BIOINFORMATICS

Vol. 25 ISMB 2009, pages i321–i329
doi:10.1093/bioinformatics/btp230

DISCOVER: a feature-based discriminative method for motif search in complex genomes

Wenjie Fu†, Pradipta Ray† and Eric P. Xing*

School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

ABSTRACT

Motivation: Identifying transcription factor binding sites (TFBSs) encoding complex regulatory signals in metazoan genomes remains a challenging problem in computational genomics. Due to degeneracy of nucleotide content among binding site instances or motifs, and intricate 'grammatical organization' of motifs within *cis*-regulatory modules (CRMs), extant pattern matching-based *in silico* motif search methods often suffer from impractically high false positive rates, especially in the context of analyzing large genomic datasets, and noisy position weight matrices which characterize binding sites. Here, we try to address this problem by using a framework to maximally utilize the information content of the genomic DNA in the region of query, taking cues from values of various biologically meaningful genetic and epigenetic factors in the query region such as clade-specific evolutionary parameters, presence/absence of nearby coding regions, etc. We present a new method for TFBS prediction in metazoan genomes that utilizes both the CRM architecture of sequences and a variety of features of individual motifs. Our proposed approach is based on a discriminative probabilistic model known as conditional random fields that explicitly optimizes the predictive probability of motif presence in large sequences, based on the joint effect of all such features.

Results: This model overcomes weaknesses in earlier methods based on less effective statistical formalisms that are sensitive to spurious signals in the data. We evaluate our method on both simulated CRMs and real *Drosophila* sequences in comparison with a wide spectrum of existing models, and outperform the state of the art by 22% in F1 score.

Availability and Implementation: The code is publicly available at http://www.sailing.cs.cmu.edu/discover.html.

Contact: epxing@cs.cmu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Deciphering the gene control circuitry encoded in the genome is a fundamental problem in developmental biology (Michelson, 2002). In multi-cellular eukaryotic organisms such as the metazoans, the time- and tissue-specific expression of essential genes during various developmental and physiological processes is carried out by an intricate interplay between the transcriptional factors (TFs), and their regulatory mechanisms which control the binding of the factors to recognition sites, known as TF binding sites (TFBSs), or motifs, within the regions of the DNA sequence called gene regulatory regions (Davidson, 2001). Motifs often appear as recurring, degenerate short string patterns (noisy copies of each other) in the non-coding, regulatory regions of the genome. It has been shown that in higher eukaryotes, instances of TFBS of each TF usually occurs clustered in several small regions of the genome (usually 200–2000 bp) known as *cis*-regulatory modules (CRMs) near the coding region of the gene being regulated. Each CRM typically contains more than one type of TFBS for implementing the logic required to regulate the gene correctly throughout the life of the organism (Davidson, 2001).

Due to the degeneracy of the nucleotide content among motif instances, pattern matching-based *in silico* motif search in higher eukaryotes remains a difficult problem, even when using formalisms such as the position weight matrix (PWM) (or nucleotide distributions at each position of the motif).

The 'grammatical organization' of motifs within CRMs that encode complex spatio-temporal regulatory information can further complicate motif search compared with similar tasks in simpler organisms such as yeast (Frith *et al.*, 2002). Extant methods based on simple pattern matching scores often yield a large number of false positives (FPs) (Sandve and Drablos, 2006), especially when the sequence to be examined spans a long region (e.g. tens of thousands of basepairs) beyond the basal promoters, where possible enhancers and CRMs could be located.

In this article, we concern ourselves with searching for instances of motifs and CRMs in higher eukaryotic genome based on not only a given description of the motif sequence patterns, such as the PWMs, but also additional features that distinguish a putative motif from the background. Our proposed approach is based on a discriminative probabilistic model known as *conditional random field* (CRF) that explicitly optimizes the predictive probability of motif presence in a large background, rather than the joint probability of both motif and background sequence under a generative model, as in many of the current methods reviewed below, whose predictive power can be seriously compromised when the amount of background sequence significantly dominates that of the motifs. See Figure 1 for a schematic workflow.

Numerous efforts have been made to predict CRMs comprising of a cluster of TFBSs (Berman *et al.*, 2002), or to use cluster-based analyses to assist TFBS prediction. Some methods directly count the number of matches of some minimal strength to given motif patterns within a certain window of DNA sequence (Donaldson *et al.*, 2005; Rajewsky *et al.*, 2002; Rebeiz *et al.*, 2002; Sharan *et al.*, 2003). From a modeling point of view, this family of algorithms assumes that motifs are uniformly and independently distributed within a fixed size window. Such methods are conceptually straightforward and often simple to implement and computationally efficient. In practice, setting the optimal window size can be difficult and optimal

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

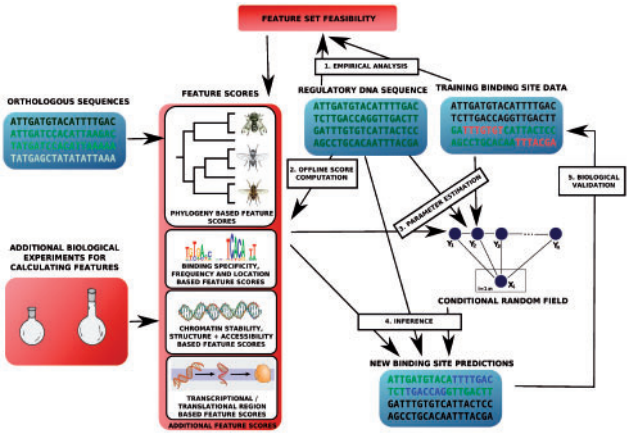

Fig. 1. A schematic view of the workflow.

parameters may not be robust on input data and may require careful analysis to calculate (Lin *et al.*, 2008). Further, an i.i.d. distribution of motifs is now known to be an unrealistic assumption (Bulyk *et al.*, 2002).

A second major class of methods adopt a generative formalism to model the occurrences of motifs and CRMs as the output of some hidden stochastic processes, such as a first-order hidden Markov model (HMM), which removes the necessity of modeling the window size. The hidden-state transition matrix within the HMM usually corresponds to a set of soft constraints on the expected CRM length and the inter-CRM distance in terms of geometric distributions. HMMs that capture motif distributions, as well as intra-CRM and inter-CRM backgrounds, have been used in several prediction algorithms, e.g. Cister (Frith *et al.*, 2002), and Cluster-Buster (Frith *et al.*, 2003). Further extensions have been made to include distinct motif-to-motif transition probabilities in programs such as Stubb (Sinha *et al.*, 2006), Module Sampler (Thompson *et al.*, 2004) and BayCis (Lin *et al.*, 2008), which also employs generalized, hierarchical HMMs. These extended models often require a significant amount of training data. Moreover, logical rules have recently been applied in a model on yeast data (Noto and Craven, 2007) in order to try and capture regulatory logic models in the spirit of Davidson (Davidson, 2001). While the HMMs and HMM-like models are capable of describing the architecture and properties of CRMs to a certain degree, the expressive power of HMMs is insufficient in that they cannot support complex representations for motifs such as non-local, sequence–composition based or epigenetic features surrounding the motif. As a result, their performances on complex CRMs such as those of the *Drosophila* early developmental genes are still unsatisfactory.

Phylogenetic conservation has been historically one of the most commonly used features used besides binding specificity to detect TFBSs (Loots *et al.*, 2002; Moses *et al.*, 2004). However, these algorithms are restricted to very closely related organisms [no more than 50 million years to the most recent common ancestor (Ray *et al.*, 2008)], because non-coding sequences are difficult to align across large evolutionary distances due to commonplace evolutionary forces like duplication and shuffling in the regulatory genome, hence making orthology prediction difficult (Davidson, 2001). Several comparative genomic methods have been applied to CRM and motif prediction (Ray *et al.*, 2008; Siddharthan *et al.*, 2004; Sinha and He, 2007; Sinha *et al.*, 2004). In this article, we concern ourselves only with motif detection within a single species, but we try and use additional features which use phylogenetic data from other species to analyze the effect of multi-species data on motif discovery.

A key motif representation used in all the above methods to score possible motif occurrence in an input DNA sequence is the PWM (Staden, 1984), also known as position-specific scoring matrix (PSSM) (see Supplementary Material for details). Several motif detection algorithms work based on designing hard constraints on features associated with motifs, like distance to transcription start site (TSS) (Sinha *et al.*, 2008). Recently, there has been a number of works in the literature that focus on refining predictive models for individual TFBS by using a wide range of features that have been shown to correlate well with regulatory regions in general and with TFBSs in particular, without necessarily modeling the CRM structure (Narlikar *et al.*, 2007; Naughton *et al.*, 2006; Pudimat *et al.*, 2004; Sharon and Segal, 2007). Using biologically motivated features like presence or absence of CpG islands, nucleosome sites, and helical structures, they appear to be able to significantly outperform models based on PWM motif representation alone. Pudimat *et al.* (2004) models a variety of features to assist in predicting binding sites, but selects the set of features in a greedy fashion, and models the features as nodes of a generic graphical model, causing topology selection of the graphical model to be NP-hard (Pearl, 1988). Sharon and Segal (2007) uses Markov networks to associate specific features with subsets of TFBS positions, causing the difficult problem of estimating the network structure to arise. Ernst (2008) analyzes a set of features to derive informative priors for TFBS prediction, using logistic regression-based classifiers for the choice of each feature. Such discriminative, integrative models have also achieved some success on other problems like protein fold recognition (Damoulas and Girolami, 2008).

