Incorporating Chromatin Interaction Data to Improve Prediction Accuracy of Gene Expression

by

Xue Li

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Bioinformatics and Computational Biology

May 2015

APPROVED:

Dr. Zheyang Wu, Advisor

Dr. Dmitry Korkin, Reader

ABSTRACT

Genome structure can be classified into three categories: primary structure, secondary structure and tertiary structure, and they are all important for gene transcription regulation. In this research, we utilize the structural information to characterize the correlations and interactions among genes, and involve such information into the Linear Mixed-Effects (LME) model to improve the accuracy of gene expression prediction. In particular, we use chromatin features as predictors and each gene is an observation. Before model training and testing, genes are grouped according to the genome structural information. We use four gene grouping methods: 1) grouping genes according to sliding windows on primary structure; 2) grouping anchor genes in chromatin loop structure; 3) grouping genes in the CTCF-anchored domain; and 4) grouping genes in the chromatin domains obtained from Hi-C experiments. We compare the prediction accuracy between LME model and linear regression model. If all chromatin feature predictors are included into the models, based on the primary structure only (Method 1), the LME models improve prediction accuracy by up to 1%. Based on the tertiary structure only (Methods 2-4), for the genes that can be grouped according the tertiary interaction data, LME models improve prediction accuracy by up to 2.1%. For individual chromatin feature predictors, the LME models improve from 2% to 26 %, in which improvement is more significant for chromatin features that have lower original predictive ability. For future research we propose a model that combines the primary and tertiary structure to infer the correlations among genes to further improve the prediction.

ACKNOWLEDGEMENTS

This project takes nearly one year and is supported by Dr. Zheyang Wu, my academic advisor in Worcester Polytechnic Institute. I wish to express my sincere thanks to Dr. Zheyang Wu for all the patient guidance, kindly suggestions and continuous encouragements during my master study.

I am grateful to Dr. Zhiping Weng's group in UMASS Medical Schoool, especially Dr. Xianjun Dong, who was a post-doctor in Weng's lab and now is a faculty member in Brigham and Women's Hospital. My project was first started by Dr. Dong and Dr. Wu two years ago and they did a lot of work about the data pre-processing part.

I am grateful to Dr. Dmitry Korkin, the reader of my master thesis, for the time and energy he spent on my thesis.

I take this opportunity to express gratitude to all of faculty members in Bioinformatics and Computational Biology Program for their help and support.

Contents

1 Introduction

1.1 Gene expression

Gene expression is a very important biological process in which genetic information is transferred from DNA to protein. Gene expression includes two main steps: transcription and translation. In the transcription step, a specific segment of DNA will be copied through RNA polymerase into an RNA chain, which contains the gene code to be further translated. In the translation step, RNA is used as a template to generate polypeptide chain that will finally fold into proteins. The proteins can be enzymes that catalyze biological process, or structural proteins that control the organism's structure. Gene expression guarantees that the genotype rises to the phenotype, and genetic code is expressed and inherited.

1.2 Gene regulation

Gene regulation controls gene expression and plays a critical role in all aspects of growth and reproduction of organisms. Human are originated from zygote, which produces many embryonic cells through mitosis, and embryonic cells differentiate into different types of cells. During these symphonic processes genes cannot be expressed at the same level all the time; they need to be conducted and adjusted by gene regulation. Gene regulation not only maintains the normal functions in cell, but also helps cells to respond immediately under stimulation. For example, when a cell is infected by virus or bacteria, the immune-related genes will be activated and transcribe many copies of RNA chain, which will translate into immune-related proteins and protect the cell from the damage (Saccani and Trabucchi 2015).

Gene regulation has two major mechanisms, one is transcription and the other is post-transcription.

1.2.1 Transcriptional regulation

Transcriptional regulation is the process by which cells can regulate the transcription from DNA to RNA. It happens at the DNA level and directly influences gene activity. It determines whether one gene is on or off and how many copies of RNA should be transcribed. Transcription factors (TF) and histone modifications (HM) are the key regulators in transcriptional regulation.

Transcription factors can bind to specific DNA elements and influence the initiation of gene transcription. RNA polymerase II is the most famous TF, which can recognize the TATA box in promoter region and start the transcription (Sainsbury et al. 2015). Transcription factor can also bind to *cis*-regulatory elements, like enhancer and silencer, and influence the transcription through long-range interactions (Hwang et al. 2013).

In addition to transcription factors, histone modifications also play an important role in transcription regulation. Histones are highly alkaline proteins, DNA wraps around the core histones and constitute nucleosome. The modification often happens on core histone H3 and H4 since they have long tails outside the nucleosome which can be modified by methyl or ethyl molecules (Barski et al. 2007). The modifications can change the histone structure, influence their interactions with DNA and expose DNA to other transcription factors. Different modifications can function in a very different way. H3K4me1 and H3K4me3 have been proved as positive marks and activate the transcription (Koch et al. 2007). However, H3K9me3 and H3K27me3 have been shown as negative marks and repress the transcription (Hublitz et al. 2009).

1.2.2 Post-transcriptional regulation

Post-transcriptional regulation occurs after transcription, controls the regulation at RNA level. It is more complex and contains many strategies that can be classified into two categories. Category I includes strategies: polyadenylation, capping and alternative splicing, which contribute to the stabilization of RNA and help RNA to be translated (Izaurralde 2002). Polyadenylation adds ploy-A tail to a messenger RNA and contributes to the stabilization of RNA. It also helps RNA transcript to be recognized by nuclear membrane protein and transported to cytoplasm. Capping can change 5-prime end to 3-prime end and protect RNA transcript from degradation. Alternative splicing can remove introns, even some of exons, and make RNA transcript be able to be translated. Category II contains only one strategy: RNA interference, which will destruct RNA chain and inhibit gene expression (Saurabh et al. 2014). This is for sequence-specific gene regulation, microRNA or small interfering RNA can bind to mRNAs and inhibit it from translating.

Here, we mainly focus on the transcription regulation since it is a universal mechanism while post-transcriptional regulation is more complex and meticulous. We have very mature technique like ChIP-Seq to measure protein-binding signals and histone modification signals, but it is harder to detect the overall protein-binding situation to RNA. Although ribosome profiling has been used to study which mRNAs are being activated (Ingolia 2014), we still don't know when and how they go through capping, polyadenylation, or alternative splicing.

1.3 Previous methods to study gene regulation

1.3.1 Linear regression model

Linear regression model is expressed in the following formula.

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} \dots \beta_k X_{ik} + \varepsilon_i.$$

For the *i*th gene, the response variable Y_i measures the gene expression level, $X_{i1}, \dots, X_{i\mathrm{k}}$ are the independent variables of the signals of *k*th gene regulation factors within the location (with proper definition) of this gene, β_0 is the parameter of the baseline gene expression level, $\beta_1, \dots, \beta_{\mathrm{k}}$ are the coefficients of the *k*th regulatory factors contributing to the gene expression, ε_i is the error term for the random variation of gene expression level. The model assumption is that $\varepsilon_i, i = 1, \dots, n$, are independently follow $N(0, \sigma^2)$. All the coefficients $\beta_{\mathrm{j}}, j = 0, \dots, k$, and the variance σ^2 are the unknown parameters to be estimated by data.

Cheng et al. first related transcription factor binding and histone modification signals to gene expression level through linear regression model in mouse embryonic stem cells (ESC) and achieved a high prediction accuracy at 0.77 (Cheng and Gerstein 2012). The principle is that transcription factor binding signals are predictive of gene expression levels. In mouse ESC cell, signal profiles of most TFs exhibiting a peak at the transcription start sites (TSSs). These binding signals and gene expression levels have a positive correlation especially at the TSSs. In contrast, HM signals show different signal profiles and correlation patterns from one another. Positive histone marks like H3K4me1, H3K4me2 and H3K4me3 are similar to TFs; they show strong signals around TSSs and are positively correlated with gene expression levels. Negative histone marks like H3K27me3 also exhibit strong signals around TSSs but are negatively correlated with gene expression. From all these knowledge, Cheng et al. built a linear regression model to study the relationship. The TF binding signals around TSS region are participated into 40 bins with length 100bp (Figure 1). The average signals at bin *j* from TFs are used as input. Following the formula above, all unknown parameters are estimated and predicted gene expression level is

obtained. The prediction accuracy for bin *j* is the correlation between predicted and observed gene expression level. TF binding signals from different bins have different prediction accuracy and the bin besides the TSS achieves the best prediction.

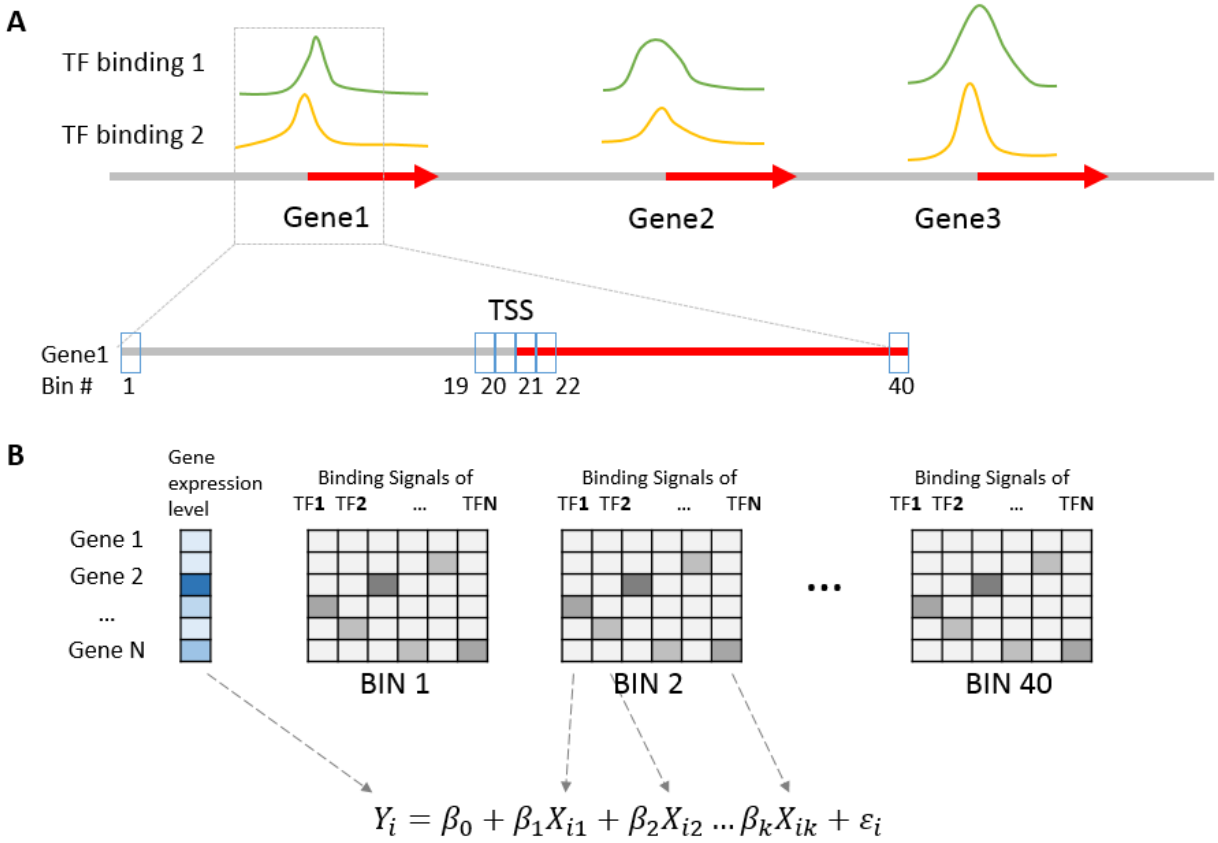

Figure 1. Illustration of using linear regression to model gene expression regulation. (a) Transcription factor binding signals around TSSs are obtained from Chip-Seq experiment. TSS region are participated into 40 bins with 100bp long in each bin. The binding signals from each transcription factor at gene *i* is the average signals in the bin. (b) The signals in each bin is a dataset, for bin *j*, predictors are binding signals from transcription factors, response variables are gene expression level. The prediction accuracy is calculated from predicted and observed gene expression levels.

In 2012, Dong et al. took advantage of ENCODE project and explored this relationship between chromatin features and gene expression levels from more than 100000 promoters measured in different cell compartments from seven human cell lines (Dong et al. 2012). They also used linear regression model but the input is a little different. They didn't involve TF binding signals from all bins but select a best bin instead. The best bin can be the maximum signals among the 40 bins or the bin that achieves the highest correlation with gene expression level. The signals of individual HM have a bad prediction accuracy, most of which are less than 0.5. However, when using all the 15 chromatin features, including histone marks, DNase I binding signals and CpG content, the prediction accuracy can be very high which is around 0.83.

1.3.2 Cross-validation

Cross-validation is also called rotation estimation. It is used in the problem whose goal is prediction and helps to evaluate predictive ability. In such problems, there is always a known dataset as the training set to estimate the unknown parameters and build a prediction model. To get the prediction accuracy, we use another dataset, testing set, which haven't been used in training process to evaluate the prediction ability. Predictors in testing set are inputs and get a set of predicted values. Comparing these values with observed values can get the prediction accuracy. Sometimes we don't have enough observations and every observation is significant to build the model. Cross-validation can help us in this situation and makes the maximum utilization of the dataset (Wang et al. 2010). In 4-fold cross-validation method, dataset is participated into 4 parts (Figure 2). Each time we select one part as testing set and the other three parts as training set. When all observations have been one time in the testing set, we can get predicted values against all observed values. The prediction accuracy is calculated from the overall dataset. The fold can be changed according to the specific dataset and situation.

