UNIVERSITY OF CALIFORNIA

Los Angeles

Penalized Bayesian Model Selection and Prediction for Gene Regulation in Higher Organisms

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Matthew David Levinson

2013

Abstract of the Dissertation

Penalized Bayesian Model Selection and Prediction for Gene Regulation in Higher Organisms

by

Matthew David Levinson

Doctor of Philosophy in Statistics

University of California, Los Angeles, 2013

Professor Qing Zhou, Chair

Penalization and sparse model selection have become topics of intense research interest in the era of big data, newly available through ubiquitous computing power, advancing data collection technologies, and internet connectivity. In genomics, chromatin immunoprecipitation, microarrays, and next generation sequencing have made available a wealth of information which continues to accumulate and which we have only begun to understand and fully utilize. We propose two penalized Bayesian techniques, one to select a sparse set of DNA binding factors (DBFs) from a large library with enriched binding to the genome in a set of regions of interest and to predict joint binding landscapes for the selected DBFs, and another to predict gene expression from joint binding landscapes.

Cellular processes are controlled, directly or indirectly, by the binding of hundreds of different DBFs to the genome. One key to deeper understanding of the cell is discovering where, when, and how strongly these DBFs bind to the DNA sequence. Direct measurement of DBF binding sites (e.g. through ChIP-Chip or ChIP-Seq experiments) is expensive, noisy, and not available for every DBF in every cell type. Naive and most existing computational approaches to detecting which DBFs bind in a set of genomic regions of interest often perform poorly, due to the high false discovery rates and restrictive requirements for prior knowledge.

We develop a penalized iterative sampling Bayesian method for identifying DBFs active in the considered regions and predicting a joint probabilistic binding landscape. Utilizing a sparsity penalization, SparScape is able to select a small subset of DBFs with enriched binding sites in a set of DNA sequences from a much larger candidate set. This substantially reduces the false positives in prediction of binding sites. Analysis of ChIP-Seq data in mouse embryonic stem cells (ESCs) and simulated data show that SparScape dramatically outperforms the naive motif scanning method and the comparable computational approaches in terms of DBF identification and binding site prediction.

We also propose an extension of Bayesian treed regression to predict gene expression from joint binding landscapes. Rather than sampling from the space of possible partitioning trees, we follow a broad optimization approach, forking the growing partitioning tree at each possible split if multiple possible splits yield similar results in the given objective function. After growing the tree, we select variables at each leaf node of each forked partitioning tree, then take the union of these selected variables and the splitting variables at each internal node and re-grow the partitioning tree considering only the selected variables.

The dissertation of Matthew David Levinson is approved.

Matteo Pellegrini

Yingnian Wu

Kenneth Lange

Qing Zhou, Committee Chair

University of California, Los Angeles

2013

To my ragamuffin co-conspirator…

Table of Contents

List of Figures

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Qing Zhou, for his guidance and support. I feel lucky to have had him as my advisor and would not have developed into the statistician and researcher I am today without him. He has guided me from novice to collaborator, and I am forever grateful.

I would also like to thank my committee members Ken Lange, Matteo Pellegrini, and Yingnian Wu for their help and guidance along the way.

But mostly I would like to thank my wife Teryn. Her love, patience, and endless dedication make each day possible, and worthwhile. I would not be nearly the person I am today without her.

VITA

2003	B.A., Computer Science Pomona College Claremont, CA
2001–2003	Programmer/Research Assistant University of Pennsylvania School of Medicine Philadelphia, PA
2004–2006	Programmer/Analyst Center for Neurobehavioral Genetics University of California, Los Angeles Los Angeles, CA
2006–2008	Research Assistant Center for International Development Harvard University Cambridge, MA
2009-2011,2013	Teaching Assistant, Reader, and Graduate Student Researcher Department of Statistics, Department of Civil and Environmental Engineering University of California, Los Angeles Los Angeles, CA
2011–2013	Fellow, Burroughs Wellcome Fund IT-MD Program

PUBLICATIONS

Levinson M, Zhou Q (2013) A penalized bayesian approach to predicting sparse protein-DNA binding landscapes. *Bioinformatics, forthcoming.*

Field E, Levinson M, Pande R, Visaria S (2008) Segregation, rent control, and riots: the economics of religious conflict in an Indian city. *American Economic Review*, **98** (2), 505–510.

Jasinska A, Service S, Levinson M *et al.* (2007) A genetic linkage map of the vervet monkey (Chlorocebus aethiops sabaeus). *Mammalian Genome*, **18** (5), 347–60.

Herzberg I, Jasinka A, et al. (2006) Convergent linkage evidence from two Latin-American population isolates supports the presence of a susceptibility locus for bipolar disorder in 5q31. *Human Molecular Genetics*, **15** (21), 3146–53.

Levinson D, Levinson M, Segurado R, Lewis CM (2003) Genome scan meta-analysis of schizophrenia and bipolar disorder, part I: methods and power analysis. *American Journal of Human Genetics*, **73** (1), 17–33.

CHAPTER 1

Introduction

1.1 Motivation and Gene Regulation

A key to the control of many complex processes in the cell is the binding of various factors to the DNA sequence. In particular we are interested here gene regulation. Genes are regulated at various points along the line from transcription to final degradation of a protein or RNA. In all cases, the regulation of a gene begins with the promotion or inhibition of transcription. While other factors, such as environmental conditions and biochemical modifications of the DNA or of histones, also contribute directly to transcriptional regulation, a key factor in this process is the set of DNA binding factors (DBFs). These DBFs are proteins or RNAs such as transcription factors (TFs), nucleosomes (complexes formed by histones), or microRNAs that bind directly to the genome, most with some level of sequence specificity or preference. A key to understanding gene regulation is determining where these DBFs bind in the genome in a given cell type and set of conditions.

Depending on the combinatorial logic of binding and regulation, the same DBF can have an inhibitory impact in some cases and a promotional impact in others [OZW09]. A full understanding of the regulatory effects of DBF binding requires an inclusive understanding of the joint binding of all DBFs with non-negligible levels of binding in the genomic regions of interest. It is also well known that some DBFs are more sequence specific than others and bind more strongly to preferred sequences, while others are less sequence specific and bind less strongly to a larger collection of possible binding sequences. We are thus motivated to develop a new method to achieve better prediction of this joint binding. A joint estimation or prediction of binding sites and binding strengths for all DBFs over genomic regions of interest is called a binding landscape. More formally, we define a binding landscape as the base pair (bp) specific probability of binding for each of a library of DBFs over a set of genomic regions.

Once we have a prediction or estimate of the binding landscape we are further interested in utilizing this information to predict mRNA expression. Because the same DBF can have different effects in different joint binding configurations, simple methods that estimate a single, universal effect of binding by some DBF are inadequate. So we further develop a method for partitioning genes into sets where the binding for each DBF has a similar regulatory impact over all genes in the set. This algorithm can be used with binding landscapes predicted by a landscape prediction algorithm such as ours. It can also be used with a landscape estimated experimentally through chromatin immunoprecipitation followed by use of microarray or sequencing technologies (ChIP-Chip and ChIP-Seq, respectively).

1.2 Existing Methods

1.2.1 Binding Landscape Prediction

The first requirement for statistically predicting DBF binding is some model of DNA sequences preferred by that DBF, known as a binding motif. The ubiquitous model for these motifs that we also employ is the position-specific weight matrix (PWM). A PWM describes a product-multinomial model for a motif w base pairs long, where the probability of DBF binding for a sequence is the product of independent 4-state multinomial models that give the probability for each of the four nucleic acids at each of the w positions in the motif. Much research has been done in recent decades to find the binding motifs for many DBFs, most especially TFs [HS99, HKB02, GL03].

In this work we draw motif models from TRANSFAC [MFG03], one of the primary databases that aggregates these PWM estimates from the literature.

Armed with a library of estimated PWMs, and given that a satisfactory view of transcriptional regulation requires a joint binding landscape, it is tempting to employ the simplest approach of building a joint binding landscape by scanning the genome or genomic regions of interest with each PWM separately and aggregating these scans. Unfortunately, considering DBFs one at a time leads to many false positives, both in determining which DBFs have significantly enriched binding sites in a set of genomic regions and in predicting the exact locations of binding sites. This results in a limited and misleading view of the processes controlled by these DBFs. This has motivated recent work on developing more sophisticated methods for jointly predicting binding landscapes for a set of DBFs.

Currently, only a few algorithms have been developed to predict joint binding landscapes at single base pair (bp) resolution for all DBFs. [WH09] develop the software COMPETE for use genome wide in yeast. They formulate the problem as a Boltzmann chain, a generalization of a hidden Markov model (HMM) that allows transition and emission weights to be an non-negative number. This requires a silent central state and results in a model quite similar to the portion of our model that deals with the sequence likelihood (see Chapter **??**). In principle this formulation would allow for estimation of DBF concentrations. They found, like we also found in our formulation, that this learning can result in over-fitting and poor results. Instead of counteracting this by employing penalization as we do, they simply set each concentration to the K_d of that DBF's PWM (see [GN05]) multiplied by a global scalar. They choose this scalar over a small set of values by choosing the value that maximizes the correlation between the nucleosome occupancy in the predicted binding landscape and the experimental occupancy from [KMF09]. We find quite poor results in applying COMPETE to data relevant to the questions we investigate.

[HCH09] developed STAP, designed primarily to interrogate ChIP-Seq or ChIP-

Chip data for one TF and to choose co-factors one at a time by choosing co-factors that increase the correlation between the predicted two factor landscape and the ChIP-Seq counts for the primary TF. This is the only other method that jointly estimates factor concentrations, something made possible by the consideration of only a single pair of TFs at a time. They investigated the same ChIP-Seq data we explore, and achieved promising results, including and some agreement with our own. But like with single DBF raw score scanning, we have found with our algorithm that false positives in co-factor prediction are significantly increased when considering only one DBF at a time.

A number of other groups have developed binding landscape prediction algorithms for use in higher eukaryotes, but they have all required a small, pre-selected set of DBFs and either forgo a concentration parameter or require one to be given as input [HSB10, KLS11, LYL09, RLS09]. [HCH09] optimizes the model parameters by maximizing correlation between predictions and ChIP data. [RLS09] mostly explore the theoretical implications of binding on gene expression through examination of stylized, artificial data. Only [KLS11] also utilized ChIP-Seq data as a source of direct information aside from their implicit use in DBF motif discovery, and then only for the nucleosome.

There has also been a fair amount of work developing approachs that employ different types of data or quite different methodological formulations. [AEP11] incorporates multiple sequences and phylogenetic information while still basing the bp-by-bp score on a PWM ratio as we do. The model is quite sophisticated with a number of modeling elements and many free parameters introduced to incorporate the multiple aligned sequences. Including multiple species can add valuable information, but the results they report are mostly from running their algorithm a single DBF at a time only on a set of the 900 regions most highly enriched for that DBF in ChIP-Seq experiments. They report a single result predicting a binding landscape with multiple DBFs simultaneously, and then only on 76 carefully chosen regions with 5 DBFs in the fly.

A very different approach is taken by [MRA12], who integrate four types of input data. They predicted regulatory network edges using two methods. First they used a sum rule across six data type specific networks they built, pruning predicted edges to include only the top-scoring 2%. Second they used a logistic regression with the edge score from each of their six networks as the predictors and the existence or absence of an edge in a small validated network as the dependent variable. Comparison of the regulatory edges implied by our results with the networks predicted by their and other similar work is promising future work.

Similarly, [EPS10] predict general binding likelihood for a DBF in a set of regions of interest with logistic regression. They first estimate the general likelihood of binding in a region using similar non-sequence specific information to [MRA12], such as conservation, distance to a transcription start site (TSS), and sequence conservation. Then they assess the overall strength binding motif scores across the region and combine these two steps to make a general prediction of the likelihood of binding by a given DBF in a particular region of interest. They do not provide bp specific binding predictions.

Other groups have also worked on combining different sources of information to predict DBF binding. [RKK10] concentrated on the effects of histone acetylation (HAc), combining HAc information with sequence specific motif scores. [WRW10] also developed a method to combine histone modification data with motif scores.

While these methods utilize information we do not, they do not directly utilize the most direct information available on DNA binding, ChIP-Seq and ChIP-chip experiment results. Most use it only as a source of information for validation. They are also not designed for our motivating problem, which is to accurately select a small set of DBFs bound to a set of genomic regions of interest from a complete candidate library. In the next chapter we develop a method that performs quite well investigating this question.

1.2.2 Regression Trees

Most famously explored early on by [BFS84], there has been extensive study of classification and regression trees. The original, stilly widely used algorithm, named CART, was a non-parametric model to partition the space of a set of predictor variables into regions with relatively homongeneous responses through recursive partitioning. The idea was to forgo parameter estimation by simply predicing the response at each leaf node as the mean response of the observations in the partition of the predictor variables assigned to that leaf. This is accomplished by building a partitioning tree where at each internal node one has a splitting rule where the observations remaining at that node are split into two or more subsets. This continues until some stopping rule is reached. This is either considered the final partition or the tree is then pruned back to reduce the risk of over-fitting, shrink the tree, and increase interpretability. Many developments have been made since this first basic method was proposed, including using some form of regression both to predict the response at each leaf node and to split the observations at internal nodes. In our work we propose some extensions and alternate optimizations of the regression tree approach as applied to the prediction of gene expression from the expression of DBFs and from DBF binding data. In the following sub-sections we discuss other attempts to accomplish this prediction and general methodological developments pertinent to our work.

1.2.3 Regression Trees and Partitioning

A number of groups have developed sophisticated methodologies for building regression trees with parametric prediction of the response at each leaf, some with extensive theoretical frameworks. Most pertinent to our work are the methods developed by Loh and Hornik, Hothorn, and Zeileis [KL01, Loh02, HHZ12, ZH07, ZHH08]. Hornik, Hothorn, and Zeileis derive that the distribution of the decorrelated score function for some process over an ordered variable converges to standard Brownian motion under the functional central limit theorem.

In a regression tree framework where we wish to assess possible splits in the data at an internal node, the score function is the least squares error between the predicted value and the response. Intuitively, if the parameter is stable across the ordering of some predictor variable, then we expect this error (residual) to fluctuate around its expected value of zero. If the parameters are stable across the range of the ordered variable, these residuals follow standard Brownian motion across that range, and they derive tests for parameter stability across the range of the ordered variable based on this fact. In their framework, applied to our problem of interest, at each internal node one uses their test for parameter stability across each predictor variable and chooses to split the observations at that internal node at the variable and cut point that gives the lowest p-value in their test. If no p-value is below some pre-determined threshold ($\alpha -$level), no further splits are made.

When considering numerical variables that we are interested in, Loh [KL01, Loh02] instead uses various $\Xi -$squared tests on the counts of signs of the residuals at each internal node grouped by quartiles of some ordered numerical variable to determine parameter instability. A significant portion of their motivation is that in the case where categorical and numeric variables are considered, in the independent case, the numerical variables are much more likely to falsely split simply because of the much larger number of possible splits. This approach also has the advantage that they develop a number of explicit tests for parameter instability over the joint space of two variables.

We seek to develop beyond these approaches for a number of reasons. First, we prefer a Bayesian approach which allows exploration of the parameter space beyond a greedy approach. The aforementioned approaches expend significant energy showing the unbiasedness (or near unbiasedness) of their approaches. This is a classic element of the debate between Bayesian and classical approaches, and we come down on the Bayesian side in this debate, believing that unbiasedness is not itself an end goald and should not be pursued over predictive accuracy. Second, while these tests are attractive because of their theoretical properties, we prefer to spit nodes based on predictive accuracy, not theoretical properties or goodness-of-fit with asymptotic distributions. Third, in a genomics environment with a very large number of predictor variables, it is desireable to include some measure of variable selection in the methodology, as including extraneous variables with no influence on the outcome which we want to predict can only degrade our predictions and the selection of variables itself can yield interesting insights.

Our work is most directly a development of the work on Bayesian treed models proposed in [CGM00, CGM01]. In that work, Chipman et al. propose a Bayesian formulation of the regression tree model where partitions at the internal regression tree nodes and prediction of response(s) at the leaf nodes are based on a Bayesian formulation of ordinary least squares (OLS) regression. They devise an MCMC algorithm for sampling from the space of possible trees based on a set of options for proposing an alteration to the current tree, adding leaf nodes, deleting any node, or swapping splitting rules between nodes. The demonstration of their methods are on data sets with a small number of variables and observations, and they are not interested in variable selection. Even with such smaller data sets, their sampling method tends to get trapped in possible local modes. Their solution is running a number of restarts of their MCMC algorithm. This is an unsatisfying solution with much larger data sets where the number of local modes is likely to be very large and all of these modes are unlikely to be explored through a computationally feasible number of restarts.

1.2.3.1 Predicting Gene Expression

Our problem of interest, predicting gene expression from the wealth of modernly available genomic information, has also attracted a wide array of interest. These include methods predicting gene expression only from other gene expression and from a combination of gene expression and DBF binding data. A number of groups have used some score to combine both the DBF binding strength, usually measured through direct ChIP experiments, and DBF expression [TLV11, CXR08], or just DBF expression [GFB04] to predict the overall regulatory effect of DBFs through a regression framework. Others have used a regression tree framework to allow for different combinatorial effects of regulators but have only used a classic CART framework trying to build homogeneous responses at leaf nodes using either DBF expression and binding [RZ06] or just DBF binding [PLL04]. Ruan et al. extended their algorithm to do prediction using ensemble decision trees, but limited themselves to using DBF binding as predictors [RDP09]. A partial step towards incorporating the fact that DBFs may activate or repress in different settings was taken by [XV05] who used DBF scores in gene promoters in a regression setting to predict the absolute value of gene expression relative to baseline over a vector of responses in various conditions.

Another popular approach has been to use some sort of clustering to group genes into regulatory modules. [SYK03] clustered gene expression values to create initial modules, then refined these modules by incorporating promoter DBF motif scores and predicting gene expression from motif and cluster information interatively. [BGL03] also developed an iterative clustering algorithm using ChIP data and DBF expression as predictors of target gene expression.

Other groups have takEn alternate methodological approaches such as SVD [YTC02], Bayesian factor analysis [SJ06], and discretizing regulatory effect into up-regulated and not-up-regulated and predicting up or down regulation using boosting [MKW04]. Perhaps most similar to our approach is that of [xWD04], who combine gene expression and ChIP data through a partitioning EM algorithm to partition genes into regulatory modules via a Bayesian network.

While many of these methods attain adequate prediction and have allowed biological insights, we believe that a few modeling decisions can be made to better capture biological phenomena of interest. It is known that some DBFs have a negligible measurable regulatory impact. This suggests that selecting out DBFs that have no direct or indirect (through shift in the combinatorial regulatory code and the regulatory impact of other DBFs) impact is a worthy goal. Directly modeling the effects of DBF binding and expression in the leaf nodes independently can also yield insights into which modules experience differing effects of the same DBF, and which do not. Doing this through a regression tree framework instead of a clustering framework allows for the same DBF to have different regulatory effects on different target genes even if those genes tend to cluster together in overall expression level. We also believe that a Bayesian treed regression framework with an optimization approach compromising between a fully greedy approach and a sampling approach is the best fit for the large data genomics problems we are interested in.

CHAPTER 2

SparScape: A Penalized Bayesian Approach to Predicting Sparse Protein-DNA Binding Landscapes

2.1 Goals and Data

Due to computational limits, it is impossible to predict a joint, bp specific binding landscape for all DBFs with unknown concentrations over the entire genome in higher eukaryotes with large genomes and many DBFs. We are interested in relaxing at least one of these constraints. We chose to allow relaxation of the cap on the number of DBFs. We are thus limited to exploring a subset of the genome. One motivating type of genomic subset is a set of regions known to be co-bound by a small group of DBFs based on ChIP-Seq data. In such a genomic subset, we do not expect most DBFs to have a significant number of binding sites. Most DBF binding sites are relatively short (6–15 bps), and non-functional but fairly strong potential sites (false positives) appear throughout the genome. Thus, the false positive binding sites in a predicted binding landscape can be substantially reduced if only the DBFs with significantly enriched binding in the regions of interest are considered. However, it is limiting to require the complete set of DBFs enriched in the considered regions to be known *a priori* as is done in the existing work on similar questions in higher eukaryotes.

We were also interested in utilizing more information than is used both directly and flexibly in existing single base pair binding landscape prediction algorithms. In existing work, ChIP-Seq is used only implicitly in that ChIP experiments were used to gather the date with which PWMs were estimated, used as gold standard information with which correlation should be maximized in optimizing the model parameters, or used only for nucleosome binding and then only to construct binding priors for the TFs considered. We believed that ChIP data should be utilized directly and flexibly for any DBFs for which it is available and formulated our model to allow for this.

With these goals in mind we developed a method that offers a principled way to select an (often small) subset of DBFs active in the regions of interest and to reduce the false positive signal in the predicted probabilistic binding landscape, eliminating the need for prior knowledge of the set of enriched DBFs or DBF concentrations. In the motivating genomic subset, our method allows for the discovery of unknown co-factors that commonly bind near the DBFs with ChIP data (ChIP DBFs). The predicted joint binding landscape provides a global and quantitive view of the binding pattern among the DBFs. This is an initial step to the study of combinatorial regulatory logic among multiple DBFs.

The input data for this method are therefore quite flexible. We require the sequence of the genomic regions of interest. This sequence could be the whole genome in organisms with smaller genomes, though our selection of DBFs from an input library is not designed for this sort of data, so should perhaps be skipped when examining entire genomes. The second requirement is the library of candidate DBFs. A number of databases that contain collections of PWMs estimated from published research are available, such as JASPAR and TRANSFAC, the database we employ [SAE04, MFG03].

A third optional data input is a set of binding windows deemed significant from a ChIP-Seq or ChIP-Chip experiment. These may be included for any of the candidate DBFs or the nucleosome, but are not required. The final input data is not utilized in the model itself, but we have found it useful in a preparatory step. Our motivating problem is the discovery of co-factors for a set of DBFs of interest with ChIP data. To be of interest these ChIP data must be from the same cell type in the same environmental conditions. If gene expression data is available from the same cell type and environment, we have found it useful to prune the candidate library to contain only those DBFs with non-negligible expression in this cell type and envrionment.

2.2 Methods

2.2.1 Overview of SparScape

Our method, SparScape, proceeds in two stages. First, from a candidate set of arbitrary size we select the DBFs with significant binding in the considered regions. Second, we do a refined prediction of the binding landscape considering only the selected DBFs. Figure 2.1 gives a schematic illustration of the method.

