A Haplotype-Based Predictive Model for Genotype/Expression Datasets

Sudheesha Perera
Department of Computer Science, Brown University

Submission for APMA-Computer Science Concentration
Adviser: Prof. Sorin Istrail
Spring 2017

Abstract

Genome/expression datasets comprise DNA sequencing data and associated gene expression data for a set of individuals. This thesis project develops a novel predictive model for imputing expression from the binary, strand specific signals contained in phased sequencing data (ie. haplotypes). The method is generalized to accept any phenotype data that has quantized expression; here we explore application to RNA-seq data which is currently prevalent in the field. We first construct a modified suffix tree to capture the substructure of input sequences, and then apply a voting theory-inspired algorithm for expression level prediction. The following work includes relevant biological background, a formalization of the expression prediction problem, and a description of the model along with its motivations. We conclude with the results of running the implemented model on simulated data as well as a limited trial of real human genome data.

Contents

1 Introduction

This thesis project aims to develop a novel predictive model for genome/expression datasets. Such datasets comprise both DNA sequencing data and associated gene expression data for individuals in a population, allowing us to investigate the relationship between genetic variation and gene expression. Diverging from current literature on the subject, we chose to investigate the problem from the standpoint of phased haplotypes rather than the unphased form of genomic data that is more often considered. Our motivation here is two-fold: not only do haplotypes permit a simple binary representation, but it is also our hypothesis that their strand-specific, binary signals will allow us to better predict expression across a population. We begin with expository material related to the nature of relevant biological data, followed by a generalized formulation of the prediction problem and a brief review of current literature. Next we introduce a modified suffix tree construction and associated voting-theory algorithm that together constitute the predictive model. Finally we present the results of running the implemented model on simulated data and a limited trial of real human genome and expression data.

2 Modelling Genetic Data

With the advent of whole-genome sequencing, the volume of genetic data available for analysis exploded. A wealth of genetic datasets now exist, including both public-access repositories, such as the the 1000 Genomes Project, and restricted-access datasets curated by the National Institutes of Health and various academic institutions. From a computational standpoint these datasets present unique challenges. Not only are they exceedingly complex, but their size alone means that efficient methods of analysis must be employed to achieve tractability. This section provides some background on the origin of genetic data utilized in this work, with specific focus on aspects that are relevant to the predictive model developed later.

2.1 Biology Background

Genotype data used in this project is derived from the process of DNA sequencing, whereby wet lab work is performed to determine the sequence of `A`, `T`, `C`, and `G` nucleotide bases that compose an individual's genome. The entire sequence is on the order of 3 billion base pairs, but most of the code does not get translated into molecular products. The short and sparse regions that do encode the directions for producing biological components are known as *genes*. Since mutations in these regions result in altered biological outcomes, humans have evolved to have highly conserved gene sequences. The sites in which variation are observed are known as *single nucleotide polymorphism* loci, or simply *SNPs*. Due to the enormity of the entire genome and the timescale of the human race's history, biologists work under the assumption that only two variants exist at a given SNP across the population.

In genome/expression datasets, the genomic data consists of overlapping fragments of DNA that can be assembled into the entire sequence of the genome. Through assembling the genomes of many individuals we are able to get a sense of which loci are SNPs. Since our objective is to model the variation across individuals in a population and use these signals to predict expression, SNPs are of chief importance. They are the fundamental unit of variation and the only loci worth incorporating into a predictive model.

Just as there is variation in gene sequences across a population, there is also variation in how these genes are expressed. *Expression* refers to how much of the gene's product is being produced at a given time. Biologists use a process called RNA sequencing (RNA-seq) to quantify expression, with the expectation that increasing quantities of RNA imply that the associated gene is being expressed at increased levels. Many genome/expression datasets contain quantified expression for every gene of each individual in units of reads per Kilobase of transcript per million mapped reads (RPKM). For the purpose of this project we will be using RPKM to quantify gene expression.

2.2 Haplotype Phasing

In discussing the genome thus far, we have ignored that it is made up of 23 pairs of analogous chromosomes. Each chromosome is a contiguous strand of DNA with a unique sequence, with the two strands of each pair being largely identical. When DNA fragments were first assembled into the human genome, the slight genetic differences between the two paired strands of each chromosome were ignored for simplicity. Since then, new methods of analysis have been developed to capture the strand-specific variation that existed in raw sequencing data. These methods are broadly know as techniques for *haplotype phasing*. We will focus on analyzing the results of such techniques, known as haplotypes, with the hypothesis that their strand-specific, binary signals will allow us to better predict expression across a population of individuals.

3 The Expression Prediction Problem

The problem of predicting expression from genotype data is not entirely unique. In the field of machine learning this task falls into a class known as small-n large-p problems, denoting that the number of samples is relatively low compared to the dimensionality of each sample. In this section we aim to abstract away the irrelevant aspects of the data, and represent the expression prediction problem in its simplest form. This provides greater clarity and allow us to apply intuition from existing learning approaches to our domain.

