Statistical and Combinatorial Methods to Predict Gene Expression and Identify eQTLs from Haplotype Sequences

by

Pinar Demetci

Submitted to the Department of Computer Science in partial fulfillment of the degree of Master of Science in Computer Science at BROWN UNIVERSITY, Providence, RI, USA

Advisor: Sorin Istrail, Ph.D.
Julie Nguyen Brown Professor of Computational and Mathematical Sciences, Professor of Computer Science

© Pinar Demetci, May 2023
Abstract

Genome-wide association studies (GWAS) have discovered thousands of significant genetic effects on disease phenotypes. By considering gene expression as the intermediary between genotype and disease phenotype, expression quantitative trait loci studies have interpreted many of these variants by their regulatory effects on gene expression. However, there remains a considerable gap between genotype-to-gene expression association and genotype-to-gene expression prediction. Accurate prediction of gene expression enables gene-based association studies to be performed post hoc for existing GWAS, reduces multiple testing burden, and can prioritize genes for subsequent experimental investigation.

In this work, we develop gene expression prediction methods that relax the independence and additivity assumptions between genetic markers. First, we consider gene expression prediction from a regression perspective and develop the HAPLEXR algorithm which combines haplotype clusterings with allelic dosages. Second, we introduce the new gene expression classification problem, which focuses on identifying expression groups rather than continuous measurements; we formalize the selection of an appropriate number of expression groups using the principle of maximum entropy. Third, we develop the HAPLEXD algorithm that models haplotype sharing with a modified suffix tree data structure and computes expression groups by spectral clustering. In both models, we penalize model complexity by prioritizing genetic clusters that indicate significant effects on expression. We compare HAPLEXR and HAPLEXD with three state-of-the-art expression prediction methods and two novel logistic regression approaches across five GTEx v8 tissues. HAPLEXD exhibits significantly higher classification accuracy overall; HAPLEXR shows higher prediction accuracy on approximately half of the genes tested and the largest number of best predicted genes ($r^2 > 0.1$) among all methods. We show that variant and haplotype features selected by HAPLEXR are smaller in size than competing methods (and thus more interpretable) and are significantly enriched in functional annotations related to gene regulation. These results demonstrate the importance of explicitly modeling non-dosage dependent and intragenic epistatic effects when predicting expression.
Author’s Declaration and Acknowledgements

A large part of the work presented here has appeared in the *Proceedings of the Intelligent Systems in Molecular Biology (ISMB) Conference* in 2020 and has been published in the following paper (\(^\ast\) denotes co-first authorship and equal contribution):

- Berk Alpay\(^\ast\), Pinar Demetci\(^\ast\), Sorin Istrail, Derek Aguiar. (2020). *Combinatorial and statistical prediction of gene expression from haplotype sequence.* *Bioinformatics.* 36:Supplement_1, i194-i202.

This paper has been co-advised by my advisor Dr. Sorin Istrail and Dr. Derek Aguiar from the Department of Computer Science at the University of Connecticut. I thank both my advisors for their input on this work, as well as my co-first author, Berk Alpay from Dr. Aguiar’s lab, for the contributions he made towards the HAPLEXR method.
# Table of Contents

Abstract ii  
Author’s Declaration iii  
List of Figures vi  
List of Tables ix  

1 Introduction 1  
1.1 Background ........................................... 1  
1.2 Summary of Contributions and Overview ......................... 3  
1.3 Related Works ........................................... 4  

2 Methods and Algorithms 7  
2.0.1 Gene Expression Regression .................................. 8  
2.0.2 Gene Expression Classification ................................ 10  

3 Experiments and Results 20  
3.0.1 Continuous Expression ..................................... 21  
3.0.2 Discrete Expression ...................................... 26  

4 Conclusion and Discussion 31
5 **Supplementary Information**

5.0.1 Fast online suffix tree construction with tractized haplotypes. . . . 34
5.0.2 Discretizing Expression Values by Entropy Maximization . . . . . 37
5.0.3 Variant Set Enrichment Analysis . . . . . . . . . . . . . . . . . . . . 40

5.1 Supplementary Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
5.2 Supplementary Tables . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

References 56
List of Figures

2.1 Overview of the HAPLEX algorithm. (A) The tractized suffix tree is a suffix tree constructed from the tractized haplotype strings. Here, strings created by appending a unique terminating character $ to haplotypes $h_0$, $h_1$, and $h_2$ are tractized and inserted into the tractized suffix tree. (B) We penalize the complexity of our model by considering the clusters $z_1, \ldots, z_t$ induced by the tractized suffix tree ($t << p$) that most distinguish gene expression $\hat{y}$ with respect to $\chi^2$ or conditional entropy $H$. The penalization measure is selected by cross-validation and the resulting ranked clusters are denoted $z_1^*, \ldots, z_t^*$. (C) After the online insertion of a new sample into the tractized suffix tree, a graph is constructed using $z_1^*, \ldots, z_t^*$ to compute edge weights between haplotypes. (D) Spectral clustering algorithms are used to compute groupings of the graph vertices to render a discrete expression prediction for the new sample. Here the prediction is denoted by red and simplified to a single haplotype. .......................................................... 12

3.1 Comparing $r^2$ and narrow-sense heritability across continuous methods. For each gene, an estimate of narrow-sense heritability ($h^2$) in blue, and regression $r^2$ on the test set in red. We compared $h^2$ and $r^2$ across gene expression in whole blood for (top-left) KNN, (top-right) PrediXcan, (bottom-left) lasso, and HAPLEXR (bottom-right). .............................. 23

3.2 Gene-wise improvement of $r^2$. The improvement in $r^2$ between HAPLEXR compared to KNN (top), PrediXcan (top-middle), and lasso (middle-bottom) sorted by improvement per gene. Bottom: improvement in $r^2$ between PrediXcan and lasso sorted by improvement per gene. ............. 28
3.3 **Heatmap for the significance of enrichment across tissues and annotations** *(Supplementary Table 5.2)* for a $|\beta|$ threshold of 0.1. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected). Variant set naming convention indicates the tissue and type of variant (s, h, and b, denoting SNP, haplotype, and both respectively).

3.4 **For each method and $E \in \{2,3,4\}$, the distribution of per-gene expression classification accuracy over all tissues.** Paired one-tailed t-tests of HAPLEXD classification accuracy with each other method for $E = 3$ and $E = 4$ all have $p < 8.14 \times 10^{-8}$.

5.1 **Selection of partition size.** Pearson correlation between actual and predicted whole blood expression on 100 randomly selected genes on chromosome 1 across settings of the haplotype partition size ($J$) used in the haplotype clustering algorithm. Genes whose associated model had no nonzero regression coefficients are excluded from the plot; the number of these genes per setting are 33, 29, 33, 30, and 30 respectively. The center line denotes the median, the box ends the first and third quartiles, and the whiskers extend $1.5 \times IQR$.

5.2 **Gene-wise $r^2$ improvements (KNN).** Gene-wise improvement of $r^2$ between KNN-predicted and actual continuous expression compared to PrediXcan (top), lasso (middle), and HAPLEXR (bottom).

5.3 **Gene-wise $r^2$ improvements (PrediXcan).** Gene-wise improvement of $r^2$ between PrediXcan-predicted and actual continuous expression compared to KNN (top), lasso (middle), and HAPLEXR (bottom).

5.4 **Gene-wise $r^2$ improvements (lasso).** Gene-wise improvement of $r^2$ between lasso-predicted and actual continuous expression compared to KNN (top), PrediXcan (middle), and HAPLEXR (bottom).

5.5 **Average features used per gene.** Across all tissues and for each continuous gene expression prediction method that uses linear regression, the average number of features per gene whose corresponding regression coefficients have magnitude above a given threshold. HAPLEXR’s vector-encoded haplotype memberships at one range of the genome are considered a single feature.
5.6 Venn diagram comparisons between sets of well-predicted genes. Each Venn diagram describes the intersections between continuous gene expression prediction methods of genes predicted above a given $r^2$ in a given tissue. PX is PrediXcan and HR is HAPLEXR.

5.7 Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.3. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).

5.8 Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.2. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).

5.9 Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.1. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).

5.10 Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.1. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).

5.11 Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.2. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).

5.12 Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.3. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).
List of Tables

3.1 For each tissue and method, the number of genes best predicted by the method, with $r^2 > 0.1$. ................................................................. 24

3.2 Micro-averaged $F_1$ score for each method and tissue, and across all tissues, with $E \in \{2,3,4\}$ (rounded to three significant figures). Bolded entries denote the highest $F_1$ score for a tissue. HD is HAPLEXD, HR is HAPLEXR, PX is PrediXcan, and LR-EN and LR-L are logistic regression with elastic net and lasso regularization respectively. .......... 27

5.1 Tag SNP counts across clusters. For each tissue and across all tissues, the average number of tag SNPs in each HAPLEXR haplotype cluster with regression coefficients in three intervals (columns). ......................... 54

5.2 Regulatory annotations used in the variant set enrichment analysis. All annotations were retrieved from [28] and [41]. ................. 55
Chapter 1

Introduction

1.1 Background

The ability to detect, prevent, and treat complex disease is enhanced by an understanding of the latent genetic and regulatory architectures of the phenotype related genes. Genome-wide association studies (GWAS) have identified thousands of associations between genetic variation and disease, providing evidence for particular genomic regions that influence complex traits [55]. However, identification of the molecular mechanisms that affect disease etiology and cause the genetic association remains difficult for a majority of these instances [84]. Motivated by the observation that most GWAS associations were discovered in non-coding regions and complex diseases are ultimately functions of molecular phenotypes, expression quantitative trait loci (eQTL) studies interpret genetic associations through their regulatory effects on gene regulation [31]. In cis-eQTL analysis, the normal-
ized and covariate corrected expression is regressed on the minor allele dosage for variants close to (typically 1Mb) the transcription start site (TSS) of the gene (i.e. cis-SNPs) [62].