In this article, we present DISCOVER : DIScriminative COnditional random field for motif recoVERy in metazoan genomes. DISCOVER is a discriminative method for motif detection in higher eukaryotic genomes that enjoys the dual advantage of modeling CRM architecture of sequences and features of individual motifs. It is a CRF model (Lafferty *et al.*, 2001), which incorporates a wide range of both CRM structure-based and individual motif-based features. CRFs have previously been used in sequence analysis, most notably in gene prediction (DeCaprio *et al.*, 2007; Gros *et al.*, 2007), since coding regions are much better characterized in terms of sequence level features with respect to regulatory regions. Bockhurst and Craven (2005) has applied a similar scheme to identify regulatory signals in prokaryotic sequences; but their model employs a simple feature set to resolve the motif sequence overlap problem, and also requires a pre-screening of motif scores via basic PWM-based models.

Our method is important in several respects in the context of the literature. First, it is a discriminative model explicitly tailored towards maximizing the conditional likelihood of predicting motifs, rather than maximizing the joint likelihood—which often confounds the analysis in the case of generative models. Secondly, it employs a comprehensive set of features carefully selected from the literature designed to capture a variety of characteristics of the motif and CRM patterns. Thirdly, it is an integrative model that allows sequence-specific features to be added at will to enhance the prediction scheme. Further, since feature scores are computed offline, it is easier

to incorporate scores involving complicated computation and long computation times as well as long-range dependencies.

We evaluate the CRF model on both simulated CRMs and actual biologically validated transcription regulatory sequences of *Drosophila melanogaster*, in comparison with a wide spectrum of existing models including, Cister (Frith *et al.*, 2002), Cluster-Buster (Frith *et al.*, 2003), BayCis (Lin *et al.*, 2008), MSCAN (Alkema *et al.*, 2004), Ahab (Rajewsky *et al.*, 2002) and Stubb (Sinha *et al.*, 2006). The results suggest that our proposed method significantly outperforms others on real *Drosophila* sequences.

The remainder of the article is outlined as follows: we discuss the model and feature design in Section 2. In Section 2.1, we describe how to learn the model from data. and then we briefly mention the inference algorithm given the model. Biological and empirical justifications for the features, experimental setup and results are presented in Section 3. We finish by some discussion on the scope of the model in Section 4.

2 METHODS

The conventional PWM representation for TFBSs is not discriminative enough to distinguish true binding sites from false binding sites. We desire a model for TFBSs and genomic sequence that supports a more complex motif representation without losing the ability to characterize sequence wide properties, which means a flexible feature design. The CRF model—a feature-based log-linear model in which features are easily incorporated—is an appropriate model choice under the circumstances. The basic inputs to such a computational model is a set of genetic sequences, a set of feature values corresponding to every nucleotide in the sequences and the PWMs of TFs that are being predicted. The output of the model is a prediction of a set of TFBSs which are being predicted, ranked in order of decreasing likelihood. The CRM boundaries can also be similarly predicted, but in this article we focus on the analysis of the TFBS predictions.

A CRF model that describes a conditional probability distribution of a genomic sequence is defined as:

$$P(\mathbf{y}\,|\,\mathbf{x},\boldsymbol{\lambda}) = \frac{1}{Z}\exp\left\{\boldsymbol{\lambda}\cdot\mathbf{F}(\mathbf{y},\mathbf{x})\right\} \tag{1}$$

$$\text{where} \qquad Z = \sum_{\mathbf{y}}\exp\left\{\boldsymbol{\lambda}\cdot\mathbf{F}(\mathbf{y},\mathbf{x})\right\} \tag{2}$$

where we use x_i to represent the type of the observed nucleotide at site i in a sequence, and y_i to represent the hidden state associated with x_i, which corresponds to the functionality of the site in the genomic sequence. The value of a hidden state is also called a state label. Vector $\mathbf{x}=\{x_i : i = 1,2,\ldots,L\}$, and vector $\mathbf{y}=\{y_i : i=1,2,\ldots,L\}$, where L is the length of the sequence. Vector $\mathbf{F}$ is the set of features, each element F of which is the sum of feature scores of a particular feature category (where feature scores refer to the numerical value of the feature). Vector $\boldsymbol{\lambda}$ corresponds to the feature weights assigned to the set of features, and is learnt from data to decide which features may be more important in predicting TFBSs. Z is a partition function that normalizes the pdf and is a function of $\mathbf{x}$ and $\boldsymbol{\lambda}$. The value space for each x_i is {A,C,G,T}. The values represent the four types of nucleotide in DNA, *adenine*, *cytosine*, *guanine* and *thymine*, respectively. The value space for hidden states y_i, however, is not so straightforward, and it will be defined subsequently.

State design: we design a set of hidden states based on the possible functionality of each nucleotide in the genomic sequence being analyzed.We incorporate each motif type as a state since this is our prediction goal. We number the types of motifs and name the state for the m-th motif type $\mathbf{M}^{(m)}$. Representationwise, a hidden state y_i being state $\mathrm{M}^{(m)}$ implies that a motif of the m-th type is located starting at site i of the sequence. Those states are all that we need to represent binding sites. Next, we know that TFs are usually working together to regulate genes, especially in genomes of higher organisms. In order to work together, different types of TFBSs often lie close to each other in the range of hundreds of base pairs forming a so-called CRM (Davidson, 2001). We use state **C** to represent all nucleotides in the CRM regions except those binding sites which have already been labeled as Ms. The nucleotides which are still unlabeled after the first two rounds are set to state **G**, which represents a global background in the genomic sequence. Hence, the set of hidden states for modeling the functionality at a nucleotide position is given by $\mathbf{S}=\{\mathbf{G},\mathbf{C},\mathbf{M}^{(1)},\ldots,\mathbf{M}^{(N_M)}\}$, where N_M is the number of motif types. We do not allow two motifs to share the same starting position, but such occurrences are infrequent. It is still an improvement on HMM-based approaches where modeling even partial overlap of motifs causes a combinatorial increase in the state space. Overlapping of starting positions of TFBSs can be accommodated in our model by using marginal probabilities in the prediction step.

Feature design: each element $F(\mathbf{y},\mathbf{x})$ of vector $\mathbf{F}(\mathbf{y},\mathbf{x})$ in Equation (1) is the sum of feature scores of a particular feature category, where feature score simply refers to the numerical value of the feature. It sums up feature function f's over the sequence, which have a common meaning and share the same weight. An example is shown in Equation (16) of Supplementary Material, after we see some concrete features. The design of f's is a critical part of CRF models. We include a rich set of features, most of which are introduced in Section 3. The set of features includes conventional features (TFBS sequence specificity, state transition probability) as well as evolutionary features (like presence of repeats, and of conservation across species), structural and epigenetic features (like melting temperature, nucleosome occupancy), features related to the protein coding mechanism (like distance to TSS, presence in 3′-UTR region), and additional discriminative features (like reverse complementarity of a site, and conservation symmetry). Their formal definitions can be found in Supplementary Material.

Features with a one-to-one correspondence with nucleotide base pairs can be easily integrated into the framework by defining as:

$$f(y_i,\mathbf{x}) = \left(\sum_m \delta(y_i,\mathrm{M}^{(m)})\right) S(i,\mathbf{x}) \tag{3}$$

where $S(i,\mathbf{x})$ is the feature score, All features are in the form of $f(\mathbf{y},\mathbf{x})$, but as for now, they have a simpler common form of $f(y_i,y_{i+1},\mathbf{x})$, which we called a chain structure CRF model.

Model Parameters: feature weights constitute the set of model parameters, some of which are fixed and some are free to be estimated. More free parameters make the CRF model more complex, which might be harder to learn. The set of free parameters are modeled to avoid redundant parameters, which will not make any contribution. Also, parameters that are not likely to be properly estimated from training data should never be included, because including them will only increase the chance of overfitting the model. Our focus is on the weight of state transition features, because they account for a large proportion of the whole parameter set and good estimation of the weights are critical for successfully predicting TFBSs. A detailed analysis is presented in Supplementary Material.

In the CRF model, we assign a parameter as a weight to each of the features defined previously which are collectively the vector $\boldsymbol{\lambda}$ in Equation (1). Not all of these parameters are free parameters. Among state transition parameters, we constrain an M state to be only directly reachable from a C state, and not from a G state, since motifs are not present outside CRMs. Thus, state transition features corresponding to taboo transitions have a weight $-\infty$ (a low enough number in practice), meaning that the transitions never occur in the CRF model. However, we want to have a reasonable number of free model parameters as more free parameters increase the expressibility of the model. With increase in the number of free parameters, the hardness of estimating model parameters increase, the running time of the learning algorithm also rises and some parameters may overfit due to data scarcity for corresponding features.

2.1 Model training and inference

In this section, we briefly describe the model training and inference procedures in which feature weights of the CRF model are learnt from training data and subsequently used to make TFBS predictions. A more thorough exposition is presented in Supplementary Material.