3.1 Generalized Formalization

Once phased, the haplotypes can be represented by a $2 \times N$ binary matrix where the rows represent the two chromosomes of the pair and the columns represent SNPs in consideration. Ideally we would like to translate the information captured by the phasing process into a standard $1 \times N$ vector form. This rationale is common in machine learning; many approaches require input data in the form of vectors, with each position in the vector representing a feature. To this effect we chose to model a haplotype as a binary vector, where the complementarity of its structure is captured by picking a common orientation across all phased individuals. For example, consider a pair of haplotypes for individual I. It is of the form:

$$\begin{array}{c|ccccc|} & s_1 & s_2 & s_3 & \dots & s_N \\ h_1 & 1 & 0 & 1 & \dots & 1 \\ h_2 & 0 & 1 & 0 & \dots & 0 \end{array}$$

...where each row represents a chromosome strand and each column is a SNP. We then define the convention that we always select the haplotype that begins with a '1', or in other words the strand that witnesses the minor allele at the first SNP position s_1. One of the strands must meet this criterion, since the two haplotypes are complementary at every locus by construction. If we later extend this model to include homogzygous loci, we can choose the haplotype based on the orientation of the first heterozygous locus. After this selection process, each individual is represented by a binary vector representing the strand-specific variation along one of their chromosomes. Now we can aggregate the variation across the population by placing these vectors in a matrix and pairing the matrix with the observed expression for a given gene.

$$\begin{array}{cc} & \begin{array}{ccccc} s_1 & s_2 & s_3 & \cdots & s_K \end{array} \\ \begin{array}{c} I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_N \end{array} & \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \end{array} \qquad \begin{array}{cc} & g_m \\ \begin{array}{c} I_1 \\ I_2 \\ I_3 \\ \vdots \\ I_N \end{array} & \begin{bmatrix} 1.1 \\ 2.3 \\ 0.1 \\ \vdots \\ 3.3 \end{bmatrix} \end{array}$$

On the left we have N individuals each represented by a row of the matrix, and K SNPs each represented by a column of the matrix. On the right we consider an arbitrary gene of interest, denoted by g_m. The values in the g_m vector are RPKM values representing expression of the m-th gene in the dataset for each individual. At this point the proposed question is: "Can we find a subset of the K SNP loci that correlates well with expression of g_m?". In narrower terms, we are searching for some strand-specific set of patterns that will allow us to impute an individual's expression of g_m given their phased haplotype.

From combinatorics we know there are 2^K subsets of the K loci that could be included in this strand-specific pattern. Given that K is typically much larger than N (ie. there are many more SNPs in the genome than individuals in the dataset), this is far too many loci to consider. We refine the loci of interest as follows:

- Only consider SNPs that are local to gene g_m. Instead of considering all K loci when searching for the pattern, we only focus on those that are "cis" or "in the vicinity" of g_m. Following the protocol of the Gamazon et al., this means choosing SNPs that are on the same chromosome as g_m and exist either in the gene region within 1 MB of its start or end [4]. As their 2015 paper notes, using estimates derived from all genotyped SNPs results in too much noise to make meaningful inferences. Additionally, from a statistical standpoint the multiple testing burden of considering all loci is insurmountable given the sample sizes afforded by current datasets.

- Use the concept of linkage disequilibrium (LD) to remove highly correlated loci [5]. This may improve our ability to impute expression, or at the very least simplify the complexity of the resulting predictive model. It also functions as a data-compression step if we find that many loci in the gene region are highly correlated.

- Re-introduce some homozygous loci that were removed before the phasing process. While these loci do not provide strand-specific information, omitting them entirely may prevent us from accurately imputing expression altogether.

Once we narrow down the set of SNPs in consideration, we can formulate a predictive model for expression based on haplotype variation. The hypothesis is that the strand-specific signals of haplotypes will provide better insight and predictive value than unphased data when modeling the link between genomic variation and expression .

3.2 Current Literature

Early work on the expression prediction problem was conducted by Beer et al. in 2004, in which RNA microarray expression data and DNA sequences upstream of genes were used for training a predictive model. They chose a Bayesian network approach to model probabilistic dependences in the data, and were ultimately able to predict expression patterns for 73% of tested genes in *Saccharomyces cerevisiae* [1]. More recent efforts generally focus on genetic data of human origin, where the degree of complexity and potential for biomedical relevance are both significantly higher. In the last three years new research on the inference task has included a multitude of computational techniques, from low-rank matrix completion to deep learning [2, 3]. Although the current state of the field shows clear progress, it is difficult to compare results between frameworks because the type of expression data and formalization of the prediction problem differ between works.

The PrediXcan platform developed by Gamazon et al. builds on a formalization that is most closely related to the one described above [4]. Their model begins by isolating the genetically-regulated component of expression for each gene and then utilizes regression approaches such as LASSO (least absolute shrinkage and selection operator) and elastic net on a database of variant-expression relationships to infer expression across the population. A chief benefit of these regression approaches is that they incorporate variable selection. In the setting of genetic variation, this manifests as an opportunity for SNP selection whereby an optimal subset of SNPs from the pool of 2^K subsets is selected as the model is being trained. Notably these subsets need not contain solely contiguous SNPs, representing an advantage over the tract-based method developed in the next section.

4 Tractized Suffix Trees

Given the formulation of the expression prediction problem described above, we propose a novel method for inferring expression from haplotypes using a modified tag-based suffix tree. Although suffix trees represent greater computational overhead than other combinatorial methods, once constructed they brilliantly capture the shared substructure of a set of related sequences. After the tree is constructed, we apply a voting theory-based approach to prediction whereby relevant nodes in the tree “elect” the most likely expression class of a chosen individual. The incorporation of voting theory allows for flexibility in handling the amorphous suffix tree tag data and, to our knowledge, represents a novel approach to the expression prediction problem.

4.1 Tract Tree Construction

The construction of our modified suffix tree draws on the Tractatus model developed by Aguiar et al. [6]. Both are developed around the concept of a *haplotype tract*, which is a contiguous segment of a haplotype shared by one or more samples in a set. Simply referred to as *tracts*, these shared contiguous sub-sequences are uniquely defined by start and end indices. Our method constructs a *tract tree* of all tracts in a haplotype set, thus capturing all repeated substrings found in distinct haplotypes. Since each unique tract is represented by a single path from root to internal node regardless of how many times it appears in the population, the tract tree also provides compression of population-wide information.

