Predicting and understanding inter-locus DNA interactions

by

Iris Xu

S.B. in Computer Science and Engineering and in Mathematics
Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering
and Computer Science
in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

September 2014

Copyright 2014 Iris Xu. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute
publicly paper and electronic copies of this thesis document in whole or in
part in any medium now known or hereafter created.

Signature redacted

Author ..............................................................

Department of Electrical Engineering and Computer Science

Signature redacted

Certified by............. ........................................

Manolis Kellis, Professor, Thesis Supervisor

Signature redacted

Accepted by ..... ........................................

Prof. Albert R. Meyer
Chairman, Master of Engineering Thesis Committee
Predicting and understanding inter-locus DNA interactions

by

Iris Xu

Submitted to the
Department of Electrical Engineering and Computer Science

September 8, 2014

in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

Most computational methods analyze DNA as a linear sequence of information when in fact the 3D architecture and organization contains important structural and functional elements that provide valuable information on DNA’s role in mediating cellular processes. However, this 3D conformation has been extremely difficult to profile, requiring vast experimental resources and limiting the number of cell types for which it becomes available.

In this thesis, I seek to address this limitation using a computational approach for predicting 3D conformation using diverse genomic annotations that are much more easily and broadly available. Hi-C maps for lineage-committed IMR90 cells and pluripotent H1 cells will provide information on features that are inherent to long-range interactions in all cell types as well as in specific cell types. Previous work in the lab used support vector machines on 5C data to find important features. While SVM performance is competitive, it becomes difficult to reveal which features are useful.

I use alternating decision trees, a type of supervised learning technique that potentially provides a more transparent relationship between features, to analyze proximal and distal genome interactions to determine the sequence and regulatory elements that are important for these interactions. In particular, I separated the data set by interaction distances to investigate how the mechanisms for chromatin organization vary spatially. Additionally, I extended the alternating decision tree learning algorithm to model the distance-dependent nature of these interactions.

Thesis Supervisor: Manolis Kellis
Title: Professor
Acknowledgments

Much gratitude is due to my supervisor Professor Manolis Kellis. From having him as a professor as a freshman in 6.006 and doing my first UROP in his lab to having him again in 6.878 as a senior and doing research for this thesis, he has been a huge part of cultivating my interest in computational biology and in pursuing this project.

Extraordinary thanks to my mentor Wouter Meuleman. His insight has inspired me. His patience and support have always guided me in the right direction. Most importantly, his enthusiasm has made this project fun and exciting to work on.

Additional thanks to Nezar Abdennur for help in getting Hi-C data and the rest of the Kellis lab for answering any questions I may have.

And of course, thanks to my friends and family for their neverending support.
Contents

1 DNA interactions mediate cellular function .......................... 9
   1.1 Chromosomes are non-randomly organized in the cell nucleus .... 9
   1.2 Chromosome conformation capture technologies reveal DNA interactions . 10
   1.3 3D organization of the chromosome is hierarchical .................. 11
   1.4 Chromosome architecture effects cellular function .................. 13
   1.5 Initial prediction results on 5C Data ............................. 14

2 Building the dataset by interaction distance ......................... 17
   2.1 Selecting interactions based on cell types ........................ 17
   2.2 Encoding biological values into features .......................... 19
       2.2.1 Calculating motif probabilities from sequence data ........... 20

3 Modeling hierarchy of DNA folding using alternating decision trees .... 23
   3.1 Introduction to Alternating Decision Trees (ADT) .................. 24
       3.1.1 ADT Algorithm details .................................. 25
       3.1.2 Interpreting an ADT .................................... 26

4 Modifying ADTs to have linear decision nodes ....................... 29
   4.1 Introducing multivariate nodes .................................... 30
   4.2 Heuristics-based fine-tuning of distance-dependent decision rules ...... 31
   4.3 Limiting the exponential search space of potential decision rules from building an ADT ........................................... 31
   4.4 Generalized multivariate regression for decision nodes ............... 32
5 Different biological mechanisms at different DNA interaction scales 35
  5.1 Highly selected features are disjoint between distance groups ............. 35
  5.2 ADT-motivated model for proximal interactions .............................. 38
  5.3 Using STRING to validate the selected model and motivate protein-mediated DNA interactions ......................................................... 43

6 Improvements to feature encoding and multi-variate ADT 47
  6.1 Creating a more separable feature set for better classification .......... 47
  6.2 Targeting regulatory elements in DNaseI hypersensitive regions ......... 48
  6.3 Using heuristic-based approaches to reduce runtime of multi-variate ADT algorithm ................................................................. 49
  6.4 Modeling the selection of decision rule coefficients as a linear program to efficiently minimize loss for the multi-variate ADT algorithm .......... 50

7 Conclusions 53

A Supplement 55
  A.1 Important ADT features in additional scenarios ............................. 55
  A.2 Foreground and background feature usage by interaction distance ....... 55
Chapter 1

DNA interactions mediate cellular function

1.1 Chromosomes are non-randomly organized in the cell nucleus

Chromosomes are incredibly complex units of information segmented into hundreds of domains with distinct functional characteristics. These include functional elements and disease-causing regulatory variants found throughout the non-coding region of the genome [1]. Understanding transcriptional control of gene expression has been one of the central problems of cell biology.

Strikingly, the genomes of most species are non-randomly organized and show distinct patterns in sequence composition. Regulating genes requires complex machinery to achieve subtle and precise timing and quantity of expression in the crowded nucleus and to avoid misregulation. In addition to a promoter, enhancer and repressor elements may reside in introns or upstream and downstream of the transcription unit.

Chromosomes consist of hundreds of domains with different protein compositions, with spatial organization characterized by many proximal and distal associations. In more compact genomes, such as yeast, a gene and its regulatory elements form an uninterrupted regulatory expression unit [2]. However, in more complex genomes, such as that for hu-
mans, regulatory elements and their target genes can be located throughout large genomic regions [3], [4]. For genes with complex expression patterns, often those that play a key role in developmental control, the regulatory domains can extend long distances outside the transcribed region. Enhancers were discovered to often be located far from the genes they regulated. In some cases, powerful enhancers were discovered that could activate clusters of genes, one model example being the locus control region of the beta-globin locus [5].