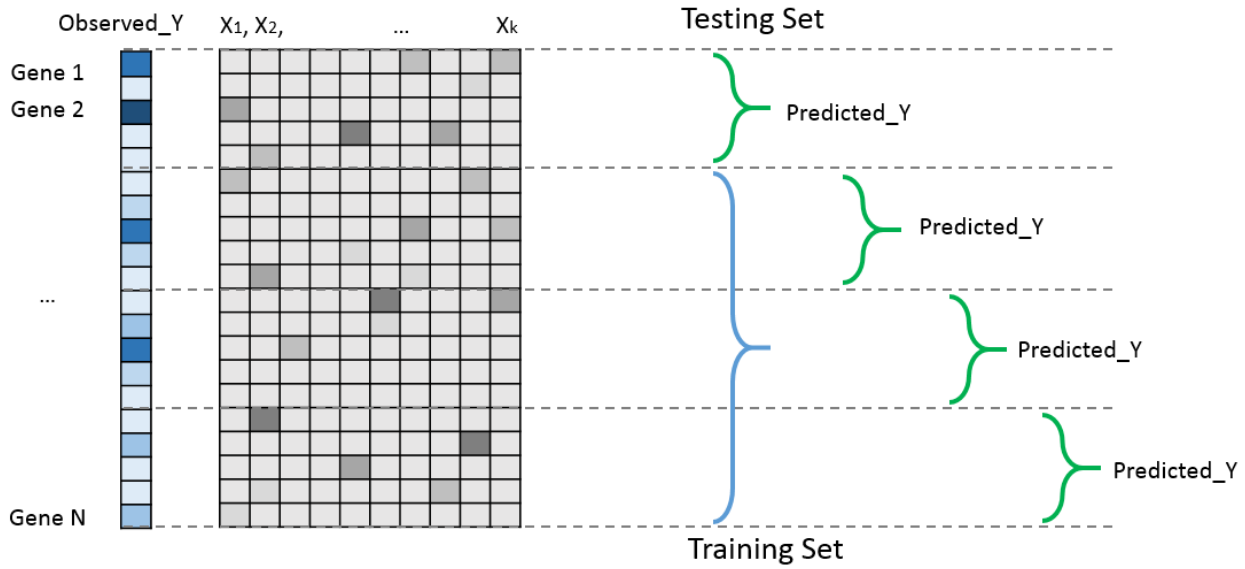

Figure 2. Illustration of cross-validation technique. Dataset is participated into 4 parts before fitting the model. Taking one part as testing set and using the other three parts to train the model. When the model is built, we use predictors from testing set as input and get a set of predicted values. Then we use another part as testing set and repeat this procedure. We can get predicted values against all observed values. All observations are one time in the testing set and three times in the training set.

1.3.3 Classification

In addition to linear regression model, classification model is also used to study the relationship between chromatin features and gene expression levels. Genes are not always active and have a positive expression level. Sometimes they are repressed and we cannot detect any RNA signals transcribed from these genes. But linear regression model can hardly classify which genes are active and which are not. This problem can be solved by classification model. Dong et al. used Random Forest method to distinguish genes which are "on" and which are "off" and improve the prediction accuracy to 0.95 in K562 cell line (Dong et al. 2012). However, this is not true for all cell lines at all conditions. Cheng et al. didn't see any improvements from classification model

when studying relationship between transcription factor binding and gene expression (Cheng et al. 2012).

1.4 Three-dimension genome structure

Human genome is organized into 22 paired chromosomes and two X chromosomes in female or one X one Y chromosome in male. The total length of human genome is over 3 billion base pairs and compressed in the nucleus with diameter around 7 μm. How the linear chromatin is packed into high density chromosome has been widely studied (Risca and Greenleaf 2015). The most recognized theory believes that chromatin is compacted in a hierarchical manner according to different structural length scales (Figure 3). Transcription regulation is related to genome structure since TFs must bind to DNA and start to function. Genome structure decides which DNA elements are exposed to TFs and which are not. Long-range interactions also play a role in transcription regulation: enhancers are always far from promoters linearly, but long-range interactions can bring them very close to promoter and influence the transcription (Shlyueva et al. 2014).

1.4.1 Spatial hierarchy of genomic compaction at three length scales

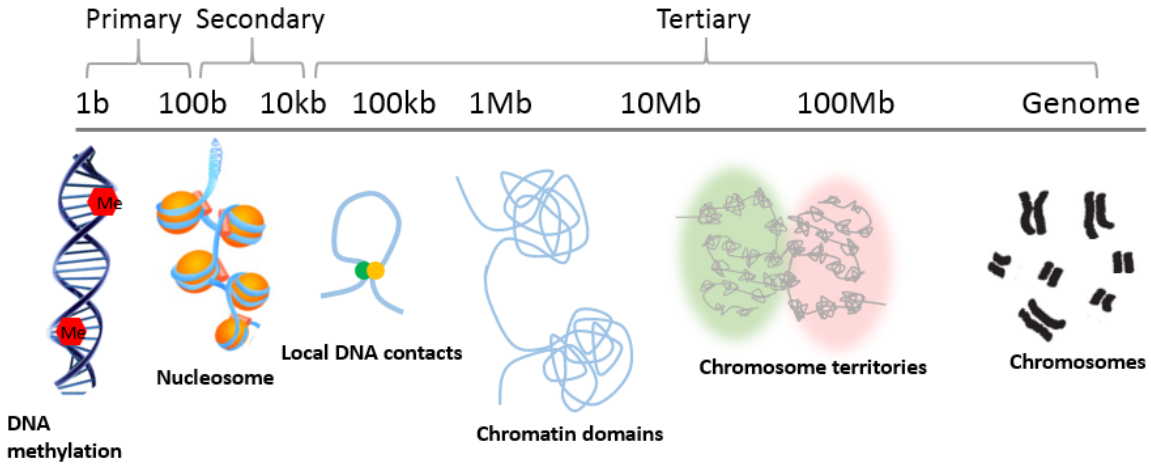

Figure 3. Overview of chromatin structure at different length scales. The primary structure is the linear chromatin structure, which includes DNA sequence, DNA-binding proteins and nucleosomes. The secondary structure is DNA chain wrapping around histones constitutes nucleosome and the nucleosome-nucleosome contacts. The tertiary structure includes local DNA contacts, like promoter-enhancer interaction, and megabase-scale structure: chromatin domain and chromosome territories.

1.4.1.1 Primary and secondary structure

The primary structure is at linear chromatin stage, it contains DNA sequence, DNA-binding proteins and nucleosomes. DNA methylation and transcription factor binding happens in the primary structure and plays a significant role in transcription regulation. These signals can be modelled by linear regression and are highly predictive of gene expression levels.

Secondary structure contains the local nucleosome-nucleosome interactions in which histone modifications can regulate the accessibility of DNA to DNA-binding proteins. Although we don't have any sequence-based method to probe the nucleosome-nucleosome interactions, we can get the histone modification signals from many techniques, such as ChIP-Seq. These histone

modifications signals can also be considered as input for linear regression model and have high prediction accuracy.

1.4.1.2 Tertiary structure

The tertiary structure is the 3D structure of genome. It includes local DNA-contacts, which happen in hundreds of kilobases scale and chromatin domains at megabase scale. The loop structure of local DNA contacts plays a central role in the communication between regulatory elements and remote targeted promoters (Tolhuis et al. 2002). Fullwood et al. shows that multiple DNA-binding sites may function as anchors to form multiple loop structure (Fullwood et al. 2009) (figure 4). This loop structure forms a transcription hub in which genes are all active and shows concentrated active marks, like transcription factors and positive histone modifications. In Section 1.3.1, we talked about the linear regression model to predict gene expression level from transcription factor binding and histone modification signals. In linear regression model, we assume individuals are independent and don't have influence on others which means genes should be independent, the protein binding or histone modification signals of one gene shouldn't have influence on other genes. However, in the loop structure, the transcription factors and histone marks in the transcription hub not only function on one specific gene, but also work cooperatively to genes around the anchor region. In this situation, linear regression model cannot capture all the information and we will introduce a new model, linear mixed-effects model, to solve this problem.

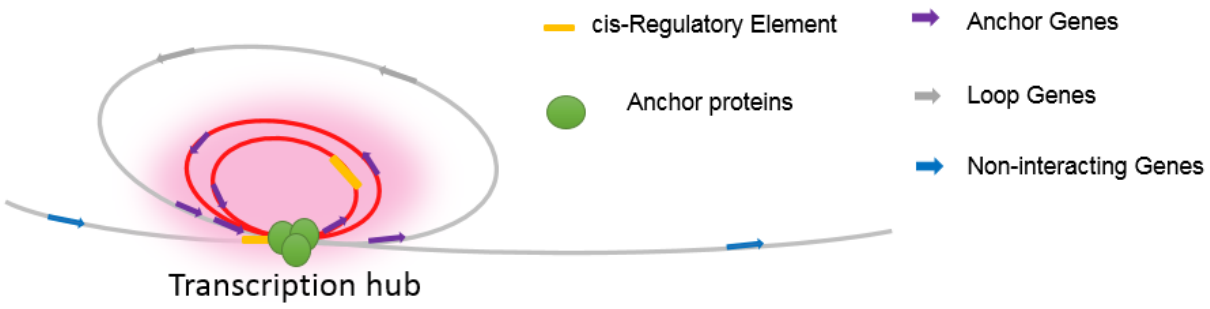

Figure 4. Protein-mediated chromatin loops. Multiple DNA-binding sites are tethered together by anchor proteins forming multiple loops. The genes (purple) around the anchor region interacted with each other. *Cis*-regulatory elements (orange) are also involved. The loop structure may form a transcription hub with intensive transcription factors and positive histone marks. Loop genes and non-interacting genes (gray and blue) are not considered.

A chromatin domain is the region in which local DNA contacts most often happens (Nora et al. 2013). It makes sure that *cis*-regulatory elements can only function to promoters in the same domain and provides the physical isolation of sets of *cis*-regulatory elements and genes from their neighbors. Chromatin domain can also be folded and constitute chromosome territories. Active domains may form an active territory and repressed domains may form a repressed territory. Similar as loop structure, genes in the same domain are dependent to each other and this information should be involved in statistical model for predictions.

1.4.2 Methods to assay chromatin structure

There are a lot of methods to study chromatin structure. At the primary and secondary stage, ChIP-Seq is widely used to get the transcription factor binding signals and histone modification signals. At the tertiary stage, Hi-C and ChIA-PET are mostly used to obtain the interactions among genome.

1.4.2.1 Hi-C experiment

Hi-C is an extension of 3C and it can identify genome wide long-range interactions. In Hi-C experiments (Figure 5), DNA elements are cross-linked with formaldehyde; Long DNA sequence is digested with restriction enzyme which leaves a 5' overhang to the DNA elements; The 5' overhang is filled and a biotin marker is added. The new blunt-ends will be ligated under specific conditions in which the ligation between crosslink DNA elements will mostly happen. The DNA elements with biotin markers are purified by the streptavidin beads and the purified DNA elements constitute a Hi-C library that will be sequenced. After massively parallel sequencing, there will be a catalog of interaction events among the whole genome (Lieberman-aiden et al. 2009; Dekker et al. 2013).

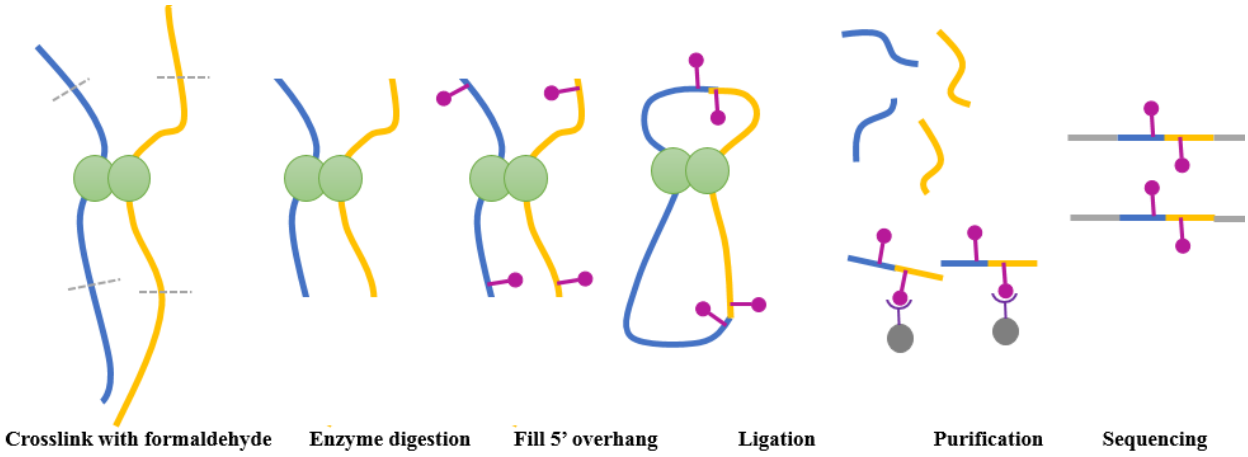

Figure 5. Architecture of Hi-C experiment. The first step is crosslinking with formaldehyde in which interacted elements are bound together. The restriction sites are represented by the gray dashed line. After enzyme digestion, there is a 5' overhang to DNA elements. A biotin marker is added when filing the 5' overhang. The blunt ends are ligated before sonication. DNA elements with biotin marker are purified by the streptavidin beads and constitute a Hi-C library. The Hi-C library will finally be sequenced using paired-ends.

