $D_k \in \{D\}$ ($k = 1, 2, \cdots, K$) with its non-negative coefficient $\beta_k$ ($k = 1, 2, \cdots, K$ and $\beta_k \geq 0$), the coefficient for each domain represents its intensity of interaction in the nuclear space. Domains
CHAPTER 2. METHOD

location could be indicated by its coordinates of upper left corner \((x_0, y_0)\) and lower right corner \((x_1, y_1)\), for example \(D_k = \{(x_0, y_0), (x_1, y_1)\}\). The interaction matrix \(M\) could be melted into a \(n^2 \times 1\) vector \(Y\) where \(Y = (y_1, y_2, \cdots, y_{n^2})'\) and \(y_i \geq 0, i = 1, 2, \cdots, n^2\), then \(Y\) could be treated as the outcome variable and predicted by the following model:

\[
Y = X\beta + \epsilon
\]

where \(\beta = (\beta_1, \beta_2, \cdots, \beta_K)'\) is a \(K \times 1\) vector. \(X\) is the design matrix with dimensions \(n^2 \times K\) indicating whether one element is covered by a specific domain, for example: \(X_{ij} = 1\) means element \(y_i\) in \(Y\) is covered by domain \(D_j\) in the contact map. And \(\epsilon\) are noises following normal distribution with mean 0.

The question to be solved in this study is: Given an \(n \times n\) matrix \(M\) (or its melted format \(Y\)), according to \(Y = X\beta + \epsilon\) with restriction of \(y_i \geq 0\) and \(\beta_k \geq 0\), how to estimated the number of topological domains \(K\) and design matrix \(X\), along with the coefficients \(\beta\). Once we could estimate \(X\), the domains location \(D_k = \{(x_0, y_0), (x_1, y_1)\} (k = 1, 2, \cdots, K)\) would be obtained.

2.2 Algorithm

The algorithm consists of three main steps: Raw blocks generation, simple-filter and CV-filter. First it generates a large number of raw blocks as the potential domains. Then it runs the simple-filter step based on t-test to select
CHAPTER 2. METHOD

a subset from the raw blocks set. The final estimated domains are selected by the CV-filter based on optimization and cross-validation.

2.2.1 Raw blocks Generation

In this step, the algorithm first detects a set of potential domain boundaries by Binary Segmentation changepoint detection method. We then use the intersections of those segments as the potential upper left and lower right corners to derive the raw blocks.

Changepoint detection

Considering only one specific row in the contact map (the blue dashed line in Figure 2.1), the intensity of interaction along this row will change abruptly on domain boundaries. Those locations of abruptly changing are called changepoints (the changepoint locations are highlighted by the yellow lines in Figure 2.1), and there are several established methods to detect them. In this study, we implement the Binary Segmentation proposed by Scott and Knott (1974) [12] to detect the changepoints in all rows and columns in matrix M.

Suppose one row in Hi-C matrix $M$ could be denoted by an ordered sequence of data $M_{row} = (m_1, m_2, ..., m_n)$, and the number of true changepoints in $M_{row}$ is $S$, together with their positions $\tau = (\tau_1, \tau_2, ..., \tau_S)$ where each position in $\tau$ is an integer between 1 and $n - 1$, and $\tau_i < \tau_j$ if and only if $i < j$. Then $S$ changepoints will split the data $M_{row}$ into $S + 1$ segments, and the $i$-th segment
Figure 2.1: The lower plot is the result of changepoint detection for the row of the dashed blue line in the upper plot. Yellow lines in the lower plot indicate the locations of detected changepoints.
CHAPTER 2. METHOD

is denoted by $m_{(\tau_1 + 1)} : m_{\tau_1}$. The approach of Binary Segmentation to detect multiple changepoints is to solve this problem:

$$
\text{minimize } \sum_{i=1}^{S+1} [C(m_{(\tau_1 + 1)} : m_{\tau_1})] + \beta S
$$