The immediate challenge of constructing a suffix tree from haplotypes is the high degree of internal substructure intrinsic to binary vectors. Any tree constructed purely from `0/1` strings is guaranteed to capture *intra*-haplotype relationships (ie. identity between different sub-sequences of the same haplotype), while we are only concerned with index-dependent *inter*-haplotype relationships (ie. tracts). To obscure the internal sub-structure of the binary haplotype vectors we apply a transform to each haplotype, originally developed by Aguiar et al. Letting h_i represent haplotype i, the allele of h_i at position j is given by $h_{i,j}$. Here "allele" refers to the bit at position j, such that $h_{i,j} \in \{0, 1\}$. We now model each $h_i = h_{i,1}, h_{i,2}, ...h_{i,n}$ with a new string $d_i = (h_{i,1}, 1), (h_{i,2}, 2), ..., (h_{i,n}, n)$ for $1 \leq i \leq n$. To simplify the information stored in these position-allele tuples, we replace each tuple $(h_{i,j}, j)$ with the integer $2 * j + h_{i,j}$. The transformed strings are known as *tractized* haplotypes.

Once the haplotypes have been tractized, we can begin constructing a suffix tree that represents the set of all sequences in the population. Before insertion into the tree each haplotype is given a unique tag, represented in Figure 1 with distinguishing colors. While the tree is being constructed, tags are placed on nodes that represent tracts found in the corresponding haplotypes. In this way each node in the tree stores the tags of haplotypes containing its unique tract, and by definition also stores all of the tags found on its descendants. Herein the tree differs from a standard suffix tree in that both its nodes and its edges contain information about the sequences underlying the tree. More specifically the nodes store information about which individual haplotypes share common tracts and the edges store the exact sequences of the tracts themselves.

The fundamental concept behind this approach is that every node in the tree uniquely represents a tract found in the set of haplotypes. Conveniently, the tagging method introduces no new complexity to the task of suffix tree construction. The need for a node to inherit all tags found below it in the tree can be met with relatively few additions to a standard recursive $O(n^2)$ suffix tree construction approach. Figure 1 illustrates a simple example of a tract tree constructed from a population of 4 haplotypes.

4.2 Voting Theory-Based Prediction Approach

Once the tract tree is constructed, its sets of tags collectively represent all shared tracts in the population. The hypothesis underlying our prediction technique is that these shared subsequences in the tractized haplotypes are related to expression. To this end, the tract tree provides a space-efficient and straightforward representation of all such relationships. We begin by defining H to be the set of all haplotypes in the population, such that h_i represents haplotype i. For a given gene, each $h_i \in H$ has a strictly positive, continuous value representing how much individual i expresses the gene of interest. Due to the discrete nature of the tract tree (and the field of combinatorics more broadly), we transform the regression problem of predicting expression to one of classification. First let K be a chosen number of expression levels, and let each h_i now be associated with one of the levels based on its true expression value. With E as the set of expression values for the gene of interest, we let $e_i \in E$ denote the expression level attributed to h_i. Accordingly we have that $e_i \in \{0, ..., K - 1\}$ where $e_i = 0$ if h_i expresses the gene

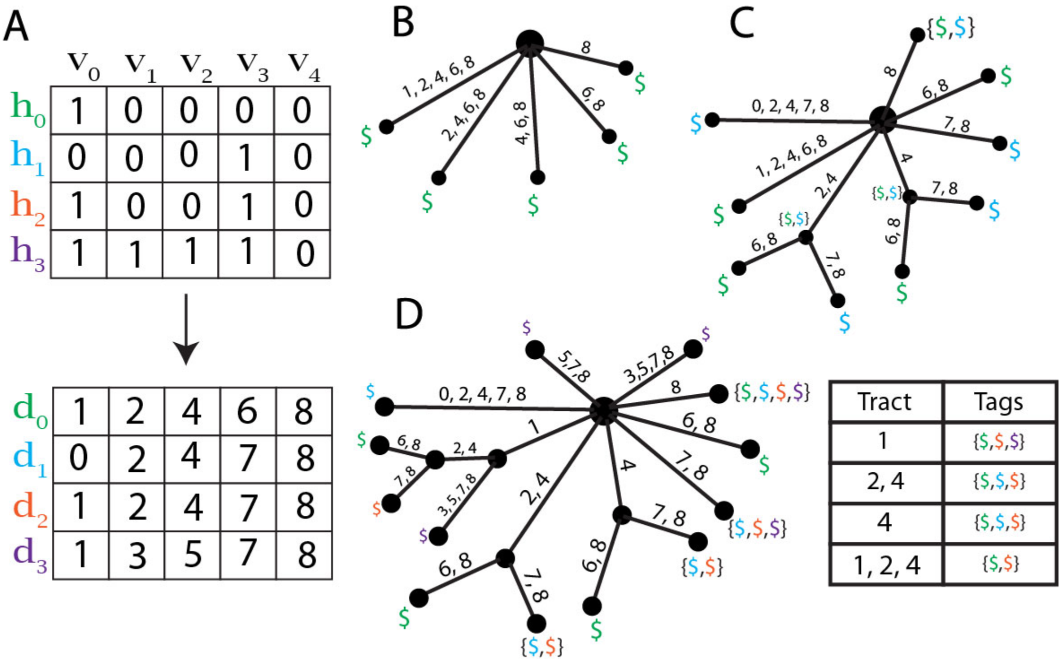


Figure 1: An example of constructing a tract tree. (A) First each h_i in the set of haplotypes is given a unique tag, represented by one of four colors. Before suffix tree construction, the set of haplotypes is also *tractized* to remove intra-haplotype similarity. The position-allele pairs are represented as integers for simplicity. (B) Tractized haplotype d_0 is inserted into the tract tree. Note that the first haplotype always inserts p nodes into the tree, where the indices in each h_i range from $\{v_0, ..., v_{p-1}\}$. (C) Tractized haplotype d_1 is now inserted into the tree, forming two new internal nodes representing the tracts $\{2, 4\}$ and $\{4\}$. Both are labeled with a set of tags containing the unique identifying colors of d_0 and d_1. These tag sets capture the fact that d_0 and d_1 share two such tracts in common. Another way to understand the tag sets is to consider that every internal node in the tree "inherits" the set of all tags associated with the nodes that descend from it. (D) The final two tractized haplotypes are added to the tree. The tag sets on each of the four internal nodes are depicted in a separate table to avoid clutter.

