The Pennsylvania State University
The Graduate School

LEARNING SIGNALS IN GENOMIC SEQUENCE ALIGNMENTS FOR IDENTIFICATION OF FUNCTIONAL ELEMENTS

A Thesis in
Computer Science and Engineering
by
James Taylor

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2006

The thesis of James Taylor was reviewed and approved* by the following:

Webb Miller
Professor of Biology and Computer Science and Engineering
Thesis Advisor, Chair of Committee

Francesca Chiaromonte
Associate Professor of Statistics and Health Evaluation Sciences

Ross Hardison
Professor of Biochemistry and Molecular Biology

Hongyuan Zha
Professor of Computer Science and Engineering

Piotr Berman
Associate Professor of Computer Science and Engineering

Vijaykrishnan Narayanan
Associate Professor of Computer Science and Engineering
Graduate Program Officer

*Signatures are on file in the Graduate School.

ABSTRACT

The structure of genomes, how they encode function, and how they evolve is still quite mysterious. Even the best understood functional elements – regions that code for proteins – are far from exhaustively annotated. Other functional elements, such as the *cis*-regulatory modules that control gene transcription, are even more poorly understood. Comparisons between the genomes of different species can be a useful tool to understand the structure of these elements and improve our ability to identify them. However, such comparisons also raise new questions as we observe regions with distinctly atypical evolutionary patterns but no clear relationship to any known function. Other sequence signals, such as base composition and specific motifs, are also useful for identifying functional regions, but the specific signals to use for identifying a given class of elements are not always obvious. When training data for a class of functional elements is available, applying a machine learning method to learn the relevant sequence and evolutionary patterns has the potential to better identify functional elements. In this work we describe a computational method, called ESPERR (Evolutionary and Sequence Pattern Extraction through Reduced Representations), which uses training examples to learn encodings of multi-genome alignments into a reduced form for predicting a chosen class of functional elements. We show that ESPERR gives excellent performance on several problems. We first describe using ESPERR for discriminating two classes of regions, with particular focus on discriminating *cis*-regulatory regions from neutral DNA, producing a score called "Regulatory Potential" that has excellent predictive power. We also consider additional pairwise discrimination problems: discrimination of DNAseI hypersensitive sites using training data produced by the ENCODE project; and screening highly conserved regions for developmental enhancer activity using training data from the VISTA Enhancer Browser. We also demonstrate the flexibility in the ESPERR procedure with respect to the type of problem addressed by showing a generalization to multi-class classification: predicting whether cDNA $5'$ ends are tissue-specific promoters, widely expressed promoters, or not promoters.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This work draws heavily from two papers written as part of my Ph.D. research at Penn State:

- "ESPERR: Learning strong and weak signals in genomic sequence alignments to identify functional elements", written with Svitlana Tyekucheva, David King, Ross Hardison, Webb Miller and Francesca Chiaromonte, (currently in review for *Genome Research*)
- "Leveraging ENCODE data to predict widely expressed and tissue-specific transcriptional promoters in the human genome", written with Nathan Trinklein, Ross Hardison, Webb Miller and Francesca Chiaromonte, (currently in review for *Genome Research*)

I would like to thank all of my co-authors, this work would not have been possible without them. Additionally, this work uses a substantial amount of data provided by the ENCODE project, and I would like to thank all of the labs involved in that project for making their data publicly available.

CHAPTER 1

INTRODUCTION

A genome is the DNA sequence that encodes the full genetic information for an organism, and, to the best of our current knowledge, contains the complete set of information necessary to create a living example of that organism. Genomes contain regions associated with a variety of known functions, including regions that code for proteins or functional RNAs (*genes*), regions that control gene expression, regions involved in controlling the structure of the DNA and its location in the cell's nucleus, and others. Additionally, genomes likely contain both non-functional regions and regions associated with functions that are not currently understood. Identifying all of the functional regions in genomes, and assigning specific functions to those regions, is a critical step toward understanding the complexity and variety of living organisms.

1.1 GENES AND GENE REGULATION

The class of genomic functional elements that is currently best understood is protein coding genes. These regions of the genome encode the information the cell uses to create proteins by assembling strings of amino acids. In eukaryotic genomes, each gene is encoded in the genome as a series of *exons*, separated by *introns*. The process of reading a gene and producing a protein product is complex, but roughly consists of these stages:

- The gene is copied from DNA into an RNA transcript by a process called *transcription*.
- The introns are removed by a process called *splicing*.
- A sequence of amino acids corresponding to the protein coding portion of the spliced transcript is produced by a process called *translation*. The coding region is read as triplets of bases, each of which specifies an amino acid determined by the *genetic code*.

Controlling the amount of gene expression at a given time, place, and set of environmental conditions is critical, and the process of gene expression can be regulated at every stage. We are going to focus our attention on one regulatory step: the regulation of gene transcription, which controls when and at what rate transcripts are produced from a gene.

Figure 1.1 illustrates many of the components involved in transcriptional regulation. Genomic DNA is normally tightly packed into *chromatin*. For genes to be transcribed the chromatin must be unpacked so that the machinery of transcription can access the DNA. Once the gene is accessible, the *transcription initiation complex*, consisting of an RNA polymerase and other factors assembles at the *promoter* region of the gene. Once fully assembled this complex then produces the transcript.

Accessibility of the promoter region and transcription initiation are both regulated by *transcription factors*, proteins that bind to the DNA. The targets of these factors are called transcription factor binding sites or TFBSs. These sites can be found in the promoter region, proximal to the gene start site, as well as far from the gene they control. TFBSs often occur in clusters called *cis*-regulatory modules or CRMs. Transcription factors bound at different sites, along with co-factors, work combinatorialy to allow complex control of gene expression.

1.2 SEQUENCE SIGNALS ALLOW COMPUTATIONAL PREDICTION OF FUNCTIONAL ELEMENTS

The structure of functional DNA is constrained – changes in functional regions would affect their associated function. Functional regions that encode instructions for building products such as proteins are constrained by the need to interact properly with the machinery that reads them, and to encode the correct product. Functional regions where the DNA interacts directly with other factors are constrained by their biochemical interactions. In both cases, this results in constraint on the DNA sequence in these regions. The amount and type of constraint on the sequence is highly variable however, ranging from a preference for a certain base composition to ensure a certain DNA structure (different bases behave differently biochemically, and thus

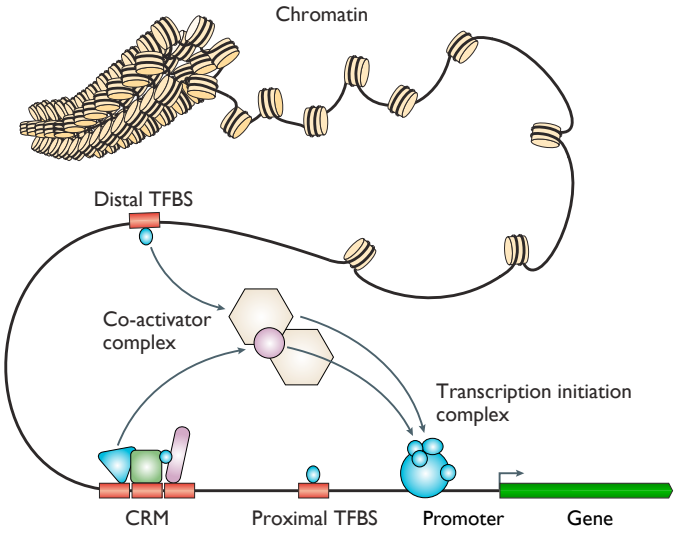

Figure 1.1: Diagram illustrating the regulation of gene transcription. Adapted from Wasserman and Sandelin (2004).

change the shape of DNA), to a requirement that a precise sequence of bases be present.

Computational methods can take advantage of these constraint signals to predict functional elements in the genome. Consider, for example, the signals associated with genes. Specific sequence signals are required at the start of the gene for the transcription initiation complex to bind and initiate. Signals near the intron/exon boundaries are required to guide the splicing process. Translation reads the spliced transcript as a series of triplets called codons, and only certain sequences of codons result in a viable protein being produced. Because the signals associated with genes are so well characterized, a wide variety of computational gene prediction algorithms based solely on sequence signals (and no other biological evidence) have been created. For example, GENSCAN (Burge and Karlin, 1997), one of the most popular gene predictors, uses a probabilistic model that takes into account a variety of signals including splicing signals, the length distribution and 3-periodic structure of exons, and the base composition of genic features compared to non-genic DNA.

While the mechanisms for non-genic functional elements are much less well

understood, properties that confer sequence-level constraint can still be identified. Transcription factors bind specifically to their DNA targets (TFBSs), and the specificity is determined by the biochemical interaction between the protein and the DNA (see Figure 1.2, top). Binding site specificity is important to ensure that genes are expressed correctly. Thus the DNA sequence at these sites is constrained by the need to bind the right transcription factor at the right rate in the right conditions.

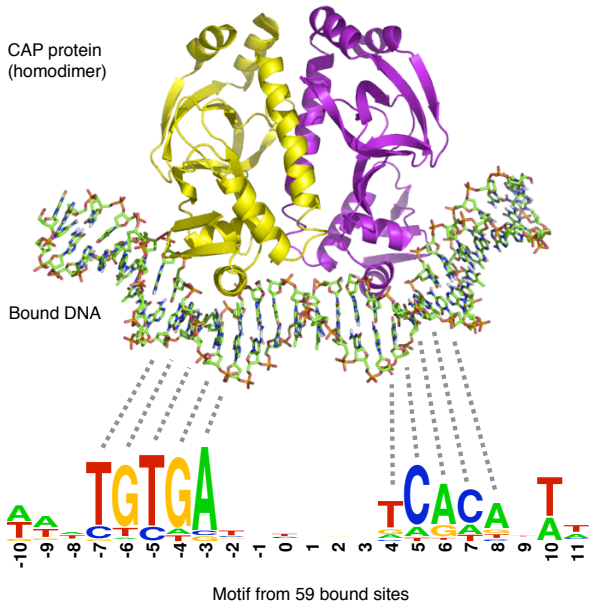

Figure 1.2: Example of a protein-DNA binding interaction (Top). The protein is the *E. coli* transcription factor CAP (Schultz et al., 1991). Also shown is a sequence logo for a position weight matrix determined using 59 CAP binding sites (Crooks et al., 2004).

Computational methods have also been applied to the identification of the targets of DNA binding proteins, specifically transcription factors. In many of these methods a large number of experimentally identified TFBSs are used to infer a common binding site sequence or *motif*, a short sequence of DNA bases frequently associated with binding of that factor. Motifs are often generalized to a probabilistic form called a *position weight matrix* or PWM, in which each position in the motif is represented by probability distribution of observing each DNA base at that position. An example

of a PWM for the binding site of the *E. Coli* transcription factor CAP is shown in Figure 1.2 (bottom).

1.3 COMPARATIVE GENOMICS IMPROVES FUNCTIONAL ELEMENT IDENTIFICATION

Evolution is the process by which genomes change over time. Changes to the genomic sequence are generated by many mechanisms that result in changes of a single base to another base (*point mutations*), as well as the insertion or deletion of multiple consecutive bases (*indels*). However, the persistence of changes in a population is determined by the process of *selection*. Changes to sites in functional regions may affect their functional role, resulting in differential fitness of individuals with such a change. If the change gives the organism a fitness advantage relative to other individuals in the population, that change may be favored and become more frequent in the population (this form of selection is called *positive selection*). However if the change produces a decrease in fitness, individuals with such a change might be selected against, resulting in the change becoming infrequent or vanishing from the population (*negative selection* or *purifying selection*).

Because of the different evolutionary pressures affecting functional and non-functional sites, comparisons between genomes of different organisms can be an effective tool for identifying functional elements. Computational methods for predicting functional regions using genomic sequence alone often have poor specificity in large genomes because the signals they capture also occur frequently by chance. Using multiple genomes to look for signals that are conserved between different organisms can greatly improve specificity. Further, some signals are only evident when considering multiple genomes. Consider again the example of protein coding genes: the genetic code that specifies which codon produces which amino acid is *degenerate* (multiple codons produce the same amino acid), certain changes to the genome sequence (usually at the third position in a codon) do not affect the encoded protein. Thus when comparing sequences of a gene from different organisms, changes are more frequently observed at these degenerate positions than at other

sites. This distinct pattern of constraint can be used to greatly improve computational prediction of genes – see for example the TWINSCAN (Korf et al., 2001) adaptation of GENSCAN.

A common approach for capturing the evolutionary relationships among a set of sequences is through a *sequence alignment*. A sequence alignment is an arrangement of the sequences, or portions of the sequences, that identifies DNA bases derived from a common ancestral base (called *orthologous* bases). This can be viewed as a way of "lining up" the sequences, and inserting *gaps* at positions where sequence segments were inserted or deleted, so that each column of the resulting alignment contains only orthologous bases or gaps. Such an arrangement of sequences contains a substantial amount of information about both the extant species and their phylogenetic history, and is thus a natural starting point for comparative genomic analysis. A variety of methods for pairwise and multiple sequence alignment exist, and it continues to be an active area of research (Batzoglou, 2005).

1.4 Learning the right signals for classifying functional elements

Our knowledge of the structure and function of protein-coding regions allows us to build models that exploit specific signals to effectively predict genes. However, for non-coding functional elements which are less well characterized the right signals to use are not always obvious. For example, consider signals currently used for identifying *cis*-regulatory modules, including: (1) specific sequence patterns, such as motifs associated with elements involved in protein-DNA interactions (e.g. transcription factor binding sites), (2) general sequence composition patterns, such as the high density of CpG dinucleotides found in most ubiquitous promoters, and (3) evolutionary patterns, particularly a high level of between-species conservation, which should characterize functional regions under purifying selection.

