BMC Bioinformatics

Methodology article

Optimized mixed Markov models for motif identification

Weichun Huang*[1,2,3], David M Umbach[2], Uwe Ohler[3] and Leping Li[2]

Address: [1]Bioinformatics Research Center, North Carolina State University, Raleigh, NC 27606, USA, [2]Biostatistics Branch, The National Institute of Environmental Health Sciences, National Institutes of Health, RTP, NC 27709, USA and [3]Institute for Genome Sciences & Policy, Duke University Medical Center, Durham, NC 27708, USA

Email: Weichun Huang* - huang6@niehs.nih.gov; David M Umbach - umbach@niehs.nih.gov; Uwe Ohler - uwe.ohler@duke.edu; Leping Li - li3@niehs.nih.gov

* Corresponding author

Published: 02 June 2006

BMC Bioinformatics 2006, **7**:279 doi:10.1186/1471-2105-7-279

Received: 08 May 2006
Accepted: 02 June 2006

This article is available from: http://www.biomedcentral.com/1471-2105/7/279

Abstract

Background: Identifying functional elements, such as transcriptional factor binding sites, is a fundamental step in reconstructing gene regulatory networks and remains a challenging issue, largely due to limited availability of training samples.

Results: We introduce a novel and flexible model, the Optimized Mixture Markov model (OMiMa), and related methods to allow adjustment of model complexity for different motifs. In comparison with other leading methods, OMiMa can incorporate more than the NNSplice's pairwise dependencies; OMiMa avoids model over-fitting better than the Permuted Variable Length Markov Model (PVLMM); and OMiMa requires smaller training samples than the Maximum Entropy Model (MEM). Testing on both simulated and actual data (regulatory *cis*-elements and splice sites), we found OMiMa's performance superior to the other leading methods in terms of prediction accuracy, required size of training data or computational time. Our OMiMa system, to our knowledge, is the only motif finding tool that incorporates automatic selection of the best model. OMiMa is freely available at [1].

Conclusion: Our optimized mixture of Markov models represents an alternative to the existing methods for modeling dependent structures within a biological motif. Our model is conceptually simple and effective, and can improve prediction accuracy and/or computational speed over other leading methods.

Background

Biological sequences, including DNA, RNA and proteins, contain functionally important motifs, such as transcription factor binding sites (TFBS), RNA splice sites, and protein domains. With the increasing-availability of genome sequences, identification of such functional motifs not only plays important roles in gene finding and function prediction but also is a fundamental step in reconstructing gene regulatory networks and in revealing gene evolutionary mechanisms [2-6].

A commonly used model for motif identification is the Weight Matrix Model (WMM) proposed by Staden [7], also called the Position Weight Matrix (PWM) or Mononucleotide Weight Matrix (MWM). A PWM is usually generated from a set of aligned instances of known motif sequences, using the observed position-specific base frequencies and/or prior information. Stormo and Fields [8] showed that the PWM score of a motif is proportional to the total binding energy contributed by individual bases. PWM has been used by many motif identification pro-

grams, *e.g.*, MatInspector [9] and Match [10], and performs reasonably well for motif identification. While a PWM can capture both nucleotide preferences at each position and different levels of position specificity, it does not account for functional dependencies between positions. Recent studies [11-15] indicate that there are often important interactions between positions, adjacent as well as non-adjacent, within a motif. The inability of the PWM to capture such dependencies is a limitation as the PWM model often produces a large number of false positives in a genome-wide scan [16].

Many models have been developed to incorporate position dependencies. Motif models, such as the Dinucleotide Weight Matrix Model (DWMM) [17] and the Weight Array Model (WAM) [18], can incorporate dependencies between adjacent positions. To incorporate further dependencies of non-adjacent positions, Ponomarenko *et al.* [19] extended DWMM by introducing the Oligonucleotide Weight Matrix model, which includes a comprehensive set of oliogonucleotide matrices classified into 5 biological function categories. A WAM could also be extended to a high order WAM in principle, *e.g.*, windowed *2*nd order WAM [2]. However, the exponentially increased number of parameters of these models makes them impractical due to insufficient training data. To address the weaknesses of WAM in incorporating long-range interactions, Burge and Karlin [2] proposed the Maximal Dependence Decomposition (MDD) model, which has a binary tree structure formed by a set of conditional WAMs. While the MDD model can capture non-adjacent dependencies through the conditional WAM models, it still requires a rather large number of training sequences, which are partitioned into smaller subsets to train all conditional WAMs. To alleviate the requirement of a large training set, Cai *et al.* [20] developed a Bayesian tree to model dependencies within RNA splice sites; Ellrott *et al.* [21] suggested a position order optimized Markov chain model, which reorders motif positions to bring distant but dependent positions into near neighbors. More recently, several other models have been developed, including Bayesian networks for modeling protein-DNA binding sites [22], Maximum Entropy Model (MEM) for splice site identification [23], Permuted Variable Length Markov Model (PVLMM) for finding transcription factor binding sites and splice sites [24]. For a biological motif with position dependencies, these models can show improvement in prediction accuracy over the models that assume independence. Incorporating position dependencies can also improve the accuracy of *de novo* motif discovery [25].

In this paper, we present a new and flexible motif model, the OMiMa, to incorporate position dependencies within a motif. OMiMa can not only adjust model complexity according to motif dependency structures but also minimize model complexity without compromising prediction accuracy. As an integrated part of OMiMa, we also introduce the Directed Neighbor-Joining (DNJ) method to optimally rearrange positions to minimize Markov order. We then describe and discuss the methods for selecting the best model. We implement our model into the OMiMa system that is freely available to the public.

Results

Mixed Markov models

Let X_i be the discrete random variable associated with position i in a biological motif **X** of length w. For DNA sequences, X_i takes values from set B = {A, C, G, T}; and for protein sequences, X_i takes values from 20 different amino acids. X_i follows a multinomial distribution. Let $\mathbf{X}_i^k = X_{i-k}...X_{i-1}$ and $\mathbf{x}_i^k = x_{i-k}...x_{i-1}$, where k = 0,..., w - 1; upper case **X** (X_i) is a random variable and lower case **x** (x_i) is a particular value. The x_i^0 denotes an empty sequence and $\Pr(\mathbf{X}_i^0 = x_i^0) = 1$. Additionally, let $X_{-j} = X_{w-j}$, $x_{-j} = x_{w-j}$, where j = 0...w - 1. If one uses the k^{th} order Markov model (M_k), the probability of observing a motif sequence x is just the product of conditional/transition probabilities. Let M_k^L be a k^{th} order Markov model of a linear chain, and M_k^C be a k^{th} order Markov model of a circular chain. The probability of a motif sequence is given by equation (1) for a linear chain and equation (2) for a circular chain, respectively.

$$\Pr(\mathbf{x} \mid M_k^L) = \Pr(\mathbf{X}_{k+1}^k = \mathbf{x}_{k+1}^k) \prod_{i=k+1}^{w} \Pr(X_i = x_i \mid \mathbf{X}_i^k = \mathbf{x}_i^k) \quad (1)$$

$$\Pr(\mathbf{x} \mid M_k^C) = \prod_{i=1}^{w} \Pr(X_i = x_i \mid \mathbf{X}_i^k = \mathbf{x}_i^k) \quad (2)$$

Compared to a linear Markov chain, a circular Markov chain incorporates additional dependencies that may contain subtle signals that allow the model to distinguish true motifs from false ones, especially when false motifs are similar to true motifs.

Suppose a motif **X** can be divided into m independent sub-motifs, that is $\mathbf{X} = \mathbf{Y}_1,... \mathbf{Y}_m$ and each sub-motif is modeled as an independent Markov chain, that is $M_X = M_{Y_1}, \ldots, M_{Y_m}$, then the probability of the sequence (x) given the Markov models is:

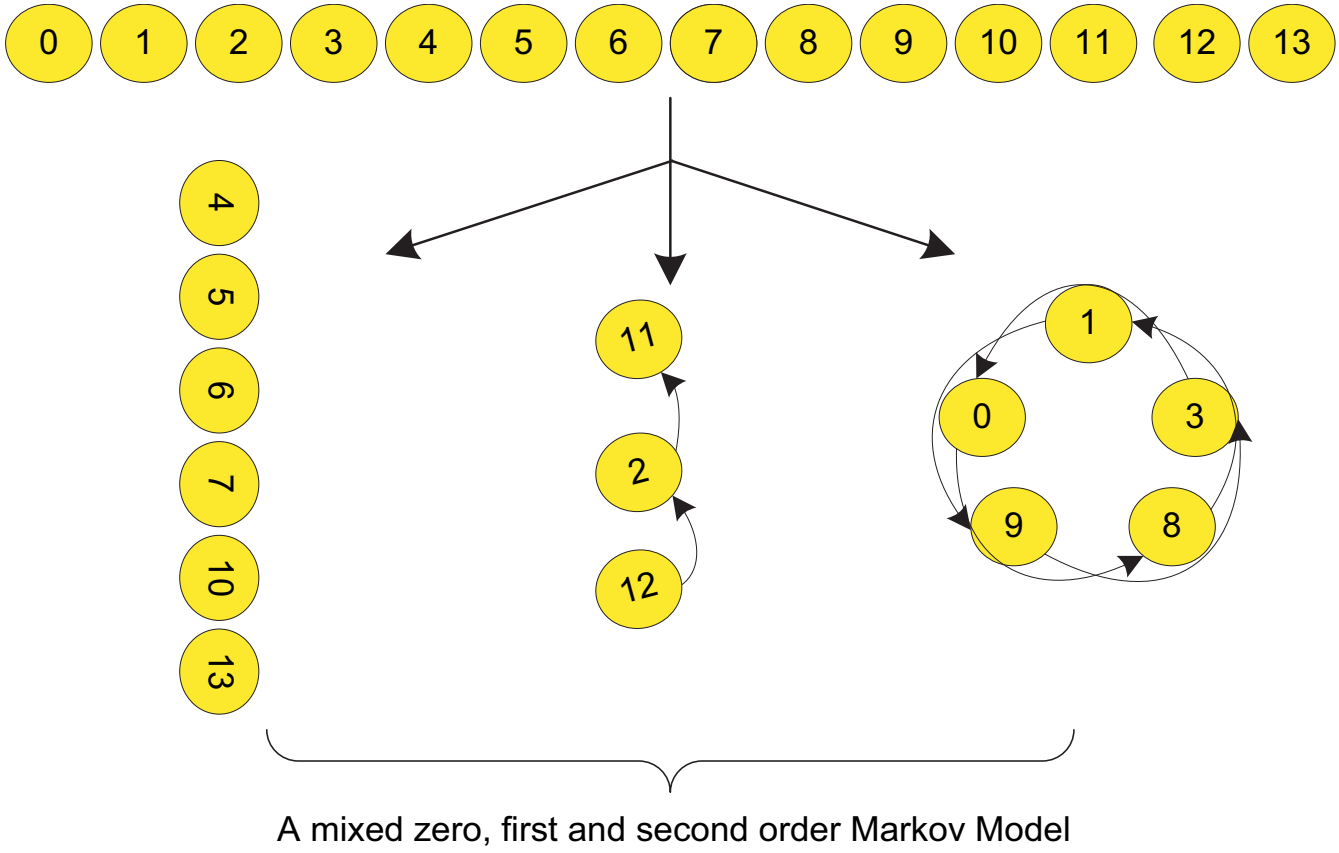


Figure 1
The graphic representation of a mixture of Markov models. A graphic representation of a mixture of Markov models. On the top is a motif of length 14 bases. On the left, 6 positions, which are independent of each other and all other positions, form a 0^{th} order Markov chain. In the middle, 3 positions form a linear chain of 1^{st} order Markov model. On the right, the remaining positions that closely depend on each other form a circular chain of the 2^{nd} order Markov model.