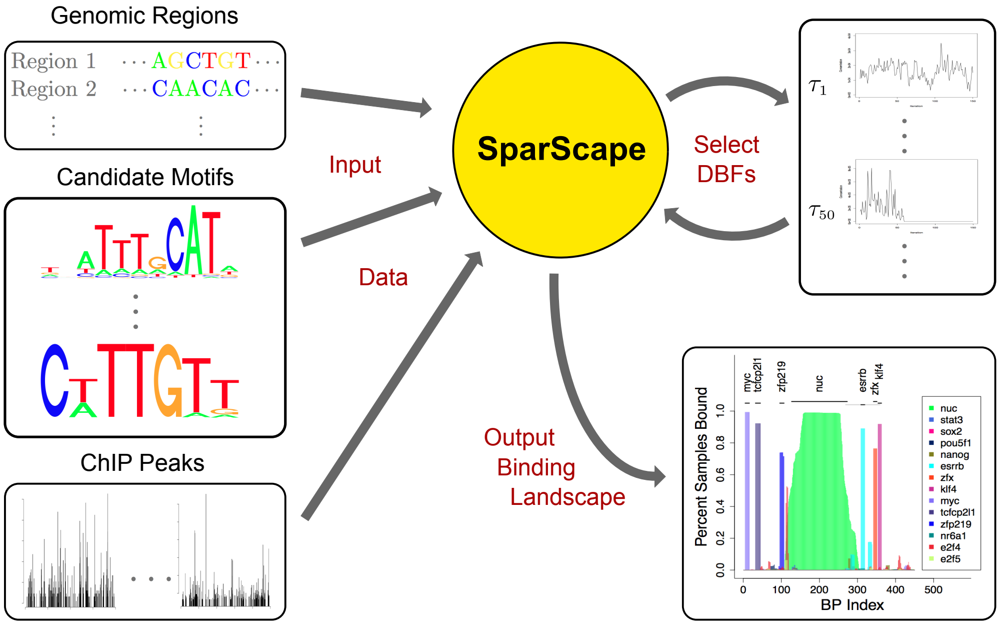


Figure 2.1: SparScape schematic. SparScape takes as input a set of genomic regions, a candidate library of DBFs with previously estimated binding motifs, and ChIP-Seq peak information where available. Using a penalized iterative sampling procedure, SparScape selects a subset of DBFs from the candidate library by estimating some of the local concentrations as exactly zero, such as τ_{50} above. SparScape then predicts a probabilistic binding landscape for the selected DBFs.

SparScape takes as input the previously estimated binding motifs for a set of candidate DBFs, the sequence of a set of genomic regions of interest, and genome-wide binding data (e.g. ChIP-Seq) for any of the candidate DBFs if available. The set of regions could be the whole genome when examining small genomes. We consider nucleosome binding because nucleosome occupancy blocks the binding of many other DBFs, and recent studies have demonstrated the utility of nucleosome models in protein binding landscapes [KMF09, RLS09, WH09]. We model nucleosome binding preferences by a position-specific Markov model proposed by [KMF09]. The 10 base pairs at the left and right ends of the 147 base pair long putative binding site are modeled as uniform in order not to capture any biasing effects of the procedure used to cleave the DNA bound by a nucleosome. The middle 127 base pairs are modeled with a position specific Markov model, where a separate 1st order Markov probability (the probability of a base appearing at that position in the putative binding site given the base in the previous position) is estimated for each of sites 11 through 137 in the bound sequence from the set of nucleosome bound sites detected in that paper. The binding of non-nucleosome DBFs is modeled by position-specific weight matrices (PWMs). A PWM describes a product-multinomial model, where each position in the motif is modeled with an independent multinomial model over the four nucleic acids that could appear at that position. The probability of a sequence representing a binding site is the product of the multinomail probabilities at each position. The background is modeled by a 5^{th}-order Markov chain estimated from a large set of sampled or simulated regions similar to the regions of interest.

We model ChIP data as a set of binary windows, where a window of bps around the center of a ChIP peak is called a ChIP window for that DBF. In this work, we use ChIP windows of 50 bps on either side of a ChIP-Seq peak. Our method estimates the probability that a binding site for a DBF with ChIP data is within one of its own ChIP windows. This is in the same spirit as the use of DNase I sensitivity measures to create an informative prior distribution for TF binding [KLS11, NGH07], though SparScape can exploit binding data for any of the candidate DBFs or the nucleosome. Moreover, ChIP windows are included as part of a generative probabilistic model with parameters related to the accuracy and sensitivity of ChIP peaks, resulting in a more principled and flexible utilization of ChIP data.

Estimating concentrations is a unique feature of SparScape. This makes it impossible to exactly calculate the binding landscape through forward-backward summation. Instead we explore the posterior distribution through a penalized iterative sampling approach, simultaneously selecting DBFs, estimating model parameters, and predicting the binding landscape. Jointly estimating the concentrations introduces a risk of excessive false positives, especially for DBFs with less informative motifs where we expect non-functional matches to occur frequently in the genome. To avoid this we employ a penalty on the predicted site counts in each iteration, penalizing in proportion to the expected number of false positive sites estimated with control regions given the current parameter values. Intuitively, this removes the expected false positive sites from the sampled sites. The level of penalization is controlled by a tuning parameter, chosen in the first stage through ten-fold cross-validation. With a proper level of penalization, concentrations of many DBFs will be estimated as exactly zero, achieving the goal of DBF selection. The final binding landscape, considering only the selected DBFs, is predicted in the second stage.

When selection and prediction are completed, SparScape reports a binding landscape, which gives the probability of binding at each bp by each selected DBF, and the estimated parameter values, including the local concentrations for the DBFs (local to the regions considered). It also reports the binding configuration and the parameter values sampled at each iteration, allowing, for example, construction of credible intervals for the parameters and examination of high-order interactions between binding at different sites.

2.2.2 The SparScape Model

Consider the sequence S of a set of genomic regions with total length $|S|$, and the set of ChIP windows D in these regions for all ChIP DBFs. Let K be the number of candidate DBFs, and Θ denote the set of binding model parameters for all K DBFs, including the nucleosome, and the background model. Under the standard steric hindrance contraint, we define a binding configuration as a partition of the sequence S into unbound background sites and binding sites for the K DBFs. Denote a configuration by $A = (a_1, a_2, \ldots, a_{|A|})$, where a_i is the index of one of the $K+1$ models; it represents a single unbound site covering $L_0 = 1$ bp when $a_i = 0$, a nucleosome covering $L_1 = 147$ bps when $a_i = 1$, and a non-nucleosome DBF from the candidate library covering L_k bps when $a_i = k \in \{2, \ldots, K\}$, where L_k is the length of the motif for the k^{th} DBF. Figure 2.2(a) illustrates an example configuration.

Let ϕ be the probability that a binding site for a ChIP DBF is within one of its ChIP windows and γ be the probability that an unbound bp is not covered by any of the ChIP windows. Write $\Phi = (\phi, \gamma)$. Denote by $T = (\tau_0, \ldots, \tau_K)$ the probabilities of initiating a background site ($k = 0$) and other DBF sites ($k = 1, \ldots, K$) at a given location $\left(\sum_k \tau_k = 1, \tau_k \geq 0\right)$. This can be thought of as a vector of local concentrations.

We wish to jointly estimate the binding configuration A, the concentrations T, and the ChIP parameters Φ. Given binding configuration A, we consider the sequence S and the ChIP windows D as independent sources of information. We further assume independent priors on T and Φ. Under these model assumptions, the joint posterior distribution is

$$\begin{aligned} P(\Phi, T, A|S, D, \Theta) &\propto P(S, D, A|\Theta, \Phi, T)\pi(\Phi, T) \\ &= P(S|A, \Theta)P(D|A, \Phi)P(A|T)\pi(\Phi)\pi(T), \end{aligned} \tag{2.1}$$

where $P(S, D, A|\Theta, \Phi, T)$ is the complete-data likelihood, regarding A as missing data. The posterior distribution of A gives the predicted binding landscape. Our primary goal is to select the DBFs with motif models in Θ that have a significant number

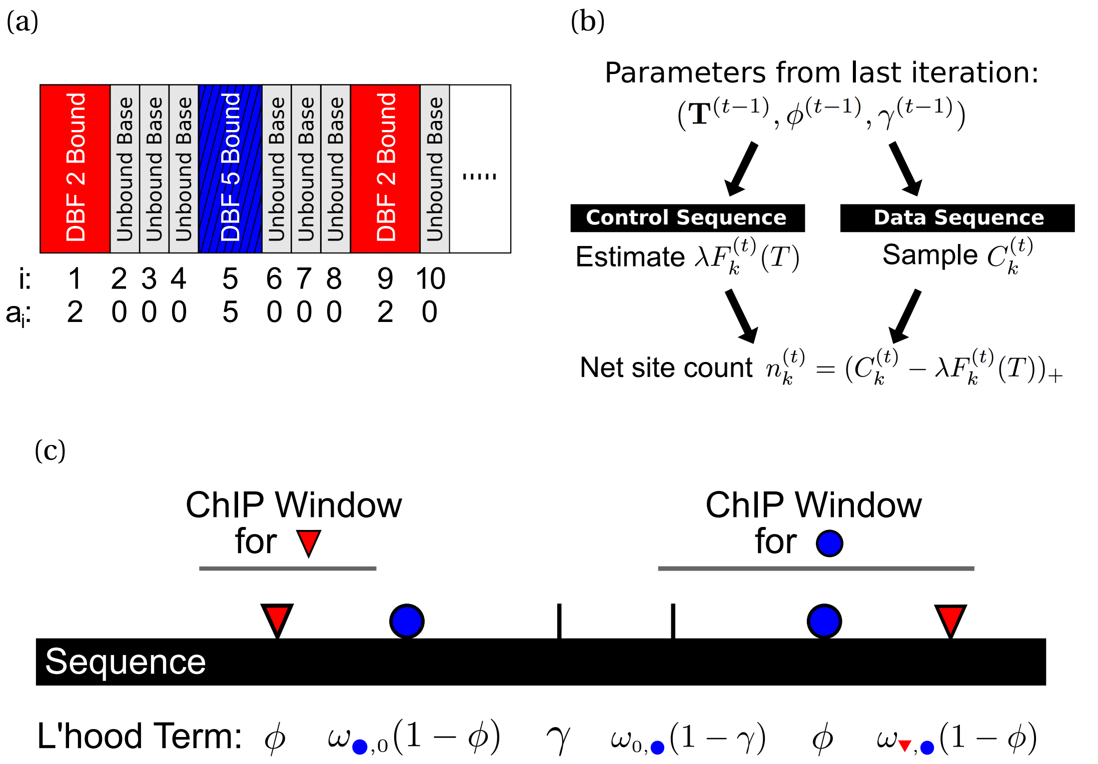

(a)
DBF 2 Bound
Unbound Base
Unbound Base
Unbound Base
DBF 5 Bound
Unbound Base
Unbound Base
Unbound Base
DBF 2 Bound
Unbound Base
.....
i: 1 2 3 4 5 6 7 8 9 10
a_i: 2 0 0 0 5 0 0 0 2 0
(b)
Parameters from last iteration:
$(\mathbf{T}^{(t-1)}, \phi^{(t-1)}, \gamma^{(t-1)})$
Control Sequence
Estimate $\lambda F_k^{(t)}(T)$
Data Sequence
Sample $C_k^{(t)}$
Net site count $n_k^{(t)} = (C_k^{(t)} - \lambda F_k^{(t)}(T))_+$
(c)
ChIP Window for ▼
ChIP Window for ●
Sequence
L'hood Term: ϕ $\omega_{●,0}(1-\phi)$ γ $\omega_{0,●}(1-\gamma)$ ϕ $\omega_{▼,●}(1-\phi)$

Figure 2.2: Elements of the SparScape model. (a) Vector A for a particular binding configuration. (b) Procedure for calculating the net site counts and penalties. (c) Terms contributed to the likelihood by unbound bps and binding of DBFs in different types of windows. The triangle and circle represent binding sites for two ChIP DBFs. The vertical bar represents an unbound site or a binding site for a DBF with no ChIP data.

of binding sites in S and predict the likely binding configurations. We achieve this by assigning a statistical weight based on Eq. (2.1) to every binding configuration and searching the space of possible configurations through Monte Carlo sampling. Note that different DBFs can bind to the same bps in different configurations, and thus, our sampling approach allows relatively large posterior binding probabilities for multiple DBFs for the same bp if BSs do overlap.

The first part of the statistical weight is the ratio between the likelihood of the sequence given a configuration A and the likelihood given the null configuration A_0 where no DBFs are bound. Let S_j be the base at position j and $S_{\text{Start}(i):\text{End}(i)}$ be the subsequence in S from the first to last bps covered by element a_i in the A. Define the single-element sequence likelihood ratio as

$$H_k(S_{\text{Start}(i):\text{End}(i)}) = \frac{P\left(S_{\text{Start}(i):\text{End}(i)}|\theta_k\right)}{P\left(S_{\text{Start}(i):\text{End}(i)}|\theta_0\right)}, \tag{2.2}$$

for $a_i = k \in \{1,\ldots,K\}$, where θ_k is the parameter (e.g., PWM) for the k^{th} binding model. For $a_i = k = 1$ we are considering the nucleosome. The numerator of Eq. (2.3) is then the model from [KMF09] described above. For $a_i = k \in \{2,\ldots,K\}$ we are considering non-nucleosome DBFs modeled by PWMs and we have

$$H_k\left(S_{\text{Start}(i):\text{End}(i)}\right) = \frac{\prod_{j=\text{Start}(i)}^{\text{End}(i)} \theta_k(j - \text{Start}(i) + 1, S_j)}{\prod_{j=\text{Start}(i)}^{\text{End}(i)} \theta_0(S_j|S_{j-1},\ldots,S_{j-5})}, \tag{2.3}$$

where $\theta_k(i,j)$ is the probability of nucleotide j at position i in a sequence bound by DBF k, and $\theta_0(S_j\left|S_{j-1},\ldots,S_{j-5}\right.)$ is the probability of nucleotide S_j in the background given the previous five nucleotides. By definition $H_0 \equiv 1$.

The likelihood ratio of a single element a_i, given only the concentrations, is simply the ratio of the concentrations τ_{a_i}/τ_0. This suggest the joint likelihood of the sequence S and the binding configuration A given the binding models Θ and concentrations T is

$$\frac{P(S|A,\Theta)P(A|T)}{P(S|A_0,\Theta)P(A_0|T)} \propto \prod_{i=1}^{|A|} \frac{\tau_{a_i}}{\tau_0} H_{a_i}(S_{\text{Start}(i):\text{End}(i)}). \tag{2.4}$$

The second part of the statistical weight is the ratio between the likelihood of the ChIP windows given a configuration A and the likelihood given the null configuration A_0. The background window is defined as the set of bps not covered by any of these windows. Thus if we have M ChIP DBFs, we will have $M+1$ sets or types of windows, where windows of type 0 are the background windows. Let $d_{\text{Start}(i):\text{End}(i)}$ be the type of the window covering the element a_i. When a_i indicates a ChIP DBF, we define

$$P(d_{\text{Start}(i):\text{End}(i)} = k|a_i = k,\phi) = \phi, \tag{2.5}$$

$$P(d_{\text{Start}(i):\text{End}(i)} = j|a_i = k,\phi) = \omega_{k,j}(1-\phi), \tag{2.6}$$

where $\omega_{k,j}$ $(j \neq k)$ is proportional to the total length of all windows of type j and $\sum_{j \neq k} \omega_{k,j} = 1$. For background sites ($a_i = k = 0$), ϕ is replaced by γ in (2.5) and (2.6). The model for a DBF without ChIP-Seq data is identical to that for background sites. See Figure 2.2(c) for an illustration. If most of the ChIP DBF binding sites are covered by a corresponding ChIP window, the parameter ϕ will be close to one. The value of γ is determined mostly by the percentage of the bps not covered by any ChIP windows. We have found that when running SparScape ignoring the ChIP data, the percentage of predicted binding sites within what would have been ChIP windows had they been considered tends to be quite similar across DBFs. Thus, we assume a single parameter ÏĘ shared among all ChIP DBFs in the current work.

SparScape ignores ChIP peak strength as we have found no correlation between ChIP peak strength and the strength of predicted binding near that peak. See [KLS11] and [KTP08] for other models of ChIP data in motif finding.

With all the terms in Eq. (2.1) defined we can compute the full likelihood ratio. Define $B(k,\ell) = P(d_{\ell:\ell'} | a_i = k, \Phi)$ with $\ell' = \ell + L_k - 1$. Then the full likelihood ratio for element $a_i = k$ in configuration A, starting at sequence position ℓ and covered by a window of type $d_{\ell:\ell'}$, is

$$\mathscr{L}(k,\ell) = (\tau_k/\tau_0)\; H_k(S_{\ell:(\ell+L_k-1)})\; B(k,\ell)/B(0,\ell). \tag{2.7}$$

Note that when $a_i = k = 0$, i.e., the i^{th} element is an unbound bp, by definition we have $\mathscr{L}(0,\ell) = 1$ for all ℓ. The product of $\mathscr{L}$ over a_i defines the complete-date likelihood ratio,

$$\frac{P(S,D,A|\Theta,\Phi,T)}{P(S,D,A_0|\Theta,\Phi,T)} = \prod_{i=1}^{|A|} \mathscr{L}(a_i, \text{Start}(i)). \tag{2.8}$$

2.2.3 Sparsity Through Penalization

The total number of candidate DBFs K is often large. We seek DBF selection because we expect that binding sites for a large majority of candidate DBFs are not enriched in the considered genomic regions. Considering DBFs that are not truly enriched when predicting the final landscape increases false positive predictions, sometimes dramatically.

One way to achieve DBF selection is to estimate many concentrations τ_k as exactly zero, as τ_k is the probability of initiating a binding site for DBF k. It can be seen from (2.8) that the log-likelihood for T given A is $\sum_{k=0}^{K} C_k \log \tau_k$, where C_k is the number of binding sites of DBF k in A. If we take the conjugate Dirichlet prior on T with prior counts $\alpha_k > 0$, for $k = 0, \ldots, K$, the conditional posterior distribution for T is a Dirichlet distribution $Dir(C_0 + \alpha_0, \ldots, C_K + \alpha_K)$. A sample from this distribution always has positive components, based on which we cannot construct a sparse estimation of T. Thus we run our algorithm in the selection stage for a burn-in period with $\alpha_k = 1$, and then set $\alpha_k = 0$. If $C_k = \alpha_k = 0$ at some iteration, then the conditional posterior distribution has a point mass at $\tau_k = 0$. This allows us to achieve sparsity in the sense that some $\tau_k = 0$ with a positive probability. DBFs that hit $\tau_k = 0$ at any sampling iteration are selected out.

Unfortunately, for most DBFs we expect relatively strong non-functional motif matches to occur randomly, leading to a non-negligible number of false positive predicted sites such that τ_k almost never hits zero for any DBF. We counteract this false positive signal with penalty terms on the parameters T and Φ, leading to a penalized complete-data log-likelihood of the form

$$\log P(S, D, A|\Theta, \Phi, T) - \lambda \sum_{k=2}^{K} F_k \log \tau_k - \rho(\Phi), \tag{2.9}$$

where $F_k \geq 0$ is the expected count of false positive binding sites for DBF k, $\lambda \geq 0$ is a tuning parameter, and $\rho(\Phi)$ denotes the penalty for Φ. Given the current concentrations, F_k is estimated empirically from a set of control sequences, possibly simulated, with no (known) true sites. We did not penalize the nucleosome concentration (τ_1) in the results presented here, but SparScape supports such penalization.

To estimate the expected false positive count for DBF k, F_k, we calculate the probability that each DBF is bound via a standard forwards-backwards summation on a set of control sequences with no (known) binding sites given the current parameters. These control sequences may either be simulated from the background model used in SparScape or sampled from the genome of interest from regions with similar characteristics to the regions of interest but with no known binding sites.

Calculating F_k for $k = 1, \cdots, K$ during every sampling iteration is computationally expensive, and is generally unnecessary, as the vector of concentrations tends not to shift dramatically over a small number of iterations, and the estimated values for each F_k do not change dramatically if the concentrations in T have not. Thus we only recalculate F_k every ten iterations. In the other iterations, F_k is estimated by a simple linear regression of the previous four estimates on the corresponding concentration values τ_k. We found this to be accurate and not to affect the posterior estimates.

Together with the prior distribution $\pi(\Phi, T)$, we obtain a penalized posterior distribution. To understand this penalized posterior, consider sampling from the conditional distributions taking the penalties into account. The conditional sampling from $[A|\Phi, T, S, D, \Theta]$ is not affected by the penalization and can be implemented by forward summation and backward sampling [GL03, ZW04]. Define the forward summation function

$$\begin{aligned} f_\ell &= P(S_{1:\ell}, D_{1:\ell}|\Theta, \Phi, T) \\ &= \sum_{A_{1:\ell}} P(S_{1:\ell}, D_{1:\ell}, A_{1:\ell}|\Theta, \Phi, T). \end{aligned} \tag{2.10}$$

Then, with $f_0 = 1$,

$$f_\ell = \sum_{k=0}^{K} f_{\ell - L_k} \mathscr{L}(k, \ell'), \tag{2.11}$$

where $\ell' = \ell - L_k + 1$ and $f_\ell = 0$ if $\ell < 0$ since we do not allow partial binding at the edge of regions. With $f_{|S|}$, we can backwards sample from the conditional posterior distribution of the DBF binding configuration $[A|\Theta, \Phi, T, S, D]$.

Consider penalized sampling from $[\Phi, T|A, S, D, \Theta]$. Let $n_k = C_k$ for $k = 0, 1$ and $n_k = (C_k - \lambda F_k)_+$ for $k \geq 2$, where $x_+ = \max(0, x)$. We think of n_k as the net site count after discounting for false positives λF_k. Then the penalized complete-data log-likelihood for the concentrations T is $\sum_{k=0}^{K} n_k \log \tau_k$. If the penalty $\lambda F_k \geq C_k$ for some iteration, then the net site count $n_k = 0$ and the conditional posterior distribution of T (with $\alpha_k = 0$) has a point mass at $\tau_k = 0$. Thereafter, we drop DBF k from further consideration. Figure **??** shows penalized and unpenalized sample paths for the concentrations of a true and a false DBF in a simulated data set. Figure **??**(a) demonstrates that moderate penalization can eliminate a false positive DBF even when its concentration would otherwise stabilize around a highly inflated value. While τ_k may hit zero exactly for many false positive DBFs even with Îż = Îśk = 0, as shown in these figures there are many DBFs that are not truly enriched where selection is achieved by using a nonzero penalty $\lambda > 0$. Setting the prior counts $\alpha_k = 0$ is mainly for mathematical rigor so that τ_k may hit zero exactly, as any positive value of Îśk will never give $\tau_k = 0$ in any sampling iteration. In addition, our choice also avoids the use of a threshold value on the estimated ÏĎk for DBF selection, which would be necessary if α_k were positive.

Penalization on the parameters in Φ is similar. The conditional posterior distributions for ϕ and γ given A are beta distributions. The counts in both are related to the site counts inside and outside the relevant set of ChIP windows. We penalize the counts for these beta distributions in the same manner as described above. Suppose we have sampled C_k sites for DBF k in the current iteration t, with $C_k^{(\text{in})}$ sites within a ChIP window for DBF k and $C_k^{(\text{out})}$ outside a ChIP window for DBF k. With no penalty ($\lambda = 0$), the conditional distribution of ϕ, the probability of a site for ChIP DBF j falling within a ChIP window of type j, is

$$\text{Beta}\left(\alpha^{(\text{in})} + \sum_{k=1}^{K} C_k^{(\text{in})}, \quad \alpha^{(\text{out})} + \sum_{k=1}^{K} C_k^{(\text{out})}\right), \tag{2.12}$$

where $\alpha^{(\text{in})} > 0$ and $\alpha^{(\text{out})} > 0$ are prior counts and $C_k^{(\text{in})} = C_k^{(\text{out})} = 0$ if DBF k has no ChIP-Seq data. With $\lambda > 0$, our total penalty on the count C_k is $\lambda F_k(T^{(t)})$. We penalize $C_k^{(\text{in})}$ and $C_k^{(\text{out})}$ proportionally with total penalty $\lambda F_k(T^{(t)})$. Let $n_k^{(\text{in})}$ and $n_k^{(\text{out})}$ denote the net counts for DBF k within and without ChIP windows, respectively. Then we have

$$
\begin{aligned}
n_k^{(\text{in})} &= \left[C_k - \lambda F_k(T^{(t)})\right]_+ \frac{C_k^{(\text{in})}}{C_k}, \\
n_k^{(\text{out})} &= \left[C_k - \lambda F_k(T^{(t)})\right]_+ \frac{C_k^{(\text{out})}}{C_k},
\end{aligned} \tag{2.13}
$$

which are used in place of $C_k^{(\text{in})}$ and $C_k^{(\text{out})}$ in the distribution (2.12).