While each of these signals is associated with some *cis*-regulatory modules, all of them have limitations (Tompa et al., 2005). Motif-based approaches can have high specificity, particularly when using a stringent consensus sequence, but when the

patterns are degenerate (often the case with transcription factors), they can have both poor sensitivity and a very high false positive rate.

Consider the binding data produced by the Transcriptional Regulation group of the ENCODE project – an effort to comprehensively annotate 1% of the human genome (Consortium, 2004). Using ChIP-chip[1] they identified genomic regions bound by 18 different transcription factors in a variety of conditions and cell lines. Of these 18 factors nine bound to sequences that were not statistically enriched for the factor's putative binding motif (Weng et al., 2006). They also applied *ab initio* motif finding[2] methods to these regions. Motif finding could identify the known TFBS motif for only seven of the nine enriched factors. These results suggest that for a comprehensive set of binding sites, motif-based approaches have weak power.

Similarly, the ENCODE Multiple Sequence Analysis group (Margulies et al., 2006) found a complex relationship between function and evolutionary constraint. They discovered that while many classes of functional elements do show significant association with constrained elements as a whole, a large number of experimental annotations of every type considered by ENCODE (other than protein-coding regions) do not overlap constrained elements, and many constrained elements do not coincide with experimental annotations. These results suggest that interspecies sequence constraint also provides only weak power for comprehensive identification of functional elements.

Thus it seems that while non-coding functional elements show association with various sequence and evolutionary characteristics, rarely will a single signal be sufficient for accurate and comprehensive prediction. While simple descriptive features can be very useful to better understand functional mechanisms, the effects of functional constraint on these elements are myriad – too complicated to be captured effectively by such features alone.

An alternative approach for identification of a class of functional elements – when

[1] ChIP-chip isolates fragments of genomic DNA bound by certain a specific protein (*chromatin immunoprecipitation*), and maps those regions back to the genome by hybridization to a *microarray*.

[2] *ab initio* motif finding attempts to find a shared motif using only sequence data. The regions identified by ChIP-chip are larger than typical protein binding sites, the motif finder attempts to identify the motif for the actual functional sites within these regions.

training data is available – is to apply a computational learning method with the potential to capture both the clear strong signals and the many subtle signals that characterize the class. In this work we describe such a method, denoted ESPERR (Evolutionary and Sequence Pattern Extraction through Reduced Representation). ESPERR uses models capable of learning patterns in multiple genomic sequence alignments that characterize specific classes of functional elements.

CHAPTER 2

THE ESPERR APPROACH

Genomic sequence alignments contain information about the primary sequence of a set of species and the evolutionary relationships among them, and thus provide useful information for discriminating functional elements. However, two major obstacles must be overcome to develop an effective method for learning characteristic patterns from genomic sequence alignments. First, the number of possible alignment columns increases exponentially with the number of sequences in an alignment. This number (over 70,000 for a seven-species alignment) is much too large an "alphabet" to use to find patterns in alignment columns, and thus a reduced representation of the alignment is required. Second, the rules for distinguishing between functional classes based on patterns in alignments are not known a priori, and thus a training regimen is required.

ESPERR solves these problems using models capable of learning patterns both among the species at a given position (evolutionary patterns) and among aligned positions (and thus across the sequence). Underlying these models is a translation or "encoding" of alignments into a simplified representation that preserves a subset of the original information. This reduced representation should remove noise and irrelevant information, but retain all the signals useful for characterizing a particular class of functional elements. We first present ESPERR in the context of pairwise classification, where our training data consists of two sets of alignments (referred to as the positive and negative sets), ESPERR is used to learn an encoding and produce a score that effectively discriminates these sets.

2.1 OVERVIEW

The key component of our method is the selection of such an encoding using (1) phylogenetic relationships to define a reasonable starting point, followed by (2)

a heuristic search procedure that optimizes the encoding based on classification performance. Encodings produced by this procedure, and the models based on them, produce excellent classification performance on a variety of problems.

The ESPERR procedure finds an encoding from multiple alignment columns into a reduced alphabet that retains information useful for discriminating a chosen class of functional elements. The procedure consists of two stages, summarized graphically in Figure 2.1. In the first stage, we reduce the "alphabet" of alignment columns to a size where fitting classification models becomes tractable. This is done by grouping multiple alignment columns based on evolutionary similarity (Figure 2.1A). We start by inferring the ancestral base probability distribution corresponding to each alignment column (section 2.3) – for this we use an extended HKY substitution model, treating alignment gaps as a fifth base (section 2.2). This inference provides a natural way to handle missing data; if data for a species is missing at a given position it is not included in the inference (Figure 2.1A, middle tree). In practice the number of missing species allowed must be limited to ensure good inference. Next, we compute the frequency with which each ancestral distribution occurs in the training data, and apply a novel clustering algorithm (section 2.4) that forms groups of columns preserving both "neighborhood" (similarity of ancestral distributions) and frequency structure. Ancestral distributions correspond to points in a 5-dimensional probability simplex, a 2-dimensional projection of which is used to visualize the clustering step in Figure 2.1B.

The clusters resulting from the initial alphabet reduction in stage 1 provide an encoding that retains a substantial amount of information from the original alignment data, and reduced representations produced in this way can be used effectively for many applications. However performance can be improved substantially by taking such an encoding as a starting point and then using classification performance to optimize the encoding for a particular problem. The second stage of the ESPERR procedure achieves this through an iterative search (Figure 2.1C, section 2.5). At each stage of this search, candidate encodings are generated from the current encoding by either joining two groups or breaking a group into two (a random sample from

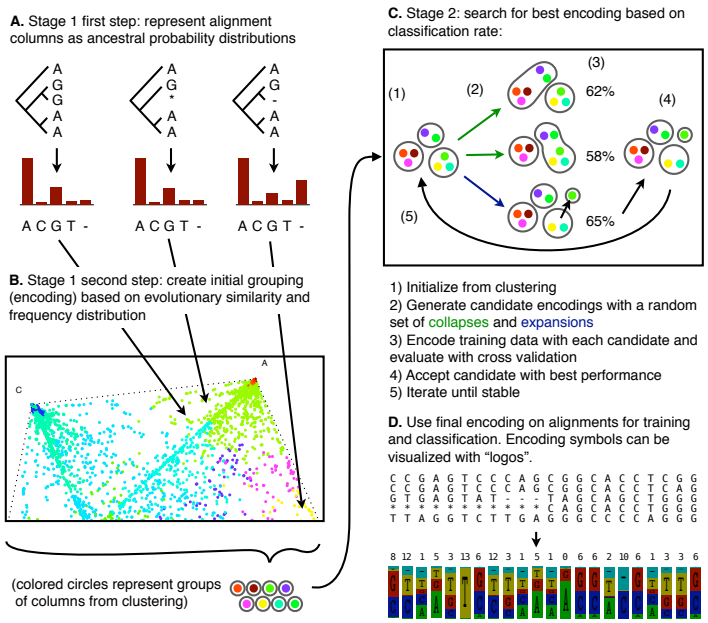

Figure 2.1: Overview of the ESPERR procedure.

each type of candidate is considered). Using the training data, cross validation is run to evaluate the prediction performance of each candidate, and the candidate with the best performance is accepted as the new current encoding. After many iterations without seeing an improvement in performance the search is terminated, yielding an optimized encoding, usually to many fewer symbols (groups) than the starting point.

While this approach could be applied using any classification method, we generally use a log-odds classifier (section 2.7) based on a type of variable-order Markov models (VOMM; Buhlmann and Wyner, 1998; see section 2.6). These models capture variable-length dependencies among positions in sequences. Thus, when applied to strings of encoded alignment columns, VOMMs are able to capture sequence and evolutionary patterns that span multiple alignment columns.

2.2 EXTENDED HKY MODEL

To make inferences on ancestral base distributions, we must first introduce a model of nucleotide substitution for estimating the probability of a given substitution event over

a given branch of the phylogenetic tree. We assume a continuous time Markov process in which a rate matrix Q specifies the instantaneous rate of each substitution event, and express the rates in Q through a smaller number of parameters. In particular, we use the parameterization provided by the HKY model of Hasegawa et al. (1985) consisting of equilibrium probabilities for each base (4 parameters; $\pi_{\mathtt{A}}$, $\pi_{\mathtt{C}}$, $\pi_{\mathtt{G}}$, $\pi_{\mathtt{T}}$), and the ratio between the rates of *transitions* and *transversions* (κ)[1]. We extend this model to accommodate gaps as if they were a fifth nucleotide, introducing an additional equilibrium probability ($\pi_{\mathtt{Gap}}$) and rate ratio (gaps to transversions σ), yielding the rate matrix:

$$Q = \begin{pmatrix} - & \pi_{\mathtt{C}} & \kappa\pi_{\mathtt{G}} & \pi_{\mathtt{T}} & \sigma\pi_{\mathtt{Gap}} \\ \pi_{\mathtt{A}} & - & \pi_{\mathtt{G}} & \kappa\pi_{\mathtt{T}} & \sigma\pi_{\mathtt{Gap}} \\ \kappa\pi_{\mathtt{A}} & \pi_{\mathtt{C}} & - & \pi_{\mathtt{T}} & \sigma\pi_{\mathtt{Gap}} \\ \pi_{\mathtt{A}} & \kappa\pi_{\mathtt{C}} & \pi_{\mathtt{G}} & - & \sigma\pi_{\mathtt{Gap}} \\ \sigma\pi_{\mathtt{A}} & \sigma\pi_{\mathtt{C}} & \sigma\pi_{\mathtt{G}} & \sigma\pi_{\mathtt{T}} & - \end{pmatrix}$$

The parameters of Q are estimated using the Expectation Maximization algorithm implemented in the PHAST software package (Siepel and Haussler, 2004), generally using a fixed tree topology and a sample of genome-wide alignments.

2.3 ANCESTRAL BASE DISTRIBUTION INFERENCE

To infer the ancestral base probability distribution corresponding to a given alignment column we use Felsenstein's algorithm (Durbin et al., 1998; Mayrose et al., 2004). We allow species to be missing for any column, in which case the corresponding leaves of the tree were left out of the inference (treated as "Felsenstein wildcards").

Given an alignment column $x = (x_1, ..., x_m)$, the posterior distribution for the

[1] DNA bases are divided into two types by their chemical structure: purines (A and G) and pyrimidines (C and T). Mutations from a purine to a purine or a pyrimidine to a pyrimidine are called transitions, all others are transversions. Transversions result in a more extreme change to the structure of the DNA, and thus occur less frequently. The rate ratio parameter κ allows for this difference in mutation rates.

base in the common ancestor of the m species is:

$$P(y|x) = \frac{P(x|y)\pi_y}{\sum_{z\in\{\texttt{A,C,G,T,Gap}\}} P(x|z)\pi_x}$$

Felsenstein's algorithm evaluates the likelihood $P(x|y)$ in this equation, recursively proceeding bottom-up along the phylogenetic tree through a series of "triangulations". For a generic stage, let y_0, y_1 and y_2 be, respectively, the position for the ancestor currently under consideration (0), and its two immediate descendants (1 and 2). The basic recursive relation is:

$$P(x(0)|y_0) = \sum_{y_1} P(x(1)|y_1)\Pi_{y_0\to y_1}(\tau(0,1)) \sum_{y_2} P(x(2)|y_2)\Pi_{y_0\to y_2}(\tau(0,2))$$

where $x(A)$ indicates the subset of x corresponding to observed species descending from A , $\tau(A,B)$ the length of the branch linking A and B, and $\Pi_{y_A\to y_B}(\tau(A,B))$ the corresponding transition probability obtained through:

$$\Pi(\tau) = \exp\{-Q\tau\} = \sum_{j=0}^{\infty} \frac{(-Q\tau)^j}{j!}$$

The Q and the π's in these equations are respectively, the rate matrix and equilibrium distribution of the of the HKY+Gap substitution model described in section 2.2.

2.4 CLUSTERING BASED ON PROXIMITY AND ENTROPY

The novel clustering algorithm underlying the first stage of ESPERR groups alignment columns agglomeratively[1], based on distance between corresponding ancestral distributions, and their frequency (occurrence counts for columns create a frequency distribution over ancestral distributions). For distance calculations, each cluster is represented by a centroid defined as the "average ancestral distribution" (weighted with frequencies). To preserve the neighborhood structure, at each stage of the agglomeration we consider merging each cluster with its nearest neighbor (Euclidean distance between centroids). To preserve the frequency distribution, the merger that is accepted at each stage is the one that maximizes the mutual information between

the distributions prior and post merging (in practice this is equivalent to accepting the merger with the maximum entropy; see below). Because the algorithm is based on entropy, clusters must not have zero frequency. Thus we perform an initial pre-clustering, grouping columns that never occur or are very seldom in the training data (occurring less than five times) with their nearest neighbor (having five or more occurrences). The agglomeration is terminated once a desired number of clusters is reached. For all applications described in this work we have stopped at 75. Regardless of the number of species in the training alignments (and corresponding number of initial alignment columns), the appropriate number of symbols at which to transition to the second stage is determined by the amount of training data. This number must be small enough so that the underlying classifier can be learned with some power, yet large enough to allow the second stage flexibility. For the applications presented here, using a VOMM with a maximum order of two as the underlying classifier, 75 is a reasonable size.