Model training: First, a learning criterion is set up, which can either be to maximize likelihood or maximize posterior probability. It is then converted to a convex optimization problem, and finally a Quasi–Newton method is applied (Avriel, 2003). Our goal here is to learn the best setting for $\boldsymbol{\lambda}$, the weights of features in the CRF model given a set of sequences as training data with their nucleotide types $\mathbf{x}$ and state labels $\mathbf{y}$. The value of feature functions $\mathbf{f}$ can be computed given necessary hyper-parameters. A reasonable criteria to learn the feature weights $\boldsymbol{\lambda}$ from nucleotide types $\mathbf{x}$ and state labels $\mathbf{y}$ (or more precisely from feature values $\mathbf{f}$) in a CRF model is to maximize likelihood of $\boldsymbol{\lambda}$ wrt $\mathbf{y}$ conditioned on $\mathbf{x}$, which equals the probability of state labels $\mathbf{y}$ given feature weights $\boldsymbol{\lambda}$ conditioned on nucleotide types $\mathbf{x}$, because the probability model itself is defined in this conditional scheme. The max likelihood estimator of $\boldsymbol{\lambda}$ can be expressed as:

$$\widehat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x})$$

$$\text{where} \qquad L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) = P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\lambda})$$

Inference: the learnt feature weights of the CRF model are used to predict TFBSs on a new genomic sequence—the inference step. There are two categories of prediction schemes analogous to the popular inference schemes for HMMs: sequence decoding by *Viterbi* algorithm and marginal decoding by forward–backward algorithm. We choose the marginal probability rank scheme as it enables us to predict overlapping TFBSs. Marginal decoding considers one hidden state at a time, making predictions based on the marginal probability, $P(y_i \mid \mathbf{x}, \boldsymbol{\lambda})$, which can be computed by the dynamic programming forward–backward algorithm in a chain structure CRF model (Lafferty *et al.*, 2001; Sha and Pereira, 2003). Variants on the marginal decoding scheme include maximum a posteriori decoding (MAP) where we predict a TFBS if the marginal probability of it is the highest among all state labels

$$\widehat{y_i} = \arg\max_{y_i} P(y_i \mid \mathbf{x}, \boldsymbol{\lambda}) \tag{4}$$

Alternatively, we make a positive prediction whenever the marginal probability is above a threshold, known as threshold decoding. It is a flexible method, but a good threshold is hard to set in practice. We use a similar scheme that takes advantage of thresholding by choosing a threshold automatically by limiting the number of predictions. Thus we calculate a list of TFBS and marginal probability pairs, sort them by probability in descending order and output the top P ones as predictions, P being the number of desired predictions. We make P for each sequence proportional to its length L, as a longer sequence tends to contain more TFBSs. The coefficient $k = P/L$ is called *prediction factor*. We call this rank decoding.

3 RESULTS

We evaluate our method of TFBS prediction on a set of real genomic transcription regulatory sequences (TRSs) of *D.melanogaster*, as well as a set of synthetic TRSs. The prediction performance is compared with six popular published methods for supervised discovery of motifs/CRMs based on a wide spectrum of models: Cister (Frith *et al.*, 2002), Cluster-Buster (Frith *et al.*, 2003), BayCis (Lin *et al.*, 2008), Stubb (Sinha *et al.*, 2006), Ahab (Rajewsky *et al.*, 2002) and MSCAN (Johansson *et al.*, 2003). In general, the prediction performance of the CRF model is superior or competitive wrt all the chosen benchmark methods on this comprehensive selection of real *D.melanogaster* dataset.

The semi-synthetic dataset was generated by artificially simulated CRM structures with a third-order Markov model for background sequences and planting real TFBSs from the TRANSFAC database (Wingender *et al.*, 2000) into the simulated background sequences based on the generative model for the HMM-based TFBS prediction tool Baycis and published in Lin *et al.* (2008). It involves 30 20 kbp-long sequences, containing 887 TFBSs of 10 types. The real *D.melanogaster* binding site data were obtained from the *Drosophila* Cis-regulatory Database at National University of Singapore (Narang *et al.*, 2006). The PWM and CRM boundary data were obtained independently of the binding site database from the REDfly CRM database (Gallo *et al.*, 2006). This TRS dataset was previously published in Lin *et al.* (2008). The dataset contains 97 CRMs pertaining to 35 early developmental genes of *D.melanogaster* (in 35 sequences). Each of the 35 sequences contains 1–4 CRMs. The lengths of sequences range from 10 000 bp to 16 000 bp, except two extremely long sequences whose lengths are 40 kb and 79 kb, respectively. There are 700 TFBSs of 44 types labeled in the dataset in all. It is worthwhile noticing that 12 out of the 44 types appear in only one sequence, which account for 10% of the binding sites. A visualization of the dataset illustrating the locations of TFBSs and CRMs is presented in Figure 2.

3.1 Input features

We include a rich set of features in our model, based on previous findings in the literature as well as some derived features which empirical evidence suggests are more discriminative than the original features from which they were derived. Most of the feature scores are accurately or heuristically calculated based solely on the sequence data, but some require external annotation (like translated and transcribed regions, and TSS). It is also easy to change feature values from sequence-derived heuristic values to actual experimental results should they become available. See the work schematic (Fig. 1) for a visual schema of feature calculation. CRFs adjust feature weights based on training data, so it is also interesting to try new features to check if they improve the predictive power of the model. The rigorous mathematical definitions corresponding to the non-trivial feature definitions is presented in Supplementary Material. Binding site positioning and characterization of the nucleotide content of binding sites in terms of binding site specificity have been the most standard features which have been used in motif finding, especially in generative models like HMMs. This is based on sound biological validation of the fact that specificity of binding sites and CRM 'architecture's, are pervasive in regulatory regions (Davidson, 2001).

PWM constraints: the basic feature we use is the PWM constraint, which implements the information present in the PWM of a motif. It represents the binding specificities of the DNA binding domain(s) of the TF in question as an ordered set of multinomials, and is an indicator of the level of evolutionary constraint and hence selection each nucleotide is under. Some PWMs tend to be more constrained (under greater purifying selection) than others. Some PWMs also tend to suffer from noisy data. Because of this, the discriminative power of the PWM constraints feature varies from PWM to PWM. For PWMs with poor discriminative power, additional features are critical for improving predictability. The PWM score provides a good baseline measure for the CRF model in motif prediction, though it is not an essential feature in our model.

State transition: state transition features are an effort to model the architecture of the regulatory region. The state transition feature

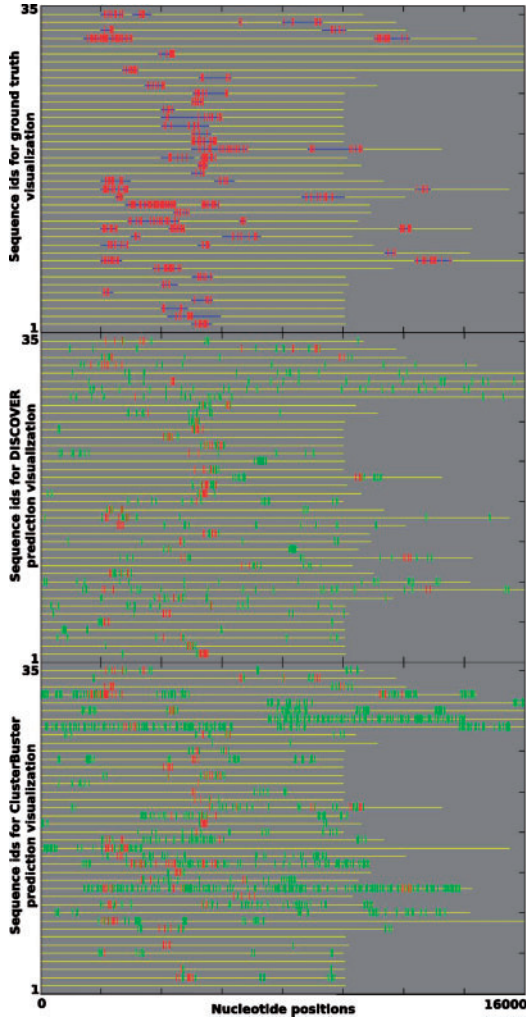

Fig. 2. Aligned data and prediction visualizations with CRMs in blue, ground truth and true positive (TP) TFBSs in red and false positive (FP) TFBSs in green. Very long sequences are broken in two for ease of depiction.

models the relationship between the functionality of neighboring nucleotides, which correspond to neighboring states in the CRF and is based on the differing likelihoods of the hidden CRF states transitioning from one to the other. Details of the mathematical modeling of this feature is provided in Supplementary Material.

Evolutionary conservation and presence or absence of evolutionary events like duplication and repeats can also play a role in identifying TFBS, as evidenced by the large body of work in phylogenetic motif finding. The basic premise in such cases is that functionally relevant nucleotides like TFBS would be under selection, and would hence be distinguishable from surrounding sequence on the basis of evolutionary parameters. While we do not explicitly use multiple species sequence data, we implicitly use evolutionary data in terms of feature data.

Presence of repeats: Interspersed repeats and low complexity DNA sequences are common elements in the genome, often near coding regions and inside regulatory sequences. The repeat feature is a simple single nucleotide-based feature indicative of whether that nucleotide is part of a repeat as predicted by RepeatMasker using the repeat database RepBase (Jurka *et al.*, 2005). On one hand, repeats with motif-like patterns may lead to a large number of FP results, but repeats have also been reported to have been under purifying selection (Britten, 1994) and to have been harnessed into the regulatory machinery (Kamal *et al.*, 2006). Thus, instead of masking out repeats to lower the FP rate, we choose to identify repeats in the sequence in a bid to find locational correlations with TFBSs.