$$\Pr(\mathbf{X} = \mathbf{x} \mid M_X) = \prod_{i=1}^{m} \Pr(\mathbf{Y}_i = \mathbf{y}_i \mid M_{Y_i})$$

These independent Markov models, each of which is position-optimized for its corresponding sub-motif, form an Optimized Mixture of Markov models (OMiMa). An example of OMiMa is illustrated in Figure 1. However, for short motifs such as transcription factor binding sites, we can use a simple mixture of Markov models consisting of only one 0^{th} order and one k^{th} order chain (Figure 2). For convenience, we refer such a mixture model as '*a 0-k mixture model*'. Since the k^{th} order Markov chain of '*a 0-k mixture model*' can be either linear or circular, we also use terms '*a 0-k mixture linear model*' and '*a 0-k mixture circular model*' to distinguish them. In the following, we describe methods for the general mixture Markov model, while we use the simple 0-k mixture model for our testing.

Conceivably, the different parts of a motif could have distinct roles in the interaction with their partners. Motif positions involved in the same role can be highly dependent, whereas those involved unrelated roles are likely independent. A mixture of Markov models seems an ideal fit by modeling different signals with different sub-models. A 0^{th} order Markov chain can effectively model strong signals such as those embedded in highly conserved positions where the probability of a certain base occurring is almost one. In addition, positions where base composition contributes little or nothing to motif function need no more complex model than a 0^{th} order Markov model. On the other hand, a higher order Markov model is necessary for detecting subtle dependency signals that can be essential for distinguishing true motifs from false ones.

Motif dissection

To apply the mixture of Markov models to a motif, the first step is to dissect the motif into several independent

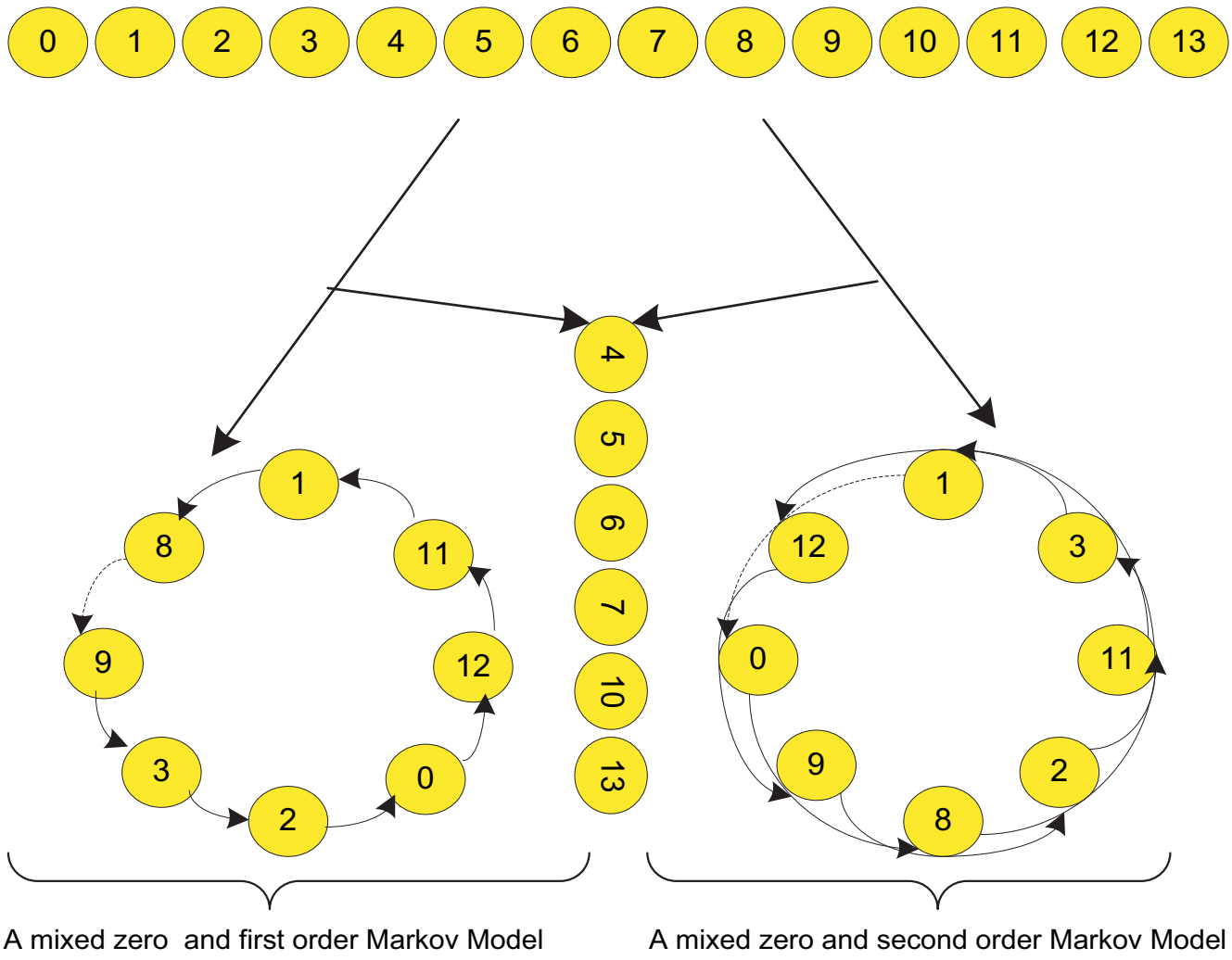


Figure 2
The graphic representation of the 0-k mixture model for TFBS. The simple mixture of Markov models for TFBS. Since TFBS are short (5–16 bases), a mixture model consisting of the 0^{th} order and $1^{st}/2^{nd}$ order Markov chains is generally adequate for predicting new binding sites. The sub-motif formed by independent positions is modeled by a 0^{th} order Markov order model. The sub-motif forming by the remaining positions is modeled by either a 1^{st} or 2^{nd} order Markov chain, which can be either linear (break at dotted arrows) or circular.

sub-motifs, each of which is modeled as a Markov chain. For a given set of sequences of a motif, we employ chi-square tests to find significant pairwise dependencies between positions within the motif (see also [21]). Based on pairwise dependencies, motif positions are grouped into independent sets, each forming a Markov chain. The outline of our procedure for grouping motif positions is described in the following steps.

1. Calculate base frequencies for each position, and find highly conserved positions where the observed frequency of a certain base (almost) equals 1. These conserved positions then are put into set H as defined below.

$$H = \{i : \max_{x \in B} f(i,x) \approx 1\}$$

where $f(i, x)$ is the frequency of base x at position i, and B is the set of bases.

2. Place remaining positions in the set M, and calculate pairwise chi-square values for every pair of positions in M.

$$\chi^2_{i,j} = \sum_{x_i \in B_i} \sum_{x_j \in B_j} \frac{(O(x_i, x_j) - E(x_i, x_j))^2}{E(x_i, x_j)} \quad (3)$$

where B_i and B_j are the sets of bases observed in positions i and j, respectively; $O(x_i, x_j)$ and $E(x_i, x_j)$ are the observed and expected counts of pair (x_i, x_j), respectively. $E(x_i, x_j)$ is the product of observed base frequencies x_i and x_j. The degrees of freedom of this test is $(|B_i| - 1) \times (|B_j| - 1)$, where $|B_i|$ and $|B_j|$ are the number of different bases in sets B_i and B_j, respectively.

3. Based on the above χ^2 tests, find all positions that show little dependence on any other positions in M, and move them to the set I, as defined by

$$I = \left\{ i = \min_{i \neq j; i, j \in M} p_{i,j} > \alpha \right\}$$

Here $p_{i,j}$ is the p-value corresponding to $\chi^2_{i,j}$, and α is the significance level, *e.g.*, 0.05.

4. The remaining positions in M are further grouped into subsets by iterating the following rules:

(a) Set s = 1.

(b) Calculate $\theta_i = \sum_{j \in M, j \neq i} \delta(p_{i,j} < \alpha)$ for each position i in M, where δ is a 0/1 indicator function. Find the largest θ_i, and move position i and all positions j that $p_{i,j} < \alpha$ from M into a new set C_s.

(c) For each remaining position, check if it significantly depends on any position in C_s. If it does, then move it from M into C_s.

(d) If M is not empty, update s = s+1 and go back to step (b).

Step 4 above essentially groups positions into independent subsets, each potentially forming a functional unit. For the special 0-k mixture model, we simply set $M = C_1$ at this step.

Markov chain optimization

The next step is to arrange the positions in each subset into a Markov chain. Since the positions in sets H and I are independent of each other, they can be arranged in their natural order to form a 0^{th} order Markov chain. The positions in H can also be treated differently from those in set I in motif identification by requiring a perfect match for a true site. Sets C_s are different. The position arrangement for each set C_s needs to be optimized so that the Markov model can account for most dependencies while minimizing the Markov order. For a given set C_s, we use the median (K_s) of θ $(\theta = \{\theta_j, j \in C_s\})$ as the maximum order of its potential Markov model. We then optimize position arrangement for the k^{th} order Markov chain $(k = 0, \ldots, K_s)$ by the Directed Neighbor-Joining (DNJ) method described below.