where $C(\cdot)$ is a cost function for a segment and $\beta S$ is the penalty term. The simple idea of Binary Segmentation method is iteratively applying single changepoint method to each segments of the sequence until satisfying a specific criteria. The initial step is searching a single changepoint on the entire sequence $M_{row} = (m_1, m_2, ..., m_n)$ to test if there exists a $\tau$ which satisfies

$$
C(m_1 : m_\tau) + C(m_{(\tau+1)} : m_n) + \beta < C(m_1 : m_n).
$$

If such a $\tau$ could not be found the method will end, otherwise the data is split into two segments: $m_1 : m_{(\tau-1)}$ and $m_\tau : m_n$, and the detection method will be applied to each of them. This procedure is repeated until no further changepoints are found or the number of changepoints is greater than or equals to $Q$ [7].

In our algorithm, we apply the Binary Segmentation method to all rows and columns in $M$ and obtain changepoint positions $\tau$ for each row and column. Since $M$ is a symmetric matrix, the detected changepoints in the $i$-th row and the $i$-th column are the same, denoted by $\tau^{(i)}$. We use R package “changepoint”
CHAPTER 2. METHOD

Figure 2.2: Black dots are the changepoints detected by Binary Segmentation method.

In our algorithm and set $Q = \left\lfloor \frac{n}{15} \right\rfloor$ for raw Hi-C interaction matrix and $Q = \left\lfloor \frac{n}{20} \right\rfloor$ for normalized Hi-C interaction matrix. Figure 2.2 shows the changepoint detection results for all rows in a contact map.

Creating boundary using the changepoints

From the changepoint detection result in Figure 2.2, it appears most of the changepoints arrange in nearly straight lines and those lines highly overlap the domain boundaries. Therefore, the potential boundaries are generated by linking nearby changepoints.

Suppose there are $n$ changepoint positions set $\tau^{(i)}$, $i = 1, ..., n$ for $n$ rows and $n$ columns in $M$, we use run length encoding to generate vertical domain boundaries using changepoints found by row, and generate horizontal domain boundaries using changepoints found by column. We only describe the method
CHAPTER 2. METHOD

to generate vertical boundary using changepoints found by row since $M$ is symmetric.

For the vertical boundary, $S - 1$ uniform distributed vertical lines will split matrix $M$ into $S$ sub-matrixes with same dimensions $n \times w$ ($w = 4$ in our algorithm). Each of the sub-matrix can be denoted by $M_{[s(w-1)+1]:sw}$ where $s = 1, ..., S$, then

$$M = \begin{pmatrix} M_{1:w} & M_{(w+1):2w} & \cdots & M_{(n-w+1),n} \end{pmatrix}.$$ 

For the $s$-th sub-matrix $M_{a:b}$ where $a = s(w - 1)$ and $b = sw$, we define an indicator vector $I(a, b) = (I_1, I_2, ..., I_n)$ for the row changepoints where

$$I_i = \begin{cases} 
1, & \text{if there exist at least one } \tau_j^{(i)} \in \tau^{(i)}(j = 1, ..., Q(i)) \text{ satisfying } a \leq \tau_j^{(i)} \leq b \\
0, & \text{otherwise}
\end{cases}$$

where $Q(i)$ is the number of changepoints detected in the $i$-th row. Then $I(a, b)$ is a binary sequence and could be represented by run length encoding using two vector: value and length. For example, if the value of $I(a, b)$ is $V(a, b) = (0, 1, 0, 1, 0, 1)$ and the length is $L(a, b) = (4, 1, 5, 10, 2, 5)$, then $I(a, b)$ could be interpreted as a sequence of four “0”, one “1”, five “0”, ten “1”, two “0” and five “1”. Joining changepoints into boundary is the same as linking continuous “1” in $I(a, b)$, while there are some noises in the sequence like the one “1” between
CHAPTER 2. METHOD

four “0” and five “0”, also like the two “0” between ten “1” and five “1”. Before linking the same value in \( I(a, b) \), we use this procedure to denoise:

1. Set the initial minimum length of the critical value \( c = c_0 \).

2. For sequence \( I(a, b) \) with length \( L(a, b) = (l_1, l_2, \ldots, l_{n_0}) \) and value \( V(a, b) = (v_1, v_2, \ldots, v_{n_0}) \), if \( l_i \leq c \), then \( v_i = v_{i-1} \) \((i \geq 2)\).