1.4.2.2 ChIA-PET experiment

In ChIA-PET experiment, distal DNA elements are brought together by anchor proteins, and they will be tightly tethered with formaldehyde cross linking; Chromatin is fractured by sonication and enriched by chromatin precipitation; Remote DNA fragments in the same chromatin complex will be linked through proximity ligation with oligonucleotide DNA linkers; After restriction digestion, paired-end tags are extracted for sequencing and then mapped to reference genome (Li et al. 2014, 2010).

After mapping to reference genome, we can get interactions with very different spans. They can be classified into 3 categories (Figure 6). The first category is intra-chromosome intra-ligation pets, which are from the ligation of the same DNA fragments. They are mapped to a very short distance

within each other and mostly used to identify the binding sites. The second category is intra-chromosome inter-ligation, which is derived from the ligation of different DNA fragments. They can be used to detect long-range interactions. The last category is inter-chromosome inter-ligation which are very similar to category II but from different chromosomes.

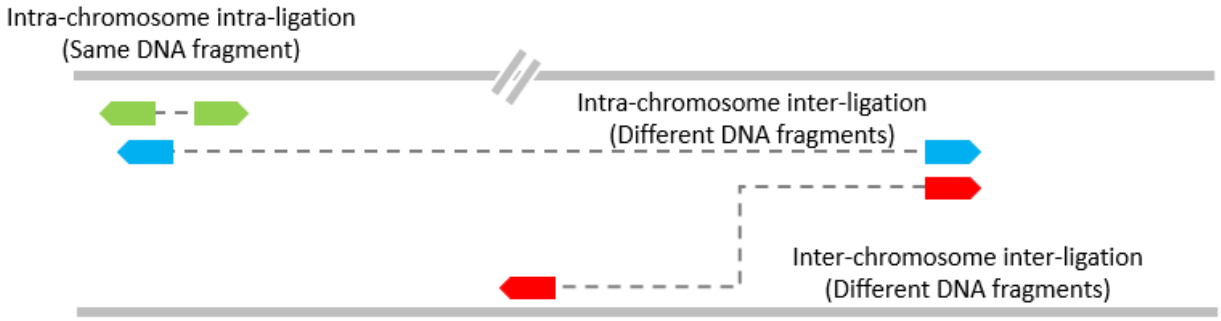

Figure 6. Three categories of ChIA-PET interactions. Intra-chromosome intra-ligation is also known as self-ligation and the two pets come from the same DNA fragment (green). Inter-ligations are the ligation for pets in different DNA fragments; they can be in the same chromosome (blue) or different chromosomes (red). The two arrows on the ends of dashed line represent the two anchors from one interaction.

1.5 Linear mixed-effects model

Linear mixed-effects (LME) models are mostly used to study the relationship between the response variables and the predictors within which a correlation structure exists. An LME model contains both fixed effect terms and random effect terms. As shown by the following formula, LME model has a random effect term B_g added into the linear regression to model the correlation structure among the observations. In particular, B_g can be used to model the clustering effect for genes in the same groups.

$$Yi = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} \ldots \beta_j X_{ij} + B_g + \varepsilon; \qquad B_g \sim N(0, \tau^2)$$

Comes to our problem, genes are not always independent with each other, we can use linear mixed effect model to capture this information. Genes in the same transcription hub or the same chromatin domain could be classified in the same group. Genes from different domains could be classified into different groups. Genes in the same group may have the same expression pattern, active or repressive. In LME model, genes classified into the same group will have the same random effect B_g. This random effect follows a Gaussian distribution with an unknown variance to be estimated when fitting the model by data.

We used the package "nlme" in R programming language to apply this model to our datasets (Bates 2005).

2 Data

2.1 Transcription factor binding and histone modification signals

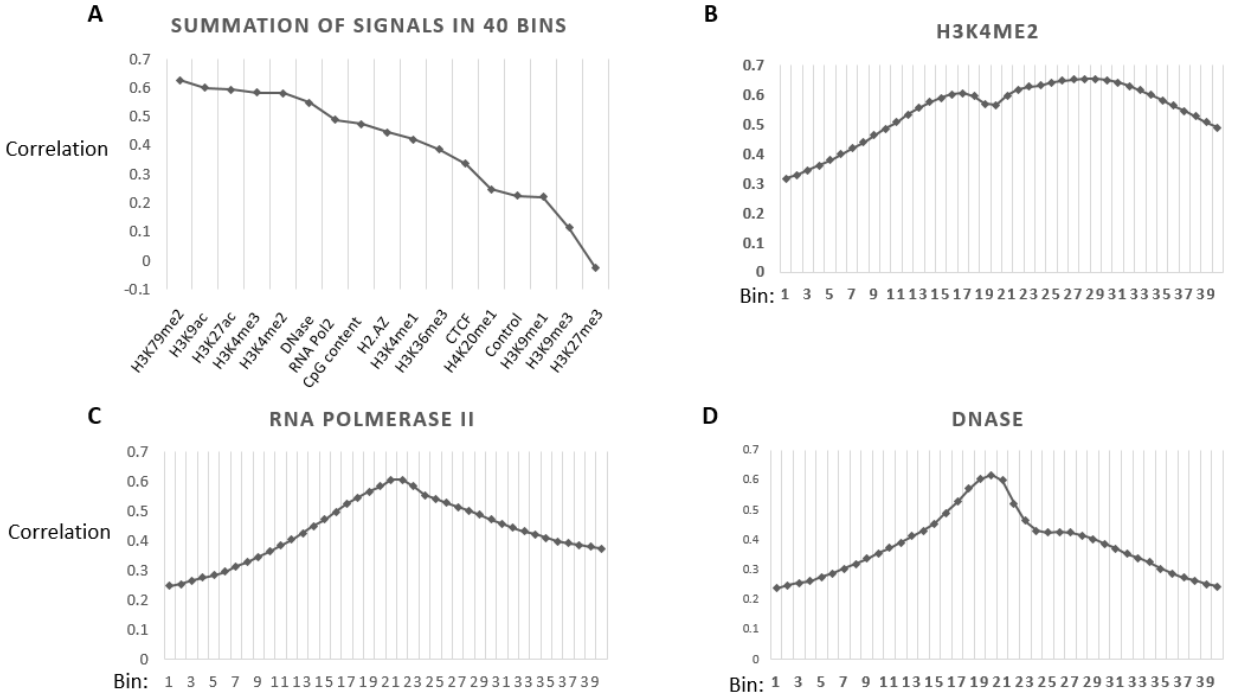

Figure 7. Correlation between histone modification signals and gene expression levels. (a) Simply taking the summation for signals from 40 bins as predictors, Y-axis is the correlation between the additive signals from 40 bins and gene expression levels in K562 cell line. (b,c,d) Considering signals in 40 bins individually, the correlation between specific chromatin feature signals and gene expression levels in 40 bins.

The same as what they did in linear regression model (Dong et al. 2012), predictors are transcription factor binding and histone modification signals around the transcription start sites. The raw binding signals are from ENCODE database and they are all in "bigwig" format. This format enables us to get the average signals in a certain bin. TSS region [-2k, 2k] are participated into 40 bins with 100bp in each bin. The average signal for each bin can be obtained from the "bigwig" dataset. Here, I simply take the summation of signals from 40 bins as the chromatin feature signals at gene *i*. There are 17 predictors and the correlation between their signals and gene expression levels are different. H3K79me2 have a highest correlation with gene expression level in K562 cell line. H3K9me1, H3K9me3 and H3K27me3 show no correlation with gene expression levels (Figure 7A). For H3K4me2, it shows a higher correlation in the nearby region of TSS than correlation at TSS. For RNA polymerase II and DNase, they both show the highest correlation at TSS.

2.2 Gene expression levels

In linear mixed effect model, the response variable is gene expression level. In our study, gene expression level is measured by RNA-Seq technique. The original datasets are from ENCODE

database. But our dataset is the observed expression level from Weng's group (Dong et al. 2012). They mapped the raw reads to reference genome "GENCODE v7" and obtained the expression level for 161,375 transcripts. The dataset contains expression levels from 3 different technologies: CAGE, RNA-PET and RNA-Seq. We only used the expression level measured by RNA-Seq technique in K562 and MCF7 cell line. For K562 cell line, the expression level from different cell compartments is available, but for MCF7 cell line, only the expression level from whole cell is available. Therefore, we have four different datasets, one is from MCF7 cell line and the other three are from different cell compartments in K562 cell line.

MCF-7 is a breast cancer cell line isolated in 1970 from a 69-year-old Caucasian woman (Levenson and Jordan 1997). K562 cell line is the first human immortalized myelogenous leukemia cell line and derived from a 53 year old female chronic myelogenous leukemia patient in blast crisis (Carmen and Lozzio 1975).

The RNA-Seq dataset contains 51,082 genes and some of them have multiple transcripts (Table 1). Most of genes have only one transcript but some genes may have more than 30 transcripts. 38% genes have more than one transcript, which are from alternative splicing. Transcripts for the same gene have different expression levels but share the same transcription factor binding and histone modification signals. When applying linear mixed effect model to the dataset, we need one transcript to represent the corresponding gene. Here, we simply select the transcript with highest expression level like the previous study (Dong et al. 2012).

Table 1 Summarization of number of transcripts for each gene.

Number of transcripts for one gene	1	2-5	6-10	11-30	>30
Number of genes	31484	11203	4759	3513	123

Among these 51,082 genes, 62% or 31,484 genes have only one transcript. 22% or 11,203 transcripts have 2 to 5 transcripts. 0.2% or 123 genes have more than 30 transcripts.

We have 161,375 transcripts in total, some of them are protein-coding genes having length from hundreds of base-pairs to kilo base-pairs. Some of them are small RNAs with length less than 100bp. There are also many processed transcripts and pseudo genes and the length of them are the same as protein coding genes. The median length of the 161,375 transcripts is 11,159bp, half of those transcripts have length around 15,000bp. The shortest length is only 7bp from a miRNA (Figure 8A).

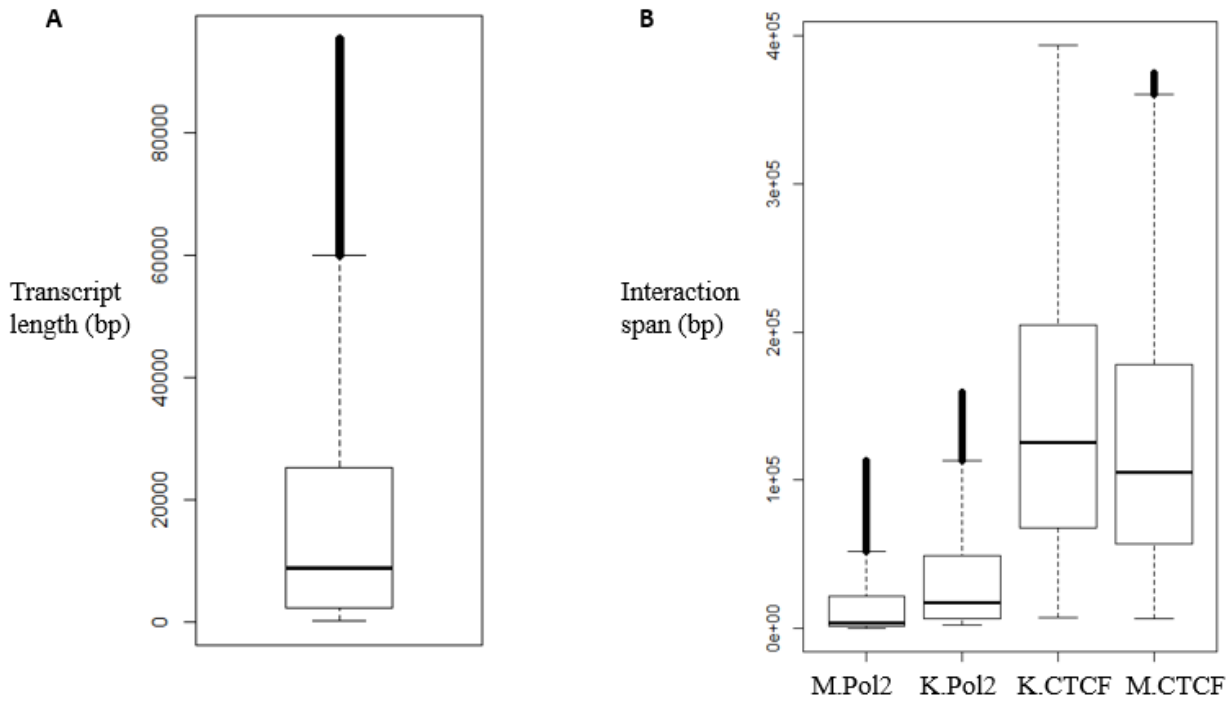

Figure 8. Boxplots for transcript length from RNA-Seq dataset and interaction span from ChIA-PET dataset. (a) Boxplot for transcript length from reference genome. 10% transcripts with super long length are removed to make the figure more descriptive. (b) Boxplot for interaction spans from 4 datasets, 10% interactions with super long length and interactions from different chromosomes have been removed.