The neighbor-joining (NJ) method proposed by Saitou and Nei [26] is a well-known distance method for phylogenetic tree reconstruction. The principle of the NJ method is to find pairs of operational taxonomic units that minimize the total branch length at each stage of clustering. Our DNJ method is based on the exactly same principle. The only major difference is that DNJ needs to consider the direction in joining two nearest neighbors to form a new node while NJ does not. Instead of producing a phylogenetic tree as the NJ method does, DNJ method creates a chain structure, which arranges closely dependent positions as the nearest neighbors. The DNJ method for constructing a k^{th} order Markov chain from a given subset (C_s) of motif positions is described in the following steps (see Figure 3 for an example).

1. For a given set C_s, put each position in the set into a different vector. Here a vector is represented by a letter, an arrow at the top of the letter may be used to indicate the direction of a vector, *e.g.*, a stands for either $\vec{a}$ or $\overleftarrow{a}$. If $\vec{a}$ = (1, 2, 3), then $\overleftarrow{a}$ = (3, 2, 1), $\vec{a}\,\overleftarrow{a}$ = (1, 2, 3, 3, 2, 1), and $\vec{a}\,\vec{a}$ = (1, 2, 3, 1, 2, 3). Initially, each vector has only one position.

2. Create an initial distance matrix (d) whose elements are $d(u, v) = p_{i,j}$, where i is the position in vector u, j is the position in vector v, and $p_{i,j}$ is the p-value of chi-square test described above.

3. Convert the distance matrix d to the transformed distance matrix D, whose elements are $D(u, v)$, by the following conversion function (see [26]):

$$D(u,v) = d(u,v) - \frac{(r_u + r_v)}{(V-2)}$$

$$\text{where} \quad r_u = \sum_{u \neq z, u, z \in C_s} d(u,z)$$

Where V is the number of vectors under consideration, and its value decreases by 1 in each iteration.

4. Find the minimum $D(u, v)$ in D. Then a new vector x is formed by joining vector u and v according to **Algorithm 1** [see Additional file 1] for a k^{th} order Markov chain.

5. Update the matrix d by replacing u and v by x. The distance of x to other remaining vector γ is defined by:

$$d(x, \gamma) = (d(u, \gamma) + d(v, \gamma) - d(u, v))/2$$

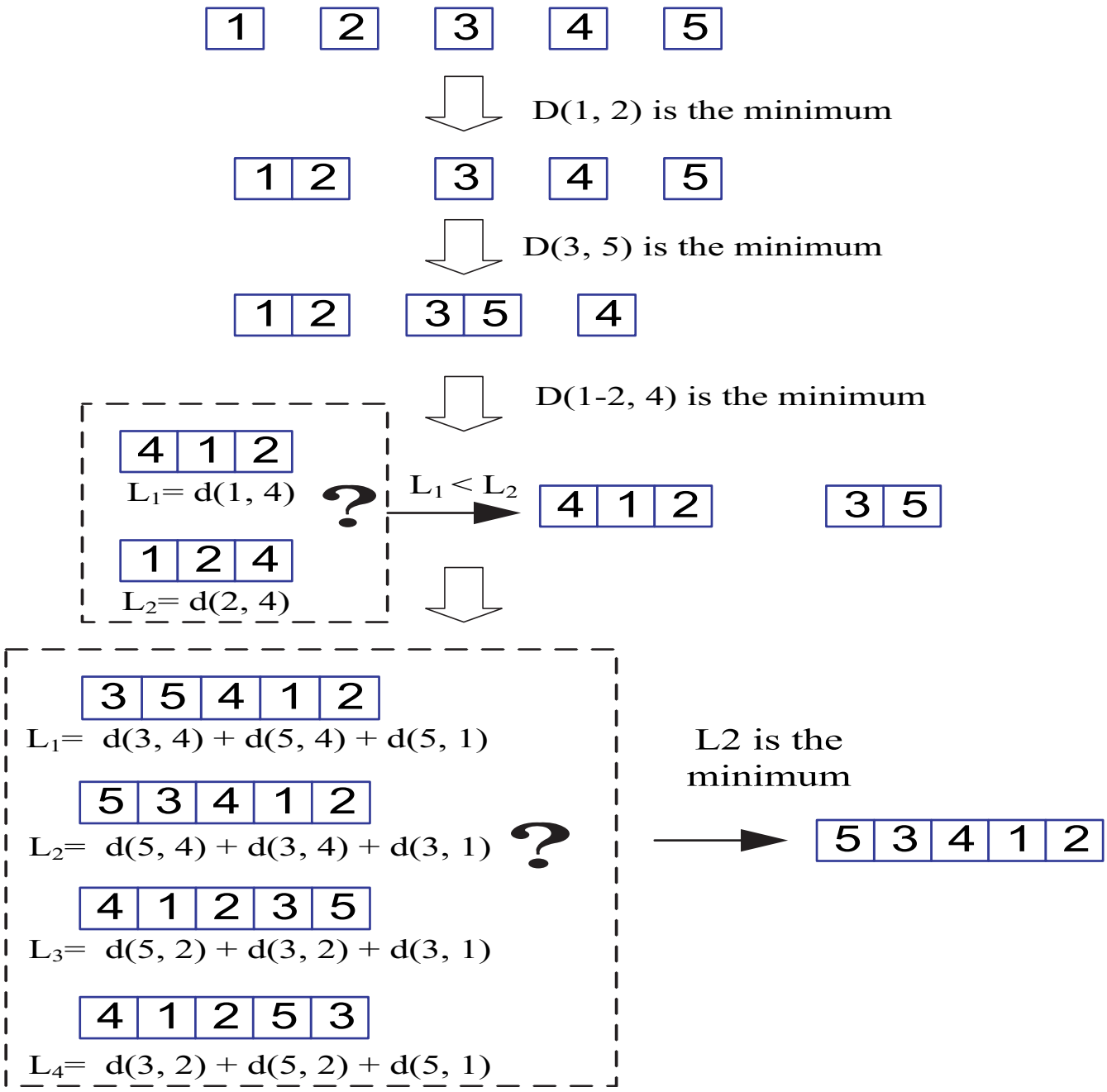


Figure 3
Illustration of the DNJ method for Markov chain optimization. An example of the DNJ method to optimize the 2nd order Markov chain.

6. Go back to step 3 if the number of vectors in C_s is larger than 2, otherwise join the last two vectors according to **Algorithm 1**.

The order of positions in the final vector is the optimized linear chain for Markov model. Joining the first position to the last position in the vector forms a circular chain. A linear chain could be further optimized by forming a circular chain first from the final vector, then breaking the circular chain between positions with the weakest dependency, *e.g.*, between positions i and j where $p_{i,j}$ is the largest or the log-likelihood of the corresponding linear chain model is maximized. DNJ not only optimizes position order for linear chain models but also improves circular chain models, particularly when the order of Markov model is low, *e.g.*, 1st or 2nd order Markov models.

Model selection

Many different mixtures of Markov models can be formed from the combination of different Markov chains. It is essential to choose the model that minimizes prediction error. In model selection, we first fit each model using maximum likelihood smoothed by a Dirichlet prior [see Additional file 1], then compute either the Akaike information criterion (AIC) [27] or the Bayesian information criterion (BIC) [28]. The model with the minimum value of AIC or BIC is selected as the potential best model. Minimizing AIC is the same as choosing the model with the minimum prediction error or loss, while minimizing BIC is equivalent to choosing the model with the largest posterior probability. Nonetheless, AIC and BIC have a similar form:

$$-2 \cdot loglik + \lambda \cdot DF$$

where λ = 2 for AIC and λ = log(N) for BIC (N is the number of sequences); *loglik* is the maximized log likelihood of data given the model; DF is the degrees of freedom (number of free parameters). We replace DF with the effective degrees of freedom (EDF) in calculating AIC or BIC of the mixture of Markov models, which enables an appropriate model to be selected (see sub-section *Effective degrees of freedom*). There is no clear better choice between AIC and BIC for model selection. AIC tends to choose a model too complex as $N \to \infty$, while BIC tends to choose a model too simple when N is small. In our test on 61 different TFBS datasets, whose sample sizes range from 20 to 130, we preferred AIC to BIC for picking an appropriate model.

Effective degrees of freedom

Let B be the set of bases (|B| denotes the number of different bases in B), *e.g.*, for DNA sequences B = {*A*, *C*, *G*, *T*} (|B| = 4). For a motif of length w, the DF for a k order Markov model is $(|B|^k - 1) \times (w - k)$ for a linear Markov chain; and $(|B|^k - 1) \times w$ for a circular chain model. That is, the DF increases exponentially as the order of Markov chain increases. As a result, AIC or BIC often pick a simpler mixture model than the best model, especially when |B| is large. Tested on 61 human regulatory motifs from the Transfac database (ver. 7.4) [29], we found that both AIC and BIC selected the 0^{th} order Markov models for all 61 DNA regulatory motifs when using the DF. To avoid picking overly simple models, we used the EDF described below to calculate AIC and BIC.

Generally, only a subset of bases from B appears in a particular position of a set of biological motifs. The more conserved a position, the fewer bases are in the subset. The EDF for a model is related to the observed bases in training samples. For example, suppose that one would like to estimate nucleotide frequencies occurring in a position in a set of DNA training motifs. If only base A is observed in the position, then one needs to estimate only the frequency of A, the remaining parameters, *i.e.*, the frequencies of C, G, T, can be derived from any prior information. Therefore, the actual DF is one in this case. For our mixture of Markov models, the EDF is defined as the number of parameters that are direct estimates of the observed bases in a training motif set. Let b_i be the base set observed in a position i of a training set of motifs. Additionally, let h^k be the sequence of motif positions in the k^{th} order Markov chain, h_i^k be the motif position in the i^{th} element of h^k, and $\sum|h^k| = w$ ($|h^k|$ is the number of positions in h^k), then we define the EDF for the k^{th} Markov chain as

$$\mathrm{EDF}_{\mathrm{k_L}} = \sum_{i=k+1}^{|h^k|} \max(b_{h_{i-k}^k} \times \cdots \times b_{h_i^k} - 1, 1)$$

$$\mathrm{EDF}_{\mathrm{k_C}} = \sum_{i=1}^{|h^k|} \max(b_{h_{i-k}^k} \times \cdots \times b_{h_i^k} - 1, 1)$$

where $h_{i-k}^k = h_{|h^k|-i+k}^k$ if $i - k \le 0$; $\mathrm{EDF}_{\mathrm{k_L}}$ and $\mathrm{EDF}_{\mathrm{k_C}}$ are for linear and circular chains, respectively. The total EDF for a mixture Markov model is just the sum of EDFs of all individual chains. For example, the total EDF for the special 0-k mixture model equals to the EDF sum of the 0^{th} and the k^{th} order chains.

