JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 26, Number 8, 2019

Pp. 846–865
DOI: 10.1089/cmb.2018.0144

Other Research Articles

# DNA Motif Match Statistics Without Poisson Approximation

WOLFGANG KOPP* and MARTIN VINGRON



## ABSTRACT

**Transcription factors (TFs) play a crucial role in gene regulation by binding to specific regulatory sequences. The sequence motifs recognized by a TF can be described in terms of position frequency matrices. Searching for motif matches with a given position frequency matrix is achieved by employing a predefined score cutoff and subsequently counting the number of matches above this cutoff. In this article, we approximate the distribution of the number of motif matches based on a novel dynamic programming approach, which accounts for higher order sequence background (e.g., as is characteristic for CpG islands) and overlapping motif matches on both DNA strands. A comparison with our previously published compound Poisson approximation and a binomial approximation demonstrates that in particular for relaxed score thresholds, the dynamic programming approach yields more accurate results.**

**Keywords:** dynamic programming, Markov model, motif enrichment.

## 1. INTRODUCTION

Transcription factors (TFs) play an essential role in the regulation of gene expression. They function by binding to short sequences known as transcription factor binding sites (TFBSs), which are typically located in promoter or enhancer regions (Alberts et al., 2002). Based on the motif descriptions of the TFBSs, many programs search for and count occurrences of the motif matches in a sequence (Chen et al., 1995; Frith et al., 2004; Cartharius et al., 2005; Bailey et al., 2009; Roider et al., 2009; Zambelli et al., 2009; McLeay and Bailey, 2010). Since the motifs typically lack specificity, the need arises to determine the statistical significance of a motif match and thereafter to evaluate how many matches one would expect to find by chance. Relative to this information, motif enrichment can be inferred, for example, for a set of promoters (Thomas-Chollier et al., 2008).

The binding motif of a TF is frequently summarized as a position frequency matrix (PFM) (Stormo, 2000). A PFM tabulates the frequency at which a certain base has been observed at a position of a TFBS. PFMs are commonly depicted as sequence logos (Schneider and Stephens, 1990), and large numbers of known motifs are available through different databases, including TRANSFAC (Wingender et al., 1996),

Computational Molecular Biology, Max Planck Institute for Molecular Genetics, Berlin, Germany.
*Present address: Berlin Institute for Medical Systems Biology, Max Delbrueck Center for Molecular Medicine, Berlin, Germany.

JASPAR (Sandelin et al., 2004), or Hocomoco (Kulakovskiy et al., 2013). Alternatively, TFBSs may be expressed via a collection of consensus strings.

An important area of research has been to determine the distributions of the number of motif matches in a random set of DNA sequences, which is at the heart of enrichment testing. For word patterns, the distribution of match counts has been studied in great depth (for review, see Reinert et al., 2000) and can even be determined exactly based on dynamic programming (Zhang et al., 2007; Marschall and Rahmann, 2008, 2010). However, the exact solutions require the enumeration of all compatible words to derive the statistics, which is only feasible if the set of words if sufficiently small (Zhang et al., 2007).

When counting PFM matches in a sequence, first, a cutoff for the match needs to be defined. Once the threshold is chosen, one can count the number of matches and evaluate the distribution of the number of matches (Rahmann et al., 2003). Unfortunately, computing the exact match count distribution is often intractable. For this reason, efficient approximative solutions have been proposed, including the binomial distribution (Thomas-Chollier et al., 2008) or the compound Poisson distribution (Pape et al., 2008; Kopp and Vingron, 2017). The accuracy of these solutions depends on the validity of their statistical assumptions, which may not always be satisfied. For instance, the binomial model assumes independence between matches in a sequence and consequently ignores self-overlapping matches, whereas the compound Poisson approximation assumes motif matches to occur only rarely (''rare hit'' assumption).

In this article, we present a novel modeling approach to delineate the distribution of the number of PFM-based motif matches that aims to account for self-overlapping motif matches and at the same time relaxes the ''rare hit'' assumption. This approach is based on our recently proposed computation of the motif match statistics (Kopp and Vingron, 2017). First, we present a novel Markov model that describes the random process for generating motif matches. This model is instrumental for determining the probability of a *clump start match*, for example, a motif match that is not overlapped by any previous matches. A similar concept has been introduced for studying word pattern matches (Marschall and Rahmann, 2008). Second, we introduce a dynamic programming approach for computing the distribution of the number of matches, which was inspired by Liu and Lawrence (1999). Finally, we present an extension of these models for scanning motif matches on both DNA strands.

We demonstrate the accuracy of the dynamic programming approach for various parameter settings and a large set of known motifs, including a palindromic motif and a repeat-like motif, in comparison to our earlier compound Poisson model (Kopp and Vingron, 2017) and a binomial model. Generally, we find that the novel dynamic programming approach yields similar or more accurate results compared with the other models, especially when a rather relaxed match score was chosen.

## 2. METHODS

### 2.1. Motifs, background, motif score, and motif hits

Let $\mathcal{A}=\{A, C, G, T\}$ denote the alphabet of DNA letters and $\mathbf{w}=w_1w_2\cdots w_N$ a sequence of length $N$ from this alphabet. The probability of $\mathbf{w}$ is given by a homogeneous order-$d$ Markov model (the *background model*), whose transition probabilities are denoted by $\pi(w_{i-d}\cdots w_{i-1}; w_i)=P(w_i|w_{i-1}\cdots w_{i-d})$ and whose stationary distribution is denoted by $\mu$. Thus, we have

$$P_B(\mathbf{w})=\mu(w_1\cdots w_d)\prod_{i=d+1}^{N}\pi(w_{i-d}\cdots w_{i-1}; w_i).$$

The transition probabilities $\pi(a_0\cdots a_{d-1}; a_d)$ are estimated via the maximum likelihood procedure described in Reinert et al. (2000):

$$\hat{\pi}(a_0\cdots a_{d-1}; a_d)=\frac{N(a_0\cdots a_{d-1}, a_d)}{\sum_{a_d}N(a_0\cdots a_{d-1}, a_d)}, \quad (1)$$

with $N$(a) denoting the count of $\mathbf{a}\in\mathcal{A}^{d+1}$ in $\mathbf{w}\in\mathcal{A}^N$ and under the additional constraints that each word occurs equally likely on both DNA strands and with reversed nucleotide order (from 5′ to 3′ and 3′ to 5′). These constraints simplify the motif matching statistics when both DNA strands are scanned for motif matches and they are enforced by utilizing the detailed balance condition (Kopp and Vingron, 2017).

We represent the DNA binding affinity by a PFM. A PFM is a $|\mathcal{A}|\times M$ matrix, where $|\mathcal{A}|$ denotes the size of the alphabet and $M$ denotes the length of the TF binding site. A PFM contains the elements $p_j(w)$, which

correspond to the frequency of observing nucleotide $w$ at position $j$. We shall further assume that all elements of the PFM are strictly positive and its columns are normalized to 1 such that they represent probabilities. Then, the likelihood of a word $\mathbf{w}' \in \mathcal{A}^M$ with respect to the PFM is given by

$$P_M(\mathbf{w}') = \prod_{j=1}^{M} p_j(w'_j).$$

We adapt the commonly used log-likelihood ratio (Rahmann et al., 2003; Li and Tompa, 2006; Touzet et al., 2007), or motif *score*, to discriminate likely bound sequences from unbound sequences according to

$$s(\mathbf{w}') := \log\left(\frac{P_M(\mathbf{w}')}{P_B(\mathbf{w}')}\right), \tag{2}$$

where $\mathbf{w}' \in \mathcal{A}^M$ and assume that $d \leq M$ for the remainder of this article.

We leverage the motif score to determine *motif hits* (or putative TFBSs) by utilizing a predetermined *score threshold*. Position $i$ in a sequence is called a motif hit if $s(w_i \ldots w_{i+M-1})$ is greater or equal to the score threshold. According to Neyman and Pearson (1933), it is reasonable to choose a score threshold $t_\alpha$, which is associated with a desired false-positive level $\alpha$. Hence, motif hits are called with significance level $\alpha$. To choose $t_\alpha$, we determine the distribution of the scores $P_B(S = s)$ using an efficient algorithm, where we assume the underlying sequence to be generated by an order-$d$ background model starting in the stationary distribution $\mu$ as described previously (Kopp and Vingron, 2017). We obtain the *score threshold* $t_\alpha$ associated with significance level $\alpha$ from $P_B(S = s)$ by computing $P_B(S \geq t_\alpha) = \alpha$.

Scanning a DNA sequence for motif matches results in a stochastic process $\{Y_i\}_{1 \leq i \leq N-M+1}$, where $Y_i := \mathbf{1}[s(w_i \cdots w_{i+M-1}) \geq t_\alpha]$ denotes an indicator random variable that reflects a TFBS occurrence at position $i$. In case both DNA strands are scanned for motif matches, an additional set of random variables, denoted by $\{Y'_i\}_{1 \leq i \leq N-M+1}$, reflects the reverse strand matches. The total number of motif matches $X$ emitted on both DNA strands is given by

$$X = \sum_{i=1}^{N-M+1} Y_i + Y'_i.$$

If only one strand is scanned, the contribution of $Y'_i$ becomes obsolete.

### 2.2. Types of matches

Scanning a DNA sequence for binding site matches might result in self-overlapping matches, depending on the structure of the motif, which influences the distribution of the number of motif matches. To account for that, the notion of a *clump* has been introduced, which refers to one or more motif matches that are mutually overlapping (Reinert et al., 2000).

Within a clump, two distinct types of motif matches are possible: A *clump start match* and *self-overlapping matches*. Without loss of generality, we scan for motif matches from left to right. Therefore, a match $Y_i = 1$ at position $i$ starts a clump if it is not overlapped by any previous matches to its left. For example, for a motif of length $M = 3$, the sequence $Y_1 = 0, Y_2 = 0, Y_3 = 1$ constitutes a clump start at position 3. Otherwise, we observe a self-overlapping match.

