ModuleFinder: A Computational Model for the Identification of *Cis* Regulatory Modules

by

Fangxue He

M.D.
Soochow University, School of Medicine, 1997

Submitted to Harvard-MIT Division of Health Sciences and Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Biomedical Informatics

at the

Massachusetts Institute of Technology

June 2005

Signature of Author ....
Harvard-MIT Division of Health Sciences and Technology
May 6, 2005

Certified by ....
Martha L. Bulyk
Assistant Professor of Medicine, Pathology and Health Sciences and Technology, HMS, BWH
Thesis Supervisor

Accepted by ....
Martha L. Gray, Ph.D.
Edward Hood Taplin Professor of Medical and Electrical Engineering
Director, Harvard-MIT Division of Health Sciences and Technology

ModuleFinder: A Computational Model for the Identification of *Cis* Regulatory Modules

by

Fangxue He

Submitted to Harvard-MIT Division of Health Sciences and Technology
on May 6, 2005 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Biomedical Informatics

ABSTRACT

Regulation of gene expression occurs largely through the binding of sequence-specific transcription factors (TFs) to genomic DNA binding sites (BSs). This thesis presents a rigorous scoring scheme, implemented as a C program termed "ModuleFinder", that evaluates the likelihood that a given genomic region is a *cis* regulatory module (CRM) for an input set of TFs according to its degree of: (1) homotypic site clustering; (2) heterotypic site clustering; and (3) evolutionary conservation across multiple genomes. Importantly, ModuleFinder obtains all parameters needed to appropriately weight the relative contributions of these sequence features directly from the input sequences and TFBS motifs, and does not need to first be trained. Using two previously described collections of experimentally verified CRMs in mammals as validation datasets, we show that ModuleFinder is able to identify CRMs with great sensitivity and specificity. We also evaluated ModuleFinder on a set of DNA binding site data for the human TFs Hepatocyte Nuclear Factor HNF1α, HNF4α and HNF6 and compared its performance with logistic regression and neural network models.

Thesis Supervisor: Martha L. Bulyk

Title: Assistant Professor of Medicine, Pathology and Health Sciences and Technology, HMS, BWH

Acknowledgments

I would like to thank my advisor Professor Martha Bulyk for her constant advice, favorable support and mentorship in all aspects of my thesis and graduate study.

I owe my special gratitude for my colleague Anthony Philippakis who has offered me great help and advice, and his willingness to share his thoughts with me.

I would like to acknowledge the contribution of John Hayden for his professional Linux support and administration.

My thanks also go to the rest members of Bulyk lab, I had great time with them and I will always cherish the fun moments.

None of this would have been possible, of course, without the constant support of my mother, Yan, to whom I am eternally grateful for her love and support.

Finally, I must thank my other half, Qiang, for all that we share, having the courage to pursue our dreams, and for continuing to grow together even when apart.

Table of Contents

List of Figures

List of Tables

1. Introduction

Recent technological advances have enabled both the sequencing of a large number of genomes and also the generation of expansive gene expression datasets for various organisms for many different cell types under a number of cellular and environmental conditions. Still, little is known about how these gene expression patterns are precisely regulated through the binding of sequence-specific transcription factors (TFs) to their DNA binding sites (BSs). Of particular interest is the organization of TF binding sites (TFBSs) into *cis* regulatory modules (CRMs) that coordinate the complex spatio-temporal patterns of gene expression, and to use that information to identify the CRMs themselves.

The problem of mapping TFs to their CRMs and thus to their target genes, however, is significantly complicated in higher eukaryotic genomes by the large proportion of non-protein-coding sequences, as they frequently contain a very large number of matches to a given set of TFBS sequences, with many of the site occurrences presumably not directly regulating gene expression. Since the BS for a typical TF can be as short as ~5 base pairs (bp), a sequence match can occur on average every few hundred bp just by chance alone. Therefore, a central challenge that must be overcome in mapping TFs to the CRMs is distinguishing functional TFBSs from spurious TFBS motifs matches.

To date, three indicators have been used to identify functional TFBSs. First, functional BSs for some TFs tend to occur in clusters, with multiple BSs occurring in close proximity (homotypic clustering). Second, searching for clusters containing BSs for 2 or more TFs that are believed to co-regulate can enrich for likely CRMs (heterotypic clustering). Finally, functional TFBSs are frequently conserved across evolutionarily divergent organisms[1]. Cross-species sequence conservation in particular has enormous potential for filtering sequence space, as many genomes have recently been sequenced, and many more are slated to be sequenced (http://www.genome.gov/10002154). The discriminatory power of phylogenetic footprinting for identifying *cis* regulatory elements is therefore expected to continue to increase through the use of more genomes[2-6]. In order to appropriately incorporate information on conservation across multiple genomes, however, a measure of TFBS conservation is required that weights each alignment genome according to its evolutionary distance not only from the query genome, but also relative to the other alignment genomes. For example, given a candidate TFBS in the human genome, observing conservation in chicken should be weighted more heavily than conservation in mouse, as mouse is evolutionarily closer to human. Moreover, if the candidate site were also conserved in rat, then this additional conservation should be weighted only slightly, given the evolutionary proximity of mouse and rat.

While numerous groups have developed approaches for the prediction of CRMs, none is optimized for practical applications. Specifically, many approaches[7-9] have been based on binary scoring schemes, wherein all regions containing a threshold number of occurrences for a given combination of TFBSs are returned. These approaches suffer from the limitation that they do not prioritize among the predictions, an important feature for experimentalists as only a limited number of candidate CRMs can feasibly be validated. Additionally, the threshold value determined in any given biological system is unlikely to be generalizeable from one set of TFs and CRM type to another; thus, the appropriate discriminatory criterion must be re-discovered with each application. Alternatively, among existing continuous scoring schemes, many require large training sets[10,11]. Such approaches cannot be applied to a system in which there are only a handful of known examples, as is frequently the case in practical applications. Finally, among approaches that employ continuous scoring schemes and do not require training[12-15], most do not systematically integrate BS clustering and conservation. We are aware of only one other approach that combines all three indicators[16], but it is computationally rather slow and requires the user to specify a single sequence window size for the search. Since CRMs are known to vary greatly in size, a scoring scheme is needed that evaluates clustering and conservation over windows of varying sizes[15].

We have developed a statistically rigorous scoring scheme that for any given genomic region integrates into a single score the degree of: (1) homotypic clustering; (2) heterotypic clustering; and (3) evolutionary conservation across multiple genomes. Similar to programs such as BLAST[17], our score is an objective measure of the statistical significance of the observed degree of clustering and conservation that is independent of the genome and TFBSs under consideration. Thus, the scoring scheme obtains all parameters needed to appropriately weight the relative contribution of each input alignment and TFBS motif directly from the sequences and motifs themselves, and so does not need to first be trained. We have implemented this scoring scheme as a *C* program called "ModuleFinder," (MF) that is algorithmically efficient and has an intuitive interface. ModuleFinder, along with pre-processed genomes and alignments for common model organisms (human, mouse, fruit fly, worm, and yeast), can be obtained from our website (http://the_brain.bwh.harvard.edu/PSB2005MFSuppl/index.html).

The remainder of this thesis is constructed as follows: Chapter 2 shows the model in details in statistics methods and implementation. Chapter 3 and Chapter 4 are dedicated to the applications of ModuleFinder to metazoan genomes. Using previously described collections of experimentally verified mammalian skeletal muscle CRMs[18], we show in Chapter 3 that ModuleFinder is able to identify CRMs with ~95% sensitivity and ~95% specificity. In Chapter 4, we evaluated ModuleFinder on a set of ChIP-chip (genome wide location analysis) binding data for the TFs HNF1 α , HNF4 α and HNF6[19], and compared its performance with logistic regression and neural networks variable selection models. Finally Chapter 5 presents a summary and possible future work to improve ModuleFinder.

2. Methods

This section is essentially the same as the methods section in:
Philippakis AA, He FS, Bulyk ML. ModuleFinder: a tool for computational discovery of cis regulatory modules. Pacific Symposium on Biocomputing. 2005; p. 519-530. (AA Philippakis and FS He contributed equally to this work)
This paper was written by Anthony A. Philippakis and Martha L. Bulyk. I did research on string searching algorithms and decided to use a modified version of suffix arrays[20,21] as our fast string searching algorithm. I implemented the ModuleFinder software in C code.

2.1. Overview and previous work

Methods for evaluating conservation among multiple genomes can be divided into two broad classes: those that evaluate the overall degree of conservation for a given region as a whole, and those that evaluate the conservation of specific TFBSs within the given region. In the first method, multiple sequences are simultaneously aligned using a metric that evaluates the overall degree of conservation for the region of interest. Although this has been successfully applied by numerous groups[2,6] to identify regulatory elements in metazoan genomes, it does not necessarily identify the CRMs through which a given set of TFs are exerting their regulatory roles (i.e., the TF's "target" CRMs).

Since our ultimate goal is to identify candidate CRMs that are targets of a given set of TFs that are known or suspected to be important in transcriptional regulation for a particular biological system, a scoring scheme that specifically considers the conservation of the input BSs is needed. To date, most approaches have focused on the case of one alignment genome, in part because of the only recent availability of additional, sufficiently closely related, alignment genomes. Additionally, several statistical difficulties must be addressed in order to account for correlations within the multi-species alignments. This question has recently been addressed by several groups. Blanchette *et al.* formalized the "substring parsimony problem", which seeks to find the set of all substring of a given length with a certain parsimony score (i.e. level of conservation)[22]. They presented a rigorous and efficient algorithmic procedure for efficiently solving the substring parsimony problem for multiple species; however, their procedure required the simplification of ignoring relative branch lengths within the phylogenetic tree. For example, conservation among human, chimp and chicken would be treated identically to conservation among human, mouse and chicken. In a later approach, Moses *et al.* used mixture models to evaluate conservation within a tree. This approach was developed and optimized for the problem of beginning with a set of co-expressed genes in order to identify candidate DNA sequence motifs responsible for the genes' co-expression[23]. Prakash *et al.* also presented a similar approach for motif finding using conservation[24]. Here we present a conceptually similar approach for the inverse problem of beginning with a set of TFBSs motifs in order to identify candidate CRMs.

2.2. Scoring scheme

We define a *word* to be a short sequence on the DNA alphabet {A, C, G, T}, and a *motif* to be a collection of words all of the same length. ModuleFinder takes as input a collection of arbitrarily many motifs $\{m_1 \ldots m_m\}$, where each motif m_i is composed of arbitrarily many words of length l_i. It also takes as input a set of sequences $G = \{g_1, \ldots g_n\}$ corresponding to genomic regions that are to be searched for instances of these motifs, as well as two sets of genomic sequences, $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$, extracted from evolutionarily divergent organisms and then aligned to the sequences of G. Here, we primarily illustrate the scoring scheme for the case of two alignment genomes, but include comments on the extension to fewer or more alignments. For any g_j, let $g_{j,k}$

denote the base at the kth position and $(g_{j,k}\ldots g_{j,k+l})$ denote the subsequence of length l beginning at position k. If there is a match to a given motif m_i at position k of sequence g_j, we define it to be *conserved in A* (respectively, B), if it is true that the subsequence $(a_{j,k}\ldots a_{j,k+l})$ (respectively, $(b_{j,k}\ldots b_{j,k+l})$) is also a word in motif m_i. Note that we are not assuming that $g_{j,k}\ldots g_{j,k+l} = a_{j,k}\ldots a_{j,k+l}$, but merely that they are both words in m_i.