Recent work has built prediction models based on the assumption that significant eQTL associations should explain variation in gene expression [7, 8, 29, 56]. The ability to accurately infer gene expression from genetic data (a) enables post-hoc gene based association tests for the hundreds of existing GWAS studies that lack gene expression data; (b) reduces the multiple testing burden in GWAS studies (approximately $10^4$ gene tests instead of $10^6$ variant tests); and (c) enables easier translation of findings to prioritize target genes for follow-up molecular experimentation. These methods assume that genetic variation either directly affects regulatory mechanisms by, e.g., altering transcription factor binding, or acts as a proxy for intermediate molecular phenotypes that influence expression, for example, variation affecting chromatin accessibility [57, 61, 21]. While these methods have great utility for prioritizing GWAS results, they have been shown to exhibit an accuracy near 0 for most genes in the Depression Genes and Networks [9] and Genotype-Tissue Expression (GTEx) cohorts [31, 51].

Explaining the gap between association and predictability requires interpreting how specific assumptions affect model robustness to varying genetic and regulatory architectures [24, 48]. All methods either explicitly or implicitly assume a particular disease model; for example, methods that predict expression as a linear function of independent common variants will misrepresent rare variant contributions to common disease or dominance effects [12, 15]. Missing or underrepresented (e.g. structural) variation that is not linked with typed variation violate minor allele dosage and linkage disequilibrium (LD) assumptions [69]. Further, intragenic epistatic interactions between variants can alter protein
conformation [6], while intergenic epistasis has been implicated in many complex human
diseases, including Alzheimer’s disease [16], type 2 diabetes [90, 19], autoimmune dis-
ease [87], and cancer susceptibility [27]. Both epistatic effects violate the independence,
linearity, and additivity assumptions of existing linear regression models.

1.2 Summary of Contributions and Overview

In this work, we consider the problem of gene expression prediction from novel modelling
perspectives. First, we introduce the gene expression classification problem, which assumes
expression can be partitioned into discrete classes. Both low and high expression groups
in RNA-seq data have previously been associated with disease risk [29, 93] and cancer
prognoses [80]. Further, allele specific expression and single-cell RNA-seq data commonly
include genes with multimodal expression [73, 47], and recently, methods have been devel-
oped to detect differential expression in discretized expression data [71].

Next, we present methods that relax the assumption of independence and additivity
between genetic markers, thereby modelling intragenic epistatic and non-dosage dependent
effects. Specifically, our methods consider shared haplotype segments (called tracts) that
are independent of allele dosages. In total, our contributions include:

• formalizing the gene expression classification problem;

• developing an expression discretization algorithm based on maximum entropy to
choose expression classes;
developing the HAPLEXD algorithm for gene expression classification, which captures the exponential haplotype tract sharing in a compact suffix tree model. We penalize model complexity by prioritizing clusters that affect gene expression. Finally, we represent the genetic effects on expression with a graph theoretic model that yields an efficient spectral clustering algorithm to classify unseen test data;

• developing the HAPLEXR algorithm for gene expression regression. HAPLEXR combines the strengths of a typical dosage model with haplotype clusters using a penalized linear model;

• demonstrating increased classification and regression accuracy on experimental data from five human tissues;

• interpreting our results with respect to regulatory annotations.

In Section 1.3 we describe prior work on predicting gene expression from genetic data. Chapter 2 describes the haplotype clustering models for classification and regression problems, penalization methods, and prediction algorithms. We present results in Chapter 3 and in a discussion of caveats, future directions, and open problems in Chapter 4.

1.3 Related Works

Given pairs of genetic sequences and normalized gene expression as training data, expression prediction models infer gene expression for previously unseen genetic sequences. Prior
methods make explicit modelling assumptions on how genetic variants interact to influence gene expression. Regularized linear models and K-nearest neighbor methods showed varying success in predicting the expression from immune precursor cells when trained on cis-SNPs from HapMap Phase II data [42, 78, 56]. Surprisingly, simple models, like using only the single SNP most correlated with gene expression, outperform similar linear models trained on all cis-SNPs for about one third of all genes. Non-linear models, e.g. K-nearest neighbors, demonstrate greater accuracy on some genes than regularized linear regression models, suggesting potential model improvements from relaxing the SNP dosage and additivity assumptions [56].

The seminal PrediXcan method imputes gene expression from genomic variants using an additive genetic model [29].

\[ Y_g = \sum_j w_{j,g} X_j + \epsilon \]  

(1.1)

where \( Y_g \) is the expression of gene \( g \), \( w_{j,g} \) is the effect size of variant \( j \) for gene \( g \), \( X_j \) counts the number of reference alleles for variant \( j \) across samples, and \( \epsilon \) is an independent error term capturing non-additive and non-genetic factors influencing expression. The effect sizes \( w_{j,g} \) can be estimated using penalized linear regression inference algorithms. While lasso was found to perform similarly to elastic net in estimating \( w_{j,g} \), elastic net produced results that were more robust to perturbations of the input variants [29].

Recent follow-up work suggests that there exists significant opportunities to improve existing linear dosage dependent models [51]. Firstly, methods based on penalized regression often infer regression models with all zero coefficients. This is reflected in the fact that
the PrediXcan DGN and GTEx models predicted the expression of only 11,538 and 6,695 genes in DGN and GTEx respectively. Secondly, most genes were found to have estimated accuracy ($r^2$) near 0. Existing methods have been shown to be useful in the prioritization of GWAS results, reducing multiple testing burden, and detecting new gene-to-phenotype associations [29]; but their usefulness with regards to prediction and imputation is fundamentally a function of their accuracy, which is limited by model assumptions.
Chapter 2

Methods and Algorithms

Linear regression methods assume that gene expression is a linear function of additive minor allele dosages. While computationally and statistically convenient, these assumptions preclude modelling non-additive gene-gene or variant-variant interaction effects (i.e. epistasis). In this section, we present two methods, HAPLEXR and HAPLEXD, that relax the additivity and independence assumptions on variant-variant interactions.

Let $H \in \{0, 1\}^{n \times p}$ denote the haplotype data matrix. We note that, although this representation of the haplotypes assumes biallelic data, our methods extend to non-biallelic sequences. For ease of exposition, we consider the problem of finding genetic predictors of gene expression for a single gene $g$ and the haplotype data is split into two sets: $n - 2$ reference haplotypes from $(n/2) - 1$ individuals and 2 test haplotypes from a distinct individual. Our methods can be applied to each gene independently, for which we will omit the $g$ subscript for convenience, and extend to more than 2 test haplotypes. Haplotypes
and individuals are indexed by $i$; haplotypes are denoted $h_i$ and have length $p$ defined by a $10^6$ bp window around the TSS of the gene. The haplotypes for individual $i$ are indicated by $(h_{2i},h_{2i+1})$. Each individual-gene pair has a corresponding normalized and covariate corrected expression value $y \in \mathbb{R}$; the collection of which is the column vector $Y$. Our goal is to learn a function of the haplotypes $f(H)$ that predicts gene expression.

Haplotype sharing of substrings, or tracts, is central for our algorithmic approaches. We define a tract for a pair of haplotypes as a shared substring that starts and ends at the same positions in both haplotypes. For example, if $h_i = 0011$ and $h_j = 1010$ are two haplotypes, then the substring 01 is a shared tract, as it starts at position 2 and ends at position 3 in both haplotypes. A common theme between HAPLEXD and HAPLEXR is to compute sets of shared tracts, called signature tract sets (STS), that are haplotypic predictors of gene expression.

### 2.0.1 Gene Expression Regression

We first consider the problem of predicting continuous expression from haplotype data. To estimate gene expression, RNA-seq reads are first mapped to the genome or transcriptome and converted to read counts. Read counts are typically normalized to control for gene lengths, the number of sequencing reads, batch effects, and statistical biases, e.g., PCR, GC-content, and genetic relatedness [17]. In eQTL analyses, the resulting expression vector $Y$ is typically assumed to be normally distributed after normalization [45].

**The Gene Expression Regression Problem.** Given a haplotype matrix $H \in \{0,1\}^{n \times p}$, and expression vector $Y$, find a function $f : (h_i,h_{i+1}) \mapsto \mathbb{R}$ for $i = 0, 2, \ldots, n - 2$
that minimizes some loss function $L(Y, \hat{Y})$ where $\hat{Y}$ is the predicted values of expression for haplotypes in $H$.

We develop the statistical model, HAPLEXR, based on signature tract sets to solve this problem (Supplementary Information).

**Genetic Clustering Model**

We cluster haplotypes using an algorithm similar to the SHAPEIT model [22]. Let $J$ be a positive integer denoting the marginal partition size of a genetic clustering. Consider the set of all unique haplotype sequences from index $j$ to index $l$; let this set be $H_{j \rightarrow l}$. The genetic clustering model starts at the first variant position $j = 0$, and grows the set $H_{j \rightarrow l}$ until $|H_{j \rightarrow l}| \geq J$. We then define a partitioning of the haplotypes using $H_{j \rightarrow l-1}$ as the clusters labels and insert each haplotype into a cluster if and only if its sequence exactly matches the cluster label. We iterate with $j = l$ and stop when $l = p$.