Note that when $n_k + \alpha_k = 0$, the improper conditional distribution for T violates the reversibility assumption in Markov chain Monte Carlo. Preliminary theoretical exploration suggests that as $|S|$ increases, with mild conditions the probability of selecting out a DBF with no true sites goes to one while DBFs with true sites will still be selected. This confirms what we have found in practice, which is that selection of true DBFs via the sampling scheme outlined above is fairly robust. See Section 2.5 for a fuller exploration.

2.2.4 The Tuning Parameter

The tuning parameter λ is a scalar that controls the level of penalization on sampled site counts in each iteration of our algorithm. We developed a method for choosing λ similar to the work by [FZ13]. We run ten-fold cross-validation on a set of decreasing values of λ, say $\lambda_1 > \cdots > \lambda_i > \cdots > \lambda_M$, and record for each λ_i the DBFs selected in at least five of the ten training sets and the mean log-likelihood across the ten test sets. In general, a lower value of λ leads to a larger number of selected DBFs, a more complex model with a higher log-likelihood. Our goal is to choose the simplest model (highest value of λ) such that choosing a lower λ and selecting more DBFs do not improve the mean likelihood of the test sets sufficiently to justify the extra complexity. Let μ_i be the mean log-likelihood over the test sets and N_i be the number of DBFs selected for λ_i. Then we define the model selection rate for λ_i as $m_i = (\mu_i - \mu_{i-1})/(N_i - N_{i-1})$. We have observed good selection results if we choose the λ_i before the first decrease in m_i. A typical plot is provided in Figure 2.4.

2.2.5 Landscape Prediction

After enriched DBFs are selected, the binding landscape is predicted considering only the selected DBFs with prior counts $\alpha_k = 1$. Recall that a final run of our iterative sampling algorithm is necessary because with unknown concentrations calculating binding probabilities exactly through forward-backward summation is impossible.

In the DBF selection stage we penalize the binding site counts to encourage sparsity in concentration estimation. Overall, the penalty biases DBF concentration estimates downward and biases the predicted binding landscape towards higher specificity and lower sensitivity. One can increase the sensitivity of binding site prediction in the final stage by not penalizing. One may also consider re-estimating PWMs, using previously published PWMs as prior information. This can increase sensitivity but may result in capture in minor modes with some concentrations highly inflated. In such cases the landscape prediction with penalized site counts and fixed PWMs must be used despite the downward bias.

2.3 Results

2.3.1 Applications to Mouse ESC and Simulated Data

We investigate two mouse data sets derived from multiple TF loci (MTL) regions defined in [CXY08]. In that paper, 12 TFs known to play a key role in maintenance of ESCs were studied with ChIP-Seq experiments. These 12 TFs are Stat3, Nanog, Klf4, Pou5f1/Oct4, Esrrb, Sox2, cMyc, Smad1, nMyc, E2f1, Tcfcp2l1, and Zfx. Two main clusters of binding sites were found, one centered around Oct4 (Nanog, Sox2, Oct4, Smad1, Stat3) and the other around cMyc (cMyc, nMyc, E2f1, Zfx). We interrogate $1,553$ MTL regions built around the Oct4 group and $1,178$ regions built around the cMyc group. ChIP windows were built around ChIP peaks in these regions. Mouse ESC gene expression data from [ZCM07] were used to cull the considered DBFs to

170 with non-negligible expression. The PWMs of the 170 DBFs were extracted from the TRANSFAC database [MFG03]. We utilize SparScape to nominate novel possible co-factors that act in concert with these known core TFs in mouse ESCs.

Two simulated data sets, each composed of 1,000 simulated regions, were designed to mimic, respectively, the two mouse data sets. One thousand regions of 450 base pairs were simulated in each case. The background (unbound) base pairs were simulated from a 5$^{\text{th}}$-order Markov chain. The Markov chain parameters were estimated from a set of regions sampled to resemble the mouse data being mimicked. The set of distances to the nearest transciption start site (TSS) for the regions in the mouse data set were sampled by chromosome with replacement. For each distance to TSS sampled, a gene was chosen randomly from the same chromosome, and the region the sampled distance from that gene was added to the set of regions used to estimate the Markov chain parameters. This is also how the control regions used to estimate the expected number of false positive sites in each iteration were sampled in the mouse data analysis.

The number of nucleosome sites was chosen randomly between zero, one, and two. The sequence inserted at each nucleosome site was sampled from those reported in Kaplan *et al.*, 2009. The total number of DBF binding sites was chosen randomly with a minimum of two and a maximum of 10. That number of DBFs was then sampled randomly with replacement from the group of DBFs chosen for this simulation (10 in the cMyc group simulation, 11 in the Oct4 group simulation). The binding site sequences were simulated from the given PWM.

To evaluate performance across a range of sample sizes, for both the mouse and simulated data, the entire Oct4 group was analyzed together, but the cMyc group was randomly divided into subsets of 100 regions. DBFs were selected using only the first random subset and binding landscapes were predicted over all subsets with the selected tuning parameters and DBFs.

The most comparable method with available software is COMPETE [WH09]. COM-

PETE requires a fixed concentration vector and a pre-selected set of DBFs as input. We chose the concentration vector following the tuning procedure outlined in their paper. Since the primary source of information for both SparScape and COMPETE is the sequence, we also compared against a naive approach considering the raw binding score at each locus. This raw score is the ratio of the PWM score over the background model score for a given w-mer (2.3).

2.3.2 Comparing DBF Selection

Both COMPETE and the raw score method use only sequence information and do not use ChIP data like SparScape. To make a fair comparison, we first show that SparScape outperforms COMPETE and the raw score method when we ignore the available ChIP data and only utilize the sequence data. This illustrates the value of penalization and joint concentration estimation in SparScape. When the ChIP data are used, SparScape outperforms the competitors more dramatically.

To select DBFs using COMPETE, we ranked the DBFs by the total predicted binding probabilities over all bps and chose rank cutoffs for comparison with other methods. For the raw score method, when considered separately by locus, the number of scores for a DBF that exceed a chosen threshold (we used 1,000 and 2,000) approximately follows a Poisson distribution when there are no true sites. An expected false positive count of scores over this threshold was estimated from control regions and used as the rate parameter of the Poisson distribution to find a p-value for the hypothesis that the DBF had no true sites in the examined regions. We considered three approaches to controlling for multiple comparisons. Our most conservative method was the ranking method where we ranked all the DBFs by their p-values and considered only the top N, where N is the number of DBFs selected by SparScape. The next most conservative approach was the use of the standard Bonferroni multiple testing adjustment to control the family-wise error rate at 5%. The least conservative approach was controlling the false discovery rate (FDR) at 5%.

			RS	*RS*	*CO*	*SS-NC*	*SS*
		Cutoff	1000	2000	-	CV	CV
(A)	***Oct4***	*Rank*	2/11	4/11	0/11	10/11	11/13
	Group	*Bonf.*	10/140	10/127	11/127	-	-
		FDR	10/148	10/134	11/134	-	-
	cMyc	*Rank*	6/12	7/12	1/12	9/12	9/13
	Group	*Bonf.*	7/19	8/14	2/14	-	-
		FDR	8/26	9/22	4/22	-	-
(B)	***Oct4***	*Rank*	2/9	3/9	1/9	4/9	8/12
	Group	*Bonf.*	6/32	4/21	3/21	-	-
		FDR	7/48	6/32	3/32	-	-
	cMyc	*Rank*	0/5	0/5	0/5	2/5	5/11
	Group	*Bonf.*	7/78	7/54	2/54	-	-
		FDR	9/100	8/84	6/84	-	-

Table 2.1: DBF selection results for the raw score (RS) method, COMPETE (CO), SparScape with no ChIP data (SS-NC), and SparScape (SS). (A) simulated data and control. (B) mouse data and sampled control. In (A), T/N represents T true DBFs out of N DBFs selected. In (B), T/N represents T ChIP DBFs selected out of N DBFs selected. For COMPETE, the number of DBFs to select from the ranked list was chosen to match that of SparScape (the *Rank* row) or the raw score method (the *Bonf* and *FDR* rows). The *Rank* row gives ranked results for the raw score method and COMPETE that match the number selected by SS-NC. The *Bonf* row gives selection results using a Bonferroni corrected p-value threshold of 0.05. The *FDR* row gives selection results using an FDR of 5%. CV indicates that the DBFs were selected via cross-validation by SparScape.

Table 2.1 summarizes DBF selection results for SparScape, COMPETE, and the raw score method. Even in the best case for the alternate methods, SparScape provides more powerful and more accurate DBF selection, in the large (Oct4 group) and small (cMyc group) data sets and in the real and simulated data, using ChIP data or not. To achieve similar sensitivity for the factors we know are enriched, the other methods nominate between 4 and 10 times as many possible co-factors, fewer of which are plausible compared to those nominated by SparScape. SparScape also did a better job selecting the DBFs around which the examined regions were built, selecting 3 of the 5 members of the Oct4 group and all 4 members of the cMyc group. When choosing the same number of DBFs selected by SparScape, COMPETE or the raw score method selected at best one member of the Oct4 or cMyc groups in the respective sets of MTL regions.

When regions are analyzed to select target DBFs for lab-based follow-up experiments, it is of critical importance to reduce the number of false leads suggested by computational analysis. A major reduction in false positives in selecting DBFs is a key advantage of our new method. The substantial improvement over COMPETE highlights the critical roles of penalizing false positive counts and estimating concentrations in SparScape.

The joint DBF selection is a key factor in the improved performance achieved by SparScape. To demonstrate this point, we compared against an individual selection approach under the same framework of SparSpace but considering one candidate DBF at a time with the nucleosome. We applied this individual approach to the cMyc group mouse data and ran SparScape 10 times for each DBF with the same λ as chosen in our joint runs on that data set. ChIP windows were considered for the ChIP DBFs. This individual approach selected a total of 112 DBFs, 97 of them in all 10 runs, including 9 of the 12 ChIP DBFs. As expected, this result is close to that of the raw score method (with FDR control). As shown in Table 1, joint runs selected only 11 total DBFs including 5 ChIP DBFs. This highlights the huge reduction in the false positives for DBF selection that results from considering all the DBFs together.

Now we consider the possible co-factors without ChIP data selected by SparScape in the Oct4 and cMyc MTL regions.

In the Oct4 regions, SparScape selected Zfp219, Egr1/Krox24, Sp1, and Nr6a1/GCNF. All four have been identified in the literature as having some association with differentiation, ESCs, or being key regulators in maintenance of pluripotency. Zfp219 and Sp1 have been identified as members of protein interaction networks for pluripotency with Oct4 and Nanog in mouse ESCs [WRC06, KCS08]. Zfp281 has also been identified as a Nanog interacting protein required for proper cell differentiation [FSA11]. The binding motifs for Zfp219 and Zfp281 are very similar, so it is possible we are picking up sites for both TFs. Sp1 was also found to be a significant co-factor in this same data set by [HCH09]. The Egr and Sox families have been shown to interact in Schwann cells [JM08]. Nr6a1 is required to suppress Oct4 and recruit co-factors to affect DNA methylation and histone modifications in the Oct4 promoter during differentiation [GXL11].

In the cMyc group, we proposed Atf5, Erf, Rfxap, Nrf1, Sp1, and Zbtb7b/cKrox/ThPok. Atf4 (with a nearly identical motif as Atf5) has been identified in a co-expression and regulation network centered around cMyc [FYK08]. Erf is key in cell differentiation mediated by cMyc repression [VPV07]. Nrf1 has been shown to interact with cMyc in regulating apoptosis and implicated as a key actor in pluripotency maintenance [MFP09, MGH02]. Sp1 interacts with cMyc [HE10]. Zbtb7b directs the CD4 and CD8 T cell differentiation, while related TFs such as Miz-1 are known to cooperate with cMyc [KPF08].

Taken together, we see that SparScape was able to select DBFs known to bind in the regions of interest and nominate a small group of likely co-factors for both the Oct4 and cMyc groups. This can provide investigators with a higher return on experimental validation and follow-up studies.

2.3.3 The Binding Landscape and Concentration

In the second stage we predict the binding landscape and estimate concentrations for the DBFs selected in the first stage. The concentration estimates for the selected DBFs (SS in Table 2.1) in the mouse and simulated Oct4 group regions are shown in Figure 2.6. It is seen that without penalization ($\lambda = 0$), the concentrations were generally overestimated for the simulated dataset (Figure 2.6a). Particularly, the unpenalized estimate for Nanog does not appear in the figure, as it was massively overestimated, indicating an enormous number of false positive predicted binding sites. This is a general danger of jointly estimating the concentrations that can be avoided with proper penalization. This also explains, at least partially, why the estimated nucleosome concentration was lower in the unpenalized run than in the penalized run. With $\lambda = 0$, many nucleosome binding sites were crowded out by the false positive sites for Nanog and other DBFs due to the competing nature between DBF binding in our model. Furthermore, nucleosome concentration τ_1 is not penalized even when $\lambda > 0$. One sees that the ratio between an estimated concentration and the true value ranged from 0.71 to 1.46 for the non-nucleosome DBFs in the penalized run ($\lambda = 0.2$). Figure 2.6(b) shows the concentration estimates in the mouse Oct4 group data from a penalized run with fixed PWMs and an unpenalized run with re-estimated PWMs. Similar patterns are observed as those in the simulated data.

Concentration estimates are very stable over different runs of SparScape. Figure **??** shows a box plot of the coefficient of variation (standard deviation divided by the mean) for the posterior mean estimates of concentration over ten independent runs of SparScape predicting the final binding landscape for the cMyc group data. These ranged from near zero to just over 0.02 for the 11 non-nucleosome DBFs selected by our two-stage run of SparScape and was less than 0.05 for the nucleosome.

Examples of predicted binding landscapes are shown in Figure **??**. The posterior binding probabilities summarized there can be used directly in further analyses, but

		SparScape			***Raw Score***		
	DBF subset	*PP*	*Sens.*	*FDR*	*Cutoff*	*Sens.*	*FDR*
Oct4 Group	*All*	0.5	0.74	0.23	900	0.74	0.42
	Non-Nuc	0.5	0.71	0.16	1,250	0.71	0.35
	ChIP	0.5	0.62	0.20	700	0.63	0.54
	No ChIP	0.5	0.82	0.11	5,000	0.82	0.14
cMyc Group	*All*	0.6	0.70	0.30	4,000	0.70	0.40
	Non-Nuc	0.6	0.67	0.26	5,000	0.68	0.33
	ChIP	0.6	0.72	0.21	5,000	0.76	0.33
	No ChIP	0.6	0.60	0.32	4,000	0.60	0.37

Table 2.2: Sensitivity and FDR in binding site prediction in the simulated data Non-Nuc is all considered DBFs except for the nucleosome. ChIP is the set of DBFs for which ChIP data were available. No ChIP is the set of DBFs for which ChIP data were not available. In the cMyc group simulation there were 10 small data sets. Combined results are reported here but the predictions were made separately in each set. PP stands for posterior probability cutoff.

we demonstrate the effectiveness of our method by choosing a posterior probability cutoff for predicting sites. A summary of results for SparScape and raw score predictions in the simulated data sets is given in Table 2.2. COMPETE results are not included in the table because the highest sensitivity achieved (< 0.1) was so low that the results are not comparable in this format. We only present results for the DBFs with true sites, ignoring in the FDR calculations the large number of false positive DBFs selected by the raw score method and those selected by SparScape.

For each category of DBFs, we chose the raw score cutoff that gave a similar sensitivity to that achieved by our method at the given posterior probability cutoff and compared the FDRs. SparScape reduces the FDR compared to the raw score method for every category of DBFs across both data sets, achieving a reduction of 21% to 63% in the Oct4 group data and 13% to 33% in the cMyc group. In the Oct4 simulation, we achieved an overall sensitivity of 0.74 with an FDR of 0.23 as compared to an FDR of 0.42 for the raw score method with similar sensitivity. In the cMyc simulation, we achieved a sensitivity of 0.7 and an FDR of 0.3, compared to an FDR of 0.4 for the raw score method. The raw score method performed relatively better on the DBFs with no ChIP data because those DBFs have, on average, stronger and more informativ motifs. In addition, SparScape utilizes ChIP data in a principled way, which led to more substantial improvement over the raw score method on the ChIP DBFs. Both methods predicted nucleosome sites with an FDR slightly below and above 50%, respectively, across a range of sensitivities. See Figure 2.8 for further details on performance of site prediction with a wide range of posterior probability cutoffs.

For the mouse datasets, true binding sites are not annotated, and therefore, we compared different methods based on the numbers of predicted binding sites for a ChIP DBF inside and outside its ChIP windows. The MTL regions were chosen by requiring multiple ChIP peaks from a small set of TFs to occur very close to each other, making it much less likely that peaks in these regions are false positives. Likewise, we expect few true binding sites further than 50 bps from a ChIP-Seq peak (the coverage of our ChIP windows) but within a few hundred bps. So we expect that the ChIP windows capture a very high percentage of true binding sites in these regions.

As reported in Table 2.3, for the selected ChIP DBFs, SparScape predicted a very high percentage of binding sites in a corresponding ChIP window for different cutoffs on the posterior binding probabilities. For example, in the Oct4 group data we predicted 317 to 3,103 binding sites in a matching ChIP window for the eight selected ChIP DBFs, with only 9 to 56 sites outside. Since the performance of the raw score method and COMPETE in DBF selection was unsatisfactory, we report their results on the sites predicted for all 12 ChIP DBFs (Table 2.3). One clearly sees a much higher percentage of binding sites predicted outside ChIP windows. In the most sensitive cases, SparScape predicted 77 – 97% of sites for ChIP DBFs within a corresponding ChIP window despite not restricting site prediction to within ChIP windows, while only 25 – 30% of sites predicted by the raw score method and 20% of sites predicted by COMPETE fell within a corresponding ChIP window. This suggests that a very high percentage (> 70%) of the binding sites predicted by COMPETE and the raw score method are false positives, demonstrating that for the questions we consider, our method is more sensitive and vastly more specific than the alternatives.

Although the joint estimation of DBF concentrations and binding landscape requires computationally expensive iterative sampling, SparScape is reasonably fast. SparScape is implemented in C++ with OpenMP, so runs most quickly on boards with more CPUs. We tested computation time with 10 independent runs on a dataset consisted of just over 52,000 bps in 100 regions and 11 DBFs, including the nucleosome. A total of 1250 iterations took on average 2,220 seconds (37 minutes, with minimal variation) on a MacBook Pro running OS X 10.6.8 with 2.53 GHz Intel core 2 duo processors and 4 GB of RAM.

2.4 Application to Gene Promoters

SparScape is not designed to carry out DBF selection on entire chromosomes or the whole genome. The enrichment of BSs for any DBF other than the nucleosome is too thin and selection results may be too sparse or inconsistent. For prediction of a binding landscape over an entire chromosome or genome we recommend running SparScape on the entire DBF library without selection and with a non-zero but small value of λ to prevent concentration inflation of DBFs with low-information or GC-rich motifs. SparScape can, however, select DBFs effectively on data less specialized than co-bound regions such as the Oct4 and cMyc groups explored above.

We demonstrate this by randomly sampling 2,000 mouse genes and running SparScape on the upstream 1,000 bps of these genes, using upstream 4,001âĹŠ5,000 bps as the

		SparScape		***COMPETE***		***Raw Score***		
	PP	*In*	*Out*	*In*	*Out*	*Cutoff*	*In*	*Out*
Oct4	0.25	3,103	56	148	584	500	1,168	3,485
Group	0.4	2,150	39	3	7	1,000	662	1,789
	0.6	1,074	25	0	0	2,000	438	1,110
	0.8	317	9	0	0	4,000	254	583
cMyc	0.25	1,328	387	12	540	500	1,311	2,930
Group	0.4	861	181	0	49	1,000	979	2,136
	0.6	497	61	0	0	2,000	707	1,500
	0.8	256	13	0	0	4,000	442	951

Table 2.3: binding site prediction inside and outside matching ChIP windows PP stands for posterior probability cutoff. Number of sites predicted by SparScape, COMPETE, and the raw score method for the ChIP DBFs inside and outside the corresponding ChIP windows. For SparScape only the ChIP DBFs selected by SparScape are considered. All 12 ChIP DBFs are considered for the raw score method and COMPETE. PP stands for posterior probability cutoff.

control regions. The candidate library consisted of all 203 DBFs in our library with unique PWMs. We ran 10-fold cross validation to select $\lambda = 0$ and the DBFs that survived selection in at least five folds. We selected 45 DBFs with $\lambda = 0$ and 41 with $\lambda = 0.05$ (the value used in the Oct4 group data). This represents 22% of the DBFs considered, a very reasonable number to be enriched in a random sample of 9% of gene promoters from the mouse genome. Table **??** shows the DBFs chosen with these two values of λ ranked by posterior mean concentration. One indication that SparScape selected truly enriched proteins is the presence of TATA box binding protein (TBP) in both lists (ranked ninth with the selected $\lambda = 0$), since the TATA box is known to be close to the transcription start site.

Table 2.4: Selected DBFs ranked by posterior mean concentration for the 2,000 upstream gene promoters sample from the mouse genome for the chosen $\lambda = 0$ and for $\lambda = 0.05$, the value chosen in our other large data sets, the Oct4 and cMyc group MTL regions.