Our basic approach is to scan each sequence in G with a series of nested windows (i.e., overlapping windows of differing sizes). In each window we count the number of occurrences of each motif and the number of these that are conserved in A and B. We then evaluate the likelihood of observing this number of matches and conserved matches under the appropriate null hypothesis, and return those windows that are statistically significant. Specifically, let $X = (X_1, \ldots, X_m)$ be the vector whose components indicate the number of occurrences for each motif individually in a given window, and let $Y = (Y_1, \ldots, Y_m)$ and $Z = (Z_1, \ldots, Z_m)$ be the corresponding vectors indicating that Y_i and Z_i out of X_i occurrences are conserved in A and B, respectively. The window score is obtained by finding the probability of observing (X,Y,Z). This quantity will vary according to the likelihood of conservation in A and/or B, the motif frequency, and the window width. Thus, this probability can be represented by:

$$P_{\Gamma,\alpha,w}(X,Y,Z) \qquad (1)$$

where Γ parameterizes conservation likelihood, α parameterizes motif frequencies, and w is the window width. Observe that:

$$P_{\Gamma,\alpha,w}(X,Y,Z) = P_{\Gamma}(Y,Z \mid X)P_{\alpha,w}(X) \qquad (2)$$

where the relevant parameters can be split between terms in the Markov decomposition, as $P_{\alpha,w}(X)$ is unaffected by conservation likelihood, and $P_{\Gamma}(Y,Z\,|\,X)$ is unaffected by motif frequency and window size.

For a single motif m_i, the term $P_{\alpha_i,w}(X_i)$ of Eq. (2) is the likelihood of observing X_i occurrences under the null hypothesis that the motif matches are distributed at random. This has been proved to be well-approximated by a Poisson distribution, provided the motif occurs infrequently and the words comprising it do not exhibit extensive self-overlap[25]. Thus, $P_{\alpha_i,w}(X_i)=e^{-\lambda_i}(\lambda_i^{X_i}/X_i!)$, where $\lambda_i=\alpha_i*w$. The value of α_i will itself be determined by both the words comprising m_i, as well as genomic word frequencies. To obtain it, we estimate the frequency of each word in m_i by a seventh order Markov approximation based on genomic word frequencies, and then sum these frequencies for all words in the motif.

For multiple motifs, the joint probability is given by assuming independence:

$$P_{\alpha,w}(X_1,\ldots X_m)=\prod_{i=1}^{m}\left(P_{\alpha_i,w}(X_i)\right)$$

This is a simplifying assumption to make the computation tractable; the error in this approximation has, however, been proved to be bounded.

The computation of the second term of Eq. (2), $P_{\Gamma}(Y,Z\,|\,X)$, is complicated by two factors. First, the score must reflect not only the evolutionary distances of A and B to G, but also the distances of A and B to each other. Thus, Γ must re-parameterize $P_{\Gamma}(Y,Z\,|\,X)$ so that it becomes smaller as A and B grow more distant from G, and as the correlation between A and B decreases. Second, the quantity $P_{\Gamma}(Y,Z\,|\,X)$ will depend not only on the phylogeny of A, B and G, but also on the degeneracy of the motifs m_i. Since we have defined a given motif match to be conserved in A or B if there is a motif occurrence (but not necessarily an exact word match) at the same position in these aligned sequences, a more degenerate motif has a greater likelihood of being conserved.

We account for these difficulties as follows. Define $\Gamma^{1}_{A,B}$ to be the covariance matrix representing the relative proportions of A and B that can be aligned against G; thus, $\Gamma^{1}_{0,0}$ gives the proportion of sequence in G for which neither A nor B could be aligned, $\Gamma^{1}_{1,0}$ and $\Gamma^{1}_{0,1}$ give the proportion for which either A or B (but not both) could be aligned, and $\Gamma^{1}_{1,1}$ gives the proportion for which both A and B could be aligned. Similarly, for each motif m_i, define $\Gamma^{i,2}_{A,B}$ to be the covariance matrix representing the relative likelihoods of exact conservation of l_i positions (i.e., $(g_{j,k}\ldots g_{j,k+l}) = (a_{j,k}\ldots a_{j,k+l})$) in A and/or B. Here, we have observed non-independence of exact conservation likelihood between adjacent positions, so we model it as a first order Markov chain.

Conservation of a completely degenerate motif is parameterized by $\Gamma^{1}_{A,B}$, and conservation of a motif composed of a single word is parameterized by $\Gamma^{i,2}_{A,B}$. The parameterization of a generic motif is between these extremes; for this, let $P_{i,j,k}$ be the matrix giving the frequency of nucleotide $j \in \{\text{A,C,G,T}\}$ at position $k \in \{1,\ldots,l_i\}$ in motif m_i, and let E_i be the average entropy of the motif:

$$E_i = -\frac{1}{2l_i}\sum_{k=1}^{l_i} \sum_{j\in\{A,C,G,T\}} P_{i,j,k} \log_2 P_{i,j,k}$$

Hence, E_i=1 for a completely degenerate motif, E_i=0 for a motif composed of a single word, and E_i increases monotonically and smoothly between these extremes as the motif degeneracy increases. Therefore, we take our parameterization of *Γ_i* for *m_i* to be a weighted average of $\Gamma^{1}_{A,B}$ and $\Gamma^{i,2}_{A,B}$:

$$\Gamma^{i} = E_i\Gamma^{i,1}_{A,B} + (1 - E_i)\Gamma^{i,2}_{A,B}$$

We then use *Γ_i* to compute $P_{\Gamma^i}(Y_i, Z_i \mid X_i)$. In a sequence window containing *X_i* matches to motif *m_i*, let *a_i* be the number that are not conserved in either *A* or *B*, let *b_i* and *c_i* be the number conserved in either *A* or *B* (but not both), and let *d_i* be the number that are conserved in both *A* and *B*. The following equations hold:

$$a_i + b_i + c_i + d_i = X_i \qquad (3\text{-}5)$$

$$b_i + d_i = Y_i \qquad\qquad c_i + d_i = Z_i$$

$P(Y_i,Z_i|X_i)$ is therefore given by the following multinomial:

$$P_{\Gamma^i}(Y_i, Z_i \mid X_i) = \sum\left(\frac{X_i!}{a_i!b_i!c_i!d_i!}\right)\left(\left(\Gamma^{i}_{0,0}\right)^{a} \cdot \left(\Gamma^{i}_{1,0}\right)^{b} \cdot \left(\Gamma^{i}_{1,0}\right)^{c} \cdot \left(\Gamma^{i}_{1,1}\right)^{d}\right) \qquad (6)$$

where the summation is performed over all values of *a_i, b_i, c_i* and *d_i* satisfying Eqs. (3)-(5). To achieve computational efficiency, we make use of the following 1-dimensional parameterization, where *X_i*, *Y_i* and *Z_i* remain fixed as *d_i* is varied:

$$a_i = X_i - Y_i - Z_i + d_i \qquad (7\text{-}9)$$

$$b_i = Y_i - d_i \qquad\qquad c_i = Z_i - d_i$$

Thus, the summation of Eq. (6) can be performed by simply taking each value of *d_i* in the range $0 \le d_i \le \min(Y_i, Z_i)$.

If one desires to only input one genome, it is sufficient to set A=B. The relevant parameters then simplify, and the preceding multinomial distribution collapses to a binomial distribution with parameter $\gamma_i = \Gamma^i_{1,1}$:

$$P_{\gamma_i}(Y_i \mid X_i) = \binom{X_i}{Y_i} \gamma^{Y_i}(1-\gamma_i)^{X_i - Y_i}$$

This parameterization can also be easily generalized to more than 2 alignment genomes by replacing the matrix Γ^i with an appropriate tensor.

This derived value of $P_{\Gamma,\alpha,w}(X,Y,Z)$ alone is insufficient for determining statistical significance, since a measurement of distance into the appropriate tail of the distribution is also required. Therefore, we perform a summation of $P_{\Gamma,\alpha,w}(X,Y,Z)$ extending from the observed value of (X,Y,Z) and including all values of (X,Y,Z) with an increased degree of clustering and conservation (we use log values to simplify the numerical analysis):

$$\begin{aligned} S_{\Gamma,\alpha,w}(X,Y,Z) &= \log_{10}\left(\sum_{\tilde{X}=X}^{\infty} \sum_{\tilde{Y}=Y}^{\tilde{X}} \sum_{\tilde{Z}=Z}^{\tilde{X}} \prod_{i=1}^{m} \left(P_{\Gamma^i,\alpha_i,w}(\tilde{X}_i,\tilde{Y}_i,\tilde{Z}_i) \right) \right) \\ &= \sum_{i=1}^{m} \log_{10}\left(\sum_{\tilde{X}_i=X_i}^{\infty} \sum_{\tilde{Y}_i=Y_i}^{\tilde{X}_i} \sum_{\tilde{Z}_i=Z_i}^{\tilde{X}_i} P_{\Gamma^i,\alpha_i,w}(\tilde{X}_i,\tilde{Y}_i,\tilde{Z}_i) \right) = \sum_{i=1}^{m} S_{\Gamma^i,\alpha_i,w}(X_i,Y_i,Z_i) \end{aligned} \qquad (10)$$

Therefore, the output score $S_{\Gamma,\alpha,w}(X,Y,Z)$ for a given window is the linear sum of scores for the input motifs, $S_{\Gamma_i,\alpha_i,w}(X_i,Y_i,Z_i)$, where each such term has been automatically weighted so that more degenerate motifs contribute less. Observe also that $S_{\Gamma_i,\alpha_i,w}(X_i,Y_i,Z_i)$=0 if and only if X_i=0, and that $S_{\Gamma_i,\alpha_i,w}(X_i,Y_i,Z_i)$ increases monotonically with increasing values of (X_i,Y_i,Z_i), as desired.

2.3. Implementation

2.3.1. String searching algorithm

To minimize the runtime, we pre-process each sequence of G with a refined suffix array[20] in order to efficiently identify the locations of all motif matches. Irving and Love has defined suffix binary search tree (SBST)[21], which is competitive in practice with classical suffix trees and suffix arrays to facilitate efficient on-line string searching. A SBST of a string σ is essentially a binary search tree with n = $|\sigma|$ nodes, with each node representing a suffix of σ. Given two strings α and σ, α is a substring of σ if and only if α is a prefix of a suffix of σ. It follows that to find a position of σ where α occurs as a substring, it suffices to search an SBST of σ to find a string with a prefix that matches the string α. Then the suffix array can be viewed as a perfectly balanced SBST and can be constructed from the SBST. Using this method, the construction of the tree can be accomplished in $O(nh)$ where the n is the length of string σ and h is the height of the tree. The substring α with length m can be searched in $O(m+\log n)$ time on average.