*2.2.1. Motif matches when scanning a single strand.* The probability of observing a clump start is denoted by

$$\tau := P(Y_i = 1 | Y_{i-1} = 0, \cdots, Y_{i-M+1} = 0). \tag{3}$$

The computation of the clump start probability shall be deferred to Section 2.3. We define the probability of a self-overlapping match by

$$\beta_k := P(Y_k = 1, Y_{k-1} = 0, \cdots Y_1 = 0 | Y_0 = 1) \tag{4}$$

for $k \in \{1, \cdots M-1\}$, which we efficiently approximate using our earlier approach (Kopp and Vingron, 2017).

*2.2.2. Motif matches when scanning both DNA strands.* In many applications, we do not know in advance on which DNA strand a TFBS might occur, which is solved by simply scanning both strands. This in turn also creates a coupling between matches on both strands that needs to be addressed. In the following, without loss of generality, we consider the ordering of events: $Y_1 Y'_1 Y_2 Y'_2 Y_3 Y'_3 \cdots$. That is, we scan the sequence from left to right, and a forward strand event $Y_i$ precedes the corresponding reverse strand event $Y'_i$ at position *i*.

A clump starts on the forward strand if matches at the $M-1$ previous positions (on both strands) are absent. Its probability is defined by

$$\tau := P(Y_i = 1 | \{Y_{i-m} = 0, Y'_{i-m} = 0\}_{i-M<m<i}). \tag{5}$$

Likewise, a clump starts on the reverse strand $Y'_i = 1$ if additionally $Y_i = 0$, in which case the probability is given by

$$\tau' := P(Y'_i = 1, Y_i = 0 | \{Y_{i-m} = 0, Y'_{i-m} = 0\}_{i-M<m<i}). \tag{6}$$

When scanning both DNA strands, overlapping motif matches might occur in three different configurations: (1) Hits might overlap on the same strand, (2) a forward strand hit $Y_0 = 1$ precedes a reverse strand hit $Y'_k = 1$ for $0 \leq k < M$, and (3) a reverse strand hit $Y'_0 = 1$ precedes a forward strand hit $Y_k = 1$ for $0 < k < M$, where $k$ denotes the shift between the positions (Fig. 1).

The associated probabilities with the overlapping match configurations are defined by

$$\beta_k := P(Y_k = 1, \{Y_j = 0, Y'_j = 0\}_{1 \leq j < k}, Y'_0 = 0 | Y_0 = 1) \tag{7}$$

$$\beta_{3',k} := P(Y'_k = 1, \{Y_j = 0\}_{1 \leq j \leq k}, \{Y'_j = 0\}_{0 \leq j < k} | Y_0 = 1) \tag{8}$$

$$\beta_{3',0} := P(Y'_0 = 1, | Y_0 = 1) \tag{9}$$

$$\beta_{5',k} := P(Y_k = 1, \{Y_j = 0, Y'_j = 0\}_{1 \leq j < k} | Y'_0 = 1). \tag{10}$$

We approximate them by using the procedure discussed in Kopp and Vingron (2017).

## 2.3. A Markov model for generating motif matches

In this section, we introduce a Markov model that resembles the process of generating motif matches as one DNA strand is scanned for match occurrences. We shall exploit this model later to determine the *clump start probability*, which constitutes an important prerequisite for Section 2.4.

*2.3.1. Model states, state transitions, and transition probabilities.* Before establishing the Markov model states, two independence assumptions are required: First, for $i + M - 1 < j$, the events $Y_i$ and $Y_j$ are assumed to be statistically independent because they are nonoverlapping. Second, a motif match $Y_i = 1$ at position $i$ renders the events upstream and downstream independent, for example, $Y_{i-1}$ is independent of $Y_{i+1}$ given $Y_i = 1$. We justify them in Section 5.

Next, we define the Markov model and use it to express the realizations of $Y_1 Y_2 Y_3 \cdots$ in terms of the states and state transition. We shall use the Markov model to analyze the stationary distribution of the model, which allows us to evaluate the unknown probability of a *clump start match* $\tau$.

The state space of the Markov model is established through a correspondence relationship with match patterns in $Y_1 Y_2 \cdots$. Accordingly, a motif of length $M$ results in $M$ states due to the assumption that nonoverlapping positions are assumed to be independent:

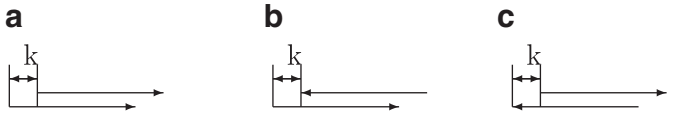


**FIG. 1.** Three types of overlapping hit with a shift of $k$ between the motif starts. **(a)**, **(b)**, and **(c)** correspond to matches on the same strand, a forward strand match followed by a reverse strand match and a reverse strand match followed by a forward strand match, respectively. The arrows pointing to the right and left represent the $(5' \rightarrow 3')$ and $(3' \leftarrow 5')$ directions of the DNA, respectively.

$$\begin{array}{lcllllll}
n & \widehat{=} & (Y_{i-M+2}=0, & Y_{i-M+3}=0, & \cdots, & Y_{i-2}=0, & Y_{i-1}=0, & Y_i=0) \\
h & \widehat{=} & (Y_{i-M+2}=\bullet, & Y_{i-M+3}=\bullet, & \cdots, & Y_{i-2}=\bullet, & Y_{i-1}=\bullet, & Y_i=1) \\
n_1 & \widehat{=} & (Y_{i-M+2}=\bullet, & Y_{i-M+3}=\bullet, & \cdots, & Y_{i-2}=\bullet, & Y_{i-1}=1, & Y_i=0) \\
n_2 & & (Y_{i-M+2}=\bullet, & Y_{i-M+3}=\bullet, & \cdots, & Y_{i-2}=1, & Y_{i-1}=0, & Y_i=0) \\
\vdots & \widehat{=} & \vdots & & & & & \\
n_{M-3} & \widehat{=} & (Y_{i-M+2}=\bullet, & Y_{i-M+3}=1, & \cdots, & Y_{i-2}=0, & Y_{i-1}=0, & Y_i=0) \\
n_{M-2} & \widehat{=} & (Y_{i-M+2}=1, & Y_{i-M+3}=0, & \cdots, & Y_{i-2}=0, & Y_{i-1}=0, & Y_i=0)
\end{array}, \tag{11}$$

where "•" represents any outcome (zero or one) at the respective position. As the motif is shifted along the sequence, the Markov chain switches from one state to another, determined by the match pattern of the motif at a position. State $h$ represents a motif match at a current position $i$ regardless of the previous events. $n_k$ denotes a memory state that indicates that the last match occurred $k$ positions upstream of the current position and $n$ denotes the absence of motif matches for $M-1$ consecutive position (including the current position).

Traversing the sequence $Y_1Y_2Y_3\cdots$ in an ordered manner imposes a restriction on the possible state transitions (Fig. 2). For example, the transition $(Z_i=n_2)\rightarrow(Z_{i+1}=n_1)$ is not viable, whereas $(Z_i=n)\rightarrow(Z_{i+1}=h)$ is. This in turn results in an $M\times M$ transition matrix:

$$\begin{bmatrix}
P(n\rightarrow n) & 0 & 0 & & 0 & P(n_{M-2}\rightarrow n) \\
P(n\rightarrow h) & P(h\rightarrow h) & P(n_1\rightarrow h) & \cdots & P(n_{M-3}\rightarrow h) & P(n_{M-2}\rightarrow h) \\
0 & P(h\rightarrow n_1) & 0 & & 0 & 0 \\
0 & 0 & P(n_1\rightarrow n_2) & & 0 & 0 \\
0 & 0 & 0 & & 0 & 0 \\
\vdots & & & \ddots & & \vdots \\
0 & 0 & 0 & \cdots & P(n_{M-3}\rightarrow n_{M-2}) & 0
\end{bmatrix}, \tag{12}$$

whose individual transition probabilities are derived from Equations (3) and (4) as

$$P(n\rightarrow n) := 1-\tau \tag{13}$$

$$P(n\rightarrow h) := \tau \tag{14}$$

$$P(h\rightarrow h) := \beta_1 \tag{15}$$

$$P(h\rightarrow n_1) := 1-\beta_1 \tag{16}$$

$$\begin{aligned}
P(n_k\rightarrow h) &:= P(Y_0=1|Y_{-1}=0,\ \cdots, Y_{-k+1}=0, Y_{-k}=1) \\
&= \frac{P(Y_k=1, Y_{k-1}=0,\ \cdots, Y_1=0|Y_0=1)}{P(Y_{k-1}=0,\ \cdots, Y_1=0|Y_0=1)} \\
&= \frac{\beta_k}{1-\sum_{i=1}^{k-1}\beta_i} \qquad \text{for } 1\le k\le M-2
\end{aligned} \tag{17}$$

$$P(n_k\rightarrow n_{k+1}) := 1-P(n_k\rightarrow h) \qquad \text{for } 1\le k< M-2 \tag{18}$$

$$P(n_{M-2}\rightarrow n) := 1-P(n_{M-2}\rightarrow h). \tag{19}$$

*2.3.2. Computing the clump start probability $\tau$.* In the previous section, we have established the states and state transitions for the Markov model. We expressed the transition probabilities solely based on

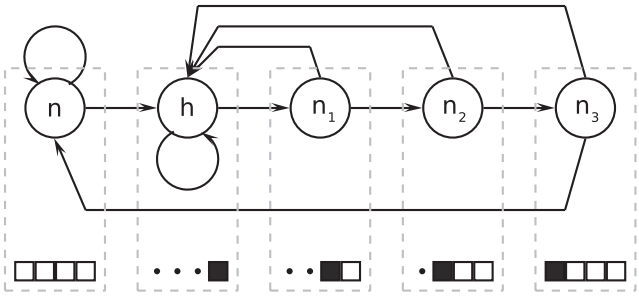


**FIG. 2.** Illustration of the Markov model. The nodes denote the states of the model using a TF motif of length $M=5$. Arrows indicate viable state transitions, which may (or may not) be associated with a positive transition probabilities. Underneath each node, the associated pattern in $Y_1Y_2\cdots$ is depicted, where the black and white boxes denote the outcomes *one* and *zero* and the bullet represents *any* outcome (*zero* or *one*) that are described by the Correspondences [Equation (11)]. TF, transcription factor.