Rank	DBF ($\lambda = 0$)	τ ($\lambda = 0$)	DBF ($\lambda = 0.05$)	τ ($\lambda = 0.05$)
1	sox10	1.494597e-03	sox10	9.642179e-04
2	pgr	1.023095e-03	pgr	9.023347e-04
3	dbp	8.192535e-04	foxa1	5.684778e-04
4	ar	7.627285e-04	dbp	4.543683e-04
5	foxm1	6.888999e-04	pitx2	4.386284e-04
6	tead1	5.896015e-04	tef	4.370527e-04
7	pou5f1	5.825057e-04	pax6	3.964371e-04
8	foxa1	5.740178e-04	ar	3.898316e-04
9	tbp	5.339879e-04	ikzf2	3.706031e-04
10	pitx2	4.755155e-04	foxm1	3.705634e-04
11	tef	4.601244e-04	pou1f1	3.669208e-04
12	ikzf2	4.426942e-04	irf9	3.338152e-04
13	pax6	4.115000e-04	tcfe3	2.978508e-04
14	irf9	3.922964e-04	cdx1	2.922822e-04
15	pou1f1	3.738187e-04	tbp	2.464555e-04
16	tcfe3	3.459292e-04	foxo3a	1.992414e-04
17	gata1	3.349708e-04	foxh1	1.886101e-04
18	en1	3.072128e-04	gata1	1.619605e-04
19	cdx1	2.956426e-04	nkx3-1	1.496764e-04
20	foxo3a	2.260526e-04	pax1	1.350059e-04
21	foxh1	2.163760e-04	tead1	1.323815e-04
22	nkx3-1	1.654200e-04	ppara	1.134400e-04

23	pax1	1.531719e-04	dmrta2	9.747900e-05
24	yy1	1.405842e-04	rreb1	9.583115e-05
25	smad5	1.350452e-04	smad5	8.601014e-05
26	ppara	1.285215e-04	yy1	8.259094e-05
27	ascl1	1.128295e-04	nr5a2	8.178382e-05
28	hmga1	1.104286e-04	foxj2	7.822326e-05
29	rreb1	1.080496e-04	onecut1	6.802819e-05
30	bptf	1.051290e-04	ascl1	6.362322e-05
31	nr5a2	1.041484e-04	zeb1	5.301816e-05
32	dmrta2	1.010508e-04	zfp219	5.078735e-05
33	onecut1	8.316074e-05	lef1	5.019112e-05
34	zfp219	6.810904e-05	dmrta1	3.553469e-05
35	lef1	6.703571e-05	foxl1	3.241955e-05
36	foxj2	6.337091e-05	prdm1	2.591156e-05
37	fosb	5.938101e-05	lhx3	2.348764e-05
38	zeb1	5.775988e-05	ctcf	2.325049e-05
39	esrrb	5.249724e-05	nkx6-1	1.780572e-05
40	dmrta1	4.089696e-05	esrrb	1.771331e-05
41	lhx3	2.798376e-05	evi1	1.155078e-05
42	nfe2l2	2.334501e-05	·	·
43	prdm1	2.165738e-05	·	·
44	ctcf	2.145135e-05	·	·
45	evi1	1.234990e-05	·	·

2.5 Theoretical Exploration: Convergence, Stationary Points, and Component Selection

In our preliminary theoretical analysis we explore the simplified case where we consider a mixture model with a $K+1$ components with known densities where we are interested in probabilistically classifying observations through an iterative sampler as coming from a component of interest, $k = 1, \cdots, K$, or from a (presumed dominant) background distribution ($k = 0$). Let $\tau^\star$ be the vector of true mixing proportions and $\hat{\tau}$ be the stationary point of the expected value recursion for τ given starting values $\tau^{(0)}$. We are interested in the properties of convergence of the expected value recursion to the stationary point $\hat{\tau}$ and in whether there exists non-trivial ($\hat{\tau} \neq (1, \mathbf{0}_K)$) stationary points when $\tau^\star \neq (1, \mathbf{0}_K)$. Let

$$\begin{aligned}
\mathscr{L}^{(t)}(x) &= \sum_{k=0}^{K} \tau_k^{(t)} f_k(x) \\
\mathscr{L}^\star(x) &= \sum_{k=0}^{K} \tau_k^\star f_k(x) \\
\hat{\mathscr{L}}(x) &= \sum_{k=0}^{K} \hat{\tau}_k f_k(x).
\end{aligned}$$

be the standard mixture density over $K+1$ dictionary components with mixture proportions equal $\tau^{(t)}$, the values from the t^{th} iteration, $\tau^\star$, and $\hat{\tau}$, respectively. Then for $k \neq 0$ we have

$$\begin{aligned}
E(\tau_k^{(t+1)}) &= \int_{\mathscr{X}} \frac{\tau_k^{(t)} f_k(x)}{\mathscr{L}^{(t)}(x)} \mathscr{L}^\star(x) \mathrm{d}x \; - \; \lambda_n \int_{\mathscr{X}} \frac{\tau_k^{(t)} f_k(x)}{\mathscr{L}^{(t)}(x)} f_0(x) \mathrm{d}x \\
&= \tau_k^{(t)} E_{f_k} \left(\frac{\sum_{j=0}^{K} \tau_j^\star f_j(x) - \lambda_n f_0(x)}{\mathscr{L}^{(t)}(x)} \right) \qquad (2.14) \\
&= \tau_k^{(t)} E_{f_k} \left(\frac{\mathscr{L}^\star(x) - \lambda_n f_0(x)}{\mathscr{L}^{(t)}(x)} \right). \qquad (2.15)
\end{aligned}$$

2.5.1 Convergence

For notational convenience let $E_i(x) = E_{f_i}(x)$, $f_i = f_i(x)$, $\mathscr{L}^\star = \mathscr{L}^\star(x)$, and so on. Then in vector form we have

$$\begin{aligned} M(\tau^{(t)}) &= E(\tau^{(t+1)}|\tau^{(t)}) \\ &= \begin{pmatrix} \tau_0^{(t)} E_0(\mathscr{L}^\star / \mathscr{L}^{(t)}) + \lambda_n \sum_{k=1}^{K} \tau_k^{(t)} E_k(f_0/\mathscr{L}^{(t)}) \\ \tau_1^{(t)} E_1\left[(\mathscr{L}^\star - \lambda_n f_0)/\mathscr{L}^{(t)}\right] \\ \vdots \\ \tau_K^{(t)} E_K\left[(\mathscr{L}^\star - \lambda_n f_0)/\mathscr{L}^{(t)}\right] \end{pmatrix}. \end{aligned} \tag{2.16}$$

With $M(\hat{\tau}) = \hat{\tau}$ and $M'(\tau)$ denoting the Jacobian matrix of $M(\tau)$, a first order Taylor expansion around $\hat{\tau}$ gives us

$$\begin{aligned} M(\tau^{(t)}) &= M(\hat{\tau}) + M'(\hat{\tau})(\tau^{(t)} - \hat{\tau}) \\ \tau^{(t+1)} - \hat{\tau} &= M'(\hat{\tau})(\tau^{(t)} - \hat{\tau}) \\ \epsilon^{(t+1)} &= M'(\hat{\tau})\epsilon^{(t)}, \end{aligned}$$

and like the EM algorithm we have a linear convergence rate dominated by the largest eigenvalues of $M'(\hat{\tau})$.

2.5.2 Stationary Points

Given linear convergence to $\hat{\tau}$, we are interested whether we can achieve a non-trivial $\hat{\tau}$. For this we must have some components k where $k \neq 0$ and $E(\hat{\tau}_k) = \hat{\tau}_k \neq 0$. This leads to

$$\hat{\tau}_k = \hat{\tau}_k E_k\left(\frac{\mathscr{L}^\star - \lambda_n f_0}{\hat{\mathscr{L}}}\right).$$

Given $\hat{\tau}_k \neq 0$, we need

$$\begin{aligned} 1 &= E_k\left(\frac{\mathscr{L}^\star - \lambda_n f_0}{\hat{\mathscr{L}}}\right) \\ &= \alpha_k^{(t)} \end{aligned} \tag{2.17}$$

for $E(\tau_k^{(t+1)}) = \tau_k^{(t)} = \hat{\tau}_k$.

Note that (2.15) gives us the condition for whether we expect $\tau_k^{(t+1)}$ to increase or decrease relative to $\tau_k^{(t)}$. If $\alpha_k^{(t)} > 1$, then $E(\tau_k^{(t+1)}) > \tau_k^{(t)}$. If $\alpha_k^{(t)} < 1$, then $E(\tau_k^{(t+1)}) < \tau_k^{(t)}$. Thus, if $\exists k \neq 0$ where $\alpha_k^{(0)} > 1$, we expect τ_k to increase from an initial value $\tau_k^{(0)}$ that was too low.

From (2.17) we can find an estimate of our non-trivial stationary point. In vector form our condition for stationarity around $\hat{\tau}$ is

$$\begin{aligned} \mathbf{1}_{K+1} &= \begin{pmatrix} E_0(\mathscr{L}^\star/\hat{\mathscr{L}}) + \lambda_n \sum_{k=1}^{K} E_k(f_0/\hat{\mathscr{L}}) \\ E_1(\mathscr{L}^\star/\hat{\mathscr{L}}) - \lambda_n E_1(f_0/\hat{\mathscr{L}}) \\ \vdots \\ E_K(\mathscr{L}^\star/\hat{\mathscr{L}}) - \lambda_n E_K(f_0/\hat{\mathscr{L}}) \end{pmatrix} \\ &= G(\hat{\tau}) \\ &= \begin{pmatrix} G_0(\hat{\tau}) \\ G_1(\hat{\tau}) \\ \vdots \\ G_K(\hat{\tau}) \end{pmatrix} \\ &= G^{(1)}(\hat{\tau}) - \lambda_n G^{(2)}(\hat{\tau}). \end{aligned}$$

Let $\hat{\tau} = \tau^\star - \Delta$. We expect Δ to be small as $n \to \infty$ and $\lambda_n \to 0$. Then we can approximate $\hat{\tau}$ with a Taylor expansion around $\tau^\star$. Let $G^{(1)\prime}(\tau^\star)$ be the Jacobian matrix for

$G^{(1)}(\tau^\star)$, with typical element

$$\begin{aligned} G^{(1)\prime}(\tau^\star)_{ij} &= \left.\frac{\partial G_i^{(1)}(\tau)}{\partial \tau_j}\right|_{\tau=\tau^\star} \\ &= \left.\frac{\partial}{\partial \tau_j} E_i\left(\frac{\mathscr{L}^\star}{\mathscr{L}}\right)\right|_{\tau=\tau^\star} \\ &= -E_i\left.\left(f_j \frac{\mathscr{L}^\star}{\mathscr{L}^2}\right)\right|_{\tau=\tau^\star} \\ &= -E_i\left(\frac{f_j}{\mathscr{L}^\star}\right) \\ &= -E_j\left(\frac{f_i}{\mathscr{L}^\star}\right), \end{aligned} \tag{2.18}$$

implying that $G^{(1)\prime}(\tau^\star)$ is symmetric, with all negative elements, and should only be singular if there are pathological relationships between the elements of the dictionary. Then

$$\begin{aligned} \mathbf{1}_{K+1} = &\begin{pmatrix} E_0(\mathscr{L}^\star/\mathscr{L}^\star) \\ E_1(\mathscr{L}^\star/\mathscr{L}^\star) \\ \vdots \\ E_K(\mathscr{L}^\star/\mathscr{L}^\star) \end{pmatrix} - \lambda_n \begin{pmatrix} -\sum_{k=1}^K E_k(f_0/\mathscr{L}^\star) \\ E_1(f_0/\mathscr{L}^\star) \\ \vdots \\ E_K(f_0/\mathscr{L}^\star) \end{pmatrix} \\ &- G^{(1)\prime}(\tau^\star)\Delta + \lambda_n G^{(2)\prime}(\tau^\star)\Delta + O(\|-\Delta\|^2) \\ = \mathbf{1}_{K+1} - \lambda_n &\begin{pmatrix} -\sum_{k=1}^K E_k(f_0/\mathscr{L}^\star) \\ E_1(f_0/\mathscr{L}^\star) \\ \vdots \\ E_K(f_0/\mathscr{L}^\star) \end{pmatrix} - G^{(1)\prime}(\tau^\star)\Delta + O(\|-\Delta\|^2) \\ \Rightarrow \Delta \approx -\lambda_n (G^{(1)\prime}(\tau^\star))^{-1} &\begin{pmatrix} -\sum_{k=1}^K E_k(f_0/\mathscr{L}^\star) \\ E_1(f_0/\mathscr{L}^\star) \\ \vdots \\ E_K(f_0/\mathscr{L}^\star) \end{pmatrix}. \end{aligned} \tag{2.19}$$

We expect the inversion required for (2.19) to be feasible, as $G^{(1)\prime}(\tau^\star)$ is symmetric with

$$G^{(1)\prime}(\tau^\star)_{ij} = G^{(1)\prime}(\tau^\star)_{ji} = -\int \frac{f_i f_j}{\mathscr{L}^\star} \mathrm{dx},$$

where one would expect this matrix to be singular only with a dictionary with pathological relationships between components.

2.5.3 Component Selection

The parameter space for mixture proportions is $\Omega = [0,1]^{(K+1)}$, and our goal is efficient selection of components with true signal. Given a large dictionary ($K >> 0$), where we expect true signals from a small number of components, our desired estimate is on the boundary of Ω in many dimensions, and thus the standard proof of $\sqrt{n}$ consistency for the MLE does not apply. However, in (Self and Liang, 1987) it was shown that under mild conditions, with sample size n, the MLE is $\sqrt{n}$ consistent even in cases where the MLE is on the boundary of the parameter space. Thus even in the case of interest where for many $k \neq 0$ the true parameter value $\tau_k^\star = 0$, we expect the unpenalized MLE $\hat{\tau}_k \in O(1/\sqrt{n})$. Also, in any interesting case, with finite n, in the unpenalized case $P(\hat{\tau}_k = 0) = 0$, and if we expect the background component to dominate we could easily have true components with $\tau_k^\star \in O(1/\sqrt{n})$ for some given finite sample size n.

In our penalized case, define $M(\tau_k^{(t)}) = M(\tau_k^{(t)})_+ = \max(M(\tau_k^{(t)}), 0)$ for $k \neq 0$ with a suitable adjustment for $M(\tau_0^{(t)})$. If

$$\begin{array}{lrcl} \text{(i)} & \lambda_n & \rightarrow & 0 \\ \text{(ii)} & \sqrt{n}\lambda_n & \rightarrow & \infty, \end{array}$$

then for k such that $\tau_k^\star = 0$, $P(\hat{\tau}_k = 0) \rightarrow 1$ at a rate $O(\sqrt{n}/\lambda_n)$. Thus our penalized method not only allows $P(\hat{\tau}_k = 0) > 0$, we can achieve a sparse selection of components at a faster rate than with widely used *ad hoc* approaches such as selecting out components with $\hat{\tau}_k < C$ for some cutoff value C. Note only can our method converge faster and select out false components, there are also cases where we can select out false components ($\tau^{\star}_k = 0$) with relatively low information densities with relatively high KL distance to the background distribution where the cutoff approach will always fail to find $\hat{\tau}_k < C$ with any reasonable, finite n.

2.6 Discussion

Generating a bp specific binding landscape for all known DBFs over the entire genome of higher organisms with large genomes, without prior knowledge of DBF concentrations, is currently prohibitively computationally expensive. When one then considers only certain genomic regions, it is expected that a large proportion of DBFs will not have true binding sites, and performance suffers if all DBFs are considered. Requiring prior knowledge of the set of DBFs active in the regions of interest and their concentrations introduces the need for (often ad hoc) user-dependent prior or iterative analysis or a need for additional experimental data. We contribute a new method, SparScape, that eliminates the need for this prior information and takes advantage of binding data where available while outperforming alternate methods. One of the key features of our method is the inclusion of penalization in Bayesian inference based on the expected false positive site counts. This significantly reduces false positive results both in DBF selection and in binding site prediction. It also introduces a level of robustness for model extensions. If an extension introduces more false positives, that will be counteracted by penalization. A similar idea has been used in the contrast motif finder [MPZ10]. The unsatisfactory performance of COMPETE for the problems we investigate suggests that tuning prior DBF concentrations when they are not given is difficult in practice and using an improper vector of concentrations can be very risky for joint prediction of landscapes. Moreover, both SparScape and the raw score method select DBFs by comparing against some control regions, but COMPETE does not have this critical component. We want to stress that our comparison against COMPETE is mostly for demonstration purpose as the method is not targeted at DBF selection or predicting sparse binding landscapes.

As demonstrated by the results in this paper, with the default method for choosing the level of penalization, a standard usage of SparScape usually works well. But in fact, SparScape is highly flexible. A user could independently choose less stringent DBF selection by setting a smaller penalty value and perform a single selection run, instead of ten-fold cross validation. This option is particularly useful when the goal is to predict sites for all DBFs that could at all plausibly have binding sites in the considered regions. The default choice in the final post-selection landscape prediction is to predict with the same penalty chosen in the selection stage and with fixed PWMs. A user may choose to make the predicted landscape less sparse by running the final prediction with no penalty, re-estimated PWMs, or both.

Our modeling framework can easily be extended to include other types of information. Some measure of absolute binding affinity, as opposed to the relative binding affinity information in a normalized PWM, is available from protein binding microarrays [BPQ06] and could be included as energy models [DSS03, FMB06, Zho10]. Further work integrating ChIP peak strength in the scoring of binding in ChIP windows could be fruitful, especially for nucleosomes. Other location-specific information on DBF binding to the sequence could also be utilized. We plan further work employing SparScape's joint binding landscape to predict gene expression and offer insights into the underlying regulatory network.

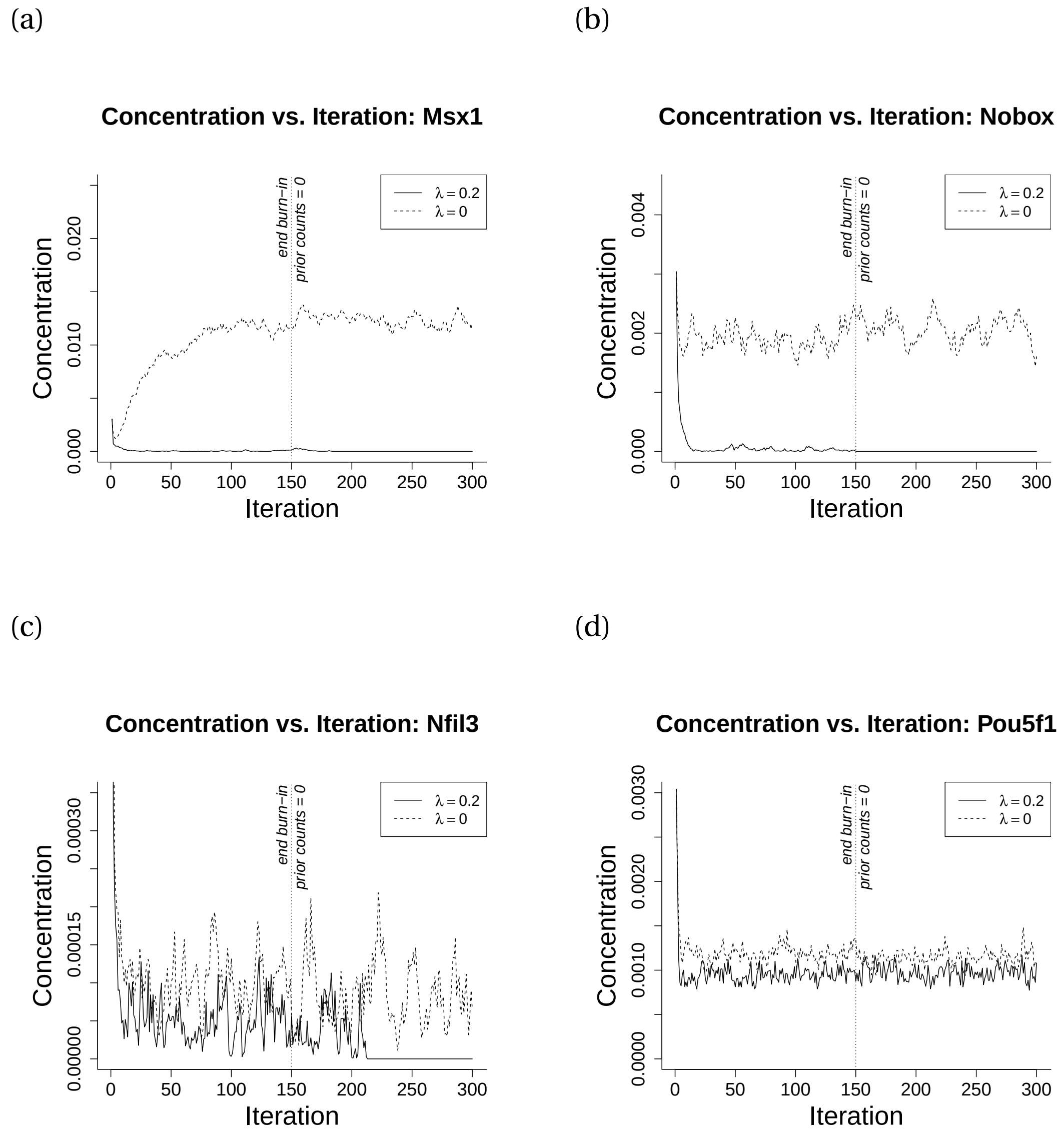


Figure 2.3: Sample paths for the concentration of a DBF in the selection stage in a simulated data set for a penalized run ($\lambda = 0.2$) and an unpenalized run ($\lambda = 0$). The dotted vertical line indicates the iteration at which the prior counts (α_k) were reduced from one to zero. (a), (b), and (c) are for DBFs with no true sites. With no penalty ($\lambda = 0$), concentration estimates can be massively inflated, as in (a), stable at a reasonable value, as in (b), or relatively stable at a very low value, as in (c). In all three of these cases, 20% penalization ($\lambda = 0.2$) pushed the concentration estimates to zero and eliminated these false positive DBFs. (d) shows the sample path for a DBF with true binding sites and illustrates the fact that concentration estimates for true positive DBFs are generally quite stable under non-excessive penalization.

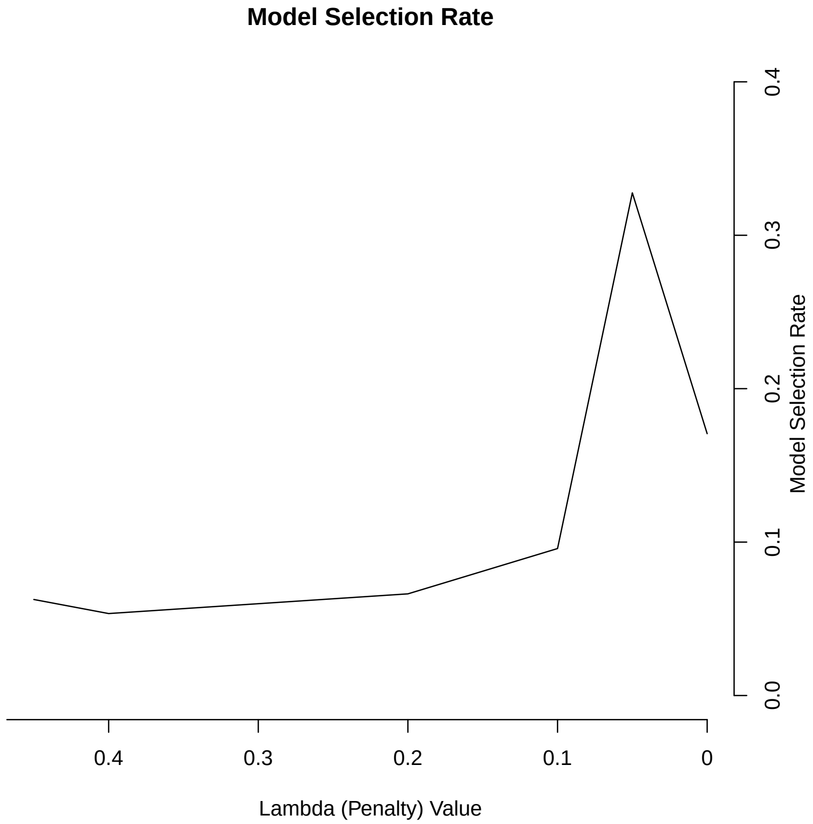


Figure 2.4: Example model selection rate graph. We consider decreasing values of the penalty λ until the first drop in the rate. In this case, the value of λ chosen was 0.05.