2.3.2. Window score look-up method

The sequences of *A* and *B* are only searched for motif matches at those positions for which *G* has a motif match, and a look-up table is kept as ModuleFinder runs that contains a list of scores for all window sizes *w* and values (X, Y, Z) for observed motif matches. Thus, when encountering each subsequent window, ModuleFinder checks to see if the corresponding score has already been computed.

2.3.3. Additional features

Our implementation also contains two additional features designed to improve the practical applicability of the program. First, it is known that transcription factors frequently bind to DNA as homo- and hetero-dimers. We have therefore added to ModuleFinder the ability to take pairs of TFBSs as input, along with minimum and maximum values for the spacing between sites. The score of the dimer is computed by first evaluating the probability of each motif in the dimer as in Eq.1, and taking the probability of observing the dimer to be the product of the individual motif probabilities at a given spacing and then summing these values over all spacings in the user-specified range. These probabilities are then summed as in Eq. (17) to give the score for each dimer.

Second, while applying ModuleFinder we observed that small adjustments to the stated locations of insertions and deletions in the alignment files often resulted in the creation of new conserved binding sites. Therefore, ModuleFinder allows a certain amount of "wiggle room" to compensate for the potential existence of small, local mis-alignments. Specifically, ModuleFinder takes a user-specified parameter r, and calls the given motif match $(g_{j,k}\ldots g_{j,k+l})$ conserved in A if there is any subsequence of $(a_{j,k-r}\ldots a_{j,k+r+l})$ that is a word in m_i. Although this does increase the likelihood of conservation, it has been our experience that this effect is miniscule for small values of r $(1\leq r \leq 5)$, and has frequently helped to identify many potentially conserved binding sites that would have been missed otherwise.

2.3.4. Software interface and performance

ModuleFinder has been implemented in *C*. It has an intuitive interface, and takes as input FASTA-formatted sequences. It returns all windows scoring above a user-defined cutoff in each sequence. Adjacent and overlapping windows scoring above this cutoff are fused into longer regions, and the fused region coordinates, along with the location and score of the maximum scoring sub-window are returned as output.

ModuleFinder can scan approximately 120 Mb/hr using window sizes ranging from 300 to 700 bps with an increment size of 50 bps on a Pentium 4 computer.

The compiled code, along with README files and appropriately formatted genomes and alignments based on the latest UCSC assemblies are available for download at our website.

3. Evaluation of ModuleFinder on Human Skeletal Muscle CRMs

This section is essentially the same as the results section in:
Philippakis AA, He FS, Bulyk ML. ModuleFinder: a tool for computational discovery of cis regulatory modules. Pacific Symposium on Biocomputing. 2005; p. 519-530. (AA Philippakis and FS He contributed equally to this work)
This paper was written by Anthony A. Philippakis and Martha L. Bulyk. I pre-processed the positive controls and negative controls, and wrote Perl scripts that we ran ModuleFinder on these datasets.

3.1. Dataset

3.1.1. Positive controls

In order to evaluate ModuleFinder, we used a set of positive control regions previously compiled by Wasserman *et al.*[18]. This test dataset comprises 27[a] skeletal muscle CRMs that have been demonstrated to direct transcription in skeletal muscle or a suitable cell-culture model system[18]. Each region contains a validated BS for at least one of the following 5 TFs: the Myf family (total of 39 TFBSs in the positive control set), Mef2 (26 TFBSs), SRF (20 TFBSs), Tef (12 TFBSs) and Sp1 (13 TFBSs). Each region was searched against human genome 16 using UCSC BLAT program[26] to find the region location on the human genome. Of these 27 regions, 23 are located upstream of translation start, and 4 are located within introns (we used translational start/stop, as transcriptional start/stop are more poorly annotated in human at present).

3.1.2. Negative controls

As negative controls, 1000 regions of size 200 bp were randomly selected to positionally match the positive control regions: 852 (=(23/27)*1000) regions were within 5 kb of

[a] The original collection gave 28 genes, but we removed the gene *Rb* as there were no confirmed TFBSs for the listed TFs.

translational Start for a randomly chosen RefGene[26] gene, and the remaining 148 were within introns. This matching of chromosomal locations was performed as ModuleFinder accounts for local word frequencies, which vary throughout the genome; in particular, promoter regions are known to be GC-rich.

3.2. Results

We ran ModuleFinder on the positive and negative control regions with window sizes of 100-200 bp (increment size = 10 bp), using human sequence alone, human/mouse/rat (H/M/R) alignments and human/mouse/chicken alignments (H/M/C) obtained from USCS Genome Browser, hg16[26]. Currently, two alternative strategies for representing TFBSs have been used by various groups in computational searches for CRMs: exact word matches to known BSs[9,15], and position weight matrices (PWMs)[7,10-13], which allow for extrapolation to additional BSs. To determine which of these representations had greater discriminatory power, we performed our searches both ways, using a PWM threshold value of 1 standard deviation (SD) below the motif average[27]. We used a "jack-knife" strategy[11] for these searches, whereby the BSs for each CRM were excluded from the construction of the PWM used to search that CRM, and similarly the exact word matches from each CRM were excluded in the search of that CRM. In addition, since *in vitro* selections (SELEX) had been performed for Mef2[28] and SRF[29], we also added those BSs to both searches.

Table 3-1: ModuleFinder results on a human skeletal muscle dataset

	Human Alone		Human/Mouse/Rat		Human/Mouse/Chicken	
	Exact	PWM	Exact	PWM	Exact	PWM
Sens.	88.9%	92.6%	92.6%	96.3%	92.6%	92.6%
Spec.	90.1%	89.2%	88.8%	94.4%	87.4%	94.4%
***p*-val**	$1.19\text{x}10^{-8}$	$6.5\text{x}10^{-10}$	$2.5\text{x}10^{-10}$	$1.4\text{x}10^{-10}$	$4.5\text{x}10^{-10}$	$7.1\text{x}10^{-10}$

The results of these evaluations are shown in **Table 3-1**. Here, we have reported those values for sensitivity and specificity which maximally discriminate between the positive and negative control sets (i.e., using the threshold score such that the difference between the sensitivity and specificity is minimized). Since there was great variability in score among the positive control regions (see **Figure 3-1**; i.e., the top positive control region received a score of -11.23 and the worst positive control region scored only -1.22 (positive controls: mean = -4.69, SD = 2.24; negative controls: mean = -0.42, SD = 0.81)), we also performed a t-test on the positive and negative control region means, in order to measure the effectiveness of ModuleFinder on regions falling far from the threshold score.

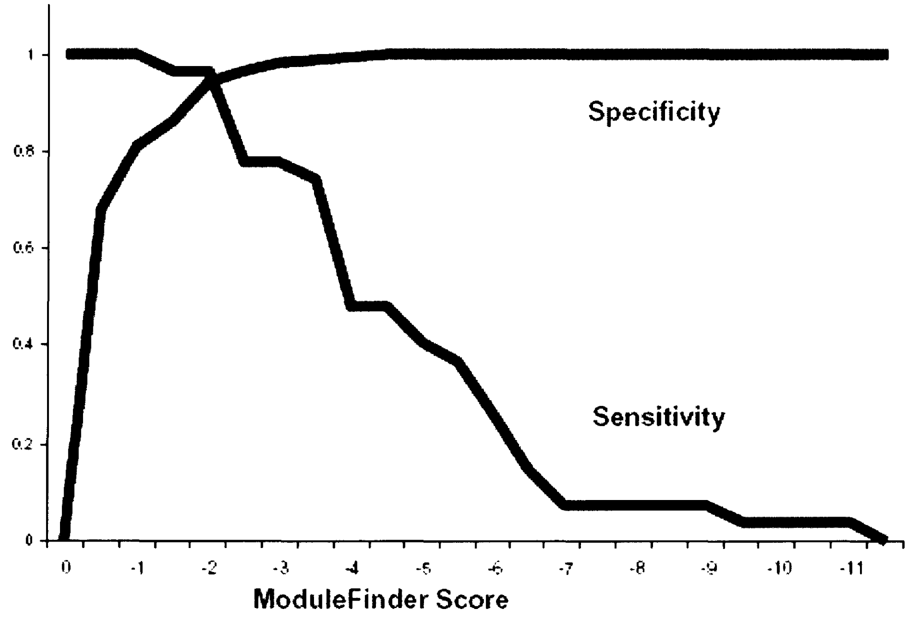


Figure 3-1: Sensitivity and specificity of ModuleFinder on skeletal muscle test regions

On this dataset, ModuleFinder achieved a maximum sensitivity of 96.3% and specificity of 94.4% on the H/M/R PWM searches. Moreover, the PWM approach consistently gave better discrimination than exact word matches. Much of this improved discrimination, however, is an artifact of the jack-knife procedure, which has a stronger effect on exact match searches. Here, using the complete set of BSs (i.e., without the jack-knife), exact word matching achieves 100% sensitivity and 95.1% specificity (we removed degenerate flanking sequences for all searches with exact words). In addition, these results indicate that the H/M/R searches reliably outperformed the H/M/C searches. There are two possible explanations for this: 1) the chicken genome is not yet complete, and the appropriate alignment regions may not have been sequenced yet; 2) the underlying mechanisms of transcriptional regulation are not actually conserved in an organism as distant as chicken. Neither of these hypotheses can be ruled out until the completion of the chicken genome.

Since ModuleFinder was specifically developed to integrate homotypic clustering, heterotypic clustering, and conservation, we wanted to determine which of these features were most contributory to discriminatory power. In order to assess this, we ran ModuleFinder on the positive and negative control regions using no alignment genomes, one alignment genome (each of mouse, rat and chicken), and two alignment genomes (H/M/R and H/M/C). These searches were repeated with each TF individually, as well as with all 5 TFs together. In **Figure 3-2**, we show the negative logarithm of the *p*-values obtained from t-tests on the positive versus negative control regions for each of these searches. Here the mouse and rat alignments improved discriminatory power, but little was gained by using both genomes, because of their evolutionary proximity. Somewhat surprisingly, using chicken actually reduced discrimination relative to human alone. This was unexpected, as it implies that our negative controls are more likely to be conserved than these 27 regions. However, this effect could be an artifact of the small size of the positive controls and gaps in the chicken genome (only 13/27 positive controls had any alignable chicken sequence).

