University of California, Berkeley

U.C. Berkeley Division of Biostatistics Working Paper Series

Year 2006 *Paper* 209

Supervised Detection of Conserved Motifs in DNA Sequences with cosmo

Oliver Bembom* Sunduz Keles†

Mark J. van der Laan‡

*Division of Biostatistics, School of Public Health, University of California, Berkeley, bembom@gmail.com

†Dept. of Statistics & Dept. of Biostatistics & Medical Informatics, University of Wisconsin, Madison

‡Division of Biostatistics, School of Public Health, University of California, Berkeley, laan@berkeley.edu

Supervised Detection of Conserved Motifs in DNA Sequences with cosmo

Oliver Bembom, Sunduz Keles, and Mark J. van der Laan

Abstract

A number of computational methods have been proposed for identifying transcription factor binding sites from a set of unaligned sequences that are thought to share the motif in question. We here introduce an algorithm, called cosmo, that allows this search to be supervised by specifying a set of constraints that the position weight matrix of the unknown motif must satisfy. Such constraints may be formulated, for example, on the basis of prior knowledge about the structure of the transcription factor in question. The algorithm is based on the same two-component multinomial mixture model used by MEME, with stronger reliance, however, on the likelihood principle instead of more ad-hoc criteria like the E-value. The intensity parameter in the ZOOPS and TCM models, for instance, is estimated based on a profile-likelihood approach, and the width of the unknown motif is selected based on BIC. These changes allow cosmo to outperform MEME even in the absence of any constraints, as evidenced by 2- to 3-fold greater sensitivity in some simulation studies. Additional improvements in performance can be achieved by selecting the model type (OOPS, ZOOPS, or TCM) data-adaptively or by supplying correctly specified constraints, especially if the motif appears only as a weak signal in the data. The algorithm can data-adaptively choose between working in a given constrained model or in the completely unconstrained model, guarding against the risk of supplying mis-specified constraints. Simulation studies suggest that this approach can offer 3 to 3.5 times greater sensitivity than MEME. The algorithm has been implemented in the form of a stand-alone C program as well as a web application that can be accessed at http://cosmoweb.berkeley.edu. An R package is available through Bioconductor (http://bioconductor.org).

Contents

1 Introduction

An important goal in contemporary biology consists of deciphering the complex network that regulates the expression of an organism's genome. A central role in this network is played by transcription factors that regulate gene expression by binding to conserved short sequences in the vicinity of their target genes (Davidson, 2001). The discovery and description of these *binding sites* or *motifs* has therefore been at the heart of efforts aimed at understanding gene regulatory networks.

Traditionally, experimental methods have been used for this purpose, leading to a set of target sites from multiple genes that could then be aligned to estimate the position weight matrix (PWM) of the motif - a $4 \times W$ matrix in which position (j, w) gives the probability of observing nucleotide j in position w of a motif of length W. Currently, however, such position weight matrix estimates are more commonly obtained by applying pattern discovery algorithms to functional genomics data. Modern high-throughput methods such as cDNA microarrays (Roth et al., 1998; Eisen et al., 1998; Bussemaker et al., 2001)or SAGE (Powell, 2000), for example, can identify sets of co-regulated genes whose promoter sequences can then be scanned for statistically over-represented patterns that are likely transcription factor binding sites (Lawrence et al., 1993; Bussemaker et al., 2001).

While this approach has proven fruitful for the discovery of such binding sites in yeast, its application to metazoan genomes has met with considerable difficulty since binding sites tend to be spread out over much larger regions of genomic sequence. Efforts at tackling this signal-to-noise problem have concentrated mostly on phylogenetic footprinting, i.e. cross-species sequence comparisons that remove noise by focusing on sequences under selective pressure (Fickett and Wasserman, 2000). Sandelin and Wassermann (2004), however, recently described an alternative approach that is based on prior knowledge about the structural class of the mediating transcription factor of interest. Such knowledge is often available on the basis of genetics or similarities between biological systems. Studies in *Caenorhabditis elegans*, for example, have suggested that downstream insulin response pathways are mediated by forkhead transcription factors (Ogg et al., 1997). For most structurally related families of transcription factors, there are clear similarities in the sequences of the sites to which they bind (Luscombe et al., 2000). Eisen (2005), for example, has demonstrated that motifs bound by proteins with structurally similar DNA binding domains tend to have similar information content profiles (Schneider et al., 1986). Prior knowledge about the structural class of the mediating transcription factor thus often translates into constraints on the unknown position weight matrix that can be used to enhance the sensitivity of pattern discovery algorithms. Sandelin and Wassermann (2004) show that the benefit of such prior knowledge is comparable to the specificity improvements obtained through phylogenetic footprinting.

Currently, only a few motif finding algorithms such as ANN-Spec (Workman and Stormo, 2000) or the Gibbs motif sampler (Neuwald et al., 1995; Thompson et al., 2003) are capable of incorporating prior knowledge about the unknown motif. These algorithms generally require the user to supply an appropriate prior distribution on the entries of the corresponding position weight matrix. van Zwet et al. (2005) recently described an algorithm that instead allows the user to place restrictions on the order of the information content across the motif.

Keleş et al. (2003) introduced a constrained motif detection algorithm - `COMODE` - that generalizes this approach by allowing the user to specify a set of arbitrary constraints that the unknown position weight matrix must satisfy. Their algorithm is based on a probabilistic model that describes the DNA sequences of interest through a two-component multinomial mixture model as first introduced by Lawrence and Reilly (1990), with estimates of the position weight matrix entries obtained by maximizing the observed data likelihood over the smaller parameter space corresponding to the imposed constraints.

This article focuses on a number of methodological improvements and extensions to the algorithm developed in Keleş et al. (2003), relating mostly to the data-adaptive selection of various model parameters. Keleş et al. propose to use likelihood-based cross-validation for this purpose. In particular, their algorithm relies on this approach for the sake of estimating the unknown motif width. Furthermore, the authors suggest that likelihood-based cross-validation can be used to choose an appropriate constraint set from a whole collection of candidate constraint sets. Keleş et al. base their advocacy for this approach primarily on certain finite-sample optimality results derived by van der Laan et al. (2003) rather than on simulation studies in the given setting of motif detection.

In this article, we present detailed simulation results that compare the performance of likelihood-based cross-validation to that of a number of other model selection techniques. Among the other techniques we consider are model selection based on the E-value of the resulting multiple alignment, model selection by AIC or BIC, as well as cross-validation based on the Euclidean norm between two position weight matrices. We examine the performance of these estimators not only in the context of choosing the motif width and an appropriate constraint set, as proposed by Keleş et al., but also in the context of choosing the appropriate model type (OOPS, ZOOPS, or TCM).

We introduce a fast and scalable new implementation of the algorithm originally proposed by Keleş et al. called `cosmo` that not only makes use of more targeted model selection approaches but also improves on `COMODE` in several other ways. First, a number of changes to the algorithm allow `cosmo` to outperform one of the most commonly used motif finding algorithms, `MEME` (Bailey and Elkan, 1995a), even in the absence of constraints. Second, various computational modifications now allow realistic jobs to be run in five to ten minutes, making `cosmo` at least competitive with `MEME` in this aspect as well. Third, while `COMODE` requires the user to specify constraints by supplying two `C` functions, one for evaluating the constraints themselves and one for evaluating their gradient, `cosmo` allows them to be defined in a simple text file according to a straightforward standard. Lastly, we make our implementation available not only as a stand-alone program and an `R` package, but also in the form of a web application that allows users to run jobs on a designated web server.

The remainder of this article is organized as follows: Section 2 describes the probabilistic models that are commonly used in the context of motif detection. In particular it contains a proposal for evaluating the exact likelihood function corresponding to the TCM random process, which previously has been based on approximations to this random process. Section 3 discusses the various constraints that can be used to supervise the motif search. The following section briefly reviews the various model selection techniques that we consider

in later sections. After discussing the design of the simulation studies used to assess the performance of the algorithm in section 5, we first present simulation results pertaining to the choice of starting values, the estimation of the intensity parameter in the ZOOPS and TCM models, and the comparison between exact and approximate methods in the TCM model. Section 7 then describes the simulation studies we have conducted for identifying the optimal model selection approaches for the purposes of choosing the motif width, the model type, and the constraint set. The following section provides a detailed example to illustrate the use of our web application. We end with a brief discussion of our methods and the simulation studies we have conducted.

2 Probabilistic models

In this section, we formally define the observed data structure and describe three different probabilistic models that are used to model the distribution of this data structure. We use the following notation in describing these models. Let $X_{il} \in \{1 \equiv A, 2 \equiv C, 3 \equiv G, 4 \equiv T\}$ denote the nucleotide in position l of sequence i. Let the length of sequence i be denoted by L_i and let $\mathbf{X}_i = \{X_{il}\}_{l=1}^{L_i}$ denote the entire sequence of nucleotides in sequence i. The observed data are then given by N i.i.d. random variables $\{\mathbf{X}_1, ..., \mathbf{X}_N\}$.

2.1 Motifs and background

All of the models described below assume that sequences are generated according to a multinomial mixture model with two components, one that describes the distribution of nucleotides in the motif, and one that describes the distribution of nucleotides in the background.

Nucleotides that are part of the transcription factor binding site are assumed to be generated according to the following statistical model: The nucleotide at position w, denoted by M_w, is drawn from a multinomial distribution with parameter vector

$$\mathbf{P}_w \equiv (P_{w1}, P_{w2}, P_{w3}, P_{w4}) \equiv (P(M_w = 1), P(M_w = 2), P(M_w = 3), P(M_w = 4))$$

such that individual positions are independent of each other and the $\mathbf{P}_w$ are allowed to be different for each position. Note that the width W of the motif is usually unknown *a priori*. The $4 \times W$ matrix with column w given by $\mathbf{P}_w$ is referred to as the position weight matrix (PWM) of the motif.

`COMODE` allows the user to specify a set of constraints C that the position weight matrix is required to satisfy. The nature of these constraints is described in detail below, but some possible examples include constraints on the information content profile, the probabilities of individual nucleotides, or the palindromicity of subsequences of the motif. We extend this approach by allowing the user to specify a number of constraint sets $C_1, ..., C_d$, leading to the weakened assumption that the true position weight matrix only satisfy at least one of the supplied constraint sets. We note that, in particular, it is possible to include an empty constraint set in the collection $C_1, ..., C_d$, which in essence protects the user from the risk of

model mis-specification through the imposition of a wrong set of constraints on the position weight matrix.

We assume that nucleotides that are not part of a motif are generated according to a k-th order Markov model. Let $\bar{X}_i(l,m) \equiv (X_{il}, ..., X_{im})$ denote nucleotides l through m of sequence i. Then a k-th order Markov model for the distribution of background nucleotides assumes that a background nucleotide at position l is drawn from a multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$, i.e. the parameter vector of the multinomial distribution is allowed to depend on the previous k nucleotides.

Let B_{il} be the indicator that a motif starts in position l of sequence i. Our main parameter of interest then consists of the position weight matrix of the motif as well as the collection $K \equiv \{(i,l) : B_{il} = 1\}$ of true motif start sites in our data set.

2.2 OOPS

The one-occurrence-per-sequence (OOPS) model assumes that every sequence contains exactly one occurrence of the motif. For a given sequence $\mathbf{X}_i$, any of the $L_i - W + 1$ eligible motif starts are equally likely to be the start site of the motif. At a given start site, the motif is equally likely to be present in either one of the two possible orientations. For example, the motif ATGCCC may be present as ATGCCC or in its reverse complement orientation as GGGCAT. Specifically, the OOPS model assumes that a given sequence $\mathbf{X}_i$ is generated according to the following random process:

1. Draw a motif start site S_i from a uniform discrete distribution with support $\{1, ..., L_i - W + 1\}$. Let $\mathbf{B}_i \equiv (I(S_i = 1), ..., I(S_i = L_i))$ be a vector of indicator variables whose l-th element is 1 if the motif start site is equal to the l-th position in sequence i.

2. Set $l = 1$. If $l < S_i$ continue with step 3, else continue with step 4.

3. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$. Set $l = l + 1$. If $l < S_i$ continue with step 3, else continue with step 4.

4. Set $w = 1$. Draw Y from a Bernoulli(0.5) distribution. If $Y = 1$, go to step 5, else go to step 6.

5. Draw X_{il} from the multinomial distributions with parameter vector $\mathbf{P}_w$. Set $l = l+1$, $w = w + 1$. If $w \leq W$, continue with step 5; else if $l \leq L_i$ continue with step 7; else stop.

6. Draw X_{il} from the multinomial distributions with parameter vector $\mathbf{P}_{W-w+1}$. Set $l = l + 1$, $w = w + 1$. If $w \leq W$, continue with step 6; else if $l \leq L_i$ continue with step 7; else stop.

7. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$. Set $l = l + 1$. If $l \leq L_i$ continue with step 7, else stop.

Let $\tau(i,l,W) \equiv \{l, ..., l+W-1\}$ denote the sites that are part of a motif of length W given a particular motif start site l on sequence i. Let $c(j)$ denote the complement of nucleotide j, $j = 1, \ldots, 4$; thus, $c(1) = 4$ and $c(2) = 3$, for example. The likelihood $P(\mathbf{X}_i|\theta)$ of a given sequence $\mathbf{X}_i$ under the OOPS model can then be calculated as

$$\frac{1}{L_i - W + 1} \sum_{l=1}^{L_i-W+1} \prod_{k \notin \tau(i,l,W)} \prod_{j=1}^{4} P_{0j}^{I(X_{ik}=j)} \frac{1}{2} \Big[\prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)} + \prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=c(j))} \Big]$$

2.3 ZOOPS

The zero-or-one-occurrence-per-sequence (ZOOPS) model assumes that a given sequence $\mathbf{X}_i$ contains one occurrence of the motif with probability π and no occurrences of the motif with probability $1 - \pi$. For a given sequence $\mathbf{X}_i$ that contains a motif, any of the $L_i - W + 1$ eligible motif starts are equally likely to be the start site of the motif. At a given start site, the motif is equally likely to be present in either one of the two possible orientations. Specifically, the ZOOPS model assumes that a given sequence $\mathbf{X}_i$ is generated according to the following random process:

1. Draw V_i from a Bernoulli(π) distribution. If $V_i = 1$, draw $\mathbf{X}_i$ according to the random process of the OOPS model above. Otherwise continue with step 2.
2. Set $l = 1$.
3. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$. Set $l = l + 1$. If $l \leq L_i$ continue with step 3, else stop.

The likelihood $P(\mathbf{X}_i|\theta)$ of a given sequence $\mathbf{X}_i$ under the ZOOPS model can then be calculated as

$$\frac{\pi}{L_i - W + 1} \sum_{l=1}^{L_i-W+1} \prod_{k \notin \tau(i,l,W)} \prod_{j=1}^{4} P_{0j}^{I(X_{ik}=j)} \frac{1}{2} \Big[\prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)} + \prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=c(j))} \Big] + (1-\pi) \prod_{l=1}^{L_i} \prod_{j=1}^{4} P_{0j}^{I(X_{ik}=j)}$$

2.4 TCM

The OOPS and ZOOPS models allow at most one occurrence of the motif per sequence. However, there are many biological examples of DNA sequences that contain multiple occurrences of the same transcription factor binding site. Bailey and Elkan (1995a) propose a

two-component mixture (TCM) model for this situation that allows each sequence to contain an arbitrary number of non-overlapping occurrences of the motif.

This model assumes that a given sequence $\mathbf{X}_i$ is generated by repeatedly deciding whether to insert a background nucleotide or a motif of width W. As before, a motif is inserted in either one of the two possible orientations with equal probability. We denote by λ the probability that a motif is inserted at a given position rather than a background nucleotide. Specifically, the TCM model assumes that a given sequence $\mathbf{X}_i$ is generated according to the following random process:

1. Set $l = 1$.

2. Draw B_{il} from a Bernoulli(λ) distribution. If $B_{il} = 0$, go to step 3, else set $w = 1$ and go to step 4.

3. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$. Set $l = l+1$. If $l \leq L_i - W + 1$ continue with step 2, else continue with step 7.