**Regression Model**

We represent cluster membership as a one-hot encoded feature in our model. Since humans are diploid, a single sample has two cluster membership vectors. We sum the two vectors for a single sample element-wise to generate the genetic model feature vector and append SNP dosages to create the design matrix $X_d$. We then fit an elastic net regression with penalization parameters $\lambda_1$ and $\lambda_2$ such that

$$\hat{\beta} = (||y - X_d\beta||^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||^2)^{-1} \beta,$$  

(2.1)
with \( \frac{\lambda_2}{\lambda_1 + \lambda_2} = 0.5 \). We perform 10-fold cross-validation to determine \( \lambda_2 \) with respect to mean squared error \([64, 29]\). The signature tract set is then identified by the set of variants with positive \( |\hat{\beta}| \) values.

**HAPLEXR Algorithm**

```plaintext
input : training haplotypes \( H \) and gene expression \( y \), test haplotypes \( H_t \), gene \( g \), and parameter \( J \)
output: expression \( y_t \in \mathbb{R} \) for test individuals

// Genetic model for gene \( g \)
idx ← SubsetVariants(\( g \))
\( G_g \leftarrow \text{GenModel}(H[idx]) \)

// Build prediction model
\( X_d \leftarrow (G_g,H[idx]) \)

// Regression
\( \hat{\beta} = \beta \left( ||y - X_d\beta||^2 + \lambda_1||\beta||_1 + \lambda_2||\beta||^2 \right) \)
\( X_t \leftarrow (G_g,H_t[idx]) \)
return \( \hat{\beta}X_t \)
```

**Algorithm 1:** The HAPLEXR method. For simplicity, we consider the algorithm for a single gene \( g \). The function \( \text{SubsetVariants} \) returns the indices of variants within a cis-window of \( g \). The function \( \text{GenModel} \) constructs the genetic clustering.

### 2.0.2 Gene Expression Classification

Next, we consider predicting discrete gene expression, for which we require classes of expression values. While discretizing gene expression can be implemented directly on the RNA-seq read counts, it is unclear how one could then correct for experimental covariates. Instead, we consider discretizing the covariate corrected expression from the continuous modelling section into \( E \) groups using the principle of maximum entropy.
Expression Discretization

We define expression discretization as the grouping of the \(n/2\) input expression values \(Y\) into \(E \in \mathbb{Z}_{>0}\) clusters. By sorting \(Y\) in ascending order, we can partition the elements into clusters with ascending mean expression by choosing \(E - 1\) breakpoints (with ties, if any, in the same cluster). Each clustering of the expression values induces a clustering of the \(n\) haplotypes. We choose a method for computing the \(E\)-clustering that is free from distributional assumptions based on the method of information entropy maximization [43] (Supplementary Information).

Let \(a\) be the average expression of the \(n\) haplotypes in \(Y\). We want to compute based on general principles (“maximum ignorance”) a partition of expression values \(Y\) into \(E\) clusters based only on the information given by \(n, a, E, \) and \(\sigma\), where \(\sigma\) is the set of \(E\)-averages for a particular partition of \(E\) clusters \(\sigma = \{a_1, a_2, \ldots , a_E\}\). Note that for any \(E\) there are \(\binom{n-1}{E-1}\) feasible \(\sigma\), assuming no empty clusters. We can reformulate this partition in terms of a random variable \(W\) with outcomes \(a_1, a_2, \ldots , a_E\).

Consider cluster \(i\) whose \(n_i\) elements have average expression \(a_i\). We view cluster \(i\) as a multi-set with \(n_i\) elements all equal to \(a_i\), that is, each expression value in the cluster is replaced by its discretized value (the cluster average). In this way, the random variable \(W\) has the set of outcomes \(\sigma\), and a corresponding discrete probability distribution defined by the \(E\)-clustering. That is, the probability \(p_i\) of an observation \((a_i)\) is given by the solution of the entropy maximization problem.

For each \(E \in \{2, 3, 4\}\), we approximate the maximum entropy solution for discretization of the expression values using a heuristic algorithm. We compare the entropy of 100
Figure 2.1: **Overview of the HAPLEX algorithm.** (A) The tractized suffix tree is a suffix tree constructed from the tractized haplotype strings. Here, strings created by appending a unique terminating character $\$ to haplotypes $h_0$, $h_1$, and $h_2$ are tractized and inserted into the tractized suffix tree. (B) We penalize the complexity of our model by considering the clusters $z_1, \ldots, z_t$ induced by the tractized suffix tree ($t << p$) that most distinguish gene expression $\hat{y}$ with respect to $\chi^2$ or conditional entropy $H$. The penalization measure is selected by cross-validation and the resulting ranked clusters are denoted $z^*_1, \ldots, z^*_t$. (C) After the online insertion of a new sample into the tractized suffix tree, a graph is constructed using $z^*_1, \ldots, z^*_t$ to compute edge weights between haplotypes. (D) Spectral clustering algorithms are used to compute groupings of the graph vertices to render a discrete expression prediction for the new sample. Here the prediction is denoted by red and simplified to a single haplotype.

randomized configurations to partition expression values and select the partition that yields the highest entropy $- \sum_i^E p_i \log_2(p_i)$ where $p_i$ is the empirical probability of a haplotype belonging to expression class $i$.

**The Gene Expression Classification Problem.** Given a haplotype matrix $H \in \{0,1\}^{n \times p}$, and discretized expression vector $Y_d$, find a function $f : (h_i, h_{i+1}) \mapsto \{1, \ldots, E\}$ for $i = 0, 2, \ldots, n - 2$ that minimizes loss function $L(Y_d, \hat{Y}_d)$ where $\hat{Y}_d$ is the predicted expression classes for haplotypes in $H$. Here, we develop a discrete mathematics model, HAPLEXD, to solve this problem (Figure 2.1).
**Tractized Suffix Tree**

Suffix trees are data structures for string representation and used for intra- and inter-string compression and pattern matching [34]. We summarize the sharing of haplotype segments, or tracts, and their gene expression in a *tractized suffix tree* [2]. The tractized suffix tree is a generalization of suffix trees over a finite alphabet $A$, and is defined as follows: a string $S$ over the alphabet $A$ is transformed into a string $S'$ of the same length, where each symbol $a$ at index $j$ of $S$ is replaced by a pair $(a, j)$ in $S'$. For our purposes, we can encode a haplotype $h_i \in \{0, 1\}$ as a tractized haplotype $h^t_i \in \{0, 1, \ldots, 2p\}$ where each integer incorporates the allele and positional information, i.e., $h^t_{ij} = 2j + h_{ij}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, p$. For example, the tractized haplotype for $h_i = 0011$ is $h^t_i = (0, 2, 5, 7)$.

Formally, a tractized suffix tree $G^T(V^T, E^T)$ is a rooted directed tree containing only tractized strings and having $O(np)$ leaves. The tractized suffix tree encodes the sharing of haplotype sequences between distinct haplotypes as internal nodes (Figure 2.1A). Vertices $v^T_k \in V^T$ correspond to a position in $\{1, \ldots, p\}$ and each node has exactly 2 children besides the root which has $\geq 2$ children. An edge $(v^T_k, v^T_l)$ is labeled with a non-empty common substring for a subset of tractized haplotypes. The tractized suffix tree’s characteristic property is that any root-to-leaf path corresponds to a suffix of a subset of tractized haplotypes. Importantly, tractized suffix trees allow inter-string compression while enforcing zero intra-string compression.

Given the tractized haplotype sequences $h^t_0, \ldots, h^t_{n-1}$, we construct a tractized suffix tree using a modified Ukkonen’s algorithm [82] (Supplementary Information). A *tract* is created by concatenating the edge labels on a root-to-node path and represents a shared
substring among a set of haplotypes. Note that tracts represent identical substrings in two or more haplotypes and are compressed in the tractized suffix tree if and only if all substrings start and end at the same position. Therefore, only inter-haplotype sharing is compressed in the tree.

**Tractized Suffix Tree Augmentation**

We build on prior work by augmenting the tractized suffix tree to support expression prediction. We label the tractized suffix tree vertices with sets of haplotypes in order to evaluate genomic tracts that affect gene expression in subsequent algorithmic steps. First, all children of the root are labelled with their set of descendent haplotypes. Due to the infinite sites assumption, all non-root, internal vertices have two children; let the parent and its two child nodes be denoted $v_p^T$, $v_{c_1}^T$, and $v_{c_2}^T$ respectively. Each internal vertex is labeled by the tractized haplotype indices that are no longer descendent from that vertex after traversing the edge from $v_p$. That is, $v_{c_1}$ is labeled with the tractized haplotype indices that are descendent from $v_{c_2}$ and conversely. We refer to a tree cluster as the two sets of haplotypes created by the diverging edges from a non-root internal vertex. For each of the $2p$ possible paths, a haplotype appears at most once in these sets, yielding a $O(np)$ space complexity.