$\tau$ and $\{\beta_k\}_{1\leq k\leq M-1}$. While $\{\beta_k\}_{1\leq k\leq M-1}$ can be efficiently approximated, the computation of $\tau$ remains to be discussed.

Recall that due to the choice of the score threshold $t_\alpha$, motif matches occur with probability $P(Y=1)=\alpha$. This implies that the stationary distribution of the Markov model should visit the state $Z=h$ also with probability $\mu(Z=h)=\alpha$, where $\mu$ denotes the stationary probability. Thus, we introduce an optimization procedure whose objective is to establish $\mu(h;\tau)=\alpha$ with respect to the unknown parameter $\tau$. We utilize the objective function:

$$f(\mu(\tau)) := -\alpha\log(\mu(h;\tau))-(1-\alpha)\log(1-\mu(h;\tau)), \tag{20}$$

which has a unique minimum at $\mu(h;\tau)=\alpha$ as can be verified by studying $df(\mu)/d\mu=0$. This function is motivated by the cross-entropy error, which is widely used to fit supervised statistical classifiers, for example, the logistic regression model (Bishop, 2006). While the objective $(\mu(h;\tau)-\alpha)^2$ would be another (perhaps more intuitive) possibility, it may result in numerical issues due to the fact that $\alpha$ is usually rather small. In contrast, the chosen objective function [Equation (20)] is slightly better behaved in this respect. Furthermore, other objective functions may be envisioned as well, so long as they have a minimum at $\mu(h;\tau)=\alpha$. But since we have not noticed any numerical issues with the current choice (as by the experiments presented in this work), we believe that the current objective works well in practice.

We minimize Equation (20) iteratively starting from $\tau_0=\alpha$ using conjugate gradients. In each iteration, the stationary distribution of the Markov model is computed with respect to the current $\tau$ using the power method (Karlin and Taylor, 1981).

### 2.4. Computing the distribution of the number of matches

In this section, we develop an algorithm for computing the distribution of the number of motif matches based on the previously determined overlapping match probabilities $\{\beta_i\}_{1\leq i<M}$ and the clump start probability $\tau$. The algorithm was inspired by Liu and Lawrence (1999), which allows to efficiently sum the probabilities of all possible permutations of placing $X$ motif matches in a sequence of fixed length $N$ via dynamic programming. For example, to compute the probability of placing two matches in $Y_1..Y_3$, we would have to determine $P(Y_1..Y_3=110)+P(Y_1..Y_3=101)+P(Y_1..Y_3=011)$. In the general case, this amounts to summing over $\binom{N}{x}$ individual permutations. While dynamic programming has been proposed for studying word-pattern-based motifs (Zhang et al., 2007), we are not aware of a comparable approach for studying PFM-based motifs directly.

We start by discussing the case where overlapping motif matches are ignored. We denote the indices associated with the events $Y_1\cdots Y_i$ by $[1:i]$. The number of motif matches in that segment is denoted by $X_{[1:i]}$ and its probability to exhibit $x$ matches is denoted by $P(X_{[1:i]}=x)$.

According to equation (18) in Liu and Lawrence (1999), given that $P(X_{[1:i]}=x)$ has already been computed, $P(X_{[1:j]}=x+1)$ can be established recursively by

$$P(X_{[1:j]}=x+1)=\sum_{i<j}P(X_{[1:i]}=x)P(Y_{i+1}=1,X_{[i+2:j]}=0), \tag{21}$$

provided that neighboring events in $\{Y_i\}$ are considered independent. As a consequence, the resulting distribution of the number of matches would be identical to a binomial distribution.

At this point, we would like to convey the intuition behind Equation (21) via an example as it is of fundamental importance for the dynamic programming algorithm that we introduce below. Consider the probability of observing two matches in the sequence $Y_1\ldots Y_4$. That is $P(X_{[1:4]}=2)$. Since the example is small enough, it is illustrative to enumerate the permutations:

$$\begin{aligned}P(X_{[1:4]}=2)=&P(Y_1..Y_4=0011)+\\&P(Y_1..Y_4=0101)+\\&P(Y_1..Y_4=1001)+\\&P(Y_1..Y_4=0110)+\\&P(Y_1..Y_4=1010)+\\&P(Y_1..Y_4=1100).\end{aligned}$$

By ordering these permutations as shown above, we can see that the right-hand side of Equation (21) is obtained

$$P(X_{[1:1]}=1)P(Y_2=1, X_{[3:4]}=0)=P(1100)$$
$$P(X_{[1:2]}=1)P(Y_3=1, X_{[4:4]}=0)=P(1010)+P(0110)$$
$$P(X_{[1:3]}=1)P(Y_4=1)=P(1001)+P(0101)+P(0011).$$

The generalization of this example yields Equation (21).

Next, we adapt Equation (21) to account for self-overlapping matches. This requires to memorize at which position in the segment $[1:i]$ the last match occurred because this influences the probability of a match at $i+1$. Toward this end, we introduce the number of matches $X^a_{[1:i]}$ in the segment $[1:i]$ with an additional index $a$ indicating the location of the last match in that segment. For $1 \leq a < M$, the last match occurred at position $i-a+1$, and for $a=M$, the last match occurred at $M$ or more positions upstream of $i+1$. In other words, $1 \leq a < M$ describes self-overlapping matches, whereas $a=M$ results in a clump start match. Note also that all nonoverlapping upstream positions can be aggregated due to the assumption that nonoverlapping events are assumed to occur independently. The recursive definition for $P(X^a_{[1:j]}=x+1)$ now becomes

$$P(X^a_{[1:j]}=x+1)=\sum_{i<j}\sum_{b\leq M} P(X^b_{[1:i]}=x)\times h(b)\times z(j-i), \tag{22}$$

where we make use of Equations (3) and (4) to define

$$a := \begin{cases} j-i & \text{if } j-i < M \\ M & \text{o.w.} \end{cases} \tag{23}$$

$$h(b) := \begin{cases} \beta_b & \text{if } b < M \\ \tau & \text{o.w.} \end{cases} \tag{24}$$

$$z(b) := \begin{cases} 1 & \text{if } b < M \\ (1-\tau)^{b-M} \cdot \sum_{i<M} 1-\beta_i & \text{o.w.} \end{cases} \tag{25}$$

The purpose of the auxiliary function $h(.)$ is to incorporate one more match at position $i+1$, which can happen through an overlapping match or a clump start match. However, $z(.)$ accounts for the absence of additional matches in the segment $[i+2, j]$, where we defer the incorporation of the absence of matches for the case $b < M$ to the termination step of the algorithm.

For convenience, we define

$$\delta_i := 1-\sum_{j=1}^{i}\beta_j. \tag{26}$$

We initialize the procedure for $P(X^a_{[1:j]}=1)$ according to

$$P(X^a_{[1:j]}=1)=\begin{cases} (j-M+1)\times(1-\tau)^{j-M}\tau\delta_{M-1} & \text{for } a=M \\ (1-\tau)^{j-a}\tau & \text{ow.} \end{cases} \tag{27}$$

for $1 \leq a \leq M$ and $1 \leq j \leq N-M+1$.

Then, we evaluate Equation (22) for $x=2$ to the maximal number of matches to be considered $x_{\max}$. Finally, the algorithm terminates by

$$P(X_{[1:N-M+1]}=x)=P(X^M_{[1:N-M+1]}=x)+\sum_{a=1}^{M-1} P(X^a_{[1:N-M+1]}=x)\delta_{a-1}. \tag{28}$$

Together with the fact that $P(X_{[1:j]}=0)=(1-\tau)^j$, this establishes the distribution of the number of matches $P(X_{[1:N-M+1]}=x)$.

We have also developed an extension of the Markov model and dynamic programming procedure for the case of studying the number of motif matches on both DNA strands. Since they are based on similar considerations, we relegate their description to Appendix.

*2.4.1. Runtime.* The asymptotic runtime of the dynamic programming algorithm is given by $O(x_{\max}(N-M+1)^2M)$, where $x_{\max}$ denotes the maximum number of hits after which the distribution is truncated and $N$ denotes the length of the DNA sequence and $M$ denotes the length of the TF motif.

Typical values for $N$, $M$, and $x_{\max}$ are $N=200$, $M=15$, and $x_{\max}=30$. Thus, since $N \gg M$ and $N \gg x_{\max}$, in practice, the dominant factor of the runtime, stems from the square of the DNA sequence length $N$.

*2.4.2. Analyzing multiple distinct DNA sequences.* In many cases, it is of interest to determine the distribution of the number of motif hits across $S$ distinct pieces of DNA, for instance, across a set of promoter regions.

Let us assume that the $S$ individual sequences are of equal length $N$ and disjoint. In this case, we need to compute $P(X_{[1:N-M+1]})$ only once. Subsequently, we determine the distribution of the sum of matches across the $S$ sequences by employing the convolution operation recursively, using a divide-and-conquer strategy. This leads to a runtime of $O(x_{\max}\log(S))$.



## 3. EVALUATION METHODOLOGY

### 3.1. Comparison between methods

We estimated background models of various orders from a subset of Dnase-I hypersensitive sites published by the ENCODE consortium (Thurman et al., 2012) as such sequences are frequently under scrutiny when it comes to searching for motif matches.

For the experiments, we focus on evaluating the more interesting case of counting motif matches on both DNA strands and compare the dynamic programming approach $P_{\mathrm{DP}}$, our recently proposed compound Poisson approximation $P_{\mathrm{CP}}$ (Kopp and Vingron, 2017), and the binomial model $P_{\mathrm{Bin}}$, which is defined by

$$P_{\mathrm{Bin}}(X=x)=\binom{2\times(N-M+1)}{x}\alpha^x(1-\alpha)^{2\times(N-M+1)-x}.$$

We compare (1) different sequence lengths, (2) different false-positive probabilities $\alpha$ of obtaining a motif hit, (3) different background model orders $d$, and (4) various motifs (Fig. 3a–c). A summary of the setup is given in Table 1.

As a reference for the analysis, we determined an empirical distribution $P_{\mathrm{E}}$ by sampling 10,000 random DNA sequences of lengths given in Table 1 from the background models and counted the number of respective motif hits, which resulted in a highly reproducible empirical distribution.