4. Draw Y from a Bernoulli(0.5) distribution. If $Y = 1$, go to step 5, else go to step 6.

5. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_w$. Set $l = l+1$, $w = w+1$. If $w \leq W$, set $B_{il} = 0$ and continue with step 5; else if $l \leq L_i - W + 1$ continue with step 2; else continue with step 7.

6. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_{W-w+1}$. Set $l = l+1$, $w = w+1$. If $w \leq W$, set $B_{il} = 0$ and continue with step 6; else if $l \leq L_i - W + 1$ continue with step 2; else continue with step 7.

7. Draw X_{il} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((l-k) \wedge 1, l-1))$. Set $l = l+1$. If $l < L_i$, continue with step 7, else stop.

The likelihood function for the TCM model is a sum over all possible sample paths that could have produced the sequence at hand. The number of these sample paths for a sequence of length L is on the order of 2^L since the random process repeatedly has the choice to either insert a motif of width W or a background nucleotide. Due to this increased computational complexity as compared to the OOPS and ZOOPS models, exact methods based on the TCM model have been avoided and a number of computationally more tractable approximations have been proposed.

Bailey and Elkan (1995a) obtain a derived data set $\mathcal{D}'$ from the original data set $\mathcal{D}$ that consists of all overlapping subsequences of length W that are contained in the original data set. A proportion λ' of these derived sequences $\mathbf{X}'_i$ represent motifs, whereas a proportion $1 - \lambda'$ consist entirely of background nucleotides. Specifically Bailey and Elkan assume that each derived sequence $\mathbf{X}'_i$ was generated by the following random process:

1. Set $w = 1$.

2. Draw B'_i from a Bernoulli(λ') distribution. If $B'_i = 1$, go to step 3, else go to step 6.

3. Draw Y_i from a Bernoulli(0.5) distribution. If $Y_i = 1$, go to step 4, else go to step 4.

4. Draw X'_{iw} from the multinomial distributions with parameter vector $\mathbf{P}_w$. Set $w = w+1$. If $w \leq W$, continue with step 4; else stop.

5. Draw X'_{iw} from the multinomial distributions with parameter vector $\mathbf{P}_{W-w+1}$. Set $w = w + 1$. If $w \leq W$, continue with step 5; else stop.

6. Draw X'_{iw} from the multinomial distribution with parameter vector $\mathbf{P}_0(\bar{X}_i((w-k) \wedge 1, w-1))$. Set $w = w + 1$. If $w \leq W$ continue with step 6, else stop.

Bailey and Elkan estimate the parameters in this approximate model based on a modified EM-algorithm that includes a smoothing step after the E-step to reduce the degree to which any two overlapping subsequences can both be assigned to the motif component. These estimates are then taken as estimates of the parameters in the original model, except that λ is estimated by

$$\lambda_n = \frac{1}{\frac{1}{\lambda'_n} - W + 1}$$

A drawback of this approach is that the derived sequences $\mathbf{X}'$ are far from independent of each other since they are constructed from overlapping portions of the original sequences. The likelihood function, however, is based on a sample of i.i.d. sequences. The impact of this violated independence assumption is not quite clear.

As shown by Keleş et al. (2003), it is computationally advantageous in the context of constrained motif detection to maximize the observed data likelihood directly rather than to use the EM-algorithm. Since the approximation proposed by Bailey and Elkan is based on an additional smoothing step after each E-step of the EM-algorithm, it cannot be implemented in a straightforward way as part of a constrained motif detection algorithm.

Keleş et al. instead propose applying the ZOOPS model to a derived data set $\mathcal{D}_U$ that is obtained from the original data set $\mathcal{D}$ by dividing each of the original sequences into subsequences of cut length U. The authors test this method on the even skipped gene (*eve*) of *Drosophila* for different choices of the cut parameter U and report that it is fairly robust with respect to the choice of U.

A potential problem with this approach is that it cannot detect motif occurrences that straddle a cut point in one of the original sequences. We consider a modified approach that combines elements of the previous two proposals. We divide the original data set into subsequences of length U such that each subsequence contains the first $W-1$ nucleotides of the following subsequence. The overlaps of length $W-1$ ensure that any motif occurrence that is present in the original sequences can also be detected in the new data set $\mathcal{D}_U$.

We compare these three approximate TCM models to an exact one that is based on our following proposal for a computationally efficient algorithm to calculate the likelihood of sequence i, denoted by $P(\mathbf{X}_i|\theta)$. Let $\pi_i(l|\theta) \equiv P(X_{i1}, ..., X_{il}|B((l-W+2)\wedge 1) = 0, ..., B(l) = 0, \theta)$ denote the likelihood of the first l nucleotides given that no motif starts at any of the

last $W-1$ nucleotides, i.e. given that position $l+1$ has positive probability of being a motif start site. The conditional likelihoods $\pi_i(l)$ are sums over all sample paths that could have generated the observed first l nucleotides with the sole restriction that no motifs start in positions $l-W+2$ through l. Suppose we have already calculated $\pi_i(k)$ for $1 \leq k < l$. Then we can calculate $\pi_i(l)$ by conditioning on whether or not the nucleotide X_{il} was generated by the motif distribution or the background distribution. In the former case, X_{il} must represent the last column of a motif starting in position $l-W+1$, or else the nucleotide $X_{i(l+1)}$ could not possibly be a motif start site; the probability of all sample paths that contain a motif in positions $l-W+1, \ldots, l$ is given by $\pi(l-W+1) \times \lambda$ multiplied by the probability of the last W positions under the motif distribution. In the latter case, the probability of all sample paths that contain a background nucleotide in position l is given by $\pi(l-1) \times (1-\lambda)$ multiplied by the probability of nucleotide X_{il} under the background distribution. Thus, letting $\pi_i(l) \equiv 0$ for $l \leq 0$, we have that $\pi_i(l)$ is given by

$$\pi_i(l-W+1) \times \lambda \times \frac{1}{2}\Big[\prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)} + \prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=c(j))}\Big]$$
$$+ \pi_i(l-1) \times (1-\lambda) \times \prod_{j=1}^{4} P_{0j}^{I(X_{il}=j)}$$

Finally, we recognize that the likelihood of sequence i under the TCM model, $P(\mathbf{X}_i|\theta)$, is given by $\pi_i(L_i|\theta)$ since no motif can start at any of the last $W-1$ nucleotides of the sequence.

This algorithm for evaluating the exact likelihood function under the TCM model allows us to estimate the parameters of the TCM model directly, without relying on one of the approximations described above. Furthermore, we can use it as part of the approximate methods for calculating the posterior probability of a motif starting in each of the positions in the N sequences, given our parameter estimates. As described in more detail below, these posterior probabilities form the basis for predicting motif sites. It is advantageous to base their calculation on the exact TCM likelihood rather than the likelihood under the approximate model since we avoid, for example, the possibility of declaring two overlapping motifs. Below, we report simulation results that compare the performance of the exact TCM method to that of the various approximate ones.

3 Constraints

Imposing constraints on the position weight matrix, derived from prior biological knowledge, corresponds to reducing the parameter space that is to be searched. Hence such constraints can be viewed as part of the statistical model that is to be applied to the data at hand. `COMODE` is very general in terms of the constraints that can be imposed on the position weight matrix. However, it requires that the user supply a `C` function that evaluates the constraints for a given candidate position weight matrix along with another `C` function that evaluates the derivative of the constraint functions at this position weight matrix. `cosmo` is

based on a more user-friendly, but still very flexible system for specifying constraints that does not require the user to code `C` functions. Instead, the user supplies a description of the constraints in a text file that follows a certain standard format. We next describe the different types of constraints that can be imposed on the position weight matrix, while leaving a detailed description of the format for specifying them in the constraint file to appendix A.

3.1 Motif intervals

Many motifs can be conceptually divided into separate intervals that each correspond to a distinct set of constraints on the position weight matrix. In order to specify constraints for `cosmo`, we hence first specify how the motif can be divided into separate intervals. Since the true motif width is usually unknown, forcing `cosmo` to search a range of candidate values, we have to specify how the width of each interval changes with varying motif widths. We offer three possibilities: The length of an interval may be a fixed number of based pairs no matter what the length of the whole motif is; alternatively, the length of an interval may always be a fixed proportion of the length of the whole motif; finally, a motif may contain one interval that for each motif width is assigned whatever number of base pairs is left after all intervals of the first two kinds have been allocated.

Let I_k denote the positions of the motif that fall into interval k, $I_k = \{w_1(k), ..., w_{p_k}(k)\}$. Once the motif has been divided into separate intervals $I_1, ..., I_d$, we can add a number of different constraints to individual intervals or to the motif as a whole.

3.2 Bound constraints on the information content across an interval

An important summary measure of a given position weight matrix is its information content profile (Schneider et al., 1986). The information content at position w of the motif is given by

$$IC(w) = \log_2(J) + \sum_{j=1}^{J} p_{wj} \log_2(p_{wj}) = \log_2(J) - entropy(w)$$

where J denotes the number of letters in the alphabet from which the sequences have been derived so that here $J = 4$. The information content is measured in bits and, in the case of DNA sequences, ranges from 0 to 2 bits. A position in the motif at which all nucleotides occur with equal probability has an information content of 0 bits, while a position at which only a single nucleotide can occur has an information content of 2 bits. The information content at a given position can therefore be thought of as giving a measure of the tolerance for substitutions in that position: Positions that are highly conserved and thus have a low tolerance for substitutions correspond to high information content, while positions with a high tolerance for substitutions correspond to low information content.

Mirny and Gelfand (2002) have shown that the information content at a given position of a motif is proportional to the number of contacts between the protein and the base pair at that position. We therefore expect higher information content in regions of the motif that

are bound by the transcription factor than in the remaining regions. If the transcription factor contains two DNA-binding domains whose target sequences in the motif are separated by a short stretch of sequence that does not interact with the protein, we would expect that the information content of the motif follows a high-low-high pattern. In this case, it may be useful to give bounds IC_{low} and IC_{up} on the information content across an individual interval k. This corresponds to the constraints

$$IC_{low} \leq IC(w) \leq IC_{up} \ , \quad w \in I_k$$

3.3 Shape constraints on the information content profile across an interval

We may want to exclude position weight matrices from consideration whose information content profile is sharply discontinuous across a given interval k. This can be achieved by requiring the information content profile across that interval to follow a linear or monotone shape. Both of these functional forms are parameterized by the information content at the left edge, $IC(w_1(k))$, and right edge, $IC(w_{p_k}(k))$, of the interval. In particular, requiring a linear information content profile corresponds to the constraints

$$IC(w) = IC(w_1(k)) + \frac{w - w_1(k)}{w_{p_k}(k) - w_1(k)}(IC(w_{p_k}(k)) - IC(w_1(k))) \ , \quad w \in I_k$$

A monotone increasing information content profile corresponds to the constraints

$$IC(w) \geq IC(w-1) \ , \quad w \in I_k \setminus w_1(k)$$

A monotone decreasing information content profile corresponds to the constraints

$$IC(w) \leq IC(w-1) \ , \quad w \in I_k \setminus w_1(k)$$

In each of these cases, we may give bounds on $IC(w_1(k)$ and $IC(w_{p_k}(k))$. Furthermore, we may relax each of these constraints by specifying an error tolerance ϵ that gives an upper limit on deviations from the specified shape at any given position in the interval. In the linear case, for example, this would require

$$-\epsilon \leq IC(w) - IC(w_1(k)) - \frac{w - w_1(k)}{w_{p_k}(k) - w_1(k)}(IC(w_{p_k}(k)) - IC(w_1(k))) \leq \epsilon \ , \quad w \in I_k$$

3.4 Lower bounds on nucleotide frequencies across an interval

We may suspect that a given nucleotide occurs with high frequency across a certain interval k. In that case, we may require that the average frequency of a given nucleotide j across all positions in interval k is no less than some lower bound p_{min}:

$$p_{min} \leq \frac{1}{w_{p_k}(k) - w_1(k) + 1} \sum_{w \in I_k} p_{wj} \leq 1$$

Similarly, we may require that the GC-content or AT-content across an interval is no less than some lower bound p_{min}. If the length of interval k does not change with varying motif width, we may also impose lower bounds for nucleotide frequencies at a single position $w_l(k)$ in that interval, $p_{w_l(k)j} \geq p_{min}$.

3.5 Palindromic intervals

If the DNA-binding domains of the transcription factor are homodimeric, the DNA stretches that are bound by the transcription factor will be palindromes of each other. `MEME` includes an option to require the entire motif to be palindromic. `cosmo` instead allows the user to specify two intervals k_1 and k_2 that are thought to be palindromic with respect to each other.

In particular, we require that the frequency of nucleotide j at position l in interval k_1 equal the frequency of the complement of nucleotide j, denoted by $c(j)$, at position l from the right edge of interval k_2:

$$p_{w_l(k_1)j} = p_{(w_{p_{k_2}}(k_2)-l+1)c(j)} \ , \quad l \in \{1, ..., p_{k_1}\}, \ j = 1, 2, 3, 4$$

Again, we can relax this constraint by specifying an error tolerance ϵ that gives an upper limit on deviations from the above equality:

$$-\epsilon \leq p_{w_l(k_1)j} - p_{(w_{p_{k_2}}(k_2)-l+1)c(j)} \leq \epsilon \ , \quad l \in \{1, ..., p_{k_1}\}, \ j = 1, 2, 3, 4$$

3.6 Submotifs

Families of transcription factors are often characterized by the occurrence of a certain submotif within the motif. The exact location of the submotif within the motif, however, can vary widely. DNA sequences bound by transcription factors with an ETS domain, for example, all contain the stretch GGAA somewhere within the binding site.

To specify such constraints, let the nucleotides in the submotif be denoted by $M = \{m_1, ..., m_d\}$. Then we would like to require that, for a specified minimum nucleotide frequency p_{min} such as $p_{min} = 0.8$,

$$p_{(w+l-1)m_l} \geq p_{min} \ , \quad l = 1, .., d$$

for some $w \in \{1, ..., W - l + 1\}$, i.e. we require that there exist a window of length d in the motif such that the nucleotides corresponding to the submotif occur in consecutive positions each with a frequency of at least p_{min}. This could be formulated as a single constraint of the form

$$g(PWM) \equiv \min_{w \in \{1,...,W-d+1\}} \sum_{l=1}^{d} [p_{min} - p_{(w+l-1)m_l}]_+ \leq 0,$$

where $[x]_+ \equiv \max(0, x)$ denotes truncation of x at 0. In this formulation of the constraint, we can think of $[p_{min} - p_{(w+l-1)m_l}]_+$ as a penalty for position $w + l - 1$ if we consider the window of length d starting at position w. Such a constraint requires then that there exists a window of length d such that the sum of all such penalties across the window is 0.

Unfortunately, such constraints do not work well in practice, presumably because the function g is not smooth enough for the sequential quadratic programming algorithm used to perform constrained maximization. We formulate the constraint instead as

$$\min_{w \in \{1,\ldots,W-l+1\}} \frac{1}{d} \sum_{l=1}^{d} e^{-5p_{(w+l-1)m_l}} \leq e^{-5p_{min}} \tag{1}$$

Thus we have introduced smoother penalties given by $e^{-5p_{(w+l-1)m_l}}$ and only require that there exists a window such that the average penalty across this window is no larger than the penalty evaluated at p_{min}. This no longer ensures that the original constraint

$$p_{(w+l-1)m_l} \geq p_{min} \ , \quad l = 1, .., d$$

is satisfied, but the approximation is reasonably close and works far better in the context of constrained maximization.