PhastCons score and related features: We use the PhastCons score as an evolutionary score-based feature. PhastCons (Margulies *et al.*, 2003) is a phylogenetic 2-state HMM which predicts if nucleotide positions in a multiple alignment are in an evolutionarily conserved state or not. The PhastCons score at a nucleotide position is merely the posterior probability that the nucleotide was generated from the conserved state based on the 15-way Multiz (Blanchette *et al.*, 2004) alignment of the *Drosophilae* species, *Apis mellifera*, *Anopheles gambiae* and *Tribolium castaneum*. We also use two other derived binary features which we feel to be discriminative based on an empirical analysis of PhastCons score distributions (Fig. 3): 'Is PhastCons score <0.05' and 'Is PhastCons score >0.95'. We also keep an additional feature indicating whether PhastCons data are available or not for bookkeeping purposes.

It is well established in the literature that the distance of the TFBS to the TSS plays an important role of the efficacy of the TFBS in regulating the gene (Defrance and Touzet, 2006; Kim *et al.*, 2008; Tharakaraman *et al.*, 2005), and of the nature of function of the TFBS (Elnitski *et al.*, 2006). We therefore incorporate several features which contain information of the distance to the TSS, the locations of the transcribed and translated regions, and the positioning of binding site with respect to the gene transcription–translational direction.

Distance to TSS and translated: TFBS are typically present near coding sequences, and we utilize two features indicative of that fact. The binary feature 'Translated' indicates at each nucleotide position whether it is translated or not by the gene translation/transcription machinery. It has also been shown that TFBSs are not uniformly distributed wrt their distance from the TSS (Defrance and Touzet, 2006), and the Distance to TSS feature is a score of the distance of each nucleotide from the TSS in question.

5′-UTR and 3′-UTR: The position of the TFBS wrt directionality of the gene being coded has been shown to be a discriminative feature for identifying TFBS. We use two binary features indicative of this fact, the '5′UTR' feature indicates for each nucleotide if it is located in the 5′ untranslated region, and the '3′UTR' feature indicates likewise for the 3′ untranslated region.

Recent work in the literature has approached the TFBS prediction problem as a non-binary classification problem, instead choosing to model the affinity of a TF to bind to a particular oligonucleotide sequence with an affinity score (Ward and Bussemaker, 2008). This has led to the realization that TFBSs may also be effective gene regulators in cases of low binding affinity but high chromatin stability and accessibility (Ozsolak *et al.*, 2007). While we model our TFBS prediction as a sort of classification problem, we still incorporate the notions of chromatin accessibility and stability.

GC content and melting temperature: The GC content feature of a genomic sequence or the fraction of G+C bases in a sequence is a simple heuristic which can be used to estimate several factors reflective of the stability of the chromatin structure like the melting temperature and in higher eukaryotes is a determining factor for

identifying CpG islands (Zhang, 2007), thus being indicative of how easy it might be for a TF to actually bind in the locality. The window size w for the genomic neighborhood over which to estimate the GC content is a hyperparameter that must be determined ahead of time, and is usually chosen to be of the order of magnitude of the binding site. The melting temperature feature is defined as the temperature for which half the DNA strands of an oligonucleotide are in the double helical structure, while the other half are in a random coil formation. It corresponds strongly to chromatin stability, and has been shown as a feature to correlate well with TFBS (Ponomarenko *et al.*, 1999).

Nucleosome occupancy: Recent research has suggested that nucleosome occupancy has a strong correlation with binding preference of TFs (Segal *et al.*, 2006). This is due to the non-feasibility of access to the chromatin by the TF when a nucleosome is already bound there. Some research has successfully used nucleosome occupancy scores to improve TFBS predictions (Narlikar *et al.*, 2007).

We also tried several other features directly computable from sequence information, and found that the following features can help in discriminating between TFBS and non-TFBS. The cause of the discriminative power of these tracks may stem from the nature of the binding specificities of the TFs in question, and a closer investigation is warranted.

Reverse complementarity and conservation symmetry: We also try two additional features for the CRF based on symmetry of the oligonucleotide in question. The reverse complementarity feature indicates as a fraction between 0 and 1 how similar a nucleotide sequence is to its reverse complement. It is exactly 1 only when an oligonucleotide sequence is identical to its reverse complement. The conservation symmetry feature models how symmetric the degree of conservation in the PWM is wrt the center of the binding site. This is based on the empirical observation that DNA binding domain binding specificities often have symmetric sequence conservation profiles.

The design of new features has exciting new possibilities. Long-range regulatory effects have been reported in the literature (Carroll *et al.*, 2005). The CRF model also readily enables us to model long-range dependencies if we deviate from the chain structured CRF structure. It can also be used as a form of ensemble learning by incorporating predictions by other independent tools as features. Other features which have been shown in the literature to correlate well with the data and which are candidates for future inclusion on this and other datasets include the presence of the nucleotide in the first intron of the regulated gene, and presence of the nucleotide in the neighborhood of a CpG island.

We tested the discriminative nature of these features on the dataset in Figure 3. Figure 3a shows the difference in mean values for background, CRM and motif nucleotides for two of the most discriminative features: GC content and PhastCons score. Figure 3b shows the distribution of PhastCons scores in motif versus non-motif nucleotides, with the most discriminative bins being at either end of the score range, which offered us some insight as to how to define a derived feature which is more discriminative than the original one. Figure 3c shows the interesting multimodal distribution of the normalized and transformed values of the feature distance to the TSS, suggesting a complicated, non-uniform distribution worth additional investigation.

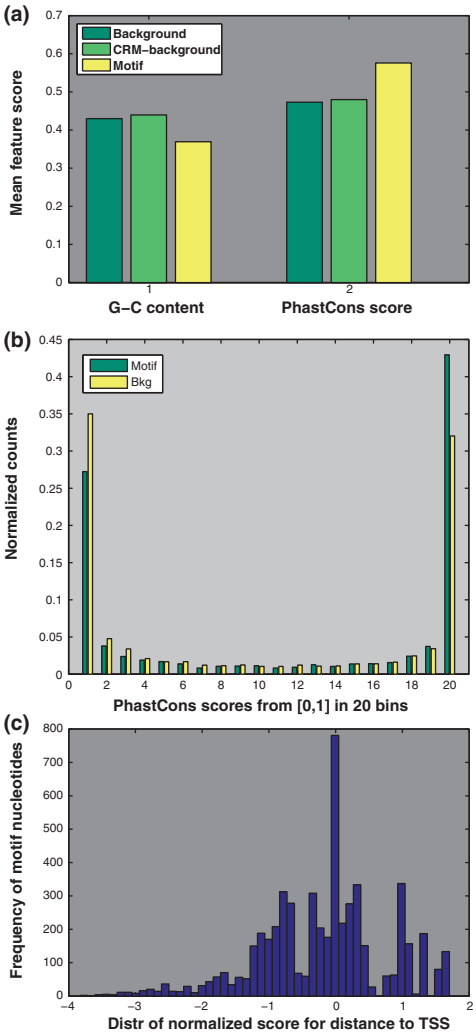

Fig. 3. (**a**) Means of two discriminative features plotted for GC content and PhastCons score for Motifs, CRMs and background nucleotides, (**b**) distribution of PhastCons scores in motifs versus non-motifs and (**c**) multimodal empirical distribution of feature values for the transformed distance to TSS feature.

3.2 Experimental setup

In this part, we include biological and empirical bases for selection of some features, data preparation, hyper-parameter setting, test scheme and evaluation scheme. For training data, we use a part of the sequences with ground truth labels. For testing, the required hyper-parameters in the CRF model are the window size used in GC percentage calculation and pseudo-counts used to smooth the probabilities in PWMs to allow for greater tolerance in motif discovery. We set the window size of GC percentage to 8 bps (approximately the average length of a motif) and pseudo-count for smoothing PWM probabilities to 0.5.

Our evaluation is based on a leave-one-out cross-validation (LOOCV) scheme. Each time we take all but one sequences as training data, and predict on the remaining sequence by the model with parameters learnt from the training data. We use the rank decoding scheme with the prediction factor k set to 0.0015 by

default. This threshold is obtained by analyzing the empirical density of TFBSs in training data. Varying the value of the threshold results in increasing one of the performance metrics of precision (P) or recall (R) at the cost of the other. For evaluating performance, we use the standard definitions of P, R and the F1 score using counts of TP, FP and false negative (FN) prediction instances. The exact method of calculating the evaluation metrics is given in Supplementary Material.

Specificity scores and ROC curves are not shown as these evaluation schemes are inappropriate in the context of motif detection. True negative (TN) instances in ground truth for motif data is rare as instances labeled as negatives in the ground truth may be discovered to contain motifs in the future. Also, the number of positive instances and number of predictions are much smaller than the number of total instances, causing the specificity to be very close to 1 almost always.

3.3 Tests on features

We have empirically established the discriminative nature of our feature set, but we also examine the soundness of the designed features in the context of the CRF model after incorporating some basic features, before including all of them in the model to test for feature redundancy and compatibility in the CRF framework. The state transition features and sequence conservation features are fundamental, so we check the validity of the other features based on predictions made by a basic model consisting of only state transition features and sequence conservation features. The soundness of additional feature is shown by comparing the distributions of the set of TPs and the set of FPs as predicted by the basic model.