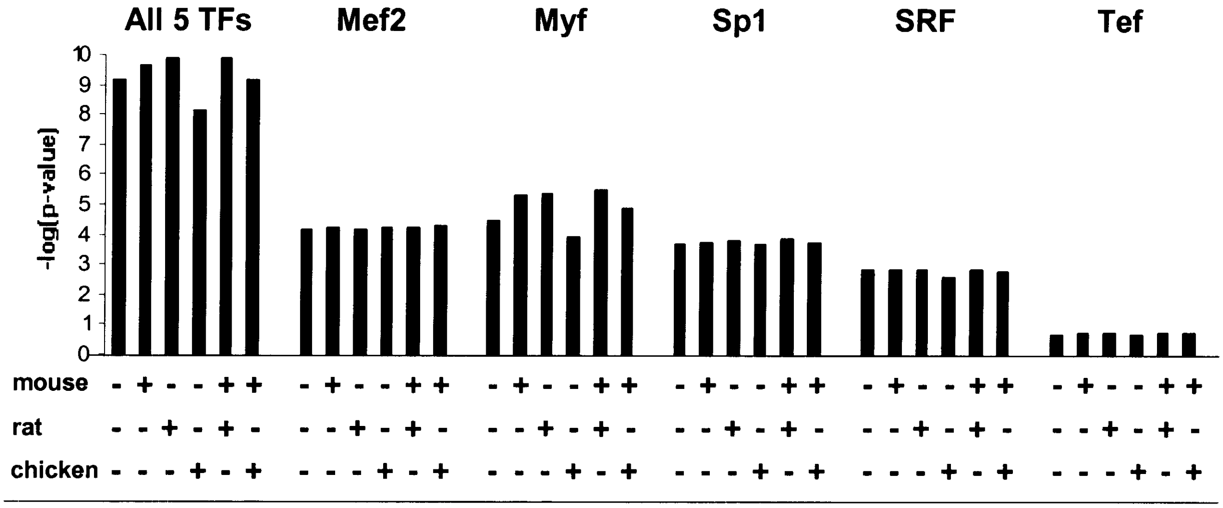


Figure 3-2: Negative log of p-values obtained from t-test between positive and negative controls

Finally, at least four other algorithms have used overlapping subsets of this dataset as positive controls, achieving sensitivities between 59% and 66%, and specificities between 95.3% and 97.1% (see **Table 3-2**). Thus, ModuleFinder appears to have comparable specificity but greater sensitivity. However, note the following caveats for this comparison. First, because ModuleFinder uses evolutionary conservation as a central component and because few vertebrate genomes have been sequenced, we limited our searches to the subset of the original compilation for which human/rodent alignments were available. The other algorithms tested on this dataset did not consider conservation, and thus used the original, larger compilation that included CRMs obtained from diverse organisms including chicken, hamster, rabbit, pig and cow. Frith *et al.* trimmed this larger set to a subset of 27 regions, but their subset overlapped with ours by only 15 genes.

Second, each group used a different set of negative controls. The original paper by Wasserman *et al.* used a set of negative control regions similar to our set; it was composed of 200 bp regions selected from the Eukaryotic Promoter Database. Comet and Cister were each tested on 300 bp regions that were selected to overlap well-characterized transcriptional Starts. Finally, MSCAN[30] measured specificity by looking at the "hit rate" in contiguous stretches of the *Fugu* genome.

Table 3-2: Relative performance of ModuleFinder

Algorithm	**Sensitivity**	**Specificity**
Logistic Regression[11]	60%	96%
Cister[12]	59%	97.1%
COMET[13]	59%	95.3%
MSCAN[30]	66%	N/A
ModuleFinder	96%	94%

4. Evaluation of ModuleFinder on HNFs ChIP-chip Binding Data

4.1. Background

The genome-wide location analysis method[31](also called ChIP-chip) allows protein-DNA interactions to be monitored across the entire genome. The method combines a **ch**romatin **i**mmuno**p**recipitation (ChIP) procedure, with DNA microarray (chip) analysis. Briefly, cells are fixed with formaldehyde, harvested, and disrupted by sonication. The DNA fragments crosslinked to a protein of interest are enriched by immunoprecipitation with a specific antibody. After reversal of the cross links, the enriched DNA is amplified and labeled with a fluorescent dye (Cy5) with the use of ligation-mediated-polymerase chain reaction (LM-PCR). A sample of DNA that was not enriched by immunoprecipitation is also subjected to LM-PCR, in the presence of a different fluorophore (Cy3), and both immunoprecipitation (IP) enriched and unenriched pools of labeled DNA are hybridized to a single DNA microarray containing intergenic sequences of the genome.

Extensive studies revealed that the Hepatocyte Nuclear Factor (HNF) HNF1 α (a homeodomain protein), HNF4 α (a nuclear receptor), and HNF6 (a member of the onecut family) often operate cooperatively in the liver and pancreatic islets, and are required for normal function of liver and pancreatic islets. Mutations in HNFs are causes of the type 3 and type 1 forms of the maturity-onset diabetes of the young (MODY3 and MODY1), a genetic disorder of the insulin-secreting pancreatic beta cells. Odom *et al.* performed ChIP-chip on HNF1 α , HNF4 α , and HNF6, using microarrays containing the proximal promoter regions of the liver and pancreatic islet genes whose expression is regulated by these transcription factors to identify systematically the genes occupied by HNF1 α , HNF4α and HNF6 (**Figure 4-1**). They then used this information to map the transcriptional circuitry in human liver and pancreatic islets[19].

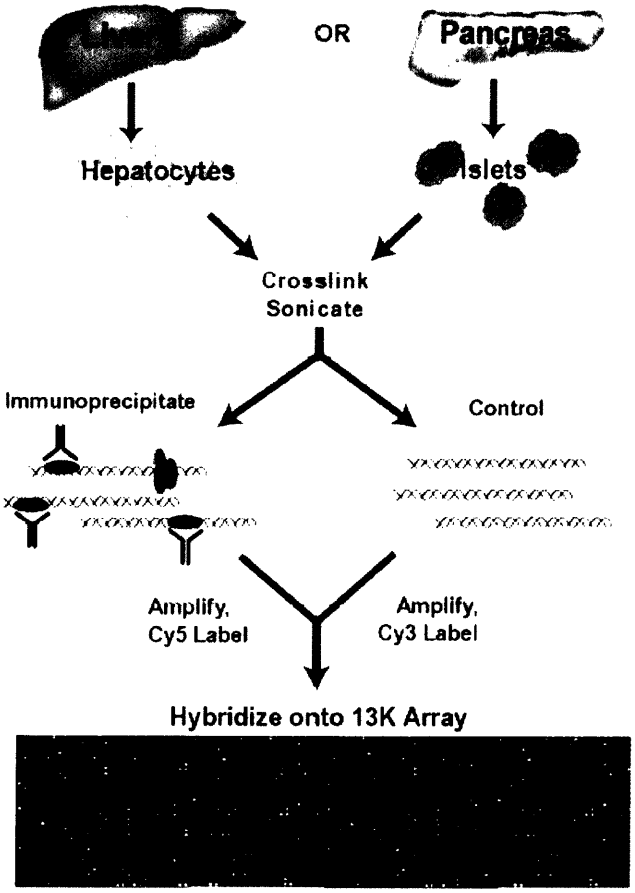


Figure 4-1: ChIP-chip analysis of HNF regulators in human tissues (Reproduced from Odom *et al.*[19])

To evaluate how well ModuleFinder can predict the genes regulated by these three transcription factors, we ran ModuleFinder on the ~7400 proximal promoter regions sequences printed onto the hu13k array and compared the results of ModuleFinder with the ChIP-chip binding data. We also used logistic regression and neural networks as alternative prediction models, and compared their performances with that of ModuleFinder.

4.2. Datasets

4.2.1. Promoter region sequences

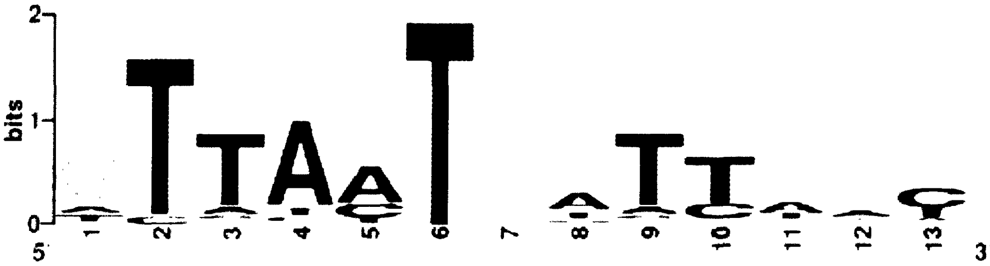
In the ChIP-chip experiment, a custom DNA microarray containing portions of promoter regions of ~13,000 human genes (Hu13K array) was constructed to identify enriched DNA fragments. Because a significant percentage of TFBSs in proximal promoters are located within 1 kb of transcription start site, primers in the Odom *et al.*[19] study were designed to amplify the genomic region -750 bp to +250 bp relative to the transcriptional start sites. Each of these sequences was BLATed against the hu16 genome and ~7400 sequences were found to be actually located at the promoter regions. For our evaluation of different computational models, we took sequences within 2 kb upstream of the transcriptional starts of these ~7400 genes and their mouse alignment sequences.

4.2.2. HNFs transcription factor binding site data

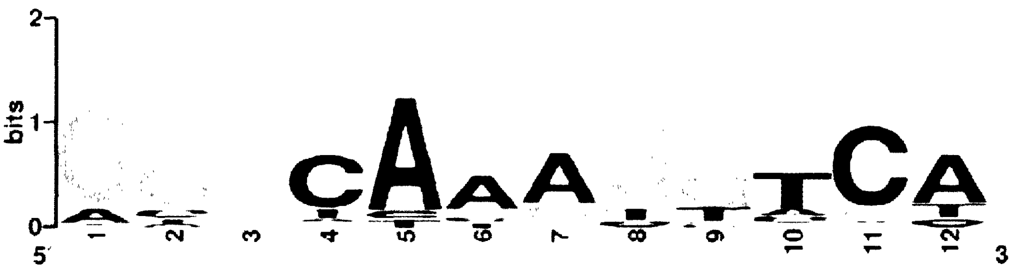
We used the HNFs transcription factor binding sites compiled in the TRANSFAC database[29] to search for motif matches in the ~7400 promoter regions. **Figure 4-2** shows the sequence logos for HNF1 α , HNF4 α and HNF6 generated by WebLogo[32,33]. A sequence logo is a graphic representation of an aligned set of binding sites. It displays the frequencies of bases at each position, as the relative heights of letters, along with the degree of sequence conservation as the total height of a stack of letters, measured in bits of information content[34], which is defined to be

$$\mathrm{I}_i = 2 + \sum_{b=A}^{T} f_{b,i} \log_2 f_{b,i}$$

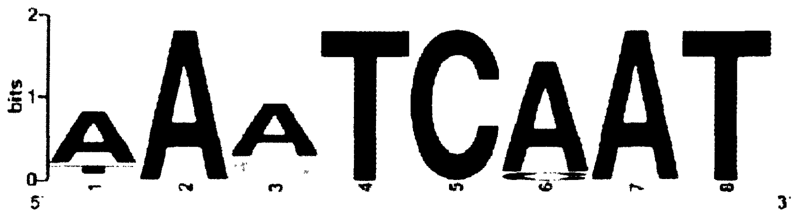
where i is the position with the site, b refers to each of the possible bases, and $f_{b,i}$ is the observed frequency of each base at that position. I_i is between 0, at positions that contain the four base with equal frequency, and 2 bits for positions that permit just one particular base.