These large strands of information roughly 6 billion base pairs long are folded into a cell nucleus with a diameter of 6-20μm [6]. How the folded chromosome maintains accessibility to this content, in addition to mediating regulatory mechanisms and cellular function remains largely an open question [7]. It was proposed that these spatially distant regulatory elements could make physical contact with their target promoters, forming a looping interaction helped by site-specific DNA binding complexes. These loops have been shown to facilitate the recruitment of regulatory factors, histone remodeling activities, or RNA polymerase to the target region. Chromosome conformation capture (3C) and related technologies confirmed this model and showed that regulatory DNA interactions can be observed over much larger distances.

1.2 Chromosome conformation capture technologies reveal DNA interactions

Genome-wide studies have provided the opportunity to analyze the linear and 3D conformation of genomes. The advent of chromosome conformation capture (3C) technologies in the past decade has allowed analysis of any two interacting DNA fragments in close physical proximity, enabling the investigation of higher-order genome architecture at unprecedented resolution and throughput.

3C and related 5C [8] and Hi-C [9] are techniques that capture these possibly linearly distant associations by formaldehyde crosslinking in the nucleus followed by high-throughput identification of cross-linked fragments, which depends on the physical proximity of chromatin fragments in interactions. Formaldehyde is added to cells to link DNA fragments in
close proximity. A restriction enzyme is added to the cross-linked fragments to separate interacting regions from the rest of the chromatin. The cross-linked fragments are then ligated into a hybrid molecule before purification and amplification. While the underlying technique is the same between the various 3C methods, they differ in how the hybrid molecule is detected and quantified (Fig. 1-1).

In 5C, universal primers are used to identify potentially millions of interactions between two large sets of restriction fragments. The resolution of the 5C data relies on the type of restriction enzyme used and analyses can cover large portions of the genome. In Hi-C, biotin is used to fill DNA ends of digested fragments. This then allows ligation of ends and fragmentation to reduce size before biotin pulldown and deep sequencing. Hi-C provides an unbiased all-by-all genome-wide interaction map and resolution depends on the number of reads.

1.3 3D organization of the chromosome is hierarchical

Hi-C maps have revealed the hierarchical nature of chromosome organization [9]. Data suggests a model with different levels of organization and function at each level of the scale (Fig. 1-2). At the highest level, individual chromosome occupy separate spatial territories.
Within these territories, Hi-C data also confirmed the separation of the genome into open 'A' and closed 'B' chromatin compartments, typically 3Mb in size.

Further, each compartment contains regions of DNA organized into a unit of genome organization termed Topologically Associated Domains (TADs), which are Mb-sized modules defining fragments with higher probabilities of interaction, but which are separated from neighboring TADs by insulating elements. In this way, genes are confined to a small neighborhood of regulatory elements and interact with only a small fraction of the genome. Within TADs are sub-TADs and looping interactions at the sub-Mb scale that are characterized by additional indications of organization and interactions. These include long-range looping interactions between enhancers and promoters. Although intra-chromosomal interactions largely occur within TADs, inter-TAD loops have also been identified.

Evidence suggests that chromatin folding plays an important role in cellular function. While chromosomes occupy distinct territories, shorter-range DNA interactions may bring regulatory elements into close proximity with target elements. The relationship between these regulatory elements and distal gene targets remains largely unexplored. While DNA organization is still not fully understood, chromatin folding is non-random; protein complexes mediate interactions between distal genomic loci and specific anchor sites provide structure (Fig. 1-3). Understanding the mechanisms for DNA folding provides information about
these interactions, including regulatory mechanisms and how they act in concert to regulate genes.

1.4 Chromosome architecture effects cellular function

While both short- and long-range DNA interactions may play a role in gene regulation and genome organization, different organizing principles govern different levels of the folding hierarchy [10]. Between chromosomes, shorter chromosomes are more likely to interact with each other and not longer chromosomes. Within a chromosomal territory, studies have found that 'A' and 'B' compartments show significant differences between pluripotent and lineage-committed cell types [11]. This evidence suggests that these compartments may serve some function in determining cellular phenotype.

TADs generally remain unchanged through differentiation. One hypothesis is that the grouping of chromatin into TADs facilitates gene regulation by limiting the space a regulatory element searches for its target. Gene expression in embryonic stem cells for a 4.5 Mb region on chromosome X is more correlated within TADs than between TADs [12], suggesting that TADs group co-regulated genes. However, there are also many examples of co-regulated genes that are not contained within TADs, leaving the role of TADs unclear.

In contrast to the invariance of Mb-sized TADs, chromatin is reorganized at the sub-Mb level during differentiation. Intra-chromosomal looping interactions have been identified
within TADs. Both interactions that remain through differentiation and those that are present only in pluripotent cells and lost upon differentiation or only acquired upon differentiation in lineage-committed cells have been identified. Additionally, architecture changes significantly as somatic cells are re-programmed to induced pluripotent cells. These findings suggest that architecture is dynamic during differentiation and reversal, particularly at the sub-TAD scale.

As function changes at each level of organization, folding is driven by different classes of architectural proteins at different length scales within TADs. The existence of different classes of looping interactions suggests that different regulatory mechanisms are employed. Between the different cell types and length scales, a variety of CTCF- and cohesin-mediated looping occurs.

As chromatin is organized in a hierarchical manner, so are the mechanisms facilitating DNA interactions and their functional outcomes. Understanding how the role of DNA interactions varies at different levels of the organizational hierarchy becomes crucial in understanding how gene expression is regulated, and, further, the role genome architecture plays in maintaining cell activity.

1.5 Initial prediction results on 5C Data

Prior to me joining the group, previous work done in the lab by Wouter Meuleman used 5C data [13] from discovered interactions in the one percent of the human genome that is part of the ENCODE pilot project regions [14], including those between transcription start sites (TSS) and distal elements. These 5C maps were generated for GM12878, K562, HeLa-S3, and H1 cells, with tens of thousands of putative long-range interactions between promoters and distal elements, including some resembling enhancers, promoters, and CTCF-bound sites, and including information on the likelihood of the interaction. To generate the 5C maps, the study analyzed interactions between 628 transcription start site (TSS)-containing HindIII restriction fragments and 4,535 other ‘distal’ restriction fragments in the regions selected by the ENCODE pilot project, ranging in size from 500kb to 1.9Mb. Resolution was determined by HindIII cut sites and, in practice, was at ~ 1kb resolution for that 1
percent of the genome.