2.3 Interactions among genome

The interactions among genome are detected by ChIA-PET experiment. The dataset can be downloaded from ENCODE database (Dunham et al. 2012). In ChIA-PET experiments, the antibodies to purify protein complex are different, the most widely used antibodies are CTCF antibody and RNA polymerase II antibody. Here, we use four ChIA-PET datasets with CTCF and RNA polymerase II antibody in K562 and MCF7 cell line (Table 2). The number of interactions detected with RNA polymerase II antibody are more than 10 times to interactions detected with CTCF antibody. This is because there are more RNA polymerase II binding sites among genome. It binds to the promoter region and initiate the transcription.

Based on the experiment using RNA polymerase II antibody, the distance between interactive loci is around thousands of base pairs. While based on the experiment using CTCF antibody, the distance is hundreds of thousand base pairs. The difference is due to the different functions of the two proteins. RNA polymerase II usually binds to promoter region and mediate local DNA contact like enhancer-promoter, promoter-promoter interactions (Li et al. 2012). CTCF usually binds to insulator and demarcating the boundaries between active domain and repressive domain and this certainly need a long span (Handoko et al. 2011) (Figure 8B).

Table 2. Summarization of interactions in 4 datasets.

	K562. RNA Pol2	MCF7.RNA Pol2	K562.CTCF	MCF7.CTCF
# of interaction	449,941	425,284	75,861	25,721
# of interactions with count > 3	66,220	67,022	20,015	6,812
# of intra-chromosome interactions (count > 3)	64,814	64,874	19,365	6,726
Median span of interactions (bp)	5,011	21,556	140,751	117,515

The first row is the total number of interactions detected by ChIA-PET. The second row is the number of interactions with counts larger than 3. Most of interactions only have count at 2, but we only use the interactions that have more than 3 count for further analysis. The median span of interactions is different in different dataset. Experiment with RNA polymerase II antibody detects interactions at a shorter span than those with CTCF antibody.

2.4 Chromatin domains

The chromatin domains are the area within which the local DNA contacts mostly happen. Rao et al. used Hi-C technique to study the chromatin domains and get the contact matrix among the whole chromosome at different resolutions (Rao et al. 2014). What we use here is the contact matrix at 5kb resolution and 25kb resolution, which can be downloaded from Gene Expression Omnibus (GEO http://www.ncbi.nlm.nih.gov/geo/).

3 Method

3.1 Data normalization

For both gene expression levels and histone modification signals, we use the logarithm transformation to normalize them. The observed datasets contain the values with a wide range and many outliers exist. Normalization can make the data more likely obey normal distribution and improve the prediction ability of the model. The normalized signals for histone modification k at gene i is $\log_2(X_{ik}+a_k)$. "a_k" is added to avoid $\log_2(0)$ and it can be a small value. The normalized expression level for gene i is $\log_2(Y_i+a)$.

3.2 Methods to classify genes

3.2.1 Sliding windows

In this method, chromatin is simply participated into equal-length domains. The genes located in the same domain will be in the same group (Figure 9). Here, we assume genes that are very close to each other linearly may have the same expression pattern. The length for each domain is not fixed and we will try different length to make a better grouping.

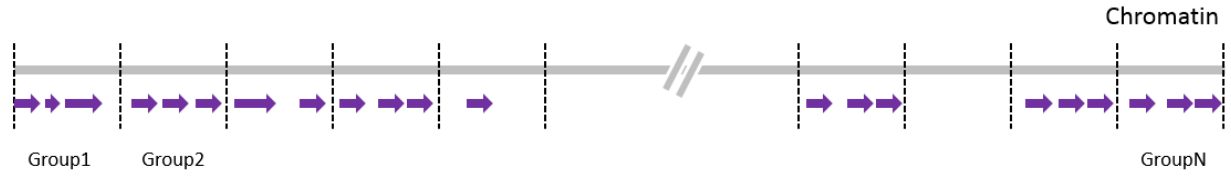

Figure 9. Illustration of using sliding-window method to classify genes. In this method, chromosomes are participated into equal-length domains. The genes located in the same domain are in the same group. The black dashed line is the cutting point of each domain.

3.2.2 Chromatin loops

In Section 1.4.1.2, we discussed chromatin loops and chromatin domains. From ChIA-PET datasets, we can get the interactions between two distal sites, and these interactions may come from the chromatin loop structure. In this situation, we classify genes around the same anchor region into the same group (Figure 10). Loop genes and non-interacting genes are not considered when doing classification. Since not all genes are in groups, we focus on the subset of genes that can be grouped.

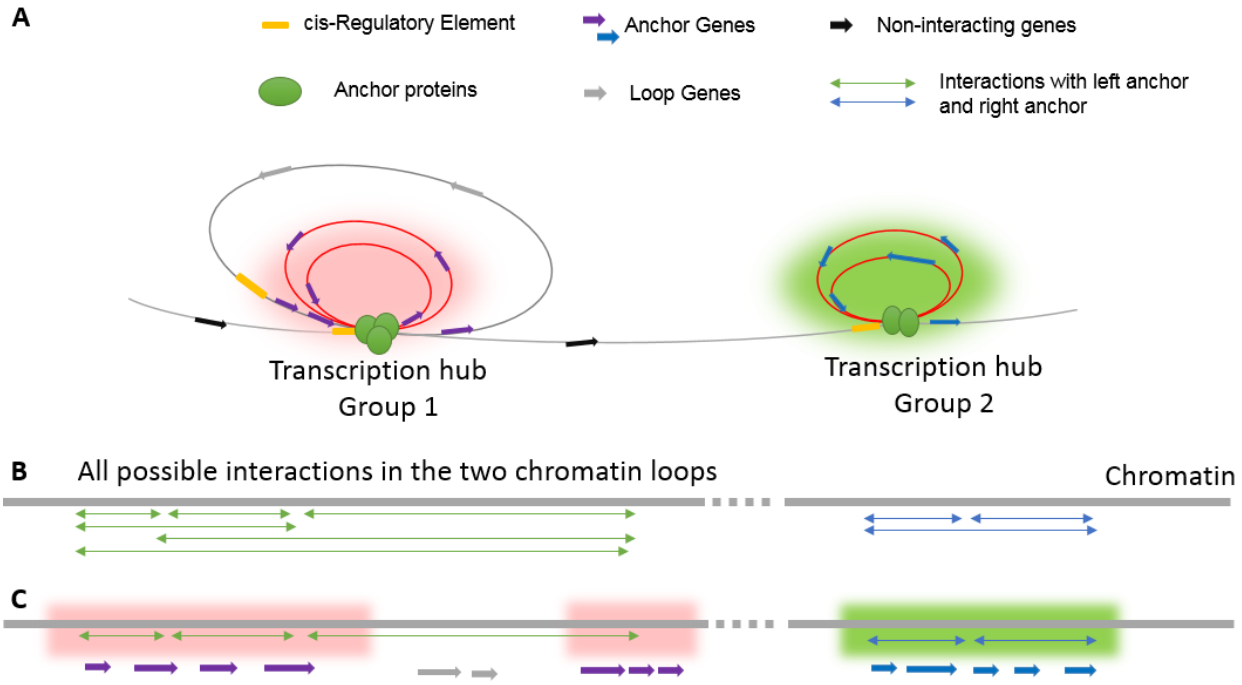

Figure 10. Illustration of using chromatin loops to classify genes for linear mixed effect model. (a) Chromatin loops are anchored by the anchor proteins (green ball). The pink shade and green shade are two transcription hubs. Genes in the same transcription hub should be classified into one group. (b) When applying ChIA-PET experiments, inter-ligation interactions are detected in the loop structure (green and blue lines with two arrows on each side), the arrows represent the two anchors from one interaction. (c) Genes around the same anchor region are classified into the same group (pink or green shade).

3.2.3 CTCF-anchored domains

CTCF is the most widely studied insulator-binding proteins and is known to organize the chromatin structure by demarcating the boundaries between chromatin domains (Bell et al. 1999; Kim et al. 2007; Vietri Rudan et al. 2015). In this method, we classify genes in the same chromatin domain, which is insulated by CTCF anchor as the same group (Figure 11). CTCF proteins often binds to cohesin and function as a whole. They always bind to insulator and isolate regulatory elements and genes in the same domain from their neighbor domains.

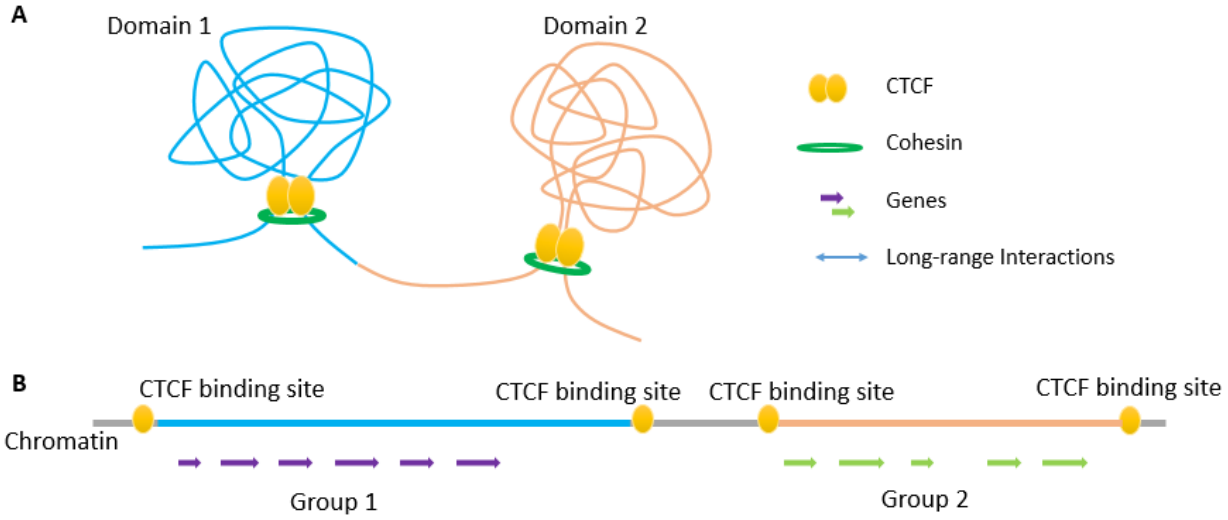

Figure 11. CTCF-anchored chromatin domains. (a) CTCF-anchored chromatin domain structure. CTCF dimers are shown in yellow balls, cohesins are shown in green circle. CTCF and cohesin always bind together and function as a whole. Here, CTCF demarcates the boundaries for two domains. Each domain can be considered as one big loop. The genes located in the same domain will be in the same group. (b) Method to classify genes according to the chromatin domain insulated by CTCF. The yellow ball shows CTCF binding sites on the chromatin. The purple arrows represent genes located in the first domain and green arrows represent genes in the second domain. The genes will be classified into two different groups.

3.2.4 Chromatin domains

To classify genes according to chromatin domain, we take advantage of the datasets described in 2.4. The dataset contains the contact between any two loci. At 5k resolution, each locus is a 5,000 base-pair DNA region in genome. Each cell in the matrix shows the contact frequency between locus in the row and locus in the column. Dark red color means a high contact frequency, light red color means a low contact frequency, while white color means there is no contract between these two loci. Genes are classified according to the domain. Genes in the same domain are classified into the same group. They may have the same expression pattern: active or repressive. There are 7 domains in the chromosome and genes are classified into 7 groups (Figure 12). In Rao's dataset, the contact matrix is more complex and contains more contacts and loci. When classifying genes according to domains, we should remove the contact with low frequency. The contact matrix will be much clear when omitting the low-frequency contacts.