3. Reconstruct the new sequence \( I(a, b) \) using the results in step 2.

4. Set new critical value \( c = c + 1 \) and go back to step 2.

5. Repeat step 2-4 until \( c = C \).

In this study, we use \( c_0 = 3 \) and \( C = 6 \). After denoising, the continuous “1” in the denoised \( I(a, b) \) are linked into several segments. After applying this method to all sub-matrix \( M_{[s(w-1)+1]:sw} \) \((s = 1, \ldots, S)\), potential vertical boundary as a set of segments \( \{Seg_v\} \) is generated. Using the same way to split \( M \) by the horizontal lines and linking all continuous “1”, potential horizontal boundary \( \{Seg_h\} \) could be obtained. The intersections of those two sets of segment are the initial potential corners: \( \{D_0\} = \{Seg_v\} \cap \{Seg_h\} \). One can then determine whether an intersection is an upper left corner or lower right corner by comparing the elements in its upper left region with the elements in its lower right region (the region is a \( 3 \times 3 \) square in this algorithm).

Suppose there are \( n_1 \) upper left corners and \( n_2 \) lower right corners found in this step, there will be \( \tilde{K}^{(0)} = n_1 \times n_2 \) raw blocks generated as \( \tilde{D}^{(0)} \).
CHAPTER 2. METHOD

2.2.2 Simple-filter

After the “Raw blocks generation” step, we have $\tilde{K}^{(0)}$ raw blocks as the potential domains. For each block, we run four t-tests to determine whether each boundary is a reasonable boundary. For example, if the mean of the elements in the left side of the vertical boundary is smaller than the mean of the elements in the right side of the boundary, we consider this vertical boundary as a reasonable left boundary of one domain. In this study, we use the elements in the rectangle region with two columns as rectangle’s width on the two sides of the left or right boundary for the t-test (rectangle’s height is the same as the length of boundary), and use the elements in the rectangle regions with two rows as rectangle’s height on the two sides of the top or bottom boundary (rectangle’s width is the same as the length of boundary). This step will reduce the number of potential domains to $\hat{K}^{(1)}$ ($\hat{K}^{(1)} \leq \tilde{K}^{(0)}$), and the remaining blocks are denoted by set $\{\hat{D}^{(1)}\}$.

2.2.3 CV-filter

The simple-filter in the previous step outputs $\hat{K}^{(1)}$ selected blocks denoted by $\{\hat{D}^{(1)}\}$. Then the initial design matrix $\hat{X}^{(1)}$ could be generated based on $\{\hat{D}^{(1)}\}$ and $M$. The statistical model described in Chapter 2.1 could be written as $Y = \hat{X}^{(1)} \beta + \epsilon$, then the problem to be solved becomes an optimization problem (we use “Gurobi Optimization” in R [5] to solve the optimization prob-
CHAPTER 2. METHOD

lems):

\[
\begin{align*}
\text{minimize} & \quad \| \hat{X}^{(1)} \beta - Y \|^2 \\
\text{subject to} & \quad \beta_i \geq 0, \ i = 1, 2, \ldots, \hat{K}^{(1)}.
\end{align*}
\]

The optimization result is \( \hat{\beta}^{(1)} \) which is the initial estimation of \( \beta \). Then, cross-validation is implemented to further reduce the number of blocks. After cross-validation, the estimated number of domains is \( \hat{K} \), meanwhile the estimated locations of \( \hat{D} \) and \( \hat{X} \) could be obtained. Finally, by solving

\[
\begin{align*}
\text{minimize} & \quad \| \hat{X} \beta - Y \|^2, \ \text{subject to} \ \beta_i \geq 0, \ i = 1, 2, \ldots, \hat{K}
\end{align*}
\]