**Construction of the Tractized Suffix Tree**

Although linear for constant sized alphabets, the McCreight and Ukkonen algorithms construct a suffix tree for a single $p$ length sequence and $O(p)$ alphabet in $O(p \log(p))$
time [58, 82]. Farach’s suffix tree algorithm closed the constant-polynomial sized alphabet gap, proving that this construction can be achieved in linear time [26]. However, Farach’s algorithm requires reading the full input at once and is considered to be largely a theoretical result due to large constants hidden in the complexity [72]. The construction of the tractized suffix tree was originally proposed using Farach’s algorithm, but, this construction is not online, a requirement for HAPLEXD. Using the lemma that follows and algorithm details in the Supplementary Information, we can construct an online tractized suffix tree for \( n \) tractized haplotypes each of length \( p \) in time \( O(np) \) using a modified Ukonnen’s algorithm.

**Lemma 1** Given an input tractized haplotype matrix of size \( 2^p n \), the number of nodes in the tractized suffix tree is \( < 2^p n \) for \( n \geq 3 \).

See Supplementary Information.

**Penalization of model complexity**

Given a decomposition of the expression for gene \( g \) of the \( n/2 \) individuals into \( E \) percentiles, our goal is to search for a set of shared tracts in \( T \) that captures the differences in assignment of haplotypes to expression percentiles (i.e. an STS). We parse the tractized suffix tree using a depth first search. We keep an active haplotype list which is set initially when we reach a child \( v_c \) of the root node \( v_r \) to the set of haplotypes descendent from \( v_c \). Consider a parent internal node \( v_P \) with two internal child nodes \( v_{c_1} \) and \( v_{c_2} \). When traversing from \( v_P \) to \( v_{c_1} \), we remove haplotype elements from \( v_{c_2} \). Likewise, when we traverse from \( v_{c_1} \) to
\(v_P^T\), we add haplotype elements from \(v_{c_2}^T\). Using the labels on the nodes that we created when constructing the tree, we can selectively remove or add sets of haplotypes to track the descendent haplotypes at any child node of the current node.

Consider an arbitrary internal vertex in the tractized suffix tree, which has 2 child vertices and recall our discretization of the normalized gene expression values into \(E\) classes. We evaluate the effectiveness of a tree cluster to separate expression classes using two methods (Figure 2.1B). In the first method, we create a \(2 \times E\) table where the cell \((i,e)\) counts the number of haplotypes in tree cluster \(i\) that have expression \(e\). For each tractized suffix tree node we compute:

- a \(\chi^2\) test statistic with \((E - 1)\) degrees of freedom, and
- the conditional entropy of the observed haplotypes in expression classes by normalizing the entries in the tree cluster contingency tables and interpreting them as empirical probabilities.

By retaining a subset of the tract tree clusters, we penalize the classification model complexity.

Spectral Clustering and Classification

The HAPLEXD classification model (a) creates a similarity matrix over haplotypes, (b) associates this matrix with an undirected weighted graph, and (c) classifies a new individual with respect to a spectral clustering of the graph [44]. Let \(G = \{V, E\}\) be an undirected graph with weights on the edges represented by \(W = (w_{i,j})\), the weighted adjacency matrix
of $G$. The vertices $v \in V$ represent haplotypes and the edge weights $w$ are defined by a similarity measure based on tract-sharing. We take the top $t$ clusters in the tractized suffix tree and create a similarity weight $w(h_i, h_j) \in [0,1]$ between haplotypes $h_i$ and $h_j$.

$$\begin{align*}
w(h_i, h_j) = \begin{cases} 
\frac{c(h_i, h_j)}{t}, & \text{if } c(h_i, h_j) \geq \frac{t}{r} \\
0, & \text{otherwise}
\end{cases}
\end{align*}$$

where $r$ is a regularization parameter and $c(h_i, h_j)$ counts the number of co-occurrences of haplotypes $h_i$ and $h_j$ in tracts across the top $t$ clusters (i.e., part of the STS). We represent the graph $G$ with weights $w(h_i, h_j)$ for $i, j = 1, \ldots, n$ as an $n \times n$ adjacency matrix (Figure 2.1C).

We implement the Shi-Malik normalized spectral clustering algorithm to group haplotypes with similar expression signatures [85]. Given the graph $G$, the similarity matrix (the weighted adjacency matrix of $G$) $W$, and the number of clusters $E$, we first compute the unnormalized graph Laplacian matrix be $L = D - W$. Next, we compute the first $E$ eigenvectors $v_1, \ldots, v_E$ of the generalized eigenproblem $Lv = \lambda Dv$ and the matrix $X = [v_1; \ldots; v_E]$. Let the rows of $X$ be $x_i, 1 \leq i \leq n$ where each row corresponds to a node in $V$. We cluster the $x_i$ with $k$-means clustering into clusters $C_1, \ldots, C_E$ and $V$ into clusters $A_1, \ldots, A_E$, where $A_i = \{ j \mid j \in C_i \}$. We assign expression groups to clusters based on co-occurrence with expression groups in the training data. Finally, we predict the expression of an individual by clustering the test haplotypes in $G$. In case the haplotypes of an individual are clustered in separate groups, we output the expression class with the highest purity (Figure 2.1D and Supplementary Information).
Given a similarity measure, the spectral clustering algorithm partitions the points of a set (nodes in the graph) into different subsets according to their pairwise similarities (edge weights). The algorithm partitions the graph by enforcing a bi-criteria optimization that maximizes “within cluster” similarity and minimizing “between cluster” similarity. In other words, the edges between different partition-subsets have a low weight (points in different subsets are dissimilar from each other), and the edges within a partition-subset have high weights (points within the same cluster are similar to each other). Formally, let the degree of node \( v_i \) be \( d_i \), and for a set of vertices \( A \subset V \), let \( \bar{A} = V - A \). The sum of the weights of edges between \( A \) and \( \bar{A} \) is denoted \( \text{cut}(A, \bar{A}) \), and the volume is \( \text{vol}(A) = \sum_{i \in A} d_i \). The Shi-Malik Normalized spectral clustering algorithm minimizes the objective function

\[
N\text{cut}(A_1, ..., A_E) = \sum_{i=1}^{E} \frac{\text{cut}(A_i, \bar{A}_i)}{\text{vol}(A_i)}.
\]

(2.2)
HAPLEXD Algorithm

**input**: training haplotypes $H$ and gene expression $y$, test haplotypes $H_t$, gene $g$, and parameters $E, t, r$

**output**: expression class $y_t \in \{1, \ldots, E\}$ for test individual

$H^t \leftarrow \text{tractizeHaps}(H)$

// Genetic model for gene $g$
idx $\leftarrow \text{SubsetVariants}(g)$
$T_g \leftarrow \text{Ukkonens}(H^t[idx])$

// Penalization
classes $\leftarrow \text{maxEntropy}(y,E)$
clusters $\leftarrow \text{SortedList}()$
for $c \in \{\text{DFS}(T_g)\}$ do
  $e \leftarrow \chi^2(c,\text{classes})$
  clusters.add((c,e))
adj_matrix $\leftarrow \text{float}[n][n]$
for $c \in \text{clusters}[0,\ldots,t-1]$ to $n$ do
  for $(i,j) \in c$ do
    if clusters[c] contains $i$ and $j$ then
      adj_matrix[i][j]++
  for $(i,j) \in \text{adj_matrix}$ do
    if adj_matrix[i][j] $\geq t/r$ then
      adj_matrix[i][j]/ = $t$
    else
      adj_matrix[i][j] = 0

// Classification
sc $\leftarrow \text{SpectralClustering}(\text{adj_matrix})$
return predict(sc,classes);

**Algorithm 2**: The HAPLEXD method. For simplicity, we consider the algorithm for a single gene $g$. The function *tractizeHaps* tractizes the haplotypes. The function *SubsetVariants* returns the indices of variants within a cis-window of $g$. The function *Ukkonens* constructs a tractized suffix tree of tractized haplotypes. *maxEntropy* computes the expression classes by the principle of maximum entropy and *predict* performs the classification.
Chapter 3

Experiments and Results

We evaluated HAPLEXR, PrediXcan (elastic net regression) [29], lasso regression [79], K-Nearest Neighbors (KNN) [56], and two logistic regression approaches modelled after lasso and PrediXcan on the Genotype-Tissue Expression (GTEx) project version 8 data (phs000424.v8.p2). For KNN, we used $K = 19$, which was the best $K$ on average found in a previous study [56]. We selected data from 5 of the GTEx tissues with high sample count (> 500): skeletal muscle, sun exposed skin, thyroid, lung, and whole blood. The cis-window around the TSS of each gene was set to $10^6$bp, a commonly used window in eQTL studies and in PrediXcan [29]. We normalize the expression $Y$ using the trimmed mean of M-values (TMM) method [68]. We then correct the expression by regressing out the covariates provided by GTEx (RNA-seq platform/protocol, PEER factors [77], and sex) and retaining the residuals. For testing, we held out 10% of the samples from each tissue, removed variants with MAF < 0.05, performed LD pruning with PLINK (plink --indep-pairwise 200 100 0.8), removed indels, and removed clusters with less than
5% of the training sample haplotypes [66, 32]. The continuous results include 15000 genes from the 5 tissues and, due to the increased number of haplotype clusters in the discrete case, the discrete results include 7500 genes from 5 tissues.

Because PrediXcan, lasso, HAPLEXR, and both configurations of logistic regression employ regularized regression, some of their fitted models have all-zero regression coefficients. For all subsequent comparisons between methods, we retain only the genes for which all compared methods produced nonzero models.