at the lowest level across the population and $e_i = K - 1$ if h_i expresses the gene at the highest level. All expression levels betwee 0 and $K - 1$ group together haplotypes with corresponding intermediate levels of expression.

Our goal is to define a classifier F such that $F(h_i)$ predicts e_i with minimal false classification across the population. For training, the tract tree gives us sets of tags representing shared tracts. Let S_i be the set of tag sets associated with h_i, such that $s_j \in S_i$ if s_j is a set of tags containing h_i's tag and at least one other tag. Simply put, s_j represents a tract shared by h_i and every other haplotype represented in the tag set. Collectively, S_i contains all tracts that h_i shares with all other members of the haplotype set. Thus F becomes a function of S_i, E, and K that imputes the expression class e_i of h_i.

A key challenge to defining F is the non-standard dimensionality of the S sets. This is a consequence of the fact that we make no guarantee that all haplotypes in the population have the same number of relevant tagged nodes or that each relevant node contains a predetermined number of other tags. Intuitively, haplotypes that share great identity with others in H will have more relevant nodes and tagged relationships than those that display less identity with other haplotype samples. The amorphous nature of the tag sets means that it is difficult to vectorize the set of relationships captured by the tract tree, or define a notion of “features” for each sample based on its set of tags. These difficulties led to the idea of a voting theory-based algorithm in which the non-uniformity of data across samples would not represent an issue.

> **Voting theory**, originally developed in the field of economics, is the mathematical formalization of the dynamics that take place in voting processes. Although planning a fair election may at first appear simple, economist Kenneth Arrow proved in 1952 there is no consistently fair method to distinguish between three or more candidates when using preferential voting [7]. In place of an optimal voting scheme, theoreticians have proposed a set of 4 fairness criteria (Monotonicity, Condorcet, Majority, and Independence of Irrelevant Alternatives) that should be optimized however possible. Our prediction strategy aligns well with the four criteria, albeit only in a qualitative sense. Interestingly, the concept of voting theory has been applied at least once before in the field of computational biology, specifically to the problem of deriving consensus sequences [8].

In our case, we allow each $s_j \in S_i$ to vote on the most likely expression class for h_i based on the expression classes of the other haplotypes in the tag set. Although there are no objective criteria for developing such a voting scheme, it is immediately clear that not all s_j should be valued equally. Here we settle on two guiding notions:

1. the contribution of s_j's vote should be proportional to length of the tract that s_j represents (quantified by $\mathcal{L}(s_j)$)
2. the contribution of s_j's vote should be proportional to the margin by which s_j determines its preferred class (quantified by $\mathcal{C}(s_j)$)

The first criterion is intuitive: we want longer shared tracts to have more influence on the prediction relative to shorter ones. The latter criterion is less obvious. It attempts to capture the concept of s_j's *confidence*, or how ‘split’ s_j was when casting its vote. The guiding principle here is that if a tag set contains equal proportions of tags from each of the K expression classes, then s_j's tract is equally represented across all expression levels. Accordingly, s_j provides no indication of how to classify h_i. Thus we want the contribution of s_j's vote to vanish as the confidence of s_j goes to 0. With these notions in mind, we define $\mathcal{L}(s_j)$ and $\mathcal{C}(s_j)$ as follows:

$$\mathcal{L}(s_j) = \frac{L_{s_j}}{\sum_{p=1}^{J} L_{sp}} \qquad \mathcal{C}(s_j) = \frac{n - \frac{1}{k}}{\frac{K-1}{K}}$$

...where n is the proportion of the tags in s_j associated with the most common expression level, L_{s_j} is the length of the tract associated with L_{s_j}, and J is the number of tag sets in S_i. Note that $\mathcal{C}(s_j) \to 0$ as $n \to \frac{1}{k}$ (that is, as the winning class approaches equal proportional representation with all of the other classes). In both quantities above the denominators act purely as normalization terms to ensure that $\sum_{j=1}^{J} \mathcal{L}(s_j) = 1$ and $\mathcal{C}(s_j) \in [0, 1]$. The results of the voting scheme are unchanged if these factors are removed. Combining these terms, we define the classifier F as:

$$F(h_i) = \underset{k \in \{0,...,K-1\}}{\text{argmax}} \quad X_k \ ,$$

$$\text{where } X_k = \sum_{s \in S_i} \mathbb{1}_{(s,k)} \mathcal{C}(s_j) \mathcal{L}(s_j) \quad \text{and} \quad \mathbb{1}_{(s,k)} = \begin{cases} 1 & \text{if } s \text{ votes for } k \\ 0 & \text{otherwise} \end{cases}$$

5 Applications

We chose to implement our predictive model in Python using a standard $O(n^2)$ algorithm for suffix tree construction, modified to incorporate the system of node-tagging described above. Our method was then applied to a variety of simulated datasets to gauge success while varying the number of expression classes, the length of the input haplotypes, and the amount of noise in the generated expression. Finally, two limited trials were performed on real biological data from the National Institute of Health's GTEx Database [9].