Due to the limited coverage of the data, interactions were selected using a stratified version of cross-validation in order to ensure independence of test and train sets. To create cross-validation folds, interactions were clustered based on overlap between interacting fragments and clusters were placed in the same train or test set. However, the process itself of choosing entire clusters of interacting fragments introduces bias into the train and test distributions.

The feature set was built using a binarization of sequence and regulatory elements associated with the interacting fragments, including chromatin states, protein motifs, transcription factor binding peaks, and DNaseI hypersensitive regions. Due to the non-polarity of interacting regions, features are independent of fragment order.

A support vector machine (SVM) [15] was used for the supervised learning algorithm. Performance curves for the learned models (Fig. 1-4) show that while performance was competitive in some scenarios of cell type and interaction type, it becomes difficult to build a biological model from selected regulatory features, particularly to show dependencies between factors. Additionally, quantifying the contribution a feature makes in classification is not straightforward.

From this study, we get a sense of the scenarios that are most difficult to predict, as well as the features that most-strongly contribute to DNA interactions, including AT-content, as well as some chromatin states and transcription factor binding events.

Our goal is then to provide additional insight into why DNA interactions occur, including what functional purpose they serve and how they are formed, and, more importantly, to build a biological model.
(a) Constitutive interactions for pluripotent cells (H1) and lineage-committed cells (GM12878) can be reasonably predicted using only sequence-based features.

(b) Cell-type specific interactions for pluripotent cells (H1) and lineage-committed cells (GM12878) are harder to predict. While pluripotent cell type interactions are still reasonably predicted by sequence-based features, transcription factor binding contributes to prediction power. Lineage-committed cell type interactions are much harder predict, with all features performing poorly.

Figure 1-4: The ROC curves and AUC values from previous work show that constitutive interactions can be predicted by sequence, and that cell-type specific interactions are more difficult to predict, particularly for lineage-committed cells. The performances for these scenarios will be used as a baseline for subsequent work.
Chapter 2

Building the dataset by interaction distance

Given that experimental evidence suggests that interaction mechanisms and function vary through different scales of chromatin folding, we attempt to model these underlying characteristics and processes for each level of the genome. Genome-wide Hi-C [9] interaction data were used, with maps generated for H1 Human embryonic stem cells [11] and IMR90 fibroblasts [16]. The maps provide a comprehensive and unbiased view of the interactions throughout the genome at a lower 10kb resolution. In order to investigate the biological mechanisms involved at different distance ranges, data were separated into six interaction groups by distance between interacting fragments: \{20 – 50kb, 50 – 100kb, 200 – 300kb, 450 – 550kb, 900 – 1100kb, 9800 – 10200kb\}. Each interaction was provided with the fraction of interactions that it represents, which was use as a confidence measure.

2.1 Selecting interactions based on cell types

The different cell types provide information on features that are inherent to long-range interactions in all cell types as well as in specific cell types. In particular, the project uses data from lineage-committed IMR90 cells compared to pluripotent H1 stem cells, which may have very different biological motivation for genome architecture at the sub-TAD level. By comparing the interactions in the various cell lines, we can see which interactions may be
Figure 2-1: Selected interactions for the foreground and background sets for the 20-50kb distance set for H1 constitutive interactions. Interaction distances and fragments sizes (all 10kb) are matched between sets, while maintaining separation in interaction frequencies and rank.

important for both differentiated and non-differentiated cell function versus interactions that are specific to pre- and post-differentiation.

In order to select foreground (strongest) and background (weakest) interactions for both cell-type specific and constitutive interactions, interactions in each cell line were ranked according to the available confidence measure, which was the fraction of interactions.

For constitutive interactions, the mean rank was calculated across cell lines, and the interactions were then re-sorted according to mean rank.

For cell-type specific interactions, we want to find the interactions that are strongest in the given cell line, but which are weak in other cell lines. Thus, the rank difference was calculated across cell lines, which is the difference in rank between the assessed cell line and the other cell lines, and then re-sorting was done per cell line according to this difference.

To eliminate dependencies between interactions based on sharing interacting fragments, overlapping interactions were removed. Interactions were sorted by mean rank, and the top 1000 non-overlapping interactions were used for the foreground set. In order to remove dependencies on interaction distance, which may be a confounding factor for other features, the background set was produced by matching interaction distances with the foreground set by starting from the lowest-ranked, non-overlapping interactions and selecting the first 1000 hits.
2.2 Encoding biological values into features

In order to provide biological motivation for DNA interactions, we analyze the sequence and regulatory elements associated with the interacting fragments. In addition to AT-content and interaction distance, we use a binarization of the occurrence of transcription factor motifs, transcription factor read peaks, chromatin states, and DNaseI read peaks. A prior is calculated for each feature by taking the probability of occurrence over the entire genome. The value $v$ for a given fragment in each sample is then divided by the prior $p$, and is binarized to 1 if the ratio is greater than 1 and 0 otherwise:

$$v_i = \begin{cases} 
1 & : \frac{v}{p} > 1 \\
0 & : \frac{v}{p} \leq 1.
\end{cases}$$

Thus, we have two binarized values $v_r$ and $v_f$ for the two interacting fragments. Because each interaction consists of two non-ordered fragments, the final calculated feature values attempt to lose as little information as possible and to remain independent of fragment ordering in the dataset. In particular, features consist of the sum as well as the absolute difference of binarized values for the two fragments. The goal is for the encoding process to be completely lossless so that the original values $v_r$ and $v_f$ can be reconstructed from the feature set. This gives us the following features:

$$f(v_r, v_f) = \begin{cases} 
v_r + v_f & \text{(Abin)} \\
|v_r - v_f| & \text{(Mbin)}. 
\end{cases}$$

The labels Abin (add binarization) and Mbin (minus binarization) describe the type of feature. In order to also model the co-occurrence of two interacting proteins with the two interaction fragments, the $L_1$ norm is calculated for the binarized values for two motifs for each fragment before also taking the sum or difference of these values. For values $v_{r,i}$, $v_{r,j}$ and $v_{f,i}$, $v_{f,j}$ for motifs $i$ and $j$, we also encode the following features:

$$f(v_{r,i}, v_{r,j}, v_{f,i}, v_{f,j}) = \begin{cases} 
L_1(v_{r,i}, v_{r,j}) + L_1(v_{f,i}, v_{f,j}) & \text{(coAbin)} \\
|L_1(v_{r,i}, v_{r,j}) - L_1(v_{f,i}, v_{f,j})| & \text{(coMbin)}. 
\end{cases}$$
The labels *coAbin* (added co-occurrence binarization) and *Mbin* (minus co-occurrence binarization) describe the type of co-occurrence statistic.