### 3.0.1 Continuous Expression

First, we selected the partition size in the haplotype clustering ($J$) by grid search on a random sample of 100 genes on chromosome 1 and $J = \{2^4, 2^5, 2^6, 2^7, 2^8\}$ (Supplementary Figure 5.1). We fit our model on each gene in the sample on 90% of the samples for whole blood, and measured the Pearson correlation between the true and inferred expression on the remaining 10%. We selected the haplotype partition size which yielded the highest median Pearson correlation ($J = 32$) for all further analysis.

Next, we computed the narrow-sense heritability between cis-SNPs and gene expression levels in whole blood using the genome-based restricted maximum likelihood (GREML) method in the genome-wide complex trait analysis (GCTA) software tool [91] (Figure 3.1). Narrow-sense heritability is the proportion of expression variation due to additive genetic variation and represents a theoretical upper bound on additive methods. In concordance with previous results, an increased $r^2$ was indicative of increased $h^2$ [29, 51]. All four methods appear to capture non-additive genetic components of expression variation, but
the proportion of genes where \( r^2 > h^2 \) was greatest in KNN (0.483) and HAPLEXR (0.335) compared to PrediXcan (0.292) and lasso (0.287); however, we note that most of the contribution of this statistic for KNN is for genes with low \( h^2 \) due to KNN producing a model for all genes. This behavior is exemplified by the abundance of predictions for low \( h^2 \) genes and comparatively fewer predictions above \( h^2 \) for highly heritable genes (Figure 3.1 top-left).

When we restricted the results to genes that all methods constructed models for, we observed similar predictive performance (mean \( r^2 \)) among HAPLEXR (0.0968), PrediXcan (0.0985), and lasso (0.0986), but comparatively poor performance from KNN (0.0455). To evaluate the relative performance, we compared the \( r^2 \) improvement pairwise for each method (Figure 3.2 and Supplementary Figures ??). HAPLEXR shows a considerable improvement on approximately 2/3 of the genes with respect to KNN and about half of the genes with respect to PrediXcan and lasso (Figure 3.2, top three plots). There is a large overlap between the subset of genes whose expression is well-predicted by PrediXcan, lasso, and HAPLEXR, but there is a significant subset of genes in each tissue for which HAPLEXR renders predictions above given \( r^2 \) thresholds (Supplementary Figure 5.6). KNN demonstrates this advantage mainly at a relatively low threshold of \( r^2 = 0.1 \). HAPLEXR is also the best-performing model for an average of about 37% of genes per tissue that were predicted by any model with \( r^2 > 0.1 \) (Table 3.1).

These findings suggest that (1) HAPLEXR captures some non-additive signal in a subset of the GTEx genes in each tissue, (2) HAPLEXR’s non-additive signals are generally of higher quality than KNN’s, but (3) dosage-only additive models are still preferable to haplotype clustering based models for a subset of genes. Further, we observed that lasso
Figure 3.1: **Comparing $r^2$ and narrow-sense heritability across continuous methods.** For each gene, an estimate of narrow-sense heritability ($h^2$) in blue, and regression $r^2$ on the test set in red. We compared $h^2$ and $r^2$ across gene expression in whole blood for (top-left) KNN, (top-right) PrediXcan, (bottom-left) lasso, and HAPLEXR (bottom-right).
and PrediXcan capture a similar additive signal with most genes having little difference in $r^2$ between the two methods (Figure 3.2, bottom).

HAPLEXR tends to select fewer features than PrediXcan and lasso (Supplementary Figure), making HAPLEXR more interpretable than both methods at high $|\beta|$ thresholds. Each significant haplotype feature that HAPLEXR selects represents a cluster of about 9 SNPs (Supplementary Table 5.1). Because we perform LD pruning before generating candidate haplotype clusters, this finding indicates that the haplotype clusters that HAPLEXR selects span multiple LD blocks.

<table>
<thead>
<tr>
<th>Tissue</th>
<th>HAPLEXR</th>
<th>PrediXcan</th>
<th>LASSO</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>blood</td>
<td>865</td>
<td>656</td>
<td>721</td>
<td>116</td>
</tr>
<tr>
<td>thyroid</td>
<td>1525</td>
<td>1072</td>
<td>1198</td>
<td>230</td>
</tr>
<tr>
<td>skin</td>
<td>914</td>
<td>706</td>
<td>694</td>
<td>150</td>
</tr>
<tr>
<td>muscle</td>
<td>946</td>
<td>674</td>
<td>712</td>
<td>92</td>
</tr>
<tr>
<td>lung</td>
<td>1035</td>
<td>788</td>
<td>881</td>
<td>245</td>
</tr>
</tbody>
</table>

Table 3.1: For each tissue and method, the number of genes best predicted by the method, with $r^2 > 0.1$.

**Variant Set Enrichment Analysis**

To characterize the regulatory function of variant and haplotype features used in our model, we performed a variant set enrichment (VSE) analysis on 76 annotations, predominately related to gene regulation (Supplementary Table 5.2) [3]. VSE compares the enrichment of an associated variant set across genomic annotations to null variant sets computed by a permutation procedure from reference GWAS tag SNP and 1000 Genomes Project data (Supplementary Information). We consider subsets of variants defined by thresholds
{0.1,0.2,0.3} on the $|\beta|$ coefficients of HAPLEXR and by feature type with respect to SNPs, haplotypes, and both (Figure 3.3, Supplementary Figures ??). Here, we focus on a $|\beta|$ coefficient threshold of 0.1 because, as we increased this threshold, the enriched variant set and overall enrichment decreased (Supplementary Figures ??).

We observed significant enrichment for regulatory annotations across all variant features, tissues, and within tissues (Figure 3.3). All UCSC gene region and ENCODE annotations were significantly enriched (VSE test, Bonferroni-corrected $p \leq 0.001$ and $p \leq 0.05$ respectively), which is consistent with the known cis-regulatory role of variation within transcription factor binding sites, intronic, and untranslated regions [4, 14, 40]. We also observed enrichment in enhancer and promoter regions, H3K9ac, H3K4me3, H3K4me1, and H3K27ac epigenetic modifications, and DNase I hypersensitive site. These findings recapitulate similar results for expression QTLs in GTEx v3 and v7 data [32, 31]. Interestingly, several enhancer annotations were not significant when considering only SNP variants, but when considering haplotypes variants or SNP and haplotype variants combined, raised to the level of significance (AncientEnhancer_e lung, Human_Enhancer_V SEC skin and lung); this result is supported by known haplotype specific enhancer effects, e.g., in human disease and *Drosophila* pigmentation [70, 30].

Our high $|\beta|$ variants were depleted in mammalian genomic regions conserved across taxonomic groups and in regions associated with background selection. This is likely due to these regions (a) not specifically being associated with genomic regulation and (b) being depleted of genetic variation due to negative selective pressures [59, 41]. The depletion of high $|\beta|$ SNP and haplotype features within repressed regions is consistent with the depletion of eQTLs in repressed annotations across cell lines and diseases [10, 63,
We also observed tissue specific significance patterns, including depletion of enhancer and H3K4me1 modifications in skin tissue; these results provide opportunities for future investigation.

### 3.0.2 Discrete Expression

We use discretized expression values computed by maximum entropy for $E = \{2,3,4\}$ in the training and evaluation of all discrete models. After training models on 90% of the data, we evaluated their performance for discrete expression prediction on the remaining 10% of the data based on classification accuracy and $F_1$ score. For continuous models PrediXcan, lasso, KNN, and HAPLEXR, we discretized their predictions based on the same partitions yielded by maximum entropy search.