We measure the quality of the analytic approximations by evaluating the total variation distance relative to $P_{\mathrm{E}}$

$$d(P_{\mathrm{E}}, Q) := \sum_{x\geq 0}|P_{\mathrm{E}}(x)-Q(x)|, \tag{29}$$

where $Q$ denotes a placeholder for $P_{\mathrm{DP}}$, $P_{\mathrm{CP}}$, and $P_{\mathrm{Bin}}$. The smaller $d(P_{\mathrm{E}}, Q)$, the more accurate the approximation is. Additionally, we measure the discrepancy on the 5% significance region only

$$d_{5\%}(P_{\mathrm{E}}, Q) := \sum_{x\geq q_{95\%}}|P_{\mathrm{E}}(x)-Q(x)|. \tag{30}$$

where $q_{95\%}$ denotes the 95% percentile of $P_{\mathrm{E}}$.

### 3.2. Comparison of the models on JASPAR motifs

We compared $P_{\mathrm{DP}}$, $P_{\mathrm{CP}}$, and $P_{\mathrm{Bin}}$ on all JASPAR core motifs with a minimum length of 6 bps contained in the *MotifDb* Bioconductor package (444 motifs in total). An order-1 background model was obtained from ENCODE Dnase-I hypersensitive sites as described above. The distribution was determined for scanning $10\times100$ bp sequences using $\alpha=0.01$ as well as for scanning $100\times100$ bp sequences using $\alpha=0.001$. As a reference, we determined the sampling-based distribution $P_{\mathrm{E}}$. To assess

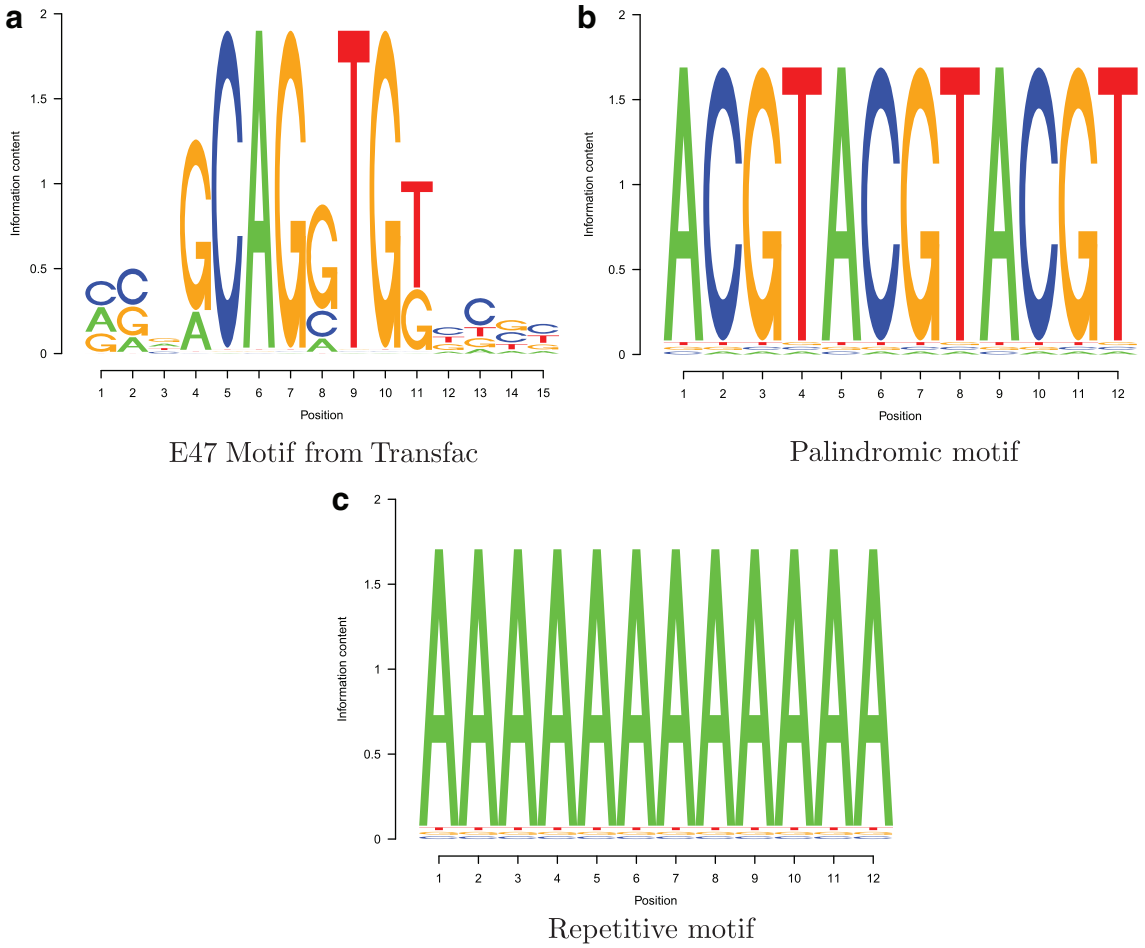


**FIG. 3.** DNA motifs.

how $P_{DP}$ compares with the other models, we determined the differences between the total variations according to

$$\Delta d_{DP-CP} := d(P_{DP}, P_E) - d(P_{CP}, P_E) \tag{31}$$

$$\Delta d_{DP-Bin} := d(P_{DP}, P_E) - d(P_{Bin}, P_E) \tag{32}$$

for each motif where negative values $\Delta d_{DP-CP}$ and $\Delta d_{DP-Bin}$ indicate that the dynamic programming approach yields more accurate results, whereas positive values suggest the opposite.

TABLE 1. PARAMETER CHOICES FOR THE COMPARATIVE ANALYSIS

| d | α | *seqlen, bp* |
|---|---|---|
| 0 | 0.01 | 50×100 |
| 0 | 0.01 | 10×500 |
| 0 | 0.001 | 500×100 |
| 0 | 0.001 | 100×500 |
| 1 | 0.01 | 50×100 |
| 1 | 0.01 | 10×500 |
| 1 | 0.001 | 500×100 |
| 1 | 0.001 | 100×500 |

## 4. RESULTS

### 4.1. Comparison between models

In this section, we systematically compare $P_{DP}$, $P_{CP}$, and $P_{Bin}$ for a range of parameter settings and motifs and focusing on scanning both DNA strands.

Since the motif structure influences the shape of the distribution of the number of motif matches, we selected a nonself-overlapping motif (Fig. 3a) and two self-overlapping motifs (a palindrome motif and a repeat-like motif; Fig. 3b, c) for the model comparison.

While the binomial distribution generally yields accurate results for nonoverlapping motifs (Fig. 4a), it is not suitable for modeling the distribution of the number of matches for self-overlapping motifs (Fig. 4b, c). However, both the dynamic programming approach $P_{DP}$ and the compound Poisson model $P_{CP}$ are



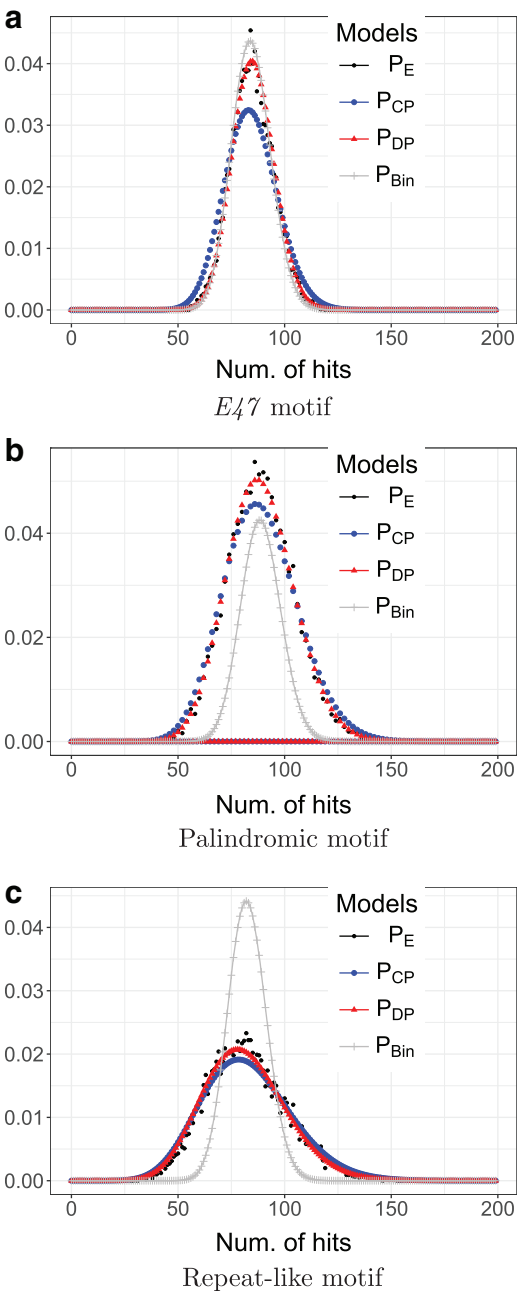


**FIG. 4.** Distribution of the number of motif matches for the motifs depicted in Figure 3. The distributions were computed for $\alpha = 0.01$ and background order $d = 1$ using 50 sequences of length 100 bp (see Table 1).

TABLE 2. TOTAL VARIATION DISTANCES DEFINED BY EQUATION (29) BETWEEN $P_E$, $P_{DP}$, $P_{CP}$, AND $P_{Bin}$ FOR E47 (FIG. 3A)

| d | α | *seqlen* | $d(P_E, P_{DP})$ | $d(P_E, P_{CP})$ | $d(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.0801** | 0.218 | 0.106 |
| 1 | 0.01 | 100 | **0.0661** | 0.218 | 0.0998 |
| 0 | 0.01 | 500 | 0.0926 | 0.227 | **0.0891** |
| 1 | 0.01 | 500 | **0.0627** | 0.217 | 0.101 |
| 0 | 0.001 | 100 | **0.0545** | 0.0574 | 0.0712 |
| 1 | 0.001 | 100 | **0.066** | 0.0676 | 0.0833 |
| 0 | 0.001 | 500 | 0.0544 | **0.0528** | 0.0707 |
| 1 | 0.001 | 500 | **0.071** | 0.0736 | 0.0868 |

Bold values mark the most accurate result in each row.



applicable for nonself-overlapping and self-overlapping motif matches, as they both account for the structural properties of the motifs (Fig. 4a–c).