Additionally, we use probabilities of motif occurrence, calculated using a logistic regression model.

### 2.2.1 Calculating motif probabilities from sequence data

To model protein binding occurrences, we want to calculate a confidence measure for protein binding occurring within an interaction fragment. Because we do not have global protein-binding (ChIP-seq) data for all proteins, we use the probability that a motif occurs, given the fragment sequence, as a proxy for protein binding. Scores are calculated using work done by Segal [17], [18] and using position frequency matrices from TRANSFAC.

For each sequence, we have the motif binding event \( R \), which is *true* if motif \( i \) appears in the interaction sequence of length \( n \), which we call \( S = \{S_1, S_2, \ldots, S_n\} \). We have a position frequency matrix, which can then be converted to a *position specific scoring matrix*, which assigns a weight to each position in the motif and for each potential nucleotide \( l \in \{A, C, G, T\} \). This weight is the probability that that position of a motif is the nucleotide \( l \). Probabilities are taken over each of the possible \( n-p+1 \) motif positions, where \( p \) is the length of the motif and \( n \) is the length of the sequence. Then, using a standard binary logistic model, the probability of motif occurrence given the sequence \( S \) is:

\[
P(R = \text{true}|S_1, \ldots, S_n) = \frac{w_0}{n - p + 1} \sum_{j=1}^{n-p+1} \exp \left( \sum_{i=1}^{p} w_1[S_{i+j-1}] \right)
\]

where \( w_0 = \frac{P(R=\text{true})}{P(R=\text{false})} \), and where \( P(R = \text{true}) \) is the prior on binding occurrence.

A probability of motif occurrence is calculated for both fragments and summed. While we lose some information about the values for both fragments, the final feature is a value between 0 and 2 and gives an idea of the average probability of occurrence between the two fragments.

While the current feature set does provide information about potential regulatory and
binding events, potential improvements are described later.
Chapter 3

Modeling hierarchy of DNA folding using alternating decision trees

While there are a variety of supervised learning algorithms, the goal of this project is to train a model that can be used to infer biological processes, which involves modeling many dependent elements. After constructing the features for each interaction, we train a learning algorithm to make further classifications and to determine which features play a role in determining an interaction. We define our training set as \((x_1, y_1), \ldots, (x_m, y_m)\), where \(x_i \in \mathbb{R}^d\), where \(d\) is the number of features, and \(y_i \in \{-1, +1\}\). Here, we denote the foreground set (strong interactions) as +1 and the background set (weak interactions) as −1.

Decision trees provide a natural hierarchical view of dependencies between features from its tree structure. The classification algorithm consists of following a path through decision rules, e.g. Is AT-content > 0.5?, until reaching a terminal node, which gives the classification, i.e. \(y_i \in \{-1, +1\}\) in the binary case. For each sample, only one terminal node is reached. While decision trees can be successful, if a large number of features play an important role in classification, the tree can become very large and hard to interpret. The number of nodes grows exponentially in the number of required features. These problems are solved by using alternating decision trees (ADT) [19], a generalization of decision trees, voted decision trees, and voted decision stumps.
3.1 Introduction to Alternating Decision Trees (ADT)

A combination of boosting and decision trees, the algorithm creates trees that have comparable performance to other decision tree algorithms, such as C5.0 and CART, but also creates a smaller and more interpretable tree. Additionally, instead of outputting a classification of +1 or −1, ADTs output a score, given as a confidence measure for the classification.

Alternating decision trees consist of alternating layers of prediction nodes and splitter nodes. A sample is fed through the tree and decisions are made based on the sample’s features. A final score is produced by summing all the traversed prediction nodes. Usually classification is +1 if greater than zero, and −1 otherwise.

ADTs use boosting to make a strong classifier (tree) from many weak classifiers (decision stumps or decision rules). ADTs force dependencies between weak hypotheses by forcing new decision rules to build off existing rules. The example (Fig. 3-1) shows us how an ADT might predict heart disease, where a negative classification is disease. Only if the number-vessels-colored equals 0 will cholesterol play a role in classification. For a patient with the following features {thall = normal, number-vessels-colored = 0, chest-pain type is
asymptomatic, oldpeak = 2.5, cholesterol = 250, sex = male}, we can calculate a final score of
\[
0.062 + 0.541 + 0.425 - 0.536 - 1.495 - 0.444 = -1.447.
\]

3.1.1 ADT Algorithm details

Fundamentally, the algorithm builds a tree based on decision rules \( r \) based on the following form:

\[
\text{if (precondition) then}
\]
\[
\quad \text{if (condition) then output } p_1
\]
\[
\quad \text{else output } p_2
\]
\[
\quad \text{else output } 0
\]

where the set of base conditions is denoted \( C \).

Generally, the precondition is the conjunction of all the decisions that lead to this decision node. We let \( \mathcal{P}_t \) denote the set of all preconditions at time \( t \). The condition is the decision that is made at this decision node and we let \( R_t \) represent all the decision rules used at time \( t \). In the above rule, \( p_1 \) and \( p_2 \) are the predictions for the two children prediction nodes of this decision node. The selected output is denoted \( r(x) \in \{p_1, p_2\} \).

The algorithm re-weights the training examples at each iteration, where \( w_{i,t} \) is the weight for example \( i \) at time \( t \). Initially, \( w_{i,0} = 1 \) for all \( i \). Let \( W(c) \) represent the total weight of training examples that satisfy predicate \( c \), then \( W_+(c), W_-(c) \) are the total weights of examples satisfying the predicate, and classified as \( +1 \) and \( -1 \), respectively.