5.1 Simulated Data

Given the difficulties of handling real genetic data, we began by simulating coupled haplotype/expression data to get a sense for how our voting scheme would perform in practice. Simulated data was also valuable because it allowed us to directly test hypotheses about the model's performance, such as its predictive ability when tract length is varied and its robustness to noise.

When simulating, we begin by letting n be the number of simulated haplotypes, with each haplotype represented by a binary vector of length p. The sequence of `0`s and `1`s for each haplotype is initially generated at random such that there is equal probability of seeing either bit at every position in each of the n strings. Once the random haplotypes are generated, common tracts were inserted and associated expression values were generated. More formally, let K be the number of expression classes we wish to simulate and R be a set of K functions that determine expression values based on expression class. For each $h_i \in H$, a random expression class k is assigned and its expression is given by:

$$R_k(\mu, T, e) = x_1 + kT + x_2(eT)$$

...where $k \in \{0, ..., K-1\}$, x_1 is generated from a uniform distribution $\mathcal{U}\big(0, 2\mu\big)$ and x_2 is generated from the uniform distribution $\mathcal{U}\big(-T, T\big)$. In simple terms, we can say that a haplotype in expression level k has an associated level of expression that is uniformly distributed about a mean proportional to k. The parameter T allows us to toggle the difference in means between the simulated expression classes. Noise is constrained by e, which is strictly non-negative and can be arbitrarily large. Intuitively, increasing T makes prediction easier and increasing e makes prediction more difficult. Based on the given function R_k chosen to generate expression, a class-specific tract is inserted into h_i to ensure that it contains a conserved subsequence shared with those of its class. This subsequence is generated randomly and inserted into the same location in each chosen member of class k.

Testing began with n-fold cross-validation on the simulated populations. Every data point consists of a single haplotype and its paired expression value. Each was removed from the population and had the voting algorithm applied to the sets of tags at all shared nodes in the modified suffix tree. In the strip plots of Figures 2 and 3, individuals are