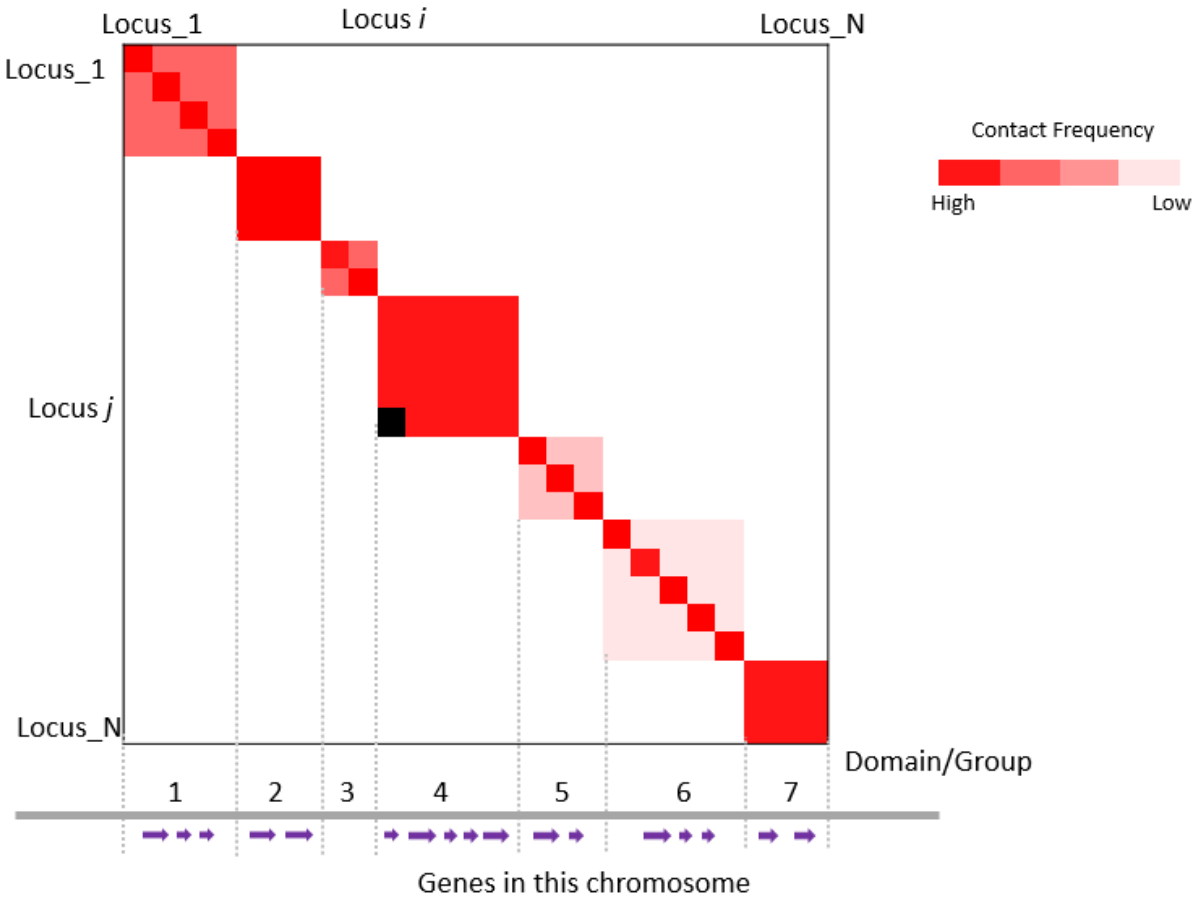

Figure 12. Illustration of using chromatin domains to classify genes for linear mixed effect model. The X and Y axis are loci from 1 to N in one chromosome. Each cell in this matrix represents the contact frequency between locus in the row and locus in the column. The cell in black shows the contact frequency between locus *i* and locus *j*. White color means no contact and as the color going darker, the contact frequency is higher. The contact frequencies are always very high between the neighboring loci and form a domain with a high contact frequency in it. In this matrix, chromosome can be classified into 7 domains. Genes in the same domain will be in the same group.

3.3 Modeling

Linear mix effect model is very similar to linear regression. A predictor *Xi* is the *i*th chromatin feature signal around TSS region described in 2.1. Observed values are the gene expression levels from RNA-Seq technique, and predicted values are obtained by the model that is trained from training set.

The difference between the two models is how to select training set. Linear regression model selects training set randomly. However, there is a group label added to linear mixed effect model and we should take them into account when training model. From linear mixed effect model, we want to estimate the random effect and the correlation structure inside the group and use this information to predict new genes in this group. Therefore, we need one part of genes in this group as training set to estimate the structure inside the group and another part of genes as testing set to test the structures obtained from training (Figure 13).

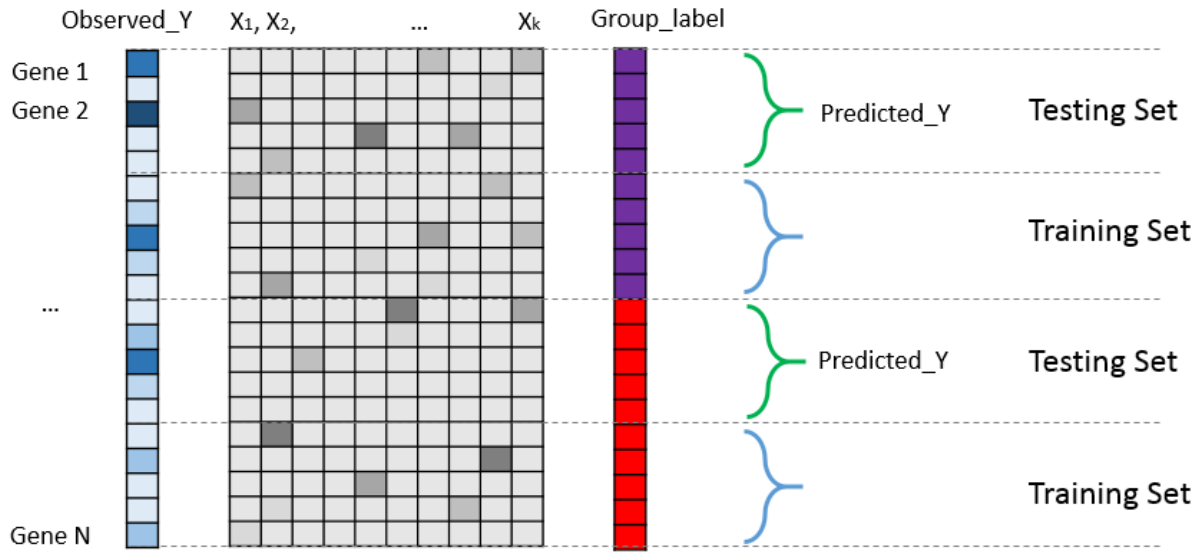

Figure 13. Illustration of using linear mixed effect to model gene expression regulation. The method here is very similar to linear regression model, but group information is added. We still use cross-validation method to mostly exploit the dataset. Assume we have two groups here and each group contains 10 genes. The training sets are selected according to the group labels. The genes in the same group will be participated into two parts, one part in training set and one part in testing set.

4 Results

4.1 Simulation

To prove the idea that linear mixed effect model can improve the prediction accuracy when individuals are not independent, we simulated a dataset in which individuals are classified into 20 groups. The random term B_g is the same for the individuals in the same group. We assume the predictors X_i follows the normal distribution with mean equals to 1 and standard deviation equals to 1: $X_i \sim N(1,1)$ and error term $\varepsilon \sim N(0,1)$. Whether individuals are independent or not depends on the random effect B_g, which also follows the normal distribution. If the standard deviation τ is very small, then B_g is very close to 0 and individuals are considered to be independent. But when τ is big, B_g is very different for different groups and it makes the linear regression model less efficient.

Prediction accuracy is the correlation between predicted Y_i and observed Y_i. When τ is small, there is no significant difference between linear regression model and linear mixed effect model. As τ grows, the differential prediction accuracy also grows. This proves that when individuals are not independent, like there is a structure among individuals, linear mixed effect model can overcome these random effects and achieve a better prediction (Figure 14).

$$Yi = \beta_0 + \beta_1 X_i + \boldsymbol{B_g} + \varepsilon;$$

$$\beta = 1 \quad Xi \sim N(1,1) \quad \varepsilon \sim N(0,1^2) \quad Bg \sim N(0, \tau^2)$$

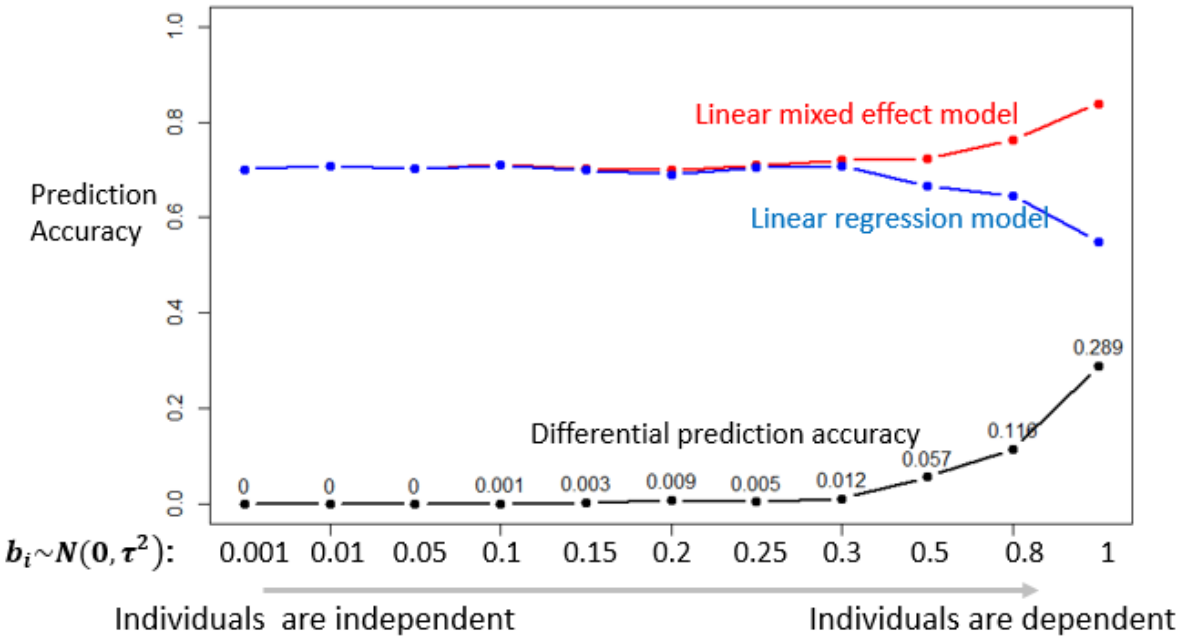

Figure 14. Prediction accuracy for simulated datasets with different standard deviation τ. Y-axis shows the prediction accuracy. X-axis shows different standard deviation τ. Red line represents the prediction accuracy from linear mixed effect model, blue line represents the prediction accuracy from linear regression model. The differential prediction accuracy is on the bottom, which is shown in black.

4.2 Real data

According to the model in Section 3.3, we used the 4 datasets from two cell lines to predict gene expression levels. The group information is different from different methods.

4.2.1 Method according to sliding windows

Adding the group information obtained by sliding-window method in Section 3.2.1. Chromatin is participated with different domain length. When using domain length equal to 200kb, the accuracy from linear mixed effect model is 0.4% higher than the accuracy from linear regression method (Figure 15A). There is no big difference between the grouping at 200kb and 900kb (Figure 15B).

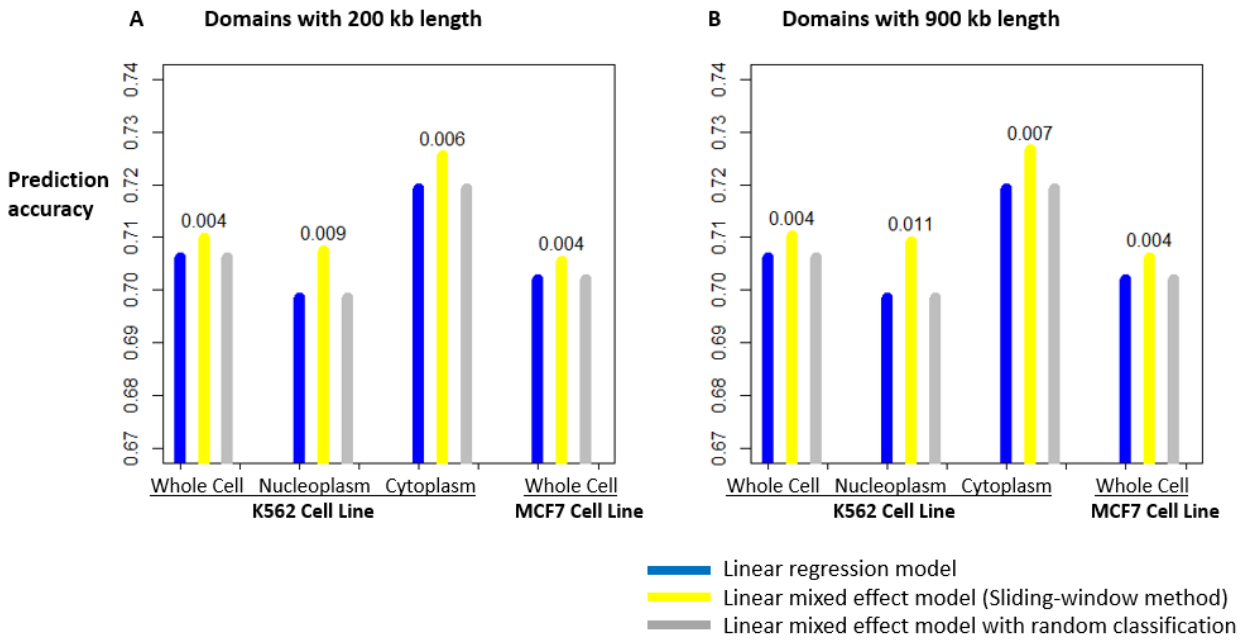

Figure 15. Prediction accuracy from 4 datasets with group information obtained by sliding-window. (a) Blue bar show the prediction accuracy from linear regression model and yellow bar shows the prediction accuracy from linear mixed effect model with sliding-window method. The numbers on the top of bars are the differential prediction accuracy for each cell line. (b) The group information here is obtained from domain length at 900 kb.

4.2.2 Method according to chromatin loops

Adding the group information obtained by the method in Section 3.2.2 to linear mixed effect model and we got a higher prediction accuracy than linear regression model. The improvement is around 0.3% for the whole dataset with 51,082 genes. It is small but constant for the four datasets. To prove that the improvement is not generated from the model itself, but from the additional group information, we randomly classified genes into groups as the control and the prediction accuracy of it is the same as linear regression model (Figure 16A). Since linear mixed effect model need we have enough observations in the same group, we selecting a subset with genes in the group with bigger group size. The differential prediction accuracy is much higher than using the whole dataset, is around 2% (Figure 16B).