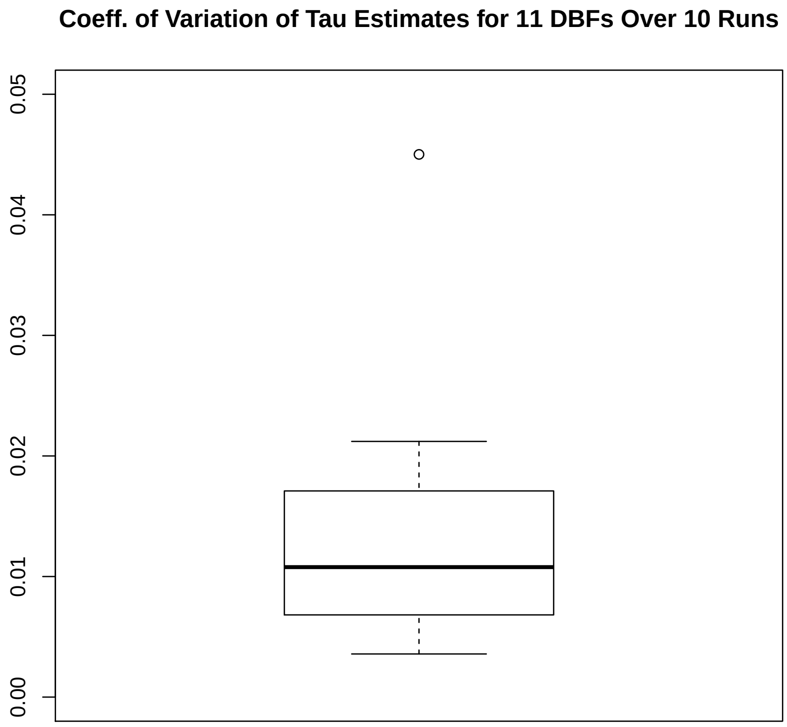


Figure 2.5: Coefficient of variation (σ/μ) of the posterior mean estimates of DBFs over 10 runs on the same genomic sequences with the same DBFs. The outlier is he nucleosome.

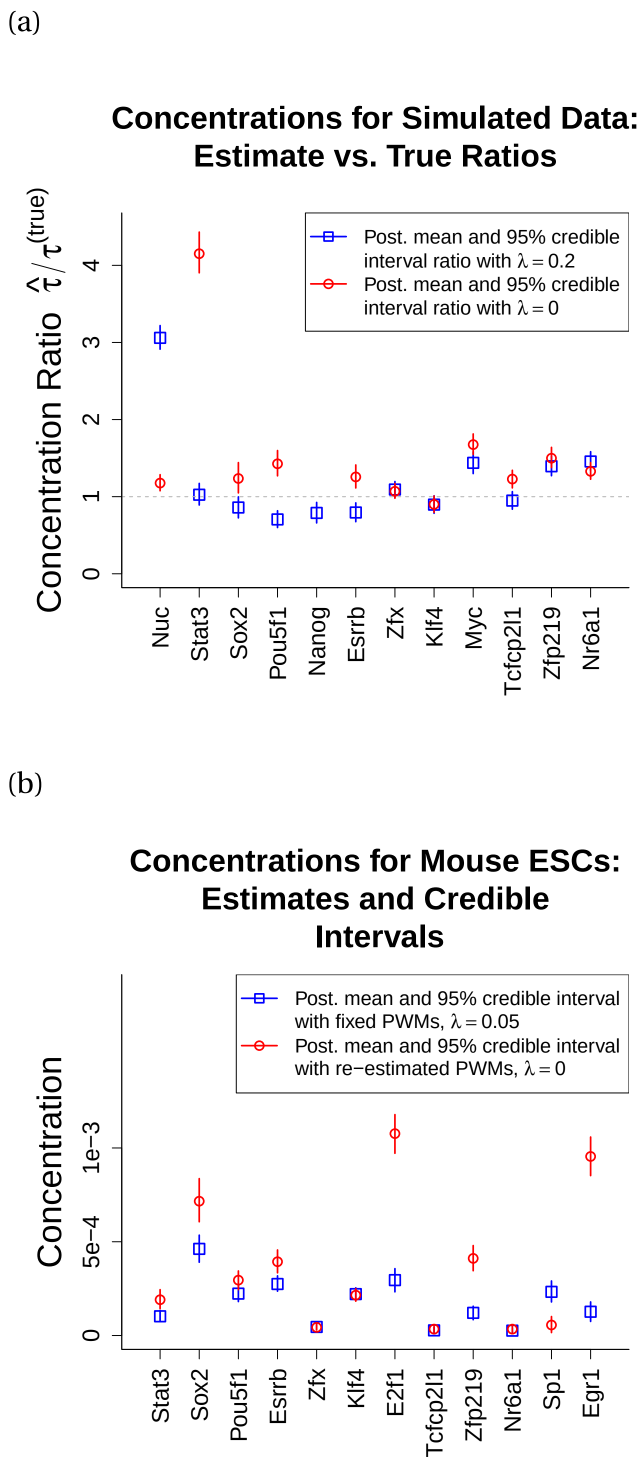
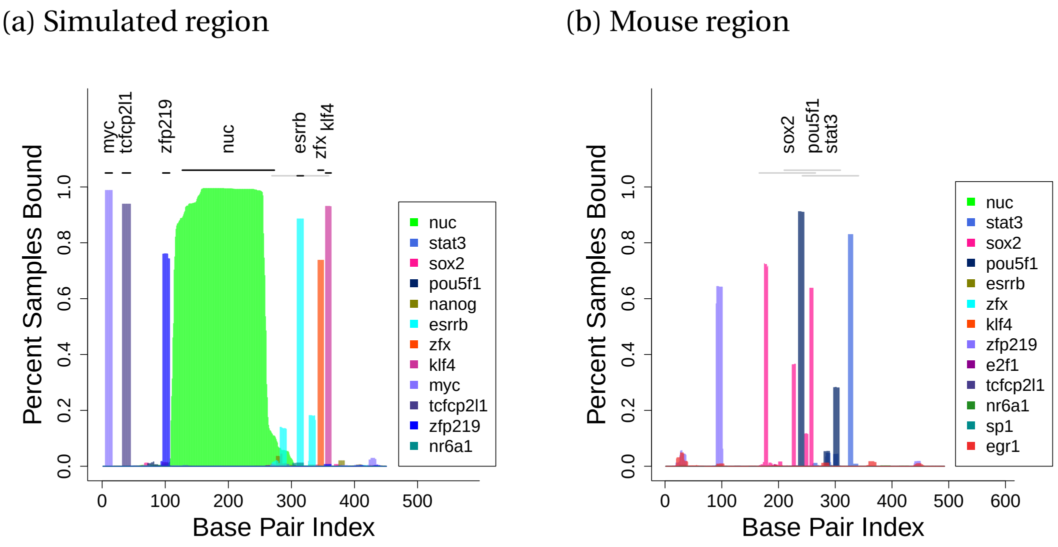


Figure 2.6: Concentration estimates. (a) Ratios of the posterior mean concentration estimates and 95% credible intervals over the corresponding true concentrations for the correctly selected DBFs in the Oct4 group simulated data. (b) Posterior means and 95% credible intervals for the selected DBFs in the Oct4 group mouse data. Penalized estimates ($\lambda = 0.2$) are squares. Unpenalized estimates ($\lambda = 0$) are circles.


(a) Simulated region

(b) Mouse region

Figure 2.7: Predicted binding landscapes from a simulated and a mouse region

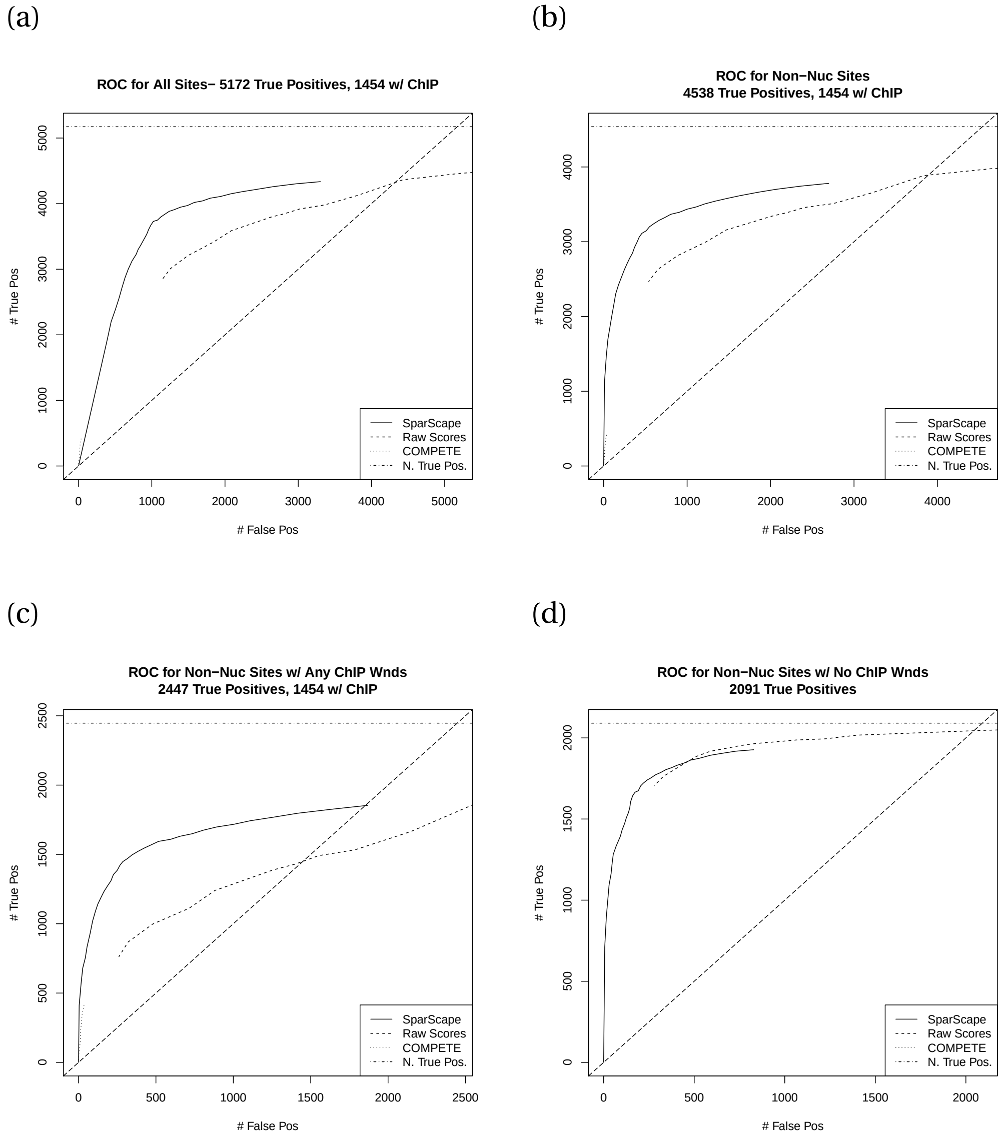


Figure 2.8: ROC style curves for binding site prediction in the Oct4 group simulated data, plotting false predicted site count versus true predicted site count for SparScape, COMPETE, and raw scores. (a) shows results for all DBFs. (b) shows results for all DBFs except the nucleosome. (c) shows results for the DBFs with ChIP data. (d) shows results for the DBFs without ChIP data, excluding the nucleosome. COMPETE results are invisible in **??** because COMPETE did not predict a single site for those DBFs with a probability of greater than 0.25.

CHAPTER 3

A Bayesian Treed Model

3.1 Introduction

The primary motivation for predicting or experimentally measuring the DBF binding landscape is to leverage this information to increase our understanding of genetic regulation and the internal processes in the cell. The key question is what effect DBF binding has on the expression of the nearby gene(s). We are interested in the combinatorial regulatory effects of DBF binding across all genes, in different cell types and conditions. Armed with a DBF binding landscape and gene expression data for DBFs and their targets, one could naively estimate the effect of DBF binding as a global effect of some score quantifying DBF binding on a per-gene basis. However, it is well known that the same DBF can have different effects on different genes in different environments or cell types, perhaps depending on distance of the binding site to the TSS, the combinatorial pattern of other DBF binding sites nearby, the expression level of the DBF in question, or other factors. To capture these heterogeneous effects we partition genes via recursive partitioning in a Bayesian treed regression framework, estimating a unique regression to estimate the effects of binding of each DBF in each subset of genes and conditions. In estimating the effects of DBF binding we create a single gene-DBF score from the set of DBF binding sites relatively near the gene and the DBF expression based on the generalized logistic function. This is motivated by the fact that below or above some thresholds, even weaker or even stronger binding near some gene will not further decrease or increase the regulatory effect. In partitioning the genes and conditions we consider all the available data as splitting variables, partitioning over gene-DBF binding scores and DBF expression.

When Bayesian treed models were developed in [CGM00, CGM01], the parameter and tree spaces were explored exclusively through MCMC sampling. The data sets explored in those papers were relatively small, with hundreds of observations and less than 20 variables. Even with data sets of such size, the sampling over the tree space tended to get trapped in a single mode fairly quickly, and multiple sampling restarts were needed to explore significantly different trees. When exploring much bigger genomic data sets with many thousands of genes in multiple conditions or cell types and possibly hundreds of DBFs, a strict sampling approach becomes less feasible, with an unreasonable number of restarts required to adequately explore the space of possible high scoring trees.

A fully greedy optimization is also not satisfying. It is possible for there to be trees that describe optimal partitions where one of the splits would have been non-optimal during a greedy tree-growth routine. We thus propose a compromise approach. Given some arbitrary proportion ζ_d, dependent on the depth d of the node under consideration, at each possible split we fork off new trees that take alternate splits at the internal node being considered if those alternate splits result in total likelihoods a proportion ζ_d or greater of the highest likelihood split. Depending on the data, some user-chosen parameters can be set to allow a broader exploration of the space of possible trees than is possible with a strictly greedy approach without requiring an unreasonable computational investment.

We also consider the problem of variable selection. At the simplest level, when genes are partitioned, we expect that some of the DBFs to have no activity related to a given gene subset. It is also known that some DBFs have no measurable direct regulatory activity. In these cases it is informative to select variables on a by-partition-subset basis independently at each leaf node. Another case is that where an investigator is interested in regulatory analyses on a subset of genes or pathways, in which case we expect that only a subset of DBFs will be responsible for the regulation. Here we may make better predictions with an iterative process where we build a tree, select variables at each leaf node, take the union of selected variables over each leaf node and the variables used for splitting, and then rebuild the tree using only variables from that union.

3.2 Methods

3.2.1 Overview

We consider the problem of linear regression with partitions on the space of independent variables. We assume there is a separate linear regression relationship between the predictors and response in each part. Our method is designed to allow for examination of data sets with a large number of observations and a large library of candidate variables that may be used for partitioning or in the independent linear regressions in each part of the partition. We also allow for a separate library of candidate partioning variables. The partitioning and response prediction is carried out using a regression tree. In some problems, some of the variables in the candidate library may not be relevant either to the partitioning or the regressions. To account for this we propose a tree growth, variable selection cycle. See Section 3.2.3.4.

Our regression tree framework is based on Bayesian regression, similar to the framework developed in [CGM00, CGM01] but with Zellner's g-prior [Zel83, GZ86] (see Section 3.2.3.2) instead of the Normal-Inverse-Gamma prior as used in Chipman's work. We optionally allow the use of separate sets of variables for partitioning the observations and for predicting the responses in each part or subset. In our problem of interest, we take a DBF binding landscape, either one predicted by SparScape or a similar method or one built through collecting ChIP data for a group of DBFs, and gene expression data for both the DBFs of interest and the target genes of interest. This is usually the set of all genes in the organism under investigation or a group of pathways of interest.

With this information we partition the set of genes in the standard way popularized by Breiman *et al.* with the CART algorithm [BFS84]. At each node we consider splits at some number of quantiles of each variable, where we rank the observations, split the observations at that node according to whether the feature value is above or below the specified quantile, and estimate two separate Bayesian OLS regressions with Zellner's g-prior, one on each subset of observations. Then the sum of the marginal log posterior densities for the two split parts is compared the sums from other possible splits. When the marginal posterior for a potential split, which includes a prior term that penalizes growth of the tree, no longer increases relative to the potential parent node for any considered split, growth from that node halts. If there are splits that increase the posterior, the node is split using the best scoring feature quantile as the splitting criterion.

We propose two novel developments to this framework. The first is that we explore the sample space of trees with a forking greedy approach instead of an MCMC sampling or greedy optimization approach. Even with a relatively small number of observations and independent variables as investigated in Chipman's work, the sampling tends to get stuck quite easily in a single local mode. Chipman advocates dealing with this by running the sampling numerous times with fresh restarts. This is unsatisfactory when the number of observations and predictor variables can both be large as in the problems we explore. One obvious alternative is to take the standard fully greedy approach, as is done in [BFS84, KL01, Loh02, HHZ12, ZH07, ZHH08]. With the complex combinatorial control in genetic regulation, it is plausible that the optimal partition of genes may not be achievable if we take the greedy path and follow only the highest scoring splits at the highest levels of the tree. We propose to find a middle ground between the greedy approach and the sampling approach. We follow a greedy approach, but when different splits not on the same variable score within some threshold of each other, we fork off a copy of the current tree with an alternate split at the current node and continue growth independently for the original and forked trees. The threshold grows towards one as the tree grows in depth, with the initial threshold and the growth rate controlled by parameters to allow the user to control the tradeoff between speed and the breadth of the exploration of the tree space. The user may further set the maximum number of trees that may be forked off at a single potential node split.

Our second contribution is to introduce a cycle of tree growth, variable selection in the leaf nodes of the original and all forked trees, then a fresh restart of tree growth considering only the variables selected in at least one leaf node in the previous cycle. If the partitioning variables are the same as the predictor variables in the regressions, all variables used to split the data in the previous cycle are also considered in the following cycle. This cycle continues until no new variables are selected out at the end of a cycle. In the problems we explore, this process is advantageous when one is examining only a subset of genes. One still wants to consider the entire library of DBFs with available binding data, but some DBFs may have no regulatory role in the gene subset under investigation. Which DBFs are not active in regulating the gene subset of interest is of itself useful information. Also, including them in the partitioning and leaf node regression estimation introduces an unnecessary source of noise into the analysis.

3.2.2 Input Data and Variable Transformation

3.2.2.1 Gene-DBF Binding Association

We consider the problem of predicting target gene expression from DBF binding and DBF gene expression. We consider DBF binding for the 12 TFs with genome-wide ChIP-Seq binding data from [CXY08]. Our first problem is to define a single association score to integrate the ChIP peak intensity (or posterior binding probability were we considering binding landscape predictions as input) and proximity to genes. For this we follow [OZW09]. We assume that the association strength of DBF j on gene i

is a weighted sum of intensities of all of the peaks of TF j:

$$a_{ij} = \sum_k c_k e^{-d_k/d_0}, \tag{3.1}$$

where c_k is the intensity (number of reads aligned to the coordinate) of the k^{th} binding peak of DBF j, d_k is the distance (number of nucleotides) in the reference genome between the TSS of gene i and the k^{th} binding peak, and d_0 is a constant. In theory, the summation is over all binding peaks of a given DBF. But the effect of a peak decays exponentially when d_k increases where the speed depends on d_0. When d_k/d_0 is very large the contribution of the peak will be effectively zero. We set $d_0 = 500$ bps for E2f1 and $5,000$ bps for other DBFs because E2f1 tends to be closer to TSSs. To save computation time, we only consider peaks within a sufficiently large distance (say, 1 Mbps) of a gene.

3.2.2.2 Predictor Variable Transformation

In many biological systems there are regions in the domain of an input where change in the level of the input will produce a change in the response. But there are points below which and above which further reduction or increase in the input variable produces no further significant change in the response. We believe this likely to be the case for the relationship between DBF binding, DBF expression, and target gene expression. This suggests a sigmoid transformation of the input variables will be useful. We also believe that target gene expression is a function of the interaction between gene-DBF association and DBF expression that is unlikely to be well modeled by a standard interaction term in a linear regression. Considering this, we choose to transform the gene-DBF association score (3.1) and the DBF expression into a single gene-by-DBF predictor variable using the generalized logistic function (GLF). This takes the form

$$s_{ijk} = \left[1 + Q\left(e^{\beta_1(a_{ij}-m_1)+\beta_2(g_{jk}-m_2)}\right)\right]^{-1/\nu}, \tag{3.2}$$

where a_{ij} is the association strength of DBF j with gene i and g_{jk} is the gene expression measurement for DBF j in condition k. The parameters Q, β_1, β_2, m_1, $m_2 \in \mathbb{R}$ and $\nu > 0$ are user defined and control the shape of the GLF curve. The β_i values control the growth rate of the logistic curve between the lower asymptote and the upper asymptote. Q and ν have control over the the range and shape of the curve. Regardless of these values, the asymptotes are both in the range $[0,1]$. For $Q = \nu$, the m_i values control the time of maximum growth. For our choice of these parameters (see below) we have $s_{ijk} \in (0,1)$.

Because the distribution of a_{ij} is not consistent across DBFs, if we use an untransformed value of a_{ij} as input to this GLF transformation, some DBFs will have $s_{ijk} \approx 0$ for almost all i, j, and k, while other DBFs will have almost exclusively values of $s_{ijk} \approx 1$. This will eliminate our predictive and discriminative power. We thus choose to standardize the binding association scores for each DBF independently through quantile normalization to the standard normal distribution. With these normalized binding scores, we chose GLF parameters values $Q = \nu = 0.01$, $\beta_1 = -3$, $\beta_2 = -0.85$, $m_1 = 0$, and $m_{2j} = \bar{g_j}$, where $\bar{g_j}$ is the mean gene expression value for DBF j over all the conditions and m_{2j} is the value of m_2 for the transformation of data for DBF j.

For similar reasons, we also standardized the gene expression scores for both the DBFs and the target genes. We chose a single condition, $k^{\star}$, as our reference condition, and set $g'_{jk} = \log(g_{jk}/g_{jk^{\star}})$. Figure 3.1 shows histograms of the normalized binding association scores for each DBF. Figure 3.2 shows plots of the normalized binding association score versus the GLF transformed value for each DBF and condition. Figure 3.3 shows barplots of the normalized DBF expression values.

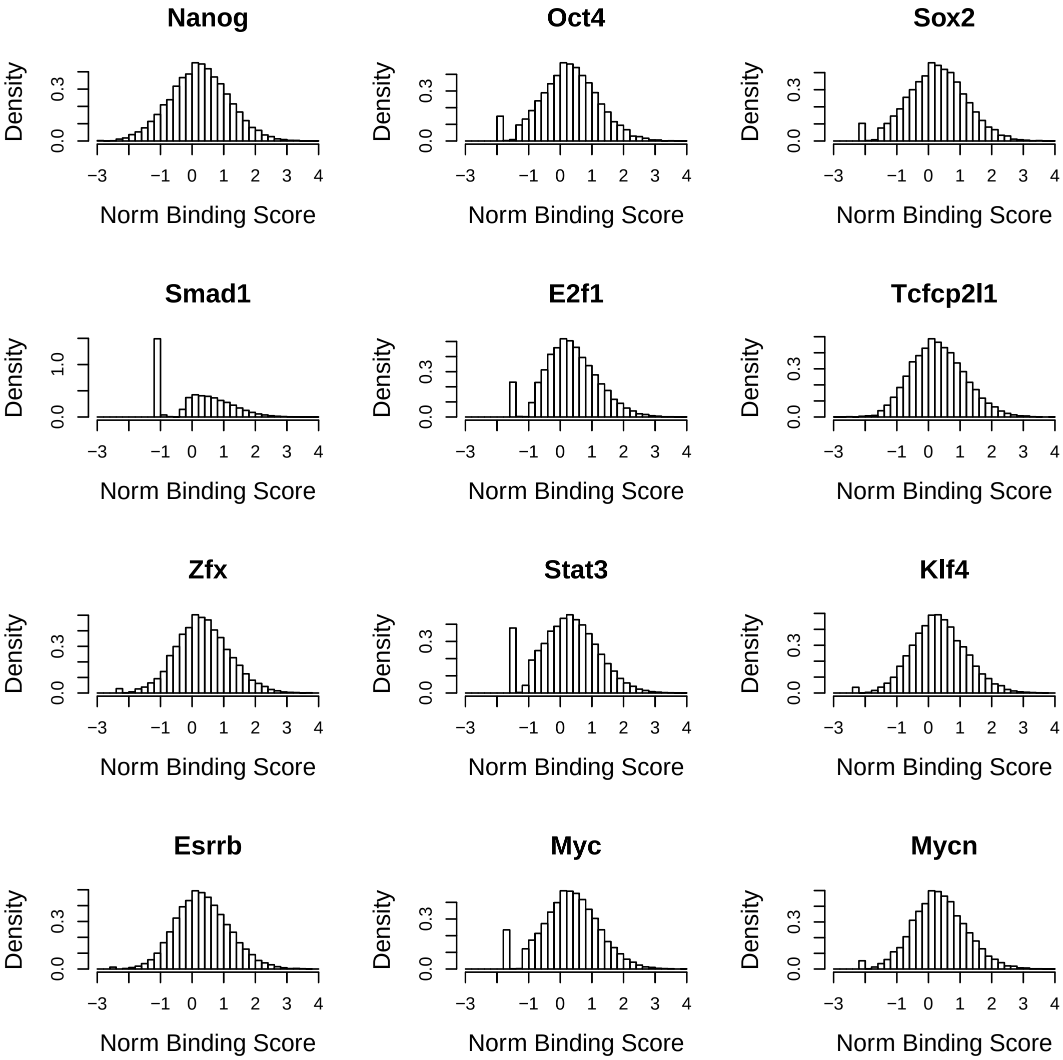


Figure 3.1: Histograms of the quantile normalized binding association scores for the 12 DBFs. Many DBFs had a point mass at zero which results in a point mass at the minimum of the normalized score and a skewed distribution despite the normalization to the standard normal distribution.