HNF1 α TRANSFAC binding site sequence logo



HNF4 α TRANSFAC binding site sequence logo



HNF6 TRANSFAC binding site sequence logo

Figure 4-2: Sequence logos of TRANSFAC binding sites for HNFs

Two alternative representations of TFBSs (exact word matches and PWM) could be used in this analysis. To determine which one had better discriminatory ability and therefore to determine which representations of TFBSs we should use, we ran ModuleFinder on human/mouse/rat alignments to perform searches with the three HNFs together, using exact word matches and separately using a PWM threshold value of 2 SD below the motif average.

Table 4-1: ModuleFinder results on HNFs ChIP-chip binding data using two motif representations

	Exact	PWM
Sensitivity	0.59	0.56
Specificity	0.65	0.55
p-value	0.17	0.21

The results are shown in **Table 4-1.** We used the ModuleFinder threshold score that minimizes the difference between the sensitivity and specificity and reported the values for sensitivity and specificity which maximally discriminate between HNFs ChIP-chip positive regions and HNFs ChIP-chip negative regions. We also performed t-tests on these two regions, in order to measure the effectiveness of ModuleFinder on regions falling far from the threshold score.

Since exact word matches had slightly higher sensitivity and specificity, although none of the results showed significant discriminatory power of ModuleFinder, we chose to use the exact word matches as our motif representation. Note that the HNF1 motif has a palindromic pattern, with forward half-sites having good matches to the consensus GTTAAT, while the reverse half-sites appear to be more degenerate. The original HNF1 sites from TRANSFAC were then constructed as dimer motifs; each contains words of length 6.

4.3. Logistic regression and neural networks Models

4.3.1. Logistic regression

Logistic regression (LR) (**Figure 4-3**) models the dependence of a dichotomous (yes/no) outcome variable on a set of observed variables that may be continuous, discrete, dichotomous, or a mix of any of these. The dependent variable can take the value 1 with a probability of success θ, or the value 0 with probability of failure 1 - θ.

Logistic regression makes no assumption about the distributions of the independent variables. They do not have to be normally distributed, linearly related, or of equal variance within each group. Logistic regression also provides knowledge of the relationships and strengths among the variables. The relationship between the predictor and response variables is not a linear function in logistic regression; instead, the logistic regression function is used, which is the logit transformation of θ:

$$\theta = \frac{e^{Logit}}{1 + e^{Logit}}$$

where the logit function is

$$\text{Logit} = \beta_0 + \beta_1 x_1 + \ldots + \beta_i x_i$$

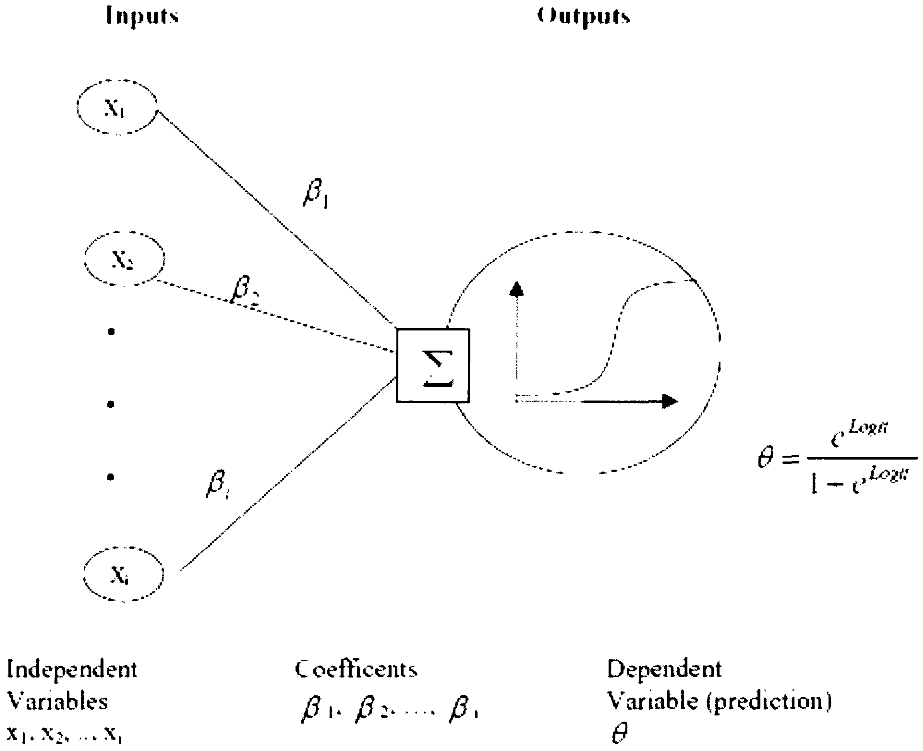


Figure 4-3: Logistic regression model

4.3.2. Neural networks

Neural networks (NN) are analytical techniques modeled after the hypothesized processes of learning in the cognitive system and the neurological functions of the brain. They are capable of predicting new observations (on specific variables) from other observations (on the same or other variables) after executing a process of so-called learning from existing data.

Inspired by the structure of the brain, a neural network consists of a set of highly interconnected entities, called *nodes* or *units*. Each unit is designed to mimic its biological counterpart, the neuron. Each accepts a weighted set of inputs and responds with an output.

A simple network has a feedforward structure (**Figure 4-4**): signals flow from inputs, forwards through any hidden units, eventually reaching the output units. A typical feedforward network has neurons arranged in a distinct layered topology. The input layer serves to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer.

When the network is executed, the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.

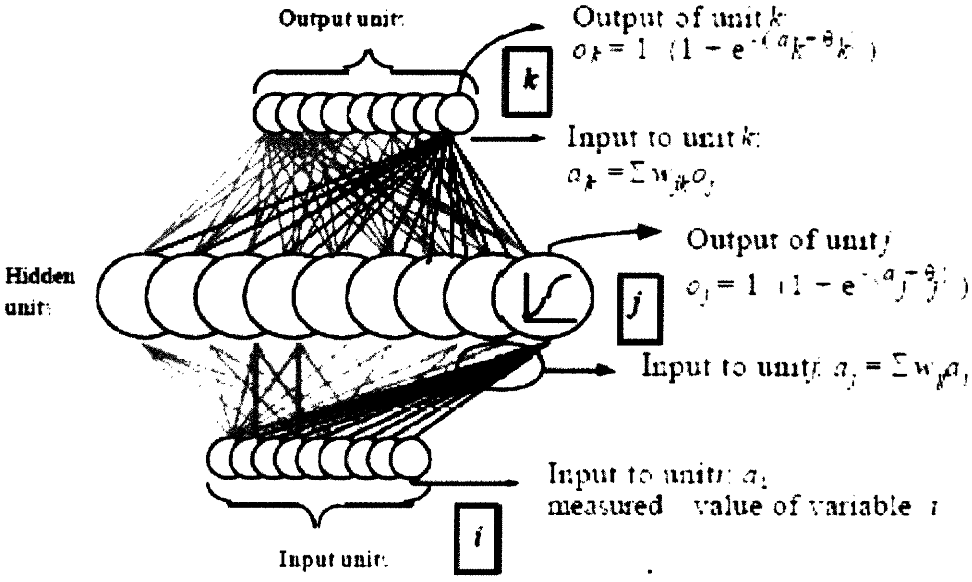

Output unit
Output of unit k
o_k = 1 / (1 - e^-(a_k - θ_k))
k
Input to unit k:
a_k = Σ w_jk o_j
Output of unit j
o_j = 1 / (1 - e^-(a_j - θ_j))
Hidden unit
j
Input to unit j: a_j = Σ w_ij a_i
Input to unit i: a_i
measured value of variable i
i
Input unit

Figure 4-4: Neural networks model

4.3.3. Variable selection

As stated in the first chapter, three indicators have been used to identify functional TFBSs (homotypic clustering, heterotypic clustering and evolutionary conservation across multiple genomes). To evaluate how interaction between the HNF under consideration and other two TFs can influence the model's predictive power, each HNF factor was modeled individually using LR and NN. The heterotypic clustering was not considered to be a predictor; instead, this information was encoded by the TF interaction scores of the other two TFs (see below). We used the following two variables to quantify the degree of homotypic clustering and evolutionary conservation:

Homotypic Clustering: each promoter region was scanned by searching the word matches (occurrences) of the HNFs TRANSFAC TFBSs, and the number of matched sites was used as a measurement of degree of clustering for each particular HNF factor.

Evolutionary Conservation: the number of the HNF TFBSs motif matches in the human sequence that were conserved in the mouse sequence.

Furthermore, we were interested in investigating the degree of binding affinity in the TFBSs prediction and therefore included this variable as another predictor. Note that we don't have the actual binding affinity data for these DNA-protein interactions. Instead, we used PWM score as a proxy for affinity; this approach has been used before[27,35]:

Binding affinity:

Let *l* be the motif length for the HNF factor. Each possible *l*-mer in the promoter sequences was scored according to the following scoring function:

$$\sum_{i=1}^{l} \log(\frac{M[i,b_i]+\sqrt{N}}{N+4\sqrt{N}})$$

where *len* is the length of motif, *N* is the number of compiled binding sites in the TRANSFAC database, and $M[i, b_i]$ is the entry in the motif PWM for nucleotide b_i at position *i*. We then measured the binding affinity using the maximum score.

Among the regions bond by HNF1/4/6 *in vivo*, we examined those sequences that had HNF1 α / HNF4 α /HNF6 TRANSFAC binding site motif matches and plotted the binding sites positional bias for all three HNF TFs (**Figure 4-5**). The binding sites appear to be enriched within the upstream 1 kb from the transcriptional start site.