3.7 Bounds on differences of shape parameters

Sometimes we may wish to impose constraints on the shape of the information content that cannot be specified by the shape constraints described above. For example, we may wish to require that the information content across a certain interval k is constant, or that the information content profile be continuous at the junction between two intervals.

Such constraints can be formulated by giving bounds on the difference between two shape parameters. Recall that shape constraints on the information content profile across interval k are parameterized using the information content at the left and right edge of the interval, $IC(w_1(k))$ and $IC(w_{p_k}(k))$. Hence we may require that

$$0 \leq IC(w_1(k)) - IC(w_{p_k}(k)) \leq 0$$

corresponding to a constant information content profile across interval k. As another example, we might require that

$$0 \leq IC(w_1(k_1 + 1)) - IC(w_{p_1}(k_1)) \leq 0$$

corresponding to the constraint that the information content profile be continuous at the junction between interval k_1 and $k_1 + 1$. As a final example, we might specify a linear information content profile across interval k and require that

$$-2 \leq IC(w_1(k)) - IC(w_{p_k}(k)) \leq 0$$

corresponding to the constraint that the information content be increasing across the interval.

4 Model selection techniques for the union model

The probabilistic models described above are indexed by by following four parameters:

1. The order k of the background Markov model, $k \in \{1, 2, ...\}$.
2. The width of the motif W, $W \in \mathcal{W} = \{\texttt{minw}, ..., \texttt{maxw}\}$.
3. The type of model used to describe the data-generating process, $M \in \mathcal{M} \subset \{OOPS, ZOOPS, TCM\}$.
4. The set of constraints on the position weight matrix of the motif, $C \in \mathcal{C} = \{C_1, ..., C_d\}$.

Here `minw` and `maxw` are user-supplied bounds on the range of motif widths to consider. The user is given the choice, for each one of these four parameters, to either make a manual selection, presumably based on available *a priori* knowledge, or to have `cosmo` select the appropriate index data-adaptively. This last approach corresponds to working in the larger union model that only assumes that at least one of the models is true out of the entire collection of models indexed by the parameters that are chosen data-adaptively.

We next review some model selection techniques that can be used to select the index in a data-adaptive manner. As it turns out, some of the criteria described below will also feature in the estimation process for a fixed index.

4.1 Likelihood-based validity functionals

The models indexed by k and W are nested in the sense that models with a smaller index are special cases of models with a larger index. Likewise, the OOPS model is contained in the ZOOPS model. Finally, any model with a non-empty constraint set is contained in the corresponding model with an empty constraint set. In this context of a collection of nested models, the maximum-likelihood principle invariably leads to choosing the model of the highest dimension and can therefore not be used as a criterion for model selection.

For this reason, a number of model selection criteria have been proposed that penalize the likelihood function by some measure of the dimension of the model. These criteria generally take the form $I = -2\log(L) + q$, where q is a penalty and L denotes the likelihood function evaluated at the maximum-likelihood estimates. Two prominent examples are Akaike's Information Criterion AIC (Akaike, 1973) with $q = 2p$, where p denotes the number of parameters, and the Bayesian Information Criterion BIC (Schwarz, 1978) with $q = p\log(n)$, where n denotes the number of observations.

AIC and BIC are aimed at different model selection scenarios and hence their performance will depend on the particular case at hand. AIC is aimed at a situation in which the true model is high-dimensional, requiring many, possibly infinitely many, parameters to describe it. The goal is to find the best approximating model from a collection of lower-dimensional models. The dimension of this approximating model will be low if the amount of available data is small and will increase as more information becomes available. In particular, if

the collection of candidate models contains the true model, AIC has been shown to be an inconsistent estimator of the true dimension (Hannan, 1980; Woodroofe, 1982). In general, models selected by AIC tend to overfit the data.

By contrast, BIC is based on the assumption that the collection of low-dimensional candidate models contains the true model. In this context it has been shown to be a consistent estimator of the dimension of the model for a broad range of data-generating distributions (Haughton, 1988; Csiszar and Shields, 2000).

4.2 E-value of the resulting multiple alignment

`MEME` compares different models based on a measure of the statistical significance of the multiple alignment obtained by aligning the predicted motifs. It computes a test statistic based on a likelihood-ratio test for comparing the null hypothesis that the aligned motifs were generated by the background distribution to the alternative hypothesis that they were generated by the motif distribution. The measure of statistical significance used by `MEME` is then the expected number of multiple alignments with such a test statistic as great or greater than the observed one that can be formed from a set of sequences $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$ that was generated entirely from the background distribution. This E-value also gives an approximate p-value for the hypothesis that the given alignment was obtained from a set of sequences $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$ generated by the background distribution. We next review how `MEME` arrives at a set of predicted motifs and then computes the corresponding E-value, based on methods originally proposed by Hertz and Stormo (1999).

Given a candidate position weight matrix as well as a candidate value $\hat{\pi}$ or $\hat{\lambda}$ for the intensity parameter in the ZOOPS and TCM models, respectively, we obtain a set of predicted motif start sites $\hat{K}$ as follows. For each position l in a given sequence i, we calculate the posterior probability that $B_{il} = 1$ given our parameter estimates. Denote this posterior probability by $\tilde{p}_{il}$. Now a candidate value $\hat{\pi}$ in the ZOOPS model corresponds to an expected number of motif occurrences of $E = \hat{\pi} \times N$; a candidate value $\hat{\lambda}$ in the TCM model corresponds to an expected number of motif occurrences of $E \approx \hat{\lambda} \times \sum_{i=1}^{N} L_i$.

For the TCM model, estimate K by $\hat{K} = \{(i, l) : \tilde{p}_{il} \geq \tilde{p}_{(E)}\}$, i.e. by the sites with the E highest posterior probabilities $\tilde{p}_{il}$. For the OOPS model estimate K by the set $\hat{K} = \{(i, l) : \tilde{p}_{il} = \max_l \tilde{p}_{il}\}$ of positions with highest posterior probabilities in the individual sequences. Finally, for the ZOOPS model, choose the E sequences with highest posterior probabilities from the set $\hat{K}$ of estimated start sites in the corresponding OOPS model.

We next calculate a p-value $\tilde{p}(MA)$ for the likelihood ratio test of the hypothesis $\tilde{H}_0(MA)$ that the E aligned subsequences were generated by the background distribution. Note that this hypothesis depends on the chosen alignment MA. We then adjust this p-value to take into account that the E aligned subsequences were derived from the larger data set $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$, particularly with the aim of obtaining an alignment of subsequences that are unlikely to have been generated by the background distribution.

`MEME` approximates $\tilde{p}(MA)$ as follows. It computes a set of log-likelihood ratios LLR_w, $1 \leq w \leq W$, for comparing the null hypothesis that the E nucleotides in position w of these

sequences were generated from the background distribution to the alternative hypothesis that they were generated from the motif distribution. Next, it computes a set of p-values p_w for these W hypothesis tests based on an exact method proposed by Hertz and Stormo (1999). It now uses as test statistic the product of these W p-values. `MEME` arrives at an approximation to $\tilde{p}(MA)$ based on the observation that, under the null hypothesis $\tilde{H}_0(MA)$, this test statistic is the product of W independent random variables that each follow a uniform distribution on $[0, 1]$. The authors report that this approximation behaves very similar to the exact p-value in practice, which are computationally much more expensive to obtain.

This p-value is now adjusted as follows to take into account the fact that the E aligned subsequences were derived from the larger data set $\{\mathbf{X}_1, ..., \mathbf{X}_n\}$. For each of the three models (OOPS, ZOOPS, TCM), we can calculate the number A of possible alignments that could have been formed from the original data set. A will differ between the three models since sequences are allowed to contain different numbers of occurrences of the motif (see Hertz and Stormo (1999) for details). Furthermore, A is a function of the chosen motif width W. If each of the A alignments were independent of each of the other possible alignments, the probability that a data set generated entirely from the background distribution would give rise to at least one multiple alignment MA with a log-likelihood ratio $\tilde{L}(MA)$ for $\tilde{H}_0(MA)$ as large or larger than the observed log-likelihood ratio $\tilde{l}(MA)$ could be computed as

$$\begin{aligned} P(\tilde{L}(MA) \geq \tilde{l}(MA) \; \forall \; MA \in \{\mathbf{X}_1, ..., \mathbf{X}_n\}) &= 1 - P(\tilde{L}(MA) < \tilde{l}(MA))^A \\ &= 1 - (1 - \tilde{p}(MA))^A \\ &\approx 1 - \exp\left(-A\tilde{p}(MA)\right) \\ &\approx A\tilde{p}(MA) \end{aligned}$$

`MEME` now uses the quantity $A\tilde{p}(MA)$ to measure the statistical significance of the aligned motifs. It gives an approximation to the expected number of alignments to observe under the background distribution that have log-likelihood ratios $\tilde{L}(MA)$ as great or greater than the observed $\tilde{l}(MA)$, i.e. an approximation to the E-value of the multiple alignment. `cosmo` uses the same algorithm for computing E-values as does `MEME`.

4.3 Likelihood-based cross-validation

Cross-validation is a general approach for selecting among candidate models that is based on dividing the original data set into a training set that is used to estimate the parameters of a given model and a validation set that is then used to evaluate the performance of this estimated model. Likelihood-based cross-validation uses as a criterion for this second step the value of the likelihood function, using the parameter estimates obtained from the training sample, evaluated at the observations in the validation sample. Specifically, given a collection of candidate models, indexed by $h \in \mathcal{H}$, V-fold likelihood-based cross-validation makes its choice from this collection as follows. First, the data set is split into V groups by

drawing G_i from a uniform distribution on $\{1, 2, ..., V\}$, for $i = 1, ..., N$. For $v = 1, ..., V$, let $\mathcal{T}_v \equiv \{i : G_i \neq v\}$ and $\mathcal{V}_v \equiv \{i : G_i = v\}$ denote the v-th training and validation samples, respectively. Let $\mathcal{P}_h^v$ denote an estimate of the data-generating distribution based on training sample v and the model with index h. The choice for the index h is then given by

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{V} \sum_{v=1}^{V} \frac{1}{\sum_{i=1}^{N} I(G_i = v)} \sum_{i \in \mathcal{V}_v} -\log Pr(X_i | \mathcal{P}_h^v),$$

where $Pr(X_i|\mathcal{P}_h^v)$ is the likelihood of observation X_i under the estimated data-generating distribution $\mathcal{P}_h^v$.

Like AIC, likelihood-based cross-validation is aimed at situations in which the true data-generating distribution is believed to be a member of a very large model such as the non-parametric model. The goal is to select from a collection of lower-dimensional models the one that, given the amount of data available, best approximates the true density. As with AIC, the dimension of this approximating model will be low if the amount of available data is small and will increase as more information becomes available.

van der Laan et al. (2003) recently showed that likelihood-based cross-validation performs asymptotically as well as an optimal benchmark model selector that depends on the true density. One of the hypotheses for this result is that the candidate density estimates are bounded away from zero and infinity. We note that this assumption may be violated when the estimated position weight matrix contains entries very close to zero. Especially in the case of the OOPS model, it is then possible for the observed data likelihood to approach zero if it so happens that all of the motif likelihoods

$$\frac{1}{2}\Big[\prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)} + \prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=c(j))}\Big]$$

for a given sequence are close to zero. For the TCM model as well as the ZOOPS model with $\pi < 1$, position weight matrices with entries close to zero are less likely to lead to an unbounded loss function since the contributions from the background will ensure a non-zero likelihood. In the TCM model, for example, the likelihood function is a sum over all possible sample paths, in particular the one that consists of all background nucleotides.

The potential problem of an unbounded loss function in the OOPS model may be addressed by truncation as follows. For a given parameter estimate, we can calculate say the 0.9 quantile over all N sequences of the loss function evaluated at this estimate. In estimating the expected loss by computing the empirical mean over the validation sample, we then truncate any observed loss by this quantile to prevent sequences with likelihoods close to zero from exerting undue influence.

4.4 Cross-validation based on the Euclidean norm

As mentioned above, likelihood-based cross-validation is aimed specifically at density estimation. If the parameter of interest is only a low-dimensional functional of the density rather

than the whole density, it is often advantageous to target the loss function directly at this parameter, rather than estimating the density first and then obtaining the parameter estimate by integrating out the components that are not of interest. While the latter approach is asymptotically efficient in parametric models, it suffers considerably in semi-parametric and non-parametric models. Its performance is further compromised even in parametric models if the size of the model is large as compared to the number of observations.

In the context of constrained motif detection, the primary parameter of interest is the position weight matrix. For selection problems in which W is fixed, notably in selecting between candidate constraint sets, we may hence want to use the following loss function that is targeted directly at the position weight matrix. Let $P_{wj}^{-v}(h)$ denote the estimate of the position weight matrix based on training sample v and constraint set h. Let $P_{wj}^{v}(0)$ denote the unconstrained estimate of the position weight matrix based on validation sample v. Then

$$\frac{1}{V}\sum_{v=1}^{V}\sum_{w=1}^{W}\sum_{j=1}^{4}\left(P_{wj}^{-v}(h)-P_{wj}^{v}(0)\right)^2$$

gives an unbiased estimate of the Euclidean norm between the true position weight matrix and the position weight matrix estimated in constraint set h. Hence its expectation is uniquely minimized by the true position weight matrix so that it can be used as a loss function for the sake of estimating the position weight matrix. For this method, we use 2-fold cross-validation since it relies on sensible parameter estimates from each validation sample.

5 Simulation study design

5.1 Test data sets

We assessed the performance of different candidate versions of `cosmo` on six collections of simulated data sets, which we will refer to below as dOOPS, dZOOPS1, dZOOPS2, dTCM1, dTCM2, and dTCM3. Each of these collections contains 255 data sets, each consisting of 25 750-bp sequences. The background nucleotides of all sequences were simulated according to a third-order Markov model whose transition matrix we estimated from the human test sequences provided by Tompa et al. (2005). The target motifs inserted into these sequences consist of the 51 human transcription factor binding site profiles available in the curated JASPAR core database (Sandelin et al., 2004). This collection includes a wide spectrum of different motifs, both in terms of width and average information content: Their widths range from 5 to 20, with a median of 10, and their average information content ranges from 0.76 to 1.73, with a median of 1.21. For each collection of test data sets, we simulated five data sets for each of these motifs. The data sets simulated for a given motif only contain occurrences of that motif; no competing motifs were inserted. The dOOPS collection was generated by inserting motifs according to the OOPS random process; dZOOPS1 and dZOOPS2 were generated according to the ZOOPS random process with intensity parameters $\pi = 0.25$

and $\pi = 0.75$, respectively, corresponding to an expected number of 6.25 and 18.75 motif occurrences in the 25 sequences, respectively; dTCM1, dTCM2, and dTCM3 were generated according to the TCM random process with intensity parameters $\lambda = 0.00067$, $\lambda = 0.0013$, and $\lambda = 0.004$, respectively, corresponding to an expected number of 12.5, 25, and 75 motif occurrences in the 25 sequences, respectively.

5.2 Measuring performance

The performance of a particular algorithm on a collection of test data sets is assessed through its average sensitivity, positive predictive value (PPV), and receiver-operation characteristic (ROC), with each of these three measures computed at the site level rather than the nucleotide level. Specifically, following Tompa et al. (2005), we take a predicted site to identify a true site if it overlaps the true site by at least one quarter of the length of the true site. For a given data set, sensitivity is then defined as the proportion of all true sites that have been identified, and positive predictive value is defined as the proportion of true sites among the predicted sites. The ROC statistic is the integral of the ROC curve which plots sensitivity against the proportion of sites not representing a motif occurrence that have been falsely identified as a motif occurrence. To calculate this statistic, the discovered position weight matrix was used to compute posterior probabilities for a motif occurrence for all eligible sites. For a given true site, the overlapping predicted site with the highest posterior probability was defined as identifying that site, with all other overlapping predicted sites defined as misses. All sites were then ranked by posterior probability, and the integral of the ROC curve was determined by numerical integration using the trapezoid rule.