3.2.2.3 Partitioning Variables

We combine the gene-DBF binding association and DBF gene expression into a single gene-DBF score. But the target genes could plausibly be partitioned into parts with different regulatory relationships with DBFs from the candidate library based directly on gene-DBF binding associations or DBF gene expression. We thus use these directly as partitioning variables, rather than the transformed predictor variables.

3.2.3 Partitioning, Forking, and Variable Selection

3.2.3.1 Tree Structure Prior

With our inputs normalized and transformed we turn our attention to partitioning, prediction, and variable selection. Our first task is to define a prior on the structure of our partitioning tree T. We follow Chipman *et al.* [CGM01] and build the prior on the structure of the tree recursively. The prior probability of a split at a node η of depth δ is

$$t\left(\eta\right) = P\left(\text{split}\,\eta|\delta\right) = \alpha_t\ (1+\delta)^{-\beta_t}, \tag{3.3}$$

where the root node has depth 0, $t\left(\eta_{(\text{root})}\right) = \alpha_t$, and α_t and β_t are fixed parameters. For details on choosing these parameters, see [CGM01]. One issue not discussed in that work is the fact that as the number of observations and variables grows, the relative contribution of this tree prior to the total posterior shrinks. It may be necessary to increase the value of β_t in this case to prevent an overly large tree from being grown. In the work presented here we set $\alpha_t = 0.25$ and $\beta_t \in (100, 150)$.

We allow an additional fine tuning of the tree structure prior. By default the prior probability of a split is zero if the split would result in a child node with fewer observations than predictor variables. Let n_t be the number of observations that will be directed to the smaller of the two potential child nodes after a split and let J be the number of predictor variables and $N \cdot K$ be the total number of observations at the root node (see next section). One may further restrict child node size by setting a prior probability of zero on potential splits which would result in either $n_t < \gamma_1 \cdot N \cdot K$ or $n_t < \gamma_2 J$. This results in setting a minimum number of observations in a leaf node in proportion to the total number of observations at the root node or the number of predictor variables under consideration. In this work we choose to restrict using the first criterion and do not allow splits with a child node with fewer than 4% of the total number of observations $N \cdot K$.

3.2.3.2 Node Splitting and Tree Growth

Let N be the total number of target genes, J be the number of DBFs, and K be the number of cell conditions under investigation. Let X be the $N \cdot K \times J$ matrix of GLF transformed gene-DBF scores. Notice that we consider each gene as K separate observations, one for each observed cell condition or state. Let Y be a vector of length $N \cdot K$ containing the gene expression measures of all N target (non-DBF) genes over K conditions. Finally let $X_{(t)}$ and $Y_{(t)}$ be the $n_t \times J$ sub-matrix of X and the sub-vector of length n_t of Y, respectively, corresponding to the n_t observations in the part of the partition associated with tree node η_t.

In the tree growth (partitioning) phase, we begin with all the observations assigned to a single root node $\eta_{(\text{root})}$ of depth $\delta_{(\text{root})} = 0$. Let Leaves$(T)$ be the set of leaf nodes – nodes with no children – of tree T. At the beginning when T consists only of the root node, Leaves$(T) = \left\{\eta_{(\text{root})}\right\}$. We grow the tree with a depth-first approach, first considering splitting the root node, the considering splitting the left child node of the split chosen for the root node (if we split the root node), and so on.

To split the nodes, we employ Bayesian OLS regression between the transformed gene-DBF score and target gene expression with Zellner's single parameter g-prior. This is a particular form of the conjugate Normal-Inverse-Gamma prior family. We model target gene expression as $Y_{(t)} = X_{(t)}\beta_{(t)} + \varepsilon$, where $\beta_{(t)}$ is the set of regression coefficients at node η_t, $\varepsilon \sim N(0, \sigma^2_{(t)} I)$, and I is the identity matrix. Zellner's g-prior takes the form

$$\begin{aligned}\pi\left(\sigma^2_{(t)}\right) &\propto \frac{1}{\sigma^2_{(t)}}\\ \pi\left(\beta_{(t)}\middle|\sigma^2_{(t)}\right) &\sim N\left(0, g\sigma^2_{(t)}\left(X^T_{(t)}X_{(t)}\right)^{-1}\right).\end{aligned} \tag{3.4}$$

So the prior covariance of our coefficient estimates is simply a scalar multiple g of the Fisher information matrix, the variance of the estimates $\hat{\beta}$, from standard least squares regression. This form allows for a simple and computationally efficient marginal posterior in closed form,

$$\begin{aligned} P\left(Y_{(t)}\middle|\,g\right) &= \int P\left(Y_{(t)}|\beta_{(t)},\sigma^2_{(t)}\right)\,\pi\left(\beta_{(t)}|\sigma^2_{(t)}\right)\,\pi\left(\sigma^2_{(t)}\right)d\beta_{(t)}d\sigma^2_{(t)} \\ &= \frac{\Gamma\left(\frac{n_t-1}{2}\right)}{\pi^{(n_t-1)/2}\sqrt{n}}\|Y_{(t)}-\bar{Y}_{(t)}\|^{-(n_t-1)}\frac{(1+g)^{(n_t-1-J)/2}}{\left[1+g\left(1-R^2\right)\right]^{(n_t-1)/2}}, \end{aligned} \tag{3.5}$$

where J, the number of DBFs, is the number of independent variables and R^2 is the typical coefficient of determination at the MLE coefficient estimate $\hat{\beta}_{(t)}$. There are a number of theoretical issues in the choice of g and the use of this prior formulation, but in our use these are not of practical importantance and we choose $g = \max(N \cdot K, J^2)$.

Now consider testing for splitting the observations at a leaf node η_t. Let $O_{(t)}$ be the vector of observation indices of the observations assigned to η_t. We test a set of possible splits where the observations in $O_{(t)}$ are split into the observations in the left leaf, $O^{(L)}_{(t)}$, and the right leaf $O^{(R)}_{(t)}$. We cycle through the J variables represented in the columns of $X_{(t)}$, ordering the observations by their value for the variable currently under consideration. For each variable j we consider a set of quantiles in $(q_{j1}, q_{j2}, \ldots, q_{jQ})$, where q_{ju} represents a split of the observations at the u^{th} considered quantile of variable $j \in \{1, 2, \ldots, J\}$. When considering splitting at quantile q_{ju}, observation $o_{(t),i}$ is split into the potential left leaf, $O^{(L)}_{(t)}$, if its value of the j^{th} variable is less than the splitting cutoff, $x_{(t),ij} < q_{ju}$, and into the potential right leaf, $O^{(R)}$, otherwise. Remember that we may have a separate matrix of variables only considered for splitting, M. In this case, the test is $m_{(t),ij} < q_{ju}$.

For each potential split we perform two regressions as described above, one on the observations in $O^{(L)}_{(t)}$ and one on the observations in $O^{(R)}_{(t)}$. For a potential split on quantile u of variable j we obtain the log marginal posteriors for the left child node, $\ell^{(L)}_{(t)} = \log\left(P\left(Y^{(L)}_{(t)}\middle|\,g\right)\right)$, and the right child node, $\ell^{(R)}_{(t)} = \log\left(P\left(Y^{(R)}_{(t)}\middle|\,g\right)\right)$, as given in Eq. (3.5). The log marginal posterior for the potential parent node is $\ell_{(t)} = \log\left(P\left(Y_{(t)}\middle|\, g\right)\right)$. If $\ell^{(L)}_{(t)} + \ell^{(R)}_{(t)} + \log\left(t\left(\eta_t\right)\right) > \ell_{(t)}$, this potential split on q_{ju} is inserted into the list A. A is ranked by $\ell^{(L)}_{(t)} + \ell^{(R)}_{(t)}$, as $\log\left(t\left(\eta_t\right)\right)$, the prior probability of a split of node η_t of depth δ_t is identical for all possible splits of node η_t. When we have tested all potential splits, the node is split according to the top ranked (first) element in A. If we are allowing forking of multiple splits (see Section 3.2.3.3), we may perform splits at more than one of the top ranked splits in A. If $\ell^{(L)}_{(t)} + \ell^{(R)}_{(t)} + \log\left(t\left(\eta_t\right)\right) < \ell_{(t)}$ for all possible splits, then further splitting from this node would decrease our marginal posterior and we halt splitting from this node. Then η_t is a leaf node in the final tree and the observations in $O_{(t)}$ form a part of our final partition.

3.2.3.3 Forking Multiple Trees

At each potential split, we find the set of ranked potential splits A as described above. In a purely greedy approach, we would simply choose to split the tree at the criteria described by A_1, the split that leads to the maximum marginal posterior. It is plausible, especially in the first few levels of the tree, that a slightly lower scoring split could lead to a better final tree after further tree growth. Due to this fact, we optionally allow multiple splits if two or more potential splits on different splitting variables (i.e. not splits on different quantiles of the same splitting variable) score similarly. In particular, let $\ell_i = \ell^{(L)}_i + \ell^{(R)}_i$ be the total log marginal posterior for the i^{th} potential split in A. We fork off an additional tree with the current node split by A_i $(i \neq 1)$ if

$$\ell_i > \log\left(1 - \alpha_s^{(d+1)^{\beta_s}}\right) + \ell_1. \tag{3.6}$$

This means that at the root node, we split at more than one splitting variable if the marginal posteriors of alternate potential splits are at least a proportion $(1-\alpha_s)$ of the marginal posterior of the top scoring split. This proportion grows to one at a speed depending on α_s and β_s. In this work we use rather mild values of $\alpha_s = 0.5$ and $\beta_s = 0.5$. The proportions given by these values for nodes of depth one through ten are shown in Figure 3.4. A more stringent set of forking proportions may be necessary in cases with a low signal to noise ratio, where a very large number of forked trees may cross a more lenient forking threshold leading to an explosion in computation time. With $\beta_s > 1$, alternate potential splits must have a nearly identical marginal posterior as the top ranking split once we descend more than a few levels into the tree.

3.2.3.4 Variable Selection Cycle

In many cases, including our genomics application, we have a large collection of candidate variables but expect some of them to have no relationship with the response of interest in any part of the partition. Considering these noise variables could potentially lower our power to find the optimal tree or trees. We combat this problem by adopting an iterative approach to tree growth. We begin with all observations assigned to the root node, considering the entire candidate library of predictor variables. When the tree growth is complete for the original tree and all forked trees, we perform an independent lasso regression at each leaf node to select the variables with a relationship with the response in each part of the partition represented by that tree [Tib96]. The lasso regression estimates are defined by

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \frac{1}{2} \| Y - X\beta \|_2^2 + \lambda \| \beta \|_1 \right\}, \tag{3.7}$$

where $\|\cdot\|_2^2$ denotes the squared L_2 norm and $\|\cdot\|_1$ denotes the L_1 norm. The key to selecting variables with the lasso is the choice of the tuning parameter λ. We choose λ by calculating the full λ solution path as described in [EHJ04] and then choosing λ using 10-fold cross-validation. Let $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{|\Lambda|})$, $\lambda_1 > \lambda_2 > \cdots > \lambda_{|\Lambda|}$, be the vector of λ values in the solution path. At the max value λ_1, $\hat{\beta}_j^{\text{lasso}} = 0$, $\forall\ j$. At each λ_i we calculate the mean prediction error over the 10 test sets, μ_i^{lasso} and the standard deviation of the 10 prediction errors, σ_i^{lasso}. We choose λ_i that minimizes μ_i^{lasso} with the restriction $\mu_i^{\text{lasso}} \geq \mu_1^{\text{lasso}} - \sigma_1^{\text{lasso}}$. This method compromises between selecting too many significant variables, as can be the case when choosing the λ_i that minimizes the test set prediction errors, and choosing no significant variables by selecting λ_1. We consider the variables with non-zero coefficient estimates at the selected λ_i as the selected variables at that leaf node.

We then find the union of the variables selected across all the leaf nodes in the original and all the forked trees. Let $S_{(t)} = \left\{ i \middle| \beta^{\text{lasso}}_{(t),i} \neq 0 \right\}$ be the set of indices of variables selected in leaf node η_t. Let $S_{(T_0)} = \underset{t \in \text{Leaves}(T_0)}{\cup} S_{(t)}$, the union of all the sets $S_{(t)}$, be the set of variable indices selected in any leaf node of the tree T_0. Now let $S_{(T_i)}$ be the set from forked tree $i = 1, \ldots, N_T$. Let $S^{\text{start}}_{(T_0)}$ be the set of variable indices considered in building trees T_i. In other words, what variables were considered at the root node of the original tree. If $\cup_{i=0}^{N_T} S_{(T_i)} \neq S^{\text{start}}_{(T_0)}$, we built tree T while considering variables that do not have a significant relationship with the response in any part of the partition. We seek to grow a tree considering only the significant variables, so we clear all the trees, reassign all of the observations to a new root node, set $S^{\text{start}}_{(T_0)} = \cup_{i=0}^{N_T} S_{(T_i)}$, and restart the tree growth process considering only this smaller set of significant variables. Note that if we are not using a separate set of variables to split the internal nodes, then we are using the same variables for prediction and splitting, and we add to $S_{(t)}$ all of the variables used to determine the splitting criteria on the path from leaf η_t back to the root node.

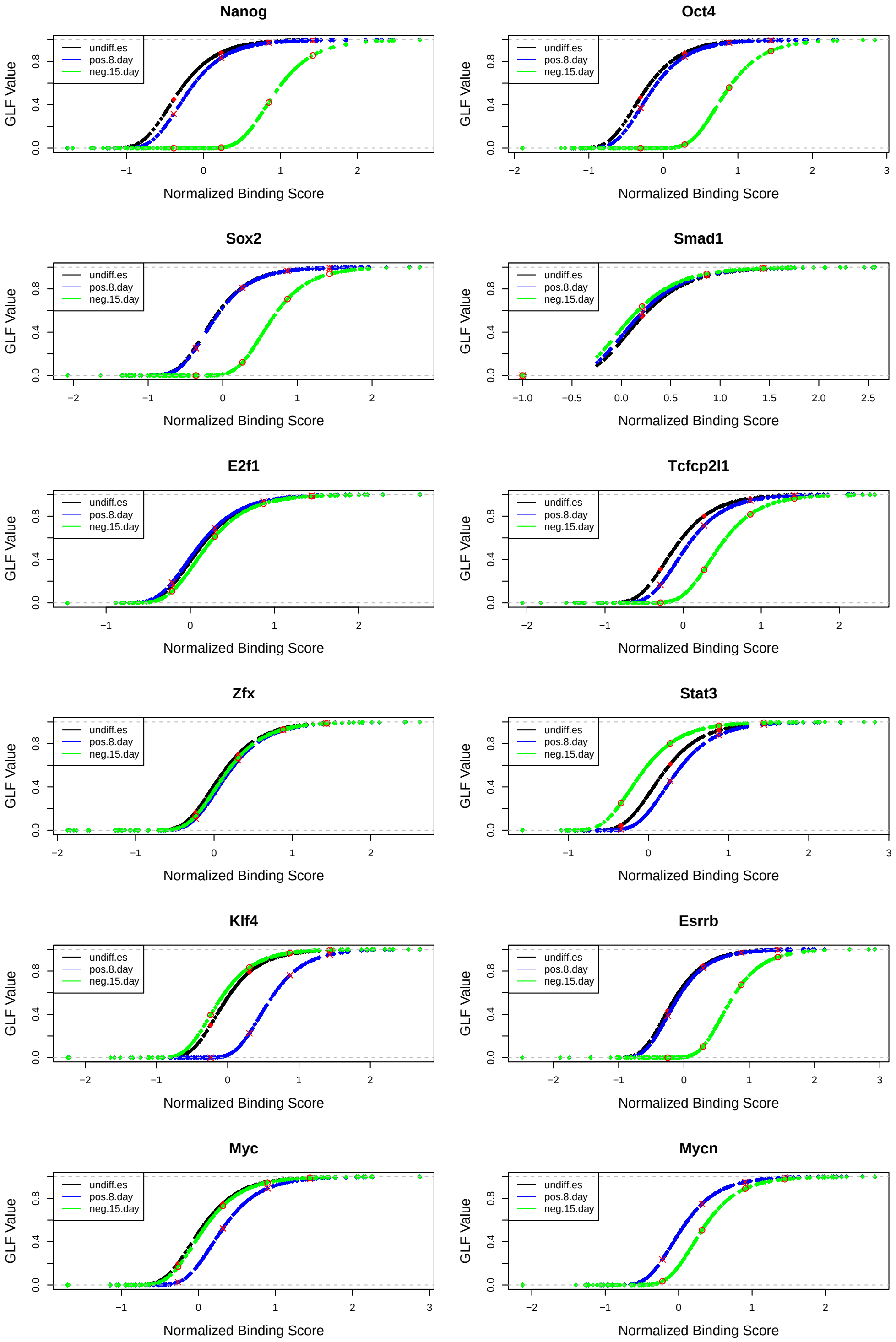


Figure 3.2: Plots of the normalized binding association scores versus the GLF-transformed scores used as the predictors in our method for each of the 12 DBFs. Each plot shows values for that DBF from all three considered conditions.

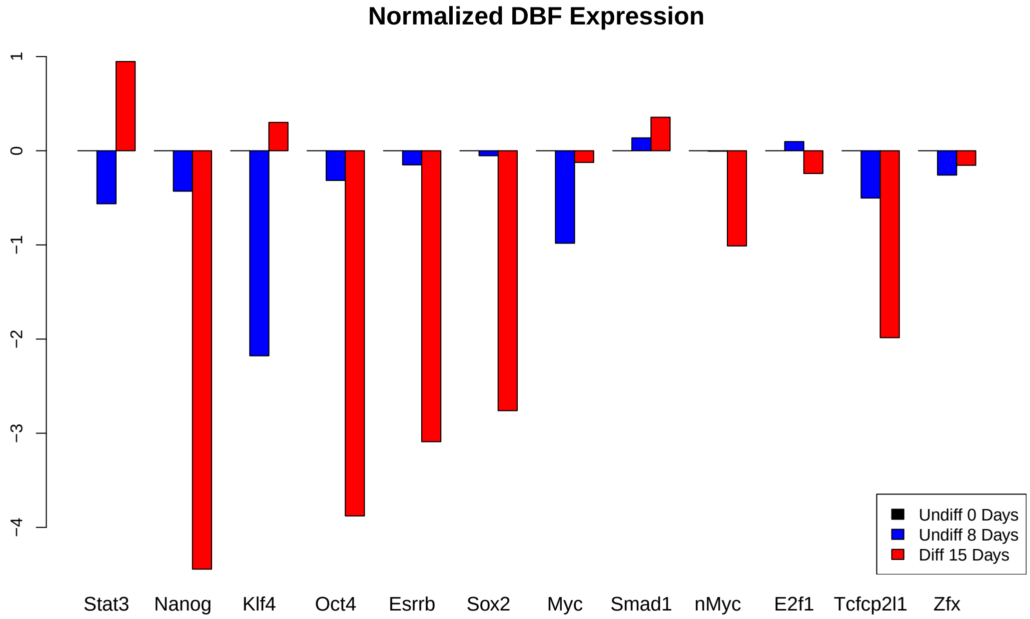


Figure 3.3: Plots of the normalized DBF expression scores over the three cell states considered: undifferentiated cells at 0 days, undifferentiated cells at 8 days, and differentiated cells at 15 days. Each DBF's expression measurements were normalized by dividing by that DBF's expression level in the undifferentiated at 0 days state and then taking the log.

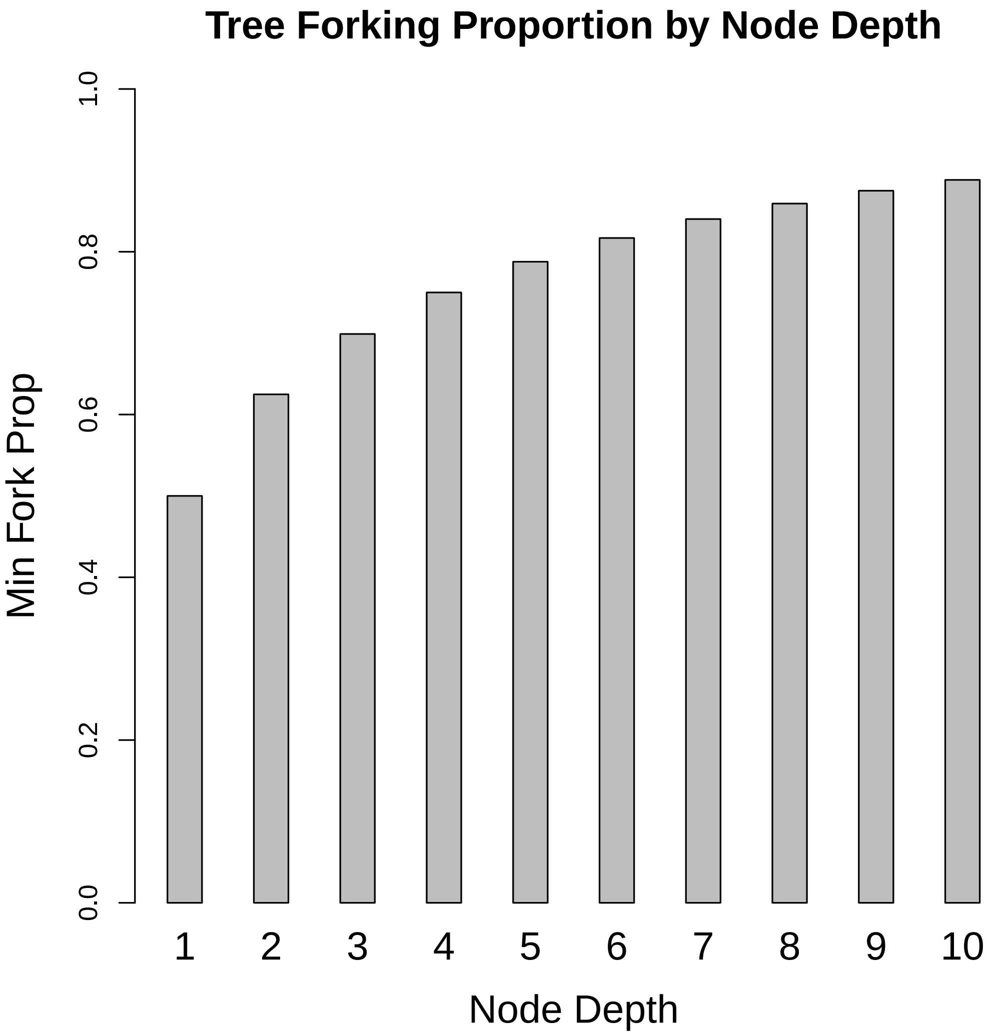


Figure 3.4: The proportion of the marginal posterior of the optimal split that must be cleared by the marginal posterior of alternate splits to induce a forking off of a new tree with this alternate split at this node.