We learn a CRF model using the two kinds of fundamental features, and use it to get a set of predictions of TFBSs, which contains both TP predictions and FP predictions. We split the predictions into two groups, TP group and FP group, and compute the GC percentage score, reverse complementary score and conservation symmetry score for each of the instances in the two groups. We can show the soundness of a feature by a statistical analysis on the difference between scores of the two groups. There are 193 instances in TP group and 499 instances in FP group. Comparisons of cumulative distribution function (CDF) curves between TP group and FP group on GC percentage scores, reverse complementary scores and conservation symmetry scores are shown in Figure 4. The scores plotted are raw scores without an offset, such as p, s and cs in Equations (9), (11) and (13) of Supplementary Material. We can see that the CDF curve of TP group is almost always lower than that of FP group in GC percentage score and reverse complementary score, while the CDF curve of TP group is almost always higher than that of FP group in conservation symmetry score.

For the feature of GC percentage, the scores in TP group have a mean at 0.4641 and sample variance at 0.0043, and the scores in FP group have a mean at 0.4323 and sample variance at 0.0065. Assuming that they both follow Gaussian distributions, we have a difference between means at 0.0318 with a SD at 0.0059, which gives us a confidence value at $1–4\times10^{-8}$ that the mean of TP group is bigger than the mean of FP group. It is credible that GC percentage feature is informative. Following a similar analysis, for the feature of reverse complementarity, the mean TP score is 0.3041

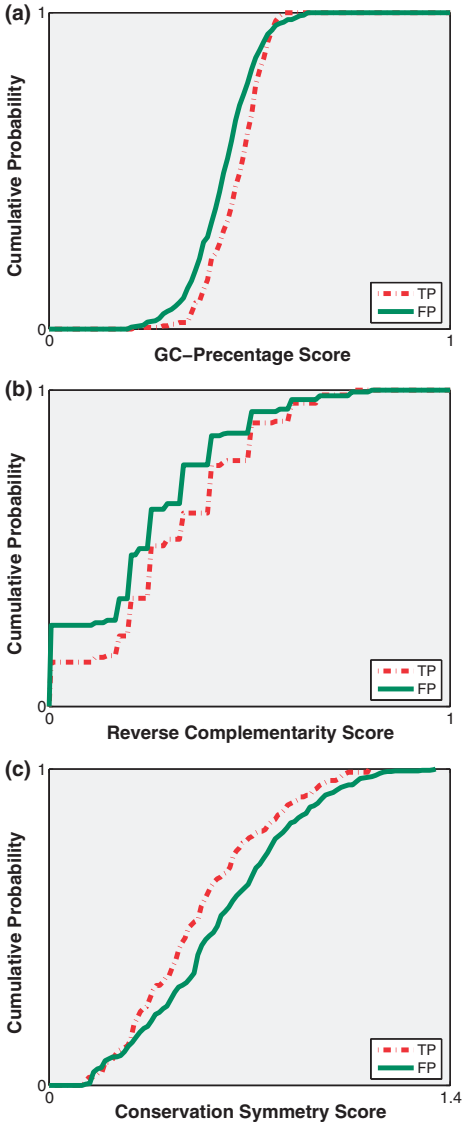

Fig. 4. On (**a**) GC percentage score, (**b**) reverse complementary score and (**c**) conservation symmetry score, a comparison of CDF curves between TP group and FP group.

and sample variance 0.0349, and the mean FP score is 0.2413 and sample variance 0.0360. With a difference between means at 0.0159 with a SD at 0.0059, we have a confidence value at $1–4\times10^{-5}$ that the mean of TP group is bigger than the mean of FP group. For the feature of conservation symmetry, the TP scores have mean 0.5215 and sample variance 0.0541, and the FP scores have a mean 0.5950 and sample variance 0.0666. The confidence value that TP group has a smaller average score than FP group is $1–1.5\times10^{-4}$.

3.4 Performances on TFBS prediction

Synthetic dataset: We compare the CRF model with BayCis, ClusterBuster and Cister on the synthetic TRS dataset. CRF model outperforms ClusterBuster and Cister but not BayCis (Fig. 5a) on the synthetic dataset. BayCis has an advantage over the other tools

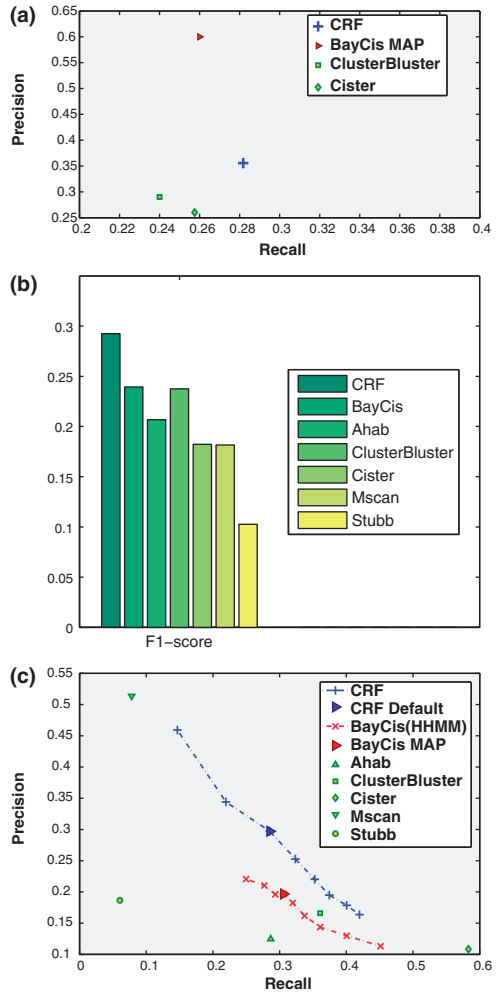

Fig. 5. (**a**) P–R performance of CRF, BayCis, Cluster-Buster and Cister on the synthetic dataset, (**b**) F1 score and (**c**) P–R curve of the CRF model in comparison with other algorithms at their default settings on the real *D. melanogaster* TRS dataset.

having the same background model as the simulation scheme, but we outperform Baycis on the real dataset.

***Drosophila* dataset:** We compare the CRF model with BayCis, Ahab, Cluster-Buster, Cister, Mscan and Stubb on the real *D. melanogaster* TRS dataset. The overall F1 scores of the CRF model and six comparing methods are shown in Figure 5. All the algorithms are set to default configurations. The feature-based CRF model outperforms all other methods on the F1 score measure. It is 22% higher than the best competing tool. We also show the P–R curves of the our methods and BayCis, as well as points in the P–R landscape for other tools in Figure 5. We plot P-R curves of the CRF model by varying the prediction factor k (from 0.0005 to 0.0040). For BayCis, we plot a P-R curve resulting from different thresholds for predictions, in addition to its default MAP setting. The CRF model outperforms BayCis, Ahab, ClusterBuster and Stubb in their default settings. The other two methods strike extremely different balances between P and R in their default output. MSCAN focuses on very high P predictions, while Cister is geared towards high values of R. It is noticeable that Stubb's performance is much below the rest, possibly because it uses distinct motif-to-motif transition probabilities, which can only be properly learned without overfitting from datasets richer in scope than the present one. Addition of further non-redundant features like other epigenetic feature scores is expected to improve performance further. A set of predictions by the CRF model with default setting comparing with that of Cluster-Buster is shown in Figure 2. While they have comparable TP predictions, CRF model makes much less FP predictions than Cluster-Buster does. In a way, the performance gap between the CRF model and the HMM-based models may be looked upon as a combination of two factors: the discriminative nature of the analysis, and the availability of features besides PWM and transition data.

4 DISCUSSION

We propose DISCOVER, a discriminative model using CRFs for motif discovery. Among advantages of the CRF model are the facts that the user can incorporate new features at will (with the model automatically adjusting feature weights to weed out uninformative features) and can configure our publicly available tool to add new genetic and epigenetic features. It can even be used for ensemble learning by incorporating predictions from other models as features. In the future, a Bayesian version of the work can be tried by putting priors on parameters as long as they do not break the concavity of the target function. We will model higher order CRFs by moving beyond chain structure CRFs with only edges between neighboring hidden states to incorporating feature functions with long-range dependencies to handle features like motif co-occurrence, distance models for CRM lengths and inter-motif spacer runs. A detailed discussion on the scope of the model can found in Supplementary Material.

ACKNOWLEDGEMENTS

The authors thank Geir Kjetil Sandve and Veronica Hinman for comments and suggestions.

Funding: National Science Foundation (CAREER Award grant DBI-0546594 to E.P.X.); Alfred P. Sloan Research Fellowship (to E.P.X.).

Conflict of Interest: none declared.

REFERENCES

Alkema,W.B. *et al.* (2004) Mscan: identification of functional clusters of transcription factor binding sites. *Nucleic Acids Res.*, **32**, W195–W198.

Avriel,M. (2003) *Nonlinear Programming: Analysis and Methods*. Dover Publishing, Mineola, NY.

Berman,B.P. *et al.* (2002) Exploiting transcription factor binding site clustering to identify cis-regulatory modules involved in pattern formation in the Drosophila genome. *Proc. Natl Acad. Sci. USA*, **99**, 757–762.