Then, at each round, loss is calculated and minimized based on precondition \( c_1 \) and condition \( c_2 \), as
\[
2(\sqrt{W_+(c_1 \land c_2)W_-(c_1 \land c_2)} + \sqrt{W_+(c_1 \land \neg c_2)W_-(c_1 \land \neg c_2)}) + W(\neg c_2).
\]
After selecting $c_1$ and $c_2$, formulas for calculating the best predictions for a given partition of the input space gives the two prediction nodes:

$$p_1 = \frac{1}{2} \ln \frac{W_+(c_1 \land c_2)}{W_-(c_1 \land c_2)}, p_2 = \frac{1}{2} \ln \frac{W_+(c_1 \land \neg c_2)}{W_-(c_1 \land \neg c_2)}.$$ 

Finally, the set of preconditions and the set of rules are updated to include the new decision rule, and weights of training examples are readjusted:

$$w_{i,t+1} = w_{i,t} e^{r_t(x_i) y_i}.$$

Training generally continues until test error converges.

For this project, JBoost [20], a java implementation was used, using AdaBoost for the boosting algorithm. The only input parameter is the number of rounds of training.

### 3.1.2 Interpreting an ADT

Most simply, each decision node can be evaluated on its own, where the magnitudes of $r(x)$ give some indication of the importance of that decision rule in classification.

However, the use of a decision tree over the original SVM is for the structure it gives to the selected features. The hierarchical nature of the tree allows representation of dependencies between decision rules. A path through the tree can be interpreted as a network of dependencies between features. In the heart disease case, the patient’s sex is only relevant if the chest pain-type is not asymptomatic.

Correspondingly, parallel subtrees represent decision paths that are more independent. Otherwise, the parallel path could have been included as a child node instead of as separate child of the root node. In the example, the results of a Thallium (thal) heart scan are independent from the number of colored vessels and cholesterol levels.

The prediction scores give a confidence measure for how predictive that predicate is. Intuitively, while traversing the tree, if the cumulative score is high or low enough, regardless of the remaining nodes, the prediction will remain unchanged. Thus, scores with greater magnitude are more predictive or provide more confidence toward a prediction.
By using ADTs, while there may be tradeoffs in performance or computational cost depending on the dataset, we gain interperability of a model with many dependencies.
Chapter 4

Modifying ADTs to have linear decision nodes

Literature and experimental evidence both suggest that the mechanisms determining interactions are usually distance-dependent, with interaction distances ranging from one kilobase to megabases. One way to incorporate this dependency is by modifying JBoost to use multivariate decision nodes in order to separate this distance dependency from other features.

Multivariate decision trees have been well-studied since the 1980’s [21] and implementations using multiple decision features per rule have been used to possess various properties dependent on purpose, such as reducing bias [22]. Here, we provide a simple adaption of the JBoost code to allow decision nodes which are linear functions of the feature in consideration.

Ordinary decision trees can be rigid in terms of classification and may require many nodes to reach a decision since they use only axis-parallel planes to divide the sample space. Additionally they do not efficiently represent dependent relationships between features. In this project, I introduce a linear alternating decision tree to more conveniently encompass these relationships. Current results suggest that it may be beneficial to create a learning algorithm that takes interaction distance as a parameter. In particular, it may be that at different levels of the hierarchy of DNA structure, interaction behavior changes, making different features important. In order to account for this change, we create a learning algorithm that adapts to these differences.
Figure 4-1: Linear decision rules can more efficiently split a sample space. In this example, a single linear decision rule splits the samples. This would take many more axis parallel splits (Brodley et al. 1995).

4.1 Introducing multivariate nodes

We do this by adapting the alternating decision tree algorithm to use linear functions at decision and prediction nodes instead of values drawn from samples. Each node is a function of the chosen parameter, in this case interaction distance, and the feature at that node. In the boosting steps, decision nodes would be chosen from the space of linear functions of the form $mx + ny + p > 0$, where $m$, $n$, and $p$ are constants that minimize loss, $x$ is the distance (or the feature that we want to use as a parameter for the tree) and $y$ is the feature being evaluated at that node. This can be rewritten as $-ny < mx + p$ or $y > -\frac{m}{n}x - \frac{p}{n}$. Since this is a decision rule, both this and the converse $y < -\frac{m}{n}x - \frac{p}{n}$ are included in a single rule. Now, we can let $a = -\frac{m}{n}$ and $b = -\frac{p}{n}$, and write the decision rule as $y < ax + b$.

We consider a sample scenario with two values that vary with interaction distance and how our decision tree presents this information. Biologically, AT-content may be an indicator or necessary component for long-range interactions, whereas motif occurrences and protein interactions may be more important for short-range interactions.

Then a decision node for AT-content may be $y < 0.1x + 0.5\dagger$, where $y$ is AT-content, with output $-1$ for yes and $+1$ for no. Conversely, a decision node for motif occurrences may be $y < -0.1x + 0.5\dagger$, where $y$ is motif probability, with output $+1$ for yes and $-1$ for no.
4.2 Heuristics-based fine-tuning of distance-dependent decision rules

Literature on multivariate decision nodes suggests many methods of selecting coefficients for decision rules [21]. Many involve searching for local minima/maxima for performance using heuristics and linear programming. However, in order to simplify the implementation and resource usage, our modification searches through a preselected range of values. Because our $y$ values fall between 0 and 2, we want to first scale $x$ so that it is on that order of magnitude. This gives us an upper bound for $ax$, and thus, an upper bound for $a$. For example, if $x$ is on the order of magnitude of $10^5$, then we must first scale $x$ by $10^{-4}$ or less to get within range of $y$. Then we can choose a range of appropriate values of $b$ to search through. Ideally, the range of $b$ values to search over would be the same as what was done for the constant features except for $y - ax$ instead of for $y$. We search through corresponding negative values for the coefficients as well.

4.3 Limiting the exponential search space of potential decision rules from building an ADT

While implementations exist for multivariate decision trees, one of the greater technical challenges is making the search for a linear function for each decider node computationally efficient. The set of base rules ("weak hypotheses") that are considered grows with each round of training. The set of existing potential base rules is inherited by a child decision rule except for the rule that caused the split. Then it only inherits the rules that potentially apply to the portion of samples that reaches it.