3.3 Results

3.3.1 Simulated Data

We first explore some simulated data. We utilized the GLF transformed scores from our mouse data set. We considered only the $10,861$ genes with at least three non-marginal GLF transformed DBF scores. We created a simple partitioning tree with 13 total nodes and 7 leaf nodes shown in Figure 3.5. We classify the genes into class 1 through class 7, corresponding to which leaf node they are directed to. In six of the leaves we simulated a response with a regression including between four and eight of the DBFs with a signal variance to error variance ratio ranging from 4 to 16. The response for the observations in the seventh leaf were iid normal random variables with no relationship to the predictors. Two of the twelve DBFs had no true relationship with the response, either in the regressions or as splitting variables.

When the candidate library consisted only of the 12 DBFs with mouse data, we predicted a tree shown in Figure 3.6 with 19 total nodes and 10 leaf nodes. The true partition tree, shown in Figure 3.5, has 12 nodes with 7 leaves. Our predicted tree has only three extra splits compared to the true tree. If we consider the class label of a leaf node to be the majority class of the member genes, then only 0.5% of the genes are mislabeled, meaning we achieve almost perfect separation. We ran two tree growth cycles and selected out the two DBFs from the twelve total DBFs that did not have a true effect as splitting or predictor variables. Remember that we select significant variables through a lasso regression in each leaf node after the tree is grown.

If we take a weighted average of the percentage of variables correctly selected (in or out) at each leaf node, we correctly select 85% of the variables. The extra nodes stem from correction for minor inaccuracies in the cut points chosen at the high level splits. In the higher levels the splits chosen were for the correct variables and quite close to the true split point, but not exactly. This required a few refining splits further down in the tree. Table 3.1 gives the breakdown of how many members of

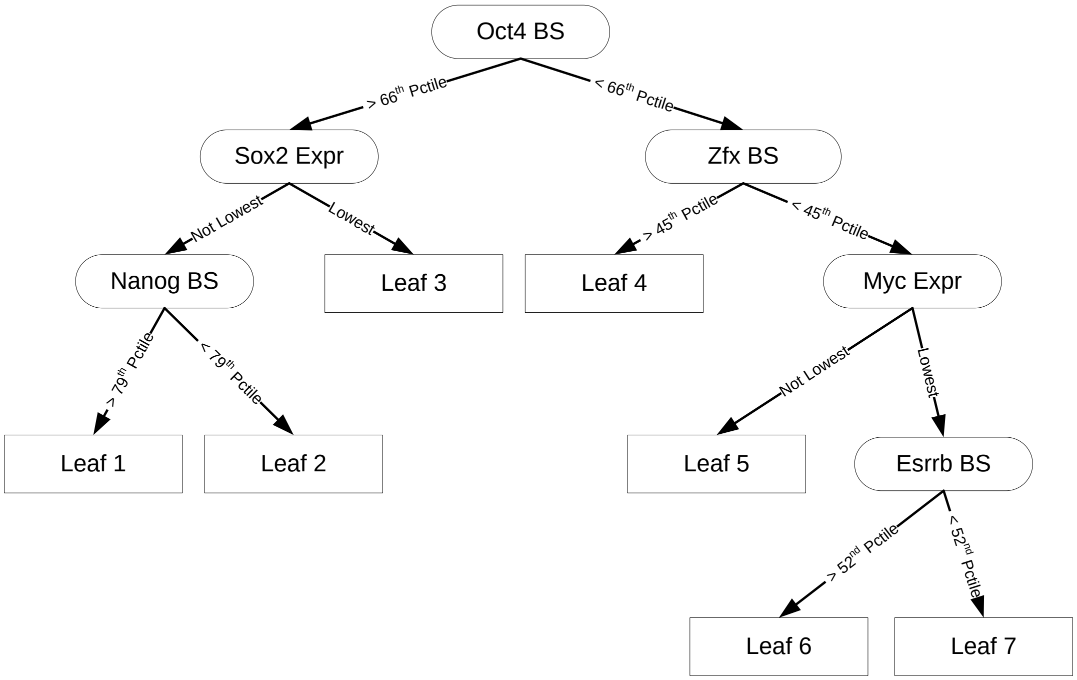


Figure 3.5: The true partitioning tree from our simulation. The text in the internal nodes corresponds to the true splitting variable at that node. BS stands for binding score. Expr stand for gene expression value over the three conditions. The text on the arrows from the internal nodes gives the splitting condition.

each part of the true partition appear in the predicted leaf nodes.

To test our variable selection we added 68 false variables to this same data set. These false variables were created by choosing a random one of the 12 DBFs with a true relationship with the response and then sampling with replacement from that DBFs' distribution of GLF transformed scores. Through four tree growth and variable selection cycles we correctly selected all of the true DBFs and kept only 3 of the 70 false DBFs. All three of these false variables had a non-zero coefficient estimate in only a single leaf. Overall, we mislabeled only 2.2% of genes and correctly selected 87% of the true variables. Table 3.2 gives the breakdown of how many members of each part of the true partition appear in the predicted leaf nodes.

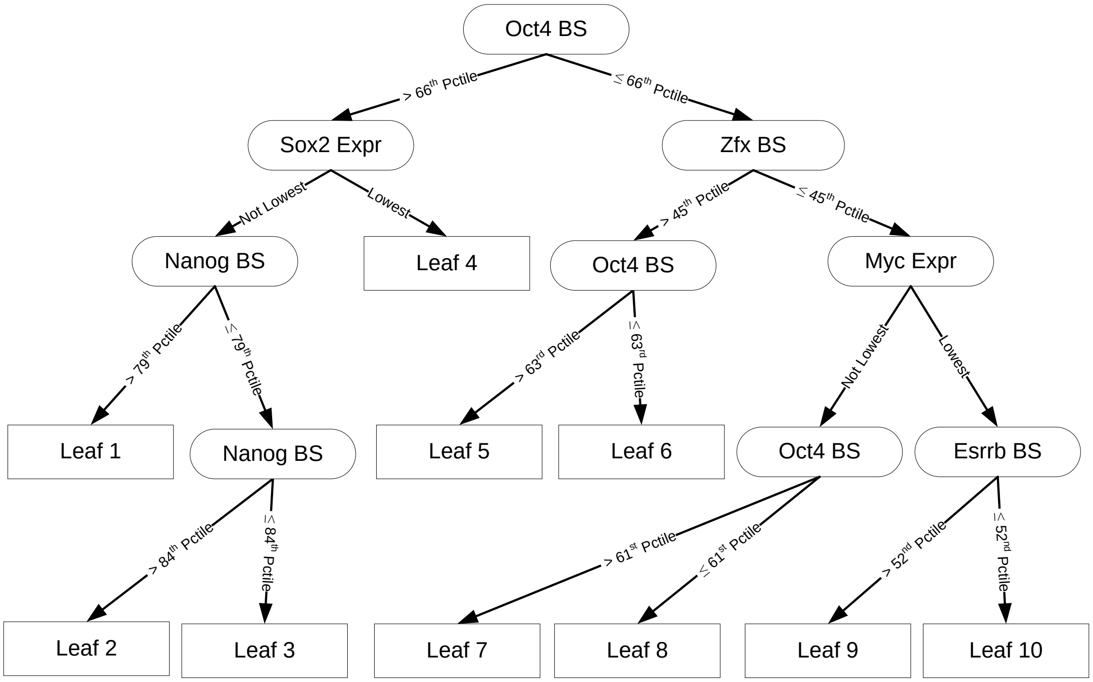


Figure 3.6: The predicted partitioning tree for the prediction with no extraneous variables. The text in the internal nodes corresponds to the splitting variable predicted at that node. BS stands for binding score. Expr stand for gene expression value over the three conditions. The text on the arrows from the internal nodes gives the splitting condition.

3.3.2 Mouse Data

The mouse data analyzed is the same as in Chapter 1. We calculated gene-DBF binding association scores for 14,812 mouse genes, then normalized and transformed them as discussed above. A heat map of the correlation between these 12 GLF transformed scores is shown in Figure 3.9. The correlation structure agrees with the groupings observed originally by Chen *et al.*, but is relatively low across the board. We predicted a tree with 7 leaves with diverse sets of significant DBFs and signs of the coefficients for the DBFs significant in that leaf. Remember that the GLF transformed combination of binding association score and DBF expression rises with more significant binding and higher DBF expression, and the log transformation of the gene expression values is monotonic, so a positive coefficient indicates that DBF is an ac-

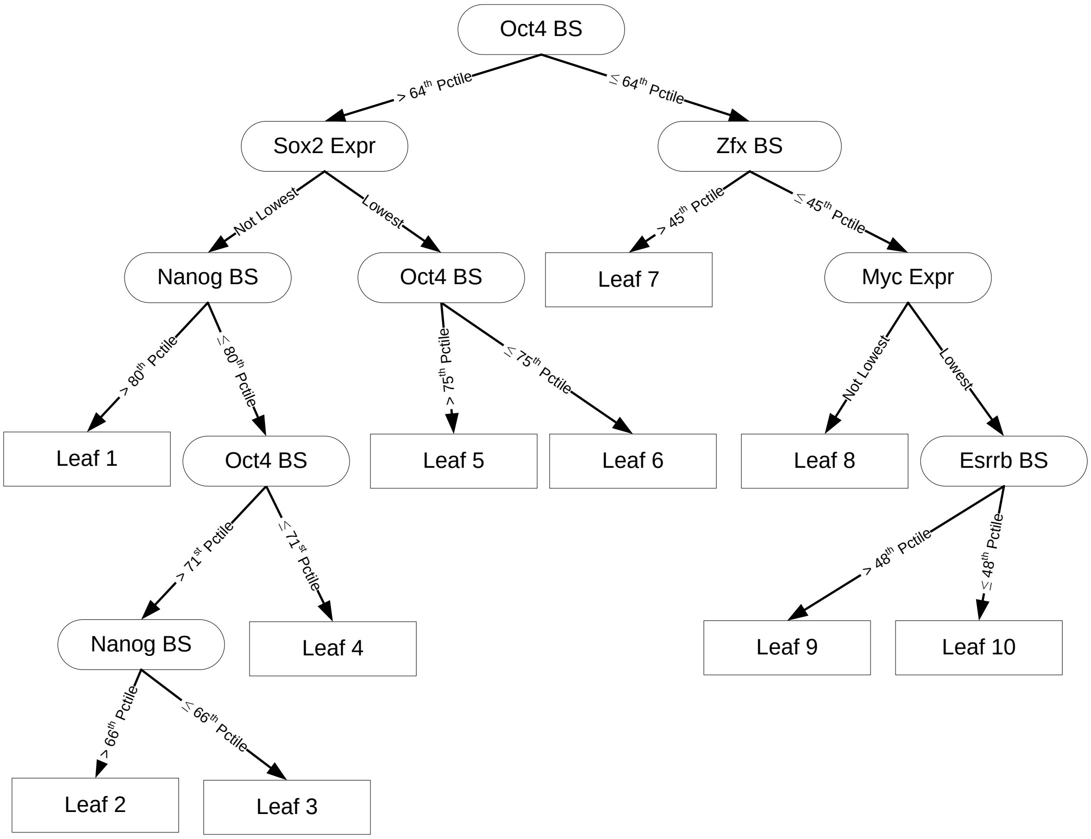


Figure 3.7: The predicted partitioning tree for the prediction with 68 extraneous variables. The text in the internal nodes corresponds to the splitting variable predicted at that node. BS stands for binding score. Expr stand for gene expression value over the three conditions. The text on the arrows from the internal nodes gives the splitting condition.

tivator in the set of genes represented by that leaf node and a negative coefficient indicates that DBF is a repressor in the set of genes in that leaf node. Figure 3.8 shows the predicted partitioning tree. Table **??** summarizes which DBFs were selected in each leaf and the sign of the estimated regression coefficient for the selected DBFs in each leaf node. The leaf names match betwen the figure and table. Table 3.3 gives the percentage of times a split predicted in the full mouse data, shown in Figure 3.8, was predicted over 20 random samples of one half of the mouse genes. We see that some of the top level splits are highly consistent, while the others are less consistent.

Table 3.1: The number of members of each class or part of the true partition (C1, C2, ..., C7) present in each leaf node in the tree predicted in the simulation considering only the 12 true DBFs, 10 of which have a true relationship to the response in at least one part of the partition. Leaf node names correpsond to those shown in Figure 3.6.

	C1	C2	C3	C4	C5	C6	C7
Leaf 1	1755	0	0	0	0	0	0
Leaf 2	467	29	0	0	0	0	0
Leaf 3	0	5146	0	0	0	0	0
Leaf 4	0	1	3700	0	0	0	0
Leaf 5	6	46	25	406	0	0	0
Leaf 6	0	0	0	10700	3	1	0
Leaf 7	2	15	17	0	529	0	0
Leaf 8	0	0	0	0	6302	0	1
Leaf 9	2	5	0	0	0	1249	0
Leaf 10	0	10	0	0	0	8	2158

The partitioning described in Figure 3.8 makes some sense given what we know about the grouping of these 12 DBFs, with one group centered around Oct4 (Oct4, Nanog, Sox2, Stat3, Smad1) and the other around cMyc (cMyc, nMyc, E2f1, Zfx). The first partition is on the expression level of Oct4, where high expression is a known indicator of stem cell state. In our three cases, Oct4 had high and similar expression levels in two of them, and low expression in the other. In the Oct4 high expression states, the leaves where there were non-zero regression coefficients either had very low nMyc activity, very high Nanog activity, or high Sox2 activity. The only leaf on this side of the intial partition with no significant regression coefficients is the set of genes with extremely low Nanog and Sox2 activity.

In the cell state with low Oct4 expression, only one leaf node had a significant

Table 3.2: The number of members of each class or part of the true partition (C1, C2, ..., C7) present in each leaf node in the tree predicted in the simulation with 68 false variables in addition to the 12 true DBFs, 10 of which have a true relationship to the response in at least one part of the partition. Leaf node names correspond to those shown in Figure 3.7.

	C1	C2	C3	C4	C5	C6	C7
Leaf 1	2147	0	0	38	10	4	6
Leaf 2	79	1131	0	0	0	0	0
Leaf 3	0	3061	0	0	0	0	0
Leaf 4	6	1059	0	118	81	35	46
Leaf 5	0	0	1078	79	91	0	0
Leaf 6	0	1	2664	0	0	0	0
Leaf 7	0	0	0	10851	0	0	0
Leaf 8	0	0	0	13	6652	0	1
Leaf 9	0	0	0	2	0	1219	116
Leaf 10	0	0	0	5	0	0	1990

Table 3.3: Consistency of splits predicted in the full set of mouse genes over 20 random samples of half the mouse genes.

Split	Repeat %
Oct4 Expr	100
nMyc BS	85
cMyc BS	20
Nanog BS	40
Smad1 BS	10
Sox2 BS	20

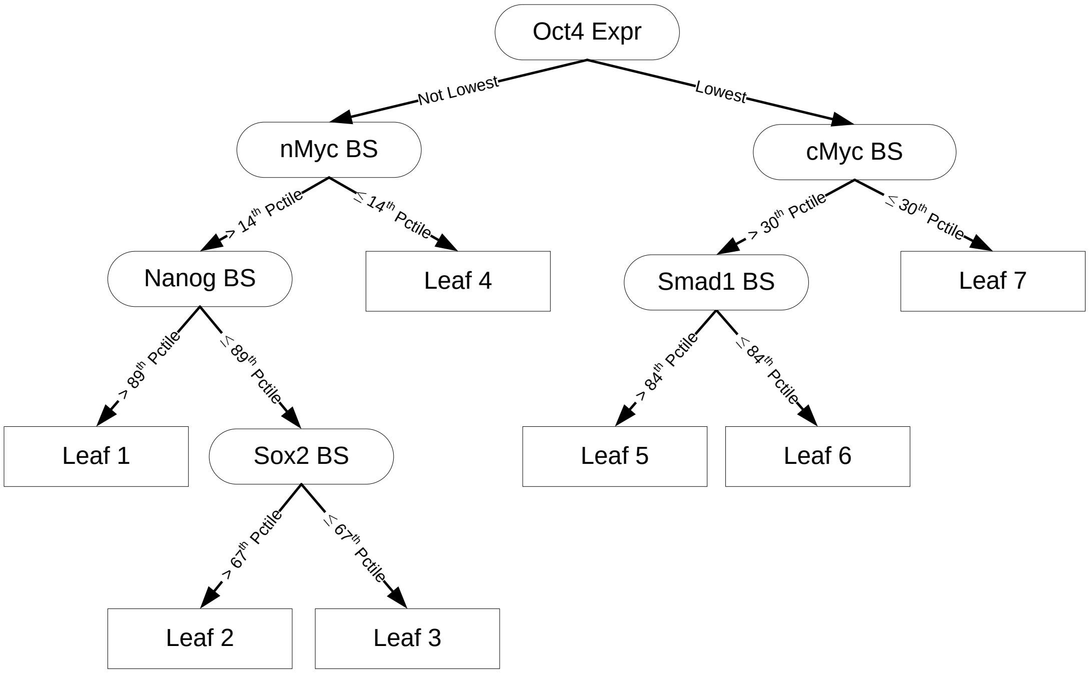


Figure 3.8: The predicted partitioning tree for the 12 TFs in mouse ESCs.

number of non-zero coefficients. This node contains genes with at least reasonable levels of cMyc activity and without very high Smad1 activity. Only cMyc group DBFs have non-zero coefficients in this group.

These partitions point to the activity of the Oct4 centered group of DBFs being more important than the cMyc group in determining the changes in strength and direction of regulatory effect due to combinatorial binding in stem cells. In the low Oct4 expression branch, where the cells have for the most part begun differentiating, cMyc and Oct4 group DBF binding both serve as markers of partition category, but only cMyc group DBFs have non-zero coefficients in these differentiated cells. Note that it is known that the Oct4 group DBFs still have regulatory activity in differentiated cells. This activity is simply not captured well by our large scale partitioning scheme.

To further examine the biological meaning of our partition, we performed simple gene ontology enrichment tests on the lists of genes in each partition using the

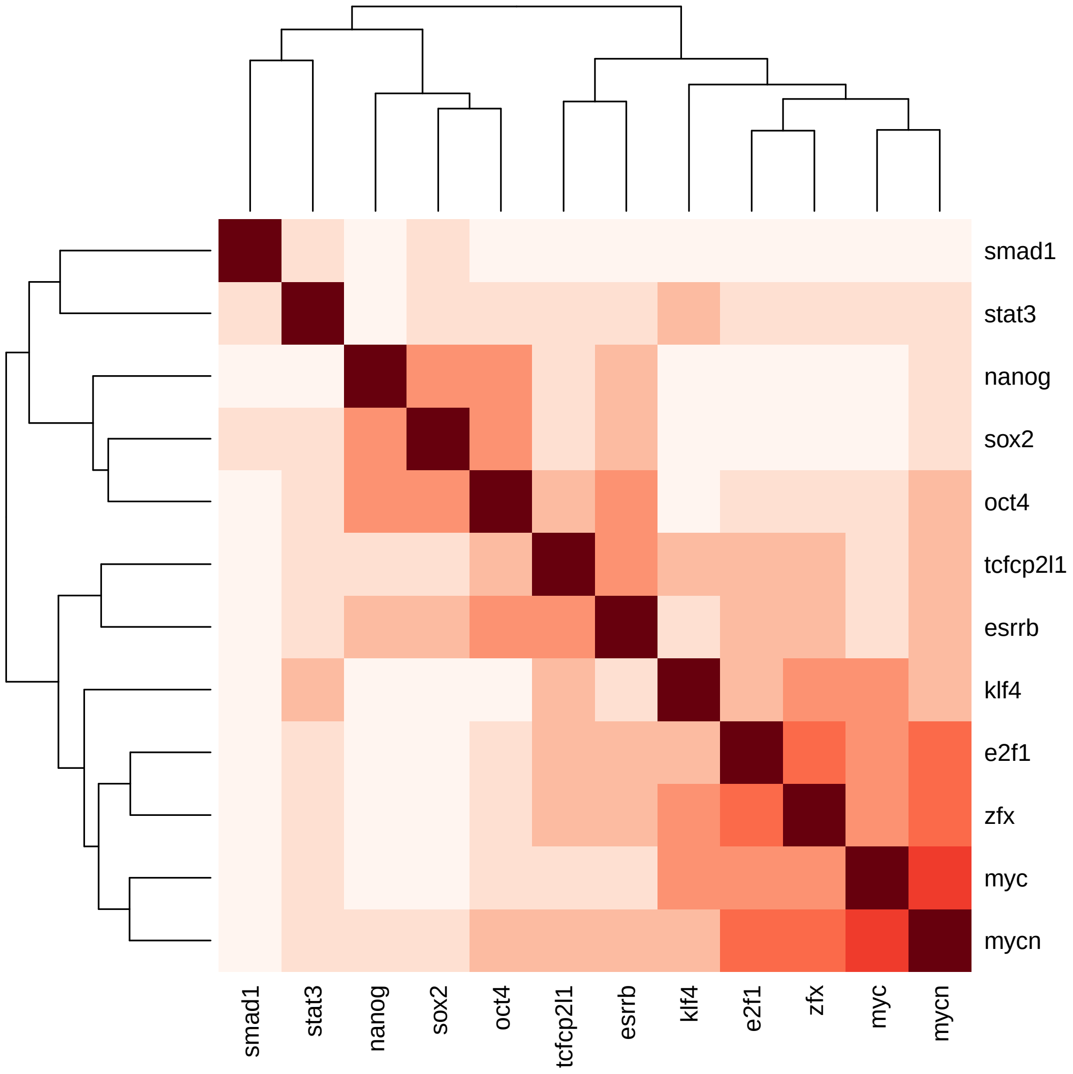


Figure 3.9: A heat map of the correlation between the GLF transformed scores of the 12 examined mouse TFs.

software described in [BWG04]. We tested for enrichment of gene function annotations, gene process annotations, and gene component annotations at each leaf, and among the leaves 1, 2, and 4, the group of leaves with non-zero regression coefficients in the high Oct4 expression branch of the tree. The results are give in Table 3.4, Table 3.5, and Table 3.6. These tables show the annotations enriched in each leaf (or group of leaves) but not enriched in any of the other leaves (or group of leaves).

Leaf 4, where Oct4 expression is high and nMyc activity is extremely low, seems to be particularly enriched for a variety of functions and processes that are not enriched in any of the other leaves. Also of note is the fact that relative to the other leaves, the leaves with non-zero coefficients in the high Oct4 expression branch, leaves 1,2, and 4, are enriched for sequence-specific DNA binding TF activity and nucleic acid binding TF activity, an expected enrichment among groups of genes targeted by TFs key in processes such as gene regulation and differentiation.