Performance assessment

We test the effectiveness of our method on TFBS data and the donor splice sites, where training data for OMiMa are a set of sequences of a motif. For prediction results, we use the following abbreviations for empirical quantities: *TP* (# true positives), *TN* (# true negatives), *FP* (# false positives), *FN* (# false negatives), *Ac* (Accuracy), *Sn* (sensitivity), *Sp* (specificity), and *Mc* (Matthews correlation coefficient). *Sn*, *Sp*, *Ac*, and *Mc* are defined as:

$$Sn = \frac{TP}{TP + FN}$$

$$Sp = \frac{TN}{TN + FP}$$

$$Ac = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Mc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}}$$

Matthews correlation coefficient [30], also called Phi (correlation) coefficient, has a value between -1 and 1, with 1, 0, and < 0 indicating a perfect prediction, a random prediction, and a worse than random prediction, respectively.

OMiMa can use two ways to score a motif site **x**: log-likelihood and log-likelihood ratio, which are defined by

$$\log-\text{likelihood} = \log \Pr(\mathbf{x} \mid M_s)$$

$$\log-\text{likelihood ratio} = \log \frac{\Pr(\mathbf{x} \mid M_s)}{\Pr(\mathbf{x} \mid M_b)}$$

where M_s is the signal model trained by true motif sites, and M_b is the background model or false signal model trained by background sequences or false motif sites. A sequence **x** is predicted as a positive site if the score of **x** is larger than a certain threshold. We select a cutoff threshold using one of the following three criteria: balanced sensitivity and specificity, the maximum prediction accuracy, and the maximum Matthews correlation coefficient. Each potential threshold yields an estimated true positive rate and a false positive rate. The plot of true positive rates against false positive rates generates a Receiver Operating Characteristic (ROC) curve, which can be used for comparing and selecting the best model.

We used a three-symbol notation 'k-m-s' to distinguish different models, where 'k' stands for a 0-k mixture Markov model, 'm' is either 'L' or 'C' to indicate whether the k^{th} order chain is linear ('L') or circular ('C'), and 's' is either 0 or 1 to indicate whether log likelihood score (0) or log-likelihood ratio score (1) is used. For example, '1-L-l' stands for a 0–1 mixture of linear Markov models that uses log-likelihood ratio to score a motif site.

Effectiveness of DNJ method for optimization

To assess the ability of our DNJ method for optimizing a Markov chain, we compared the DNJ method with random permutation method. In this evaluation, we used a 0-k mixture model (k = 0, 1, 2) (Figure 2) to model transcription factor binding sites from the Transfac database. For each TFBS, we first fitted a 0-k mixture model (denoted as M_{DNJ}) with its k^{th} order Markov chain optimized by the DNJ method. We calculated the log-likelihood of the data given the model M_{DNJ} (log $\Pr(data|M_{DNJ})$). Second, with the same data, we fitted a new 0-k mixture model (denoted as M_R), which is the same as M_{DNJ} except that the positions in its k^{th} order chain are ordered by random permutation, and calculated log$\Pr(data|M_R)$. This step was repeated 1,000 times, so we have 1,000 log-likelihoods of the randomly permuted models $(M_{R_1}, \ldots, M_{R_{1000}})$. We then calculated the empirical *p_value* of the DNJ-optimized model as follows:

$$p-value = \frac{\sum_{i=1}^{1000} \delta(\log \Pr(data \mid M_{DNJ}) < \log \Pr(data \mid M_{R_i}))}{1000}$$

where δ is an indicator function with value 1 if condition is true, and 0 otherwise. The smaller the *p_value*, the better the DNJ optimization is; and *p_value* = 0 means the DNJ-optimized model performs better than any one of the 1,000 randomly permuted models. The *p_value* approximates the probability of observing log $\Pr(data| M_{R_i})$ larger than (log $\Pr(data|M_{DNJ})$.

Fifty-three human transcription factors, whose binding sites contain at least four dependent positions by the χ^2 test given by equation (3), are selected for this evaluation (Table 1). The assessment was performed on four 0-k mixture models: 1st order linear chain, 1st order circular chain, 2nd order linear chain, and 2nd order circular chain.

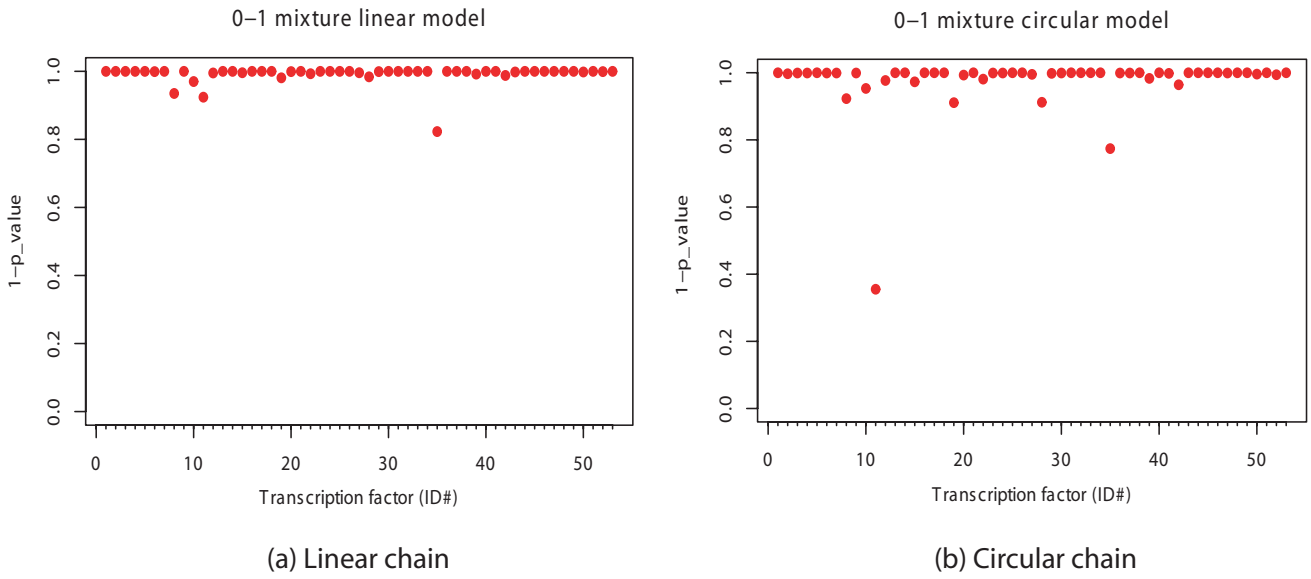
Results suggest that DNJ method performed remarkably well in optimizing the 1st order linear Markov chains, that in 49 out 53 cases, the DNJ optimized models were the best or close to the best (Figure 4a). The optimization for the 2nd order linear chains was slightly worse than that for the 1st order linear chains, partially because the DNJ method relies only on the pairwise dependencies between two single positions. Nevertheless, most of the DNJ optimized models were still close to the best [see Additional file 1 Figure 1a]. Although our DNJ method was designed for optimizing linear Markov chains, it still worked well in optimizing the 1st order circular Markov chains (Figure 4b). However, the DNJ method did not perform well in optimizing the 2nd order circular Markov chains [see Additional file 1 Figure 1b].

We used AP1 (activating protein 1) transcription factor binding sites (Transfac ID V$AP1_Q4_01) as an example of how DNJ optimization can improve performance of a 0–1 or 0–2 mixture model. We plotted the histogram of the log-likelihood per instance given a model M_{R_i}, log $\Pr(data| M_{R_i})/N$, where N is the number of sequences in the data set, and i = 1,..., 1000 for 1,000 mixture models of randomly permuted Markov chains. The histogram represents a simulated null-distribution of log-likelihood per instance given a mixture model. Then we mapped the location of the likelihood per instance given DNJ optimized model, (log $\Pr(data|M_{DNJ})/N$, in the histogram. For the transcription factor V$AP1_Q4_01, we found that for the 0–1 mixture model of either linear or circular structure, DNJ optimized models are better than any models from 1,000 random permutations (Figure 5).

Theoretically, the optimal model can be found by exhaustively searching through all possible models. An exhaustive search is not always possible in practice, however, as

Table 1: The optimized 1st order Markov chains for TFBS. The optimized arrangement of dependent positions within TFBS for the 1st order Markov model. N and N_D are total number of motif positions and the number of positions significantly dependent, respectively.