Blanchette,M. *et al.* (2004) Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, **14**, 708–715.

Bockhurst,J. and Craven,M. (2005) Markov networks for detecting overlapping elements in sequence data. *Proc. Adv. Neural Inform. Process. Syst.*, **17**, 193–200.

Boyd,S. and Vandenberghe,L. (2004) *Convex Optimization*. Cambridge University Press, Cambridge.

Britten,R. (1994) Evolutionary selection against change in many Alu repeat sequences interspersed through primate genomes. *Proc. Natl Acad. Sci. USA*, **91**, 5992–5996.

Bulyk,M. *et al.* (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.*, **30**, 1255–1261.

Carroll,J. *et al.* (2005) Chromosome-wide mapping of estrogen receptor binding reveals long-range regulation requiring the forkhead protein FoxA1. *Cell*, **122**, 33–43.

Damoulas,T. and Girolami,M.A. (2008) Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection. *Bioinformatics*, **24**, 1264–1270.

Davidson,E.H. (2001) *Genomic Regulatory Systems*. Academic Press, San Diego, CA.

DeCaprio,D. *et al.* (2007) Conrad: gene prediction using conditional random fields. *Genome Res.*, **17**, 1389–1398.

Defrance,M. and Touzet,H. (2006) Predicting transcription factor binding sites using local over-representation and comparative genomics. *BMC Bioinformatics*, **7**, 396.

Donaldson,I.J. *et al.* (2005) Tfbscluster: a resource for the characterization of transcriptional regulatory networks. *Bioinformatics*, **21**, 3058–3059.

Elnitski,L. *et al.* (2006) Locating mammalian transcription factor binding sites: a survey of computational and experimental techniques. *Genome Res.*, **16**, 1455–1464.

Ernst,J. (2008) *Computational Methods for Analyzing and Modeling Gene Regulation Dynamics*. PhD dissertation, Carnegie Mellon University, MLD.

Frith,M.C. *et al.* (2002) Statistical significance of clusters of motifs represented by position specific scoring matrices in nucleotide sequences. *Nucleic Acids Res.*, **30**, 3214–3224.

Frith,M.C. *et al.* (2003) Cluster-buster: finding dense clusters of motifs in dna sequences. *Nucleic Acids Res.*, **31**, 3666–3668.

Gallo,S.M. *et al.* (2006). Redfly: a regulatory element database for drosophila. *Bioinformatics*, **22**, 381–383.

Gros,S. *et al.* (2007) CONTRAST: a discriminative, phylogeny-free approach to multiple informant de novo gene prediction. *Genome Biol.*, **8**, R269.

Johansson,O. *et al.* (2003) Identification of functional clusters of transcription factor binding motifs in genome sequences: the mscan algorithm. *Bioinformatics*, **19 (Suppl. 1)**, i169–i176.

Jurka,J. *et al.* (2005) Repbase Update, a database of eukaryotic repetitive elements. *Cytogenet. Genome Res.*, **110**, 462–467.

Kamal,M. *et al.* (2006) A large family of ancient repeat elements in the human genome is under strong selection. *Proc. Natl Acad. Sci. USA*, **103**, 2740–2745.

Kim,N.K. *et al.* (2008) Finding sequence motifs with Bayesian models incorporating positional information: an application to transcription factor binding sites. *BMC Bioinformatics*, **9**, 262.

Lafferty,J. *et al.* (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*. Williamstown, MA.

Lin,T.-H. *et al.* (2008) Baycis: a bayesian hierarchical hmm for cis-regulatory module decoding in metazoan genomes. In *Proceedings of RECOMB 2008*. Singapore.

Loots,G.G. *et al.* (2002) rVista for comparative sequence-based discovery of functional transcription factor binding sites. *Genome Res.*, **12**, 832–839.

Margulies,E.H. *et al.* (2003) Identification & characterization of multi-species conserved sequences. *Genome Res.*, **13**, 2507–2518.

Michelson,A.M. (2002) Deciphering genetic regulatory codes: a challenge for functional genomics. *Proc. Natl Acad. Sci. USA*, **99**, 546–548.

Moses,A.M. *et al.* (2004) Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. In *Proceedings of Pac. Symp. Biocomput. 2004*, Hawaii, pp. 324–335.

Narang,V. *et al.* (2006) Computational annotation of transcription factor binding sites in D. melanogaster developmental genes. In *Proceedings of The 17th International Conference on Genome Informatics*. Yokohama.

Narlikar,L. *et al.* (2007) A nucleosome-guided map of transcription factor binding sites in yeast. *PLoS Comput. Biol.*, **3**, e215.

Naughton,B. *et al.* (2006) A graph-based motif detection algorithm models complex nucleotide dependencies in transcription factor binding sites. *Nucleic Acids Res.*, **34**, 5730–5739.

Noto,K. and Craven,M. (2007) Learning probabilistic models of cis-regulatory modules that represent logical and spatial aspects. *Bioinformatics*, **23**, e156–e162.

Ozsolak,F. *et al.* (2007) High-throughput mapping of the chromatin structure of human promoters. *Nat. Biotechnol.*, **25**, 244–248.

Pearl,J. (1988) *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

Ponomarenko,J. *et al.* (1999) Conformational and physicochemical DNA features specific for transcription factor binding sites. *Bioinformatics*, **15**, 654–668.

Pudimat,R. *et al.* (2004) Feature based representation and detection of transcription factor binding sites. In *Proceedings of the German Conference on Bioinformatics 2004*, Bielefeld, pp. 43–52.

Rajewsky,N. *et al.* (2002) Computational detection of genomic cis-regulatory modules applied to body patterning in the early drosophila embryo. *BMC bioinformatics*, **3**, 30.

Ray,P. *et al.* (2008) Csmet: comparative genomic motif detection via multi-resolution phylogenetic shadowing. *PLoS Comput. Biol.*, **4**, e1000090.

Rebeiz,M. *et al.* (2002) 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. USA*, **99**, 9888–9893.

Sandve,G.K. and Drablos,F. (2006) A survey of motif discovery methods in an integrated framework. *Biol. Direct*, **1**.

Segal,E. *et al.* (2006) A genomic code for nucleosome positioning. *Nature*, **442**, 772–778.

Sha,F. and Pereira,F. (2003) Shallow parsing with conditional random fields. *Proc. Hum. Lang. Tech.-NAACL*, **1**, 134–141.

Sharan,R. *et al.* (2003) Creme: a framework for identifying cis-regulatory modules in human-mouse conserved segments. *Bioinformatics*, **19** (Suppl. 1), i283–i291.

Sharon,E. and Segal,E. (2007) A feature-based approach to modeling protein-dna interactions. *Lect. Notes Comput. Sci.*, **4453**, 77–91.

Siddharthan,R. *et al.* (2004) Phylogibbs: a gibbs sampler incorporating phylogenetic information. In Eskin,E. and Workman,C. (eds), *Regulatory Genomics*, Vol. 3318 of *Lecture Notes in Computer Science*, Springer, Berlin, pp. 30–41

Sinha,S. and He,X. (2007) MORPH: probabilistic alignment combined with hidden Markov models of cis-regulatory modules. *PLoS Comput. Biol.*, **3**, e216.

Sinha,S. *et al.* (2004) PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. *BMC Bioinformatics*, **5**, 170.

Sinha,S. *et al.* (2006) Stubb: a program for discovery and analysis of cis-regulatory modules. *Nucleic Acids Res.*, **34**, W555–W559.

Sinha,S. *et al.* (2008) Systematic functional characterization of cis-regulatory motifs in human core promoters. *Genome Res.*, **18**, 477–488.

Staden,R. (1984) Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res.*, **12**, 505–519.

Tharakaraman,K. *et al.* (2005) Alignments anchored on genomic landmarks can aid in the identification of regulatory elements. *Bioinformatics*, **21 (Suppl. 1)**, i440–i448.

Thompson,W. *et al.* (2004) Decoding human regulatory circuits. *Genome Res.*, **14**, 1967–1974.

Ward,L. and Bussemaker,H. (2008) Predicting functional transcription factor binding through alignment-free and affinity-based analysis of orthologous promoter sequences. *Bioinformatics*, **24**, i165–i171.

Wingender,E. *et al.* (2000) TRANSFAC: an integrated system for gene expression regulation. *Nucleic Acids Res.*, **28**, 316–319.

Zhang, M. (2007) Computational analyses of eukaryotic promoters. *BMC Bioinformatics*, **8 (Suppl. 6)**, S3.

Supplementary Material for DISCOVER: A feature-based discriminative method for motif search in complex genomes

A1. Formal definitions of some features

Sequence Conservation

The PWM offers a simple and straightforward way to formally model a TFBS specific to a single TF, and to score the DNA segments according to its likelihood of being a true motif or not based only on the sequence content within the segment in question in Naughton *et al* [*Nuc Acids Res*, 2006] takes a slightly different direction and models a motif type as a bag of instances, using a graph cut method to identify motifs in a set of n-grams. We use the standard PWM definition. The feature captures the degree of conservation of a potential motif binding site i given the position weight matrix of the motif, $\theta^{(m)}$.