In more detail: ancestral base distributions are points in the 5D simplex. At each iteration, a merger is chosen among a set of candidates so as to maximize entropy of the resulting partition. Let G indicate the current partition of alignment columns into groups $g \in G$, each of which contains a fraction of the alignment columns: $p(g) = \frac{n_g}{n}$, where n is the total number of alignment columns in the training data. Let C indicate a set of candidate mergers $c \in C$, and G_c the partition in groups $g_c \in G_c$ (each containing a fraction $p(g_c)$ of the occurrences) resulting from applying merger c to G. We select the merger:

$$c^* = \arg\max_{c \in C} \left(H(G_c) \right) = \arg\max_{c \in C} \left(- \sum_{g_c \in G_c} p(g_c) \log(p(g_c)) \right)$$

Because G_c is "nested" in G, the entropy of the former coincides with the mutual information between the two so that, at each iteration, selecting a merger to maximize entropy is the same as selecting a merger to retain maximal information relative to the current partition. Specifically, the information content between G and G_c is:

[1] In general, *agglomerative* clustering algorithms initially start with each data point forming a single cluster, and then progressively join clusters together to form a smaller set of clusters.

$$I(G, G_c) = \sum_{g \in G} \sum_{g_c \in G_c} p(g, g_c) \lg \frac{p(g, g_c)}{p(g)p(g_c)}$$

Because G_c is a partition of G, the probability mass of any of its clusters is $p(g_c = \sum_{g \in g_c} p_g$, and the joint probability of two clusters is $p(g, g_c) = p(g)$ if g is in g_c, otherwise 0. Thus:

$$\begin{aligned} I(G, G_c) &= \sum_{g_c \in G_c} \sum_{g \in g_c} p(g) \lg \frac{p(g)}{p(g)p(g_c)} \\ &= \sum_{g_c \in G_c} \left(\sum_{g \in g_c} p(g) \right) \lg \frac{1}{p(g_c)} \\ &= - \sum_{g_c \in G_c} p(g_c) \lg p(g_c) \\ &= H(G_c) \end{aligned}$$

the entropy of G_c. Because it employs entropy, this algorithm tends to create clusters of similar mass (number of alignment columns), located depending on the frequency of occurrences in the simplex.

Proximity is used as a constraint; by limiting the set of candidates C in each iteration to mergers involving "neighboring clusters", we ensure that clusters remain spatially contiguous. This can be implemented in several ways, as to give stronger of weaker roles to proximity. For the applications presented in this work, we consider merging each cluster with its nearest neighbor. In other words, we let a merger $c = g_1, g_2$ be a candidate if:

$$d(g_1, g_2) = \min_{g \in G} d(g_1, g) \text{ or } \min_{g \in G} d(g, g_2)$$

2.5 ITERATIVE SEARCH

The second stage of ESPERR – the search – generates candidate encodings, accepts the best based on a figure of merit (FOM), and repeats until an optimal encoding is found. The FOM is the cross validation success rate (see section 2.8), and does not include

"unclassifiable" elements. The search is initialized with the encoding determined by agglomerative clustering in the first stage. We refer to the symbols (groups) produced by clustering as "atoms", because they are never split during the search. At each stage, candidates are generated from the current encoding by either merging two symbols (groups) or extracting an atom from one of the symbols. When the current encoding is large, many candidates will perform close to (a poor) best. Thus we evaluate only a random sampling, e.g. $\gamma = 50$ mergers and $\eta = 20$ extractions, which reduces computations while still producing reasonable moves with high probability. As the current encoding shrinks, γ represents a larger fraction of the possible mergers, and η random extractions continue to afford a degree of reversibility to the search.

Large encodings require more parameters, are more susceptible to over-fitting and thus score more elements in the unclassifiable range, reducing the FOM. Consequently, the search strongly prefers small encodings, and it is possible that evaluating single atom extractions will not be enough to by-pass local optima. We overcome this problem with a heuristic: if the FOM does not increase over w (e.g. 20) consecutive iterations, we consider only extractions for e (e.g. 5) consecutive steps, which allows us to move out of local optima through poorer performing, larger encodings.

Even with this heuristic, it is still possible for the search to make bad moves which then take a long time to be reversed. To recover efficiency, we add a "restarting" heuristic: if we proceed for r (e.g. 50) iterations without reaching an encoding better than the best seen so far, we restart the search at that best encoding. Termination is similar but extends to a much larger number of iterations – we stop if we go for 1,000 iterations without reaching an encoding better than the best seen so far, and adopt that best encoding as the final one.

2.6 VARIABLE ORDER MARKOV MODELS AND THEIR ESTIMATION

A Markov model of fixed order T on a state space S containing symbols $s \in S$ is usually represented through a $\#(S)^T$ by $\#(S)$ transition probability matrix, whose entries $p(s|s_{-1}...s_{-T})$ express the chances of s conditional to the symbols in the T preceding positions. An alternative and more intuitive way of representing Markov

models is through a tree structure; each node in the tree correspond to a context of a given length, say a, b of length 2, and contains transition probabilities $p(s|b, a)$. The children of such node correspond to contexts extended forward by one symbol, say a, b, c, and contain transition probabilities $p(s|c, b, a)$. A tree comprising all contexts up to length T contains in its leaf nodes all the transition probabilities required to specify a Markov model of fixed order T. A variable order Markov model (VOMM) of maximal order T can be thought of as a "pruned" version of such a tree, where a reduced number of leaf nodes correspond to contexts of variable lengths with distinct transition probabilities.

Fitting a VOMM on training data consists of extending contexts and estimating the corresponding transition probabilities. We extend contexts using a pruning criterion; considering each order t from 0 to T, we augment the tree to include a node for each context $s_{-t}...s_{-1}$ that occurs more than p (e.g. 10) times in both the positive and the negative training sets. While this criterion is naïve compared to other VOMM pruning strategies, it does not require the maximal model (where all contexts are considered) to be built before pruning, and thus allows quicker model fitting. For each node included in the tree, we then need to compute the transition probabilities $p(s|s_{-1}..s_{-t})$. Of course a node may not have a full set of children, and there may even be extended contexts $s_{-t}...s_{-1}, s$ that never occur in the data. To produce non-zero estimates for the corresponding probabilities, we use a "discount" smoothing rule, which redistributes a small amount of mass d (e.g. 0.01) through the formula:

$$p(s|s_{-1}...s_{-t}) = (1-d)\frac{\#(s|s_{-1}...s_{-t})}{\sum_{\tilde{s}\in X}\#(\tilde{s}|s_{-1}...s_{-t})} + dp(s|s_{-1}...s{-}(t-1))$$

where $\#(\cdot|s_{-1}...s_{-t})$ indicates number of occurrences after $s_{-t}...s_{-1}$ (in other words, the rule reallocates d mass relative to the distribution of the parent context $s_{-(t-1)}...s_{-1}$). For order zero (empty context) we set $d = 0$.

Note that the maximal order is a hard limit on the size of a VOMM, since contexts cannot extend beyond T. Pruning also limits the size of the model, as it determines how many transition probabilities need to be estimated. Preliminary investigations

showed that our fits are robust to changes in p and d, at least for relatively small values of these parameters.

2.7 LOG-ODDS CLASSIFICATION

For classification, we fit two variable-order Markov models on the positive and negative training sets, as described in section 2.6. Any training or independent alignment segment, say $a = (a_1 \ldots a_n)$ comprising n columns, is then scored with the equation:

$$\mathcal{L}(a) = \sum_{i \in (1 \ldots n)} \log \left(\frac{p_{\mathrm{POS}}(a_i | a_{\mathrm{POS}}^{(i-)})}{p_{\mathrm{NEG}}(a_i | a_{\mathrm{NEG}}^{(i-)})} \right)$$

where $a_{\mathrm{POS}}^{(i-)}$ and $a_{\mathrm{POS}}^{(i-)}$ represent the relevant contexts (symbols in position $i-1$, $i-2$, ...) under the positive and negative model. $\mathcal{L}(a)$ is positive if the patterns in a resemble those characteristic of the positive training data, and negative if the resemblance is to the negative training data, so the segment can be classified by the sign of its score.

2.8 EVALUATION OF ENCODINGS THROUGH CROSS VALIDATION

To evaluate the classification performance of an encoding during the iterative search, we use k-repeated h-fold cross validation. The training data is partitioned at random into h (e.g. 10) folds, a fold is withheld, and two VOMMs are estimated with the remaining positive and negative data. The estimated models are used to produce log-odds scores for all the data (including the withheld fold). If the sets of scores for positive and negative data used in training overlap, withheld data is classified into positive and negative based on the sign of their scores. If the sets do not overlap, the withheld data is classified as positive if their score is larger than the minimum score of the positive data, as negative if it is smaller than the maximum score of the negative data, and as "unclassifiable" if it falls in between. This yields counts of correctly classified, erroneously classified, and unclassifiable elements in the withheld fold. The process is repeated for the h folds, and for k (e.g. 10) random partitions of the

data. Counts are averaged in correct classification (success), erroneous classification, and unclassifiable rates associated with the alphabet.

Unlike the success rates used to evaluate encodings during the search, those reported in outcome of ESPERR applications reported below, i.e. the success rates obtained on optimal encodings, are recomputed with leave-one-out cross validation for stability (instead of withholding folds, the data elements are withheld one at a time).

2.9 IMPLEMENTATION DETAILS

The ancestral base inference, agglomerative clustering, and iterative search were implemented in Python with performance-critical portions implemented in C – these include code for estimating VOMMs and scoring alignment segments, which are run to perform cross-validation over thousands of candidate encodings. The simple pruning and smoothing rules used in VOMM estimation are amenable to efficient implementation, making the iterative search tractable. The search can be spread over multiple cluster nodes using MPI. Running time for the search varies depending on specific data and the random component; for the RP application (chapter 3) convergence is generally achieved in $\sim 10,000$ iterations, requiring a day on a 2Ghz Athlon machine – substantially less on a small cluster.

CHAPTER 3

REGULATORY POTENTIAL SCORES

Despite years of intense study, *cis*-regulatory elements remain difficult to predict. In previous work (Elnitski et al., 2003; Kolbe et al., 2004) it has been shown that using Markov models to capture patterns in encoded genomic alignments can be effective for discriminating these regions from ancestral repeats – a model for likely neutral regions (Waterston et al., 2002). Applying ESPERR to learn an encoding for this problem yields a substantial improvement in discrimination over previous methods.

3.1 LEARNING REGULATORY POTENTIAL WITH ESPERR

To discriminate regulatory elements from ancestral repeats with ESPERR we defined two training sets. The positive training data consists of a set of 97 experimentally validated regulatory elements (Elnitski et al., 2003), and the negative training data of repetitive elements that stabilized before the divergence of human, mouse, and dog. Alignments of seven species (human, chimpanzee, macaque, mouse, rat, cow, and dog) corresponding to the training regions were extracted from the UCSC Genome Browser (Karolchik et al., 2003). To improve the resolution of our cross-validation procedure, these alignments were chopped into 100-column pieces, and the ancestral repeats were randomly sampled to produce a training set equal in size to the positive set. We allowed alignment columns to be considered if they had no more than three missing species, and required each 100-column segment to have at least 50 such columns. This resulted in positive and negative training sets containing 357 elements, covering approximately 31,000 human bases each. ESPERR with a log-odds classifier based on VOMMs (with a maximal order of 2) yielded a final encoding into 17 symbols, with a leave-one-out cross-validation success rate of ~ 94% on the training data.

This performance is a considerable improvement over previous Regulatory Po-

tential scores (~ 82% for the scores of Kolbe et al. (2004), based on human-rodent alignments). Cumulative distributions of RP scores computed on the training sets and similarly prepared random samples of exonic and all ("bulk") genomic regions are show in Figure 3.1A. RP scores do an excellent job discriminating regulatory regions from bulk and neutral DNA, as well as separating them from exons.

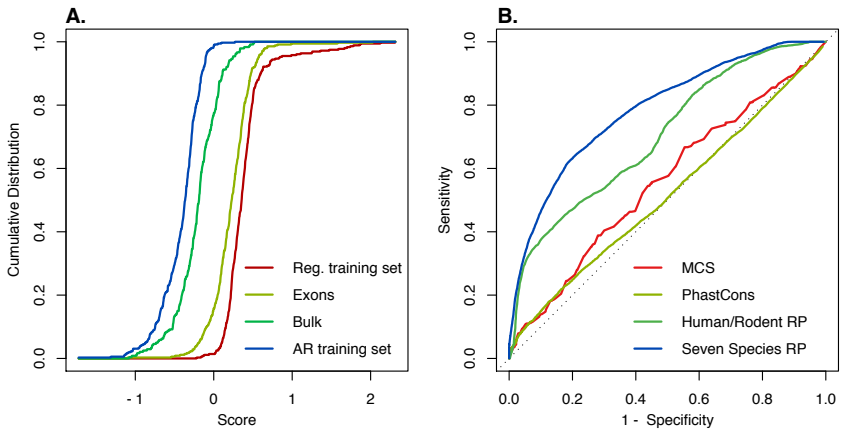

Figure 3.1: RP score performance demonstrated by (A) cumulative distributions of scores on various genomic elements and (B) ROC plots for discrimination of 23 elements in the human beta-globin locus.