6 Estimation of parameters for fixed index

In this section, we describe how parameter estimates are obtained for a model with a given index. The following section then describes how this index is selected data-adaptively. As described in more detail in the following section, the background distribution is estimated in a preliminary, separate step. The parameters of this distribution are then fixed at their estimated values during the estimation process targeting the actual parameters of interest.

We begin this section with a short argument for why `cosmo` is based on maximum-likelihood estimates rather than Bayesian estimates as employed by `MEME`, followed by a description of how starting values for the constrained maximization of the likelihood function are created from the original data set. Next we discuss different approaches to estimating the intensity parameter in the ZOOPS and TCM models. Finally, we compare the exact TCM model to the various approximate models described above.

6.1 Maximum-likelihood vs. Bayes estimates

The current implementation of `MEME` is based on a modified EM-algorithm that finds the mean posterior probability estimates of the entries of the position weight matrix during the

M-step rather than the maximum-likelihood estimates (Bailey and Elkan, 1995b). For DNA sequences, it uses a Dirichlet prior distribution for the entries of the position weight matrix with parameter vector β equal to the average letter frequencies in the data set.

This deviation from the classical EM-algorithm is based on the following two problems in the context of motif detection. First, if any entry of the position weight matrix is ever estimated to be zero during an iteration of the EM-algorithm, it remains zero. Second, maximum-likelihood estimates tend to have a high variance in the presence of a weak signal or a small data set. MEME addresses both of these issues by using a prior distribution on the entries of the position weight matrix.

`cosmo` is not based on Bayesian estimation for a number of reasons. First, it does not use the EM-algorithm so that the first problem does not apply. Second, prior knowledge about the position weight matrix is incorporated explicitly through the use of constraints, which can be hoped to reduce the variance of the corresponding estimates. Third, applying Bayesian estimation in a fashion analogous to the one used by `MEME` in the context of an algorithm that directly maximizes the observed data likelihood without resorting to the EM-algorithm is likely to lead to final estimates that do in fact not satisfy the constraints originally imposed on the position weight matrix. Lastly, a few simulation studies showed that, even in the context of unconstrained motif detection with `cosmo`, there appears to be little or no benefit in postulating a prior distribution. This is in agreement with observations by Bailey and Elkan (1995b) who report that prior distributions are mostly of use for the detection of motifs in protein sequences, rather than DNA sequences. Maximum-likelihood estimates, on the other hand, can be expected to be asymptotically efficient since the models considered here are parametric.

For these reasons, `cosmo` estimates the parameters of interest by performing a constrained maximization of the likelihood function. As shown by Keleş et al. (2003), it is computationally advantageous to maximize this likelihood function directly rather than to use the EM-algorithm with a constrained maximization during each M-step. `cosmo` performs this maximization using the `C` function `donlp2()` by Spellucci (1996) (see appendix B.1).

6.2 Choice of starting values

Starting values for the entries of the position weight matrix are often created from a nucleotide sequence $\mathbf{S} = (S_1, ..., S_W)$ of length W by the mapping

$$PWM_{ij}(\mathbf{S}) = \begin{cases} p_c & \text{if} \quad S_i = j \\ (1 - p_c)/3 & \text{if} \quad S_i \neq j \end{cases}$$

where common choices for p_c are $p_c = 0.5$ or $p_c = 0.7$. Keleş et al. propose to obtain starting values for the constrained maximization routine by evaluating the model likelihood for all 4^W candidate position weight matrices that can be generated by the above mapping from the set of all possible length W sequences and choosing the `k` candidates that yield the highest likelihood values. The authors show that starting values obtained in this manner perform well in practice. However, the usefulness of this approach is limited by its enormous computational time and space requirements, especially for larger candidate values of W.

Bailey and Elkan (1995a) propose to choose starting values only from the set of candidate position weight matrices that can be generated by the above mapping from the set of all length W subsequences that actually occur in the observed sequences $\mathbf{X}_1, ..., \mathbf{X}_N$. This proposal is based on the reasoning that the sequences are likely to contain a number of length W subsequences that are close to the consensus sequence of the common motif so that the position weight matrix derived from these subsequences is likely to be close to the position weight matrix of that motif.

A possible modification of the proposal by Keleş et al. thus consists of choosing the `k` candidate position weight matrices that yield the largest likelihood values among the set of all candidate position weight matrices that can be derived from the original sequences. The current implementation of `MEME` instead performs a single E-step for each derived position weight matrix, aligns the predicted motifs, and chooses as starting value for the position weight matrix the empirical distribution of the alignment with the smallest E-value.

We examine the performance of these two different approaches on the six collections of test data sets described in section 5.1. For both approaches, a range of different candidate values are considered for the number of starting values `k` to use for each optimization. For the sake of simplicity, we treat the motif width W and the model type as known in these simulations. `cosmo` is run without any constraints on the position weight matrix.

Table 1: Mean performance statistics for different choices of starting values.

Starts	**dOOPS**			**dZOOPS1**			**dZOOPS2**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME									
1	0.34	0.34	0.92	0.10	0.09	0.90	0.25	0.26	0.91
E-value									
1	0.54	0.54	0.96	0.20	0.15	0.94	0.40	0.39	0.96
3	0.56	0.56	0.96	0.22	0.15	0.94	0.42	0.40	0.96
5	0.57	0.57	0.97	0.23	0.16	0.95	0.43	0.40	0.96
10	0.59	0.59	0.97	0.24	0.17	0.95	0.46	0.42	0.96
25	0.61	0.61	0.97	0.26	0.17	0.95	0.50	0.44	0.97
Likelihood									
1	0.18	0.18	0.94	0.06	0.04	0.94	0.09	0.09	0.94
3	0.26	0.26	0.95	0.09	0.06	0.94	0.15	0.14	0.95
5	0.32	0.32	0.95	0.10	0.06	0.94	0.19	0.18	0.95
10	0.39	0.39	0.96	0.13	0.08	0.95	0.26	0.23	0.95
25	0.51	0.51	0.97	0.16	0.10	0.95	0.37	0.34	0.96

Tables 1 and 2 summarize the results of this simulation study. We note that E-value based starting values lead to consistently better performance than can be achieved with likelihood-based starting values. As is to be expected, the performance of `cosmo` improves

Table 2: Mean performance statistics for different choices of starting values.

Starts	**dTCM1**			**dTCM2**			**dTCM3**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME									
1	0.14	0.13	0.89	0.29	0.33	0.90	0.42	0.68	0.93
E-value									
1	0.29	0.19	0.95	0.47	0.41	0.96	0.52	0.76	0.98
3	0.31	0.20	0.95	0.51	0.44	0.97	0.53	0.77	0.98
5	0.33	0.22	0.96	0.51	0.44	0.97	0.54	0.79	0.98
10	0.34	0.22	0.96	0.53	0.46	0.97	0.55	0.79	0.99
25	0.36	0.23	0.96	0.54	0.47	0.97	0.56	0.80	0.99
Likelihood									
1	0.12	0.06	0.95	0.24	0.21	0.95	0.35	0.56	0.97
3	0.20	0.11	0.95	0.32	0.28	0.96	0.43	0.68	0.98
5	0.22	0.13	0.95	0.39	0.33	0.96	0.47	0.72	0.98
10	0.27	0.14	0.95	0.46	0.39	0.97	0.52	0.76	0.99
25	0.33	0.19	0.96	0.52	0.44	0.97	0.55	0.78	0.99

somewhat as the number of starting values `k` is increased. Using 25 starting values instead of a single starting value improves the sensitivity of the algorithm by 7 to 29%; improvements in positive predictive value lie in the range from 4 to 20%. In both cases, the algorithm is most sensitive to the number of starting values used when the data set contains a relatively weak signal, as in the dZOOPS1 and dTCM1 data sets. Since the time requirement of the algorithm scales linearly in the number of starting values used, a reasonable trade-off between performance and computing time is needed. The default setting of `cosmo` therefore uses five starting values. We note, however, that users may wish to increase this number for smaller jobs to achieve better performance, or decrease it if reductions in computing time are needed.

The default setting of five starting values allows `cosmo` to achieve mean sensitivities that are 1.3 to 2.4 times greater than those achieved by `MEME`, with simultaneous mean positive predictive value improvements in the range from 1.2- to 1.7-fold. In fact, even a version of `cosmo` based on a single starting value outperforms `MEME` on all test cases considered here. A possible explanation for this somewhat surprising performance differential might be that, while both algorithms are based on the same two-component multinomial mixture model, only `cosmo` reports true maximum-likelihood estimates which are known to be asymptotically efficient in this parametric model. `MEME` deviates from the maximum-likelihood principle, for instance, in using an M-step that is based on mean posterior probability estimates (see 6.1) as well as in estimating the intensity parameter in the ZOOPS and TCM models based on the E-value criterion (see 6.3).

6.3 Estimation of the intensity parameter in the ZOOPS and TCM model

`COMODE` estimates the intensity parameters π and λ in the ZOOPS and TCM model by maximum likelihood. Bailey and Elkan (1995a) propose instead to estimate these intensity parameters based on the E-value of the aligned predicted motifs. We next describe a number of different approaches incorporating aspects of these two ideas for estimating the intensity parameter λ in the TCM model; estimation of π in the ZOOPS model is carried out in an analogous fashion.

We first choose a small number of candidate values for the expected number of motif occurrences over the whole data set. By default, the lowest number of expected motif occurrences to consider, `minSites`, is set to two, and the highest number of expected motif occurrences to consider, `maxSites`, is set to the minimum of 50 and five times the number of sequences. Candidate values for the expected number of motif occurrences are then generated as a geometric progression from `minSites` to `maxSites`, with each following candidate value being twice as large as the current one. Next, we map these expected numbers of motif occurrences into a set $\Lambda_1 = \{\lambda_1, ..., \lambda_d\}$ of candidate values for λ. Let $\Lambda_2 \supset \Lambda_1$ denote the larger set of candidate values for λ obtained from the set of all integer candidate values for the expected number of motif occurrences between `minSites` and `maxSites`, $\{\texttt{minSites}, \texttt{minSites} + 1, ..., \texttt{maxSites}\}$.

`MEME` now obtains an estimate $\hat{PWM}_k$ of the position weight matrix for each candidate value λ_k, holding λ fixed at λ_k. It then selects from the collection of estimates $\{(\hat{PWM}_k, \lambda_k) : k\}$ that pair which minimizes the E-value criterion. In a last step, `MEME` arrives at final estimates $(\hat{PWM}, \hat{\lambda})$ by holding the selected estimate of the position weight matrix fixed and selecting that $\hat{\lambda} \in \Lambda_2$ that minimizes the E-value for the corresponding multiple alignment. The E-value based estimator we study here modifies the algorithm used by `MEME` only slightly in that it carries out this last update step for the estimate of λ for each pair $(\hat{PWM}_k, \lambda_k)$, before selecting a pair based on the E-value criterion, rather than only once at the end of the algorithm. We observed that this modification lead to a moderate improvement in performance of the estimator.

Other estimators may be defined by following the approach taken by `MEME` of holding λ constant while estimating the position weight matrix, but then selecting the final estimates based on criteria other than the E-value of the corresponding multiple alignment. If we use the value of the likelihood function for this purpose, we obtain a profile-likelihood estimator of the intensity parameter. Alternatively, we may use likelihood-based cross-validation or truncated likelihood-based cross-validation.

The full maximum-likelihood estimator (MLE), lastly, obtains maximum-likelihood estimates $(\hat{PWM}_k, \hat{\lambda}_k)$ of the position weight matrix as well as of λ for each candidate value $\lambda_k \in \Lambda_1$, with Λ_1 now representing no more than a set of possible starting values for the optimization routine. The final estimates $(\hat{PWM}, \hat{\lambda})$ are then given by that pair in the collection $\{(\hat{PWM}_k, \hat{\lambda}_k) : k\}$ that achieves the greatest value of the likelihood function.

We examine the performance of these different approaches for estimating λ on the collections of test data sets dTCM1, dTCM2, and dTCM3. The same approaches for estimating π

Table 3: Mean performance statistics for different approaches to estimating the intensity parameter.

	dOOPS			**dZOOPS1**			**dZOOPS2**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.37	0.43	0.92	0.10	0.09	0.90	0.25	0.26	0.91
MLE	0.58	0.59	0.97	0.22	0.15	0.94	0.41	0.41	0.96
Eval	0.57	0.58	0.96	0.18	0.15	0.93	0.41	0.38	0.94
Lik	0.59	0.59	0.97	0.23	0.16	0.95	0.43	0.40	0.96
likCV	0.39	0.60	0.96	0.16	0.20	0.93	0.37	0.45	0.95
trCV	0.44	0.60	0.96	0.18	0.19	0.94	0.37	0.44	0.95

Table 4: Mean performance statistics for different approaches to estimating the intensity parameter.

	dTCM1			**dTCM2**			**dTCM3**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.14	0.13	0.89	0.29	0.33	0.90	0.42	0.68	0.93
MLE	0.26	0.24	0.94	0.43	0.46	0.95	0.50	0.76	0.97
Eval	0.22	0.19	0.93	0.46	0.44	0.95	0.51	0.74	0.97
Lik	0.33	0.22	0.96	0.51	0.44	0.97	0.54	0.79	0.98
likCV	0.22	0.24	0.94	0.40	0.49	0.96	0.52	0.78	0.98
trCV	0.23	0.24	0.94	0.40	0.48	0.96	0.52	0.78	0.98

in the ZOOPS model are evaluated on dOOPS, dZOOPS1, and dZOOPS2. For the sake of simplicity, we treat the motif width W as known and run `cosmo` without any constraints on the position weight matrix. Tables 3 and 4 summarize the results of this simulation study.

Since the ZOOPS and TCM models are parametric, the MLE is asymptotically efficient for estimating the position weight matrix and the intensity parameter. Its finite-sample performance on the test data sets considered here also compares favorably to more *ad-hoc* estimators like the one based on the E-value criterion. Somewhat surprisingly, the profile-likelihood estimator, performs even better than the MLE. The two estimators are based on the same maximum-likelihood principle and differ only in the algorithmic approach taken to identify the parameter estimates that maximize the likelihood of the observed data. We speculate that holding the intensity parameter fixed while estimating the entries of the position weight matrix helps to improve the finite-sample performance of the MLE in the presence of a weak signal. In other simulations, we have seen that the full MLE tends to overestimate the intensity parameter in such instances, leading to the identification of a spurious high-abundance motif, so that holding the intensity parameter fixed may help the chances of finding the true low-abundance motif.

The estimators that are based on cross-validation tend to select smaller values for the intensity parameter and thus predict fewer motif occurrences than the other estimators. Hence they tend to achieve a greater positive predictive value, but a smaller sensitivity than the other two likelihood-based estimators. This more conservative trade-off between sensitivity and specificity makes sense since each candidate estimator is evaluated on an independent validation sample rather than on the same data set that was used to obtain the estimates. The ROC statistics suggest that the performance of these estimators is overall comparable to that of the MLE.

Based on these simulation results, `cosmo` defaults to the profile-likelihood approach for estimating the intensity parameter.

6.4 Exact vs. approximate methods in the TCM model

Above we described three approximations to the exact TCM likelihood: The approximation used by `MEME` is based on applying a slightly modified ZOOPS likelihood to overlapping subsequences with length equal to the candidate motif width under consideration. The proposal by Keleş et al. (2003) more generally applies the ZOOPS likelihood to subsequences of length U. Finally, we proposed to derive subsequences of length $U + W - 1$ that overlap each other by $W - 1$ nucleotides to ensure that each possible motif start site in the original data set remains a possible motif start in the derived data set. In this section, we assess the performance of these three approximations relative to the exact approach.