We compared the discretized expression prediction of HAPLEXD to competing methods for the five tissues (Table 3.2). We find that for $E = 3$ and $E = 4$, HAPLEXD has statistically significantly higher classification accuracy as determined by paired one-tailed t-tests against each other method; in each test, we found that $p < 8.14 \times 10^{-8}$ (Figure 3.4). When considering the $F_1$ score for each method and tissue, the results are more mixed. As the discretization approaches the continuous limit, PrediXcan and lasso appear to outperform their logistic regression counterparts; we conjecture this is due to the regression methods being aware of the underlying ordering among the discrete expression classes. Interestingly, the performance of HAPLEXD relative to its continuous (and discrete) competitors increases with $E$ despite the method not explicitly modelling the ordering of expression classes.
<table>
<thead>
<tr>
<th>$E$</th>
<th>Tissue</th>
<th>HD</th>
<th>LR-EN</th>
<th>LR-L</th>
<th>HR</th>
<th>PX</th>
<th>LASSO</th>
<th>KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>blood</td>
<td>0.498</td>
<td>0.576</td>
<td>0.575</td>
<td>0.574</td>
<td><strong>0.579</strong></td>
<td><strong>0.579</strong></td>
<td>0.536</td>
</tr>
<tr>
<td>2</td>
<td>thyroid</td>
<td><strong>0.606</strong></td>
<td>0.594</td>
<td>0.594</td>
<td>0.598</td>
<td>0.602</td>
<td>0.602</td>
<td>0.553</td>
</tr>
<tr>
<td></td>
<td>skin</td>
<td>0.539</td>
<td>0.579</td>
<td>0.578</td>
<td>0.582</td>
<td><strong>0.585</strong></td>
<td>0.584</td>
<td>0.546</td>
</tr>
<tr>
<td></td>
<td>muscle</td>
<td>0.562</td>
<td>0.574</td>
<td>0.576</td>
<td>0.577</td>
<td><strong>0.580</strong></td>
<td><strong>0.580</strong></td>
<td>0.542</td>
</tr>
<tr>
<td></td>
<td>lung</td>
<td>0.575</td>
<td>0.578</td>
<td>0.580</td>
<td>0.579</td>
<td><strong>0.584</strong></td>
<td>0.583</td>
<td>0.539</td>
</tr>
<tr>
<td></td>
<td>all</td>
<td>0.559</td>
<td>0.581</td>
<td>0.581</td>
<td>0.583</td>
<td><strong>0.587</strong></td>
<td>0.586</td>
<td>0.544</td>
</tr>
<tr>
<td></td>
<td>blood</td>
<td>0.378</td>
<td>0.393</td>
<td>0.392</td>
<td>0.365</td>
<td>0.392</td>
<td><strong>0.394</strong></td>
<td>0.347</td>
</tr>
<tr>
<td>3</td>
<td>thyroid</td>
<td><strong>0.430</strong></td>
<td>0.411</td>
<td>0.411</td>
<td>0.390</td>
<td>0.422</td>
<td>0.422</td>
<td>0.362</td>
</tr>
<tr>
<td></td>
<td>skin</td>
<td><strong>0.439</strong></td>
<td>0.396</td>
<td>0.397</td>
<td>0.375</td>
<td>0.407</td>
<td>0.407</td>
<td>0.354</td>
</tr>
<tr>
<td></td>
<td>muscle</td>
<td>0.359</td>
<td>0.392</td>
<td>0.391</td>
<td>0.369</td>
<td><strong>0.396</strong></td>
<td><strong>0.396</strong></td>
<td>0.351</td>
</tr>
<tr>
<td></td>
<td>lung</td>
<td><strong>0.456</strong></td>
<td>0.393</td>
<td>0.393</td>
<td>0.372</td>
<td>0.403</td>
<td>0.403</td>
<td>0.351</td>
</tr>
<tr>
<td></td>
<td>all</td>
<td><strong>0.413</strong></td>
<td>0.398</td>
<td>0.398</td>
<td>0.375</td>
<td>0.405</td>
<td>0.405</td>
<td>0.353</td>
</tr>
<tr>
<td></td>
<td>blood</td>
<td>0.334</td>
<td>0.294</td>
<td>0.293</td>
<td>0.281</td>
<td>0.304</td>
<td>0.304</td>
<td>0.260</td>
</tr>
<tr>
<td>4</td>
<td>thyroid</td>
<td><strong>0.392</strong></td>
<td>0.312</td>
<td>0.312</td>
<td>0.301</td>
<td>0.331</td>
<td>0.330</td>
<td>0.271</td>
</tr>
<tr>
<td></td>
<td>skin</td>
<td><strong>0.348</strong></td>
<td>0.297</td>
<td>0.297</td>
<td>0.286</td>
<td>0.314</td>
<td>0.314</td>
<td>0.265</td>
</tr>
<tr>
<td></td>
<td>muscle</td>
<td>0.302</td>
<td>0.294</td>
<td>0.293</td>
<td>0.281</td>
<td><strong>0.304</strong></td>
<td><strong>0.304</strong></td>
<td>0.264</td>
</tr>
<tr>
<td></td>
<td>lung</td>
<td><strong>0.344</strong></td>
<td>0.292</td>
<td>0.293</td>
<td>0.283</td>
<td>0.310</td>
<td>0.311</td>
<td>0.261</td>
</tr>
<tr>
<td></td>
<td>all</td>
<td><strong>0.346</strong></td>
<td>0.298</td>
<td>0.298</td>
<td>0.287</td>
<td>0.313</td>
<td>0.314</td>
<td>0.264</td>
</tr>
</tbody>
</table>

Table 3.2: Micro-averaged $F_1$ score for each method and tissue, and across all tissues, with $E \in \{2,3,4\}$ (rounded to three significant figures). Bolded entries denote the highest $F_1$ score for a tissue. HD is HAPLEXD, HR is HAPLEXR, PX is PrediXcan, and LR-EN and LR-L are logistic regression with elastic net and lasso regularization respectively.
Figure 3.2: **Gene-wise improvement of $r^2$.** The improvement in $r^2$ between HAPLEXR compared to KNN (top), PrediXcan (top-middle), and lasso (middle-bottom) sorted by improvement per gene. Bottom: improvement in $r^2$ between PrediXcan and lasso sorted by improvement per gene.
Figure 3.3: Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.1. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected). Variant set naming convention indicates the tissue and type of variant (s, h, and b, denoting SNP, haplotype, and both respectively).
Figure 3.4: For each method and $E \in \{2, 3, 4\}$, the distribution of per-gene expression classification accuracy over all tissues. Paired one-tailed t-tests of HAPLEXD classification accuracy with each other method for $E = 3$ and $E = 4$ all have $p < 8.14 \times 10^{-8}$. 
Chapter 4

Conclusion and Discussion

In this work, we introduced the problem of gene expression classification and presented two methods, HAPLEXR and HAPLEXD, for predicting continuous and discretized gene expression from haplotypes. HAPLEXR and HAPLEXD consider haplotype sharing that encodes non-linear effects between variants. We evaluated both methods on GTEx experimental data across 5 tissues and demonstrated that our methods capture a haplotype signal not effectively modelled by the linear and additive variant dosage approaches. We develop two additional linear models in the discrete case, and show clear performance gains. In the continuous case, our methods perform similarly to lasso and PrediXcan when aggregated across tissues, but deeper analysis on well-predicted genes shows that HAPLEXR is complementary to the linear and additive models, capturing a distinct signal. Finally, we demonstrated that our methods capture biologically meaningful patterns supported by eQTL studies. Our results show that both methods capture epistatic interactions that are not characterized by purely additive linear models, but, are complementary to additive
and linear dosage models as they capture distinct signatures.

There are several opportunities to expand on the methods and results presented here. HAPLEXR and HAPLEXD make the additional assumption that we have access to phased haplotype data, which can be difficult to experimentally derive or computationally infer [11, 53]. Additionally, there are many choices for haplotype clustering model and an increased computational burden of computing the clustering. In the continuous case, we presented a computationally simple model that lacks robustness to rare variants, errors in haplotyping, or varying LD. Due to the computational burden of generating the clustering, we inferred parameters via cross-validation on a held out sample, but it is likely that individual genes have unique regulatory architectures. In this case, a cross-validation procedure per gene would likely yield more accurate models [56].

We computed the proportion of haplotype cluster features among all HAPLEXR gene models. For every subset of genes whose proportion of haplotype features are greater than those defined by the thresholds \( \left\{ \frac{1}{10}, \frac{2}{10}, \frac{3}{10}, \ldots, \frac{9}{10} \right\} \), the median improvement in \( r^2 \) of HAPLEXR relative to PrediXcan and lasso across all tissues was zero. However, there are two distinct sets of genes: one where the linear models have significantly better performance and another where HAPLEXR produced the most accurate model. This suggests that the linear and additive assumptions more accurately model the regulatory architecture of the former, whereas the combination of SNP and haplotype features more accurately model the latter. We conjecture that to see a consistent advantage in \( r^2 \) with respect to the proportion of haplotype features, a genetic clustering model that is more robust to varying LD, rare variation, and haplotyping errors is required.
An underlying assumption of all models that predict gene expression from genetic data is that the genetic markers act as a proxy for intermediate molecular phenotypes that influence expression. These include histone modifications, chromatin accessibility, and DNA methylation. New studies like the Enhancing GTEx project aim to characterize genetic and intermediate molecular phenotypes in multiple tissues per sample [23]. Open problems and future work in expression prediction include (a) determining how to combine these regulatory markers with genetic models, (b) incorporating other genes, pathways, and trans-eQTLs in expression prediction, and (c) simultaneous modelling of correlated tissues or conditions.

Finally, we note that HAPLEXR is distinct from, but shares similarities with, the group lasso defined on dosages and genetic model clusters [92]. Group lasso introduces an $\ell^2$ penalty on groups of covariates, preferentially forcing all covariates in a group to be 0 or nonzero. HAPLEXR considers the haplotype clusters themselves to be covariates. The sparse group lasso is a convex combination of lasso and group lasso penalties, and while more difficult to fit, is a more comparable statistical model to HAPLEXR and subject of future work [65, 76].
Chapter 5

Supplementary Information

5.0.1 Fast online suffix tree construction with tractized haplotypes.

Here we show that Ukkonen’s suffix tree algorithm can be used to efficiently construct an online tractized suffix tree in linear time.

**Lemma 2** Given an input tractized string matrix of size $2^n p$, the number of nodes in the tractized suffix tree is $< 2^n$ for $n \geq 3$.

A new vertex is created in the tractized suffix tree for each unique suffix, and so the number of nodes in the tractized suffix tree is $O(np)$. An upper bound on the number of unique suffixes is the set of binary representations of integers. Consider the case where the input is the binary representation of integers from 0 to $2^n$. Insertion of the first index over
all input will create a single root node and two leaves; during insertion of the second index, each leaf node becomes an internal node and gains two children. Generalizing, all leaf nodes at index \( i \) become internal nodes at iteration \( i + 1 \) because the addition of another character index generates a new unique sequence. Another way to recognize this is, the tractized suffix tree encodes two full binary trees (one for each allele) from the root of sizes \( 2^i - 1 \) for \( i = 1, \ldots, p \). The number of nodes in the tractized suffix tree is then,

\[
1 + \sum_{i=1}^{p} 2^{i+1} - 2 = 2^{p+2} - 2p - 3 \tag{5.1}
\]

If the size of the input is \( 2^p n \), then the number of nodes in the tractized suffix tree is less than the size of the input if \( 2^{p+2} - 2p - 3 < 2^p m \Rightarrow n > 2.7 \), which is true for all non-trivial instances.