As an additional evaluation of RP performance, we considered 23 experimentally confirmed regulatory elements in the hemoglobin beta gene cluster. These likely include most of the sequences with regulatory function for this extensively studied locus, and only five are part of our regulatory training set – providing reasonably exhaustive and independent test data for sensitivity and specificity assessments (King et al., 2005). The ROC plot (Figure 3.1B) shows that performance of the ESPERR-based RP scores on this dataset, in terms of both sensitivity and specificity, is uniformly better than previous RP scores (from human-rodent alignments) and two conservation scores: phastCons (Siepel et al., 2005) and MCS (Margulies et al., 2003). Figure 3.2 shows a portion of the beta globin locus control region. Known regulatory elements are shown along with the ESPERR based RP score and the previous human/rodent RP scores of Kolbe et al. The ESPERR based score clearly identifies more regulatory elements, in particular HS3.1, HS3.2, and HS4 were all missed by the older score.

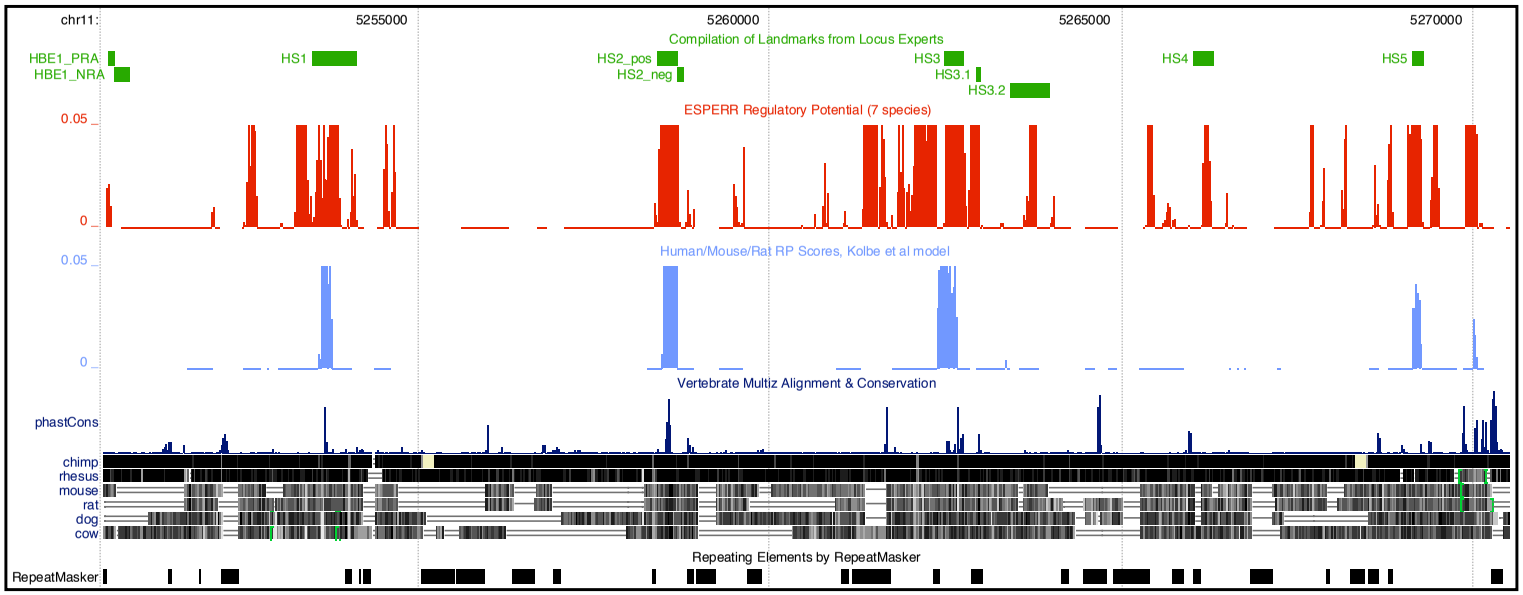

Figure 3.2: UCSC browser image for the hemoglobin beta gene cluster locus control region. Known regulatory elements (landmarks track) are shown in green, ESPERR RP scores in red, Human/rodent RP scores from Kolbe et al. (2004) in blue, as well the conservation and repeat tracks.

3.2 ESPERR CAPTURES A VARIETY OF SIGNALS IN REGULATORY ELEMENTS

To begin unraveling the signals that contribute to the excellent performance of the ESPERR based RP score, we must examine the variability structure in the training data, and how this structure relates to the score. We would like to understand which features of the training data lead to good performance, but this is a challenging prospect given that a very large number of alignment columns are grouped together by ESPERR in each symbol of the reduced representation. Because RP is a log-odds score based on VOMMs with maximal order 2, we consider the frequencies of words of length 3 in the training data after applying the encoding learned by ESPERR. One approach for understanding the variability structure of a dataset is principal component analysis (PCA), which finds a transformation of a dataset to a new coordinate system in which the first component has the greatest variance, the second (orthogonal to the first) and has the next greatest variance, and so on. Applying PCA to these word frequencies shows that a large amount of their variability is explained by the first few principal components (Figure 3.3 top panel). However a substantial amount of variability is spread across the many remaining components, consistent with the presence of both strong and weak signals in this dataset.

Our first insight into the nature of the strong signals comes from our analysis of the performance of RP scores. We note that while RP can discriminate regulatory elements better than conservation scores can, exons also have very high RP values (Figure 3.1A). Two signals often associated with exons (as well as regulatory elements) are conservation and GC content. Computing a regression of RP score on GC content and conservation (measured as the average phastCons score) shows that these two quantities alone explain $\sim$ 68% of the variability in RP. Another factor typically associated with ubiquitous promoter regions is CpG dinucleotide density (Cooper et al., 2006); however, while this does explain some within-class variability of the regulatory elements in our training set, it does not significantly improve our ability to explain RP (when CpG density is added to the regression its coefficient is not significant, and the correlation is almost unchanged).

Pinpointing the nature of other factors that systematically contribute to RP is

complicated, due to the enormous reduction induced by our encoding, and the random component involved in the search algorithm. Nevertheless, these factors are crucial for discrimination; about a third of RP is likely a composite of weaker signals. A practical way to measure this composite is to consider the residuals from the regression of RP on GC content and conservation, which we will denote as $F = RP - (\hat{\beta}_0 + \hat{\beta}_1 \cdot GC + \hat{\beta}_2 \cdot Cons)$, where $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ are the coefficients from the RP score regression. The bottom panel of Figure 3.3 shows the correlation of RP and each of these three quantities with the first 25 principal components. We see that the strongest component that has high correlation with RP also has high correlation with conservation and GC content. However RP also shows substantial correlation with many of the weaker components, which are less exclusively dominated by the strong conservation and GC content signals.

To further explore the difference between these strong signals, and the composite of weak, subtler signals represented by F, we correlate each of these three quantities with individual word frequencies in the training data. Figure 3.4 (bottom panel) shows box-plots of these correlations. The positive correlation with both conservation and GC content are dominated by a small number of words, which are the outliers at the top of the distribution. In contrast, F shows far fewer dominant outliers and is associated with many different words. Further insight into the nature of these signals is obtained by examining the specific words that have the strongest positive correlation with each feature. Figure 3.4 (top panel) shows "logos" for the words most strongly correlated with each signal (the height of each character in the logo is determined by the ancestral probability distribution centroid for the columns encoded to that symbol). Again, conservation and GC content are dominated by words clearly associated with these signals (the search procedure has grouped fully conserved C and G columns together, so the symbol with strong G and C components shows up frequently in the highly conserved set). The words most strongly associated with F on the other hand are more diverse, consistent with indications that a variety of different patterns contribute to F.

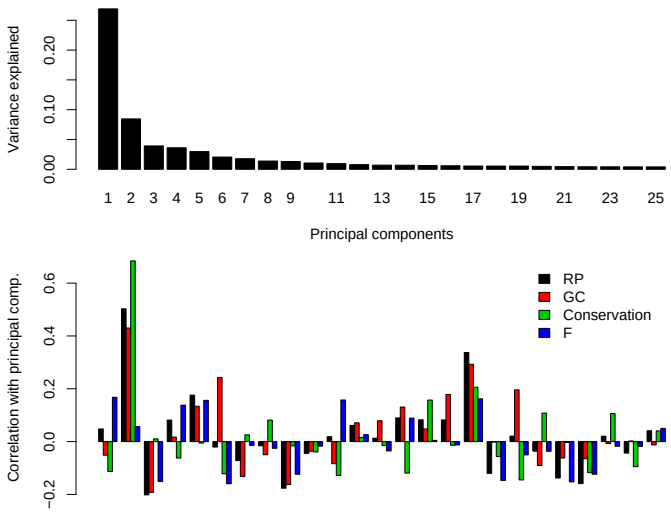

Figure 3.3: Share of variance explained by each of the first 25 principal components of the RP training data word frequencies (top) and correlation of RP score, GC content, conservation, and the residuals *F* with each principal component (bottom).

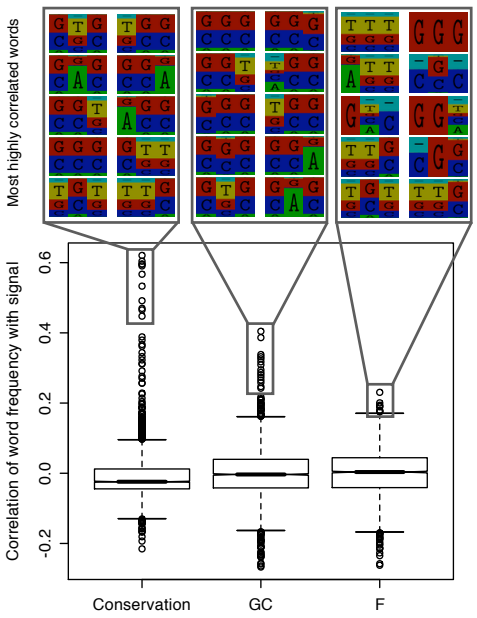

Figure 3.4: Distributions of the correlations between word frequencies in the RP training data and three component signals (GC content, conservation, and the residuals F). For each signal, representative logos of the most strongly correlated words are shown.

3.3 RP WEAK COMPONENTS HELP TO IDENTIFY TRULY DISTAL REGULATORY ELEMENTS

Distal *cis*-regulatory elements – those that are far from the start site of any gene – have proven particularly difficult to identify. The ENCODE Transcriptional Regulation group (Weng et al., 2006) used ChIP-chip to identify binding sites for a variety of transcription factors and other DNA binding proteins. We selected a subset of their experiments, emphasizing experimental platforms with high-resolution site identification and sequence-specific binding not exclusively associated with transcription start sites. We eliminated all sites overlapping repetitive regions or coding exons, and expanded the remaining sites to cover at least 100bp. Next, we restricted attention to sites supported by at least one additional line of experimental evidence suggesting regulation, such as ChIP-chip evidence for certain histone modifications associated with activation or factors associated with general chromatin modification[1], as well as DNaseI hypersensitivity and nucleosome depletion[2] (Stamatoyannopoulis et al., 2006). Finally, to focus on distal regulation, we removed sites falling within 2.5kb of a transcription start site (Guigó et al., 2006). This resulted is a set of 617 elements with multiple lines of evidence suggesting a distal regulatory function. Aggregate characteristics of these regions suggest that they are enriched for function; in particular, they show evidence of evolutionary constraint both in terms of average phastCons scores (Siepel et al., 2005) and in terms of overlap with evolutionarily constrained regions (moderate MCS set from Margulies et al., 2006). This set may also contain some non-functional elements, as well as unannotated promoters and proximal elements, because there are likely transcription start sites that have not been identified (Guigó et al., 2006; Weng et al., 2006).

Sufficient aligning sequence was available to calculate the RP score, GC content, average phastCons score, and F for 583 of the 617 elements in our set. For each of

[1] Histones are the major proteins that DNA is packed around in its packed form (chromatin). Certain changes to the histones have been found to be associated with the activation of transcription.

[2] In packed chromatin, histones and DNA are arranged into larger structures called nucleosomes. Nucleosomes prevent polymerase from accessing the promoter regions of genes. Thus absence of nucleosomes is another marker for activation.

the three RP components (GC, conservation and F) we examined the 50 highest scoring elements. Among those with high GC content we see a strong enrichment for possible unannotated promoters: 21 overlap an ENCODE ChIP-chip binding site for factors associated with transcription initiation (PolII, Taf250, TFIIB). Elements with high conservation also appear to contain possible unannotated promoters, with 12 regions overlapping such a binding site. In contrast, among the 50 putative distal elements with the highest F, only three overlap such a binding site. This suggests that, although strong signals such as GC content and constraint are still likely to play a role, true distal elements may be better characterized by the subtler, weaker signals proxied by F.

3.4 DATA PREPARATION

RP training sets were prepared using the 17-species Multiz alignments from the UCSC genome browser (Karolchik et al., 2003; Blanchette et al., 2004). We used the subset of mammalian species in these alignments with higher sequence quality; namely human (hg17), chimpanzee (panTro1), macaque (rheMac2), mouse (mm7), rat (rn3), dog (canFam2), and cow (bosTau2). Alignments corresponding to each element of a training set were extracted. Gaps between alignment blocks were annotated as such, and all other gaps (including complex insertion/deletion events) were annotated as missing data. The training sets were then chopped to 100 column alignment segments. Training elements were also required to have at least 50 good alignment columns (those having three or fewer missing species) to be included.

CHAPTER 4

APPLICATION TO OTHER PROBLEMS

Next we consider the application of ESPERR to two other pairwise classification problems. In the computation of RP scores, we discriminate experimentally validated regulatory regions from a neutral model – these two training sets represent extremes relative to the entire genome. In the following two examples we instead construct positive and negative training sets from the result of experimental assays which were applied comprehensively to a set of regions. Specifically, for the first example we construct a training set from a comprehensive evaluation of DNaseI hypersensitivity in the ENCODE regions. For the second, the training set is constructed from highly conserved elements tested for developmental enhancer activity. Thus the overall composition for these training sets is different from that used to train RP. The DNAseI hypersensitivity training set as a whole is representative of the genome since the assay was performed comprehensively. In contrast, both the positive and negative training sets for distinguishing developmental enhancers are drawn from the most highly conserved regions of the genome. ESPERR performs well for both of these problems, indicating that our method is able to effectively learn encodings for a variety of problems.