As described previously (Reinert et al., 2000), the compound Poisson model rests on the ''rare hit'' assumption, which requires a sufficiently stringent choice for α. For instance, for $\alpha = 0.001$, the compound Poisson model yields accurate results regardless of the motif structure (Tables 2–4), whereas for a relaxed choice of α, the ''rare hit'' assumption becomes violated, which causes biases. This is evident for the nonself-overlapping motif (Fig. 3a) and $\alpha = 0.01$ where the compound Poisson model overestimates the variance of the distribution and results in a less accurate approximation compared with the binomial model (Table 2).

In comparison, the dynamic programming approach does not explicitly rest on the ''rare hit'' assumption; thus, it yields similar or more accurate results especially for a relaxed choice of $\alpha = 0.01$ compared with the other models. For example, for the nonself-overlapping motif and $\alpha = 0.01$, $P_{DP}$ yields similar results to $P_{Bin}$ and more accurate results than $P_{CP}$ (Table 2), whereas for self-overlapping motifs, $P_{DP}$ yields more accurate results compared with both other model types (Tables 3 and 4).

Inspecting the performance of the dynamic programming approach at fairly stringent significance level $\alpha = 0.001$, we observe that for nonself-overlapping motifs, all three approximations ($P_{DP}$, $P_{CP}$, and $P_{Bin}$) yield comparably accurate results (Table 2), whereas for self-overlapping motifs, only $P_{DP}$ and $P_{CP}$ seem to be adequate and yield similar results (Tables 3 and 4).

Next, to exclude biases due to variations of the background model and sequence length, we varied the background model order ($d \in \{0, 1\}$) and the individual sequence length (100 or 500 bp). We observe a similar relationship between the accuracies of $P_{DP}$, $P_{CP}$, and $P_{Bin}$, regardless of variations of the background model order and sequence length (Tables 2–4). In other words, the dominant effect on the model accuracies is determined by the choice of α and the model's ability to account for self-overlapping matches, rather than $d$ and the sequence length.

TABLE 3. TOTAL VARIATION DISTANCES DEFINED BY EQUATION (29) BETWEEN $P_E$, $P_{DP}$, $P_{CP}$, AND $P_{Bin}$ FOR THE PALINDROME MOTIF (FIG. 3B)

| d | α | *seqlen* | $d(P_E, P_{DP})$ | $d(P_E, P_{CP})$ | $d(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.0487** | 0.103 | 1 |
| 1 | 0.01 | 100 | **0.0487** | 0.11 | 1 |
| 0 | 0.01 | 500 | **0.0573** | 0.103 | 1 |
| 1 | 0.01 | 500 | **0.056** | 0.105 | 1 |
| 0 | 0.001 | 100 | **0.0446** | 0.047 | 1 |
| 1 | 0.001 | 100 | 0.051 | **0.0504** | 1 |
| 0 | 0.001 | 500 | 0.0396 | **0.0386** | 1 |
| 1 | 0.001 | 500 | **0.0595** | 0.0605 | 1 |

Bold values mark the most accurate result in each row.

TABLE 4. TOTAL VARIATION DISTANCES DEFINED BY EQUATION (29) BETWEEN $P_E$, $P_{DP}$, $P_{CP}$, AND $P_{Bin}$ FOR THE REPEAT-LIKE MOTIF (FIG. 3C)

| d | α | *seqlen* | $d(P_E, P_{DP})$ | $d(P_E, P_{CP})$ | $d(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.103** | 0.115 | 0.718 |
| 1 | 0.01 | 100 | **0.0964** | 0.117 | 0.714 |
| 0 | 0.01 | 500 | **0.0945** | 0.115 | 0.727 |
| 1 | 0.01 | 500 | **0.0743** | 0.103 | 0.712 |
| 0 | 0.001 | 100 | 0.0952 | **0.0694** | 0.587 |
| 1 | 0.001 | 100 | 0.107 | **0.0808** | 0.591 |
| 0 | 0.001 | 500 | 0.0623 | **0.0597** | 0.589 |
| 1 | 0.001 | 500 | 0.0849 | **0.0844** | 0.595 |

Bold values mark the most accurate result in each row.



Finally, we explicitly compared the total variation on the right tail of the distribution of the number of matches using Equation (30), since this regime is of relevance for motif match enrichment testing. In particular, we chose to compare the discrepancy on the 5% percentile of $P_E$ since these regions can be accurately and highly reproducibly approximated via sampling in a reasonable time. The discrepancies measured by Equations (29) and (30) produce highly concordant results (Tables 2–7). That is, there are no cases where the discrepancy measured by Equations (29) and (30) substantially disagrees.

### *4.2. Performance evaluation on JASPAR motif*

In the previous section, we have studied the accuracies $P_{DP}$, $P_{CP}$, and $P_{Bin}$ relative to $P_E$ in depth for three selected motifs. In this section, we assess the adequacy of the approximations for a large set of known motifs from JASPAR (Sandelin et al., 2004). Accordingly, for each motif, we determine the difference in total variation between $P_{DP}$ and the alternatives, $P_{CP}$ and $P_{Bin}$, as defined by Equations (31) and (32), respectively. We asked whether the distribution of $\Delta d_{DP-Bin}$ and $\Delta d_{DP-CP}$ over all JASPAR motifs is biased toward negative values, which would indicate that $P_{DP}$ is more accurate compared with the alternatives. To this end, we conducted the Wilcoxon rank sum test using the null hypotheses median $\Delta d_{DP-Bin} = 0$ and $\Delta d_{DP-Bin} = 0$, respectively. We observe that for a relaxed score cutoff with $\alpha = 0.01$, the dynamic programming approach significantly outperforms the binomial and the compound Poisson approximation (Table 8), suggesting that it generally yields more accurate results for known motifs in this regime. Furthermore, we observe that for a stringent cutoff with $\alpha = 0.001$, the dynamic programming approach also establishes the most accurate results. However, the differences are less pronounced, especially for $\Delta d_{DP-CP}$ (Table 8). This suggests that for stringent cutoff the compound Poisson approximation and the dynamic programming algorithm yield similar results.

TABLE 5. TOTAL VARIATION DISTANCES ON THE 5% TILE DEFINED BY EQUATION (30) BETWEEN $P_E$, $P_{DP}$, $P_{CP}$, AND $P_{Bin}$ FOR E47 (FIG. 3A)

| d | α | *seqlen* | $d_{5\%}(P_E, P_{DP})$ | $d_{5\%}(P_E, P_{CP})$ | $d_{5\%}(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.0103** | 0.0447 | 0.0139 |
| 1 | 0.01 | 100 | **0.0113** | 0.0407 | 0.0161 |
| 0 | 0.01 | 500 | **0.0113** | 0.0437 | 0.0157 |
| 1 | 0.01 | 500 | **0.00674** | 0.0339 | 0.0192 |
| 0 | 0.001 | 100 | **0.00684** | 0.00691 | 0.0103 |
| 1 | 0.001 | 100 | 0.0111 | **0.0083** | 0.0171 |
| 0 | 0.001 | 500 | **0.00921** | 0.00992 | 0.0104 |
| 1 | 0.001 | 500 | 0.0118 | **0.00924** | 0.0181 |

Bold values mark the most accurate result in each row.

Table 6. Total Variation Distances on the 5% Tile Defined by Equation (30) Between $P_E$, $P_{DP}$, $P_{CP}$, and $P_{Bin}$ for the Palindrome Motif (Fig. 3B)

| d | α | *seqlen* | $d_{5\%}(P_E, P_{DP})$ | $d_{5\%}(P_E, P_{CP})$ | $d_{5\%}(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.00675** | 0.0159 | 0.0527 |
| 1 | 0.01 | 100 | **0.00613** | 0.022 | 0.0523 |
| 0 | 0.01 | 500 | **0.0104** | 0.019 | 0.0607 |
| 1 | 0.01 | 500 | **0.0114** | 0.0205 | 0.054 |
| 0 | 0.001 | 100 | **0.00454** | 0.00595 | 0.0533 |
| 1 | 0.001 | 100 | **0.00661** | 0.00736 | 0.061 |
| 0 | 0.001 | 500 | 0.00801 | **0.00786** | 0.0631 |
| 1 | 0.001 | 500 | **0.0101** | 0.0106 | 0.0536 |

Bold values mark the most accurate result in each row.



## 5. DISCUSSION

In this article, we have introduced a novel statistical model and a dynamic programming algorithm that are jointly used to approximate the distribution of the number of motif matches. First, we have described the Markov model that we have employed to determine the previously unknown probability of a *clump start match*. Second, using the clump start and the overlapping match probabilities, we have derived a dynamic programming algorithm for determining the distribution of the number of matches in a finite-length sequence. Furthermore, we have also developed an extension that accounts for overlapping motif matches on both DNA strands simultaneously.

The Markov model and the dynamic programming algorithm are tightly linked, which can be appreciated by the fact that both are completely determined by the clump start and overlapping match probabilities. Furthermore, we illustrate that the dynamic programming algorithm can be rearranged such that it makes use of the transition probabilities of the Markov model in Appendix A.

Focusing on the more interesting case of studying motif matches on both DNA strands, we have systematically compared the accuracy of novel dynamic programming approach with a binomial model and compound Poisson approximation (Kopp and Vingron, 2017).

Our results suggest that the dynamic programming approach yields more accurate results if relaxed significance levels of $\alpha$ are considered (e.g., $\alpha = 0.01$) for nonself-overlapping as well as for self-overlapping motifs. This can be explained due to the relaxation of the ''rare hit'' assumption (also known as Poisson assumption) and by accounting for self-similarity of a motif at the same time. For stringent choices of $\alpha$ (e.g., $\alpha = 0.001$), however, we find that the compound Poisson and the dynamic programming approach generally yield highly concordant results, regardless of the motif structures.