the coefficients \( \hat{\beta} \) can be estimated.
Chapter 3

Result

In this study, we conducted real data analysis and Monte Carlo simulations. We obtained the Hi-C data of mouse ES cells from the website of “Hi-C project at Ren Lab” (http://chromosome.sdsc.edu/mouse/hi-c/download.html, GEO accession: GSE35156) and analyzed the interaction matrix of chromosome 2 and chromosome 16. The simulated data were generated by the estimated domains \{\hat{D}\} of chromosome 16 with their coefficients \hat{\beta} using the model \( Y = \hat{X} \hat{\beta} \), where \( \hat{X} \) is generated by \{\hat{D}\}. Detailed steps of generating simulated data are described in “3.2 Simulation Data”. We used 80 kb as the window size in all studies and limited the length of domain boundaries between 400kb (five times of window size) and 800kb (100 times of window size) considering computing power.

To understand the ability and efficiency for identifying significant domains in our algorithm, we introduce a term “Identification Rate” (IR) for a domain
set \{D'\} given a reference domain set \{D^R\}. IR given an allowable range of errors \(r\) is denoted by \(IR(r)\). The IR for \{D'\} in reference to \{D^R\} is defined below:

where \#\{\text{\textit{D}_\text{\textit{identifiable}}}\} is the number of domains in \{D'\} that can be cover by the reference domain set \{D^R\}

Let \(r\) denotes the “tolerance rate”. For one domain \(D'_i = \{(x'_0, y'_0), (x'_1, y'_1)\}\) where \(D'_i \in \{D'\}\) and \(i \in \{1, 2, \cdots, K'\}\), if there exists at least one reference domain \(D^R_j = \{(x^R_0, y^R_0), (x^R_1, y^R_1)\}\) satisfying

\[
(x'_0 - x^R_0)^2 + (y'_0 - y^R_0)^2 \leq r^2 \quad \text{and} \quad (x'_1 - x^R_1)^2 + (y'_1 - y^R_1)^2 \leq r^2
\]

where \(D^R_j \in \{D^R\}\) and \(j \in \{1, 2, \cdots, K^R\}\) (\{\text{\textit{D}}^R\} is the reference domain set), then domain \(D'_i\) is a identifiable domain. The IR given a tolerance rate \(r\) is the percentage of identifiable domains in domain set \{D'\}:

\[
IR(r) = \frac{\#\{\text{\textit{D}_\text{\textit{identifiable}}}\}}{\#\{D'\}}
\]

where \{\text{\textit{D}_\text{\textit{identifiable}}}\} is the collection of all identifiable domains. Fig 3.1 is a simple graphic explanation for the concept of IR.

Let \{\text{\textit{D}}\} indicate the true topological domain set, and \{\hat{\text{\textit{D}}}\} indicate the estimated domain set obtained from the algorithm. The IR for \{\text{\textit{D}}\} in reference to \{\hat{\text{\textit{D}}}\} is the percentage of true domains that can be identified by the estimated
domains, which is the sensitivity of the identification method. The IR for \( \{\hat{D}\} \) in reference to \( \{D\} \) is the percentage of estimated domains obtained from the algorithm that can be covered by the true domains, which is \( 1 - \text{false discovery rate (1-FDR)} \) of the identification method. In addition, \( 1 - \text{FDR} \) is a measure of the precision.

The reason for allowing a range of tolerance rate \( r \) is the noises in the interaction matrix make the boundaries not clear enough to be located. In addition, since domain boundaries are formed by short segments in the contact map where interaction frequency changes rapidly [2], each boundary’s location should be a range of number instead of a specific location such as the \( i \)-th row or the \( j \)-th column in the contact map. Therefore, measuring sensitivity and precision under a tolerance rate is reasonable.