4.1 DISCRIMINATING ENCODE DNASEI HYPERSENSITIVE SITES WITH ESPERR

Sensitivity to cutting by the non-specific endonuclease DNaseI in vivo can be used to identify local disruptions in the DNA chromatin packaging structure. These disruptions, which span sequences of length ~ 250bp, are reliable markers for regions that are functional in the nucleus, including transcriptional regulatory elements. High-throughput quantitative chromatin profiling techniques (Sabo et al., 2004; Dorschner et al., 2004) allow the identification of large numbers of DNaseI hypersensitive sites in specific cell lines (e.g. K562 erythroid cells).

The ENCODE chromatin and chromosomes group (Stamatoyannopoulis et al., 2006) has assayed a large portion of the ENCODE regions in several cell lines for hypersensitivity to DNaseI. From their data we extracted a set of high-confidence positive calls (empirical p-value < .001 and plate quality > 0.5 in any cell line; 369 elements), and high-confidence negative calls (empirical p-value > .1 for all cell lines with plate quality > 0.5, and no overlap with other ENCODE functional elements as compiled in (Margulies et al., 2006); 477 elements). Prior work on predicting DNaseI hypersensitive sites with a linear support vector machine based on short motifs in the primary genomic sequence (length 1-6, ignoring strand) showed good performance (Noble et al., 2005). Using their methods and training data we were able to confirm their reported success rate of ~85%. However applying this approach to the ENCODE dataset achieves a success rate of ~64%, suggesting that this more comprehensive set of sites is substantially more difficult to discriminate.

We applied ESPERR to this dataset, using the same seven-species alignments as for the RP scores. Training data consisted of 319 positive elements and 379 negative elements with sufficient alignments, prepared as was done for RP scores (except that these elements were not chopped to 100 column segments since the training sets are larger and the elements are of less variable length). The procedure identified an encoding to 18 symbols, which achieved a success rate of ~80%. Thus, for this more comprehensive dataset, the additional information available in multiple alignments and captured by ESPERR achieves substantially better performance than does a linear SVM using sequence motifs.

4.2 Identifying conserved regions with developmental enhancer activity

It has been observed that many of the highly conserved non-coding regions of the genome are transcriptional regulatory enhancers common to vertebrates. For example, Woolfe et al. (2005) tested regions conserved between human and pufferfish (*F. ruprides*). Of 25 elements tested (all near four developmental genes), 23 showed significant enhancer activity in at least one tissue. Thus, screening conserved elements

appears to be an excellent approach for experimentally identifying more enhancer elements. However, computational methods could improve the efficiency of these screens by predicting which elements are more likely to show activity in a given experiment.

The VISTA Enhancer Browser (`http://enhancer.lbl.gov`) contains 253 conserved regions that have been tested for consistent enhancer activity in transgenic mouse embryos. A region was declared positive (validated) if at least three embryos showed the same pattern of expression for that element. Here, ESPERR produces a score to predict which of the numerous other conserved regions in the genome would be validated by this assay. For positive and negative training sets we used 108 validated and 138 non-validated regions (a small number of regions with ambiguous results were excluded).

Because both the positive and negative training sets for this problem consist of highly conserved elements, alignments spanning a much deeper evolutionary tree were used as compared to the previous applications. Specifically, we used alignments of human, mouse, opossum, chicken, frog, zebrafish, and pufferfish. Training elements were not chopped, and alignment columns with at most three missing species were considered valid, with at least 50 such columns required for an element to be used, resulting in a positive set of 108 elements covering 143,688 human bases and a negative set of 134 elements covering 165,272 human bases. ESPERR identified an encoding to 15 symbols, and yielded a very good cross-validation success rate of ~83%. Thus, using our method to score conserved elements for potential enhancer activity could greatly increase the rate of discovery and validation of new conserved embryonic enhancers.

4.3 DATA PREPARATION

Training sets were again prepared using the 17-species Multiz alignments from the UCSC genome browser (Karolchik et al., 2003; Blanchette et al., 2004). For the computation of hypersensitive site predictions we used the same subset of mammalian species used in RP scores; namely human (hg17), chimpanzee (panTro1), macaque

(rheMac2), mouse (mm7), rat (rn3), dog (canfam2), and cow (bosTau2). For prediction of highly conserved regions with embryonic enhancer activity we used a subset of species spanning a larger evolutionary distance; namely human (hg17), mouse (mm7), opossum (monDom2), chicken (galGal2), frog (xenTro1), zebrafish (danRer3), and pufferfish (fr1). Alignments corresponding to each element of a training set were extracted. Gaps between alignment blocks were annotated as such, and all other gaps (including complex insertion/deletion events) were annotated as missing data. For both applications, training elements were also required to have at least 50 good alignment columns (those having three or fewer missing species) to be included.

CHAPTER 5

PREDICTING PROMOTER ACTIVITY WITH ESPERR

A transcriptional promoter, by definition, is the genomic segment required for initiation of transcription. Increased precision of the definition of a promoter can be attained by tying it to a specific experimental protocol. Here we focus on computational prediction of the promoter data generated by Cooper et al. (2006), who identified 921 potential promoters based on full length cDNA libraries. A cDNA is generated by reverse transcribing (copying back into DNA) a mature mRNA, these sequences are thus good evidence for the presence of a gene. Potential promoters were evaluated in 16 diverse cell lines by a transient transfection assay, in which a construct is made containing the tested region followed by a reporter gene in a piece of circular DNA called a *plasmid*, which is then introduced into the cell. If the promoter is active in that cell the reporter gene is expressed. Usually the reporter gene codes for a luciferase protein that produces light when expressed, which can then be used to measure the expression level.

We thus address the following problem: given the genomic position of the 5′ end of a putative full-length human cDNA, predict whether transient transfection reporter assays will determine that the surrounding genomic sequence (1) has promoter activity in almost all human tissues (i.e., is a ubiquitous promoter), (2) has promoter activity in just a few tissues (tissue-specific promoter), or (3) has no promoter activity.

Several kinds of genomic data are potentially useful for predicting promoter activity. It is well known that ubiquitous promoters are frequently associated with a high density of CpG dinucleotides, and some predictive success can be attained using this association (e.g. Down and Hubbard, 2002; Bajic and Seah, 2003). The presence of a TATA box (a sequence motif thought to be associated with transcription initiation) is potentially informative, but the Stanford team reported that only 16% of the functional promoters contain one (Cooper et al., 2006). Available promoter

predictors, which typically rely heavily on the presence of CpGs and/or TATA boxes, have limited overall success with the Stanford data. For instance, promoters predicted by FirstEF (Davuluri et al., 2001) overlap 87% (92 of 106) of the ubiquitous Stanford promoters but only 11% (14 of 130) of the tissue-specific promoters. Similarly, only 45% of ubiquitous and 5% (7 of 130) of specific Stanford promoters are within 500bp of a transcription start site predicted by Eponine (Down and Hubbard, 2002). Evolutionary constraint has been measured in promoters and found to have only slight predictive value (Margulies et al., 2006; Weng et al., 2006). For example, Cooper et al. (2006) report that the fraction of conserved bases is very similar between functional and non-functional promoters – 12.5% vs. 10%. Another potential data source for promoter prediction are the genomic locations of 5′ ends of transcripts (Liu and States, 2002), which are employed here. Finally, a variety of data showing associations with transcriptional promoters, such as ChIP-chip data on binding by PolII or TAF1, are available for the ENCODE regions. However, this study is limited to data that are currently available genome-wide.

Here we extend ESPERR to predict the activity of potential promoter regions using genomic positions of the 5′ ends of human putatively full-length cDNAs and genome wide multiple alignments of human with four other mammals – chimpanzee, mouse, rat, and dog – obtained from the UCSC Genome Browser (Kent et al., 2002; Karolchik et al., 2003). We compare the performance of ESPERR with other machine learning methods trained on additional signals computed from primary sequence and alignments: G+C content and CpG frequencies in arbitrary genomic intervals, and genomic locations of regions predicted to be under purifying selection, and the average phastCons score for arbitrary intervals (Siepel et al., 2005).

For training data, we used the Stanford results to label each assayed 5′ end position as a ubiquitous, tissue-specific or nonfunctional promoter. There are a variety of computer techniques that can combine these sources of data to classify 5′ ends according to promoter activity; we tried linear discriminant analysis (see for instance Seber, 1984), support vector machines (Cortes and Vapnik, 1995), classification trees (Breiman et al., 1984) and a multi-class generalization of the ESPERR approach. Our

results show that ESPERR significantly out-performs the other approaches, presumably because it is not limited to pre-defined signals, but can extract the sequence and evolutionary patterns relevant to a particular problem from the training data.

5.1 Assessment of promoter activity in the ENCODE regions

Cooper et al. (2006) identified 921 potential promoters based on full-length cDNA libraries. Of these they tested all those associated with multi-exon transcripts (528) and a sample of those associated with single exon transcripts (114) in 16 diverse cell lines using transient transfection reporter assays. They additionally assayed 102 negative control fragments and declared a tested promoter fragment as functional in a given cell line if it showed significant activity relative to negative controls.

The criteria used by Cooper et al. for declaring positive fragments are stringent, although fragments not declared positive by this standard might still have substantial activity. For labeling of negative fragments we also need stringent criteria for absence of activity. Thus we declare each tested promoter fragment as positive in a cell line if the activity is more than four standard deviations over the mean of the negative controls, negative if the activity is less than one standard deviation over the mean, and ambiguous otherwise. We then group fragments into three classes: "ubiquitous" having positive calls in all 16 cell lines (106 fragments), "specific" with positive calls in only one to five cell lines (130 fragments), and "negative" with negative calls in all 16 cell lines (123 fragments). The remaining 304 fragments assayed by the Stanford group, which could not be clearly assigned to one of these stringently defined classes, were not used for this analysis.

5.2 Signals that distinguish different classes of promoters

To distinguish these three classes of promoters we first considered three signals: density of CpG dinucleotides, GC content, and multi-species conservation. Conservation was measured both as the fraction of non-coding bases overlapping a highly constrained region (moderate MCS Margulies et al., 2006) and as the average non-coding phastCons score (Siepel et al., 2005), which can be computed for all

intervals with alignments, not just those with strong constraint. Figure 5.1 shows the distributions of these signals across the three classes of tested regions. Clearly these signals vary among classes, and in particular CpG content is very strongly associated with ubiquitous promoters.

We evaluated the predictive power of each of these signals (Table 5.1, left columns). Performance was evaluated using leave-one-out cross validation. Each element in the training data is withheld, a threshold on the score is found that best classifies the remaining elements, and the withheld element is classified according to that threshold. The success rate is the percentage of elements classified correctly. As expected, CpG density does a very good job of separating ubiquitous promoters from the other two classes (over 90% success rate). Additionally, GC content does quite well for separating ubiquitous from negative promoters (over 80% success rate). Otherwise these signals discriminate fairly poorly, with all other success rates less than 65%. In particular all of these signals do a very poor job of distinguishing tissue specific promoters from negative regions.

As an alternative to classification based on predefined signals computed from sequences or alignments, we can instead attempt to learn the relevant signals for classification from training data using ESPERR. Training data was extracted from Multiz alignments of chimpanzee (panTro1), mouse (mm6), rat (rn3), and dog (canFam1) to the human genome. Positions overlapping coding sequences were eliminated from the training data. We allowed any column in the training data with at most two missing species to be used. Handling missing data in this way allowed us to cover most potential promoter regions, however a small number of training regions (4

Datasets	**phastCons**	**MCS overlap**	**GC**	**CpG**	**ESPERR**
Ubiquitous vs. Negative	54.15%	61.14%	80.79%	90.83%	96.31%
Ubiquitous vs. Specific	46.19%	53.81%	64.41%	90.68%	96.21%
Specific vs. Negative	52.96%	60.08%	63.24%	58.50%	83.81%

Table 5.1. Pair-wise classification success rates using quantities computed from genomic sequence (GC content and CpG density), alignments (phastCons and MCS overlap), and ESPERR.