ID#	Name	N	N_D	Position order
1	V$AP1_Q4_01	8	8	7-3-1-2-0-6-5-4
2	V$AP1_Q6_01	9	8	2-3-1-4-5-7-8-6
3	V$AP1_Q2_01	12	9	4-3-5-6-7-10-11-9-1
4	V$CDPCR1_01	10	9	3-4-2-9-6-5-7-8-1
5	V$ATF_01	14	8	1-0-10-9-11-2-13-12
6	V$CHOP_01	13	10	5-4-6-7-9-10-0-8-11-12
7	V$CDPCR3_01	15	10	3-0-1-8-9-13-4-6-2-5
8	V$CDPCR3HD_01	10	5	1-8-9-2-7
9	V$CREB_Q2_01	14	8	1-11-12-0-2-3-9-8
10	V$CREB_Q4_01	11	6	7-6-1-8-9-10
11	V$CREB_Q3	6	4	4-5-1-0
12	V$CEBP_Q3	12	9	8-9-5-6-4-11-3-2-10
13	V$CEBPB_01	14	4	0-13-11-3
14	V$E2F_Q4_01	11	4	1-8-7-0
15	V$E2F_Q6_01	12	8	8-3-7-0-2-11-9-10
16	V$E2F1DP1_01	8	5	3-4-0-6-7
17	V$E2F1DP2_01	8	5	5-6-7-3-4
18	V$E2F4DP1_01	8	4	3-4-0-1
19	V$E2F4DP2_01	8	5	4-3-7-1-0
20	V$ETS_Q4	12	8	11-2-5-10-4-3-0-1
21	V$ELK1_02	14	4	10-11-2-3
22	V$FAC1_01	14	12	12-6-10-11-13-4-9-8-5-1-0-7
23	V$FOXD3_01	12	11	1-3-8-7-9-10-11-2-0-4-6
24	V$FOXO1_02	14	11	8-9-10-12-7-6-2-0-11-1-3
25	V$HNF4_Q6	9	7	4-3-2-6-8-1-7
26	V$HNF1_Q6	18	15	3-11-12-1-4-8-13-5-9-0-6-16-14-2-10
27	V$HNF3_Q6	13	11	1-10-7-5-3-4-12-9-0-2-8
28	V$E2F1DP1RB_01	8	5	1-7-3-0-4
29	V$IRF7_01	18	13	3-2-0-16-15-17-1-7-6-8-14-9-12
30	V$LUN1_01	17	8	8-9-10-7-12-11-14-13
31	V$MZF1_01	8	4	0-1-4-5
32	V$MYC_Q2	7	4	4-5-3-1
33	V$NFAT_Q4_01	10	4	6-8-9-5
34	V$NFKAPPAB_01	10	4	5-7-9-2
35	V$NKX22_01	10	6	9-8-6-1-0-7
36	V$OCT_Q6	11	10	8-2-0-10-5-3-9-6-4-7
37	V$PAX_Q6	11	10	10-6-7-0-9-3-1-5-4-2
38	V$PAX6_01	21	21	15-17-16-18-6-8-19-13-11-3-2-1-0-20-7-10-4-9-5-14-12
39	V$PBX1_02	15	10	6-12-2-0-3-1-11-13-14-4
40	V$RSRFC4_Q2	17	6	6-7-0-13-2-3
41	V$RSRFC4_01	16	8	6-7-9-1-2-13-12-8
42	V$STAT5A_01	15	7	8-12-1-13-0-4-5
43	V$SOX9_B1	14	9	1-13-0-2-11-5-3-10-4
44	V$SRY_01	7	4	4-6-0-1
45	V$SRY_02	12	4	1-3-11-4
46	V$STAT5A_02	24	16	7-12-20-15-16-17-18-19-22-1-21-13-5-6-9-23
47	V$SP1_Q2_01	10	7	7-3-8-0-4-9-5
48	V$SP1_Q4_01	13	13	0-2-11-12-6-1-3-10-9-8-7-4-5
49	V$SP1_Q6_01	10	10	3-5-8-9-0-7-4-2-6-1
50	V$USF_Q6_01	12	8	3-11-4-5-7-2-1-8
51	V$XBP1_01	17	9	13-5-3-4-15-11-10-12-0
52	V$ZID_01	13	8	6-7-4-8-12-10-9-11
53	I$DRI_01	10	7	6-9-8-7-0-1-2



(a) Linear chain (b) Circular chain

Figure 4
The performance of the DNJ optimized 0–1 mixture models. The performance of the DNJ optimized 0–1 mixture models of TFBS. The y-axis is 1-*p_value* measuring performance of the DNJ optimized models relative to the randomly permuted models. The values on x-axis are the ID numbers of 53 TFBS in the first column of Table 1. (a) 0–1 mixture linear models, (b) 0–1 mixture circular models.

the search space can be very large. The number of possible Markov chains is the factorial of the length of the Markov chain and dramatically increases as the length of chain increases. For example, the computational time for a motif of 15 bases (15! = 1.307674e + 12) can be practically unacceptable. Our DNJ method can deal with such long motifs because of its computational efficiency.

TFBS identification

One interesting application of our mixture model is TFBS identification. In this assessment, we used a couple of examples to show how OMiMa can improve prediction accuracy when there are position dependencies within a TFBS. We first tested our method on simulated data where the exact dependency structure of a TFBS is known. We tested whether OMiMa can capture such dependency and optimize the Markov model accordingly. Next, we tested our method on real motif data for AP1. In both examples, we compared OMiMa performance to PWM, PVLMM, and the 1^{st} order Markov model (1stMM) with its motif positions in the natural order. PVLMM, run on Microsoft Windows, is based on the variable length Markov model (VLMM) [31,32]. Except for its order and depth parameters, PVLMM was run under its default settings in all comparisons.

Simulated TFBS prediction

Many TFBS are palindromic sites bound by heterodimers/homodimers (*e.g.*,: Jun-Fos, Myc-Max, Max-Max and p50-p50). The sequences in two half sites of a palindromic TFBS are usually not perfectly complementary, and the strong binding to one half site may compensate for weak binding to the other. We simulated two imperfect palindromic TFBS (named A and B) of length 12 bases. For each TFBS, the bases in each position were generated from the uniform distribution (the frequency of each base is 0.25). The base in one half site and its reverse complementary base in the other half were generated using the probabilities listed in Table 2. Therefore, there are only pairwise position dependencies in the simulated TFBS. The position pair 0–11 has the strongest dependency, whereas the pair 5–6 has the weakest dependency (they are independent). Overall, motif A has stronger position dependencies than motif B. The false sites of TFBS were simulated from the uniform distribution of four nucleotides without any constraints of base pairing between the two half sites. The simulated data of each TFBS consist of a training set with 150 true sites, and a testing set with 150 true sites and 150 false sites. Using these simulated training sets, OMiMa found all true dependencies significant at level $\alpha = 0.05$ (see Table 2). For both TFBS, OMiMa was also able to arrange the positions of each dependent pair to be the nearest neighbors in their 0–1 mixture models: (See table 7)

In our simulation, the positions 5 and 6 were generated independently from all other positions, so they should be in the 0^{th} order chains. However, based on OMiMa's pair-

Table 2: Simulation of two palindromic TFBS. Simulation of two palindromic TFBS A and B. The first 2 columns are the complementary positions of the palindromic TFBS. The 3rd and 4th columns are simulation parameters, which specify the probabilities of forming a complementary base pair. The last 2 columns are the p-values of OMiMa's pairwise χ^2 tests of position dependency for the simulated data.

Position pair		Complementary Prob.		p-value	
1st	2nd	A	B	A	B
0	11	0.99	0.90	4.88e-88	3.84e-63
1	10	0.95	0.85	6.62e-72	2.66e-56
2	9	0.90	0.75	3.84e-69	2.25e-35
3	8	0.65	0.65	1.44e-19	5.89e-24
4	7	0.50	0.50	2.00e-07	3.05e-03
5	6	0.25	0.25	3.35e-01	1.43e-01

wise χ^2 tests for the training data, the position pair 5–8 (with p-value = 0.03) in TFBS A, and the position pair 6–10 (with p-value = 0.04) in TFBS B were declared dependent. That is why the positions 5 and 8 were arranged together in the model for TFBS A, and positions 6 and 10 were together for TFBS B. We compared the prediction results of OMiMa's 0–1 mixture model with those of PWM, 1stMM and the 1st order PVLMM (with depth 1). Results (Table 3) showed that OMiMa outperformed all other models, and PVLMM performed better than IstMM and PWM. Additionally, we used smaller training sets to access the performance of these methods on the same testing set. Smaller training sets, in sizes ranging from 15 to 150, were independently sampled (without replacement) from the original 150 sites for training. Results suggested that OMiMa performed consistently better than the other methods, regardless the size of a training set (Figure 6) [see Additional file 1 Figure 2].

AP1 TFBS prediction

We chose human AP1 TFBS (see Figure 7a) for this evaluation because of its relatively large number of known sites. In total, we had 119 true sites and 5950 false sites. The true sites were extracted from Transfac database (Transfac ID V$AP1_Q4_01), and false sites were randomly sampled from the non-coding regions of the human genome. Our χ^2 tests on the 119 true sites suggested that all positions showed some level of dependency with the neighboring pairs 0–2, 4–5, 5–6, and 4–6 showing strong dependencies (p-value < 1.0e-6). Noticeably, the positions 4, 5 and 6 are also the most conserved positions, so we expect that PWM would be reasonable good model for the TFBS. We randomly split both the true sites and false sites into 10 roughly equal-sized parts, and used a 10-fold cross validation to compare the performance of OMiMa's 0–1 mixture model with the others. OMiMa had advantage over the other three models in predicting TFBS that do not have strong long-range dependencies (Table 4). Results also showed the first order PVLMM did not perform better than 1stMM and PWM. We found that the first order PVLMM arranged the position pair 5–6, which showed the strong dependency, differently from OMiMa and 1stMM. In only 3 out 10 times, PVLMM arranged positions 5 and 6 as direct neighbors, while OMiMa did in 9 out 10 times, and 1stMM did naturally all times. This is

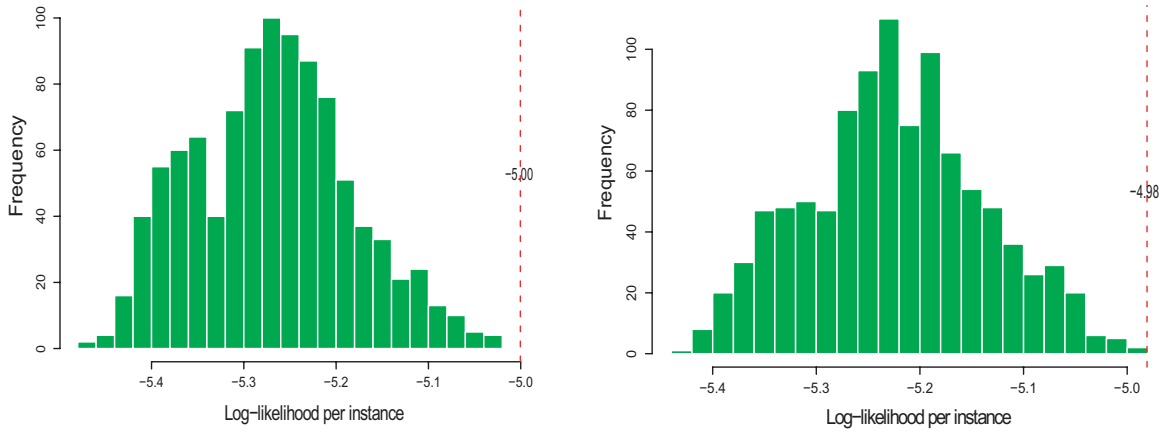


Figure 5
Modeling TFBS V$AP1_Q4_01. The performance of the optimized model of TFBS V$AP1_Q4_01. The histogram is the log-likelihood score distribution of 1,000 randomly permuted mixture models. The red reference line indicates the relative performance of the DNJ optimized model (a) 0–1 mixture linear model (b) 0–1 mixture circular model.

Table 3: Performance evaluation using simulated palindromic TFBS. Performance comparison of OMiMa (1-L-0) with PWM, 1stMM, and PVLMM (order 1 and depth 1) for predicting two simulated TFBS A and B. The performance was measured as the maximum *Mc* achieved by each model.

Motif	PWM	1stMM	PVLMM	OMiMa
A	0.306	0.414	0.807	0.914
B	0.253	0.428	0.647	0.794

one possible reason why PVLMM performed slightly worse.