Our approach makes use of two simplifying assumptions: First, events at nonoverlapping positions are considered independent, and second, events separated by a match are independent. The first assumption holds exactly for order-zero background models, whereas for higher-order background models, it amounts to an simplifying assumption. However, since nonoverlapping events are coupled only very weakly, this assumption

Table 7. Total Variation Distances on the 5% Tile Defined by Equation (30) Between $P_E$, $P_{DP}$, $P_{CP}$, and $P_{Bin}$ for the Repeat-Like Motif (Fig. 3C)

| d | α | *seqlen* | $d_{5\%}(P_E, P_{DP})$ | $d_{5\%}(P_E, P_{CP})$ | $d_{5\%}(P_E, P_{Bin})$ |
|---|---|---|---|---|---|
| 0 | 0.01 | 100 | **0.013** | 0.0187 | 0.0547 |
| 1 | 0.01 | 100 | **0.0105** | 0.0199 | 0.0513 |
| 0 | 0.01 | 500 | **0.0117** | 0.0155 | 0.0519 |
| 1 | 0.01 | 500 | **0.0135** | 0.0177 | 0.0511 |
| 0 | 0.001 | 100 | 0.0148 | **0.0107** | 0.0567 |
| 1 | 0.001 | 100 | 0.0115 | **0.00941** | 0.0495 |
| 0 | 0.001 | 500 | 0.00866 | **0.00817** | 0.0509 |
| 1 | 0.001 | 500 | 0.00889 | **0.00885** | 0.0496 |

Bold values mark the most accurate result in each row.

Table 8. Median Difference Between Total Variation Distances Across All JASPAR Motifs

| α | *Median* $\Delta d_{DP-Bin}$ (*p-value*) | *Median* $\Delta d_{DP-CP}$ (*p-value*) |
|---|---|---|
| 0.01 | –0.034 ($<10^{-16}$) | –0.045 ($<10^{-16}$) |
| 0.001 | –0.0089 ($<10^{-16}$) | –0.00064 (0.0005) |

*p*-Values were determined using the Wilcoxon rank sum test.

is usually justified even for higher order background models. Furthermore, this assumption allows to limit the state space of the Markov model to $M$ states and influences the runtime of the dynamic algorithm. The second assumption is simultaneously affected by the background model order and the choice for the score cutoff $t_\alpha$.

Regarding the background model, the assumption is compatible with background model orders zero and one, but it amounts to a simplifying assumption for orders greater than one, as the background model then influences positions that span the separating event $Y_i = 1$. Regarding the stringency of the score cutoff, the assumption is met exactly only if a motif match corresponds to a single word. For cutoffs $t_\alpha$ that are associated with a set of multiple compatible words, which is usually the case, this assumption is only approximately satisfied, and for too relaxed choices of $t_\alpha$, the assumption might not be justified. Therefore, while the dynamic programming approach does not explicitly rely on the Poisson assumption, it still requires reasonably stringent choices for $t_\alpha$. Our comparative analysis suggests that the assumption still holds reasonably well for $\alpha = 0.01$, but we recommend against using too relaxed choices of $\alpha$ (e.g., $\alpha = 0.05$).

Finally, while the dynamic programming algorithm achieves more or similarly accurate results compared with the other models, its runtime requirement is significantly more demanding. For instance, it scales quadratically with the sequence length. Therefore, it is best suited for analyzing a small set (e.g., some 100 sequences) of relatively short length (e.g., some 100 bp in length). If long sequences are subjected to the study, we recommend to use the compound Poisson model instead.

# Appendix

## A. RELATIONSHIP BETWEEN THE MARKOV MODEL AND THE DYNAMIC PROGRAMMING ALGORITHM

The Markov model and the dynamic programming procedure discussed in this article are tightly linked. This can be appreciated by the fact that both are completely determined by $\tau$ and $\beta_i$. Therefore, by algebraic rearrangement, one could transform one representation into the other. In this section, we illustrate the equivalence relationship between the transition probabilities of the Markov model and the quantities used in the dynamic programming algorithm by an example.

Consider the case of computing the probability $P(Y_1 \dots Y_5 = 01010)$ using a motif of length $M = 3$, which involves the evaluation of $\tau$, $\beta_1$, and $\beta_2$ [defined by Equations (3) and (4)].

First, using the correspondence relationship [Equation (11)], the transition probabilities of the Markov model can be used to express the desired probability as

$$P(Y_1 \dots Y_5 = 01010) = P(n \to n)P(n \to h)P(h \to n_1)P(n_1 \to h)P(h \to n_1),$$

and then, using the definitions for the transition probabilities [Equations (13)–(19)], we obtain

$$P(Y_1 \dots Y_5 = 01010) = (1-\tau)\tau(1-\beta_1)\frac{\beta_2}{(1-\beta_1)}(1-\beta_1)$$

and finally by cancellation

$$P(Y_1 \dots Y_5 = 01010) = (1-\tau)\tau\beta_2(1-\beta_1).$$

However, by definition, $\tau$, $\beta_1$, and $\beta_2$ can be used directly to express

$$P(Y_1 \ldots Y_5 = 01010) = (1-\tau)\tau\beta_2(1-\beta_1),$$

which yields the same results as above.

The latter approach was adapted by the dynamic programming algorithm to sum the probabilities $P(Y_1..Y_N)$.

## B. MARKOV MODEL FOR GENERATING MOTIF MATCHES IN BOTH DNA STRANDS

While in the main text, we discussed the Markov model for generating $Y_1Y_2Y_3\cdots$, that is, when a single DNA strand is scanned for matches, in this section, we introduce the extension of the Markov model to the case when both DNA strands are scanned. Without loss of generality, we seek to describe the random process in the order $Y_1Y'_1Y_2Y'_2\cdots$, where matches are considered from left to right and forward strand matches precede reverse strand matches at the same position.

### *B.0.1. Model states, state transitions, and transition probabilities*

Like in the main text, we define the state space of the Markov model via match patterns in $Y_1Y'_1Y_2Y'_2\cdots$. We obtain

$$h_f \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=\bullet, & Y'_{-1}=\bullet, & Y'_0=\bullet \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=\bullet, & Y_0=1 \end{pmatrix}$$

$$h_r \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=\bullet, & Y'_{-1}=\bullet, & Y'_0=1 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=\bullet, & Y_0=\bullet \end{pmatrix}$$

$$n_0 \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=\bullet, & Y'_{-1}=\bullet, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=\bullet, & Y_0=1 \end{pmatrix}$$

$$n_1 \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=\bullet, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=1, & Y_0=0 \end{pmatrix}$$

$$\vdots$$

$$n_{M-3} \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=0, & \cdots, & Y'_{-2}=0, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=1, & \cdots, & Y_{-2}=0, & Y_{-1}=0, & Y_0=0 \end{pmatrix}$$

$$n_{M-2} \widehat{=} \begin{pmatrix} Y'_{-M+2}=0, & Y'_{-M+3}=0, & \cdots, & Y'_{-2}=0, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=1, & Y_{-M+3}=0, & \cdots, & Y_{-2}=0, & Y_{-1}=0, & Y_0=0 \end{pmatrix}$$

$$n'_1 \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=\bullet, & Y'_{-1}=1, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=\bullet, & Y_0=0 \end{pmatrix}$$

$$n'_2 \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=\bullet, & \cdots, & Y'_{-2}=1, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=\bullet, & Y_{-1}=0, & Y_0=0 \end{pmatrix}$$

$$\vdots$$

$$n'_{M-3} \widehat{=} \begin{pmatrix} Y'_{-M+2}=\bullet, & Y'_{-M+3}=1, & \cdots, & Y'_{-2}=0, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=\bullet, & \cdots, & Y_{-2}=0, & Y_{-1}=0, & Y_0=0 \end{pmatrix}$$

$$n'_{M-2} \widehat{=} \begin{pmatrix} Y'_{-M+2}=1, & Y'_{-M+3}=0, & \cdots, & Y'_{-2}=0, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=\bullet, & Y_{-M+3}=0, & \cdots, & Y_{-2}=0, & Y_{-1}=0, & Y_0=0 \end{pmatrix}$$

$$n \widehat{=} \begin{pmatrix} Y'_{-M+2}=0, & Y'_{-M+3}=0, & \cdots, & Y'_{-2}=0, & Y'_{-1}=0, & Y'_0=0 \\ Y_{-M+2}=0, & Y_{-M+3}=0, & \cdots, & Y_{-2}=0, & Y_{-1}=0, & Y_0=0 \end{pmatrix}.$$

Note that the number of states in this case is exactly twice the number of states relative to the Markov model that accounts for matches on a single strand only. The reason for this is that there are separate match

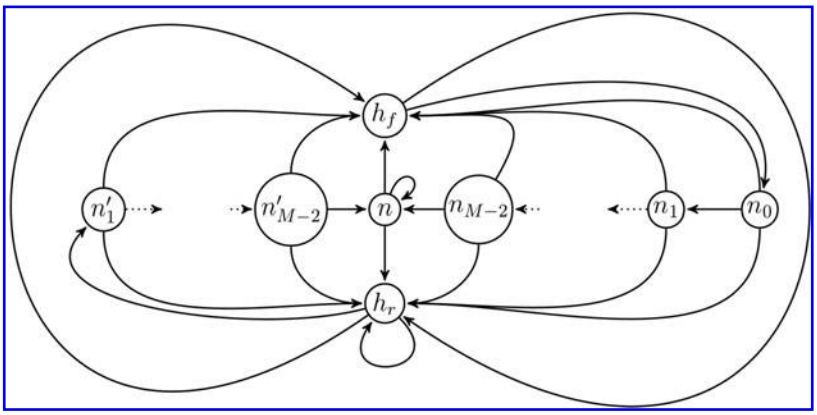


**FIG. 5.** Markov model transition diagram. The states $h_f$ and $h_r$ correspond to obtaining motif hits on the forward or reverse strand, respectively. The remaining states represent the absence of a motif match at the current position, but which memorize previous matches up to the length of the motif. They are necessary to properly account for the self-overlapping structure of the given motif. In the sketch, a self-overlapping match would occur if a match state $h_f$ or $h_r$ follows after any other state except for $n$. A transition from $n$ to $h_f$ or $h_r$ reflects a clump start match.

states $h_f$ and $h_r$ that represent a forward and a reverse strand match. Moreover, $\{n_i\}_{0\le i<M-1}$ memorizes a recent forward strand match, whereas $\{n'_i\}_{1\le i<M-1}$ memorizes a respective reverse strand match. They are necessary to account for self-overlapping matches. Finally, $n$ corresponds to the absence of matches for the last $M-1$ positions (including at the current position). Thus, $n$ is necessary to account for nonself-overlapping matches.