We note that `cosmo` only makes use of an approximation to the TCM likelihood for the purpose of maximizing the likelihood function. Starting values as well as posterior probabilities calculated for declaring motif sites are always based on the exact TCM likelihood since these two steps are computationally inexpensive.

We compare the performance of the three approximations to the TCM likelihood to that of the exact method on the three collections of test data sets dTCM1, dTCM2, and dTCM3.

For the sake of simplicity, we treat the motif width W as well as the model type as known and run `cosmo` without any constraints on the position weight matrix.

Table 5: Mean performance statistics for different approaches to evaluating the TCM likelihood function. Keleş refers to the proposal in Keleş et al. (2003). Bembom refers to the proposal made here.

	dTCM1			**dTCM2**			**dTCM3**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.14	0.13	0.89	0.29	0.33	0.90	0.42	0.68	0.93
Keleş	0.31	0.21	0.95	0.50	0.44	0.97	0.54	0.78	0.99
Bembom	0.33	0.22	0.96	0.51	0.44	0.97	0.54	0.79	0.98
Exact	0.34	0.22	0.95	0.52	0.44	0.97	0.55	0.78	0.99

Table 5 shows the result of this simulations study. In all three test cases, the estimator based on the exact TCM likelihood achieves marginally greater mean sensitivities than the other estimators. The three `cosmo` variants behave very similarly in terms of positive predictive value and the ROC statistic. This suggests that any possible benefit of working with the exact TCM likelihood is too limited to warrant the increased computational burden. A possible explanation for this observation lies in the considerable sensitivity of the algorithm to the starting values used as well as the posterior probabilities calculated at the end to declare motif sites, two steps that, as mentioned above, are always based on the exact likelihood function. Among the two approximate `cosmo` variants, the one based on overlapping subsequences performs slightly better than the one based on the proposal by Keleş et al. (2003). `cosmo` therefore defaults to this estimator for the TCM model.

7 Data-adaptive selection of the index

In this section, we describe how `cosmo` chooses the various indices of the union model in a data-adaptive manner. The background distribution is estimated in a preliminary step by likelihood-based cross-validation. For the remaining parameters, we report simulation results for comparing a range of possible model selection techniques. Finally, we describe the approach we use when different parameters are to be chosen simultaneously based on different model selection techniques.

7.1 Background model

It is often desirable to estimate the parameters of the background Markov model from a larger, independent data set such as the entire set of intergenic regions of the organism of interest. Hence we estimate these parameters in a separate, preliminary step and fix them at their estimated values during the estimation process targeting the remaining parameters.

Of course, it is still possible to specify that the background parameters be estimated from the original set of input sequences.

Csiszar and Shields (2000) have shown that BIC is a consistent estimator of the order of a Markov model. In the present context, however, we do not assume that background nucleotides are actually generated according to a k-th order Markov model. Rather, we view these models as imperfect approximations to a true data-generating process that is allowed to be more complex. Hence it is more appropriate to select the order k of the background Markov model by AIC or likelihood-based cross-validation. Since the computational burden of this preliminary step is minimal, `cosmo` uses likelihood-based cross-validation.

7.2 Estimating the motif width W

The true width of the motif to be identified is generally not known *a priori*. The ability to choose this motif width in a data-adaptive manner is therefore of great importance for any motif detection algorithm. In this section, we report simulation results for comparing a number of model selection techniques that one might consider for this purpose, namely selection by maximum likelihood, AIC, BIC, the E-value of the aligned predicted motifs, likelihood-based cross-validation, and truncated likelihood-based cross-validation.

All candidate estimators are evaluated on the six different collections of test data sets described in section 5.1. For each data set, the candidate motif widths that are considered range from $W_0 - 3$ to $W_0 + 3$ base pairs, where W_0 is the true width of the inserted motif. For the sake of simplicity, the model type is treated as known, and `cosmo` is run without any constraints on the position weight matrix.

For each simulation, we also include results obtained from `MEME`, which are based on the following algorithm. First, candidate values of W are generated according to a geometric progression from `minw` to `maxw`. `MEME` chooses the model that minimizes the E-value of the aligned predicted motifs. This multiple alignment is then trimmed to produce the longest g-alignment of width at least `minw`, where a g-alignment is an alignment with no more than g gapped sequences per column. Values of g in $\{0, 1, ...\}$ are tried until an alignment of width at least `minw` is found. The number of motif occurrences is then adjusted to minimize the E-value of the alignment, followed by a final trimming step aimed at optimizing the E-value further.

Tables 6 and 7 summarize the results of this simulation study. First we note that there does not appear to be a significant price for having to select the motif width W data-adaptively. For the sake of comparison, we have included a version of `cosmo` for which W is fixed at the true value. In many cases, data-adaptive estimators in fact outperform this reference estimator in terms of mean sensitivity, positive predictive value, and ROC statistic. We note that this observation is likely tied to the decision of only requiring a predicted site to overlap the true site by one quarter the length of the true site in order to be considered a hit. Thus, an algorithm may perform well even if the selected motif width does not match the true motif width.

Among all data-adaptive candidate versions of `cosmo`, the estimator that selects W based on BIC leads to the most favorable overall performance. It achieves the highest mean sensi-

Table 6: Mean performance statistics for different approaches to selecting the motif width.

	dOOPS			**dZOOPS1**			**dZOOPS2**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.35	0.35	0.92	0.08	0.08	0.90	0.23	0.23	0.91
known	0.57	0.57	0.97	0.23	0.16	0.95	0.43	0.40	0.96
Lik	0.58	0.58	0.97	0.20	0.14	0.94	0.42	0.39	0.96
AIC	0.59	0.59	0.97	0.23	0.15	0.94	0.47	0.42	0.96
BIC	0.60	0.60	0.97	0.25	0.17	0.95	0.48	0.43	0.97
Eval	0.58	0.58	0.96	0.20	0.14	0.94	0.42	0.37	0.95
likCV	0.54	0.54	0.97	0.26	0.21	0.95	0.44	0.47	0.96
trCV	0.58	0.58	0.97	0.25	0.21	0.95	0.45	0.46	0.96

Table 7: Mean performance statistics for different approaches to selecting the motif width.

	dTCM1			**dTCM2**			**dTCM3**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.13	0.12	0.89	0.28	0.32	0.91	0.41	0.67	0.93
known	0.33	0.22	0.96	0.51	0.44	0.97	0.54	0.79	0.98
Lik	0.28	0.18	0.95	0.49	0.44	0.96	0.53	0.76	0.98
AIC	0.34	0.21	0.95	0.53	0.46	0.97	0.55	0.79	0.98
BIC	0.36	0.23	0.96	0.54	0.47	0.97	0.54	0.77	0.98
Eval	0.32	0.26	0.95	0.51	0.49	0.97	0.53	0.78	0.98
likCV	0.34	0.27	0.96	0.51	0.50	0.97	0.54	0.78	0.98
trCV	0.34	0.26	0.96	0.52	0.49	0.97	0.54	0.78	0.98

Table 8: Mean error in selected width.

	dOOPS	dZOOPS1	dZOOPS2	dTCM1	dTCM2	dTCM3
MEME	0.26	0.52	0.57	0.37	0.63	0.10
Lik	2.55	2.72	2.57	2.74	2.60	2.29
AIC	0.98	1.33	0.96	1.37	1.04	0.88
BIC	-0.84	-1.37	-1.23	-1.44	-0.96	-0.25
Eval	0.40	1.03	0.68	0.65	0.48	0.31
likCV	-1.44	-0.71	-0.62	-0.58	-0.37	0.18
trCV	-0.87	-0.41	-0.36	-0.09	-0.17	0.33

tivity in all test cases except for dZOOPS1 and dTCM3, where it comes in a close second. The cross-validation based estimators perform somewhat better in terms of mean positive predictive value, which, as before in section 6.3, can be attributed to a more conservative trade-off between sensitivity and specificity. The slight advantage of the BIC-based estimator in terms of mean ROC statistic, however, indicates that this latter estimator behaves somewhat better on the whole. In the dOOPS, dZOOPS1, and dZOOPS2 test cases, this estimator outperforms the E-value based estimator on all three measures of performance considered here. In the three TCM test cases, the E-value based estimator achieves a slightly higher mean positive predictive value, but does not match the performance of the BIC-based estimator in terms of mean sensitivity and ROC statistic.

The mean errors in the selected widths, reported in table 8, help to illustrate the behavior of the different estimators. As expected, maximum likelihood tends very strongly to select motif widths that are too large. In fact it is somewhat surprising that it does not select the largest width in all simulations. Perhaps this results from the constrained maximization routine failing to identify the true maximum-likelihood estimates, at least in a small number of instances. AIC likewise tends to overestimate the width of the unknown motif, although to a smaller extent. BIC and likelihood-based cross-validation, on the other hand, tend to underestimate the motif width somewhat. This behavior appears to be mostly attributable to those test data sets that contain a very weak signal and thus do not give the algorithm enough information to identify the unknown motif. In such instances, BIC as well as likelihood-based cross-validation err on the more conservative side of predicting shorter rather than longer motifs. On the whole, we would expect BIC to be a consistent estimator of the unknown motif width since this is a problem of selecting the correct dimension of the model.

We note that, in the dOOPS test case, likelihood-based cross-validation tends to underestimate the unknown motif width quite dramatically. Since the truncated version of this estimator selects motif widths closer to the truth and also achieves a better performance in terms of sensitivity, positive predictive value, and ROC statistics, this behavior is most likely due to the problem of a likelihood function that is not bounded away from zero, as described above. We conjecture that the problem of an unbounded loss function increases with increasing W since there are more possibilities for entries close to zero. As expected, truncating the loss function has minimal impact on likelihood-based cross-validation in the ZOOPS and TCM models.

Based on the results described in this section, `cosmo` defaults to selecting the motif width W based on BIC. The comparison of this estimator to `MEME` is even more favorable than in the case of a known motif width, with mean sensitivity and positive predictive value improvements consistently around two-fold. In the presence of weak signals, as in the dZOOPS1 and dTCM1 test cases, `cosmo` in fact achieves a three-fold greater mean sensitivity than `MEME`. Finally, we note that `cosmo`'s performance at selecting W can be expected to benefit significantly from constraints that the user may have imposed on the structure of the position weight matrix.

7.3 Selecting a model type

The distribution of motif occurrences among the sequences at hand may not be known *a priori*. In particular, one may often not be comfortable with the assumption that each sequence contains at most one occurrence of the motif. As a consequence, one would be forced, in such instances, to resort to the largest, most general model, the TCM model. However, one would expect an increase in performance associated with working in the smaller OOPS and ZOOPS models if their assumptions do happen to hold. Hence we are interested in model selection techniques that allow us to choose between the three different model types in a data-adaptive fashion.

In this section, we report simulation results for comparing a number of model selection techniques that one might consider for this purpose, namely model selection by maximum likelihood, AIC, BIC, the E-value of the aligned predicted motifs, likelihood-based cross-validation, and truncated likelihood-based cross-validation. Each candidate estimator was asked to select from among the three different model types on the six collections of test data sets described in section 5.1. We compare the performance of these estimators to that of `MEME` in the TCM model to examine the advantages and disadvantages of working in the larger union model. For the sake of simplicity, the motif width W is treated as known and `cosmo` is run without any constraints on the position weight matrix.

Table 9: Mean performance statistics for different approaches to selecting the model type.

	dOOPS			**dZOOPS1**			**dZOOPS2**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.31	0.36	0.90	0.09	0.09	0.88	0.23	0.23	0.89
OOPS	0.57	0.57	0.97	0.16	0.07	0.94	0.40	0.28	0.95
ZOOPS	0.59	0.59	0.97	0.23	0.16	0.95	0.43	0.40	0.96
TCM	0.56	0.47	0.97	0.29	0.16	0.95	0.44	0.37	0.96
Lik	0.61	0.60	0.97	0.28	0.16	0.95	0.47	0.42	0.96
AIC	0.60	0.60	0.97	0.27	0.15	0.95	0.47	0.41	0.96
BIC	0.60	0.60	0.97	0.27	0.15	0.95	0.47	0.40	0.96
Eval	0.58	0.58	0.97	0.21	0.15	0.94	0.43	0.37	0.96
likCV	0.56	0.48	0.97	0.29	0.18	0.95	0.44	0.41	0.96
trCV	0.58	0.53	0.97	0.28	0.17	0.95	0.45	0.41	0.96

The results of this simulation are summarized in tables 9 and 10. For the sake of comparison, we have included three estimators that work within the smaller models in which the model type is set *a priori* rather than chosen data-adaptively. Somewhat surprisingly, the dOOPS, dZOOPS1, and dZOOPS2 test cases show that there is only a small price, if any, to be paid for working in the larger TCM model if in fact the OOPS or ZOOPS assumptions are satisfied. For dZOOPS1, the TCM model in fact leads to slightly better performance than the ZOOPS model, with the two models achieving comparable results for dZOOPS2.

Table 10: Mean performance statistics for different approaches to selecting the model type.

	dTCM1			**dTCM2**			**dTCM3**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.14	0.13	0.89	0.29	0.33	0.90	0.42	0.68	0.93
OOPS	0.14	0.07	0.94	0.28	0.28	0.95	0.26	0.72	0.98
ZOOPS	0.21	0.18	0.95	0.32	0.43	0.96	0.25	0.73	0.98
TCM	0.33	0.22	0.96	0.51	0.44	0.97	0.54	0.79	0.98
Lik	0.32	0.22	0.95	0.52	0.46	0.97	0.54	0.79	0.99
AIC	0.31	0.21	0.95	0.51	0.46	0.97	0.54	0.79	0.99
BIC	0.30	0.21	0.95	0.51	0.46	0.97	0.54	0.79	0.99
Eval	0.22	0.18	0.94	0.43	0.42	0.96	0.52	0.77	0.98
likCV	0.31	0.23	0.96	0.50	0.47	0.97	0.53	0.78	0.98
trCV	0.30	0.23	0.96	0.50	0.47	0.97	0.53	0.78	0.98

On dOOPS, the TCM model performs only slightly worse than the OOPS or ZOOPS model. The data-adaptive estimators we consider here appear to be able to capitalize on the favorable performance of the TCM estimator in some of these instances, allowing them in fact to outperform the OOPS and ZOOPS estimators on data sets satisfying their respective assumptions. The TCM test cases illustrate that the OOPS and ZOOPS estimators perform considerably worse than the TCM estimator if the ZOOPS assumption does not hold. The poor performance of these estimators, however, appears to have only a minimal effect on the data-adaptive estimators, which still achieve mean sensitivities, positive predictive values, and ROC statistics close to those of the TCM estimator in these cases. These observations show that data-adaptive estimators offer an improvement in performance over *a priori* estimators if the OOPS or ZOOPS assumptions are satisfied, with no appreciable drop in performance relative to the TCM estimator if these assumptions are violated.

The different data-adaptive estimators we consider achieve largely comparable levels of performance, with maximum likelihood enjoying perhaps a slight edge. Since this approach is also computationally attractive, we have made it the default criterion employed by `cosmo` for selecting between different model types. If the OOPS or ZOOPS assumptions hold, this estimator achieves 2- to 3-fold greater mean sensitivity than the `MEME` TCM estimator, with improvements for the TCM test cases in the range from 1.30 to 2.35.