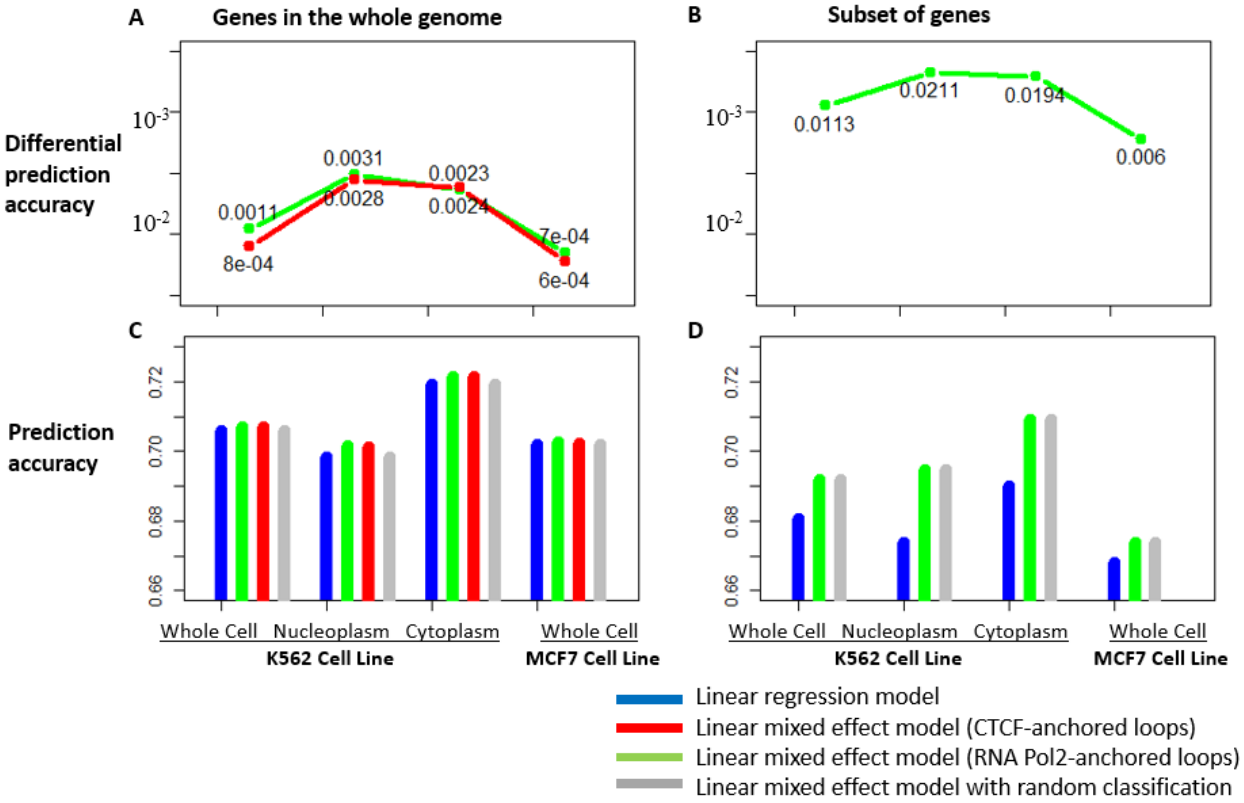

Figure 16. Prediction accuracy from 4 datasets with group information obtained by chromatin loops. (a) The differential prediction accuracy between linear regression model and linear mixed effect model. Green line is for group information obtained from RNA Polymerase II-anchored chromatin loop. Red line if for group information obtained from CTCF-anchored loops. (b) The differential prediction accuracy between linear regression and linear mixed effect model with group information obtained by RNA polymerase II anchor (green). Subset genes are selected according to their group size. 4293 genes in big groups are selected. (c) Prediction accuracy for linear regression and linear mixed effect model. Gray bar is the control with randomly classified group label. (d) Prediction accuracy for the 4293 selected genes.

4.2.3 Method according to CTCF-anchored domains

Adding the group information obtained by the method in 3.2.2.1 to linear mixed effect model. The prediction accuracy from the whole dataset is very similar to the model in 4.2.1 and the improvement is around 0.3%. But the improvement 1% from subset genes is less than the model in Section 4.2.1, which is around 2% (Figure 17).

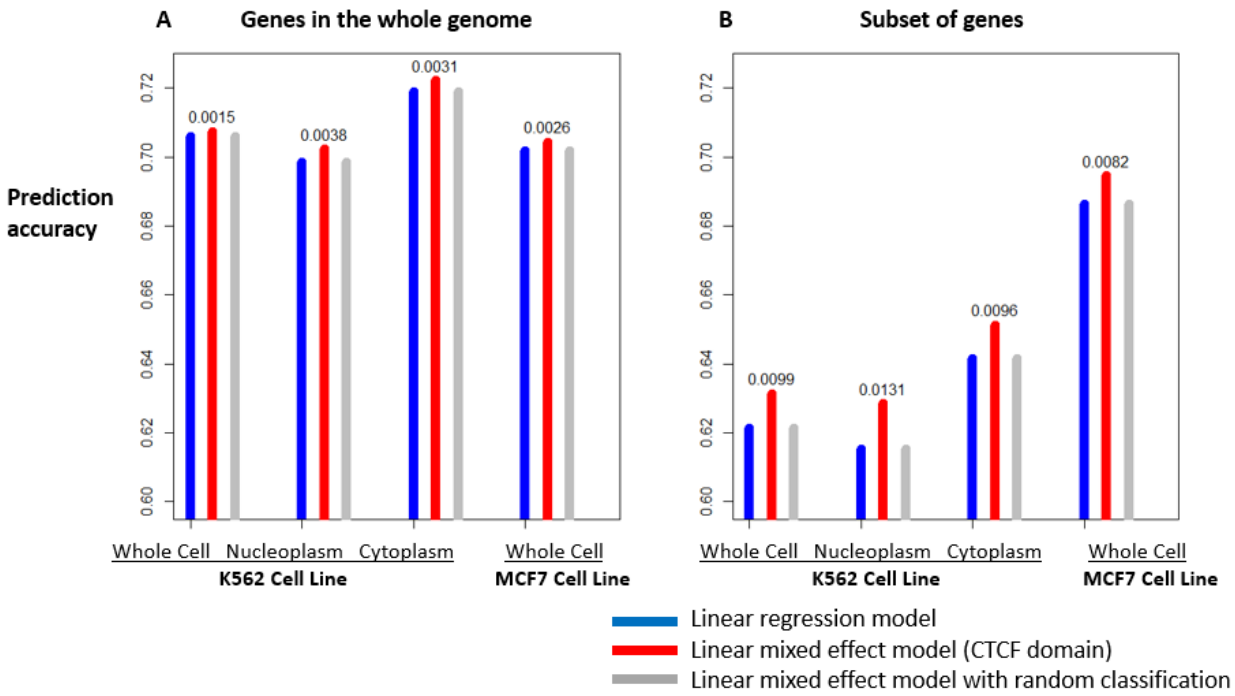

Figure 17. Prediction accuracy from 4 datasets with group information obtained by CTCF-anchored domain. (a) Prediction accuracy from the whole dataset including 51,082 genes. Blue bar shows the prediction accuracy from linear regression model and red bar shows the prediction accuracy from linear mixed effect model. The numbers above the red bars are the differential prediction accuracy between the two models. Linear mixed model used the group information obtained from ChIA-PET dataset with CTCF anchor. The gray bar is the control in which genes are randomly classified. (b) Prediction accuracy from a subset of genes. This subset is selected by the group size. Only grouped genes within top 300 largeest groups were retained. 12,900 genes were selected in K562 cell line and 14,417 genes were selected in MCF7 cell line.

4.2.4 Method according to chromatin domains

Adding the group information obtained by the method in 3.2.2.2 to linear mixed effect model. There is a better improvement in with this method. The differential prediction accuracy from method with 5k resolution is around 0.5% and the differential prediction accuracy from method with 25k resolution is around 0.3%. From the three methods, the naïve chromatin domain method with 5k resolution have the highest improvements (Figure 18A). The result is worse the resolution is low (Figure 18B).

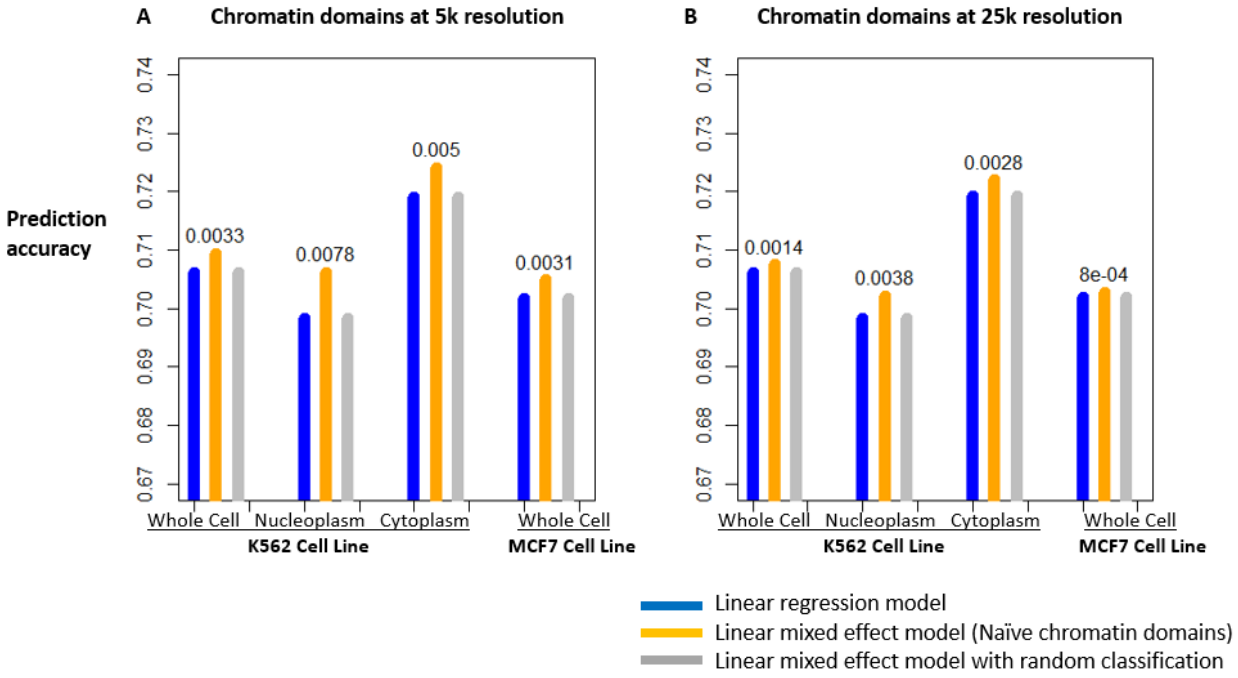

Figure 18. Prediction accuracy from 4 datasets with group information obtained by naïve chromatin domains. (a) Blue bar show the prediction accuracy from linear regression model and orange bar shows the prediction accuracy from linear mixed effect model with naïve chromatin domains method. The numbers on the top of bars are the differential prediction accuracy for each cell line. (b) The group information here are obtained from naïve chromatin domain method at a lower resolution: 25kb.

4.2.5 Predicting expression levels with individual chromatin feature

For each of the 16 chromatin features in K562 cell line, we want to know the individual predictive powers with different group information: one is obtained from method with chromatin loops and the other is from method with chromatin domains. The differential prediction accuracy is very similar form the two methods (Figure 19). Histone marks with low predictive power like H3K27me3 and H3K9me3 have a big improvement when group information is added. In the contrast, histone marks with high predictive power like H3K27ac and H3K4me2 don't have too much improvement when group information is added.

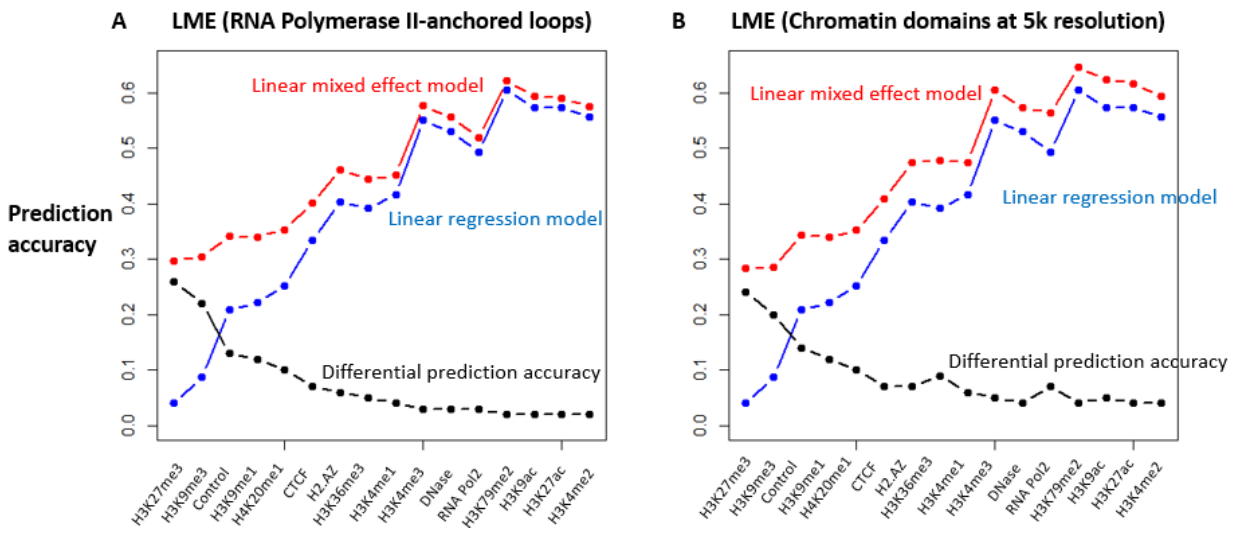

Figure 19. Prediction accuracy from individual chromatin feature. (a) Linear mixed effect model with group information obtained from chromatin loop method in K562 cell line. The red line is prediction accuracy from linear mixed effect model and blue line is prediction accuracy from linear regression model. Black line is the differential prediction accuracy. The X axis shows the histone marks and transcription factors. (b) Group information obtained from naïve chromatin domain method at 5k resolution.