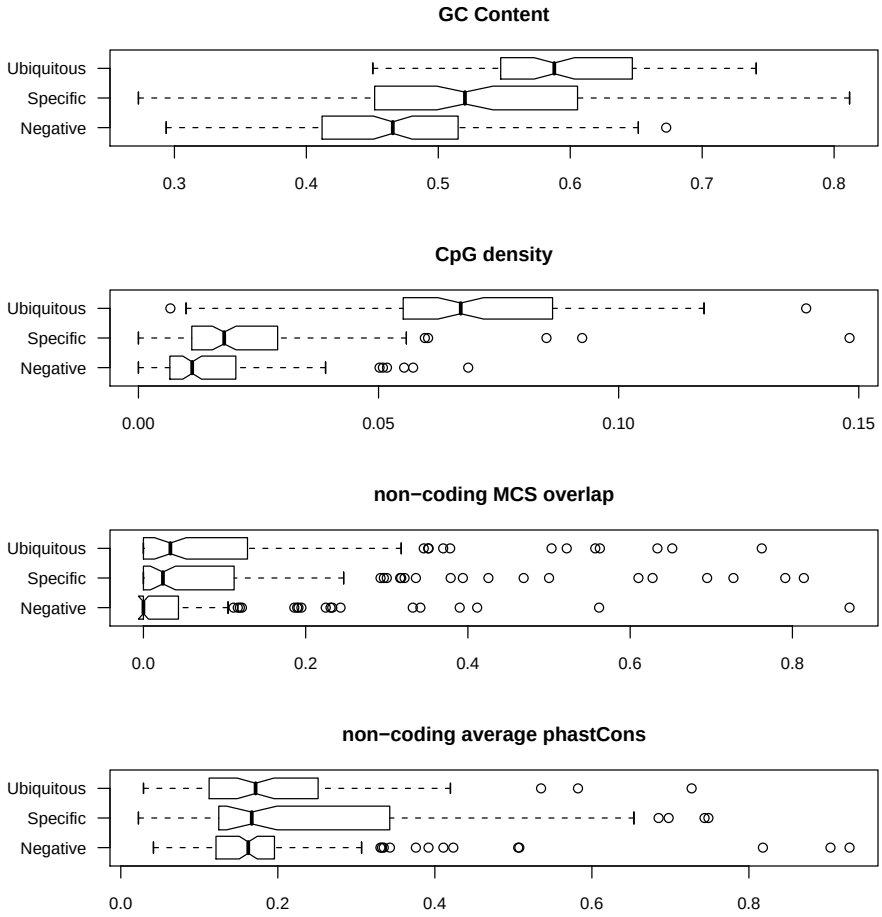

Figure 5.1: Distributions of (A) GC content, (B) CpG density, (C) non-coding phastCons and (D) non-coding MCS overlap for widely expressed, tissue specific, and negative regions.

ubiquitous, 12 specific, 12 negative) were eliminated from the analysis because they did not have at least 50 valid alignment columns.

Applying ESPERR to the pair-wise classification of ENCODE promoter regions is very effective (Table 5.1, right column). In addition to performing better than CpG density for classification of ubiquitous promoters, this method is the first to show good performance for discriminating specific from negative promoters, having a success rate of ~84%.

5.3 EXTENSION TO SIMULTANEOUS THREE-WAY CLASSIFICATION

To effectively identify both tissue specific and widely expressed promoters across the human genome, pair-wise discrimination is insufficient. We would instead like to be able to predict for a given element which of the three classes it most likely belongs to. This problem is even more challenging than distinguishing two classes, especially given that the signals do not separate the classes very cleanly, since there are now more ways to misclassify each element.

A variety of machine learning methods exist for multi-way classification, and we have evaluated the performance of several on this problem (Table 5.2). We first considered linear discriminant analysis (LDA) using each of the predefined signals independently as predictors, as well as combinations of the signals. At best, using CpG density, GC content, and MCS overlap as predictors, LDA achieved a leave-one-out cross validation success rate of ~66%. Because the effectiveness of the different predictor signals for classification varies substantially depending on which classes are considered, we next attempted to use classification trees (Breiman et al., 1984). Several combinations of predictors were again attempted, and the best performance was again achieved combining CpG density, GC content, and MCS overlap, giving a leave-one-out cross validation success rate of ~63%. Finally, because the classes are not easily linearly separable we tried to discriminate them using a support vector machine, achieving a 10-fold cross validation success rate of ~69%. For SVM 10-fold cross validation was used rather than leave-one-out for this portion of the analysis due to the computational cost of parameter optimization.

ESPERR can also be generalized to multi-way classification. Rather than producing a log-odds score for two probability models, a model (VOMM) is trained for each class and test segments classified according to the model under which they have the highest probability. Using the correct classification rate under this scheme as the new "figure of merit" to evaluate candidate encodings in the heuristic search allows selection of an encoding optimized for multi-way classification. On our 3-class problem this approach yields a much better performing classifier than any of the other methods evaluated, with a final success rate of ~81%.

5.4 PREDICTING WIDELY EXPRESSED AND TISSUE SPECIFIC PROMOTERS GENOME WIDE

While our approach relies on ENCODE experimental data for training, for prediction it only requires genomic alignments. This allows us to predict promoter activity in regions immediately upstream of existing cDNAs in the entire human genome. Following the approach of Cooper et al. we used alignments of Genbank cDNAs to the human genome to identify 79,616 possible 500bp promoter regions associated with 36,416 gene models (clusters of cDNAs with exonic overlap). From these candidates we predicted 19,239 widely expressed promoters (of which 13,967 overlap CpG islands) and 23,315 tissue specific promoters. We also predict that 29,988 of the candidate regions would not be active in the transient luciferase assay, while 7,704 could not be reliably classified due to insufficient alignment data. Among our gene models, 14,692 have a predicted ubiquitous and 15,500 have a predicted specific promoter. Interestingly, 6,584 models have both a ubiquitous a and specific promoter. Predicted ubiquitous promoters have the strongest association with existing gene annotations: 61% of our predictions are within 500bp of a RefSeq annotated start site. Predicted specific promoters coincide less frequently – 20% – consistent with lower quality

Method (predictors)	**Performance**
LDA (MCS)	39.83%
LDA (phastCons)	34.09%
LDA (GC)	48.60%
LDA (GC, CpG)	65.46%
LDA (MCS, GC, CpG)	66.85%
LDA (phastCons, GC, CpG)	65.06%
Tree (GC, CpG)	57.94%
Tree (phastCons, GC, CpG)	63.07%
Tree (MCS, GC, CpG)	63.23%
SVM (MCS, gc, cpg)	63.83%
ESPERR	80.98%

Table 5.2. Multi-way classification success rates using several machine learning methods and predictors: Linear discriminant analysis (LDA), classfication trees (Tree), support vector machines (SVM), and ESPERR.

annotation for tissue specific transcripts. The predicted negative promoters show by far the least associated with RefSeq start sites, with only 12% falling within 500bp.

Inspection of the predictions for several well-studied tissue-specific and ubiquitous promoters shows that they are accurate in these cases. An example of a tissue-specific gene is *NFE2*, which encodes the erythroid subunit of the protein NF-E2 that binds to strong enhancers such as hypersensitive site 2 of the beta-globin locus control region. Two different mRNAs (L13974 and S77763, Figure 5.2) arise from different promoters and encode the same protein, but they are differentially synthesized during development (Ney et al., 1993; Pischedda et al., 1995). Both of these are correctly predicted as tissue-specific promoters by our method. A third promoter is inferred from large scale cDNA sequences (e.g. mRNA CR450284), but no further information is available on it.

Another example of a tissue-specific promoter that is correctly predicted is that of *ZFPM1*. This gene encodes the protein FOG-1 (Friend of GATA-1), a multi-zinc-finger protein that interacts specifically with the erythroid transcription factor GATA-1 and other proteins (Tsang et al., 1997). Although the promoter is within a CpG island, the gene is expressed primarily in hematopoietic tissues. Most predictors would classify this as a ubiquitous promoter because it is a CpG island, but our method correctly predicts it as being tissue-specific (Figure 5.3).

An example of a gene with ubiquitous and tissue-specific promoters that are correctly predicted is *POU2F1* which encodes PO2F1, also known as OCT1, an important transcriptional regulator. The transcript beginning at the most 5' exon (Figure 5.4) is ubiquitously expressed, associated with a CpG island, and correctly predicted by our method. However a second isoform initiating ~108kb downstream exhibits tissue specific expression, predominantly in B cells, in activated T cells, and in the nervous system (Luchina et al., 2003). This highly tissue specific promoter is also correctly identified by our method.

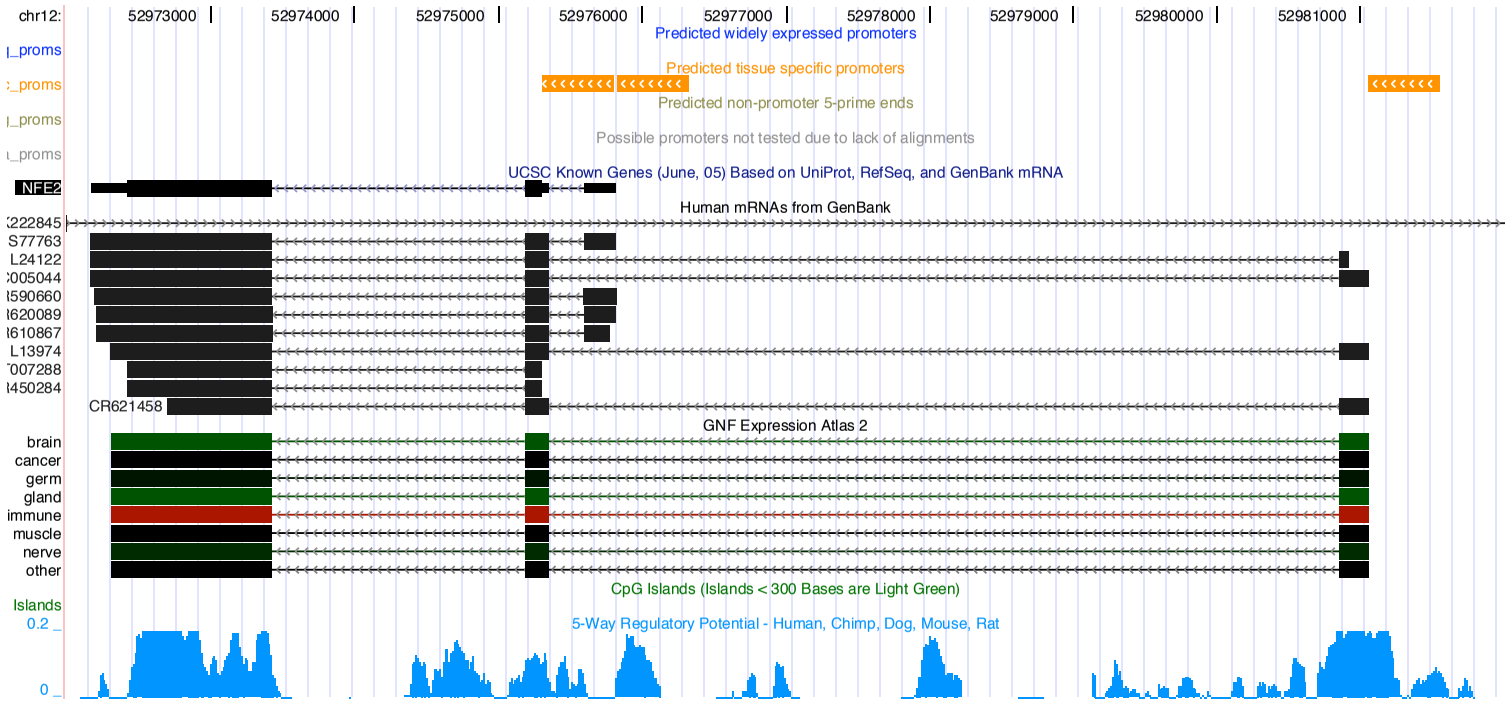

Figure 5.2: UCSC genome browser snapshot of promoter predictions in the neighborhood of *NFE2*.

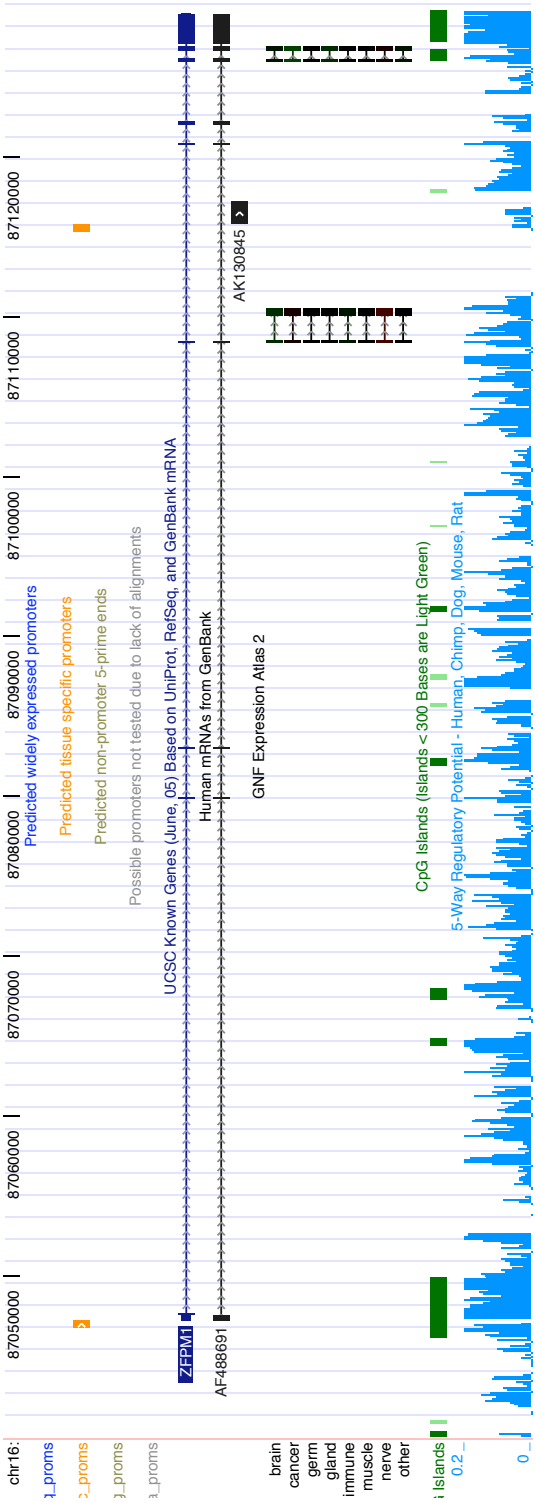

Figure 5.3: UCSC genome browser snapshot of promoter predictions in the neighborhood of *ZFPM1*.