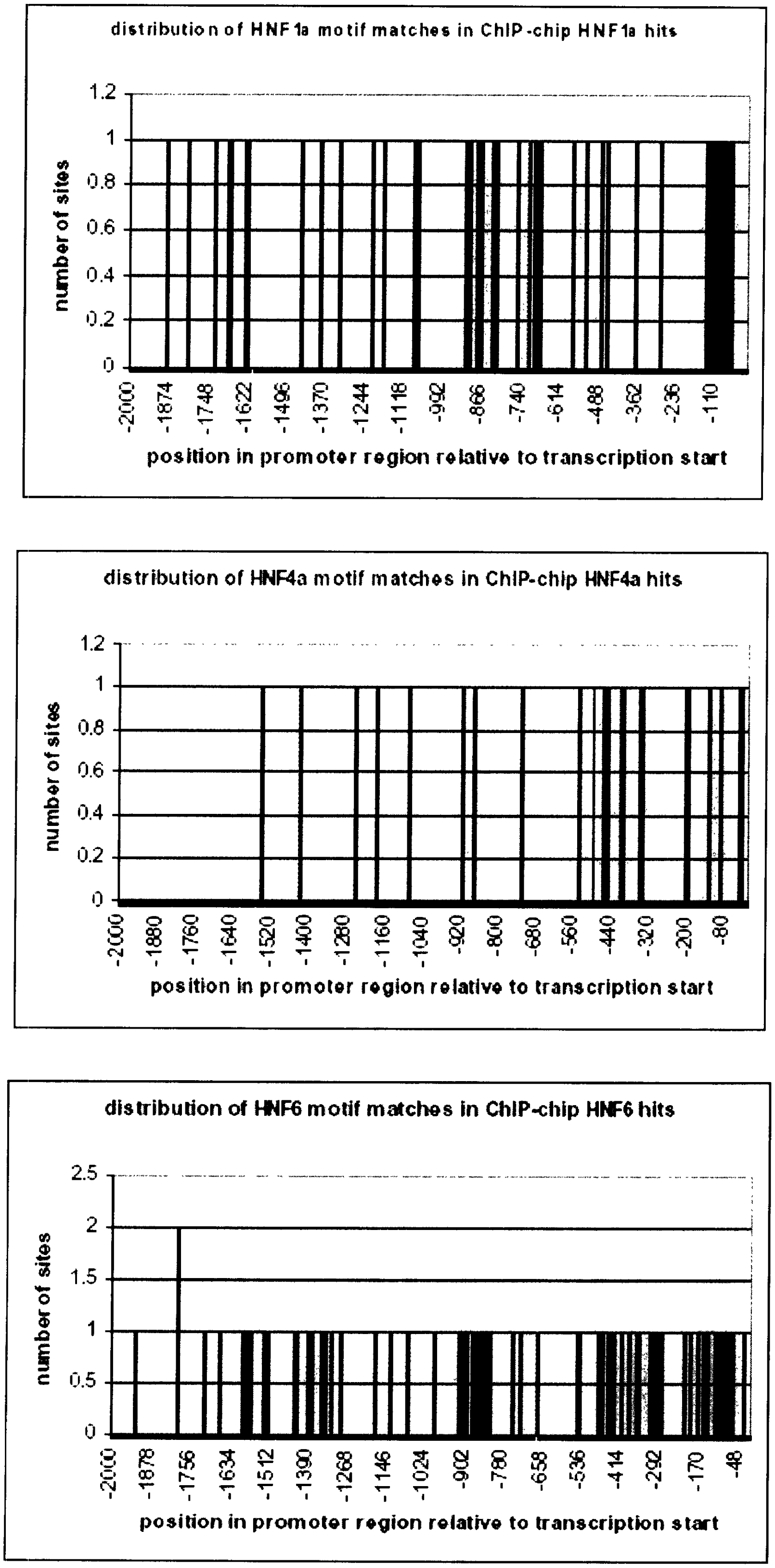


Figure 4-5: Distribution of HNF1α /4α /6 TFBS motif matches among ChIP-chip hits

To evaluate whether this offset information can influence the prediction of HNF regulatory regions, we encoded this information as an offset likelihood ratio:

Let *offset_bound* be a vector of size 2000 that denotes an offset distribution of the site with highest affinity score among the regions bound by an HNF TF *in vivo*. For example, *offset_bond*[*i*] = *k* means that there are *k* promoter sequences that were bound by a given HNF TF at position *i*, with the site having the highest affinity score. Likewise, *offset_nbond* is a vector of size 2000 that denotes an offset distribution of the site with highest affinity score among the regions not bound by an HNF TF *in vivo*. To overcome the problem that most elements in these vectors have zero values due to under sampling, we used a smoothing function to approximate the offset distribution using a sliding window method. Specifically, *s_offset_bond* and *s_offset_nbond* denote the "smoothed" offset distribution. Then

$$s\_offset\_bond[i] = \frac{\sum_{j=i-50}^{j=i+50} offset\_bond[j]}{100}$$

$$s\_offset\_nbond[i] = \frac{\sum_{j=i-50}^{j=i+50} offset\_nbond[j]}{100} \quad \text{if } 50 \le i \le 1950 \qquad (1)$$

$$s\_offset\_bond[i] = \frac{\sum_{j=0}^{j=i+50} offset\_bond[j]}{i+50}$$

$$s\_offset\_nbond[i] = \frac{\sum_{j=0}^{j=i+50} offset\_nbond[j]}{i+50} \quad \text{if } i < 50 \qquad (2)$$

$$s\_offset\_bond[i] = \frac{\sum_{j=i-50}^{1999} offset\_bond[j]}{2050-i}$$

$$s\_offset\_nbond[i] = \frac{\sum_{j=i-50}^{1999} offset\_nbond[j]}{2050-i} \quad \text{if } i > 1950 \tag{3}$$

where the offset likelihood ratio was calculated as:

$$w[i] = \frac{s\_offset\_bond[i] / \sum_{i=0}^{1999} s\_offset\_bond[i]}{s\_offset\_nbond[i] / \sum_{i=0}^{1999} s\_offset\_nbond[i]} \tag{4}$$

Here, we used a sliding window of size 100 bp arbitrarily. When the position *i* is between 50 and 1950, the window size is 100. When the position *i* is at the left edge (*i*<50), the window size corresponds to the length between the left most position (i.e., 0) and 50 bp to the right of position *i* (i.e., *i*+50); when the position *i* is at the right edge (*i*>1950), the window size corresponds to the length between 50 bp to the left of position *i* (i.e., *i*-50) and the right most position (i.e., 1999) (**Figure 4-6**). Therefore, the denominator is *i*+50 when *i*<50 in equation (2) and 2050-*i* when *i*>1950 in equation (3).

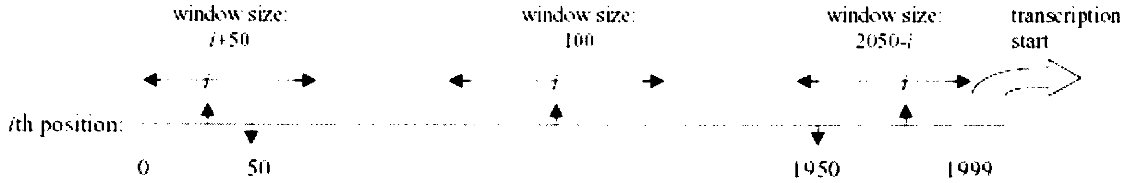


Figure 4-6: Window size calculation for the offset variable in the upstream 2 kb of transcription start site.

The final predictor is the TFs interaction which was calculated as the best binding affinity score of the other two HNFs.

These six predictors for HNF1, HNF4 and HNF6 are summarized in **table 4-2**:

Table 4-2: Predictors in the LR and NN models

Predictors	parameters for HNF1α	parameters for HNF4α	parameters for HNF6
Homotypic Clustering	Occ	Occ	Occ
Evolutionary Conservation	Con	Con	Con
Binding affinity	Score	Score	Score
Binding site offset	Offset	Offset	Offset
TF1 interaction	HNF4Score	HNF1Score	HNF1Score
TF2 interaction	HNF6Score	HNF6Score	HNF4Score

4.3.4. Data randomization

One dataset containing 7425 genes (records) was generated for each HNF. Each record consisted of six predictors and a binary outcome variable indicating whether the given gene was regulated by the given HNF. For both the LR and NN model building, the dataset of 7425 cases was randomly split into five non-overlapping sets. To build the five-fold cross-validation, each time one of the five sets (1485 cases) was held out as the test set and the remaining four sets (5940 cases) were used as training set. A t-test was performed on each predictor among the training and test datasets, and random splitting was continued until there was no significant difference (p-value > 0.05) on any of the variables between the training and test datasets.

4.3.5. Variable selection models

R for Windows, version 1.8.0[36] was used for building the LR models. We first built six LR models using the six predictors individually. Based on the result, we used different combinations of the predictors to build five additional models **(Table 4-3 to 4-5)**.

NevProp3 version 3[37] was used for building the neural network models. NevProp3 requires a user-supplied random initialization seed to set the initial weights in the network, and also to randomly split the training set for the "AutoTrain" option. For all neural network models, we built two-layer networks using the same variable selection as used in LR models. The number of inputs and hidden units varied within each model. We used NevProp's "AutoTrain" option with NSplit = 10, which randomly splits the allotted training data into training and holdout sets 10 times, and takes the average of all of the runs to obtain a target error. The final model is then built by training on all training data, using the target error as the stopping criterion to stop the optimization.

Table 4-3: Variable selection models of LR and NN for HNF1 α

Model	Occ	Con	Score	HNF4Score	HNF6Score	Offset
1	x					
2		x				
3			x			
4				X		
5					x	
6						x
7	x	x	x	X	x	x
8			x			x
9			x		x	x
10	x		x		x	x
11	x		x	X	x	x

Table 4-4: Variable selection models of LR and NN for HNF4 α

Model	Occ	Con	Score	HNF1Score	HNF6Score	Offset
1	x					
2		x				
3			x			
4				X		
5					x	
6						X
7	x	x	x	X	x	X
8				X		X
9			x	X		X
10			x	X	x	X
11	x		x	X	x	X

Table 4-5: Variable selection models of LR and NN for HNF6

Model	Occ	Con	Score	HNF1Score	HNF4Score	Offset
1	x					
2		x				
3			x			
4				X		
5					x	
6						x
7	x	x	x	X	x	x
8			x			x
9			x	X		x
10	x		x	X		x
11	x		x	X	x	x

4.4. Results

We used the Receiver Operating Characteristic (ROC) curve and the C-index to evaluate model accuracy. The ROC curve is a commonly used tool for evaluating the performance of prediction models[38,39]. The ROC curve is a plot of sensitivity (i.e., true positive rate) vs. (1-specificity) (i.e., false positive rate) over a range of threshold values. The area under the ROC curve is a standard measure of discrimination, which is one measure of the accuracy of a model. The greater the area under the ROC curve, the greater the ability of the test to discriminate between two cases. The C-index is equivalent to the area under an ROC curve, which can range from 0.5 (no discriminating ability) to 1.0 (perfect discrimination).

The results shown are average C-index values from 5 different randomization splits for each of the variable selection models. The best average C-indices (0.638) for the HNF1 α dataset (**Table 4-6**) were obtained by model 10 (built on binding affinity, offset, HNF6 interaction and homotypic clustering) and model 3 (built on binding affinity alone) from LR and NN, respectively. Since higher C-index indicates better discrimination ability of the model, model 10 is the best LR model and model 3 is the best NN model for HNF1 dataset. From the average C-index for each individual predictor, we can see that binding affinity (Score) alone had almost the same discrimination ability as the best models. ModuleFinder had a C-index of 0.549 in predicting genes bound by HNF1 α . (**Figure 4-7**).