The current implementation in JBoost uses different types of splitter and predictor node objects. It enumerates potential splits during a round of boosting and the split that minimizes loss is chosen. In that case, the search is done over feature values. For example, if a split is being built on $AT$-content, then the potential split points are taken from the training set’s $AT$-content values.
The technical challenge here comes from efficiently finding $a$ and $b$ to form a split. In our case, there are now an infinite number of values that each constant can take on. To make this more practical, we limit our search space to powers of two over some practical positive and negative range between `Double.MIN_VALUE` and `Double.MAX_VALUE`. Limiting to 20 powers of 2 for each value would create $20^2 = 400$ candidates. The number of actual powers will depend on the dataset and resource availability.

While in the original implementation, the number of splits that a child decision rule inherits decreases, the number of rules in the current multivariate modification remains constant. Thus, depending on the size of the sample set and the size of the constant, this can mean either more or less computation time. Currently our implementation uses

$$A = \{ \pm 0.0001, \pm 0.00001, \pm 0.000001, \pm 0.0000001, \pm 0.00000001, 0 \}$$

and

$$B = \{ \pm 2, \pm 1, \pm 0.5, \pm 0.25, \pm 0.125, \pm 0.0625, 0 \}$$

as potential values for $a$ and $b$.

### 4.4 Generalized multivariate regression for decision nodes

Ideally, performance should be no worse than for the original implementation and potentially much better for a dataset where many features are dependent on one feature. Because of our simplification in searching for coefficients, the current performance is only roughly on par with the original implementation, but computation is slower.

Currently, the range for $b$, $B$, is more coarse than it could be. This implementation still needs tuning in terms of coefficient selection and how to efficiently limit our search space.

Additionally, instead of minimizing the loss as described by Freund, Brodley [21] gives four candidate methods for splitting at each decision node, including the Recursive Least Squares procedure, the Pocket Algorithm, the Thermal training procedure, and explicit reduction of impurity (CART). Theses splitting methods may provide additional benefits over the current implementation.
We have chosen to use decision nodes with a single parameter value in order to simplify the process and limit computational costs. However, the algorithm can easily be extended to use many more parameters. Even in the space of linear decision nodes, we can create decision nodes that are limited to one other feature or decision nodes which are linear functions of many features. For example, our decision nodes can consist of linear functions of the form:

\[ y_1 < a_1 x_1 + b_1, y_2 < a_2 x_2 + b_2, \ldots, y_n < a_n x_n + b_n \]

where each \( x_i \) is chosen from a set of parameter features. Alternatively, we could have

\[ y_1 < a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + a_0 \]

where again the \( x_i \) are a specified set of \( n \) parameter features. This would give us a more generalized multi-variate tree, but at the cost of additional time and/or memory.
Chapter 5

Different biological mechanisms at different DNA interaction scales

A java implementation of the ADT learning algorithm, JBoost, using AdaBoost for the boosting algorithm was run for 100 rounds using 10-fold cross validation. Cross-validation was used to assess performance of the learned models, and the number of rounds was selected by looking at both train and test error and selecting a point of convergence. After analyzing performance, we can select the learned models with good performance to build a final classifier.

The scenarios include comparing constitutive interactions that are highly-conserved in both differentiated IMR90 and pluripotent H1 cell lines to motivate what types of interactions are instrumental to both cell types. Cell-type specific interactions that are primarily present in one cell type are analyzed to determine which interactions are important in pluripotent versus lineage-committed cell types.

5.1 Highly selected features are disjoint between distance groups

Feature importance was determined by ranking features based on number of cross-validation trees they occurred in. Further, feature usage across foreground and background samples was
Figure 5-1: Cell-type specific interactions in IMR90.
Figure 5.2: Constitutive interactions in Hey.

CDPCR1_01 SREBP_Q3
NKKX2.5_01 ATF4_02
coAbin_Tx_EnhG OCT1_06
TBX5_Q6 AP4_Q6_01
IRF7_01 GATA3_03 Adh_Ques
ATF4_01 FOXO4_01 OCT1_03
IRF7_01 OCT1_06
HES1_Q9 CEIPEDELTA_Q6
SREBPQ3NKX25 Q2
ATF4 Q2
coAb-in Tx EnhG
OCT1 06TBX505AN406
IRF7 01
GATA3_03
Adh_Ques
ATF4_01
FOXO4_01
MSX1_01
E2A_02
OCT1_06
PLZF_02
HES1_09
CEIPEDELTA_Q6
AML1_01
STAT6_01
LUN1_01
coAbin_Tx_Quies
EVI1_01
EGR5_01
HNF4ALPHA_Q6
coAbin_Enh_Ques
HIF2A_06
HDAC3_01
CTX_03
MEIS1BH6X9_02
WHN_B
coAbin_GATA_known8_STAT_known13
CEBP_C
STAT6_02
ZTA_02
coAbin_TxWk_Quies
ATF3_06
T3R_06
coAbin_Het_TssBIV
coAbin_JUN_RF5X
APNT_02
coAbin_RepPCWk_Quies
coAbin_HW-TNK_Quies
GATA2_02
OCT1_06
BRCA3_01
MIF1_01
STAT1_03
STAT1_01
TITF1_03
TCF4_Q6
DBP_Q6
MAZ_Q6
STAT_01
compared to see what feature values are predictive of interactions. Features names refer to either a TRANSFAC motif name (e.g. P300_01) or the sum (A) or difference (M) between different types of either continuous (mean, min, max) or binarized (bin) statistics calculated for individual elements or co-occurrences, as described in chapter 2.

In each scenario (Figs. 5-1 and 5-2), we include three panels. The first panel shows the class, averaged training set classification, and the test set classification for 10-fold cross validation. The second panel selects the top 10 most selected features across folds for each distance and aggregates the results. The third graph shows ROC curves across the ten folds and gives the average AUC.

Remarkably, all scenarios exhibit highly disjoint sets of the most-used features across distance groups. This suggests that the interaction mechanisms may be very different at different interaction scales. For cell-type specific interactions in lineage-committed IMR90 cells, we see that the co-occurrence of a transcribed region and an enhancer is a commonly selected feature in all folds, as well as the occurrence of the motif for protein EP300, which is involved in enhancer looping.