To show that we can construct an online tractized suffix tree for \( n \) tractized haplotypes each of length \( p \) in time \( O(np) \) we consider tractized strings and Ukkonen’s algorithm. Ukkonen’s algorithm for online suffix tree construction proceeds by constructing a series of implicit suffix trees. Consider the case for concatenated haplotype strings

\[
h_0, \ldots, h_{n-1}, \]

where \( h_i \) is the \( i \)th haplotype’s terminating character appended to the end of \( h_i \). An implicit tractized suffix tree is a tractized suffix tree with all terminating symbols \( \$, \ldots, \$ \) removed and all edges with empty labels as a result removed as well. Denote a substring of tractized haplotype \( i \) at positions \( j \) through \( k \) as \( h_i[j \ldots k] \). An implicit tractized suffix tree for prefixes \( \{h_i[1 \ldots j]\}_i=1^n \) is denoted \( T_j \). Ukkonen’s algorithm constructs a series of implicit suffix trees \( T_1 \) to \( T_p \) and then converts \( T_p \) into a tractized suffix tree.
Given tractized haplotype strings, we show that the three time saving rules of Ukkonen’s suffix tree algorithm apply.

1. If a tractized haplotype mismatches with an existing edge at the position $j$, then it will not merge with any tractized haplotypes from positions $j + 1, \ldots, n$. We no longer need to update these nodes if we simply reference their end with the index of the current iteration implicit suffix tree.

2. If we represent the edges as a starting and ending index instead of a string, we can compress all edges to constant size. This makes it trivial to identify the next node to traverse on any path using the starting index and an ending index (which we can access in constant time) and recognizing that there is a constant number of outgoing edges from any internal node. This also enables the extension of all leaves in the tractized suffix tree in constant time by having them refer to a current end index which we increment each iteration.

3. For a particular tractized haplotype, if we encounter an extension of a prefix in Ukkonen’s algorithm that does not require generation of a new node (it is already captured by the implicit suffix tree) then that extension applies to all following prefixes for that tractized haplotype.

The active point remains simple to specify as the maximum number of possible children for any node (besides the root) is a small constant (2 for biallelic haplotypes) and we can use a hash table to lookup edges adjacent to the root. We can guarantee constant time hashing because the maximum number of edges adjacent to the root is $2p$ so we can set an
appropriate capacity to ensure low collisions and no resizing and we are hashing integers of size at most $2p$ so the hashing function is fast.

5.0.2 Discretizing Expression Values by Entropy Maximization

Given a sequence of expression values, the goal is to cluster the sequence into $E$ groups. We compute the $E$-clustering and the underlying probability distribution that has the maximum entropy.

The method of entropy maximization, better named, information entropy maximization was formalized by E. T. Jaynes in his seminal book [43]. Information entropy (Shannon entropy) is a better name as it distinguishes it from the “experimental entropy” of statistical mechanics; information entropy is a property of probability distributions, while the experimental entropy of statistical mechanics is a property for the thermodynamics of some physical system. Information entropy maximization is a fundamental method for inferring rigorous priors (of “maximum ignorance”) for a given probability distribution.

Let us consider a set $\sigma = \{a_1, \ldots, a_E\}$ with $E$ different numerical values. Suppose that we observe a sequence of $n$ numbers from $\sigma$ and their average $a$. Suppose now that we are given only the average value $a$ of the above sequence of $n$ numbers (but not the sequence) and also suppose that we know $\sigma$. The intuition about the information entropy maximization method is as follows: “The knowledge of average values does give the robot a reason for preferring some possibilities to others, but we would like it to assign a probability distribution which is as uniform as it can be while agreeing with the available information. The most conservative, non-committal distribution is the one which is as ‘spread-out’ as
possible... the information available defines constraints fixing some properties of the initial probability distribution, but not all of them. The ambiguity remaining is to be resolved by the policy of honesty; frankly acknowledging the full extent of its ignorance by taking into account all possibilities allowed by its knowledge” [43].

For our clustering problem, we are given as input a sequence of $n$ numerical expression values and the number of clusters $E$. Let $a$ be the average expression value. We want to compute, based on general principles (maximum ignorance), the most general partition into $E$ clusters of the input sequence based only on the information given by $n,a,E,\sigma$, where $\sigma$ is the set of $E$-averages $\sigma = \{a_1, a_2, ..., a_k\}$ for the $E$ clusters. Note that for any $E$ there are several such feasible $\sigma$s (different $E$ clusters will give different $\sigma$s). We can reformulate this in terms of a random variable $W$ with an underlying a probability distribution. We think of each cluster as follows: if its average is $a_i$ and it has $n_i$ elements, then we view it as a multi-set with $n_i$ elements all equal to $a_i$ (each expression value in the cluster is replaced by the cluster average). In this way, the random variable has the outcomes set $\sigma$, and the probability distribution that the $E$-cluster decomposition defines. We would like to compute the $E$-cluster decomposition that maximizes the information entropy, that is, among all such $E$-cluster probability distributions, to find the one that has the maximum information entropy.

Formally, the maximum information entropy principle is defined as follows. Suppose a random variable $A$ can take on $E$ different discrete values $(a_1, ..., a_n)$ which correspond to $E$ different propositions $\pi_1, ..., \pi_E$; and that there are $m$ different functions of $A$ (representing constraints), $\{f_k(A), 1 \leq k \leq m < n\}$ and we want them to have expectations equal to constants i.e.,
\[ F_k = \langle f_k(A) \rangle = \sum_{i=1}^{E} p_i f_k(a_i), \quad 1 \leq k \leq m \]

where \( F_k \) are numbers given to us in the statement of the problem.

What probabilities \((p_1, \ldots, p_n)\) could be assigned to the possibilities \((a_1, \ldots a_n)\)? We will find the set of \( p_i \)'s which has the maximum information entropy subject to these constraints, together with the standard constraint that the sum of the probabilities is equal to 1. The solution of this maximization is based on Lagrangian multipliers. If \( \lambda_0 \) is the Lagrangean multiplier for “the sum of probabilities add up to 1”, and \( \lambda_j, 1 \leq j \leq m \) are the Lagrangean multipliers for the averages constraints functions \( f_j, 1 \leq j \leq m \), then the solution to the problem, namely the probability distribution of maximum entropy is given by

\[ p_i = \exp(-\lambda_0 - \sum_{j=1}^{m} \lambda_j f_j(x_j)) \]

\[ Z(\lambda_1, \ldots, \lambda_m) = \sum_{i=1}^{E} \exp(\sum_{j=1}^{m} \lambda_j f_j(a_i)) \]

\[ \lambda_0 = \log Z(\lambda_1, \ldots, \lambda_m) \]

\[ H_{max} = \left( \sum_{i=1}^{E} p_i \log(p_i) \right)_{max} = \lambda_0 + \sum_{j=1}^{m} \lambda_j F_j. \]

The term \( Z(\lambda_1, \ldots, \lambda_m) \) is called the partition function, and \( H_{max} \) is the value of the
maximum information entropy.

The principle of maximum information entropy states that the probability distribution for $A$ which best represents the current state of knowledge is the one with the largest Shannon entropy. For example, if the only prior information is that the average $a$ of the outcomes (only one constraint function $f_1(m = 1)$), then, the maximum entropy distribution is the celebrated Gibbs-Boltzmann distribution of statistical mechanics. This is the Boltzmann theorem, proved by the method of Lagrange multipliers. It is interesting to note a continuous-discrete mathematics difficulty that was resolved by seminal research. It turns out that the method of Maximum Entropy was used first by Gibbs for continuous probability distributions for canonical ensemble in classical statistical mechanics, but the continuous method solution failed to predict accurately important thermodynamic properties (e.g. heat capacity). It was Planck, Einstein, and then Bohr who showed how to fix the results. The need was a special discretization of the values, and so a discrete probability distribution was the solution; the discrete possible energy values were henceforth called energy levels. In 1927 the theory was developed from first principles by Heisenberg and Schrödinger.

### 5.0.3 Variant Set Enrichment Analysis

For our variant set enrichment analysis [3], we computed the enrichment of an associated variant set across 76 annotations (Supplementary Table 5.2). First, we lifted the annotations to human genome version hg38 to match the GTEx v8 data using UCSC liftOver [35]. Variant feature sets were defined based on their $\beta$ coefficients and type
(SNP variants, haplotypes, or both) in the HAPLEXR model. Variant set enrichment analysis is defined on tag SNP blocks and sensitive to linkage disequilibrium (LD). Thus, we computed the SNPs in the v8 GTEx data that tag \( r^2 \geq 0.8 \) the method input variants using plink (plink --show-tags <exp_prediction_input_variants> --tag-kb 200 --list-all --vcf <all_GTEx_variants>).