7.4 Selecting a constraint set

The main distinguishing feature of `cosmo` is the ability to supervise the motif detection by incorporating prior knowledge in the form of constraints on the position weight matrix of the motif to be discovered. Since such constraints may not always be easy to define with certainty, we would like to allow the user to give a number of different constraint sets $C_1, ..., C_d$ and to choose the appropriate constraint set in a data-adaptive manner. This would correspond to

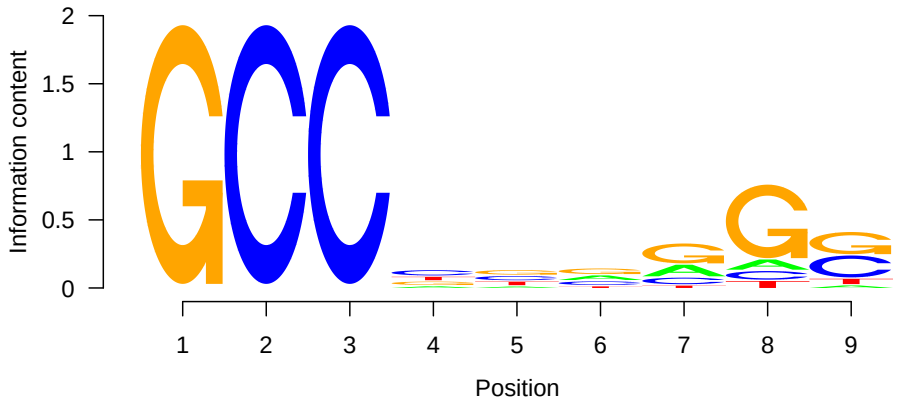

Figure 1: Sequence logo of one of the inserted motifs.

working in a larger model that only assumes that the true position weight matrix satisfies at least one of the supplied constraint sets. In particular, the user may wish to include an empty constraint set in the collection $C_1, ..., C_d$ to be protected from the risk of model mis-specification through the imposition of a wrong set of constraints on the position weight matrix.

In this section, we report simulation results for comparing a number of model selection techniques for the purpose of selecting between such different constraint sets. Apart from the model selection techniques used in previous sections, we also consider cross-validation based on the Euclidean norm between two position weight matrices. We do not, however, consider the penalized likelihood approaches based on AIC and BIC since there is no straightforward way to identify the dimension of a constrained model. In most cases, these methods can be expected to give results that are very similar to the unpenalized likelihood approach.

We evaluated each of these candidate estimators on the test data sets dOOPS, dZOOPS1, and dTCM1. For the sake of simplicity, the motif width W and the model type were treated as known. We examined the behavior of the different model selection approaches in two different scenarios. In both scenarios, `cosmo` is asked to choose between a non-trivial submotif constraint and the empty constraint set. In the first scenario, the non-trivial constraint set is correctly specified, with the submotif constraint based on the longest submotif contained in the inserted motif whose letters each roughly appear with frequency 0.9; specifically, we required that the value of the penalty function (1) be less than or equal to $e^{-5\times 0.9}$ for the selected submotif. In the second scenario, the non-trivial constraint set is incorrectly specified, based on the submotif of length four base pairs whose letters appear least frequently in the inserted motif; more precisely, we selected the submotif that maximizes the penalty function (1) for this purpose. For the motif shown in figure 1, for example, the correct constraint set requires that the identified motif contain the submotif "GCC", while the incorrect constraint set is based on the submotif "ATTT".

Tables 11-13 summarize the results of this simulation study. For the sake of comparison,

Table 11: Mean performance statistics for different approaches to selecting among constraint sets in the presence of a correctly specified constraint set.

	dOOPS			**dZOOPS1**			**dTCM1**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.34	0.34	0.92	0.10	0.09	0.90	0.14	0.13	0.89
Uncon	0.57	0.57	0.97	0.23	0.16	0.95	0.33	0.22	0.96
Con	0.65	0.65	0.98	0.39	0.25	0.97	0.47	0.31	0.98
Lik	0.62	0.62	0.97	0.29	0.19	0.95	0.37	0.23	0.96
Eval	0.60	0.60	0.97	0.29	0.21	0.96	0.44	0.31	0.97
likCV	0.62	0.62	0.98	0.36	0.25	0.97	0.43	0.31	0.97
trCV	0.63	0.63	0.98	0.34	0.24	0.96	0.43	0.30	0.97
pwmCV	0.57	0.57	0.97	0.24	0.17	0.95	0.37	0.25	0.97

Table 12: Mean performance statistics for different approaches to selecting among constraint sets in the presence of an incorrectly specified constraint set.

	dOOPS			**dZOOPS1**			**dTCM1**		
	Sens	PPV	ROC	Sens	PPV	ROC	Sens	PPV	ROC
MEME	0.34	0.34	0.92	0.10	0.09	0.90	0.14	0.13	0.89
Uncon	0.57	0.57	0.97	0.23	0.16	0.95	0.33	0.22	0.96
Con	0.27	0.27	0.92	0.13	0.11	0.94	0.16	0.13	0.94
Lik	0.57	0.57	0.97	0.23	0.16	0.95	0.33	0.22	0.96
Eval	0.57	0.57	0.96	0.21	0.16	0.94	0.29	0.22	0.95
likCV	0.48	0.48	0.96	0.21	0.16	0.94	0.32	0.23	0.95
trCV	0.55	0.55	0.96	0.22	0.16	0.94	0.31	0.22	0.95
pwmCV	0.57	0.57	0.97	0.22	0.16	0.95	0.31	0.21	0.95

Table 13: Proportion of times constraint set is chosen.

	dOOPS		**dZOOPS1**		**dTCM1**	
	Good	Bad	Good	Bad	Good	Bad
Lik	0.18	0.02	0.22	0.04	0.18	0.02
Eval	0.18	0.18	0.33	0.28	0.58	0.58
likCV	0.51	0.44	0.67	0.58	0.63	0.43
trCV	0.42	0.27	0.58	0.49	0.56	0.38
pwmCV	0.15	0.02	0.08	0.06	0.30	0.28

we have included three estimators that do not choose a constraint set data-adaptively, namely `MEME`, an unconstrained version of `cosmo`, and a version of `cosmo` that works only within the given non-trivial constraint set. The dZOOPS1 and dTCM1 test cases show that this latter estimator outperforms the unconstrained estimator considerably if the number of motif occurrences is small and the constraint set is correctly specified. The dOOPS test case shows that differences in performance are far less pronounced if the number of motif occurrences is large. As is to be expected, the constrained algorithm performs worse if the constraint set is incorrectly specified. These initial observations underscore the potential benefits of a constrained motif search and the need for a data-adaptive methodology for choosing between different constraint sets. On the whole, the methods we investigate for this purpose perform quite well in that they almost rise to the level of the constrained estimator if the constraint set is correctly specified while not sinking too far below the level of the unconstrained estimator if this is not the case. At the same time, no single method clearly outperforms all other methods on the test cases we consider.

The usefulness of the maximum-likelihood criterion in selecting between the constrained and unconstrained algorithm should be limited since its value in the larger, unconstrained model is guaranteed to be no less than in the smaller, constrained model. As seen in table 13, this selector is therefore heavily biased toward the unconstrained model. However, it still performs somewhat better than the unconstrained estimator if the constraint set is specified correctly. Presumably this is due to the numerical optimization routine succeeding more frequently in identifying the true maximum of the likelihood function in the smaller parameter space corresponding to the constrained search. This behavior makes the likelihood criterion a sensible conservative choice for instances in which the user suspects that the constraint set is mis-specified.

Truncated likelihood-based cross-validation offers perhaps the most appealing properties of all data-adaptive estimators we consider. It selects the constrained algorithm a considerable proportion of the time, allowing it to achieve levels of mean sensitivity, positive predictive value, and ROC statistic that are close to that of the constrained algorithm if the constraint set is correctly specified. At the same time it performs competitively in the presence of an incorrect constraint set, outperformed in the dZOOPS1 and dTCM1 test cases only by the conservative likelihood-based estimator. The E-value based estimator performs slightly better in the dTCM1 test case in the presence of a correct constraint set as well as in the dOOPS test case in the presence of an incorrect constraint set, but lags considerably in some of the other test cases. Truncated likelihood-based cross-validation also appears best able to distinguish between a correctly specified constraint set and an incorrectly specified one, especially if the data contain a fairly strong signal as in the dOOPS test case (table 13).

Truncation of the log-likelihood loss function greatly improves the performance of likelihood-based cross-validation in the dOOPS test case, cutting almost in half the proportion of times the incorrectly specified constraint set is selected. As before, this observation can be explained by the possibility of an unbounded loss function in the OOPS model. At the same time, truncation impacts the performance of the estimator only slightly in the dZOOPS1 and dTCM1 test cases.

Somewhat surprisingly, cross-validation based on the Euclidean norm between two position weight matrices offers in some ways the least desirable properties of all data-adaptive estimators we consider. Overall, it very rarely selects the constrained algorithm over the unconstrained algorithm, even if the constraint set is correctly specified. For test cases based on correct constraint sets, it thus consistently takes the last or second to last place, generally offering even less improvement than the very conservative likelihood-based estimator. At the same time, it does not even achieve this latter estimator's level of performance if the constraint set is mis-specified. The poor performance of this estimator as compared to likelihood-based cross-validation is somewhat surprising since it is based on a loss function that is directly targeted at the parameter of interest - the position weight matrix.

Based on these observations, `cosmo` chooses between different constraint sets by truncated likelihood-based cross-validation. In the presence of a weak signal, the mean sensitivity of this estimator is 3 to 3.5 times greater than that achieved by `MEME` when allowed to select between a correct constraint set and the empty constraint set. When asked to select between an incorrect constraint set and the empty constraint set, its mean sensitivity is still more than twice as great as that of `MEME`.

7.5 Separate model selection criteria for different parameters

In the previous sections, we examined the performance of various model selection techniques for the purpose of selecting a single fine-tuning parameter, with the remaining parameters fixed. In practice, the desired union model will generally be indexed by several choices for the motif width as well as for the model type and constraint set to use, making it necessary to simultaneously select a number of fine-tuning parameters. We do not want to require that all fine-tuning parameters are selected based on the same model selection criterion, but rather want to allow the user to specify a separate criterion for each parameter.

Thus suppose that the constraint set, model type, and motif width are to be chosen based on the respective criteria f_C, f_M, and f_W, with better choices in each case corresponding to smaller values of the criterion. Furthermore, recall that the corresponding sets of candidate parameter values are denoted by $\mathcal{C}$, $\mathcal{M}$, and $\mathcal{W}$. `cosmo` now selects these fine-tuning parameters as follows. For each given candidate constraint set C and model type M, it selects the optimal motif width $\hat{W}(C, M)$ by minimizing f_W:

$$\hat{W}(C, M) = \arg\min_{W \in \mathcal{W}} f_W(C, M, W)$$

In the next step, `cosmo` selects the optimal model type $\hat{M}(C)$ for each given constraint set C by minimizing f_M at the chosen value of $W = \hat{W}(C, M)$:

$$\hat{M}(C) = \arg\min_{M \in \mathcal{M}} f_M(C, M, \hat{W}(C, M))$$

Finally, the optimal constraint set $\hat{C}$ is chosen by minimizing f_C at the chosen values of $M = \hat{M}(C)$ and $W = \hat{W}(C, \hat{M}(C))$:

$$\hat{C} = \arg\min_{C \in \mathcal{C}} f_C(C, \hat{M}(C), \hat{W}(C, \hat{M}(C)))$$

This profiling approach allows for any combination of criteria for the different fine-tuning parameters. Furthermore, it is computationally advantageous since f_M and f_C need to be evaluated only for a subset of the candidate models. Specifically, f_M only needs to be evaluated for the candidate models

$$\mathcal{K}_M = \{(C, M, W) : C \in \mathcal{C}, M \in \mathcal{M}, W = \hat{W}(C, M)\}$$

with $\mid \mathcal{K}_M \mid = \mid \mathcal{C} \mid\mid \mathcal{M} \mid$, and f_C only needs to be evaluated for the candidate models

$$\mathcal{K}_C = \{(C, M, W) : C \in \mathcal{C}, M = \hat{M}(C), W = \hat{W}(C, \hat{M}(C))\}$$

with $\mid \mathcal{K}_C \mid = \mid \mathcal{C} \mid$.

The order in which the different model parameters are selected is in large part motivated by computational considerations. Since the selection among different constraint sets is computationally expensive due to the default reliance on likelihood-based cross-validation, this operation is best carried out once all other model parameters have already been identified. Such considerations suggest no particular order for selecting the motif width and model type since BIC as well as the likelihood are computationally easy to evaluate. Future research will investigate the impact of choosing a different order on the performance of the overall algorithm.

8 Software

A stand-alone version of `cosmo` can be downloaded at `http://cosmoweb.berkeley.edu/software.html`. An `R` package implementing this algorithm is available through Bioconductor (`http://bioconductor.org`). We furthermore created a web application `cosmoweb`, accessible at `http://cosmoweb.berkeley.edu`, that allows users to submit jobs through a simple web interface. Their jobs are then processed on a UC Berkeley server, with results posted in both HTML and XML format on a temporary web page. In addition to the detailed output obtained from the stand-alone version of `cosmo`, these results contain a sequence logo of the discovered motif as well as a plot of posterior probabilities along the entire sequences.

We next illustrate the use of `cosmoweb` through a simple example. The data set consists of 20 sequences that are each 200 nucleotides long and that each contain one occurrence of the motif with sequence logo given in figure 2. The motif is eight base pairs long, with high information content toward the edges and low information content in the middle. Furthermore we note that the two outer portions of the motif are palindromes of each other.

The first step of submitting our request to `cosmoweb` consists of pasting these input sequences, which are accessible at `http://cosmoweb.berkeley.edu/sample.seqs`, into the text box entitled '**actual sequences**'. Alternatively, we might specify the name of a file in which the sequences have been saved.

Suppose we knew *a priori* that the unknown motif represented the binding site of a homodimeric transcription factor. Then we might suspect a structure along the lines of what we described above and specify the following constraint set:

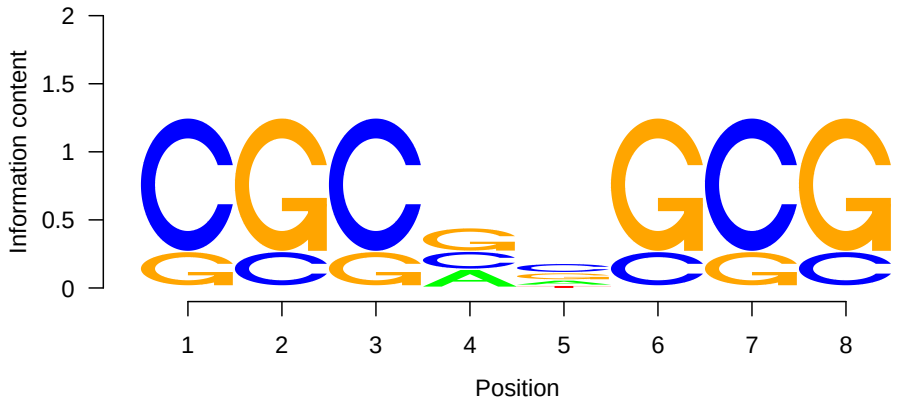

Figure 2: Sequence logo of inserted motif.

```
>IntervalSetup
Length: 3 bp
Length: variable
Length: 3 bp

>IcBounds
Interval: 1
Bounds: 1.0 to 2.0

>IcBounds
Interval: 2
Bounds: 0 to 0.8

>Pal
Intervals: 1 and 3
ErrorTol: 0.05
```

We paste this constraint set into the text box entitled '**actual constraint definitions**'. Alternatively, we might give the name of a file in which we have saved these definitions. To protect ourselves from the risk of specifying an incorrect constraint set, we will let `cosmo` choose between this constraint set and an unconstrained search by checking the option 'Add **unconstrained** case to given constraints'.