Donor splice site recognition

The transcription of most higher eukaryotic genes involves RNA splicing, in which primary transcripts become mature mRNA by removing introns. The donor or 5' splice sites and the acceptor or 3' splice sites on the boundaries of exons and introns provides critical signals for precise splicing. Therefore, splice site recognition has been widely used by gene finding tools such as GENESCAN [2] and GENIE [33] for gene prediction. The splicing process starts with U1 snRNP binding to the donor site via base-pairing of U1 snRNA. The base pairing between U1 snRNA and the donor site, however, need not be perfectly complementary in all positions [34,35]. Both experimental and computational evidence suggest that there are mutually dependent positions within the donor site: a mis-matched pair of U1 snRNA and the donor sites at one position can be compensated for by a matching base pair at another position, and vice versa [2,24,36,37]. Modeling such dependency structure within the donor site has been used to improve donor site prediction [2,23,24,33]. We used two independent datasets of human donor sites to assess the performance of OMiMa in comparison with leading competitors.

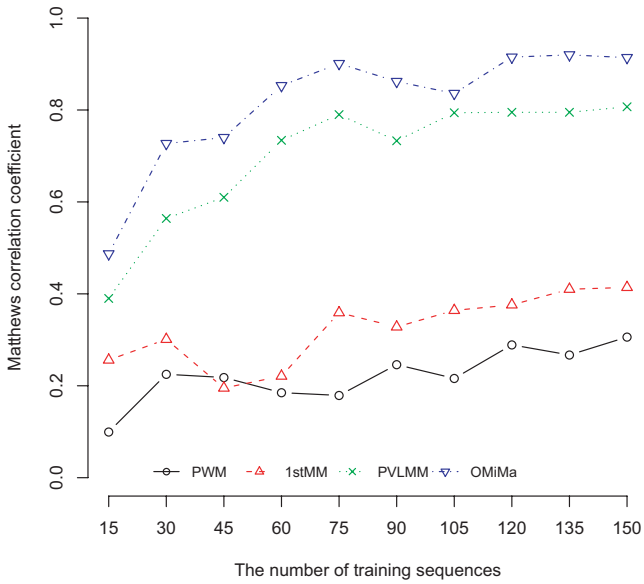


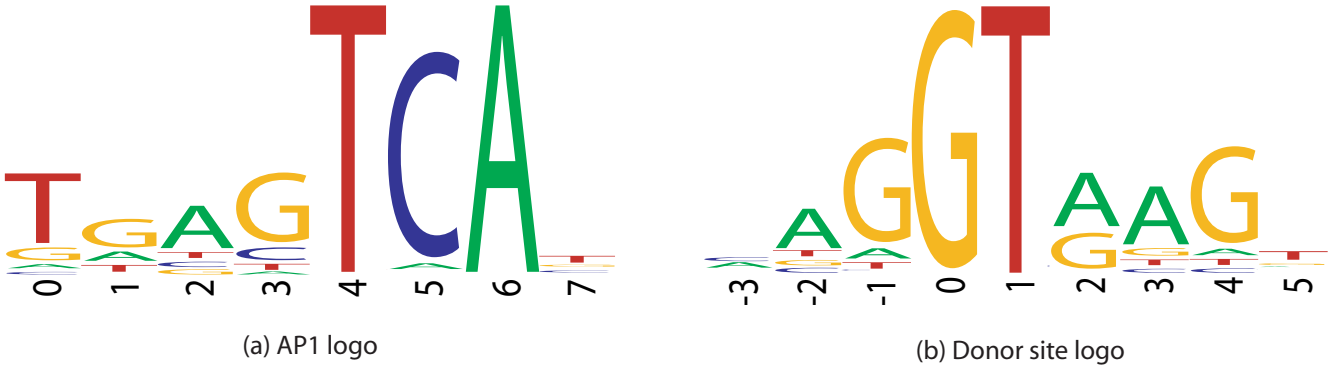
Figure 6
Performance comparison on the simulated palindromic TFBS. The performance comparison of different methods for predicting the simulated palindromic TFBS A. The x-axis shows the number of motif sequences used for training. The y-axis is the Matthews correlation coefficient of each method in predicting the same testing dataset (150 false and 150 true sites, respectively). The figure shows that OMiMa performed significantly better than the other methods, regardless the number of training samples.

Comparison with NNSplice and PVLMM

The test dataset of human donor splice sites (Reese data) was from [38]. This dataset has 6246 donor sites (1324 real and 4922 false) of length 15 bases from -7 to +8 around the conserved 'GT' dinucleotide. The dataset consists of a training set (containing $\frac{5}{6}$ of data) and a testing set (the remainder), which were previously used to assess the performance of NNSplice [33]. We used the same partitions for training and testing in the following comparisons.

First, we tested whether OMiMa which uses either AIC or BIC, can correctly pick the best model based on ROC analysis. We fitted a set of 0-k mixture models, in which the k^{th} order chains are either linear or circular and k ranges from 0 to 3, with the training data. We subsequently applied the fitted models to predict splice sites in testing data. The performances of different models were compared and evaluated by ROC analysis (Figure 8). In addition, we compared the maximum accuracy (*Ac*) and the maximum Matthews correlation efficient (*Mc*) achieved by each model (data not shown). The best models were 0–1 mixture models (Figure 8). Both the linear and circular models performed about the same. The best models picked by ROC analysis are consistent with those selected by OMiMa (AIC criterion). The selected models were further confirmed by a six-fold cross validation.

Using the best model selected above, we then compared OMiMa with NNSplice and PVLMM. NNSplice is based on a complex neural network model and is trained by both true sites and false sites. Since both OMiMa and NNSplice used the same training and testing data, their prediction results can be directly compared. We compared OMiMa's 1-L-1 and 1-C-1 models with the first order PVLMM (with depth 1) as all have similar model complexity. The results of NNSplice were reported at the NNSplice Web site [39]. We found that OMiMa had comparable prediction accuracy to NNSplice and PVLMM



(a) AP1 logo (b) Donor site logo

Figure 7

The sequence logos of AP1 TFBS and the donor site. Sequence logos of the AP1 TFBS and the donor splice site. The height of bases represents the information content at each position of a sequence motif. (a) the logo of AP1 TFBS. Note that the positions 4 and 6 of AP1 TFBS are not perfectly conserved. (b) the logo of donor splice site. The positions 0 and 1 are perfectly conserved. The logo plot was created by WebLogo [45].

(Table 5). In addition, OMiMa is much more computationally efficient than NNSplice and PVLMM [see Additional file 1].

Comparison with MEM and PVLMM

Given enough training data, we can use more complicated models than the 0–1 mixture model to improve prediction accuracy. In this evaluation, we test whether 0-k mixture models can compete with the MEM on a much larger dataset. This large donor site dataset (Yeo data), used to assess performance of the MEM, was from [40]. The dataset, extracted from 1821 non-redundant human transcripts, has 8,415 real and 179,438 decoy sites in the training set, and 4,208 real and 89,717 in decoy sites in the testing set. Each real site has length 9 bases from -3 to +6 around the conserved 'GT' of donor splice sites recognized by U-2 type spliceosome. The decoy sites are any other sequence segments in the exons and introns matching the pattern N_3GTN_4. So a decoy site can have the exactly same sequence as a real site. We applied this original training and testing sets to assess performance of OMiMa, where we used only log-likelihood ratio scoring. In addition, we ran a 3-fold cross-validation, in which the number of sites in new training and testing sets are roughly the same as those in the original ones [see Additional file 1 Table 2]. The top 4 models selected by AIC are 3-C-1, 3-L-1, 2-C-1 and 2-L-1, respectively, consistent with the ROC analysis. To find the top 4 sub-models of PVLMM by ROC analysis, we used a series of Markov orders and/or depths ($1 \le order \le 4$ and $order \ge depth$) to predict the same data sets. For convenience, we use notation "P:*k*-*d*" to denote a PVLMM of order k and depth d. We adopt notation in [23] for sub-models of MEM.

Table 4: TFBS V$AP1_Q4_01 prediction. Comparison of OMiMa (1-L-0/1-C-0), PWM, 1stMM, and PVLMM (order 1 and depth 1) for AP1 TFBS prediction. The performance results are the average values of 10-fold cross validation.

Model	*Sn*	*Sp*	*Mc*
PWM	0.857	0.997	0.860
1stMM	0.839	0.998	0.870
PVLMM	0.789	0.999	0.847
1-L-0	0.866	0.998	0.882
1-C-0	0.874	0.998	0.884

Briefly, the notation has the form "me*K*s*D*" or "me*K*x*D*", where "me" stands for maximum entropy; "*K*" is a number for the marginal order or the maximum length of an oligomer in consideration; "*D*" is the skip number or maximum skip number determining which positions the bases of an oligomer are from; "s" stands for skip number and "x" for the maximum skip. For example, model "me5s0" considers all marginal distributions of $p(x_i)$, $p(x_i, x_{i+1})$, $p(x_i, x_{i+1}, x_{i+2})$, $p(x_i, x_{i+1}, x_{i+2}, x_{i+3})$, $p(x_i, x_{i+1}, x_{i+2}, x_{i+3}, x_{i+4})$.