Table 3.4: Gene function annotations enriched in the genes at each leaf and not in the genes at the other leaves. Also listed are the additional annotations enriched only in the three leaves on the left side of the tree with non-zero regression coefficients.

Leaf 1	Enzyme binding
	Zinc ion binding
	Transition metal ion binding
	Protein domain specific binding
Leaf 2	
Leaf 3	Transferase activity
Leaf 4	serine-type endopeptidase inhibitor activity
	Gated channel activity
	Ion channel activity
	Passive transmembrane transporter activity
	Ion gated channel activity
	Peptidase inhibitor activity
	Substrate-specific channel activity
	Channel activity
	Endopeptidase inhibitor activity
	Peptidase regulator activity
	Endopeptidase regulator activity
	GABA-A receptor activity

Leaf 5	
Leaf 6	
Leaf 7	Protein homodimerization activity
	Transporter activity
	Substrate-specific transporter activity
Leaves 1,2,4 Add’l	Transferase activity
	Sequence-specific DNA binding TF activity
	Nucleic acid binding TF activity

Table 3.5: Gene process annotations enriched in the genes at each leaf and not in the genes at the other leaves. Also listed are the additional annotations enriched only in the three leaves on the left side of the tree with non-zero regression coefficients

Leaf 1	
Leaf 2	Regulation of metabolic process
Leaf 3	Establishment of localization
	Transport
	Cellular component organization
Leaf 4	Biological adhesion
	Cell adhesion
	Regulation of multicellular organismal process
	Locomotion
	Regulation of biological process
Leaf 5	
Leaf 6	Cellular protein metabolic process
Leaf 7	Positive regulation of cellular process
Leaves 1,2,4 Add'l	

Table 3.6: Gene component annotations enriched in the genes at each leaf and not in the genes at the other leaves. Also listed are the additional annotations enriched only in the three leaves on the left side of the tree with non-zero regression coefficients

Leaf 1	Cytosol Nuclear lumen
Leaf 2	
Leaf 3	Mitochondrion Organelle membrane
Leaf 4	Intrinsic to plasma membrane Integral to plasma membrane Cell periphery Plasma membrane Neuron part Ion channel complex Neuron projection
Leaf 5	
Leaf 6	
Leaf 7	Extracellular matrix
Leaves 1,2,4 Add'l	

3.4 Discussion

We present an extension of Bayesian treed regression models for applications with big data sets that are difficult to explore with a sampling approach. We specifically aim to address the problem of predicting gene expression from DBF binding data and DBF expression. Because we do not believe a standard linear regression well captures the relationship between target gene expression and the measures we have of DBF activity, we transform these DBF measures into a single logistic score, but then extend the tree splitting procedure to allow for separate matrices of predictor variables and node splitting variables. This allows us to use the GLF transformed DBF measures as predictors of target gene expression but to allow node splitting separately on both DBF binding association scores and DBF gene expression.

We have also seen in past implementations of Bayesian treed regression models that the sampling is easily trapped in a particular mode, necessitating numerous fresh restarts of the sampling algorithm to explore multiple modes and increase the chances of exploring the most promising modes. With large data this is impractical. The obvious alternative is to take the standard fully greedy approach. With complex biological systems, it is plausible that complex combinatorial relationships create situations where node splits high on the tree that are not the optimal scoring split could lead to trees with more accurate clustering and prediction further down the tree. To address this issue, we propose a compromise optimization technique. We proceed building the greedy tree, but fork off copies of the tree with alternate node splits when multiple splits on different splitting variables seem similarly promising.

We finally propose one additional extension motivated by our examination of complex problems with large data sets. One may consider a large candidate library of predictor variables, but not expect all of them to have a significant relationship with the response in any part of the partition. Or one may be interested simply in which predictors have a relationship with the response in each part. We propose an iterative tree-growth/variable-selection cycle to address this possiblity. We grow our partitioning tree, then select variables through a simple lasso regression in each leaf node. We take the union of the variables selected in each leaf, and if this union does not include all the variables originally considered, we erase the tree and restart the tree growth process considering only the selected variables. We continue this cycle until we do not select out any variables that were considered at the beginning of the current tree growth iteration.

With these extensions, we believe the Bayesian regression tree approach can be more fruitfully applied to complex problems with large data sets such as prediction of gene expression.

CHAPTER 4

Summary and Future Work

We have seen that likelihood-based modeling focused on sparsity and a tradeoff between broadly exploring the model space and computational efficiency to allow analysis of large genomic datasets with complex relationships can yield important insights into the functioning of the genome. We have taken this approach to predicting DBF binding landscapes over sets of genomic regions of interest and to utilizing these or other estimates of DBF binding to partition the genes in the genome to better understand the interplay of combinatorial binding, cell state, and gene regulation.

In predicting binding landscapes, it is prohibitively computationally expensive to predict binding landscapes considering the complete library of DBFs with known motifs but unknown concentrations over the entire genome in higher organisms. Other groups have dealt with this problem by not predicting a bp-specific binding landscape, considering only lower organisms with smaller genomes and smaller DBF libraries, predicting landscapes for only one or a few DBFs at a time with unknown concentrations, fixing concentrations *a priori*, or considering only a small subset of the DBF library and a small subset of the genome. We have relaxed all of these restrictions except for the need to consider only a subset of the genome. We consider the entire DBF library without the need for prior knowledge of DBF concentrations over a subset of regions in the genome. We also add the functionality of considering experimental binding data, such as ChIP-Seq, directly, allowing consideration of more information and improved prediction without requiring such information to be available for every DBF one is interested in considering.

For our motivating problem, considering a specialized set of genomic regions, we successfully select a sparse set of DBFs with enriched binding in those regions, including most of the DBFs expected to bind in such regions given experimental data, with a much reduced false positive and false negative rate both for DBFs with experimental binding data and for those without compared to competing methods. In a less specialized genmoic subset, a random sample of 9% of mouse genes, we select 22% of DBFs, about the proportion you might expect, including some DBFs known to bind in gene promoters close to the TSS. Even if considering entire chromosomes or genomes, our method can be used to predict a binding landscape on the entire genome using a complete library of DBFs with greatly reduced false positive predicted binding sites if one forgoes selection from the DBF library and uses our method to predict a binding landscape with a low but non-zero concentration penalty.

In the future there are a number of extensions and computational advantages that would be useful and relatively simple. SparScape was coded in C++ in an object oriented fashion using OpenMP to take advantage of all the CPUs on a single board. It could reasonable be expanded into an MPI framework to allow for much improved performance without relying on shell scripting and outside cluster job schedulers. We may also look into utilizing GPU programming to allow for much improved computational performance on everyday personal computers.

SparScape could also be extended to include further experimental data. We did not utilize experimental binding data for the nucleosome in the results presented here. This data can be included in the current implementation of SparScape. As such data becomes available in more organisms, utilizing this information could lead to better results, especially when considering regions that are not known *a priori* to be nucleosome depleted, such as gene promoters. We have also considered extending SparScape to include modification data known to influence DBF binding and gene regulation. These include histone modifications and DNA methylation. This would be a more involved extension, but would be quite feasible within the current framework.

In predicting the gene regulation action of DBF binding, we have extended the Bayesian treed regression model in a number of ways. Current treed regression models generally use the same set of variables for prediction and to determine partitioning criteria. In genomics applications, a regulatory effects, or other biological associations or effects, are not usually linear In addition, interaction terms can be essential, but the effects or associative relationship is not linear with a standard multiplicative interaction term. We extend Bayesian treed regression to allow for creation of custom scores or variable transformations as predictor variables but to retain the raw, untransformed inputs to create splitting criteria.

Few existing methods are designed for use with very large data sets as are commonly encountered in genomics applications. Because of this, variable selection in recursive partitioning algorithms has not been a priority. When the set of variables under consideration come from a large library, as is the case in our simulated application, it is feasible that only a small subset will have a true relation with the response in the problem of interest. We introduce a tree growth and variable selection cycle, where we grow our tree, then select variables in each leaf node independently using an L_1 penalized lasso regression. Then we collect the variables selected in any leaf node, including the variables used for splitting criteria at internal nodes if the separate splitting variables were not used. If this collection of selected variables does not include the entire library of variables considered at the start of this growth/selection iteration, we erase the tree and grow a new tree considering only the set of variables selected in the last iteration. We have found in simulated applications that this process is quite accurate and gives a good prediction of which variables are significant in any part of the partition and of which variables are significant in any particular part.

Finally, previous treed regression problems tend to use a greedy approach to growing the tree. Previous Bayesian treed regression proposals have relied on MCMC sampling. Even with small data sets one can easily be trapped in local modes, requiring a large number of fresh restarts to adequately explore the tree space. We propose a compromise approach to explore more possible trees than with a greedy approach but to avoid getting stuck in a non-optimal local mode. We follow the greedy path, but allow forking off copies of the current tree with alternate splits at the leaf currently under consideration when multiple splits seem promising.

In the future, exploration of the theoretical properties of our optimization and variable selection strategies could be illuminating. At the least, it would be useful to carry out a more extensive exploration of the empirical properties of these extensions, including some guidance for when these are most useful and perform best. Another promising avenue for further study is the examination of the relationship between the tree structure prior and the size of the data set, both the number of observations and the number of variables.

References

[AEP11] P Arnold, I Erb, M Pachkov, N Molina, and E van Nimwegen. “MotEvo: integrated Bayesian probabilistic methods for inferring regulatory sites and motifs on multiple alignments of DNA sequences.” *Bioinformatics*, **28**:487–494, 2011.

[BFS84] L Breiman, JO Friedman, and C Stone. *Classification and Regression Trees.* Chapman & Hall/CRC, 1984.

[BGL03] Ziv Bar-Joseph, Georg K Gerber, Tong Ihn Lee, Nicola J Rinaldi, Jane Y Yoo, FrançÌğos Robert, D Benjamin Gordon, Ernest Fraenke, Tommi S Jaakkola, Richard A Young, and David K Gifford. “Computational discovery of gene modules and regulatory networks.” *Nature Biotech*, **21**:1337–1342, 2003.

[BPQ06] M Berger, A Philippakis, QM Qureshi, FS He, PW Estep III, and M Bulyk. “Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities.” *Nat Biotechnol*, **24**:1429–1435, 2006.

[BWG04] E Boyle, S Weng, J Gollub, H Jin, D Botstein, J Cherry, and G Sherlock. “GO::TermFinder–open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes.” *Bioinformatics*, **18**:3710–3715, 2004.

[CGM00] Hugh Chipman, Edward I George, and Robert E McCulloch. “Hierarchical priors for Bayesian CART shrinkage.” *Stat Comput*, **10**:17–24, 2000.

[CGM01] Hugh Chipman, Edward I George, and Robert E McCulloch. “Bayesian treed models.” *Machine Learning*, **48**:299–320, 2001.

[CXR08] Li Chen, Jianhua Xuan, Rebecca B Riggins, Yue Wang, Eric P Hoffman, and Robert Clarke. “Identification of condition-specific regulatory modules by multi-level motif and mRNA expression analysis.” *Int J Comput Biol Drug Des*, **2**:1–20, 2008.

[CXY08] X Chen, H Xu, P Yuan, F Fang, M Huss, V Vega, E Wong, Y Orlov, W Zhang, J Jiang, Y-H Loh, HC Yeo, ZX Yeo, V Narang, KR Govindarajan, B Leong, A Shahab, Y Ruan, G Bourque, W-K Sung, ND Clarke, C-L Wei, and H-H Ng. “Integration of external signaling pathways with the core transcriptional network in embryonic stem cells.” *Cell*, **133**:1106–1117, 2008.

[DSS03] M Djordjevic, AM Sengupta, and BI Shraiman. “A biophysical approach to transcription factor binding site discovery.” *Genome Res*, **13**:2381–2390, 2003.

[EHJ04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. “Least angle regression.” *Ann Stat*, **32**:407–499, 2004.

[EPS10] J Ernst, H Plasterer, I Simon, and Z Bar-Joseph. “Integrating multiple evidence sources to predict transcription factor binding in the human genome.” *Genome Res*, **20**:526–536, 2010.

[FMB06] BC Foat, A Morozov, and HJ Bussemaker. “Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE.” *Bioinformatics*, **22**:e141–e149, 2006.

[FSA11] M Fidalgo, PC Shekar, Y-S Ang, Y Fujiwara, SH Orkin, and J Wang. “Zfp281 functions as a transcriptional repressor for pluripotency of mouse embryonic stem cells.” *Stem Cells*, **29**:1705–1716, 2011.

[FYK08] S Furuya, K Yoshida, Y Kawakami, JH Yang, T Sayano, N Azuma, H Tanaka, S Kuhara, and Y Hirabayashi. “Inactivation of the 3-phosphoglycerate dehydrogenase gene in mice: changes in gene expression and associated regulatory networks resulting from serine deficiency.” *Funct Integr Genomics*, **8**:235–249, 2008.

[FZ13] Fei Fu and Qing Zhou. “Learning sparse causal gaussian networks with experimental intervention: regularization and coordinate descent.” *J Am Stat Assoc*, **108**:288–300, 2013.

[GFB04] Feng Gao, Barrett C Foat, and Harmen J Busssemaker. “Defining transcriptional networks through integrative modeling of mRNA expression and transcription factor binding data.” *BMC Bioinformatics*, **5**:31, 2004.

[GL03] M Gupta and J Liu. “Discovery of conserved sequence patterns using a stochastics dictionary model.” *J Am Stat Assoc*, **98**:55–66, 2003.

[GN05] JA Granek and Clarke ND. “Explicit equilibrium modeling of transcription-factor binding and gene regulation.” *Genome Biol*, **6**:R87, 2005.

[GXL11] P Gu, X Xu, D Le Menuet, ACK Chung, and AJ Cooney. “Differential recruitment of methyl cpg-binding domain factors and dna methyltransferases by the orphan receptor germ cell nuclear factor initiates the repression and silencing of oct4.” *Stem Cells*, **29**:1041–1051, 2011.

[GZ86] Prem K Goel and Arnold Zellner, editors. *On assessing prior distributions and Bayesian regression analysis with g-prior distribution.* Elsevier Science Ltd, 1986.

[HCH09] X He, C-C Chen, F Hong, F Fang, S Sinha, H-H Ng, and S Zhong. “A biophysical model for analysis of transcription factor interaction and binding site arrangement from genome-wide binding data." *PLoS ONE*, **4**:e8155, 2009.

[HE10] B Herkert and M Eilers. "Transcriptional repression: the dark side of myc." *Genes Cancer*, **1**:580–586, 2010.

[HHZ12] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. "Unbiased recursive partitioning: a condition inference framework." *J Comput Graph Stat*, **15**:651–674, 2012.

[HKB02] Steven Hampson, Dennis Kibler, and Pierre Baldi. "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences." *Bioinformatics*, **18**:513–528, 2002.

[HS99] G Hertz and G Stormo. "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences." *Bioinformatics*, **15**:563–577, 1999.

[HSB10] X He, MAH Samee, C Blatti, and S Sinha. "Thermodynamics-based models of transcriptional regulation by enhancers: the roles of synergistic activation, cooperative binding and short-range repression." *PLoS Comput Biol*, **6**:e1000935, 2010.

[JM08] K Jessen and R Mirsky. "Negative regulation of myelination: relevance for development, injury, and demyelinating disease." *GLIA*, **56**:1552–1565, 2008.

[KCS08] J Kim, J Chu, X Shen, J Wang, and S Orklin. "An extended transcriptional network for pluripotency of embryonic stem cells." *Cell*, **132**:1049–1061, 2008.

[KL01] Hyunjoong Kim and Wei-Yin Loh. "Classification trees with unbiased multiway splits." *JASA*, **96**:589–604, 2001.

[KLS11] T Kaplan, X-Y Li, P Sabo, S Thomas, J Stamatoyannopoulos, M Biggin, and M Eisen. "Quantitative models of the mechanisms that control genome-wide patterns of transcription factor binding during early drosophila development." *PLoS Genet*, **7**:e1001290, 2011.

[KMF09] N Kaplan, I Moore, Y Fondufe-Mittendorf, A Gossett, D Tillo, Y Field, E LeProust, T Hughes, J Lieb, J Widom, and E Segal. "The DNA-encoded nucleosome organization of a eukaryotic genome." *Nature*, **458**:362–366, 2009.

[KPF08] L Kerosuo, K Piltti, H Fox, A Angers-Loustau, V Häyry, M Eilers, H Sariola, and K Wartiovaara. "Myc increases self-renewal in neural progenitor cells through Miz-1." *J Cell Sci*, **121**:3941–3950, 2008.

[KTP08] PV Kharchenko, MY Tolstorukov, and PJ Park. "Design and analysis of ChIP-seq experiments for DNA-binding proteins." *Nat Biotechnol*, **26**:1351–1359, 2008.

[Loh02] Wei-Yin Loh. "Regression trees with unbiased variable selection and interaction detection." *Stat Sinica*, **12**:361–386, 2002.

[LYL09] K Laurila, O Yli-Harja, and H Lähdesmäki. "A protein-protein interaction guided method for competitive transcription factor binding improves target predictions." *Nucleic Acids Res*, **37**:e146, 2009.

[MFG03] V Matys, E Fricke, R Geffers, E Gössling, M Haubrock, R Hehl, K Hornischer, D Karas, AE Kel, OV Kel-Margoulis, DU Kloos, S Land, B Lewicki-Potapov, H Michael, R Münch, I Reuter, S Rotert, H Saxel, M Scheer, S Thiele, and E Wingender. "TRANSFAC: transcriptional regulation, from patterns to profiles." *Nucleic Acids Res*, **31**:374–378, 2003.

[MFP09] M Mason, G Fan, K Plath, Q Zhou, and S Horvath. "Signed weighted gene co-expression network analysis of transcriptional regulation in murine embryonic stem cells." *BMC Genomics*, **10**:doi:10.1186/1471–2164–10–327, 2009.

[MGH02] F Morrish, C Giedt, and D Hockenbery. "c-MYC apoptotic function is mediated by Nrf-1 target genes." *Gene Dev*, **17**:240–255, 2002.

[MKW04] Manuel Middendorf, Anshul Kundaje, Chris Wiggins, Yoav Freund, and Christina Leslie. "Predicting genetic regulatory response using classification." *Bioinformatics*, **20**:i232–i240, 2004.

[MPZ10] M Mason, K Plath, and Q Zhou. "Identification of context-dependent motifs by contrasting chip binding data." *Bioinformatics*, **26**:2826–2832, 2010.

[MRA12] D Marbach, S Roy, F Ay, P Meyer, R Candeias, T Kahveci, C Bristow, and M Kellis. "Predictive regulatory models in Drosophila melanogaster by integrative inference of transcriptional networks." *Genome Res*, **22**:1334–1349, 2012.

[NGH07] L Narlikar, R Gordân, and A Hartemink. "A nucleosome-guided map of transcription factor binding sites in yeast." *PLoS Comput Biol*, **3**:e215, 2007.

[OZW09] Z Ouyang, Q Zhou, and WH Wong. "ChIP-Seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells." *PNAS*, **106**:21521–21526, 2009.

[PLL04] Tu Minh Phuong, Doheon Lee, and Kwang Hyung Lee. "Regression trees for regulatory element identification." *Bioinformatics*, **20**:750–757, 2004.

[RDP09] Jianhua Ruan, Youping Deng, Edward J Perkins, and Weixiong Zhang. "An ensemble learning approach to reverse-engineering transcriptional regulatory networks from time-series gene expression data." *BMC Genomics*, **10**:S8, 2009.

[RKK10] SA Ramsey, T Knijnenburg, K Kennedy, D Zak, M Gilchrist, E Gold, C Johnson, A Lampano, V Litvak, G Navarro, T Stolyar, A Aderem, and I Shmulevich. "Genome-wide histone acetylation data improve prediction of mammalian transcription factor binding sites." *Bioinformatics*, **26**:2071–âĂŞ2075, 2010.

[RLS09] T Raveh-Sadka, M Levo, and E Segal. "Incorporating nucleosomes into thermodynamic models of transcription regulation." *Genome Res*, **19**:1480–1496, 2009.

[RZ06] Jianhua Ruan and Weixiong Zhang. "A bi-dimensional regression tree approach to the modeling of gene expression regulation." *Bioinformatics*, **22**:332–340, 2006.

[SAE04] A Sandelin, W Alkema, P Engstrom, WW Wasserman, and B Lenhard. "JASPAR: an open-access database for eukaryotic transcription factor binding profiles." *Nucleic Acids Res*, **32**:D91–D94, 2004.

[SJ06] Chiara Sabatti and Gareth M James. "Bayesian sparse hidden components analysis for transcription regulation networks." *Bioinformatics*, **22**:739–746, 2006.

[SYK03] E Segal, R Yelensky, and D Koller. "Genome-wide discovery of transcriptional modules from DNA sequence and gene expression." *Bioinformatics*, **19**:i273–i282, 2003.

[Tib96] Robert Tibshirani. "Bayesian shrinkage and selection via the lasso." *JRSSB*, **58**:267–288, 1996.

[TLV11] Konstantin Tretyakov, Sven Laur, and Jaak Vilo. "G =MAT: Linking transcription factor expression and dna binding data." *PLoS One*, **6**:e15449, 2011.

[VPV07] M Verykokakis, C Papadaki, E Vorgia, L Le Gallic, and G Mavrothalassitis. "The RAS-dependent Erf control of cell proliferation and differentiation is mediated by c-Myc repression." *J Biol Chem*, **282**:30285–30294, 2007.

[WH09] T Wasson and A Hartemink. "An ensemble model of competitive multifactor binding of the genome." *Genome Res*, **19**:2101–2112, 2009.

[WRC06] J Wang, S Rao, J Chu, X Shen, DN Levasseur, TW Theunissen, and SH Orkin. "A protein interaction network for pluripotency of embryonic stem cells." *Nature*, **444**:364–368, 2006.

[WRW10] K-J Won, B Ren, and W Wang. "Genome-wide prediction of transcription factor binding sites using an integrated model." *Genome Biol*, **11**:R7, 2010.

[XV05] Biao Xing and Mark Van Der Laan. "A statistical method for constructing transcriptional regulatory networks using gene expression and sequence data." *J Comput Biol*, **12**:229–246, 2005.

[xWD04] Xiaojiang xu, Lianshui Wang, and Dafu Ding. "Learning module networks from genome-wide location and expression data." *FEBS Letters*, **578**:297–304, 2004.

[YTC02] M K Stephen Yeung, Jesper Tegnér, and James J Collins. "Reverse engineering gene networks using singular value decomposition and robust regression." *PNAS*, **99**:6163–6168, 2002.

[ZCM07] Q Zhou, H Chipperfield, D Melton, and WH Wong. "A gene regulatory network in mouse embryonic stem cells." *Proc Natl Acad Sci USA*, **104**:16438–443, 2007.

[Zel83] Arnold Zellner. "Applications of Bayesian analysis in ecnometrics." *JRSSD*, **32**:1–23, 1983.

[ZH07] Achim Zeileis and Kurt Hornik. "Generalized M-fluctuation tests for parameter instability." *Stat Neerl*, **61**:488–508, 2007.

[ZHH08] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. "Model-based recursive partitioning." *J Comput Graph Stat*, **17**:492–514, 2008.

[Zho10] Q Zhou. "On weight matrix and free energy models for sequence motif detection." *J Comput Biol*, **17**:1621–1638, 2010.

[ZW04] Q Zhou and WH Wong. "CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling." *Proc Natl Acad Sci USA*, **101**:12114–12119, 2004.