As a consequence of scanning the DNA sequence in the specified order $(Y_1 Y'_1 Y_2 Y'_2 Y_3 \cdots)$, a set of viable transitions is induced, which are illustrated in Figure 5. The transition network accounts for clump start and self-overlapping matches that might arise on both DNA strands.

The state transitions are quantified by the transition matrix, which is defined by

$$M = \begin{bmatrix} 0 & P(h_r \to h_f) & P(n \to h_f) & \mathbf{a}^{\mathrm{T}} & \mathbf{a}'^{\mathrm{T}} \\ P(h_f \to h_r) & P(h_r \to h_r) & P(n \to h_r) & \mathbf{b}^{\mathrm{T}} & \mathbf{b}'^{\mathrm{T}} \\ 0 & 0 & P(n \to n) & \mathbf{c}^{\mathrm{T}} & \mathbf{c}^{\mathrm{T}} \\ P(h_f \to n_0) & 0 & \cdots & & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{C} & \mathbf{0} \\ 0 & P(h_r \to n'_1) & 0 & \cdots & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D} \end{bmatrix}, \tag{33}$$

where the bold zeros denote submatrices containing only zeros and where we made use of

$$\mathbf{a}^{\mathrm{T}} = [\, P(n_0 \to h_f) \quad \cdots \quad P(n_{M-2} \to h_f) \,] \tag{34}$$

$$\mathbf{a}'^{\mathrm{T}} = [\, P(n'_1 \to h_f) \quad \cdots \quad P(n'_{M-2} \to h_f) \,] \tag{35}$$

$$\mathbf{b}^{\mathrm{T}} = [\, P(n_0 \to h_r) \quad \cdots \quad P(n_{M-2} \to h_r) \,] \tag{36}$$

$$\mathbf{b}'^{\mathrm{T}} = [\, P(n'_1 \to h_r) \quad \cdots \quad P(n'_{M-2} \to h_r) \,] \tag{37}$$

$$\mathbf{c}^{\mathrm{T}} = [\, 0 \quad \cdots \quad 0 \quad P(n_{M-2} \to n) \,] \tag{38}$$

$$\mathbf{c}'^{\mathrm{T}} = [\, 0 \quad \cdots \quad 0 \quad P(n'_{M-2} \to n) \,] \tag{39}$$

$$\mathbf{C} = \begin{bmatrix} P(n_0 \to n_1) & 0 & & 0 \\ 0 & P(n_1 \to n_2) & & 0 \\ & & \ddots & \\ 0 & 0 & & P(n_{M-3} \to n_{M-2}) \\ & & & \end{bmatrix} \tag{40}$$

$$\mathbf{D} = \begin{bmatrix} P(n'_1 \to n'_2) & 0 & & 0 \\ 0 & P(n'_2 \to n'_3) & & 0 \\ & & \ddots & \\ 0 & 0 & & P(n'_{M-3} \to n'_{M-2}) \\ & & & \end{bmatrix}.. \tag{41}$$

The individual transition probabilities contained in the transition matrix [Equation (33)] are expressed analogously to the single-stranded scanning scenario, in terms of the clump start probabilities $\tau$ and $\tau'$ and the self-overlapping match probabilities $\beta_k$, $\beta_{3',k}$, $\beta_{5',k}$ [see Equations (5)–(10) in the main text]. They are defined by

$$P(n \to h_f) := P\left( \begin{matrix} Y'_0 = \bullet \\ Y_0 = 1 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-M-1} = 0 \\ Y_{-1} = 0, & \cdots, & Y_{-M-1} = 0 \end{matrix} \right) = \tau \tag{42}$$

$$P(n \to h_r) := P\left( \begin{matrix} Y'_0 = 1 \\ Y_0 = 0 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-M-1} = 0 \\ Y_{-1} = 0, & \cdots, & Y_{-M-1} = 0 \end{matrix} \right) = \tau' \tag{43}$$

$$P(n \to n) := 1 - \tau - \tau' \tag{44}$$

$$P(h_f \to h_r) := P(Y'_0 = 1 | Y_0 = 1) = \beta_{3',0} \tag{45}$$

$$P(h_f \to n_0) := 1 - \beta_{3',0} \tag{46}$$

$$P(h_r \to h_f) := P(Y_0 = 1 | Y'_{-1} = 1) = \beta_{5',1} \tag{47}$$

$$P(h_r \to h_r) := P(Y'_0 = 1 | Y'_{-1} = 1) = \beta_1 \tag{48}$$

$$P(h_r \to n'_1) := P(Y'_0 = 0, Y_0 = 0 | Y'_{-1} = 1) = 1 - \beta_{5',1} - \beta_1 \tag{49}$$

$$\begin{aligned} P(n_k \to h_f) &:= P\left( \begin{matrix} Y'_0 = \bullet \\ Y_0 = 1 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-k} = 0 \\ Y_{-1} = 0, & \cdots, & Y_{-k} = 1 \end{matrix} \right) \\ &= \frac{P(Y_{k+1} = 1, \{Y_j = 0, Y'_j = 0\}_{1 \le j \le k}, Y'_0 = 0 | Y_0 = 1)}{P(\{Y_j = 0, Y'_j = 0\}_{1 \le j \le k}, Y'_0 = 0 | Y_0 = 1)} \\ &= \frac{\beta_{k+1}}{1 - \sum_{i=1}^{k} \beta_k - \sum_{i=0}^{k} \beta_{3',k}} \qquad \text{for } 1 \le k \le M-2 \end{aligned} \tag{50}$$

$$\begin{aligned} P(n_k \to h_r) &:= P\left( \begin{matrix} Y'_0 = 1 \\ Y_0 = 0 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-k} = 0 \\ Y_{-1} = 0, & \cdots, & Y_{-k} = 1 \end{matrix} \right) \\ &= \frac{\beta_{3',k+1}}{1 - \sum_{i=1}^{k} \beta_k - \sum_{i=0}^{k} \beta_{3',k}} \quad \text{for } 1 \le k \le M-2 \end{aligned} \tag{51}$$

$$\begin{aligned} P(n'_k \to h_f) &:= P\left( \begin{matrix} Y'_0 = \bullet \\ Y_0 = 1 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-k} = 1 \\ Y_{-1} = 0, & \cdots, & Y_{-k} = \bullet \end{matrix} \right) \\ &= \frac{\beta_{5',k+1}}{1 - \sum_{i=1}^{k} \beta_k - \sum_{i=1}^{k} \beta_{5',k}} \quad \text{for } 1 \le k \le M-2 \end{aligned} \tag{52}$$

$$P(n'_k \to h_r) := P\left( \begin{matrix} Y'_0 = 1 \\ Y_0 = 0 \end{matrix} \middle| \begin{matrix} Y'_{-1} = 0, & \cdots, & Y'_{-k} = 1 \\ Y_{-1} = 0, & \cdots, & Y_{-k} = \bullet \end{matrix} \right)$$

$$= \frac{\beta_{k+1}}{1 - \sum_{i=1}^{k} \beta_k - \sum_{i=1}^{k} \beta_{5',k}} \quad \text{for } 1 \le k \le M-2 \tag{53}$$

$$P(n_k \to n_{k+1}) := 1 - P(n_k \to h_f) - P(n_k \to h_r) \quad \text{for } 1 \le k \le M-3 \tag{54}$$

$$P(n'_k \to n'_{k+1}) := 1 - P(n'_k \to h_f) - P(n'_k \to h_r) \quad \text{for } 1 \le k \le M-3 \tag{55}$$

$$P(n_{M-2} \to n) := 1 - P(n_{M-2} \to h_f) - P(n_{M-2} \to h_r) \tag{56}$$

$$P(n'_{M-2} \to n) := 1 - P(n'_{M-2} \to h_f) - P(n'_{M-2} \to h_r). \tag{57}$$

*B.1. Computation of the clump start probability.* The transition matrix of the Markov model is fully determined by the clump start and the self-overlapping match probabilities. The self-overlapping match probabilities are approximated according to Kopp and Vingron (2017). We seek to leverage the Markov model, similarly as described in the main text, to establish the clump start probabilities.

First, note that when studying both DNA strands, two clump start events are conceivable, which occur with probabilities $\tau$ and $\tau'$. However, we can approximately express

$$\tau' \approx \tau(1 - \beta_{3',0}). \tag{58}$$

That is, for a palindromic motif, the clump is biased toward starting at the forward strand and the match at the reverse strand will be considered an overlapping match. As a consequence, we only need to identify one unknown quantity $\tau$, rather than solving for $\tau$ and $\tau'$ simultaneously, which significantly simplifies the problem. We seek to adjust $\tau$ such that in the stationary distribution the states $h_f$ and $h_r$ are visited with probability $2 \times \alpha$ and define a function

$$f(\tau) := -(2\alpha)\log(\mu(h_f;\tau) + \mu(h_r;\tau)) - (1-2\alpha)\log(1 - \mu(h_f;\tau) - \mu(h_r;\tau))), \tag{59}$$

whose minimum is obtained at $2\alpha = \mu(h_f;\tau) + \mu(h_r;\tau)$. We optimize $f(\tau)$ as described in the main text.

*B.2. Distribution of the number of matches in both DNA strands.* In this section, we discuss the extension of the algorithm for computing the number of motif matches in both DNA strands simultaneously.

We define the total number of matches (on both strands) in the sequence from positions 1 to $j$ by $X_{[1:i]}$ and denote the probability of obtaining $x$ matches in this segment by $P(X_{[1:i]} = x)$.

For the same reason as discussed in the main text, accounting for self-overlapping matches in the recursion algorithm requires memorizing position and strandedness of the last motif match in $[1 : i]$. Therefore, we define the number of matches $X^a_{[1:i]}$ and $X'^a_{[1:i]}$ that end with a *forward strand match* and a *reverse strand match*, respectively. If $1 \le a < M$, the match is located at position $i - a + 1$, whereas if $a = M$, the last match occurred at least $M - 1$ positions upstream of $i$.