We can also see that performance declines as interaction distance increases. While short-range interactions may be due to more targeted mechanisms, long-range interactions may be less precise and more related to larger-scale sequence features. We can also see from the plots for foreground and background values of the most-used features (Figs. 5-3 and 5-4) that separation between feature values decreases as interaction distance decreases. This also hints that the interaction process is less-targeted at longer ranges.

5.2 ADT-motivated model for proximal interactions

Using the hierarchical nature of the ADTs from the distance groups with high AUC values (20-50kb and 50-100kb interaction distances), we can assemble a biological model for DNA interactions by traversing paths through the tree and using the dependencies to infer regulatory associations. We can analyze the final trees (Figs. 5-5 and 5-6) generated for the models, which have been limited to 20 decision rules for simplicity.

In the tree for constitutive 20 – 50kb interactions in H1 cells (Fig. 5-5), we can see
Figure 5-3: Feature usage by foreground/background set for constitutive H1 interactions at the scale of 20-50kb.
Figure 5-4: Feature usage by foreground/background set for cell-type-specific IMR90 interactions at the scale of 20-50kb.
Figure 5-5: Final model for 20–50kb constitutive interactions in H1 cells.
Figure 5-6: Final model for 20 – 50kb cell-type specific interactions in IMR90 cells.
that highly quiescent sequences (decision node 2) are a strong indicator of non-interactions, or that the lack of these sequences is an indicator for interacting fragments. Additionally, following that path, we see that the lack of AP-4 (decision node 4), both a repressor and activator, is a strong indicator of background interactions. This suggests that regulators are important for DNA interactions at the 20 – 50kb scale.

Following the opposing child path from decision node 2, we see that the co-occurrence of transcribed regions (Tx) and genic enhancers (EnhG) on both strands (when this score is greater than 1.5) is a very strong indicator of a foreground interaction (prediction score of 1.588). Further, following this path to decision node 12, we see that the presence of Myc, a transcription factor, in addition to the presence of the co-occurrence of transcribed regions and genic enhancers on both fragments is an additionally strong indicator of DNA interaction.

The model from the selected path through the constitutive interactions in H1 (Fig. 5-7) is an example of a reasonable model for predicting strong and weak (or non-) interactions that can be learned using ADTs.

### 5.3 Using STRING to validate the selected model and motivate protein-mediated DNA interactions

STRING [23] is a database of proteins that includes protein-protein interactions and gives confidence scores based on literature and experimental data. STRING was used to build a network of interactions from highly selected motifs to give insight into potential protein-mediated interactions.

We input the proteins associated with the motifs for the same scenarios as before, for constitutive 20-50kb interactions in H1 cells and cell-type specific 20-50kb interactions for IMR90 cells, into STRING to see if we can identify any potential interactions (Fig. 5-8).

We see potential interactions between NKX2-5 and TBX5 in H1 cells, which are regulators for myocardial lineage and mesoderm differentiation, respectively. Additionally, for IMR90 cell-type specific interactions, we see that EP300 regulates chromatin remodeling and MAF
is involved in embryonic lens fiber cell development, and recruits EP300. Because these protein motifs act as strong indicators of interactions and the associated proteins co-regulate developmental processes, DNA interactions could act as potential mediators for recruitment of these proteins in these cell types at the 20-50kb scale.
(a) Network for constitutive H1 interactions at the scale of 20-50kb.

(b) Network for cell-type-specific IMR90 interactions at the scale of 20-50kb.

Figure 5-8
Chapter 6

Improvements to feature encoding and multi-variate ADT

The first most immediate goal would be to modify the feature set so that the features are no longer binarized, but, rather, use the cumulative distribution function of that feature to get a score. This would provide a better ordering of features, and, more importantly, better separation in the feature space. The second goal would be to tune the multivariate version of the ADT algorithm so that coefficient selection is more accurate and has more competitive classification and runtime performance.

6.1 Creating a more separable feature set for better classification

While the current feature set allows for reconstruction of the binarized values per interaction fragment, information is lost in terms of the strength of occurrence. For future work, the binarized values will be converted to a value between 0 and 1. For each feature $X$, a cumulative distribution function (CDF) $F_X$ will be calculated over the genome for each 10kb fragment. For ratio value $x$, the final feature value will be $F_X(x)$ for each fragment, and we can simply take the sum of the values for the two fragments. By using the CDF to calculate a score, we are able to maintain the order of the feature values between samples, which is
currently difficult with the 0, 1, and 2 binarized encoding because so many samples in both
the foreground and background sets have the same values. This becomes important when
trying to find decision rules to separate the sample space, which is important for accurate
classification and performance of the learned model.

While this new encoding will provide an ordering for feature values, we do lose the ability
to reconstruct the original values, which may be of biological importance. For example, if
the final sum for a motif occurrence is 1, this could mean that one fragment has very high
probability of motif occurrence and one is very low, or that both are average.

In addition to potentially improving classification, we will be able to reduce our feature
space significantly, which will make the process faster.

6.2 Targeting regulatory elements in DNaseI hypersensitive regions

Due to the large 10kb fragment size and that the features are binarized, the large region size
allows for peaks for many features to occur, even when not related to DNA interaction. In
order to more precisely target regions that are relevant to DNA interactions and potentially
improve the performance of our model, we plan to limit regions to DNaseI-hypersensitive
regions with open, accessible chromatin.

Feature values are calculated as before, except only over regions with DNaseI peaks. Priors are recalculated to use only regions with DNaseI peaks. Motif occurrence probabilities are recalculated for DNaseI peak regions for an interaction. For a set of DNaseI-masked regions, let the motif probabilities be \( \{p_1, \ldots, p_n\} \). Then the probability of occurrence over the entire interaction region is

\[
1 - \prod_{i=1}^{n} (1 - p_i).
\]

By limiting the analyzed interaction regions to only those with open chromatin, we are
improving the resolution of the data to only those regions where there is more information
on regulatory elements.
6.3 Using heuristic-based approaches to reduce runtime of multi-variate ADT algorithm

Current performance of our multi-variate ADTs is only on par with the original implementation while runtime has increased for some datasets. In order to improve performance, we must first fine-tune the method used to select coefficients for our linear distance-dependent decision rules, as the current search ranges are too coarse and lack the ability to find a decision rule that can efficiently separate the sample space.