Next, we construct associated variant sets (AVS) for each tissue, across all tissues, and for SNP, haplotype, and a combination of both variant feature types. AVS compress a set of linked variants into a lower dimensional set of tagging variants. Using the set of LD-corrected tagging variants, VSE generates AVS sampled from the null hypothesis (no enrichment) by computing the overlap of randomly permuted AVS with genomic annotations. The enrichment score is a statistic computed as the number of standard deviations that the count of overlapping regions from the AVS deviates from the randomly permuted AVS count median. An exact p-value is computed by fitting a density to the randomly permuted AVS null and a Kolmogorov-Smirnov test is used to check for normality of the null distribution. If deviation of the null is detected, a Box-Cox power transformation is used to enforce normality. In this work, the number of random permutations to generate the null hypothesis was 100 for tissues and increased to 250 for trans-tissue runs due to the increase in variant features.

**Relevant References / Background:**

[54, 89] for interpretation of genetic variation associations with respect to eQTLs.

[86, 13, 20] for explaining the gap between association and predictability and varying genetic architectures.
5.1 Supplementary Figures

Figure 5.1: **Selection of partition size.** Pearson correlation between actual and predicted whole blood expression on 100 randomly selected genes on chromosome 1 across settings of the haplotype partition size ($J$) used in the haplotype clustering algorithm. Genes whose associated model had no nonzero regression coefficients are excluded from the plot; the number of these genes per setting are 33, 29, 33, 30, and 30 respectively. The center line denotes the median, the box ends the first and third quartiles, and the whiskers extend $1.5 \times IQR$. 


Figure 5.2: **Gene-wise $r^2$ improvements (KNN).** Gene-wise improvement of $r^2$ between KNN-predicted and actual continuous expression compared to PrediXcan (top), lasso (middle), and HAPLEXR (bottom).
Figure 5.3: **Gene-wise $r^2$ improvements (PrediXcan)**. Gene-wise improvement of $r^2$ between PrediXcan-predicted and actual continuous expression compared to KNN (top), lasso (middle), and HAPLEXR (bottom).
Figure 5.4: Gene-wise $r^2$ improvements (lasso). Gene-wise improvement of $r^2$ between lasso-predicted and actual continuous expression compared to KNN (top), PrediXcan (middle), and HAPLEXR (bottom).
Figure 5.5: **Average features used per gene.** Across all tissues and for each continuous gene expression prediction method that uses linear regression, the average number of features per gene whose corresponding regression coefficients have magnitude above a given threshold. HAPLEXR’s vector-encoded haplotype memberships at one range of the genome are considered a single feature.
Figure 5.6: **Venn diagram comparisons between sets of well-predicted genes.** Each Venn diagram describes the intersections between continuous gene expression prediction methods of genes predicted above a given $r^2$ in a given tissue. PX is PrediXcan and HR is HAPLEXR.
Figure 5.7: Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.3. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).
Figure 5.8: Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.2. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).
Figure 5.9: Enrichment scores from variant set enrichment analysis across all tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.1. Points in red denote a significant enrichment (VSE test, Bonferroni-corrected $p \leq 0.05$).
Figure 5.10: Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.1. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).
Figure 5.11: Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.2. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).
Figure 5.12: Heatmap for the significance of enrichment across tissues and annotations (Supplementary Table 5.2) for a $|\beta|$ threshold of 0.3. Cell color denotes the level of significance for a particular variant set and annotation (VSE test, Bonferroni-corrected. Variant set naming convention indicates the tissue and type of variant (s, h, and b denoting SNP, haplotype, and both respectively).
5.2 Supplementary Tables

| Tissue | $0.1 < |\beta| \leq 0.2$ | $0.2 < |\beta| \leq 0.3$ | $|\beta| > 0.3$ |
|--------|-----------------|-----------------|-----------------|
| blood  | 9.1             | 8.7             | 8.9             |
| thyroid| 9.1             | 9.3             | 9.9             |
| skin   | 9.1             | 9.0             | 9.4             |
| muscle | 9.2             | 9.4             | 9.9             |
| lung   | 8.9             | 8.8             | 9.1             |
| all    | 9.1             | 9.1             | 9.5             |

Table 5.1: Tag SNP counts across clusters. For each tissue and across all tissues, the average number of tag SNPs in each HAPLEXR haplotype cluster with regression coefficients in three intervals (columns).
Table 5.2: Regulatory annotations used in the variant set enrichment analysis.
All annotations were retrieved from [28] and [41].

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding_UCSC</td>
<td>Coding, 3’ UTR, 5’ UTR, intronic, and promoter regions from RefSeq gene models in the UCSC Genome Browser [46]. These annotations were processed by [33].</td>
</tr>
<tr>
<td>Intron_UCSC</td>
<td></td>
</tr>
<tr>
<td>Promoter_UCSC</td>
<td></td>
</tr>
<tr>
<td>UTR_3_UCSC</td>
<td></td>
</tr>
<tr>
<td>UTR_5_UCSC</td>
<td></td>
</tr>
<tr>
<td>DGF_ENCODE</td>
<td>ENCODE [18] annotations for digital genomic footprints and transcription factor binding sites that were processed by [33].</td>
</tr>
<tr>
<td>TFBS_ENCODE</td>
<td></td>
</tr>
<tr>
<td>CTCF_Hoff</td>
<td>[38] developed a rule-based metric to classify results from the Segway [37] and ChromHMM [25] methods on six cell types into seven regulatory region types: transcriptional repressor CTCF, enhancer, promoter flanking, transcribed, transcription start sites, weak enhancer, and inactive or repressed.</td>
</tr>
<tr>
<td>Enhancer_Hoff</td>
<td></td>
</tr>
<tr>
<td>PromoterFlk_Hoff</td>
<td></td>
</tr>
<tr>
<td>Transcribed_Hoff</td>
<td></td>
</tr>
<tr>
<td>TSS_Hoff</td>
<td></td>
</tr>
<tr>
<td>WeakEnhancer_Hoff</td>
<td></td>
</tr>
<tr>
<td>Repressed_Hoff</td>
<td></td>
</tr>
<tr>
<td>DHS_Try</td>
<td></td>
</tr>
<tr>
<td>FetalDHS_Try</td>
<td>[81] processed ENCODE [18] and Roadmap Epigenomics [49] project data across many cell types to produce longer regions and peaks of DNase I hypersensitivity, and H3K4me1, H3K4me3, and H3K9ac epigenetic modifications.  [28] partitioned the DHS regions into fetal cell types (FetalDHS) and all cell-types (DHS). The epigenetic modifications were also combined across all cell types by [28].</td>
</tr>
<tr>
<td>H3K4me1_Try</td>
<td></td>
</tr>
<tr>
<td>H3K4me3_Try</td>
<td></td>
</tr>
<tr>
<td>H3K9ac_Try</td>
<td></td>
</tr>
<tr>
<td>DHS_peaks_Try</td>
<td></td>
</tr>
<tr>
<td>H3K4me1_peaks_Try</td>
<td></td>
</tr>
<tr>
<td>H3K4me3_peaks_Try</td>
<td></td>
</tr>
<tr>
<td>H3K9ac_peaks_Try</td>
<td></td>
</tr>
<tr>
<td>H3K27ac_PGC2</td>
<td>H3K27ac methylation marks generated by Roadmap Epigenomics [49] and processed by [67] (PGC2) and [36] (Hnisz).</td>
</tr>
<tr>
<td>H3K27ac_Hnisz</td>
<td>Super-enhancer regions generated by [36].</td>
</tr>
<tr>
<td>SuperEnhancer_Hnisz</td>
<td>Enhancers from the FANTOM 5 project [5].</td>
</tr>
<tr>
<td>Enhancer_Ander</td>
<td>Genomic regions in mammals from [52], and from phastCons [75] elements (46 way) in the UCSC Genome Browser [46] conserved in mammals, primates, and vertebrates (Vert) [88].</td>
</tr>
<tr>
<td>Cons_Lind</td>
<td></td>
</tr>
<tr>
<td>Cons_Mammal_phast</td>
<td></td>
</tr>
<tr>
<td>Cons_Primat_e_phast</td>
<td></td>
</tr>
<tr>
<td>Cons_Vert_phast</td>
<td></td>
</tr>
<tr>
<td>Human_Enhancer_V</td>
<td>Enhances and promoters, including loss-of-function genes from the Exome Aggregation Consortium (ExAC) [50] and enhancer-enhancer conservation count (SEC) [83].</td>
</tr>
<tr>
<td>Human_Enhancer_V_SEC</td>
<td></td>
</tr>
<tr>
<td>Human_Promoter_V_ExAC</td>
<td></td>
</tr>
<tr>
<td>Human_Promoter_V</td>
<td></td>
</tr>
<tr>
<td>AncientEnhancer</td>
<td>Ancient enhancers and promoters from [41].</td>
</tr>
<tr>
<td>AncientPromoter</td>
<td></td>
</tr>
<tr>
<td>BP_DNAMeth_MaxCPP</td>
<td>Regulatory annotations that are enriched for disease heritability [39].</td>
</tr>
<tr>
<td>BP_H3K27ac_MaxCPP</td>
<td></td>
</tr>
<tr>
<td>BP_H3K4me1_MaxCPP</td>
<td></td>
</tr>
<tr>
<td>GTEx_MaxCPP</td>
<td></td>
</tr>
<tr>
<td>BivFlnk</td>
<td>Flanking Bivalent TSS and enhancers from Roadmap Epigenomics [49].</td>
</tr>
<tr>
<td>Bckgrd_Selection_Stat</td>
<td>Regions of common SNPs undergoing background selection (1-McWicker B statistic) [41, 59].</td>
</tr>
<tr>
<td>X_e</td>
<td>Same annotation as X but with 500bp padding.</td>
</tr>
</tbody>
</table>
References