### 3.1 Real Data

Both raw interaction matrix and normalized matrix were analyzed in this study. The elements in raw interaction matrix are log2 transformed read counts between any two loci in whole genome obtained by the Hi-C experiments. The normalized interaction matrix is the generated from the raw interaction matrix by implementing the normalization method developed by Yaffe and Tanay (2011) [15]. This normalization method reduces systematic biases which are distance between restriction sites, the GC content of trimmed ligation junctions and sequence uniqueness. For each interaction matrix \( M \), we found \( \hat{K} \)
CHAPTER 3. RESULT

Figure 3.1: The blue block is a domain \( D'_i \in \{ D' \} \). The gray circular area is the tolerance area, and its radius is the tolerance rate \( r \). The block surrounded by black lines is one of the reference domain \( D'_r \in \{ D' \} \) which can be used to identify the reference domain \( D_i \).

estimated domains \( \hat{D}_k \in \{ \hat{D} \} \) using the algorithm describe in Chapter 2, along with the estimated coefficient \( \hat{\beta}_k \) \((k = 1, 2, \cdots, \hat{K})\). Therefore, the estimated values of the elements \( \hat{y}_i \) \((i = 1, 2, \cdots, n^2)\) in matrix \( M \) could be calculated by \( \hat{Y} = \hat{X}\hat{\beta} \), where \( \hat{X} \) was generated by \( \hat{D}_k \). The mean squared error (MSE) could be driven by \( MSE = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2/n^2 \) and the R squared \((R^2)\) is also calculated by \( R^2 = 1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(y_i - \bar{y})^2} \). The summarized results are shown in Table 3.1. We could use the estimated domains \( \{ \hat{D} \} \) and their coefficients \( \hat{\beta}_k \) to predict the Hi-C interaction matrix \( M \). Figure 3.2 and Figure 3.3 show the prediction results of raw interaction matrix comparing with the original contact maps, and the differences between original matrix and predicted matrix which are residuals. The result of normalized matrix is in Appendix A (see Fig-
CHAPTER 3. RESULT

Figure A.1, Figure A.2). The $R^2$ for both of the two chromosomes are reasonable and most of the domains pattern are visible in predicted contact maps, which demonstrates the algorithm has the ability to identify topological domains in real Hi-C data. The results of normalized matrices have smaller MSE compared with raw matrices. The selected domains are highlighted in contact map which could be found in Figure A.3 from Appendix.

Table 3.1: Result of chromosome 2 and 16

<table>
<thead>
<tr>
<th></th>
<th>chr2</th>
<th>chr16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Raw</td>
<td>Normalized</td>
</tr>
<tr>
<td>Number of Selected Domains</td>
<td>1530</td>
<td>1299</td>
</tr>
<tr>
<td>MSE</td>
<td>2.0382</td>
<td>0.6047</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.7321</td>
<td>0.8183</td>
</tr>
</tbody>
</table>

To compare with the established method “Directionality Index” (DI) developed by Dixon et al. in 2012, the domains selected by DI method are denoted by $\{D^{(DI)}\}$, and we calculated the IR for $\{D^{(DI)}\}$ in reference to $\{\hat{D}\}$, where $\{\hat{D}\}$ is the estimated domain set from our algorithm. The IR for $\{D^{(DI)}\}$ in reference to $\{\hat{D}\}$ is a measure of sensitivity when assuming DI selected domains are true topological domains. We also calculated the IR for $\{\hat{D}\}$ in reference to $\{D^{(DI)}\}$, which is the measure of precision assuming DI selected domains are true domains. Since Dixon et al. analyzed normalized interaction matrices, the comparisons were only based on our results for normalized matrix of chromosome 2 and 16. The IR was calculated with tolerance rate from $r = 80kb$ (the window size) to $r = 1600kb$ (20 times of the window size). The results are
CHAPTER 3. RESULT

Figure 3.2: Chromosome 2 raw matrix. (a) Contact map. (b) Prediction result. (c) Residuals.
CHAPTER 3. RESULT

Figure 3.3: Chromosome 16 raw matrix. (a) Contact map. (b) Prediction result. (c) Residuals.
CHAPTER 3. RESULT

Figure 3.4: Comparison with DI method developed by Dixon et al. (2012) for chromosome 2 normalized matrix, when assuming DI selected domains are true domains. (a) Sensitivity: IR for \( D^{(D)} \) in reference to \( \hat{D} \). (b) Precision: IR for \( \hat{D} \) in reference to \( D^{(D)} \).