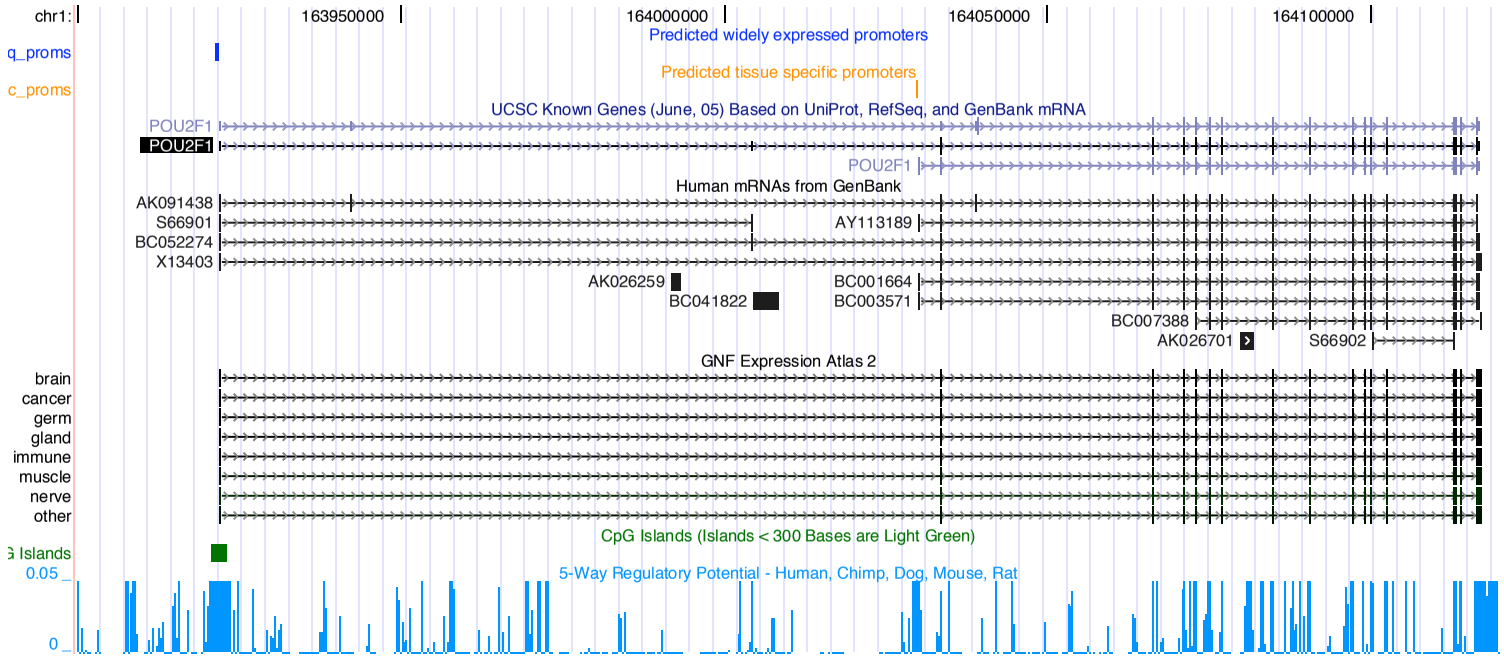

Figure 5.4: UCSC genome browser snapshot of promoter predictions in the neighborhood of *POU2F1*.

5.5 DATA PREPARATION AND METHODS

For all analysis restricted to the ENCODE regions, coding sequences were defined by Gencode gene annotations (Guigó et al., 2006). For genome-wide analysis, coding sequences were defined by UCSC knownGenes (Karolchik et al., 2003). MCS overlap for an interval was defined as the fraction of non-coding positions overlapping a moderate MCS (Margulies et al., 2006). Average phastCons scores were computed using the eight-species conservation track at the UCSC browser (Kent et al., 2002).

Linear discriminant analysis was performed using the R statistical software package (R Development Core Team, 2006). Classification tree analysis was performed using the RPART package for R, based on CART (Breiman et al., 1984). Performance for both schemes was evaluated with leave-one-out cross validation.

Support vector machine tests were performed using LIBSVM (Chang and Lin, 2001) which implements the "one-against-one" approach for multi-way classification (Hsu and Lin, 2002). Linear, Gaussian, and polynomial kernels were tested with a grid-search to select the best parameters.

Potential promoter regions for genome-wide predictions were identified following the methods of Cooper et al. (2006). The 200,825 Genbank cDNA Alignments from the UCSC browser human mRNA track were merged into 36,416 clusters of elements having any exonic overlap. Potential start sites were identified by taking the most 5′ base in each cluster and 500bp upstream. For clusters containing any multi-exon transcript we additionally required the first start site to be defined by a multi-exon transcript. Additional start sites were then identified for each cluster by progressively taking the next-most 5′ end at least 500bp downstream from the last, yielding a total of 79,616 potential promoters sites.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

We have presented ESPERR, a method to learn encodings of multiple alignments that retain useful information for a chosen classification problem. We have shown excellent performance for predicting three different types of functional elements, each of which involves a binary (e.g. positive versus negative) classification, as well as excellent performance in multi-way classification of potential promoter regions. We have also begun exploring the application of this method to gene prediction. Alignment encodings have already been used effectively to improve gene prediction; for example, TWINSCAN encodes positions as match, other aligned, and unaligned, and estimates models over the encoded sequence (Korf et al., 2001). ESPERR may be able to find effective encodings of multiple "informant" species in related gene prediction algorithms (e.g. N-SCAN, Gross and Brent, 2006). The first stage of ESPERR could also be employed to create reduced representations of multiple alignments for analyses where there is not a natural performance metric for driving the iterative search. One such application is the identification of encodings for the unsupervised characterization of alignments from highly conserved sequences (Bejerano et al., 2004).

ESPERR-based Regulatory Potential scores have proven effective for identifying enhancer elements. Wang et al. (2006) identified 75 regions having a positive RP score as well as matches to the binding site motif for the essential erythroid transcription factor GATA-1. They tested these regions with reporter gene assays in transiently transfected human K562 cells and/or after site-directed integration into murine erythroleukemia cells, and found that regions with high RP score were validated frequently (at least 50%), with even higher validation rates at higher RP scores. In contrast, segments with low RP tended to be inactive.

ESPERR is most appropriate when the loci in question are under selection among

the species examined, and at the very least requires that the loci can be aligned (though for all applications presented here we lose a small number of training sequence due to lack of sufficient alignment). For most applications, including those described in this work, elements do not necessarily exhibit strict nucleotide-level conservation. For example, binding sites in regulatory elements may change relative order or motif (Ludwig et al., 1998; Dermitzakis and Clark, 2002; Costas et al., 2003). Also, some elements may only be functional in a specific lineage – see for instance studies by Valverde-Garduno et al. (2004) on lineage-specific hypersensitive sites in the GATA1 locus in humans and mice. However, as long as the elements retain sufficient alignability, ESPERR can still achieve very good performance: in fact, our method can tolerate some degree of local change, and even capture such change if it occurs with a consistent pattern. In the future we would like to modify ESPERR to better capture rearrangements in functional elements, perhaps guided by alignments but allowing more flexibility in small scale rearrangement. This has the potential to greatly improve our ability to recognize elements containing these sorts of changes.

To infer the ancestral base distribution, ESPERR extends traditional nucleotide substitution models by treating gaps like a fifth nucleotide (see section 2.2). While this extension has been used effectively (McGuire et al., 2001), it is naïve, in that it treats indels affecting multiple consecutive positions as multiple independent events, and thus is overly sensitive to all but very short indels. Nonetheless, the extended HKY model works well in ESPERR, most likely because it is combined with a classifier that incorporates context and thus captures dependencies among neighboring sites. More sophisticated modeling of indels for ancestral distribution inferences would integrate naturally into our procedure, and we expect this to become more important as we apply ESPERR using other classifiers, as well as to unsupervised classification (clustering) problems.

All of the applications presented here are limited by the available training data and the experimental procedure that generated it; for instance, for promoter prediction we focus on predicting the result of transient transfection reporter assays. It remains unclear how accurately this assay recapitulates the regulatory behavior of promoter

fragments in their natural genomic context since the plasmids constructed for transient transfection lack distal elements, chromatin structure, and other epigenetic modifications. Because ESPERR was trained on data produced by this assay, our predictions must be interpreted in this context. Encouragingly, however, Cooper et al. (2006) do observe a strong correlation between activity in the assay and endogenous transcript levels in the cell lines studied, suggesting that a significant proportion of gene regulation is contained within promoter fragments of this size, and that this activity can be reliably measured to a large extent in the transient reporter assays. Furthermore, because precisely defined promoter fragments are assayed out of their genomic context, it is highly certain that the activity observed comes directly from that specific fragment. This decreases the search space for specific regulatory motifs and establishes defined boundaries for the regulatory module sufficient for gene regulation.

ESPERR predicts many promoters in the human genome not associated with start sites of known genes (as annotated by the UCSC genome browser). This is consistent with a number of recent findings suggesting that there are more promoters in the human genome than previously believed. Cooper et al. (2006) find a substantial number of genes with functional alternate promoters, and suggest that these might be due to highly tissue-specific alternate isoforms. The ENCODE Genes and Transcripts group (Guigó et al., 2006) annotated and verified a substantial number of novel transcription start sites. The ENCODE Transcriptional Regulation group (Weng et al., 2006) similarly integrated the results of more than 100 ChIP experiments to predict the presence of many novel promoters.

For the applications presented here we used ESPERR on alignments of at most seven species. Further increasing the number of species could add predictive power in some problems; our method can easily scale to incorporate more genomes, and because of the way we handle missing data, these could be picked from the many low-coverage genomic sequences currently becoming available. However, care must be taken in selecting what species to use. Very low coverage genomes may in some cases add more noise than exploitable signals, and in general the type of functional

elements under consideration should dictate species selection (McAuliffe et al., 2005). For example, if elements are not expected to be under very strong constraint, comparisons should be restricted to closely related species, while if the elements are very constrained, using more distant species increases the chance of seeing systematic patterns of change that can be captured by ESPERR (as in section 4.2).

Another challenge of our current methodology is a common difficulty in data mining: techniques that are strongly data-driven, and incorporate a random component, result in models that work quite well in practice but can be difficult to interpret. Developing methods to systematically interpret the encodings we produce will allow us not only to predict elements well, but describe the characteristics of such elements. This has the potential to aid in understanding the specific mechanisms involved. The integration of phylogenetic modeling and a data-driven algorithm in our current methodology is a first step in this direction. In the future we intend to develop approaches that strike a balance between adaptability to the provided training data and biological interpretability.

The intense efforts to characterize and improve predictions of regulatory regions and other functional intervals in the genome are yielding many helpful resources. Biochemical assays of protein binding and chromatin modifications at high resolution (Weng et al., 2006; Stamatoyannopoulis et al., 2006), predictions of clusters of conserved transcription factor binding sites (Blanchette et al., 2006), refined estimates of nucleotides under constraint (Siepel et al., 2005; Cooper et al., 2005), and other experimental and computational efforts provide a plethora of resources from which investigators can build hypotheses to test. ESPERR differs from other methods in its emphasis on training to discover both strong and weak signals in alignments, and in its broad applicability – as signals can be learned to discriminate potentially any functional classes for which training data are available. ESPERR can be applied to new sets of functional elements, such as those explored by the ENCODE project, to generate genome-wide predictions for many functional classes. Future efforts to better understand the many subtle signals discovered by ESPERR should provide new insights into the mechanisms underlying specific functions, which could then

be tested experimentally. Another avenue of future work will be to combine augment our classification models to take advantage of other datasets – such as ChIP-chip predictions of protein occupancy – in addition to genomic alignments data.