5 Discussion

Our study shows the prediction results in linear mixed-effects model with group information from different methods. The method according to chromatin loops focuses on local contact and transcription hub. In this method, only a small part of genes can be grouped, most of them are loop genes or non-interacted genes that are outside groups. Linear mixed-effects model need enough observations in one group and for those genes outside groups, we used linear regression model instead. Therefore, the improvement is very small, around 0.3%, when we use the whole dataset as input. We removed the genes that are outside group or in groups with small size and used the subset as input. The differential prediction accuracy is about 2.1%, higher than the differential prediction accuracy from the whole dataset (Figure 16).

The differential prediction accuracy is different from different grouping method. Methods according to chromatin loop and CTCF domain both have small improvements (Table 3). The reason for method with chromatin loop is only a small part of genes are in groups and this is hard for linear mixed effect model to estimate the random effects and correlation structures within group. Methods according to CTCF domain and chromatin domains have the same idea but different results. The reason is because CTCF-anchored domain can only capture the contact information for a small part of genes. Not all domains are anchored by CTCF. Differently, the chromatin domain method involves all the genes in the genome and generates groups with big size. For the same chromatin domain method, grouping methods at different resolutions have different results in prediction accuracy. Chromatin domains with a high resolution often have better results than those with low resolution.

Table 3. Summary of the differential prediction accuracy (%) for all genes.

	K562. Whole	K562.N	K562.C	MCF7.Whole
Sliding window (200kb)	0.4	0.9	0.6	0.4
Chromatin loop (RNA Polymerase II)	0.11	0.31	0.23	0.07
CTCF-domain	0.15	0.38	0.3	0.26
Naïve chromatin domain (5k resolution)	0.33	0.78	0.5	0.31

All values in percentage format. Each column represents one cell line. K562.N means K562 cell line with RNA extracted from nucleoplasm. K562.C means K562 cell line with RNA extracted from cytoplasm. Each row represents a grouping method described in 3.2.

Therefore, the first important reason why the improvement is lower is because there are always a large part of genes cannot be classified according to the methods we used here and linear mixed effect model don't have enough input information. Comparing all the methods, we found that the result from method 1 is the best (Table 3). One reason is that all genes can be classified according to sliding windows method; another reason is that the local contact among the neighborhood genes is also important for grouping genes. For the next step, we can combine the local contact information and the long-range interaction information together and construct a correlation map among all genes. For example, for gene *i* and gene *j*, the correlation between them can be constructed by a weighted sum of the linear relation (1D) and spatial relation (3D). The linear relation is the linear distance between the two genes (possibly based on genetic distance), the spatial relation is whether they have long-range interactions or not. If they are very close to each other linearly, the linear relation will dominate their correlation. If they are far from each other linearly, the spatial relation will dominate their correlation. For any subset of data, we can use this method to calculate the correlation matrix. Linear mixed effect model can treat this subset of data as one group and using the correlation matrix to build the model. This track of work has started, and the results will be reported elsewhere.

In our study, we assume genes in the same group are positively correlated; they are either all activated or all repressed. However, genes that are very close to each other may be competitors. The expression of one gene may suppress the expression of genes nearby. How genes in the same group correlated with each other is also important information for gene expression prediction. As we talked above, we can add a within-group correlation to the model. Except grouping methods according to genome structure, we can also seek help from pathways.

The third reason is chromatin feature signals already existed accurately in the same domain. Which means the prediction accuracy from linear regression model is very high and it is hard to improve the accuracy to a much higher level. From the predicting model with individual chromatin features, we know that linear mixed effect model can help the predictors with bad predictive power to achieve a better prediction. But for those predictors that already have a high predictive power, linear-mixed effects model cannot achieve a better prediction (Figure 19).

6 Conclusion

In our study, we compare the results from linear regression and LME model. The improvement is consistent and stable. We classified genes according to four different methods. The methods according to sliding windows have a better improvement than the other three methods for the whole gene set. However, when we looked at the subset of genes, which can be grouped according to the four methods, the improvement is higher for the last three methods. For individual chromatin feature predictors, the LME models improve from 2% to 26 %, in which improvement is more significant for chromatin features that have low original predictive**.**

Reference

Barski A, Cuddapah S, Cui K, Roh T-Y, Schones DE, Wang Z, Wei G, Chepelev I, Zhao K. 2007. High-resolution profiling of histone methylations in the human genome. *Cell* **129**: 823–37. http://www.ncbi.nlm.nih.gov/pubmed/17512414 (Accessed July 9, 2014).

Bates D. 2005. Fitting linear mixed models in R. Using the lme4 package. *R News* **5**: 27–30. http://www.ncbi.nlm.nih.gov/pubmed/23677808.

Bell AC, West AG, Felsenfeld G. 1999. The protein CTCF is required for the enhancer blocking activity of vertebrate insulators. *Cell* **98**: 387–396.

Carmen B, Lozzio B. 1975. Human With Chronic Positive. *Cancer* **45**: 321–334.

Carpenter S, Ricci EP, Mercier BC, Moore MJ, Fitzgerald K a. 2014. Post-transcriptional regulation of gene expression in innate immunity. *Nat Rev Immunol* **14**: 361–76. http://www.ncbi.nlm.nih.gov/pubmed/24854588.

Cheng C, Alexander R, Min R, Leng J, Yip KY, Rozowsky J, Yan K, Dong X, Djebali S, Ruan Y, et al. 2012. Understanding transcriptional regulation by integrative analysis of transcription factor binding data. 1658–1667.

Cheng C, Gerstein M. 2012. Modeling the relative relationship of transcription factor binding and histone modifications to gene expression levels in mouse embryonic stem cells. *Nucleic Acids Res* **40**: 553–568.

Dekker J, Marti-Renom M a, Mirny L a. 2013. Exploring the three-dimensional organization of genomes: interpreting chromatin interaction data. *Nat Rev Genet* **14**: 390–403. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3874835&tool=pmcentrez&rendertype=abstract.

Dong X, Greven MC, Kundaje A, Djebali S, Brown JB, Cheng C, Gingeras TR, Gerstein M, Guigó R, Birney E, et al. 2012. Modeling gene expression using chromatin features in various cellular contexts. *Genome Biol* **13**: R53. http://genomebiology.com/2012/13/9/R53.

Dunham I, Kundaje A, Aldred SF, Collins PJ, Davis C a., Doyle F, Epstein CB, Frietze S, Harrow J, Kaul R, et al. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**: 57–74.

Fullwood MJ, Liu MH, Pan YF, Liu J, Xu H, Mohamed Y Bin, Orlov YL, Velkov S, Ho A, Mei PH, et al. 2009. An oestrogen-receptor-alpha-bound human chromatin interactome. *Nature* **462**: 58–64. http://dx.doi.org/10.1038/nature08497.

Handoko L, Xu H, Li G, Ngan CY, Chew E, Schnapp M, Lee CWH, Ye C, Ping JLH, Mulawadi F, et al. 2011. CTCF-mediated functional chromatin interactome in pluripotent cells. *Nat Genet* **43**: 630–638.

Hublitz P, Albert M, Peters a. HFM. 2009. Mechanisms of transcriptional repression by histone lysine methylation. *Int J Dev Biol* **53**: 335–354.

Hwang YC, Zheng Q, Gregory BD, Wang LS. 2013. High-throughput identification of long-range regulatory elements and their target promoters in the human genome. *Nucleic Acids Res* **41**: 4835–4846.

Ingolia NT. 2014. Ribosome profiling: new views of translation, from single codons to genome scale. *Nat Rev Genet* **15**: 205–13. http://www.ncbi.nlm.nih.gov/pubmed/24468696.

Izaurralde E. 2002. Nuclear export of messenger RNA. *Results Probl Cell Differ* **35**: 133–150.

Kim TH, Abdullaev ZK, Smith AD, Ching K a., Loukinov DI, Green RD, Zhang MQ, Lobanenkov V V., Ren B. 2007. Analysis of the Vertebrate Insulator Protein CTCF-Binding Sites in the Human Genome. *Cell* **128**: 1231–1245.

Koch CM, Andrews RM, Flicek P, Dillon SC, Clelland GK, Wilcox S, Beare DM, Fowler JC, Couttet P, James KD, et al. 2007. The landscape of histone modifications across 1 % of the human genome in five human cell lines. 691–707.

Levenson a S, Jordan VC. 1997. MCF-7: the first hormone-responsive breast cancer cell line. *Cancer Res* **57**: 3071–3078.

Li G, Cai L, Chang H, Hong P, Zhou Q, Kulakova E V, Kolchanov N a, Ruan Y. 2014. Chromatin Interaction Analysis with Paired-End Tag (ChIA-PET) sequencing technology and application. *BMC Genomics* **15**: S11. http://www.biomedcentral.com/1471-2164/15/S12/S11.

Li G, Fullwood MJ, Xu H, Mulawadi FH, Velkov S, Vega V, Ariyaratne PN, Mohamed Y Bin, Ooi H-S, Tennakoon C, et al. 2010. ChIA-PET tool for comprehensive chromatin interaction analysis with paired-end tag sequencing. *Genome Biol* **11**: R22. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2872882&tool=pmcentrez&rendertype=abstract.

Li G, Ruan X, Auerbach RK, Sandhu KS, Zheng M, Wang P, Poh HM, Goh Y, Lim J, Zhang J, et al. 2012. Extensive promoter-centered chromatin interactions provide a topological basis for transcription regulation. *Cell* **148**: 84–98. http://dx.doi.org/10.1016/j.cell.2011.12.014.

Lieberman-aiden E, Berkum NL Van, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, et al. 2009. of the Human Genome. **33292**: 289–293.

Nora EP, Dekker J, Heard E. 2013. Segmental folding of chromosomes: A basis for structural and regulatory chromosomal neighborhoods? *BioEssays* **35**: 818–828.

Rao SSP, Huntley MH, Durand NC, Stamenova EK. 2014. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell* **159**: 1665–1680. http://dx.doi.org/10.1016/j.cell.2014.11.021.

Risca VI, Greenleaf WJ. 2015. Unraveling the 3D genome: genomics tools for multiscale exploration. *Trends Genet* 1–16. http://linkinghub.elsevier.com/retrieve/pii/S0168952515000633.

Rodrigues R, Grosso AR, Moita L. 2013. Genome-Wide Analysis of Alternative Splicing during Dendritic Cell Response to a Bacterial Challenge. *PLoS One* **8**.

Saccani S, Trabucchi M. 2015. Regulation of stimulus-inducible gene expression in myeloid cells. *Semin Immunol* **27**: 33–43. http://linkinghub.elsevier.com/retrieve/pii/S104453231500007X.

Sainsbury S, Bernecky C, Cramer P. 2015. Structural basis of transcription initiation by RNA polymerase II. *Nat Rev Mol Cell Biol* **16**: 129–143. http://www.nature.com/doifinder/10.1038/nrm3952.

Saurabh S, Vidyarthi AS, Prasad D. 2014. RNA interference: concept to reality in crop improvement. *Planta* 1–22.

Shlyueva D, Stampfel G, Stark A. 2014. Transcriptional enhancers: from properties to genome-wide predictions. *Nat Rev Genet* **15**: 272–86. http://www.ncbi.nlm.nih.gov/pubmed/24614317.

Tolhuis B, Palstra RJ, Splinter E, Grosveld F, De Laat W. 2002. Looping and interaction between hypersensitive sites in the active ??-globin locus. *Mol Cell* **10**: 1453–1465.

Vietri Rudan M, Barrington C, Henderson S, Ernst C, Odom DT, Tanay A, Hadjur S. 2015. Comparative Hi-C Reveals that CTCF Underlies Evolution of Chromosomal Domain Architecture. *Cell Rep* **10**: 1297–1309. http://linkinghub.elsevier.com/retrieve/pii/S2211124715001126.

Wang Y, Fan Y, Bhatt P, Davatzikos C. 2010. High-dimensional pattern regression using machine learning: From medical images to continuous clinical variables. *Neuroimage* **50**: 1519–1535. http://dx.doi.org/10.1016/j.neuroimage.2009.12.092.

Appendix A

Absolute prediction accuracy for figure 15~19.

Table A1. Absolute prediction accuracy for the whole dataset with grouping method according to sliding-window (Figure 15).