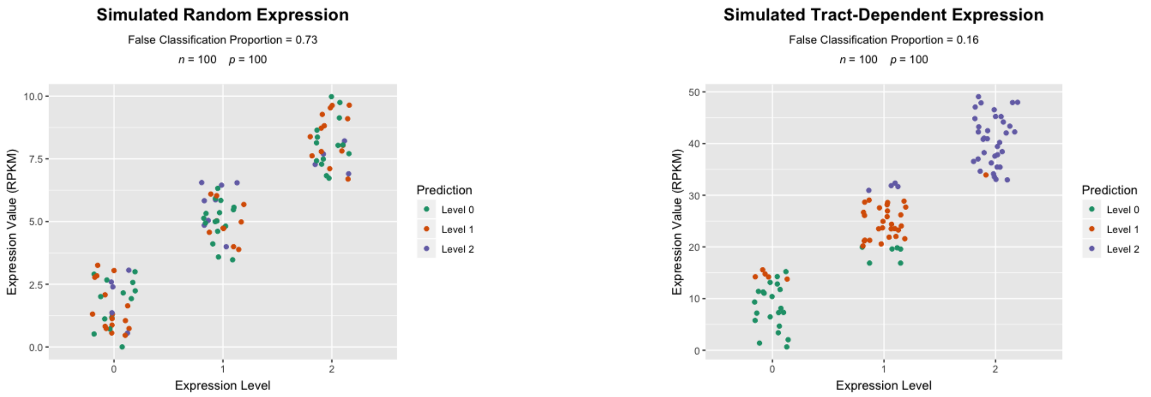


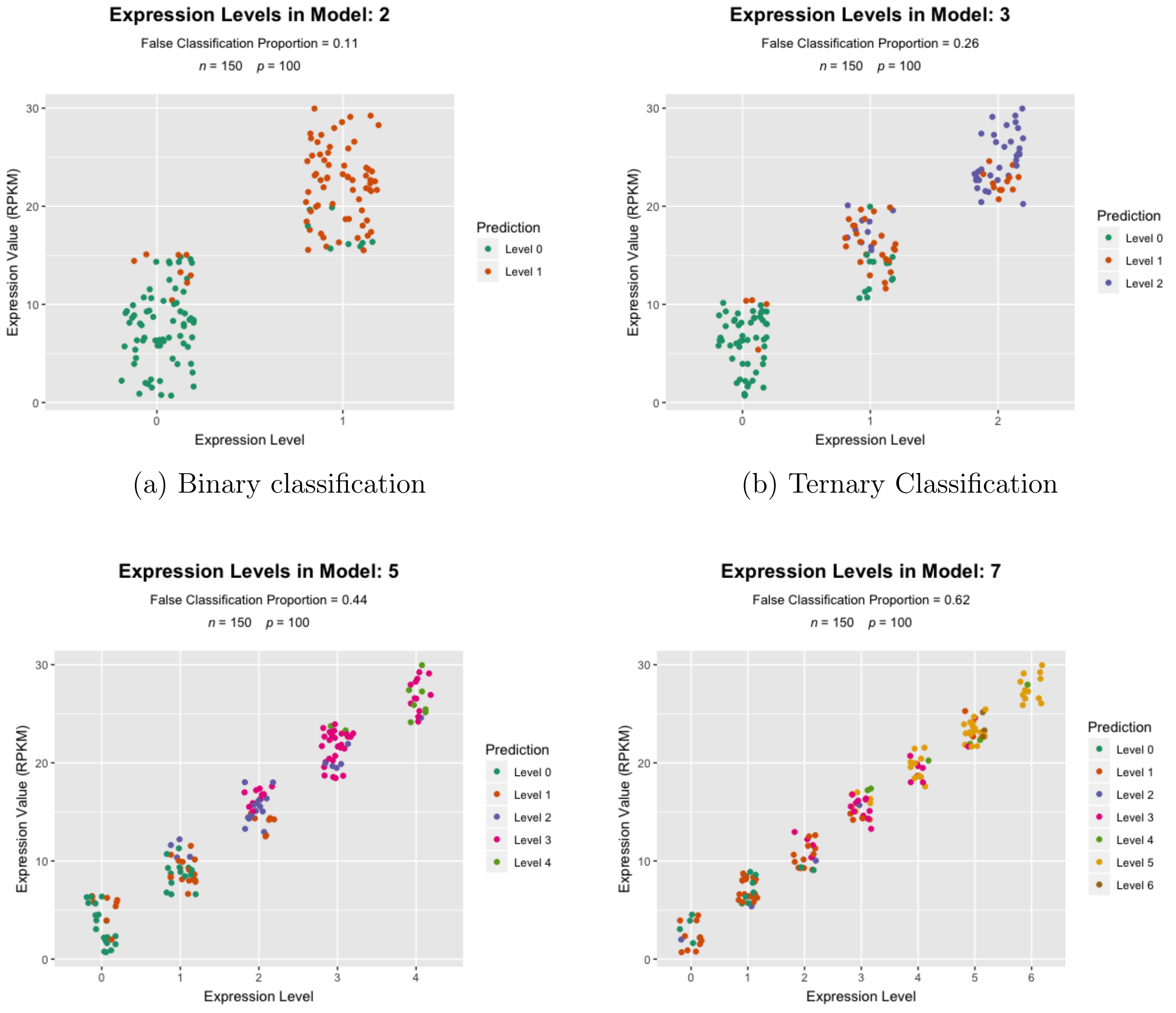
Figure 2: (Left) A sanity check. (Right) Successful n-fold cross validation on a population with simulated tract-specific expression. The generated data is noise-free ($e = 0$) with parameters $\mu = 5$ and $T = 10$.

grouped into K classes by expression on the x-axis and are plotted vertically based on their exact expression value. Each point is colored based on the prediction of the voting scheme. Ideally we hope to observe clusters of points that are colored identically, with minimal mixing.

Although data was simulated with a predetermined K, the voting algorithm and associated tagged suffix tree are not provided this value. This reflects the reality of biological data, in which we do not know *a priori* how many classes best represent the distribution of gene expression. With this in mind, we investigated how the predictive model performs when it attempts to classify the same set of individuals in separate trials where the number of classes in the model is varied. As we expected, we observe better predictive performance when the number of classes in the model is lowered below the true value of K and poorer results when the data is over-classified (ie. the number of levels in the model is greater than the true value of K).

The length of the common tracts shared between members of the same expression class is also of interest. Again, we have no way to determine in advance the length of shared subsequences in a set of haplotypes derived from genomic sequencing. To investigate how shared tract length might affect predictive performance, we considered two opposing cases: one in which the shared tract size scales with the length of the haplotypes, and another in which the tract size is constant. In the former case we assume that longer haplotypes will inherently have longer shared tracts, while in the latter we expect tracts of static bounded length. The results of these tests, summarized in Figure 4, suggest that predictive performance is only marginally worse when common tracts remain static in the face of increasing haplotype length. Although this may at first seem counter-intuitive, one explanation lies in the incorporation of the confidence term ($\mathcal{C}(s_j)$) into the voting scheme. It appears that, at least empirically, the notion of confidence-based voting allows the prediction algorithm to value highly class-conserved subsequences in the haplotype population even when these tracts represent an increasingly small portion of the growing haplotypes. This result suggests that a voting-based methods may be an effective way to capture short yet highly conserved genomic elements in future predictive models.

All results discussed thus far have been on data generated without the introduction of noise (ie. $e = 0$). However the signals in real biological data are almost always muddy, and subject to aberrations such as lab errors and genomic interactions entirely overlooked by our models. Figure 5 depicts the predictive performance of our model on simulated



(a) Binary classification

(b) Ternary Classification

(c) True Number of Classes

(d) Over-Classification

Figure 3: Varying the model parameters for a population with 5 simulated classes ($\mu = 5$, $T = 10$, $e = 0$). Predictive performance, as determined by the false classification rate, improves as the model is simplified.

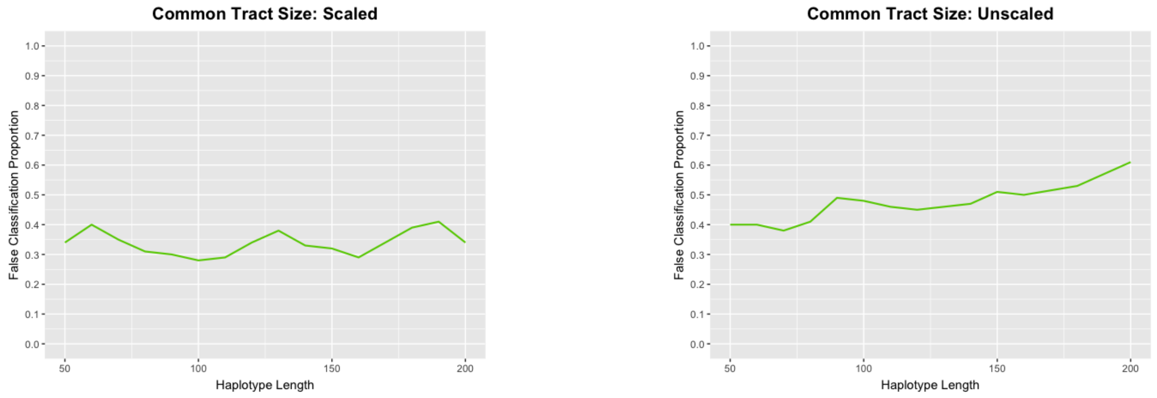


Figure 4: Results are roughly similar between trials involving scaled and unscaled tract lengths ($K = 3$, $\mu = 5$, $T = 10$, $e = 0$)