APPENDIX A

PSEUDOCODE FOR THE ESPERR SEARCH ALGORITHM

The search is initialized using some mapping, either a one-to-one mapping of the training data symbols (e.g. all alignment columns) or the result of another encoding selection procedure (e.g. the clustering based on ancestral base distributions). After each iteration, this will be replaced with the best mapping found in that iteration.

```
mapping = initialize_mapping()
```

We keep track of the best mapping seen, and its figure of merit. When the search terminates this best mapping corresponds to the final encoding.

```
best_merit_overall = −∞
best_mapping_overall = None
```

The search iterates until it has performed 1,000 iterations without any improvement over the best mapping seen.

```
while steps_since_best < 1,000:
```

Within each iteration, we keep track of the best candidate mapping found.

```
    best_merit = −∞
    best_mapping = None
```

The first set of candidate mappings is created by merging symbols in the current encoding. We consider a random sample of γ such candidates. For practical reasons we set a lower bound (e.g. 5) on the encoding size and skip this step if the encoding is already too small.

```
    if symbol_count > minimum_alphabet_size:
```

Sample γ pairs from all possible pairs of symbols that could be collapsed.

```
        for pair in sample( all_collapsible_pairs( mapping ), γ ):
```

Generate a new mapping in which that pair of symbols are merged

```
new_mapping = collapse( current_mapping, pair )
```

Evaluate the figure of merit when this mapping is applied to the training data. If it is the best so far for this iteration, save it.

```
merit = calc_merit( new_mapping )
if merit > best_merit:
   best_merit = merit
   best_mapping = new_mapping
```

The second set of candidates is created by extracting atoms which are currently grouped with other symbols. We consider a random sample of η such candidates. Again for practical reasons we only break out seeds which occur more than 10 times in the training data, since they will not comprise any context that can be incorporated in the model (see VOMM estimation).

```
for atom in sample( expandable_atoms( mapping ), η ):
```

Generate a new mapping with that atom separated.

```
new_mapping = expand( mapping, atom )
```

Evaluate the figure-of-merit when this mapping is applied to the training data. If it is the best so far for this iteration, save it.

```
merit = calc_merit( new_mapping )
if merit > best_merit:
   best_merit = merit
   best_mapping = new_mapping
```

We accept the best mapping from either the collapse or expand steps as the new mapping for the next iteration

```
mapping = best_mapping
```

When the new mapping is better that the best seen so far, we save it and reset the counters used to trigger the two heuristics and termination.

```
if best_merit > best_merit_overall:
   best_merit_overall = best_merit
   best_mapping_overall = best_mapping
   steps_since_best = 0
   steps_since_restart = 0
   steps_since_forced_expansion = 0
```

We now check if the "restarting" heuristic should be triggered. If we have gone r iterations without an improvement over the best mapping, we restart from that mapping and reset the counters for the heuristics.

```
if steps_since_restart >= r:
   steps_since_restart = 0
   steps_since_forced_expansion = 0
   mapping = best_mapping_overall
```

Next we check if the "forced expansion" heuristic should be triggered. If we have gone w iterations without improvement over the best mapping, we force e consecutive expansion steps. These expansions are part of a single iteration and do not affect the counters (the expansion procedure is otherwise identical to that above).

```
if steps_since_forced_expansion > w:
   steps_since_forced_expansion = 0
   for i from 0 to e:
      best_merit = 0
      best_mapping = None
      for atom in sample( expandable_atoms( mapping ), η ):
         new_mapping = expand( mapping, atom )
         merit = calc_merit( new_mapping )
         if merit > best_merit:
            best_merit = merit
```

```
            best_mapping = new_mapping
            mapping = best_mapping
```

Finally we increment the counters that keep track of when each heuristic is triggered and when the search terminates.

```
steps_since_best += 1
steps_since_restart += 1
steps_since_forced_expansion += 1
```

References

Bajic, V. B. and Seah, S. H., 2003. Dragon gene start finder: an advanced system for finding approximate locations of the start of gene transcriptional units. *Genome Res*, **13**(8):1923–1929.

Batzoglou, S., 2005. The many faces of sequence alignment. *Brief Bioinform*, **6**(1):6–22.

Bejerano, G., Haussler, D., and Blanchette, M., 2004. Into the heart of darkness: large-scale clustering of human non-coding DNA. *Bioinformatics*, **20 Suppl 1**:40–40.

Blanchette, M., Bataille, A. R., Chen, X., Poitras, C., Laganière, J., Lefèbvre, C., Deblois, G., Giguère, V., Ferretti, V., Bergeron, D., *et al.*, 2006. Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res*, **16**(5):656–668.

Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smit, A. F., Roskin, K. M., Baertsch, R., Rosenbloom, K., Clawson, H., Green, E. D., *et al.*, 2004. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res*, **14**(4):708–715.

Breiman, L., , Friedman, J., Olshen, R., and Stone, C., 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Buhlmann, P. and Wyner, A., 1998. Variable length Markov chains. *Annals of Statistics*, **27**:480–513.

Burge, C. and Karlin, S., 1997. Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, **268**(1):78–94.

Chang, C.-C. and Lin, C.-J., 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Consortium, E. P., 2004. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science*, **306**(5696):636–640.

Cooper, G. M., Stone, E. A., Asimenos, G., Green, E. D., Batzoglou, S., and Sidow, A., 2005. Distribution and intensity of constraint in mammalian genomic sequence. *Genome Res*, **15**(7):901–913.

Cooper, S. J., Trinklein, N. D., Anton, E. D., Nguyen, L., and Myers, R. M., 2006. Comprehensive analysis of transcriptional promoter structure and function in 1% of the human genome. *Genome Res*, **16**(1):1–10.

Cortes, C. and Vapnik, V., 1995. Support-Vector Networks. *Machine Learning*, **20**(3):273–297.

Costas, J., Casares, F., and Vieira, J., 2003. Turnover of binding sites for transcription factors involved in early Drosophila development. *Gene*, **310**:215–220.

Crooks, G. E., Hon, G., Chandonia, J. M., and Brenner, S. E., 2004. WebLogo: a sequence logo generator. *Genome Res*, **14**(6):1188–1190.

Davuluri, R. V., Grosse, I., and Zhang, M. Q., 2001. Computational identification of promoters and first exons in the human genome. *Nat Genet*, **29**(4):412–417.

Dermitzakis, E. T. and Clark, A. G., 2002. Evolution of transcription factor binding sites in Mammalian gene regulatory regions: conservation and turnover. *Mol Biol Evol*, **19**(7):1114–1121.

Dorschner, M. O., Hawrylycz, M., Humbert, R., Wallace, J. C., Shafer, A., Kawamoto, J., Mack, J., Hall, R., Goldy, J., Sabo, P. J., *et al.*, 2004. High-throughput localization of functional elements by quantitative chromatin profiling. *Nat Methods*, **1**(3):219–225.

Down, T. A. and Hubbard, T. J., 2002. Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res*, **12**(3):458–461.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G., 1998. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press.

Elnitski, L., Hardison, R. C., Li, J., Yang, S., Kolbe, D., Eswara, P., O'Connor, M. J., Schwartz, S., Miller, W., and Chiaromonte, F., *et al.*, 2003. Distinguishing regulatory DNA from neutral sites. *Genome Res*, **13**(1):64–72.

Gross, S. S. and Brent, M. R., 2006. Using multiple alignments to improve gene prediction. *J Comput Biol*, **13**(2):379–393.

Guigó, R. et al., 2006. The transcriptional landscape of the ENCODE regions of the human genome reveals long rang transcriptional networks. *Nature (submitted)*, .

Hasegawa, M., Kishino, H., and Yano, T., 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol*, **22**(2):160–174.

Hsu, C.-W. and Lin, C.-J., 2002. A comparison of methods for multi-class support vector machines. *IEEE Trans. Neural Networks*, **13**:415–425.

Karolchik, D., Baertsch, R., Diekhans, M., Furey, T. S., Hinrichs, A., Lu, Y. T., Roskin, K. M., Schwartz, M., Sugnet, C. W., Thomas, D. J., *et al.*, 2003. The UCSC Genome Browser Database. *Nucleic Acids Res*, **31**(1):51–54.

Kent, W. J., Sugnet, C. W., Furey, T. S., Roskin, K. M., Pringle, T. H., Zahler, A. M., and Haussler, D., 2002. The human genome browser at UCSC. *Genome Res*, **12**(6):996–1006.

King, D. C., Taylor, J., Elnitski, L., Chiaromonte, F., Miller, W., and Hardison, R. C., 2005. Evaluation of regulatory potential and conservation scores for detecting cis-regulatory modules in aligned mammalian genome sequences. *Genome Res*, **15**(8):1051–1060.

Kolbe, D., Taylor, J., Elnitski, L., Eswara, P., Li, J., Miller, W., Hardison, R., and Chiaromonte, F., 2004. Regulatory potential scores from genome-wide three-way alignments of human, mouse, and rat. *Genome Res*, **14**(4):700–707.

Korf, I., Flicek, P., Duan, D., and Brent, M. R., 2001. Integrating genomic homology into gene structure prediction. *Bioinformatics*, **17 Suppl 1**:140–148.

Liu, R. and States, D. J., 2002. Consensus promoter identification in the human genome utilizing expressed gene markers and gene modeling. *Genome Res*, **12**(3):462–469.

Luchina, N. N., Krivega, I. V., and Pankratova, E. V., 2003. Human Oct-1L isoform has tissue-specific expression pattern similar to Oct-2. *Immunol Lett*, **85**(3):237–241.

Ludwig, M. Z., Patel, N. H., and Kreitman, M., 1998. Functional analysis of eve stripe 2 enhancer evolution in Drosophila: rules governing conservation and change. *Development*, **125**(5):949–958.

Margulies, E. H., Blanchette, M., Haussler, D., and Green, E. D., 2003. Identification and characterization of multi-species conserved sequences. *Genome Res*, **13**(12):2507–2518.

Margulies, E. H., Cooper, G. M., Asimenos, G., Thomas, D. J., Dewey, C. N., Siepel, A., Birney, E., et al., 2006. Relationship between evolutionary constraint and genome function in 1% of the human genome. *Nature (submitted)*, .

Mayrose, I., Graur, D., Ben-Tal, N., and Pupko, T., 2004. Comparison of site-specific rate-inference methods for protein sequences: empirical Bayesian methods are superior. *Mol Biol Evol*, **21**(9):1781–1791.

McAuliffe, J. D., Jordan, M. I., and Pachter, L., 2005. Subtree power analysis and species selection for comparative genomics. *Proc Natl Acad Sci U S A*, **102**(22):7900–7905.

McGuire, G., Denham, M. C., and Balding, D. J., 2001. Models of sequence evolution for DNA sequences containing gaps. *Mol Biol Evol*, **18**(4):481–490.

Ney, P. A., Andrews, N. C., Jane, S. M., Safer, B., Purucker, M. E., Weremowicz, S., Morton, C. C., Goff, S. C., Orkin, S. H., and Nienhuis, A. W., *et al.*, 1993. Purification of the human NF-E2 complex: cDNA cloning of the hematopoietic cell-specific subunit and evidence for an associated partner. *Mol Cell Biol*, **13**(9):5604–5612.

Noble, W. S., Kuehn, S., Thurman, R., Yu, M., and Stamatoyannopoulos, J., 2005. Predicting the in vivo signature of human gene regulatory sequences. *Bioinformatics*, **21 Suppl 1**:338–343.

Pischedda, C., Cocco, S., Melis, A., Marini, M. G., Kan, Y. W., Cao, A., and Moi, P., 1995. Isolation of a differentially regulated splicing isoform of human NF-E2. *Proc Natl Acad Sci U S A*, **92**(8):3511–3515.

R Development Core Team, 2006. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Sabo, P. J., Hawrylycz, M., Wallace, J. C., Humbert, R., Yu, M., Shafer, A., Kawamoto, J., Hall, R., Mack, J., Dorschner, M. O., *et al.*, 2004. Discovery of functional noncoding elements by digital analysis of chromatin structure. *Proc Natl Acad Sci U S A*, **101**(48):16837–16842.

Schultz, S. C., Shields, G. C., and Steitz, T. A., 1991. Crystal structure of a CAP-DNA complex: the DNA is bent by 90 degrees. *Science*, **253**(5023):1001–1007.

Seber, 1984. *Multivariate Obervations*. Wiley, New York. NY.

Siepel, A., Bejerano, G., Pedersen, J. S., Hinrichs, A. S., Hou, M., Rosenbloom, K., Clawson, H., Spieth, J., Hillier, L. W., Richards, S., *et al.*, 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res*, **15**(8):1034–1050.

Siepel, A. and Haussler, D., 2004. Combining phylogenetic and hidden Markov models in biosequence analysis. *J Comput Biol*, **11**(2-3):413–428.

Stamatoyannopoulis, J. et al., 2006. Chromatin and replication architecture of the human ENCODE regions. *Nature (submitted)*, .

Tompa, M., Li, N., Bailey, T. L., Church, G. M., De Moor, B., Eskin, E., Favorov, A. V., Frith, M. C., Fu, Y., Kent, W. J., *et al.*, 2005. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, **23**(1):137–144.

Tsang, A. P., Visvader, J. E., Turner, C. A., Fujiwara, Y., Yu, C., Weiss, M. J., Crossley, M., and Orkin, S. H., 1997. FOG, a multitype zinc finger protein, acts as a cofactor for transcription factor GATA-1 in erythroid and megakaryocytic differentiation. *Cell*, **90**(1):109–119.

Valverde-Garduno, V., Guyot, B., Anguita, E., Hamlett, I., Porcher, C., and Vyas, P., 2004. Differences in the chromatin structure and cis-element organization of the human and mouse GATA1 loci: implications for cis-element identification. *Blood*, **104**(10):3106–3116.

Wang, H., Zhang, Y., Cheng, Y., Zhou, Y., King, D. C., Taylor, J., et al., 2006. Experimental validation of predicted mammalian erythroid cis-regulatory modules. *Genome Res (to appear)*, .

Wasserman, W. W. and Sandelin, A., 2004. Applied bioinformatics for the identification of regulatory elements. *Nat Rev Genet*, **5**(4):276–287.

Waterston, R. H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J. F., Agarwal, P., Agarwala, R., Ainscough, R., Alexandersson, M., An, P., *et al.*, 2002. Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**(6915):520–562.

Weng, Z., Trinklein, N. D., Fu, Y., Zhang, Z., Karaozl, U., Barrera, L., et al., 2006. The landscape of transcriptional regulatory elements in the ENCODE regions. *Nature (submitted)*, .

Woolfe, A., Goodson, M., Goode, D. K., Snell, P., McEwen, G. K., Vavouri, T., Smith, S. F., North, P., Callaway, H., Kelly, K., *et al.*, 2005. Highly conserved non-coding sequences are associated with vertebrate development. *PLoS Biol*, **3**(1).

VITA

James Taylor is a Ph.D. candidate in Computer Science and Engineering at Penn State University. He received his B.S in Computer Science (Magna Cum Laude) from the University of Vermont in 2000. He worked in industry as a software engineer at the NYBOR corporation, and later as senior software engineer at 4Lane Digital. He entered the Ph.D. program at Penn State University in July 2003, performing computational biology research in Webb Miller's lab at the Penn State Center for Comparative Genomics.