Suppose we have no prior knowledge about the distribution of sites among the individual sequences. We are thus forced to let `cosmo` select the appropriate distribution data-adaptively by checking both the 'ZOOPS' and 'TCM' options. Since there is generally no penalty for using the 'ZOOPS' model if in fact the 'OOPS' model is true, we need not check the 'OOPS' option. Suppose we have no prior knowledge about the total number of sites in the input

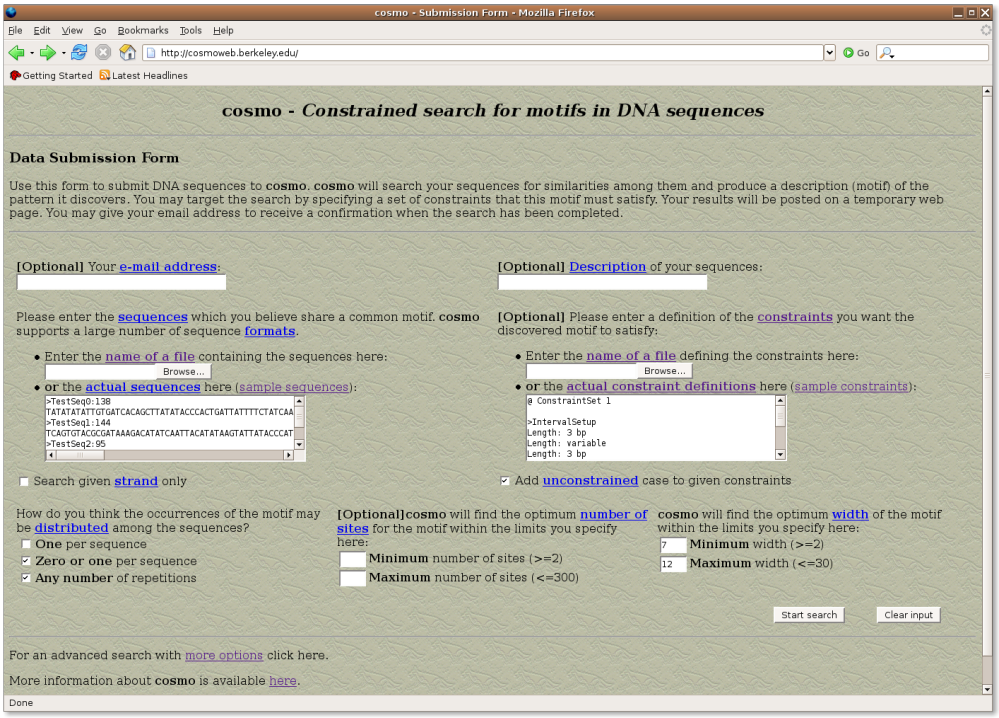

Figure 3: Interface for submitting jobs to `cosmoweb`.

sequences so that we are forced to leave the corresponding fields blank. Finally we may know that the width of the motif lies somewhere between 7 and 12, which we specify in the appropriate fields in the lower right-hand corner of the screen. Figure 3 shows the web page with all relevant information entered.

After submitting the job, we are given the name of a temporary web page on which the results will be posted on completion of the job. We are also given some summary statistics of the sequences we submitted as well as the option to check the progress of our job. The job described here takes about five minutes to be processed. Figure 4 now shows part of the output created by `cosmoweb`, consisting of the estimated position weight matrix, its sequence logo, as well as the alignment of predicted motif occurrences. The output furthermore contains information about the estimated background model, the considered candidate models, as well as a plot of posterior probabilities along the entire sequences. The output web page can be accessed at `http://cosmoweb.berkeley.edu/sample`.

Using the stand-along `C` version of `cosmo`, we would run this job using the command

```
cosmo sample.seqs -con confile -addfree -zoops -tcm -minw 7 -maxw 12
```

where `sample.seqs` is a file containing the input sequences and `confile` is a file containing the constraint definitions. The `-addfree` flag adds the unconstrained case to the collection of candidate constraint sets. The `R` package `cosmo` would use the command

```
cosmo('sample.seqs', 'confile', minW=7, maxW=12, models=c('ZOOPS','TCM'))
```

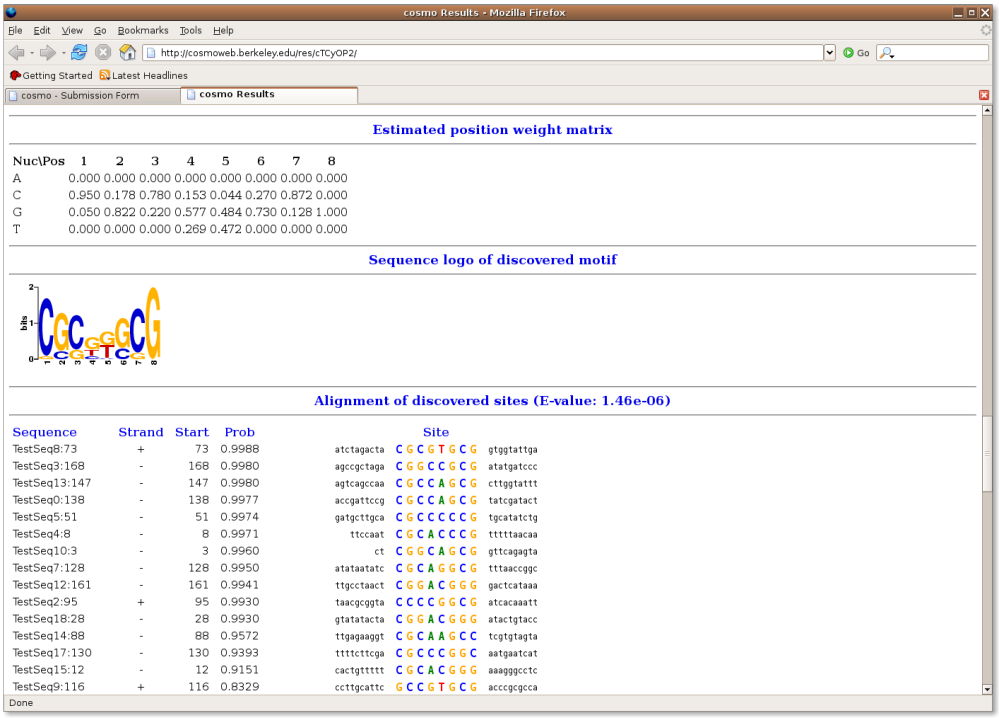

Figure 4: Output created by `cosmoweb`.

9 Discussion

In this article we present a motif detection algorithm that expands on existing methodology in that it is capable of working within a larger statistical model that encompasses the models used by previous approaches as submodels. Estimation within this larger model is made possible by allowing a number of model parameters to be chosen data-adaptively rather than having to be specified *a priori*. As a consequence of working in this larger model, our algorithm is able to rely on fewer assumptions than are necessary for previous algorithms.

We introduce a new model parameter by allowing the user to specify a collection of constraint sets for the position weight matrix to be discovered. As shown in the various simulation studies in this article as well as in Keleş et al. (2003), such constraint sets can considerably improve the performance of the algorithm in situations of low motif abundance. We furthermore demonstrate how the risk of model mis-specification can be controlled by including an empty constraint set in the collection of candidate constraint sets and allowing `cosmo` to choose the appropriate constraint set in a data-adaptive manner. By doing so, the algorithm is working within the larger unconstrained model in which it then aims to identify viable submodels. The data-adaptive selection of a constraint set is carried out by truncated likelihood-based cross-validation.

We are currently investigating to what extent the representative transcription factor familial binding profiles derived by Sandelin and Wassermann (2004) can be used to obtain corresponding representative constraint sets for the different structural classes of transcrip-

tion factors. The availability of such representative constraint sets would further improve the user-friendliness of `cosmo`.

The algorithm we present does not require the user to have prior knowledge about the distribution of motif occurrences among the input sequences. Rather, the maximum-likelihood principle can be used to select the appropriate model type data-adaptively. We show in simulation studies that this data-adaptive estimator outperforms the OOPS and ZOOPS estimators if their respective assumptions hold, while performing on the same level as the TCM estimator if these assumptions are violated.

Finally, unlike other current algorithms, `cosmo` does not require the order of the background Markov model to be specified *a priori*, but rather selects it data-adaptively by likelihood-based cross-validation. This approach optimizes the bias-variance trade-off such that more complex models are chosen as the amount of available data increases.

While `cosmo` is similar to `MEME` in many regards, the two algorithms differ in a number of important points. Instead of using the E-value criterion for estimating the intensity parameters in the ZOOPS and TCM models, `cosmo` employs a profile likelihood for this purpose, an approach that also leads to a moderate improvement in finite-sample performance over the asymptotically efficient maximum-likelihood estimator originally proposed by Keleş et al. (2003). `cosmo` furthermore employs a different approximation to the TCM likelihood that is shown to perform as well as an algorithm for evaluating this likelihood exactly. While `MEME` selects the width of the unknown motif based on the E-value criterion, `cosmo` here relies on the Bayesian Information Criterion, which has been shown to be consistent for selecting the dimension of a model. In addition, `cosmo` does not sample candidate widths in a geometric progression, but rather considers every candidate width between a lower and an upper bound.

The performance of `cosmo` compares favorably to that of `MEME`, even if the user supplies no constraints on the unknown position weight matrix. Our simulation studies demonstrate that `cosmo` achieves mean sensitivities in such cases that can be 2 to 3 times greater than those achieved by `MEME`, with simultaneous, albeit somewhat smaller improvements in mean positive predictive value. If the user supplies correctly specified constraints for data sets containing only a weak signal, we observed mean sensitivities that were 3 to 3.5 times greater than those achieved by `MEME`, even if `cosmo` is asked to select between the supplied constraint set and an unconstrained version of the algorithm in order to guard against the risk of model mis-specification.

We have implemented our algorithm in the form of a web application and a stand-alone C program, both accessible at `http://cosmoweb.berkeley.edu`, as well as in the form of an `R` package which is available through Bioconductor (`http://bioconductor.org`). As described in some detail in section B of the appendix, we have modified the original algorithm proposed by Keleş et al. (2003) in a number of places to achieve considerable speed improvements. Furthermore, we have introduced a more user-friendly way to specify a set of constraints on the position weight matrix that no longer requires the user to code these constraints up in the form of `C` functions.

Unlike other motif detection programs like `MEME` or `BioProspector` (Liu et al., 2001) that

can handle both DNA and protein sequences, `cosmo` is limited to DNA sequences. We are currently working on a version of `cosmo` that can be applied to sequences derived from a general alphabet, with DNA and protein sequences of course representing important special cases.

References

H. Akaike. *Information theory and an extension of the maximum likelihood principle.* Academiai Kiado, 1973.

T.L. Bailey. *Discovering motifs in DNA and protein sequences: The approximate common substring problem.* PhD thesis, University of California, San Diego, 1995.

T.L. Bailey and C.P. Elkan. Unsupervised learning of multiple motifs in biolpolymers using expectation maximization. *Machine Learning*, pages 51–80, 1995a.

T.L. Bailey and C.P. Elkan. The value of prior knowledge in discovering motifs with MEME. Technical Report CS95-413, Department of Computer Science, University of California, San Diego, 1995b.

H.J. Bussemaker, H. Li, and E.D. Siggia. Regulatory element detection using correlation with expression. *Nature Genetics*, 27:167–171, 2001.

I. Csiszar and P.C. Shields. The consistency of the BIC Markov order estimator. *The Annals of Statistics*, 28(6):1601–1619, 2000.

E. Davidson. *Genomic Regulatory Systems. Development and Evolution.* Academic Press, San Diego, 2001.

M.B. Eisen. All motifs are not created equal: structural properties of transcription factor - DNA interactions and the inference of sequences specificity. *Genome Biology*, 6:P7, 2005.

M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Science*, 95: 14863–14868, 1998.

J.W. Fickett and W.W. Wasserman. Discovery and modeling of transcriptional regulatory regions. *Current Opinions in Biotechnology*, 11:19–24, 2000.

E.J. Hannan. The estimation of the order of an ARMA process. *The Annals of Statistics*, 8:1071–1081, 1980.

D.M.A. Haughton. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16(1):342–355, 1988.

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

S. Keleş, M.J. van der Laan, S. Dudoit, B. Xing, and M.B. Eisen. Supervised detection of regulatory motifs in DNA sequences. *Statistical Applications in Genetics and Molecular Biology*, 2(1):5, 2003.

C. Lawrence and A. Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics*, 7:41–51, 1990.

C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

X. Liu, D. Brutlag, and J. Liu. BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Proceedings of the Pacific Symposium of Biocomputing*, 2001.

N.M. Luscombe, S.E. Austin, H.M. Berman, and J.M. Thornton. An overview of the structures of protein-DNA complexes. *Genome Biology*, 1:1–37, 2000.

L.A. Mirny and M.S. Gelfand. Structural analysis of conserved base pairs in protein-DNA complexes. *Nucleic Acid Research*, 30:1704–1711, 2002.

A.F. Neuwald, J.S. Liu, and C.E. Lawrence. Gibbs motif sampling: detection of bacterial outer membrane repeats. *Protein Science*, 4:1618–1632, 1995.

S. Ogg, S. Paradis, S. Gottlieb, G.I. Patterson, L. Lee, H.A. Tissenbaum, and G. Ruvkun. The Fork head transcription factor DAF-16 transduces insulin-like metabolic and longevity signals in *C. elegans*. *Nature*, 389:994–999, 1997.

J. Powell. SAGE. The serial analysis of gene expression. *Methods of Molecular Biology*, 99: 297–319, 2000.

F.P. Roth, J.D. Hughes, P.W. Estep, and G.M. Church. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nature Biotechnology*, 16:939–945, 1998.

A. Sandelin and W.W. Wassermann. Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics. *Journal of Molecular Biology*, 338:207–215, 2004.

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

T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht. Information content of binding sites on nucleotide sequences. *Journal of Molecular Biology*, 188:415–431, 1986.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

P. Spellucci. Solving general convex QP problems via an exact quadratic augmented Lagrangian with bound constraints. Technical report, TU Darmstadt, Department of Mathematics, 1996. URL http://www.mathematik.tu-darmstadt.de/pub/department/software/opti/qp.ps.gz.

W. Thompson, E.C. Rouchka, and C.E. Lawrence. Gibbs recursive sampler: finding transcription factor binding sites. *Nucleic Acid Research*, 31:3580–3585, 2003.

M. Tompa, N. Li, T.L. Bailey, G.M. Church, B. De Moor, E. Eskin, A.V. Favorov, M.C. Frith, Y. Fu, W.J. Kent, V.J. Makeev, A.A. Mironov, W.S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23:137–144, 2005.

M.J. van der Laan, S. Dudoit, and S. Keleş. Asymptotics optimality of likelihood based cross-validation. Technical Report 125, Division of Biostatistics, University of California, Berkeley, 2003. URL www.bepress.com/ucbbiostat/paper125.

E.W. van Zwet, K.J. Kechris, P.J. Bickel, and M.B. Eisen. Estimating motifs under order restrictions. *Statistical Applications in Genetics and Molecular Biology*, 4(1):Article 1, 2005.

M. Woodroofe. On model selection and the arc-sine laws. *The Annals of Statistics*, 10: 1182–1194, 1982.

C.T. Workman and G.D. Stormo. ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. In *Proceedings of the Pacific Symposium on Biocomputation*, pages 467–478, 2000.

A Format of the constraint file

In this section we describe the format that is used for specifying the constraints on the position weight matrix of the unknown motif.

A.1 Motif intervals

The division of the motif into separate intervals is specified in the constraint file by an entry like

```
>IntervalSetup
Length: 3 bp
Length: 30%
Length: variable
```

The entry has to start with the line `>IntervalSetup`. Each following line begins with the token `Length:` and sets up a new interval. The different interval types are then specified in the way shown above. In general, entries in the constraint file follow the above pattern in that they start with a line that gives the name of the action to be performed or the type of constraint to be added, followed by lines that are required to start with certain tokens of the form `Length:` that then specify the details of that action or constraint.