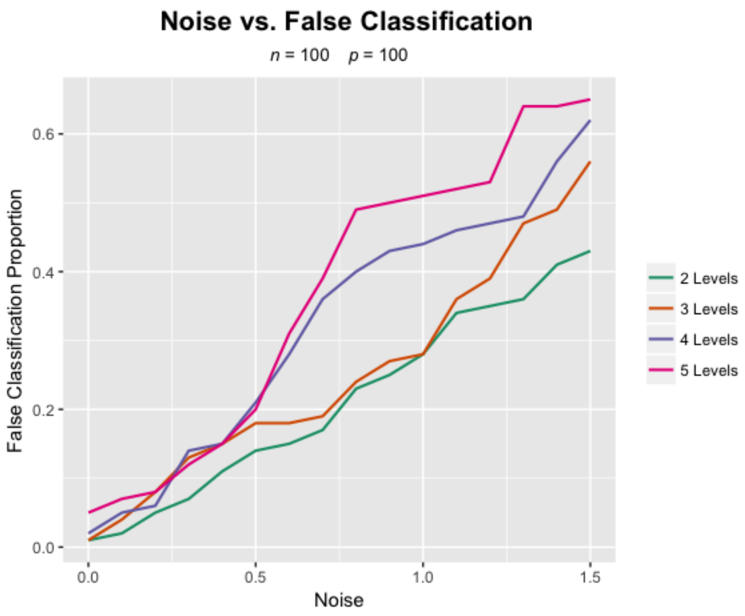


Figure 5: Results suggest that predictive performance decreases roughly linearly with increasing noise across all 4 chosen complexities ($\mu = 5$, $T = 10$). As expected, simpler models generally provided better predictions across all noise levels. Overall performance for trials with $K > 5$ was poor even when $e = 0$, making analysis of noise fruitless on these more complex models.

datasets in which the error parameter e is allowed to ranged from `0` to `1.5`. Note that a noise value of $e =$ `1.5` means that the expression value generated by the chosen function X_k is subject to the addition of a random value in the interval $[-1.5T, 1.5T]$, or 1.5 times the difference in mean expression between two adjacent classes.

5.2 Real Data

Having confirmed the viability of our method on simulated data, the next logical step was to apply the pipeline to real biological data. A number of genome/expression databases now exist, including those provided by the Nation Institutes of Health (NIH), the European Bioinformatics Institute (EBI) and the Genetic European Variation in Health and Disease Consortium (GEUVADIS). The NIH's Genome-Tissue Expression Project (GTEx) was chosen for analysis due to its prevalence in recent literature and its comprehensive nature. The controlled-access portion of the database contains 94 terabytes of data and grows periodically as independent collaborators of the GTEx Consortium provide new sequencing results. The sheer size of the database presented a host of complications, from file storage considerations to concerns surrounding computational efficiency. For perspective, the average sequence alignment map (SAM) file representing low sequencing coverage for a single individual would be on the order of 27 gigabytes, while higher coverage samples exceed 125 gigabytes. With storage limitations in mind, a population of 15 low-coverage individuals were chosen for analysis. Sequencing data was assembled into haplotypes and prediction was performed. Since coverage was low, the assembled haplotypes had few contiguous blocks shared among all 15 individuals. Consequently two genes containing the longest shared continguous blocks were chose, yielding haplotype lengths of 57 and 42 respectively. MUC6 codes for an oligomeric mucus protein found in the gut lumen, and KCNJ12 encodes a specific potassium ion channel in cell walls. Both genes happened to contain one outlier in terms of expression, and thus the population was reduced to 14. Predictive performance in both cases was relatively mediocre, ultimately suggesting that further work must be done to engage with larger volumes of real data and improve the efficacy of our suffix tree voting algorithm.

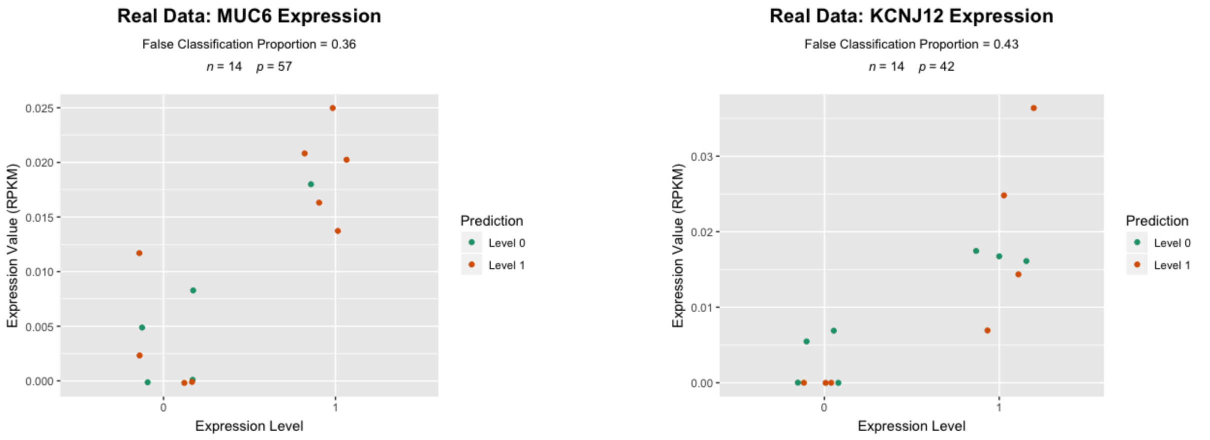


Figure 6: Results form MUC6 and KCNJ12 are encouraging, but limited by population size and haplotype length. These trials are provided as proof-of-concept rather than validation of success.