showed in Figure 3.4 and Figure 3.5. When \( r = 400 \text{ kb} \), about 80% of the domains in \( D^{(D)} \) could be identified by \( \hat{D} \) for both chromosome 2 and 16, and when \( r = 800 \text{ kb} \), about 90% of them are identifiable. Those results indicate relatively good sensitivity of the algorithm. However, the precision is not satisfying when assuming DI method selected domains are true domain. One of the reasons for that is the number of estimated domains in \( \hat{D} \) is much larger than the number of domains in \( D^{(D)} \).
CHAPTER 3. RESULT

Figure 3.5: Comparison with DI method developed by Dixon et al. (2012) for chromosome 16 normalized matrix, when assuming DI selected domains are true domains. (a) Sensitivity: IR for \(\{D^{(DI)}\}\) in reference to \(\{\hat{D}\}\). (b) Precision: IR for \(\{\hat{D}\}\) in reference to \(\{D^{(DI)}\}\).

3.2 Simulated Data

We conducted four groups of simulation studies. The simulated data were generated based on the results of real data analysis of chromosome 16 in section 3.1, using the estimated domains and estimated coefficients as the reference domains \(\{D\}\) and true coefficients \(\{\beta\}\), and then adding normal noises \(\{\epsilon\}\). The number of reference domains is \(K\) and \(K = 700\). For the noises \(\{\epsilon\}\), we divided the elements \(y_i\) \((i = 1, 2, \ldots, n^2)\) into five groups according to their value: \(y_i^{(1)} \in [0, 2), y_i^{(2)} \in [1, 3), y_i^{(3)} \in [3, 5), y_i^{(4)} \in [5, 10), y_i^{(5)} \in [10, +\infty)\). Let \(\sigma^{(s)}\) \((s = 1, 2, 3, 4, 5)\) denotes the standard error of the residuals obtained from the real data analysis in section 3.1. Then the new noises \(\{\epsilon\}\) were generated
within each element group according to the following distribution

\[ \epsilon^{(s)} \sim N\left(0, \left(\frac{\sigma^{(s)}}{p}\right)^2\right), \quad s = 1, 2, 3, 4, 5 \]

where \( p = 1, 2, 3, 4 \) in the four groups of simulation. For each simulation group with different noise level, we synthesized 30 independent datasets in the format of an interaction matrix \( M \).

For each independent dataset \( M \), we implemented our algorithm to obtain the estimated domains \( \hat{D} \) with their coefficients \( \hat{\beta} \) and the number of domains \( \hat{K} \). All elements in the interaction matrix \( M \) could be estimated by \( \hat{Y} = \hat{X}\hat{\beta} \), then the mean squared error could be calculated. We could compare the simulation results across different groups by summarizing the mean squared error and estimated number of domains \( \hat{K} \) within each group (see Table 3.2).

Table 3.2: Result of Simulation Study

<table>
<thead>
<tr>
<th>Group</th>
<th>Mean of MSE</th>
<th>CI of mse</th>
<th>Mean of ( \hat{K} )</th>
<th>CI of ( \hat{K} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.32</td>
<td>(0.16, 0.49)</td>
<td>1043</td>
<td>(848, 1237)</td>
</tr>
<tr>
<td>2</td>
<td>0.16</td>
<td>(-0.01, 0.32)</td>
<td>1774</td>
<td>(1508, 2039)</td>
</tr>
<tr>
<td>3</td>
<td>0.14</td>
<td>(-0.04, 0.32)</td>
<td>2170</td>
<td>(1876, 2465)</td>
</tr>
<tr>
<td>4</td>
<td>0.07</td>
<td>(-0.10, 0.23)</td>
<td>2537</td>
<td>(1926, 3148)</td>
</tr>
</tbody>
</table>

For each simulation results \( \{\hat{D}\} \), we calculated the IR for true domains \( \{D\} \) in reference to the selected domain \( \{\hat{D}\} \) to measure the sensitivity of the algorithm, and calculate the IR for selected domains \( \{\hat{D}\} \) in reference to the true
CHAPTER 3. RESULT