The feature function is defined as:

$$\begin{aligned} f_{SC}^{(m)}(y_i, \mathbf{x}) &= f_{SC}^{(m)}(y_i, x_{i:i+l^{(m)}-1}) \\ &= \delta(y_i, \mathrm{M}^{(m)}) \sum_{j=1}^{l^{(m)}} \beta(\theta_j^{(m)}, x_{i+j-1}) \end{aligned} \tag{5}$$

$$\beta(\theta_j^{(m)}, k) = \log \theta_{jk}^{(m)} - \log \theta_{0k}; \tag{6}$$

where $\theta^{(m)} = \{\theta_{jk}^{(m)} : j = 1, ..., l^{(m)}, k \in \{\mathrm{A,C,G,T}\}\}$ is the PWM of motif type m, $l^{(m)}$ is the length of the motif, and $\theta_0 = \{\theta_{0k} : k \in \{\mathrm{A,C,G,T}\}\}$ is the nucleotide frequency in background. The δ function equals 1 when y_i is assigned to state $\mathrm{M}^{(m)}$ and 0 otherwise.

State Transition

The State transition feature captures the relationship between neighboring states. The feature function is defined as:

$$f_T^{s_1 \to s_2}(y_i, y_{i+1}, \mathbf{x}) = \delta(y_i, s_1)\delta(y_{i+1}, s_2) \tag{7}$$

where $s_1, s_2 \in \mathbf{S}$. $\delta(y_i, s_1)$ [1] equals 1 when y_i is s_1, and 0 otherwise. $\delta(y_{i+1}, s_2)$ likewise. Thus, the feature function equals 1 only when y_i is state s_1 and y_{i+1} is state s_2, and 0 otherwise. There are $(2+N_M)^2$ features of this category in total. State transition features are an effort to model the architecture of the regulatory region.

GC-Content

A high percentage of nucleotide *guanine* (G) and *cytosine* (C) may indicate a region containing regulatory elements. The feature function is defined as:

$$f_{GC}(y_i, \mathbf{x}) = \delta(y_i, \mathrm{M}) \left(p(x_{i-w/2:i+w/2}) - p_0 \right) \tag{8}$$

$$p(x_{left:right}) = \frac{1}{right - left + 1} \sum_{i=left}^{right} \Big(\delta(x_i, \mathrm{G}) + \delta(x_i, \mathrm{C}) \Big) \tag{9}$$

[1] δ function can be viewed as an Identity function.

where w is the window size, p is the GC-percentage inside the window whose value lies in [0,1], and p_0 is the average GC-percentage over the dataset. The $\delta(y_i, \mathrm{M})$ function equals 1 when y_i is assigned to any motif state and 0 otherwise.

As an example, the sum of conservation symmetry features can be computed as:

$$F_{CS}(\mathbf{y}, \mathbf{x}) = \sum_{i=1}^{L} f_{CS}(y_i, \mathbf{x}) \tag{10}$$

where f_{CS} is defined in Eq 13 and L is the length of the sequence. $F_{CS}(\mathbf{y}, \mathbf{x})$ is one of the elements in function vector $\mathbf{F}(\mathbf{y}, \mathbf{x})$ used in a CRF model in Eq 1.

Reverse Complementarity

This feature assesses how likely a potential binding site i is reverse complementary with itself. In other words, that is how similar the site is to its counterpart on the other genomic strand. The higher similarity may suggest a true motif. The feature function is defined as:

$$f_{RC}(y_i, \mathbf{x}) = \sum_{m} \delta(y_i, \mathrm{M}^{(m)}) \left(s(x_{i:i+l^{(m)}-1}) - s_0 \right) \tag{11}$$

$$s(x_{i:i+l-1}) = \frac{1}{\lfloor l/2 \rfloor} \sum_{j=1}^{\lfloor l/2 \rfloor} \delta_{pair}(x_{i+j-1}, x_{i+l-j}) \tag{12}$$

where s is the reverse complementary score of a potential binding site whose value lies in [0,1], s_0 is an offset value that is set at the mean, and l is the length of the motif. The $\delta(y_i, \mathrm{M}^{(m)})$ function equals 1 when y_i is the state of motif type m and 0 otherwise. The $\delta_{pair}(a, b)$ function equals 1 if and only if a and b are a Watson-Crick pair.

Conservation Symmetry

This feature captures the symmetry of the degree of sequence conservation given motif PWM within a motif binding site with respect to the center. The feature is defined as:

$$f_{CS}(y_i, \mathbf{x}) = \sum_{m} \delta(y_i, \mathrm{M}^{(m)}) \left(cs(\theta^{(m)}, x_{i:i+l^{(m)}-1}) - cs_0 \right) \tag{13}$$

$$\begin{aligned} cs(\theta^{(m)}, x_{i:i+l^{(m)}-1}) = & \frac{1}{\lfloor l^{(m)}/2 \rfloor} \sum_{j=1}^{\lfloor l^{(m)}/2 \rfloor} \Big| \beta(\theta_j^{(m)}, x_{i+j-1}) - \\ & \beta(\theta_{l^{(m)}+1-j}^{(m)}, x_{i+l^{(m)}-j}) \Big| \end{aligned} \tag{14}$$

where cs averages the conservation symmetry score over a potential binding site, cs_0 is an offset value of choice, $l^{(m)}$ is the length of the motif, and β function is the conservation score of a single base defined in Eq 6.

Melting Temperature

This feature provides an estimated melting temperature of sequences within a certain size of window by a formula:

$$f_{MT}(y_{i:i+w-1}, \mathbf{x}) = 64.9 + \frac{41 * (G + C - 16.4)}{A + T + G + C} \tag{15}$$

where w is the window size, and A, T, G and C are the counts of the four types of nucleotides within the window. We set the window size to 15, which is about the length of a long TFBS.

Distance to Transcription Start Site

Sites closer to a transcription start site are more likely to be TFBSs, so we adopt this feature to assess how close each site is to a nearest transcription start site. It is easy to understand that a distance change from 0-bp to 1k-bp makes more difference than a distance change from 10k-bp to 11k-bp though both of them are shifted by 1k-bp, so the feature score should not be linear on distance. We apply a logarithm function and a small constant to avoid logarithm going to negative infinity. The feature scores are calculated as:

$$f(z) = \log(z + 5) \tag{16}$$

where z is the distance in base-pair.

A2. Model Parameters

Feature weights constitute the set of model parameters. Some of them can be fixed and the others are free. More free parameters make the CRF model more complex, which might be harder to learn. As a guide line, we want to avoid redundant free parameters, since they will not make any contribution. On the other hand, parameters that are not likely to be properly learned from training data should never be included, because including them will only increase the chance of over-fitting. In this part, our main focus is on the weight of state transition features, because they account for a large portion in the whole parameter set.

In the CRF model, we assign a parameter as a weight to each of the features defined in the previous subsection. Those are the vector $\boldsymbol{\lambda}$ in Eq 1. However, some of them are not free parameters because of the context. In state transition, it is not allowed to reach an M state directly from a G state, since it is enforced that state M's representing TFBSs are surrounded by state C representing *cis*-Regulatory Module region. Thus, the corresponding state transition features have a weight being *-inf*, which means that the transitions will never happen in the CRF model. In practice, we set the weights to a small enough number.

For the sake of a good performance, we want to have a reasonable number of free model parameters. More free parameters will promote the expressing ability of the model, but at the same time the hardness of model learning will increase, the running time of learning algorithm will rise, and some parameters may be overfitting due to the lack of data describing the related features. In our case, the state transitions from a motif state to a motif state are rare, if they ever happened, which will make those transition features an inevitable overfit if we set them free. Our solution is banning the transition between motif states and setting the matching weights to $-inf$. As a result, the number of all possible state transitions reduces dramatically.

A close look at the remaining set of state transitions will reveal redundancy. Assuming that no CRM region is on the edge, the sequence of hidden states will start with a global background state and end with a global background state. In that case, the number of transition from state G to state C will be exactly the same as the number of transition from state C to state G along the sequence of states. The models are identical to each other as long as the sum of the weight of transition feature G-C and the weight of transition feature C-G is a constant, given all the other parameters unchanged. Only one of the two weights need be a free parameter, leaving the other one to be fixed at any finite value. For simplicity, we set the weight of C-G

to zero. Similar situations happen to the pair of state transition C-M$^{(m)}$ and M$^{(m)}$-C, so we fix the weight of M$^{(m)}$-C at zero.

The free parameters of state transition features left so far are G-G, C-C, G-C and C-M's. The number of state transitions along the sequence is unchanging given the sequence, so there is one more degree of redundancy, a common offset within the weights of state transition features. We get rid of the common offset by fixing the weight of G-G at zero. The final free parameters of state transition features are those of C-C, G-C and C-M's.

For those free parameters, it is not a good idea to let them be totally free. A prior can be imposed on each of them, as a way to encode prior knowledge on them. This may help in the attempt to avoid over-fitting issues. For example, we can make a prior be a normal distribution of mean 0 and variance σ^2.

A3. Model training

In this section, we briefly describe the model training procedure in which feature weights of the CRF model are learned from training data. A more thorough exposition is presented in Supplementary Materials.Firstly, a learning criterion is set up, which can be either to maximize likelihood or to maximize posterior probability. Then, it is turned into a convex optimization problem, and finally a Quasi-Newton method is applied.

Our goal in the model learning task is to learn the best setting for $\boldsymbol{\lambda}$, the weights of features in the CRF model. What we have are a group of sequences as training data with their nucleotide types $\mathbf{x}$ and state labels $\mathbf{y}$, so the value of feature functions $\mathbf{f}$ can be computed given necessary hyper-parameters.