Comparison of the top 4 performers from each model class suggested that OMiMa performed comparably with MEM and better than PVLMM (Table 6) (all models' performances suffered on this data set because about 98% real sites appeared at least once in the decoy set). One advantage of OMiMa over MEM is that, for the models with similar performance, OMiMa's models generally have fewer parameters and thus require fewer training samples. Our test showed that OMiMa was able to retain similar performance of MEM even when trained by only

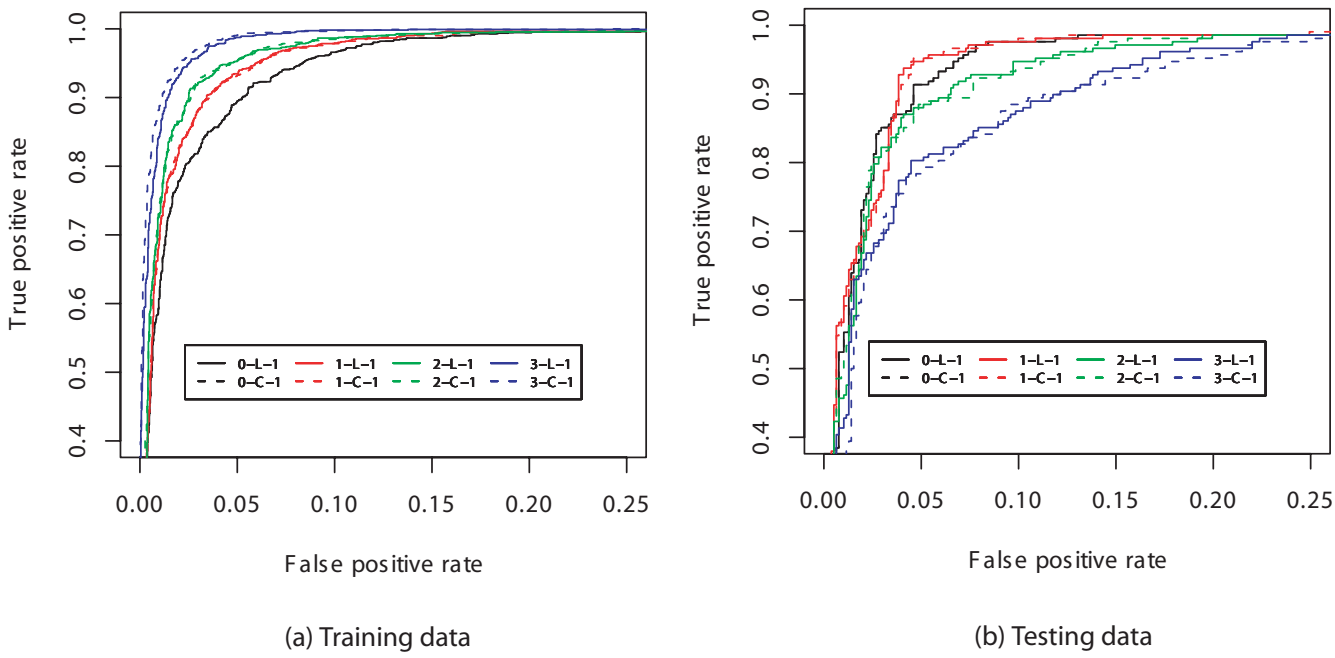


Figure 8

Comparison of different 0-k mixture models for donor splice site prediction. Comparison of different 0-k mixture models for donor splice site prediction by ROC curves. Based on the Area Under Curve (AUC) criterion, the figure indicates that: (a) For training data, the best models were 3-L-1 and 3-C-1 while the worst model is 0-L-1 (same as 0-C-1). (b) For testing data, the best models were 1-L-1 and 1-C-1 while the worst models are 3-L-1 and 3-C-1.

60% data of MEM's original training sets [see Additional file 1 Table 3].

Biological explanation

To compare OMiMa's fitted donor site models to biological knowledge about dependencies among positions, we examined the best donor models for the first donor dataset (Reese data) and for the second donor dataset (Yeo data). For convenience, let us mark the invariant 'GT' nucleotides in the boundary of exon/intron as the positions 0 and 1 of the donor site, respectively (see Figure 7b). First, based on 1,116 real donor sites in the Reese original training data, the 0–1 mixture model was selected as the best model with the following 1^{st} order chain.

-2 5 -1 3 4 -3 -7 -6 -5 -4 7 6 2

We found that this position arrangement is supported by the following biological evidence of base-pairing between U1 snRNA and the donor site: (a) 5'/3' compensation effect: a base pair at position -1 can prevent an aberrant splicing caused by a mis-matched pair at position 5 [37]; (b) Adjacent base-pair effect: a matching base pair at position 3 is rare in the absence of a matching base pair at position 4 [2,41]; (c) A matching base-pair at the non-conserved positions 6 and 7 can compensate for a mis-matched pair at position 2 [42]. Interestingly, the model also arranged non-conserved positions (-4, -5, -6, -7, 6, 7) together as it did for the other more conserved positions. Second, based on 8,415 real donor sites of the Yeo original training sites, the 0–3 mixture model was the best model. The optimized position order of its 3^{rd} order chain was:

2 5 -1 4 -2 3 -3

We can see that this model is consistent with the above evidence (a) and (b). In addition, it is well supported by experimentally verified position dependencies of position 4 on the positions -1, -2, 3 and 5 [37], and the computationally confirmed dependency of position -3 on position -2 due to the adjacent base-pair effect [2].

Discussion

The prediction accuracy of a probabilistic model is largely determined by the effectiveness of the model in characterizing a biological motif. Since there is large variation of the signals embedded in biological motifs, an effective model can be as simple as a consensus sequence or as complex as a fully connected network model. In this paper, we described a mixture of Markov models to allow adjustment of model complexity for different motifs. Also, we extended the traditional linear chain Markov model to the circular chain Markov model, which can bet-

Table 5: Comparison OMiMa with NNSplice and PVLMM for donor site prediction. Comparing two OMiMa models (1-L-1 and 1-C-1) with NNSplice's neural network model and PVLMM (order 1 and depth 1) for donor splice site prediction.

		Network	PVLMM	1-L-1	1-C-1
Ac maximized	*Ac*	0.951	0.927	0.955	0.954
	Sn	0.904	0.793	0.928	0.947
	Sp	0.963	0.963	0.962	0.955
Mc maximized	*Mc*	0.857	0.786	0.869	0.869
	Sn	0.942	0.889	0.938	0.952
	Sp	0.951	0.934	0.959	0.954

ter represent position dependencies within a motif in some cases. We presented a novel method, DNJ, for efficiently optimizing position arrangement of a non 0^{th} order Markov chain to incorporate most dependencies. We described methods for calculating the EDF and for selecting the best mixture Markov model. We implemented these methods in our motif finding OMiMa system, which is freely available. Finally, we demonstrated from different aspects in several examples that OMiMa can improve motif prediction accuracy in biological sequences.

The interaction of biological macromolecules, such as transcription factors bound to DNA sites, usually involves several highly dependent positions functioning as a unit. Many methods including Markov chains, Bayesian trees, and neutral networks have been used to model dependency structures within a motif. The Markov model is the simplest yet can be very powerful when it is optimized. Our results showed that the optimized Markov models performed better than the neural network model and PVLMM, and comparably with MEM for splice site prediction. The optimized Markov model can incorporate both local and non-local dependencies into the model, which enables it to compete with tree or network models in predicting short biological motifs. We also showed that the optimized Markov model can be an excellent motif predictor. Moreover, it is also computationally efficient due to its simplicity.

Table 6: Comparison of OMiMa PVLMM and MEM for donor site prediction. Comparing OMiMa with PVLMM and MEM for donor splice site prediction. The table shows Matthews correlation efficients (*Mc*) of top 4 models from each model class. The splice site data and results of MEM models were from Yeo and Burge [23]. In the 3-fold cross validation, the sample sizes for both training and testing sets are approximately equal to those of the original partition by Yeo and Burge.

MEM		PVLMM		OMiMa	
sub-model	*Mc*	sub-model	*Mc* (Org./3-CV)	sub-model	*Mc* (Org./3-CV)
me2x5	0.659	P:2-2	0.629/0.631	3-C-1	0.658/0.663
me2x4	0.655	P:3-2	0.626/0.632	3-L-1	0.654/0.657
me2x3	0.653	P:4-2	0.625/0.630	2-C-1	0.647/0.657
me5s0	0.653	P:4-3	0.622/0.628	2-L-1	0.643/0.653

Model complexity, measured by parameter number, is an important issue in motif modeling. The more complex a model, the more data are needed for adequate training. For many biological motifs, however, the number of known (experimentally determined) sites is small. This limits the usage of complex models, such as higher order Markov models, Bayesian trees, network models or MEM, even though these models in some cases can perform better than the simpler models given enough training data. For a standard Markov model, the number of its parameters increases exponentially as its Markov order increases. Without sufficient training data, it is difficult to accurately estimate all model parameters, even using more robust methods (*e.g.* interpolated Markov chains [43,44]). As a result, lack of sufficient training data often makes it impractical to train a higher order Markov model. On the other hand, a low order Markov model may perform poorly by failing to incorporate more distant dependencies. Several motif models and methods have been developed to address this issue. One of these models is the variable length Markov model (VLMM), whose Markov orders (also called context lengths) can vary among different positions. VLMM can effectively reduce Markov model complexity when the variation of actual context lengths is large. VLMM, however, is not the best choice to incorporate long-range dependencies. The position optimized Markov model (POMM) [21] is able to incorporate important distant dependencies without increasing Markov chain order. However, the effectiveness of this model largely depends on the optimization routine.

More recently, Zhao *et al.* [24] described the PVLMM in an attempt to combine advantages of both VLMM and POMM. The disadvantage of PVLMM is that the number of possible permutations is the factorial of motif length, which makes it more computationally expensive. In addition, the random permutation method used by PVLMM for optimization is more likely to overfit the model, *e.g.*, incorporating non-significant dependencies into the PVLMM model that can reduce its prediction power. The optimized mixture of Markov models we presented here tries to inherit advantages of these existing models while avoiding their disadvantages. In OMiMa, we replace VLMM with a mixture of several lower order Markov models, which are subsequently optimized to account for long-range dependencies.

In comparison with other leading methods, OMiMa can incorporate more than the NNSplice's pairwise dependen-

Table 7:

TFBS	0^{th} chain	1^{st} chain
A	6	7-4-1-10-3-8-5-0-11-9-2
B	5	2-9-6-10-1-8-3-7-4-11-0

cies; OMiMa avoids model over-fitting better than the PVLMM; and OMiMa requires smaller training samples than the MEM. These are primarily reasons that OMiMa showed superior performance, in terms of prediction accuracy, required size of training data or computational time, over other leading-methods in our results.

With any model selection procedure, the possibility of choosing a model that drastically over- or underfits is a concern. OMiMa employs AIC and BIC, two standard criteria, that are widely used because they tend to avoid extreme over- or underfitting. Both have theoretical support [27,28]. In our application, neither criterion worked well when using the DF (results not shown); but both, particularly AIC, performed well when using EDF. We found that models selected by AIC using EDF were consistent with models selected by cross-validation and by ROC analysis.

Our OMiMa approach has two features that can be limitations when the size of the training data is small. First, the chi-square test that partitions motif positions into those with dependencies and those without dependencies will, like any statistical test, make mistakes, and its statistical power to detect dependencies will suffer with small training samples. Although the test will not always provide a correct partition, our approach should adapt to strong or weak dependencies overall and improve prediction when dependencies are strong. In addition, weakly dependent positions mistakenly placed in the set with no dependencies are often adequately modeled by a 0^{th} order chain, whereas independent positions mistakenly assigned to the set with dependencies will be placed by the DNJ algorithm in locations with the least impact on the k^{th} order chain. Second, the EDF that we used in model selection is an estimate based on the training data. For degenerate sites, the estimate should be accurate with even small training samples; whereas for conserved sites a larger training sample might reveal additional bases and change the EDF. Still, such additions should be minimal and would generally induce small changes in the EDF, so we expect little impact on model selection. Any methods that employ chi-square techniques to test for dependent sites face similar limitations. Nevertheless, OMiMa with its relatively small parameter space should adapt to small training datasets better than many competitors. Of course, any motif finding algorithm would do better with larger training samples.