Table 4-7 shows the LR and NN results from the HNF4 α dataset. The best LR model that obtained by model 9 (built on binding affinity, HNF1 α interaction and offset) had C-index 0.5454 and the best NN model that obtained by model 10 (built on binding affinity, HNF1 α interaction, HNF6 interaction and offset) had C-index 0.545. ModuleFinder had a C-index of 0.5006. **Figure 4-8** shows ROC curves of the best LG, NN models and ModuleFiner.

Both model 9 (built on binding affinity, offset and HNF1 α interaction) of LR and NN gave best C-index for HNF6 dataset (**Table 4-8**). ModuleFinder, in this case, had a C-index of 0.55. ROC curves of best LG and NN model and ModuleFinder are shown in **Figure 4-9**.

Table 4-6: LR and NN C-indices for HNF1 α dataset

Model	LR C-index	NN C-index
1: Occ	0.5470	0.5479
2: Con	0.5200	0.5204
3: Score	0.6380	0.6383
4: HNF4Score	0.5230	0.5062
5: HNF6Score	0.5618	0.5609
6: Offset	0.5890	0.5895
7: All	0.6366	0.6259
8: Score + Offset	0.6379	0.6263
9: 8+ HNF6Score	0.6358	0.6326
10: 9 + Occ	0.6386	0.6268
11: 10 + HNF4Score	0.6371	0.6219

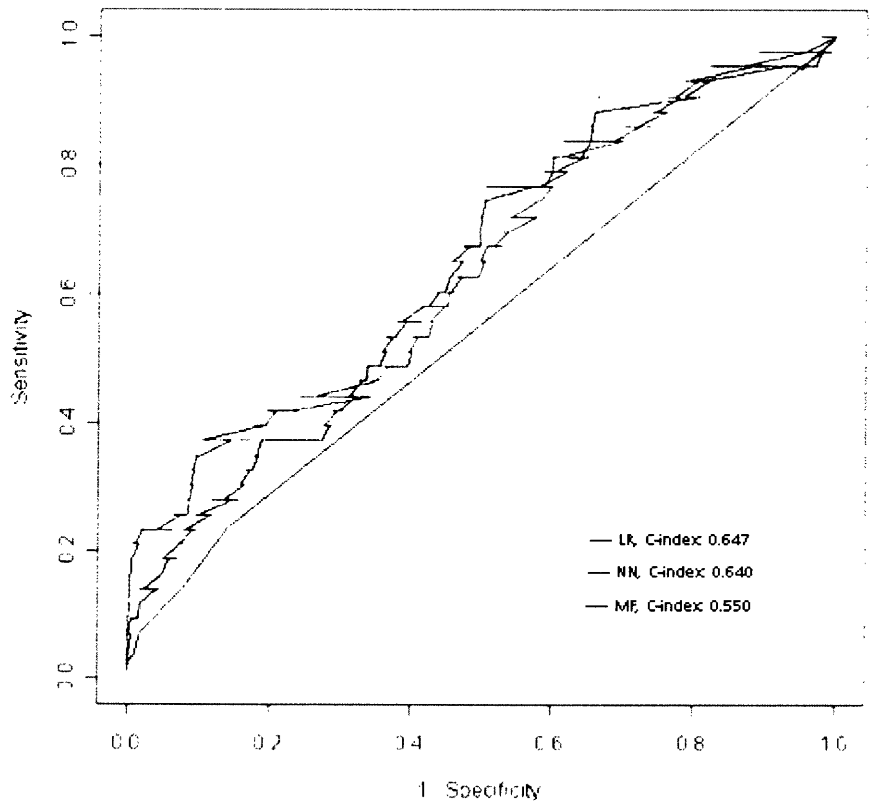


Figure 4-7: ROC curves of LR, NN and MF for HNF1 α

Table 4-7: LR and NN C-indices for HNF4α dataset

Model	LR C-index	NN C-index
1: Occ	0.5029	0.5014
2: Con	0.5009	0.5006
3: Score	0.5146	0.5120
4: HNF1Score	0.5152	0.5040
5: HNF6Score	0.5046	0.5026
6: Offset	0.5403	0.5403
7: All	0.5436	0.5435
8: Offset+ HNF1Score	0.5421	0.5420
9: 8+ Score	0.5454	0.5449
10: 9 + HNF6Score	0.5452	0.5450
11: 10 + Occ	0.5440	0.5437

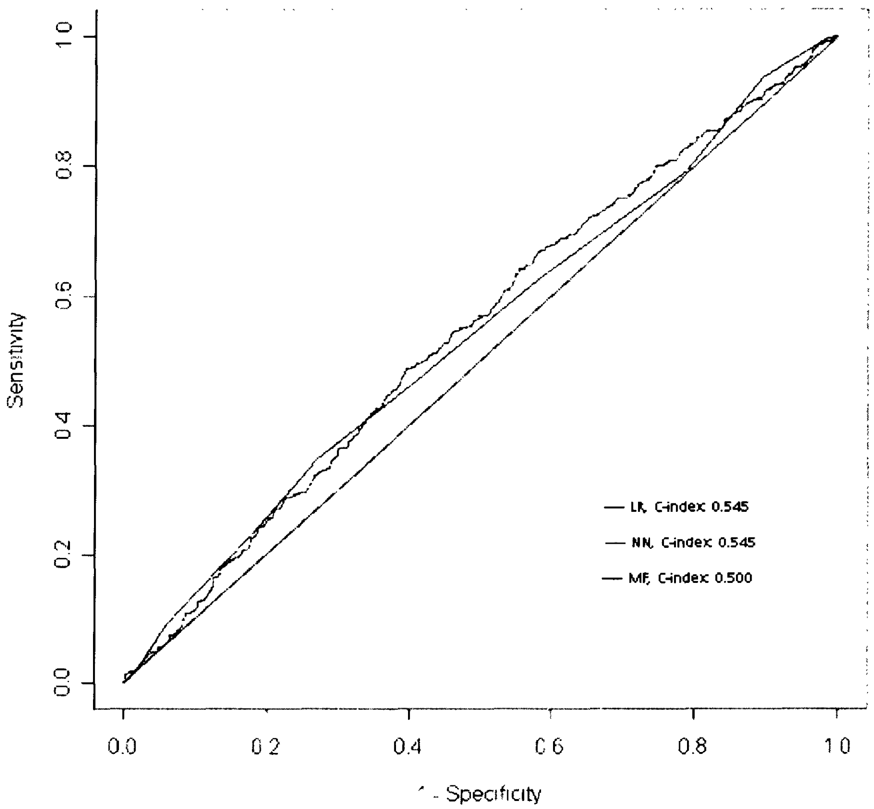


Figure 4-8: ROC curves of LR, NN and MF for HNF4α

Table 4-8: LR and NN C-indices for HNF6 dataset

Model	LR C-index	NN C-index
1: Occ	0.5491	0.5491
2: Con	0.5130	0.5130
3: Score	0.6079	0.6079
4: HNF1Score	0.5675	0.5676
5: HNF4Score	0.5209	0.5216
6: Offset	0.5979	0.5979
7: All	0.6340	0.6296
8: Score + Offset	0.6270	0.6291
9: 8+ HNF1Score	0.6360	0.6386
10: 9 + Occ	0.6358	0.6346
11: 10 + HNF4Score	0.6359	0.6368

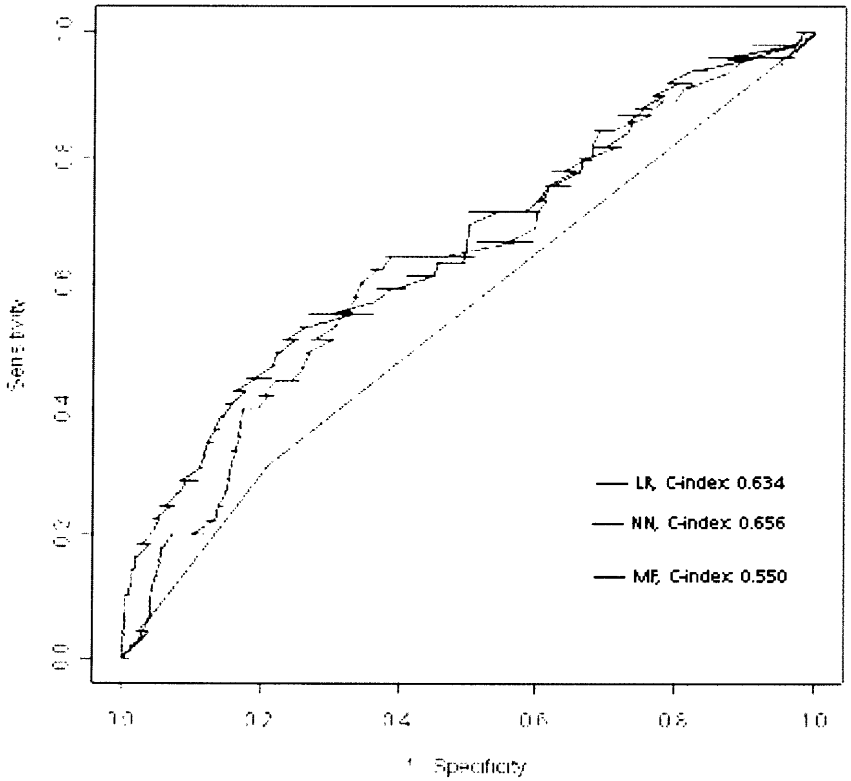


Figure 4-9: ROC curves of LR, NN and MF for HNF6

4.5. Discussion

All three approaches had very poor performances on HNF1/4/6 datasets. This is surprising since these regions should in theory have experimentally determined binding sites for these three transcription factors. It is also known that TFs can regulate transcription via two types of DNA binding mechanisms, involving direct binding and indirect binding via cross-talk with other TFs. In direct binding, a TF binds directly to promoter sequence of the regulated gene. In contrast, in indirect binding, a TF may cooperate with an unknown factor (or factors) which in turn binds to the promoter region of the gene being regulated. In the ChIP-chip experiment, the formaldehyde cross-links both proteins with each other as well as to DNA and the experiment reports both direct and indirect binding of proteins to DNA. Therefore, the positive ChIP-chip HNF promoter regions do not necessarily contain binding sites of the given HNF.

The C-indices of all models including ModuleFinder and all the variable selection models of logistic regression and neural networks that were built on the three TRANSFAC TFBS data for HNF1α/4α/6, ranged from 0.50 to 0.64. Among these methods, ModuleFinder had the worst performance.

ModuleFinder works by scoring homotypic clustering, heterotypic clustering and binding site conservation. In these ChIP-chip datasets, these three indicators seem less important than other variables such as binding site affinity and location, which ModuleFinder currently does not consider. In the variable selection models of LR and NN, the models built on the homotypic clustering variable (Occ) alone had C-indices that indicated the models had no discriminative ability. We then examined the ChIP-chip data for HNF1α/4α/6, and found that only a small percentage of the ChIP-chip HNF positive promoter regions ("hits") had matches for the TRANSFAC HNF1α/4α/6 binding sites (**Table 4-9 to Table 4-11**). Specifically, among 209 promoter regions bound by HNF1α, 49 (23.4%) had exact matches to the TRANSFAC HNF1α sites. Only 1.2% (22 out of 1820) of the HNF4α ChIP-chip hits contain exact matches to the TRANSFAC HNF4α sites, and 31.2% (72/231) had such exact HNF6 matches in the HNF6 ChIP-chip hits.