	K562. Whole	K562. Nucleoplasm	K562. Cytoplasm	MCF7. Whole
Linear regression model	0.7061	0.6985	0.7192	0.7020
LME (domain length 200kb)	0.7099	0.7075	0.7255	0.7056
LME (domain length 900kb)	0.7104	0.7093	0.7266	0.7062

Table A2. Absolute prediction accuracy for the whole dataset with grouping method according to chromatin loops (Figure 16A).

	K562. Whole	K562. Nucleoplasm	K562. Cytoplasm	MCF7. Whole
Linear regression model	0.7060	0.6984	0.7192	0.7019
LME (RNA Pol2-anchored loop)	0.7071	0.7015	0.7215	0.7026
LME (CTCF-anchored loop)	0.7070	0.7013	0.7215	0.7024

Table A3. Absolute prediction accuracy for the subset genes with grouping method according to chromatin loops (Figure 16B).

	K562. Whole	K562. Nucleoplasm	K562. Cytoplasm	MCF7. Whole
Linear regression model	0.6807	0.6737	0.6899	0.6680
LME (RNA Pol2 anchor)	0.6920	0.6948	0.7093	0.6740

Table A4. Absolute prediction accuracy for the whole dataset with grouping method according to CTCF-anchored domains (Figure 17A).

	K562. Whole	K562. Nucleoplasm	K562. Cytoplasm	MCF7. Whole
Linear regression model	0.7060	0.6985	0.7191	0.7018
LME (CTCF anchored domain)	0.7075	0.7023	0.7222	0.7044

Table A5. Absolute prediction accuracy for the subset genes from grouping method according to CTCF-anchored domains (Figure 17B).

	K562.	K562.	K562.	MCF7.

	Whole	Nucleoplasm	Cytoplasm	Whole
Linear regression model	0.6215	0.6154	0.6417	0.6864
LME (CTCF anchored domain)	0.6314	0.6285	0.6513	0.6946

Table A6. Absolute prediction accuracy for the whole dataset from grouping method according to chromatin domains (Figure 18).

	K562. Whole	K562. Nucleoplasm	K562. Cytoplasm	MCF7. Whole
Linear regression model	0.7061	0.6984	0.7191	0.7019
LME (5k resolution)	0.7094	0.7062	0.7241	0.7049
LME (25k resolution)	0.7075	0.7022	0.7220	0.7028

Table A7. Absolute prediction accuracy for individual chromatin features (Figure 19).

	Linear regression model	LME (RNA Polymerase II-anchored loops)	LME (Naïve chromatin domain 5k resolution)
H3K27me3	0.040	0.297	0.284
H3K9me3	0.087	0.305	0.285
Control	0.208	0.341	0.344
H3K9me1	0.222	0.340	0.339
H4K20me1	0.251	0.353	0.353
CTCF	0.334	0.401	0.409
H2.AZ	0.403	0.462	0.474
H3K36me3	0.393	0.444	0.479
H3K4me1	0.416	0.452	0.474
H3K4me3	0.551	0.577	0.605
DNase	0.530	0.556	0.573
RNA Pol2	0.493	0.520	0.564
H3K79me2	0.605	0.622	0.647
H3K9ac	0.573	0.594	0.624
H3K27ac	0.573	0.590	0.617
H3K4me2	0.557	0.576	0.595

Appendix B

Some codes for this study.

B1. Codes for simulation.

```
taos<-c(0.001,0.01,0.05,0.1,0.15,0.2,0.3,0.4,0.5,0.8,1,1.5,2);      #variation in random effects
sigma=1; beta = 1;groupN = 20;groupSize = 200;mu = 1;v = 1;
cor.matrix<-matrix(0,2,13);
library(nlme);

fixTerm = "Y~X"
randomTerm = "~ 1|gIndx";
loopn = 10;
cvNumber = 5;                    #5-fold cross-validation;

for(i in 1:13){
 tao = taos[i];
 cor.lme = cor.lm = matrix(0,cvNumber,loopn);
 X = Gamma = Epsilon = gIndx = c();
 for (i in 1:groupN) {
    gIndx = c(gIndx, array(i, groupSize));
    X = c(X, rnorm(groupSize, mean=mu, sd=v));      #Xs are independent;
    Gamma = c(Gamma, array(rnorm(1, mean=0, sd=tao), groupSize));
    Epsilon = c(Epsilon, rnorm(groupSize, mean=0, sd=sigma));
 }
 Y = beta*X + Gamma + Epsilon;                          #Xs are not independent when gamma added;
 dat = data.frame(Y, X, gIndx);


 for(g in 1:loopn) {                      #Repeat "loopn" times
  testObs = matrix(sample(1:groupSize, size=(groupSize %/% cvNumber)*cvNumber, replace=FALSE), ncol=
cvNumber);
   for (h in 1: cvNumber) {
    testIndx = c(); #get the index for test set
    for (k in 1:groupN) testIndx = c(testIndx, groupSize*(k-1) + testObs[, h]);
    dat.test =  dat[testIndx, ]; dat.train = dat[-testIndx, ];
    model.lme = lme(fixed = as.formula(fixTerm), random = as.formula(randomTerm), data = dat.train);
    model.lm = lm(as.formula(fixTerm), data=dat.train);

    predOut.lme = predict(model.lme, newdata=dat.test, level=1, na.action=na.omit);
    observed = dat.test$Y;
    cor.lme[g,h] = cor(observed, predOut.lme);

    predOut.lm = predict(model.lm, newdata=dat.test, na.action=na.omit);
    cor.lm[g,h] = cor(observed, predOut.lm);
   }
 }
mean(cor.lm)->cor.matrix[i,1];
mean(cor.lme)->cor.matrix[i,2];
}
```

B2. Codes for the second grouping method. With this function, we can find out which anchors are tethered together in the chromatin loop. Anchors tethered in the same chromatin loop are in the same group. For the next step, genes located around the anchors in the same group will be in the same group.

```
group_anchor <- function(pet,anchor_length,rep){

 pet_groups = vector("list",5);
 #a list of vectors to contain the group information. A group of pets will be represented by their index.
 #Each pet represent two anchors;
 pet <- pet[order(pet$chrom_left,pet$chrom_right,pet$left_middle,pet$right_middle),];
 #sorting pet according to their location
 rownames(pet) <- 1:nrow(pet);  pet -> pet0;
 for(i in 1:nrow(pet)){
  if(nrow(pet0)>0){           #Start with the first pet in pet0, if there is no more pet in pet0, then stop.
    PETindex = c();           #Store the index for the selected pets.
    petA = pet0[1,];          #Always start with the first pet in pet0;
    lmid = petA$left_middle; rmid = petA$right_middle; indx = as.numeric(rownames(petA));
    chrl = as.character(petA$chrom_left); chrr = as.character(petA$chrom_right);
                              #The attribute value of the first pet in pet0;
    PETindex = sort(c(PETindex,indx));
                              #The first pet in pet0 is always in this group;
    pet0 <- pet0[-which(as.character(rownames(pet0))==indx),];
                              #Remove the first pet in pet0 and this pet will be in group and never show up in pet0;
    index<-
as.numeric(rownames(pet0)[which(pet0$chrom_left==chrl&pet0$left_middle<lmid+anchor_length&pet0$left_middle>lmid-anchor_length)]);
    index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_right==chrr&pet0$right_middle<rmid+anchor_length&pet0$right_middle>rmid-anchor_length)])));
    index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_left==chrl&pet0$left_middle<rmid+anchor_length&pet0$left_middle>rmid-anchor_length)])));
    index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_right==chrr&pet0$right_middle<lmid+anchor_length&pet0$right_middle>lmid-anchor_length)])));
    #Pets in this group after the first loop

    for(g in 1:rep){          #Repeat this procedure "rep" times;
     n = length(index);
     if(n>0){                 #If there is no more pets added to this group, then stop.
      for(j in 1:n){
       petA = pet0[which(rownames(pet0)==index[1]),]; lmid = petA$left_middle; rmid = petA$right_middle;
       chrl = as.character(petA$chrom_left); chrr = as.character(petA$chrom_right); indx =
as.numeric(rownames(petA));
       PETindex <- sort(c(PETindex,indx)); index <- index[-1]; pet0 <- pet0[-which(rownames(pet0)==indx),];
       index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_left==chrl&pet0$left_middle<lmid+anchor_length&pet0$left_middle>lmid-anchor_length)])));
       index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_right==chrr&pet0$right_middle<rmid+anchor_length&pet0$right_middle>rmid-anchor_length)])));
```

```
        index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_left==chrl&pet0$left_middle<rmid+anchor_l
ength&pet0$left_middle>rmid-anchor_length)])));
        index<-
sort(unique(c(index,as.numeric(rownames(pet0))[which(pet0$chrom_right==chrr&pet0$right_middle<lmid+ancho
r_length&pet0$right_middle>lmid-anchor_length)])));
       }
     }
    }
    pet_groups[[i]] <- PETindex;
  }
 }
 return(pet_groups);          #A list of groups, each group here contains the index for pets. Actually, each group
contains the anchors represented by pets.
}
```

B3. Codes for the third grouping method. With this function, interactions (pair of pets) are merged according to their location. Interactions with shorter spans are merged into interactions with longer spans according the location. After merging, the interactions with high count will be used to group genes. The region between the two anchors of one interaction is one CTCF-domain and genes located in this region are in one group.

```
MergePets<-function(pet,sort,petspan,maxspan,anchor_region=1000){
#"Pet" is the input pet; Each pet represents an interaction;
#"Sort" is the sorting method. The pets are sorted before merging.
#"Petspan": The pets who have span close to "petspan" will be considered first;
#"maxspan": Only pets with span less than “maxspan” will be merged.

 pet <- pet[which(pet$intra_chrom==1),];     #Only pets in the same chromosome is considered;
 if(sort=="span"){pet <- pet[order(abs(pet$span-petspan)),]; }
                              #soring the pets according to petspan, the pet whose span is closest to "petspan" is in
the first place;
 if(sort=="count"){pet <- pet[order(-pet$count),];}
                              #Sorint the pets according to the count;
 rownames(pet) <- 1:nrow(pet); newpet <- c();  n <- nrow(pet);

 for(i in 1:n){
  if(nrow(pet)>0){
  lmid = pet$left_middle[1]; rmid = pet$right_middle[1]; chr = as.character(pet$chrom_left)[1];
                              #Starting with the first unmerged pet;
  pet0 = pet[which(pet$chrom_left==chr&pet$left_middle>(lmid-
anchor_region)&pet$right_middle<(rmid+anchor_region)),];
                              #Pets located in first umerged pet will be merged to it.
  if(rmid-lmid<maxspan){
                              #Only pets whose span are less than "maxspan" will be merged.
    pet <- pet[-1,];
    if(nrow(pet0)>1){
     for(j in 2:nrow(pet0)){
      pet0$count[1] <- pet0$count[1]+pet0$count[j]*pet0$span[j]/petspan;
                              #The merging pets will have a new count;
```

```
      pet <- pet[-which(rownames(pet)==rownames(pet0)[j]),];
                              #Pets already been merged won't be considered again;
     }
    }
    newpet <- rbind(newpet,pet0[1,]);
   }else{
    newpet <- rbind(newpet,pet[1,]);pet <- pet[-1,]
   }
  }
  }
  return(newpet);              #The pets after merging;
}
```

B4. Codes for the fourth grouping method. With this function, we can get the cutting point of chromatin domains.

```
GetDomainCutoff <-
function(contact,resolution=5000,chr,rep=5000,Max_length=1000,Min_frequency=20,overlap=10){

#Input: "contact" file has three columns, the first column is the locus on the left and the second column is the locus on the right, the third column is the contact frequency;
#"contact" is the contact frequency matrix downloaded from GEO;
#"resulotion" is the Hi-C experiment resolution;
#"chr" is the input chromosome;
#"Max_length" is the maximum length for chromatin domains;
#"Min_frequency" is the minimum contact frequncy between two loci;

 contact[,1:2]<-contact[,1:2]/resolution;
 contact <- contact[-which(contact[,1]==contact[,2]),]; #Remove contact between the locus and itself;
 contact <- contact[which(contact[,3]>cutoff_freq),];   #Only keep contacts which have large contact frequency

 cut=0;n<-nrow(contact);
 for(i in 1:nrow(contact)){
  if(nrow(contact)>0){
   left <- contact[1,1];                    #The first left locus is the left cutting point of this domain;
   indx <- which(contact[,1]==left);   #The loci having contact with the first left lucus are all in this domain;
   right <- max(contact[indx,2]);       #The maximum position of these loci is the right cutting point;
   contact<-contact[-indx,];
   for(i in 1:rep){                         #Repeat "rep" times;
    if(right-left<Max_length){
     indx=which(contact[,1]<=right+overlap);   #The loci having contact with the maximum locus are in this domain and have a new right cutting point;
     if(length(indx)>0){                #If there exists loci having contact with the maximum locus;
      right <- max(contact[indx,2]);
      contact <-contact[-indx,];
     }
    }
   }
   if(right-left<cutoff_length){
    indx<-which(contact[,1]<=right+overlap);
    if(length(indx)>0){
     contact<-contact[-which(contact[,1]<=right),];    #The loci located in this domain won't be considered for the next domain;
```

```
      }
      cut=c(cut,left,right);                    #The cutting point for chromatin domains;
     }
    }
   }
  cut<-data.frame(chrom=chr,cut=cut);
  return(cut);
}
```