Figure 3.6: Smooth Curves of IR. (a) Sensitivity: IR for \( \{D\} \) in reference to \( \{\hat{D}\} \). (b) Precision (1–FDR): IR for \( \{\hat{D}\} \) in reference to \( \{D\} \).

domains \( \{D\} \) to measure the precision, which is 1–FDR. Figure 3.6 contains the smooth curve of IR within each simulation group. Figure 3.7 contains the original contact map and analysis result of one dataset in the fourth simulation group with \( p = 4 \). From results of simulation study, we could see the mean squared error would be smaller in group with lower noise rate, but the number of selected domains will increase when noise level goes down. When the tolerance rate is 20, the algorithm could identify over 75% of the topological domains in all simulation groups (in group 3 and 4, it could identify over 85% domains). When \( r = 800kb \), the algorithm could identify over 75% of the topological domains for three groups with \( p = 2, 3, 4 \). The smooth curves of IR for \( \{D\} \) in reference to \( \{\hat{D}\} \) show relatively high sensitivity of the algorithm, but the smooth curves of IR for \( \{\hat{D}\} \) in reference to \( \{D\} \) illustrate the relatively
CHAPTER 3. RESULT

low precision of the algorithm for simulated data.

Figure 3.7: Results for one dataset in the fourth simulation group. (a) Original contact map for \{\hat{D}\}. (b) Prediction result of \{\hat{D}\}. (c) Unidentifiable domains in original domains \{D\} when \(r = 800\text{kb}\). (d) Residuals.
Chapter 4

Conclusions and Discussion

In this study, we introduce an algorithm to identify topological domains in both diagonal and off-diagonal regions with the hierarchical overlapping structure. The algorithm includes three main steps to find the boundary, filter blocks and determine final selected domains using statistical methods such as change-point detection, mathematical optimization and cross-validation. In both real data analysis for mouse ES cells and simulation studies, the algorithm has shown good ability to estimate Hi-C interaction matrix and identify overlapping domains in hierarchical structure. Compared with the DI method developed by Dixon et al. in 2012, the identification results using our algorithm will cover most of the selected domains in that method. Moreover, the algorithm we developed in this study will also identify off-diagonal topological domains and overlapping domains which are not allowed in most of the existing methods including DI.
CHAPTER 4. CONCLUSIONS AND DISCUSSION

Due to the computing power issue, the size of selected domains is limited by the length of domain boundaries between 400 kilo-bases and 800 kilo-bases which ignores potential topological domains in larger scales. Therefore, future work is to reduce the computing time cost of the algorithm to allow for more domains, even as large as a whole chromosome. In addition, this study only analyzed intrachromosomal interaction matrixes, but here may exist interchromosomal topological domains along the whole genome between chromosomes closely located in 3D structure in nuclear space. Therefore, it is important to analyze the interaction matrix across the whole genome.
Appendix A

Appendix

Figure A.1 and Figure A.2 show the prediction results along with the contact map of normalized interaction matrix of chromosome 2 and 16.
Figure A.1: Chromosome 2 normalized matrix. (a) Contact map. (b) Prediction result. (c) Residuals.
Figure A.2: Chromosome 16 normalized matrix. (a) Contact map. (b) Prediction result. (c) Residuals.
Figure A.3: Selected blocks in real data analysis. (a) Chromosome 2 raw matrix. (b) Chromosome 2 normalized matrix. (c) Chromosome 16 raw matrix. (d) Chromosome 16 normalized matrix.
Bibliography


BIBLIOGRAPHY


Curriculum Vitae

Yifan Zhou was born on March 12, 1993 in Zibo, China. She received a Bachelor of Science in Statistics from Zhejiang University in June 2015, in Hangzhou, China. After college graduation, she attended Johns Hopkins Bloomberg School of Public Health for the ScM program in Biostatistics. During her graduate study, she worked as a research assistant in the Department of Biostatistics and participated in several research projects in genomics field. She was also a teaching assistant during academic year 2016 to 2017 for the courses Statistical Methods in Public Health I-IV. She graduated from Johns Hopkins University with Master of Science in Biostatistics in May 2017, in Baltimore, MD.