Table 4-9: HNF1α site matches in HNF1α ChIP-chip positive regions

	Regions bound by ChIP-chip	Regions not bound by ChIP-chip	
Site matches	49	1021	1070
No site matches	160	6195	
	209		

Table 4-10: HNF4α site matches in HNF4α ChIP-chip positive regions

	Regions bound by ChIP-chip	Regions not bound by ChIP-chip	
Site matches	22	62	84
No site matches	1788	5553	
	1810		

Table 4-11: HNF6 site matches in HNF6 ChIP-chip positive regions

	Regions bound by ChIP-chip	Regions not bound by ChIP-chip	
Site matches	72	1530	1602
No site matches	159	5664	
	231		

Furthermore, we found that only ~10% of binding sites in the 2 kb promoter regions were conserved in the mouse genome. This finding is consistent with the fact that adding conservation (Con) variable to the LR and NN variable selection models does not improve the model performance.

The low number of occurrences and cross-species conservations of TRANSFAC HNF1 α /4 α /6 sites in promoter region may be partly due to the fact that the TRANSFAC binding sites we used are not a complete representation of these TFs' DNA binding site specificity. Indeed, TRANSFAC is continually updated as more experimental data become available. The protein-binding microarray[40,41] (PBM) technology could be applied to determine the *in vitro* binding specificities of HNF1 α /4 α /6 by assaying the sequence-specific binding of these individual transcription factors directly to double-stranded DNA microarrays spotted with a large number of potential DNA-binding sites.

Studies of muscle transcription factors reveal that the interactions between Mef-2, Myf, SRF, TEF and Sp1 are needed for muscle transcription regulation[11]. Clustering of functional sites was found in muscle regulatory sequences. We did literature searches on HNF1 α /4 α /6 regulation, but did not find evidences of binding sites clustering in the regulatory regions of these TFs.

Since none of these known CRM sequence features that were integrated into ModuleFinder seem to be present in the HNF1 α /4 α /6 ChIP-chip binding data, it is not surprising that ModuleFinder does not appear to perform well in this dataset.

The best variable selection LR and NN models obtained better C-indices than that of ModuleFinder. This is partially based on the fact that LR and NN use the training data to "learn" the relationship between the outcome and a set of observed variables and predict the outcome of new data based on the existing data, while ModuleFinder does not require training data as they are often unavailable. Moreover, LR and NN incorporated both binding affinity and binding site offset information, which turned out to be the two most informative predictors in the model structure. Binding affinity is a continuous measurement for the probability of the TF binding site. Incorporating this information could enhance the model performance especially when the binding site is more degenerate[27,35]. The result of this analysis inspired us to integrate the binding affinity information into our newer version of ModuleFinder.

5. Conclusions and Future Directions

We have presented a statistically rigorous approach for scoring windows of genomic sequence according to their likelihood of containing BSs for a collection of input TFs. The approach systematically integrates homotypic clustering, heterotypic clustering and evolutionary conservation across multiple genomes into a single, objective scoring scheme that does not require training. Additionally, our algorithm, implemented as a *C* program called "ModuleFinder," is publicly available for download, along with pre-processed genomes and alignments for yeast, worm, fly, mouse, rat, and human, at our lab website (http://the_brain.bwh.harvard.edu/PSB2005MFSuppl/index.html).

We have tested ModuleFinder on a set of human skeletal muscle CRMs using a variety of genome alignments and TFBSs, and have achieved a maximum sensitivity and specificity of 96% and 94%. On this dataset, improved sensitivity and specificity were achieved by using mouse and rat alignments in the searches, whereas chicken alignments actually decreased sensitivity and specificity. Furthermore, PWMs resulted in improved sensitivity and specificity over exact TFBS matches.

We also have tested ModuleFinder on HNFs (HNF1 α , HNF4 α and HNF6) transcription factors binding data from Odom *et al.*, and compared its performance with logistic regression and neural network variable selection models. The poor performance of all three models may due to some unknown binding specificities of these transcription factors. The results of this study also show that the TFs' binding affinity seems to be a potential additional indicator that can be integrated into ModuleFinder to identify functional TFBSs.

The current version of ModuleFinder considers up to two alignment genomes as input, and we are currently expanding it to accept arbitrarily many genomes. We expect that in the future we and others will use ModuleFinder to further refine transcriptional regulatory models for CRMs in particular biological systems and thus discover how the associated TFBSs are organized to confer specific gene expression patterns.

References

1. Bulyk, M. L. Computational prediction of transcription-factor binding site locations. *Genome Biol* **5**, 201 (2003).
2. Boffelli, D. et al. Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science* **299**, 1391-4 (2003).
3. Cliften, P. et al. Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science* **301**, 71-6 (2003).
4. Kellis, M., Patterson, N., Endrizzi, M., Birren, B. & Lander, E. S. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature* **423**, 241-54 (2003).
5. McGuire, A. M., Hughes, J. D. & Church, G. M. Conservation of DNA regulatory motifs and discovery of new motifs in microbial genomes. *Genome Res* **10**, 744-57 (2000).
6. Thomas, J. W. et al. Comparative analyses of multi-species sequences from targeted genomic regions. *Nature* **424**, 788-93 (2003).
7. Berman, B. P. et al. Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proc Natl Acad Sci U S A* **99**, 757-62 (2002).
8. Halfon, M. S., Grad, Y., Church, G. M. & Michelson, A. M. Computation-based discovery of related transcriptional regulatory modules and motifs using an experimentally validated combinatorial model. *Genome Res* **12**, 1019-28 (2002).
9. Markstein, M., Markstein, P., Markstein, V. & Levine, M. S. Genome-wide analysis of clustered Dorsal binding sites identifies putative target genes in the Drosophila embryo. *Proc Natl Acad Sci U S A* **99**, 763-8 (2002).
10. Krivan, W. & Wasserman, W. W. A predictive model for regulatory sequences directing liver-specific transcription. *Genome Res* **11**, 1559-66 (2001).
11. Wasserman, W. W. & Fickett, J. W. Identification of regulatory regions which confer muscle-specific gene expression. *J Mol Biol* **278**, 167-81 (1998).
12. Frith, M. C., Hansen, U. & Weng, Z. Detection of cis-element clusters in higher eukaryotic DNA. *Bioinformatics* **17**, 878-89 (2001).
13. Frith, M. C., Spouge, J. L., Hansen, U. & Weng, Z. Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res* **30**, 3214-24 (2002).
14. Rajewsky, N., Vergassola, M., Gaul, U. & Siggia, E. D. Computational detection of genomic cis-regulatory modules applied to body patterning in the early Drosophila embryo. *BMC Bioinformatics* **3**, 30 (2002).
15. Rebeiz, M., Reeves, N. L. & Posakony, J. W. SCORE: a computational approach to the identification of cis-regulatory modules and target genes in whole-genome sequence data. Site clustering over random expectation. *Proc Natl Acad Sci U S A* **99**, 9888-93 (2002).
16. Sinha, S., van Nimwegen, E. & Siggia, E. D. A probabilistic method to detect regulatory modules. *Bioinformatics* **19 Suppl 1**, i292-301 (2003).
17. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J Mol Biol* **215**, 403-10 (1990).

18. Wasserman, W. W., Palumbo, M., Thompson, W., Fickett, J. W. & Lawrence, C. E. Human-mouse genome comparisons to locate regulatory sites. *Nat Genet* **26**, 225-8 (2000).
19. Odom, D. T. et al. Control of pancreas and liver gene expression by HNF transcription factors. *Science* **303**, 1378-81 (2004).
20. Irving, R. W. & Love, L. Suffix binary search trees and suffix arrays. *University of Glasgow, Computing Science Department Research Report, TR-2001-82* (2001).
21. Irving, R. W. & Love, L. The suffix binary search tree and suffix AVL tree. *Journal of Discrete Algorithms* **1**, 387-408 (2003).
22. Blanchette, M., Schwikowski, B. & Tompa, M. Algorithms for phylogenetic footprinting. *J Comput Biol* **9**, 211-23 (2002).
23. Moses, A. M., Chiang, D. Y. & Eisen, M. B. Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. *Pac Symp Biocomput*, 324-35 (2004).
24. Prakash, A., Blanchette, M., Sinha, S. & Tompa, M. Motif discovery in heterogeneous sequence data. *Pac Symp Biocomput*, 348-59 (2004).
25. Reinert, G., Schbath, S. & Waterman, M. S. Probabilistic and statistical properties of words: an overview. *J Comput Biol* **7**, 1-46 (2000).
26. http://www.genome.uscs.edu.
27. Stormo, G. D. DNA binding sites: representation and discovery. *Bioinformatics* **16**, 16-23 (2000).
28. Andres, V., Cervera, M. & Mahdavi, V. Determination of the consensus binding site for MEF2 expressed in muscle and brain reveals tissue-specific sequence constraints. *J Biol Chem* **270**, 23246-9 (1995).
29. www.biobase.de.
30. Johansson, O., Alkema, W., Wasserman, W. W. & Lagergren, J. Identification of functional clusters of transcription factor binding motifs in genome sequences: the MSCAN algorithm. *Bioinformatics* **19 Suppl 1**, i169-76 (2003).
31. Ren, B. et al. Genome-wide location and function of DNA binding proteins. *Science* **290**, 2306-9 (2000).
32. Crooks, G. E., Hon, G., Chandonia, J. M. & Brenner, S. E. WebLogo: a sequence logo generator. *Genome Res* **14**, 1188-90 (2004).
33. http://weblogo.berkeley.edu/.
34. Schneider, T. D., Stormo, G. D., Gold, L. & Ehrenfeucht, A. Information content of binding sites on nucleotide sequences. *J Mol Biol* **188**, 415-31 (1986).
35. Lifanov, A. P., Makeev, V. J., Nazina, A. G. & Papatsenko, D. A. Homotypic regulatory clusters in Drosophila. *Genome Res* **13**, 579-88 (2003).
36. httP://www.r-project.org/.
37. http://brain.unr.edu/FILES_PHP/show_papers.php.
38. Zweig, M. H. & Campbell, G. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clin Chem* **39**, 561-77 (1993).
39. Hanley, J. A. & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29-36 (1982).

40. Bulyk, M. L., Huang, X., Choo, Y. & Church, G. M. Exploring the DNA-binding specificities of zinc fingers with DNA microarrays. *Proc Natl Acad Sci U S A* **98**, 7158-63 (2001).

41. Mukherjee, S. et al. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. *Nat Genet* **36**, 1331-9 (2004).