6 Future Directions

From a computational perspective, more analysis of real data from GTEx and other genome/expression databases is required. In particular, more time and computational resources will be required to allow for larger populations and higher-coverage haplotype assemblies to determine the true capabilities of our voting theory approach. As data size increases, we hope to improve code efficiency first by re-implementing the suffix tree implementation to utilize a linear time construction algorithm rather than the naive $O(n^2)$ method currently used. Ukkonen's linear time suffix tree construction algorithm is a promising starting point, however some modification will be required to incorporate the node-tagging method necessary for our voting scheme. [10]

In terms of further refining our predictive algorithm, a reasonable starting point would be to incorporate scaling factors into the voting scheme. These parameters would allow the model to toggle the relative effects of confidence and length when considering the vote of a given tag set, which should provide greater predictive performance if these parameters can be optimized during the training phase. We may also consider entirely new approaches to performing prediction based on the sets of tags given by the tract tree. For example, it may be possible to construct a Bayes classifier for expression level if we can develop a notion of the likelihood of witnessing the sets of tags at shared nodes conditioned on each of the K expression levels. Formally, we would try to compute the quantity $p(C_k|h_i)$ where C_k is the set of haplotypes in the population that belong in expression level k. We can decompose this condition probability as follows:

$$p(C_k|h_i) = \frac{p(h_i \mid C_k)p(C_k)}{p(h_i)} = \frac{p(h_i \in C_k \mid S_i)p(C_k)}{p(h_i)}$$

The first quantity is the result of applying Baye's Rule to $p(C_k|h_i)$. The latter term translates this probability into more explicit terms using the quantities we derive from the tractus tree. We disregard the denominator (it does not depend on C_k and can be treated as a constant) and use the chain rule to re-define the numerator from a conditional probability to a joint probability:

$$p(h_i \in C_k \mid S_i) = p(C_k, s_1, ..., s_j, ..., s_J), \text{ where } |S_i| = J$$

Now we assume that each $s_j \in S_i$ is conditionally independent of every other element in S_i. This assumption of "naive" conditional independence allows us to re-express the joint probability above as:

$$p(h_i \in C_k \mid s_1, ..., s_J) \propto p(C_k, s_1, ..., s_J), \text{ where}$$
$$p(C_k, s_1, ..., s_J) = p(C_k)\ p(s_1 \mid C_k)\ ...\ p(s_J \mid C_k) = p(C_k) \prod_{j=1}^{J} p(s_j \mid C_k)$$

Thus the classifier F is defined as:

$$F(h_i) = \underset{k \in \{0,...,K-1\}}{\text{argmax}}\ p(C_k) \prod_{j=1}^{J} p(s_j \mid C_k)$$

The task here is to quantify $p(s_j \mid C_k)$ or, as stated above, to develop a notion of the likelihood of witnessing a set of tags at a shared node conditioned on the expression level. Even when this quantity is defined, it is still to be seen whether the assumption of conditional independence of tag sets in the tract tree proves sound in practice.

7 Conclusion

The tractized suffix tree voting method developed in this project represents a novel take on the expression prediction problem for genome/expression datasets. Empirical results suggest that a strand-specific, tract-based approach to capturing genetic variation for imputing expression may hold promise, but the method currently requires greater refinement and experimental validation before comparison to already published frameworks. On a higher level this project is an interesting look into the intellectual product of bringing computer science, biology, applied mathematics, and economics into conversation with each other. Ultimately our model should serve as a proof-of-concept, and a modest validation of the value of cross-disciplinary study.

References

[1] Beer, Michael A., and Saeed Tavazoie. “Predicting Gene Expression from Sequence.” Cell 117.2 (2004): 185–198.

[2] Kapur, Arnav, Kshitij Marwah, and Gil Alterovitz. “Gene Expression Prediction Using Low-Rank Matrix Completion.” *BMC Bioinformatics* 17 (2016): 243. BioMed Central.

[3] Singh, Ritambhara et al. “DeepChrome: Deep-Learning for Predicting Gene Expression from Histone Modifications.” *Bioinformatics* 32.17 (2016): i639–i648.

[4] Gamazon, Eric R et al. “A Gene-Based Association Method for Mapping Traits Using Reference Transcriptome Data.” *Nature Genetics* 47.9 (2015): 1091–1098.

[5] Reich, David E. et al. “Linkage Disequilibrium in the Human Genome.” *Nature* 411.6834 (2001): 199–204.

[6] Derek Aguiar, Eric Morrow, Sorin Istrail, "Tractatus: an exact and subquadratic algorithm for inferring identity-by-descent multi-shared haplotype tracts." *RECOMB* (2014): 158-173.

[7] Arrow, Kenneth J., and Eric S. Maskin. "Social Choice and Individual Values". *Yale University Press* (2012)

[8] Day, W. H., and F. R. McMorris. “Interpreting Consensus Sequences Based on Plurality Rule.” *Mathematical Biosciences* 111.2 (1992): 231–247.

[9] GTEx Consortium. The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans. *Science* (New York, NY). 2015; 348(6235):648-660.

[10] Ukkonen, E. On-line construction of suffix trees. *Algorithmica* 14(3) (1995) 249–260