A criterion is needed to learn the feature weights $\boldsymbol{\lambda}$ from nucleotide types $\mathbf{x}$ and state labels $\mathbf{y}$, or more precisely from feature values $\mathbf{f}$. In the CRF model, a reasonable criterion is to maximize the likelihood of $\boldsymbol{\lambda}$ with respect to $\mathbf{y}$ conditioned on $\mathbf{x}$, which equals the probability of state labels $\mathbf{y}$ given feature weights $\boldsymbol{\lambda}$ conditioned on nucleotide types $\mathbf{x}$, because the probability model itself is defined in this conditional scheme. The max likelihood estimator of $\boldsymbol{\lambda}$ can be expressed as:

$$\widehat{\boldsymbol{\lambda}} = \arg\max_{\boldsymbol{\lambda}} L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x})$$

$$\text{where} \qquad L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) = P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\lambda})$$

For the simplicity of notation, we just showed likelihood function in a one-training-sequence circumstance. When multiple (for example, m) training sequences are used, as we do in our experiment, the likelihood function will be:

$$L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) = P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\lambda}) = \prod_{k=1}^{m} P(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \boldsymbol{\lambda})$$

where $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ represent the vector of nucleotide types and a vector of state labels of the k-th sequence, respectively.

Getting the maximum point of a likelihood function is equivalent to getting the maximum point of a log-likelihood function $L_\lambda = \log L(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x})$, since logarithm function is monotonically increase.

$$L_\lambda = \sum_{k=1}^{m} \left[\boldsymbol{\lambda} \cdot \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) - \log Z(\mathbf{x}^{(k)}, \boldsymbol{\lambda}) \right]$$

We can prove that the function of L_λ is concave with respect to $\boldsymbol{\lambda}$ (see supplementary material), so it turns into a typical convex optimization problem to find the maximum point [Boyd and Vandenberghe, 2004]. Gradient method or Newton's method can be adopted, and convergence is assured in theory. Both of them

are iterative methods which first get a search direction and then find a proper step length in each iteration. The update scheme is:

$$\boldsymbol{\lambda}^{(n+1)} = \boldsymbol{\lambda}^{(n)} + t\triangle\boldsymbol{\lambda}$$

where n is the iteration round, $\triangle\boldsymbol{\lambda}$ is the search direction, and t is the step length. The search direction is set to the negative of the first derivative of log-likelihood function $-\triangledown L_\lambda$ in Gradient method, and $-\triangledown L_\lambda / \triangledown^2 L_\lambda$ in Newton's method. The step length is determined by a Back-track Search method (see supplementary material). The initial point $\boldsymbol{\lambda}^{(0)}$ can be picked by experience.

It can be shown that the first derivative of log-likelihood function with respect to $\boldsymbol{\lambda}$ is:

$$\triangledown\boldsymbol{\lambda} = \sum_{k=1}^{m} \left\{ \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) - \mathbb{E}\Big[\mathbf{F}(\mathbf{y}, \mathbf{x}^{(k)}) \,\Big|\, \mathbf{x}^{(k)}, \boldsymbol{\lambda}\Big] \right\}$$

The derivative is tractable, because the conditional expectation of feature sums $\mathbf{F}(\mathbf{y}, \mathbf{x}^{(k)})$ given genomic sequence $\mathbf{x}^{(k)}$ and feature weights $\boldsymbol{\lambda}$ is computational feasible (see supplementary material).

In practice, however, gradient method is likely to converge slowly, and the second derivative term in Newton's method is hard to compute efficiently. A Quasi-Newton method [Avriel, 2003] is more practical, in which an approximation is applied to the inverse of the second derivative of log-likelihood with respect to feature weights $\boldsymbol{\lambda}$ and the rest parts are the same as Newton's method. More specifically we use BFGS approximation method (see supplementary material).

Besides choosing the likelihood of $\boldsymbol{\lambda}$ as the target function to maximize, we can instead use the posterior probability:

$$P(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) = \frac{P(\boldsymbol{\lambda}, \mathbf{y}, \mathbf{x})}{P(\mathbf{y}, \mathbf{x})} = \frac{P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\lambda})\, P(\mathbf{x} \mid \boldsymbol{\lambda})\, P(\boldsymbol{\lambda})}{P(\mathbf{y}, \mathbf{x})}$$

As long as feature weights are independent of genomic sequences, $P(\mathbf{x} \mid \boldsymbol{\lambda}) = P(\mathbf{x})$, which is constant. So,

$$P(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) \propto P(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\lambda})\, P(\boldsymbol{\lambda})$$

The full version of posterior probability for multiple (m) training sequences is:

$$P(\boldsymbol{\lambda} \mid \mathbf{y}, \mathbf{x}) \propto P(\boldsymbol{\lambda}) \prod_{k=1}^{m} P(\mathbf{y}^{(k)} \mid \mathbf{x}^{(k)}, \boldsymbol{\lambda})$$

assuming state labels of different sequences $\mathbf{y}^{(k)}$ are independent of each other given $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ is independent of $\mathbf{x}^{(j)}$ given $\mathbf{x}^{(k)}$ when $j \neq k$.

The new target function is concave, as long as the prior distribution function of $\boldsymbol{\lambda}$ is log-concave. We keep using L_λ to represent the logarithm of posterior probability. As an example, the full version for multiple (m) training sequences is:

$$L_\lambda = \sum_{k=1}^{m} \Big[\boldsymbol{\lambda} \cdot \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) - \log Z(\mathbf{x}^{(k)}, \boldsymbol{\lambda})\Big] - \frac{\boldsymbol{\lambda} \cdot \boldsymbol{\lambda}}{2\sigma^2} + C$$

if each λ follows a $\mathcal{N}(0, \sigma^2)$ as a prior. C is a constant in the equation. The equation has a similar form to a regularized log-likelihood.

A4. Evaluation metrics

We first compare the predictions from the inference step with ground truth labels to obtain counts of true positive (TP), false positive (FP) and false negative (FN) prediction instances. Predictions within a fixed 3bp tolerance window of an actual ground truth instance are taken to be TP. We sum up the TP, FP and FN counts over all sequences and calculate the overall precision and recall values from the overall TP, FP and FN counts using the definitions Precision=TP/(TP+FP) and Recall=TP/(TP+FN). Finally, F1-score = 2/(1/Precision+1/Recall). An alternative method is calculating the precision and recall values on individual sequences first and averaging them second, but this gives equivalent weights to each sequence and unequal importance to each TFBS, and hence we do not pursue this latter method.

A5. Scope of the Model

Our approach takes advantage of the CRF models, which can overcome label bias problems that often happen in HMM models. The CRF model is a discriminative method that is based on a set of feature designs. The flexible forms of feature designs make it possible and easier to encode current knowledge in the field as well as to incorporate new information on TFBS when they are available. For example, we have made use of the knowledge about *cis*-Regulatory Module architecture as well as the abundance level of *guanine* and *cytosine* in nearby region in our predictor for TFBS. A feature weight, a parameter in the CRF model, determines the degree to which the feature influences the probability model. Priors can be put on the parameters, as long as they do not break the concavity of the target function. The concavity (or convexity) is such a good characteristic that we no longer need to worry about the annoying local maximum (or minimum) issues in iterative methods, and convergence is guaranteed theoretically. As expected, our method outperforms window-based methods and HMM-based methods in the experiment.

The CRF model also allow us to put together more than necessary features, because the feature weights that we got from the learning step will decide whether they are in use or not in the final model. However, as for now, the limited data size we got may prevent us from learning out the actual value of some under-represented features, and may result in severe over-fitting if we introduce too many features at a time. On the other hand, the iterative methods in the learning step may have a higher difficulty in convergence as more and more free feature parameters are added into the model, because an approximation is being used. Sometimes, singularity may occur in the approximation to the Hessian matrix [2]. In such case, we used the identity matrix to replace it, which is the same as its initial setting. The analysis and improving of convergence speed regarding various free parameter set could be a future work.

As for now, our feature functions are limited to containing only neighboring hidden states. More variety of features, such as long distance features between two hidden states that are away from each other and features involving more than two hidden states, are desired when trying to encode some knowledge. For example, we will need long distance features to encode motif co-occurrence, some other kind to directly describe motif spacing and CRM length, etc. However, complex feature functions could make the algorithms currently used in the learning step invalid, therefore alternative algorithms need be studied. There is a (hidden) trade-off between the express power of feature functions and the efficiency of learning. This will be one of the future directions to work on.

It is noticeable that an offset is presented in Eq (8) (11) (13), which tries to move the mean value of a feature to 0. The motivation is trying to minimize the impact of adding/removing the feature to other weights. It is helpful in practice.

A special prediction scheme, rank decoding, is used in the paper. We control the number of positive predictions made rather than a common threshold for probability values. This can strike a good balance

[2] The second derivative matrix of target function, log-likelihood or log-posterior-probability, with respect to the variable vector, λ.

between sequences, because longer sequences tend to fit into a model worse when it is different from the (unknown) real model. On the other hand, this scheme is reasonable in the sense of working load when we want to verify the predictions in biology experiments. Sequence decoding, another prediction scheme, does not work at most time, which barely output positive predictions, because of the modeling error accumulated along the long sequence. MAP decoding may sometimes work well.