A.2 Bound constraints on the information content across an interval

A bound constraint on the information content profile across a given intervals is specified in the constraint file by an entry like

```
>IcBounds
Interval: 2
Bounds: 0 to 0.8
```

The entry has to start with the line `>IcBounds` or `>ICBounds`. The next line specifies which interval the bound constraint applies to. The last line gives the lower and upper bounds IC_{low} and IC_{up}, respectively.

A.3 Shape constraints on the information content profile across an interval

Shape constraints are specified in the constraint file by an entry like the following:

```
>IcShape
Interval: 1
Shape: Linear
```

```
LeftBounds: 1.0 to 2.0
RightBounds: 1.0 to 2.0
ErrorTol: 0.0
```

The entry has to start with the line `>IcShape` or `>ICShape`. The next line specifies which interval the shape constraint applies to. The following line specifies the functional form of the information content across that interval, with possible entries given by `Linear`, `MonotoneIncreasing`, and `MonotoneDecreasing`. The next two lines give bounds on $IC(w_1(k)$ and $IC(w_{p_k}(k))$, respectively. The last line sets the error tolerance ϵ.

A.4 Lower bounds on nucleotide frequencies across an interval

Nucleotide frequency constraints are specified in the constraint file by an entry like

```
>NucFreq
Interval: 2
Pos: all
Nuc: GC
LowerBound: 0.7
```

The entry has to start with the line `>NucFreq` or `>NucProb`. The next line specifies which interval the constraint applies to. The following line specifies a position in that interval, with the choice `all` or `avg` corresponding to requiring that the average nucleotide frequency across that interval be no less than the given lower bound. The following line specifies the nucleotides whose frequency is to be bounded from below, with possible entries given by `A`, `C`, `G`, `T`, `AT`, and `GC`. The last line finally gives the lower bound on the nucleotide frequency.

A.5 Palindromic intervals

A palindromic constraint is specified in the constraint file by an entry like

```
>Palindrome
Intervals: 1 and 3
ErrorTol: 0.1
```

The entry has to start with the line `>Palindrome` or `>Pal`. The next line gives the two intervals that are required to be palindromes of each other, and the last line defines the error tolerance ϵ.

A.6 Submotifs

A submotif constraint is specified in the constraint file by an entry like

```
>Submotif
Motif: GGAA
MinFreq: 0.90
```

The entry has to start with the line `>Submotif` or `>Sub`. The next two lines give the nucleotide sequence of the submotif and the approximate minimum probability p_{min}.

A.7 Bounds on differences of shape parameters

Constraints giving bounds on the difference between two shape parameters are specified in the constraint file by an entry like

```
>ParmDiff
Parameters: 2a - 1b
Bounds: -2 to 0
```

The entry has to start with the line `>ParmDiff` or `>ParameterDifference`. The next line defines the particular difference of shape parameters that we want to bound. Parameters are specified by the interval number followed by the letter `a` or `b`, denoting the left and right edge of the interval, respectively. The last line defines the bounds on this parameter difference.

A.8 Constraint file structure

A constraint file may contain the specifications for more than one constraint set. The beginning of a new constraint set is indicated through a line that starts with the character @. All commands that are encountered until the next line beginning with an @ are applied to the current constraint set. The only requirement on such constraint set sections is that they must contain the command `>IntervalSetup` to define the breakdown of the motif into intervals. Examples of valid constraint files can be found at `http://cosmoweb.berkeley.edu/constraints.html`.

B Computational improvements

In this appendix, we provide details on the computational improvements we made to the constrained motif search algorithm. These modifications lead to dramatically increased speed as compared to `COMODE`, with various test cases suggesting improvements on the order of 700-fold

B.1 Constrained maximization of the likelihood using `donlp2()`

`COMODE` relies on the proprietary NAG routine `E04UCF` for the constrained maximization of the likelihood function. We use the non-proprietary C function `donlp2()` written by Peter Spellucci (Spellucci, 1996) instead, allowing us to distribute the software freely for academic purposes. The function `donlp2()` is suited for the optimization of a non-linear, differentiable, real-valued function f subject to non-linear inequality and equality constraints. Specifically, it finds $\mathbf{x}^*$ such that

$$f(\mathbf{x}^*) = \min\{f(\mathbf{x}) : \mathbf{x} \in S \subset \mathbb{R}^n\}$$

where

$$S = \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u, \\ \mathbf{b}_l \leq \mathbf{A}\mathbf{x} \leq \mathbf{b}_u, \\ \mathbf{c}_l \leq c(\mathbf{x}) \leq \mathbf{c}_u\},$$

$\mathbf{A}$ is a matrix of dimension `nlin` $\times\, n$, and c is a vector-valued function $\mathbb{R}^n \rightarrow \mathbb{R}^{\texttt{nonlin}}$. This function solves the nonlinear constrained maximization problem with the same generality and efficiency as the NAG routine originally employed for this purpose.

B.2 Starting values

Both E-value based starting values and likelihood-based starting values require the calculation of the likelihood of a given subsequence $\mathbf{X}_i(l, l+W-1) \equiv (X_{il}, ..., X_{il+W-1})$ under a candidate position weight matrix $PWM(i^*, l^*, W) \equiv (\mathbf{P}_1, ..., \mathbf{P}_W)(i^*, l^*, W)$ that was derived from the subsequence $\mathbf{X}_{i^*}(l^*, l^*+W-1)$ according to the mapping given above. Such a likelihood can be calculated as

$$P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W)) = \prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)=j})}(i^*, j^*, W)$$

Bailey (1995) uses a dynamic programming approach to calculate this likelihood for all subsequences $\mathbf{X}_i(l, l+W-1)$ contained in the original data and all candidate position weight matrices $PWM(i^*, l^*, W)$. Their algorithm reuses the computations for $P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W))$ when calculating $P(\mathbf{X}_i(l+1, l+W)|PWM(i^*, l^*+1, W))$ based on the recursion relation

$$P(\mathbf{X}_i(l+1, l+W)|PWM(i^*, l^*+1, W)) = \\ \frac{P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W)) \prod_{j=1}^{4} P_{Wj}^{I(X_{i(l+W)=j})}(i^*, l^*+1, W)}{\prod_{j=1}^{4} P_{1j}^{I(X_{il}=j)}(i^*, l^*, W)}$$

This calculation takes only two floating-point operations as opposed to the $W-1$ that would be required to calculate $P(\mathbf{X}_i(l+1, l+W)|PWM(i^*, l^*+1, W))$ from scratch as the product of W terms given above. The recursion is based on the observation that

$$P_{wj}(i^*, l^*+1, W) = P_{(w-1)j}(i^*, l^*, W) = \begin{cases} p_c & \text{if} \quad X_{i^*(l^*+w)} = j \\ (1-p_c)/3 & \text{if} \quad X_{i^*(l^*+w)} \neq j \end{cases}$$

for $w = 2, ..., W$, i.e. $PWM(i^*, l^*+1, W)$ is a shifted version of $PWM(i^*, l^*, W)$. The first column of $\mathbf{P}_1(i^*, l^*, W)$ is dropped, and the last column of $PWM(i^*, l^*+1, W)$ is given by

$$P_{Wj}(i^*, l^*+1, W) = \begin{cases} p_c & \text{if} \quad X_{i^*(l^*+W)} = j \\ (1-p_c)/3 & \text{if} \quad X_{i^*(l^*+W)} \neq j \end{cases}$$

Specifically, `MEME` calculates these partial likelihoods according to the following algorithm:

for $W =$ `minw` to `maxw` **do**
  for $i^* = 1$ to N **do**
    for $l^* = 1$ to $L_i - W + 1$ **do**
      for $i = 1$ to N **do**
        for $l = 1$ to $L_i - W + 1$ **do**
          if $(l^* = 1)$ calculate $P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W))$ from scratch as

$$\prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)=j})}(i^*, j^*, W)$$

          else calculate $P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W))$ as

$$\frac{P(\mathbf{X}_i(l,l+W-1)|PWM(i^*,l^*,W)) \prod_{j=1}^{4} P_{Wj}^{I(X_{i(l+W)=j})}(i^*,l^*+1,W)}{\prod_{j=1}^{4} P_{1j}^{I(X_{il}=j)}(i^*,l^*,W)}$$

        end
      end
    end
  end
end

`COMODE` does not employ this dynamic approach, causing it to be very slow since the calculation of starting values comprises a considerable part of the algorithm. `cosmo` not only employs this dynamic programming approach, but also extends it by using recursion relations that hold between partial likelihoods for neighboring values of W. These recursion relations are not available to `MEME` since it only samples candidate widths in a geometric progression. To be specific, `cosmo` uses the following algorithm:

for $i^* = 1$ to N **do**
  for $l^* = 1$ to $L_{i^*} -$ `minw` **do**
    if $(l^* = 1)$ **do**
      for $i = 1$ to N **do**
        for $l = 1$ to $L_i -$ `minw` $+ 1$ **do**
          calculate $P(\mathbf{X}_i(l, l+$ `minw` $- 1)|PWM(i^*, l^*,$ `minw`$))$ from scratch as

$$\prod_{w=1}^{\texttt{minw}} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)=j})}(i^*, j^*, \texttt{minw})$$

        end
      end
    else do
      for $i = 1$ to N **do**
        for $l = 2$ to $L_i -$ `minw` $+ 1$ **do**

calculate $P(\mathbf{X}_i(l, l+\texttt{minw}-1)|PWM(i^*, l^*, \texttt{minw}))$ as

$$\frac{P(\mathbf{X}_i(l,l+\texttt{minw}-1)|PWM(i^*,l^*-1,\texttt{minw}+1))}{\prod_{j=1}^{4} P_{1j}^{I(X_{i(l-1)}=j)}(i^*,l^*-1,\texttt{minw}+1)}$$

end

calculate $P(\mathbf{X}_i(1, \texttt{minw})|PWM(i^*, l^*, \texttt{minw}))$ from scratch as

$$\prod_{w=1}^{\texttt{minw}} \prod_{j=1}^{4} P_{wj}^{I(X_{iw}=j)}(i^*, j^*, \texttt{minw})$$

end

end

for $W = \texttt{minw} + 1$ to $\texttt{maxw}$ **do**

for $i = 1$ to N **do**

for $l = 1$ to $L_i - W + 1$ **do**

calculate $P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W))$ as

$$P(\mathbf{X}_i(l, l+W-1)|PWM(i^*, l^*, W-1)) \prod_{j=1}^{4} P_{Wj}^{I(X_{i(l+W)}=j)}(i^*, l^*, W)$$

end

end

end

end

end

The two recursion relations each only require one floating-point operation as opposed to the $W-1$ that would be necessary to calculate the corresponding partial likelihoods from scratch. This modification to the original `COMODE` implementation hence leads to an immense gain in computational efficiency.

B.3 Preventing underflow

Recall that the likelihood of sequence i under the OOPS model is given by

$$P(\mathbf{X}_i|\theta) = \frac{1}{L_i - W + 1} \sum_{l=1}^{L_i-W+1} \prod_{k \notin \tau(i,l,W)} \prod_{j=1}^{4} P_{0j}^{I(X_{ik}=j)} \frac{1}{2} \Big[\prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)} + \prod_{w=1}^{W} \prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=j)} \Big]$$

Let

$$B(i,l,W) \equiv \prod_{k \notin \tau(i,l,W)} \prod_{j=1}^{4} P_{0j}^{I(X_{ik}=j)}$$

denote the likelihood of the nucleotides in sequence i contributing to the background given that a motif of width W starts in position l of sequence i. Let

$$M(i,l,W) \equiv \prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+w-1)}=j)}$$

$$M^*(i,l,W) \equiv \prod_{w=1}^{W}\prod_{j=1}^{4} P_{wj}^{I(X_{i(l+W-w+1)}=j)}$$

denote the likelihoods of the subsequence $\mathbf{X}_i(l, l+W-1)$ under the position weight matrix in the two different orientations. Then we can write the likelihood of sequence i under the OOPS model as

$$P(\mathbf{X}_i|\theta) = \frac{1}{L_i - W + 1}\sum_{l=1}^{L_i-W+1} B(i,l,W)\frac{M(i,l,W)+M^*(i,l,W)}{2}$$

For long sequences the terms $0.5B(i,l,W)[M(i,l,W)+M^*(i,l,W)]$ can become very close to zero. To avoid underflow problems, we may choose some $N(i) \approx 0.5B(i,l,W)[M(i,l,W)+M^*(i,l,W)]$ and write the log-likelihood of sequence i as

$$\log P(\mathbf{X}_i|\theta) = \log N(i) + \log\Big[\frac{1}{L_i - W + 1}\sum_{l=1}^{L_i-W+1}\frac{B(i,l,W)}{N(i)}\frac{M(i,l,W)+M^*(i,l,W)}{2}\Big]$$

Here we choose

$$N(i) = \prod_{l=1}^{L_i}\prod_{j=1}^{4} P_{0j}^{I(X_{il}=j)}$$

as the likelihood of sequence i under the background model. `COMODE` now calculates the log-likelihood of sequence i as

seqProb = 0
for $l = 1$ to $L_i - W + 1$ **do**
    logMotProb1 = 0
    for $w = 1$ to W **do**
        logMotProb1 += log(PWM[$\mathbf{X}_{i(l+w-1)}$,w])
    end
    logMotProb2 = 0
    for $w = 1$ to W **do**
        logMotProb2 += log(PWM[$\mathbf{X}_{i(l+W-w)}$,w])
    end
    logMotProb = 0.5 exp(logMotProb1 + logMotProb2)
    seqProb += exp(log(B(i,l,W) + logMotProb - log(N(i))))
end
logLik = log(N(i)) + log(seqProb)

The logarithm inside the innermost loops is taken to prevent underflow of $M(i,l,W)$ and $M^*(i,l,W)$, forcing us then to exponentiate these logarithms outside the loop to obtain an average of probabilities on the original scale. The new implementation is based on two observations regarding this likelihood computation. First, $M(i,l,W)$ and $M^*(i,l,W)$ are unlikely to cause underflow problems since the product is only taken over W terms, with W usually no bigger than 15. Second, maximizing

$$l(\theta|\mathbf{X}_1,...,\mathbf{X}_N) = \sum_{i=1}^{N} \log N(i) + \log\Big[\frac{1}{L_i - W + 1} \sum_{l=1}^{L_i-W+1} \frac{B(i,l,W)}{N(i)} \frac{M(i,l,W) + M^*(i,l,W)}{2}\Big]$$

is equivalent to maximizing

$$llr(\theta|\mathbf{X}_1,...,\mathbf{X}_N) = \sum_{i=1}^{N} \log\Big[\frac{1}{L_i - W + 1} \sum_{l=1}^{L_i-W+1} \frac{B(i,l,W)}{N(i)} \frac{M(i,l,W) + M^*(i,l,W)}{2}\Big]$$

since the terms $N(i)$ do not involve θ. The quantity $llr(\theta|\mathbf{X}_1,...,\mathbf{X}_N)$ represents the log-likelihood ratio for comparing the null hypothesis that the entire sequence was generated under the background model to the alternative hypothesis that it was generated under the OOPS model. Unlike the likelihood of the data, this quantity is unlikely to cause underflow problems so that we can calculate it more efficiently than the actual likelihood-based on the following algorithm:

```
seqLR = 0
for l = 1 to L_i − W + 1 do
    motProb1 = 1
    for w = 1 to W do
        motProb1 *= PWM[X_{i(l+w−1)},w]
    end
    motProb2 = 1
    for w = 1 to W do
        motProb2 *= PWM[X_{i(l+W−w)},w]
    end
    motProb = 0.5*(motProb1 + motProb2)
    seqLR += B'(i,l,W) * motProb
end
```

where $B'(i,l,W) = B(i,l,W)/N(i)$. This approach saves us a few calls to the computationally expensive functions `log()` and `exp()` and thus leads to further gains in computational efficiency. Extensive tests revealed no risk of incurring underflow problems.