As previously mentioned, we want to scale the value of our distance feature so that it can effectively re-index the original order of the dependent feature values. For example, if we have the following set of data:

<table>
<thead>
<tr>
<th>Sample</th>
<th>Int. Dist.</th>
<th>P(Myc Motif)</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1000000</td>
<td>0.05</td>
<td>+1</td>
</tr>
<tr>
<td>2</td>
<td>10000</td>
<td>0.9</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>1000000</td>
<td>0.1</td>
<td>-1</td>
</tr>
<tr>
<td>4</td>
<td>10000</td>
<td>0.95</td>
<td>-1</td>
</tr>
</tbody>
</table>

Our original ordering of the samples by increasing probability of Myc motif occurrence would be 1, 3, 2, 4, with classes +1, −1, +1, −1. This is very hard to find a point to separate the ordered values to create a decision rule. From the feature values, \( P(Myc) < c \) where \( c \in \{0, 0.05, 0.1, 0.9, 0.95\} \) are all potential decision rules, but none of them provide a satisfactory division.

However, if we add the distance multiplied by a coefficient, we can reorder the values. As stated previously, the linear decision rule is \( y < ax + b \), where \( y \), in this case, is probability of a Myc motif and \( x \) is interaction distance. We choose the coefficient \( a = -0.00000085 \) and \( b = 0.92 \) because the \( a \) scales the interaction distance to roughly the same order of magnitude as the Myc motif probabilities and because we suspect that motif probabilities decrease as interaction distance increases and \( b \) is chosen to separate the set. This then changes the decision rule for each sample:
We now see that the two foreground samples 1 and 2 both answer yes to their decision rule and the two background samples 3 and 4 both answer no, making them correctly classified.

### 6.4 Modeling the selection of decision rule coefficients as a linear program to efficiently minimize loss for the multi-variate ADT algorithm

Additionally, we can potentially use a linear-programming approach. The challenge will be finding a way to write loss as a linear equation in terms of selected coefficients \( a \) and \( b \). Assuming we can do so, we then are trying to minimize loss:

\[
\begin{bmatrix}
B
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
\]

Subject to:

\[
\begin{bmatrix}
x_1 & 1 \\
x_2 & 1 \\
\vdots & \vdots \\
x_n & 1
\end{bmatrix}
\begin{bmatrix}
a \\
b
\end{bmatrix}
\geq
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_n
\end{bmatrix}
\]

Here, we write \( x_1, x_2, \ldots, x_n \) as the interaction distances of the \( n \) samples that reach this decision node, and \( y_1, y_2, \ldots, y_n \) as the corresponding feature values to be evaluated. Because here \( a \) and \( b \) must be positive, we can rewrite the constraint equation to accommodate this. For example, if we want to test positive \( a \) values and negative \( b \) values, then we can rewrite
the constraint as:

\[
\begin{bmatrix}
  x_1 & -1 \\
  x_2 & -1 \\
  \vdots & \\
  x_n & -1 
\end{bmatrix}
\begin{bmatrix}
  a \\
  b 
\end{bmatrix}
\geq
\begin{bmatrix}
  y_1 \\
  y_2 \\
  \vdots \\
  y_n 
\end{bmatrix}
\]

where we have changed the 1 to a \(-1\) in the first matrix. There are efficient polynomial time algorithms for solving a linear program [24], so that if we can find \(B\) to write loss in terms of \(a\) and \(b\), we may be able to find an efficient splitting algorithm.
Chapter 7

Conclusions

Chromosomes are extremely complex entities that store billions of base pairs of information within the \( \mu m \)-scale diameter of the cell nucleus. How this vast string of genomic elements, including regulatory and transcriptional variants, is organized inside the nucleus while still maintaining the delicate regulatory balance required for proper cell function is largely an open question in cell biology.

Understanding chromatin architecture and organization, including the mechanisms behind and the functional outcomes of DNA interactions at every level of genome organization becomes critical to comprehending the role DNA plays in gene regulation and cell function.

Hi-C maps of the human genome have enabled investigation of this higher-order genome architecture at unprecedented resolution and throughput. In addition to these maps, genome-wide data for regulatory elements and chromatin states provides a richer landscape for genome-wide interactions.

In this project, I analyzed these Hi-C maps and the data for other genomic elements for pluripotent H1 and lineage-committed IMR90 cell lines. I separated the data by interaction distance and used alternating decision trees to build hierarchical models to gain insight into the complex machinery required for these potentially highly-specific interactions.

I found that the most-often-selected features for each distance range were highly disjoint, suggestive of distinct functional roles at each level of the chromosome organization hierarchy. Additionally, I was able to use the generated ADTs with high cross-validation performance to create biological models of DNA interaction mechanisms.
Further, I modified the original ADT algorithm to include decision rules that are linear functions of interaction distance. While the implementation of this modification is complete, I have future plans to make the algorithm more accurate and potentially more efficient using either a heuristics-based approach or linear programming.

The modified ADT algorithm in combination with a more informative feature set encoding can be used to build an accurate model of DNA interactions for all levels of genome organization.
Appendix A

Supplement

A.1 Important ADT features in additional scenarios

Here we include additional figures for the other scenarios comparing constitutive interactions in differentiated IMR90 cells and cell-type specific interactions in H1 cells (Figs. A-1 and A-2). We again see the disjointness in highly-used features between distances and can use this information to determine the differences in mechanisms at each interaction distance scale.

A.2 Foreground and background feature usage by interaction distance

We include feature usage for foreground (strong) and background (weak) interactions for all four scenarios: constitutive and cell-type specific interactions in H1 and IMR90 cell lines for highly selected features between cross-validation models (Figs. A-3, A-4, A-5, A-6). Each row of a scenario shows the ten most-selected features and density plots of their values. We see strong separation between foreground and background sets for shorter interaction ranges, which is consistent with the performance shown in Figs. 5-2, 5-1, A-1, and A-2.
Figure A-1: Cell-type specific interactions in H1.
Figure A-2: Constitutive interactions in IMR90.
Figure A-3: Density plots for feature usage for constitutive interactions in IMR90 cells.
Figure A-4: Density plots for feature usage for cell-type specific interactions in IMR90 cells.
Figure A-5: Density plots for feature usage for constitutive interactions in H1 cells.
Figure A-6: Density plots for feature usage for cell-type specific interactions in H1 cells.
Bibliography