Assuming that $P(X^a_{[1:j]} = x)$ and $P(X'^a_{[1:j]} = x)$ have already been established, we obtain $P(X^a_{[1:j]} = x+1)$ and $P(X'^a_{[1:j]} = x+1)$ recursively according to

$$\begin{aligned} P(X^b_{[1:j]} = x+1) = &\sum_{i=1}^{j-1} \sum_{a=1}^{M} P(X^a_{[1:i]} = x) \cdot h(a) \cdot z(j-i) \\ &+ \sum_{i=1}^{j-1} \sum_{a=1}^{M} P(X'^a_{[1:i]} = x) \cdot h_{5'}(a) \cdot z(j-i) \end{aligned} \tag{60}$$

$$\begin{aligned} P(X'^b_{[1:j]} = x+1) = &\sum_{i=1}^{j-1} \sum_{a=1}^{M} P(X^a_{[1:i]} = x) \cdot h_{3'}(a) \cdot z'(j-i) \\ &+ \sum_{i=1}^{j-1} \sum_{a=1}^{M} P(X'^a_{[1:i]} = x) \cdot h'(a) \cdot z'(j-i) \\ &+ \sum_{i=1}^{j} P(X^1_{[1:i]} = x) \cdot h_{3'}(0) \cdot z'(j-i+1) \end{aligned} \tag{61}$$

using the definitions

$$\delta_i := 1 - \sum_{k=1}^{i} \beta_k - \sum_{k=0}^{i} \beta_{3',k} \tag{62}$$

$$\delta'_i := 1 - \sum_{k=1}^{i} \beta_k - \sum_{k=1}^{i} \beta_{5',k}. \tag{63}$$

$$b := \begin{cases} j-i & \text{if } j-i < M \\ M & \text{o.w.} \end{cases} \tag{64}$$

$$h(a) := \begin{cases} \beta_a & \text{if } a < M \\ \alpha' & \text{o.w.} \end{cases} \tag{65}$$

$$h'(a) := \begin{cases} \beta_a & \text{if } a < M \\ \alpha'(1-\beta_{3',0}) & \text{o.w.} \end{cases} \tag{66}$$

$$h_{3'}(a) := \begin{cases} \beta_{3',a} & \text{if } a < M \\ \alpha' \cdot (1-\beta_{3',0}) & \text{o.w.} \end{cases} \tag{67}$$

$$h_{5'}(a) := \begin{cases} \beta_{5',a} & \text{if } a < M \\ \alpha' & \text{o.w.} \end{cases} \tag{68}$$

$$z(a) := \begin{cases} 1 & \text{if } a < M \\ \delta_{M-1} \cdot (1-\tau(2-\beta_{3',0}))^{a-M} & \text{o.w.} \end{cases} \tag{69}$$

$$z'(a) := \begin{cases} 1 & \text{if } a < M \\ \delta'_{M-1} \cdot (1-\tau(2-\beta_{3',0})^{a-M} & \text{o.w.} \end{cases}. \tag{70}$$

$h(.)$, $h'(.)$, $h_{3'}(a)$, and $h_{5'}(a)$ incorporate an additional match depending on the locations and strandedness of the previous and the next match: forward → forward, reverse → reverse, forward → reverse, and reverse → forward. $z(.)$ and $z'(.)$ account for the absence of matches in the remainder of the sequence. Therefore, notice that the first and second summations on the right-hand side of Equations (60) and (61) account for a previous forward and reverse strand match, respectively. Moreover, the third summation on the right-hand side of Equation (61) accounts for palindromic motif hits.

To establish the distribution $P(X_{[1:N-M+1]})$, we start by setting $P(X_{[1:j]}=0) = (1-\tau-\tau')^j = (1-\tau(2-\beta_{3',0}))^j$, where we made use of Equations (5) and (20). The dynamic programming is initialized for $P(X^a_{[1:j]}=1)$ and $P(X'^a_{[1:j]}=1)$ according to

$$P(X^a_{[1:j]}=1) = \begin{cases} (j-M+1)\times(1-\tau)^{j-M}\tau\delta_{M-1} & \text{for } a=M \\ (1-\tau)^{j-a}\tau & \text{ow.} \end{cases} \tag{71}$$

for $1 \le a \le M$ and $1 \le j \le N-M+1$. Then, we evaluate the recursion defined by Equations (60) and (61) in an ordered manner from $x=2$ to the maximal number of matches to be considered $x_{\max}$. Finally, the algorithm terminates by

$$\begin{aligned} P(X_{[1:N-M+1]}=x) = {} & P(X^M_{[1:N-M+1]}=x) + P(X'^M_{[1:N-M+1]}=x) + \\ & \sum_{a=1}^{M-1} P(X^a_{[1:N-M+1]}=x)\delta_{a-1} + \sum_{a=1}^{M-1} P(X'^a_{[1:N-M+1]}=x)\delta'_{a-1}, \end{aligned} \tag{72}$$

which yields the distribution of the number of motif matches in a sequence of length $N$ with a motif length of $M$.

## AUTHOR DISCLOSURE STATEMENT

The authors declare that no competing financial interests exist.

## REFERENCES

Alberts B., Johnson A., Lewis J., et al. 2002. *Molecular Biology of the Cell, Fourth Edition*. Garland Science, London.

Bailey, T.L., Boden, M., Buske, F.A., et al. 2009. Meme suite: Tools for motif discovery and searching. *Nucleic Acids Res* gkp335.

Bishop, C.M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Cartharius, K., Frech, K., Grote, K., et al. 2005. Matinspector and beyond: Promoter analysis based on transcription factor binding sites. *Bioinformatics.* 21:2933–2942.

Chen, Q.K., Hertz, G.Z., and Stormo, G.D. 1995. Matrix search 1.0: A computer program that scans dna sequences for transcriptional elements using a database of weight matrices. *Comput. Appl. Biosci.* 11:563–566.

Frith, M.C., Fu, Y., Yu, L., et al. 2004. Detection of functional DNA motifs via statistical over-representation. *Nucleic Acids Res.* 32:1372–1381.

Karlin, S., and Taylor, H.E. 1981. *A Second Course in Stochastic Processes*. Academic Press, London, UK.

Kopp, W., and Vingron, M. 2017. An improved compound poisson model for the number of motif hits in DNA sequences. *Bioinformatics* 33:3929–3937.

Kulakovskiy, I.V., Medvedeva, Y.A., Schaefer, U., et al. 2013. Hocomoco: A comprehensive collection of human transcription factor binding sites models. *Nucleic Acids Res.* 41:D195–D202.

Li, N., and Tompa, M. 2006. Analysis of computational approaches for motif discovery. *Algorithms Mol. biol.* 1:8.

Liu, J.S., and Lawrence, C.E. 1999. Bayesian inference on biopolymer models. *Bioinformatics.* 15:38–52.

Marschall, T., and Rahmann, S. 2008. Probabilistic arithmetic automata and their application to pattern matching statistics, 95–106. Eds: Ferragina, P., and Landau, G.M. In *Annual Symposium on Combinatorial Pattern Matching*. Springer, Berlin-Heidelberg.

Marschall, T., and Rahmann, S. 2010. Speeding up exact motif discovery by bounding the expected clump size, 337–349. In *International Workshop on Algorithms in Bioinformatics*. Springer.

McLeay, R.C., and Bailey, T.L. 2010. Motif enrichment analysis: A unified framework and an evaluation on chip data. *BMC Bioinformatics.* 11:165.

Neyman, J., and Pearson, E.S. 1933. The testing of statistical hypotheses in relation to probabilities a priori. *Math. Proc. Camb. Philos. Soc.* 29:492–510.

Pape, U.J., Rahmann, S., Sun, F., et al. 2008. Compound poisson approximation of the number of occurrences of a position frequency matrix (PFM) on both strands. *J. Comput. Biol.* 15:547–564.

Rahmann, S., Müller, T., and Vingron, M. 2003. On the power of profiles for transcription factor binding site detection. *Stat. Appl. Genet. Mol. Biol.* 2:Article7.

Reinert, G., Schbath, S., and Waterman, M.S. 2000. Probabilistic and statistical properties of words: An overview. *J. Comput. Biol.* 7:1–46.

Roider, H.G., Manke, T., O'keeffe, S., et al. 2009. Pastaa: Identifying transcription factors associated with sets of co-regulated genes. *Bioinformatics.* 25:435–442.

Sandelin, A., Alkema, W., Engström, P., et al. 2004. Jaspar: An open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res.* 32:D91–D94.

Schneider, T.D., and Stephens, R.M. 1990. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.* 18:6097–6100.

Stormo, G.D. 2000. DNA binding sites: Representation and discovery. *Bioinformatics.* 16:16–23.

Thomas-Chollier, M., Sand, O., Turatsinze, J.V., et al. 2008. RSAT: Regulatory sequence analysis tools. *Nucleic Acids Res.* 36:W119–W127.

Thurman, R.E., Rynes, E., Humbert, R., et al. 2012. The accessible chromatin landscape of the human genome. *Nature.* 489:75–82.

Touzet, H., and Varré, J.S. 2007. Efficient and accurate p-value computation for position weight matrices. *Algorithms Mol. Biol.* 2:1748–1788.

Wingender, E., Dietze, P., Karas, H., et al. 1996. Transfac: A database on transcription factors and their DNA binding sites. *Nucleic Acids Res.* 24:238–241.

Zambelli, F., Pesole, G., and Pavesi, G. 2009. Pscan: Finding over-represented transcription factor binding site motifs in sequences from co-regulated or co-expressed genes. *Nucleic Acids Res.* 37:W247–W252.

Zhang, J., Jiang, B., Li, M., et al. 2007. Computing exact p-values for DNA motifs. *Bioinformatics.* 23:531–537.

Address correspondence to:
*Dr. Wolfgang Kopp*
*Berlin Institute for Medical Systems Biology*
*Max Delbrueck Center for Molecular Medicine*
*Robert-Roessle-Strasse 10*
*13125 Berlin*
*Germany*

*E-mail:* wolfgang.kopp@mdc-berlin.de