OMiMa places no limit on the length of sequences that it can scan, and it could be used to find TFBS in any sequenced organism as long as a training motif set is available. The larger the genome evaluated, the more false positives are likely to be declared. Although OMiMa's prediction accuracy will help, other approaches to reducing false positives will be needed. Cross-species comparisons and relative location compared to transcription start sites have been used to reduce false positives and could be used with OMiMa too. Furthermore, OMiMa's ability to accurately and quickly identify splice sites should be easy to incorporate into probabilistic gene-prediction programs where correct prediction of splice sites is critical.

Conclusion

Our optimized mixture of Markov models represents an alternative to the existing methods for modeling dependent structures within a biological motif. Unlike existing methods, our model is conceptually simple and effective, which has advantages in a large scale motif prediction. In particular, with its ability to minimize model complexity, our method can work effectively even with limited training data. The optimized mixture of Markov models is implemented in our computational tool OMiMa, which can use a variety of mixture models for motif prediction. OMiMa, in which most parameters are configurable, is freely available to all users.

Authors' contributions

W. Huang provided the principal contributions to the conception and design of this study as well as to its analysis. D. M. Umbach and L. Li contributed to the design of the study and the interpretation of results. All authors contributed to writing and critically revising the manuscript.

Additional material

Additional file 1

The supplement includes the mathematical formulas for computing the probability of a motif site given a Markov model, the algorithmic pseudocode for the DNJ method, and the description of the parameter estimation for our model. It also contains supplemental materials for the main results as well as other additional results, such as the application for protein domain identification, the comparison of computational time, and so on.
Click here for file
[http://www.biomedcentral.com/content/supplementary/1471-2105-7-279-S1.pdf]

Acknowledgements

We thank Drs Bruce Weir and Jeffrey Thorne for critically reading the manuscript, and Drs Clarice Weinberg and Joseph Nevins for helpful comments. This research was supported by Intramural Research Programs of the NIH, National Institute of Environmental Health Sciences.

References

1. **Weichun Huang's Research Domain** [http://BioMedEmpire.org]
2. Burge C, Karlin S: **Prediction of complete gene structures in human genomic DNA.** *J Mol Biol* 1997, **268:**78-94.
3. Wray GA, Hahn MW, Abouheif E, Balhoff JP, Pizer M, Rockman MV, Romano LA: **The Evolution of Transcriptional Regulation in Eukaryotes.** *Mol Biol Evol* 2003, **20(9):**1377-1419.
4. Kellis M, Patterson N, Endrizzi M, Birren B, Lander ES: **Sequencing and comparison of yeast species to identify genes and regulatory elements.** *Nature* 2003, **423(6937):**241-254.
5. Negre B, Casillas S, Suzanne M, Sanchez-Herrero E, Akam M, Nefedov M, Barbadilla A, de Jong P, Ruiz A: **Conservation of regulatory sequences and gene expression patterns in the disintegrating Drosophila Hox gene complex.** *Genome Res* 2005, **15(5):**692-700.
6. Xie X, Lu J, Kulbokas EJ, Golub TR, Mootha V, Lindblad-Toh K, Lander ES, Kellis M: **Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals.** *Nature* 2005, **434(7031):**338-345.
7. Staden R: **Computer methods to locate signals in nucleic acid sequences.** *Nucl Acids Res* 1984, **12:**505-519.
8. Stormo GD, Fields DS: **Specificity, free energy and information content in protein-DNA interactions.** *Trends Biochem Sci* 1998, **23(3):**109-113.
9. Quandt K, Freeh K, Karas H, Wingender E, Werner T: **MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data.** *Nucl Acids Res* 1995, **23(23):**4878-4884.
10. Kel AE, Gössling E, Reuter I, Cheremushkin E, Kel-Margoulis OV, Wingender E: **MATCH: A tool for searching transcription factor binding sites in DNA sequences.** *Nucleic Acids Res* 2003, **31(13):**3576-3579.
11. Agarwal P, Bafna V: **Detecting non-adjoining correlations with signals in DNA.** In *RECOMB '98: Proceedings of the second annual international conference on Computational molecular biology* New York, NY, USA: ACM Press; 1998:2-8.
12. Man TK, Stormo GD: **Non-independence of Mnt repressor-operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay.** *Nucl Acids Res* 2001, **29(12):**2471-2478.
13. Benos PV, Lapedes AS, Fields DS, Stormo GD: **SAMIE: statistical algorithm for modeling interaction energies.** *Pac Symp Biocomput* 2001:115-26.
14. Bulyk ML, Johnson PLF, Church GM: **Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors.** *Nucl Acids Res* 2002, **30(5):**1255-1261.
15. Roulet E, Busso S, Camargo AA, Simpson AJG, Mermod N, Bucher P: **High-throughput SELEX SAGE method for quantitative modeling of transcription-factor binding sites.** *Nat Biotechnol* 2002, **20(8):**831-835.
16. Krivan W, Wasserman WW: **A Predictive Model for Regulatory Sequences Directing Liver-Specific Transcription.** *Genome Res* 2001:GR1806R.
17. Schneider TD, Stormo GD, Gold L, Ehrenfeucht A: **Information content of binding sites on nucleotide sequences.** *J Mol Biol* 1986, **188(3):**415-431.
18. Zhang MQ, Marr TG: **A weight array method for splicing signal analysis.** *Comput Appl Biosci* 1993, **9(5):**499-509.
19. Ponomarenko MP, Ponomarenko JV, Frolov AS, Podkolodnaya OA, Vorobyev DG, Kolchanov NA, Overton GC: **Oligonucleotide frequency matrices addressed to recognizing functional DNA sites.** *Bioinformatics* 1999, **15(7):**631-643.
20. Cai D, Delcher A, Kao B, Kasif S: **Modeling splice sites with Bayes networks.** *Bioinformatics* 2000, **16(2):**152-158.
21. Ellrott K, Yang C, Sladek FM, Jiang T: **Identifying transcription factor binding sites through Markov chain optimization.** *Bioinformatics* 2002, **18(Suppl 2):**S100-S109.
22. Barash Y, Elidan G, Friedman N, Kaplan T: **Modeling dependencies in protein-DNA binding sites.** In *RECOMB '03: Proceedings of the seventh annual international conference on Computational molecular biology* New York, NY, USA: ACM Press; 2003:28-37.
23. Yeo G, Burge CB: **Maximum entropy modeling of short sequence motifs with applications to RNA splicing signals.** *J Comput Biol* 2004, **11(2–3):**377-394.
24. Zhao X, Huang H, Speed TP: **Finding short DNA motifs using permuted markov models.** In *RECOMB '04: Proceedings of the eighth annual international conference on Computational molecular biology* New York, NY, USA: ACM Press; 2004:68-75.
25. Zhou Q, Liu JS: **Modeling within-motif dependence for transcription factor binding site predictions.** *Bioinformatics* 2004, **20(6):**909-916.
26. Saitou N, Nei M: **The neighbor-joining method: a new method for reconstructing phylogenetic trees.** *Mol Biol Evol* 1987, **4(4):**406-425.
27. Akaike H: **A new look at the statistical model identification.** *IEEE Trans Automat Control* 1974, **19(6):**716-723.
28. Schwarz G: **Estimating the dimension of a model.** *Ann Stat* 1978, **6(2):**461-464.
29. Wingender E, Chen X, Hehl R, Karas H, Liebich I, Matys V, Meinhardt T, Prüâ M, Reuter I, Schacherer F: **TRANSFAC: an integrated system for gene expression regulation.** *Nucleic Acids Res* 2000, **28:**316-319.
30. Matthews B: **Comparison of the predicted and observed secondary structure of T4 phage lysozyme.** *Biochim Biophys Acta* 1975, **405(2):**442-451.
31. Rissanen J: **Complexity of strings in the class of Markov sources.** *IEEE Trans Inform Theory* 1986, **32(4):**526-532.
32. Bühlmann P, Wyner AJ: **Variable length Markov chains.** *Ann Statist* 1999, **27(2):**480-513.
33. Reese MG, Eeckman FH, Kulp D, Haussler D: **Improved splice site detection in Genie.** *J Comput Biol* 1997, **4(3):**311-323.
34. Ketterling RP, Drost JB, Scaringe WA, Liao DZ, Liu JZ, Kasper CK, Sommer SS: **Reported in vivo splice-site mutations in the factor IX gene: severity of splicing defects and a hypothesis for predicting deleterious splice donor mutations.** *Hum Mutat* 1999, **13(3):**221-231.
35. Staley JP, Guthrie C: **An RNA switch at the 5' splice site requires ATP and the DEAD box protein Prp28p.** *Mol Cell* 1999, **3:**55-64.
36. Thanaraj T, Robinson AJ: **Prediction of exact boundaries of exons.** *Brief Bioinform* 2000, **1(4):**343-356.
37. Carmel I, Tal S, Vig I, Ast G: **Comparative analysis detects dependencies among the 5' splice-site positions.** *RNA* 2004, **10(5):**828-840.
38. **Berkeley Drosophila Genome Project** [http://www.fruitfly.org/seq_tools/datasets/Human/GENIE_96/splicesets]
39. **BDGP: Splice Site Prediction by Neural Network** [http://www.fruitfly.org/seq_tools/splice.html]
40. **Christopher Burge Lab** [http://genes.mit.edu/burgelab/maxent/ssdata]
41. Nelson K, Green M: **Mechanism for Cryptic Splice Site Activation During Pre-mRNA Splicing.** *PNAS* 1990, **87(16):**6253-6257.
42. Nandabalan K, Price L, Roeder GS: **Mutations in U1 snRNA bypass the requirement for a cell type-specific RNA splicing factor.** *Cell* 1993, **73(2):**407-415.
43. Salzberg S, Delcher A, Kasif S, White O: **Microbial gene identification using interpolated Markov models.** *Nucl Acids Res* 1998, **26(2):**544-548.
44. Ohler U, Harbeck S, Niemann H, Noth E, Reese M: **Interpolated markov chains for eukaryotic promoter recognition.** *Bioinformatics* 1999, **15(5):**362-369.
45. Crooks GE, Hon G, Chandonia JM, Brenner SE: **WebLogo: A Sequence Logo Generator.** *Genome Res* 2004, **14(6):**1188-1190.