Bayesian Markov models improve the prediction of binding motifs beyond first order without overfitting

Wanwan Ge, Markus Meier, Christian Roth, Johannes Söding*

Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Am Fassberg 11, 37077 Göttingen, Germany

* For correspondence: soeding@mpibpc.mpg.de

Abstract

Transcription factors regulate gene expression by binding to specific DNA motifs. Accurate models for predicting binding affinities are crucial for quantitatively understanding transcriptional regulation. Motifs are commonly described by position weight matrices, which assume that each position contributes independently to the binding energy. Models that can learn dependencies between positions, for instance, induced by DNA structure preferences, have yielded markedly improved predictions for most transcription factors on *in vivo* data. However, they are more prone to overfit the data and to learn patterns merely correlated with rather than directly involved in transcription factor binding. We present an improved, faster version of our Bayesian Markov model software, BaMMmotif2. We tested it with state-of-the-art motif discovery tools on a large collection of ChIP-seq and HT-SELEX datasets. BaMMmotif2 models of fifth-order achieved a median false-discovery-rate-averaged recall 13.6% and 12.2% higher than the next best tool on 427 ChIP-seq datasets and 164 HT-SELEX datasets, respectively, while being 8 to 1000 times faster. BaMMmotif2 models showed no signs of overtraining in cross-cell line and cross-platform tests, with similar improvements on the next-best tool. These results demonstrate that dependencies beyond first order clearly improve binding models for most transcription factors.

INTRODUCTION

Gene expression is regulated through the binding of transcription factors (TFs) to specific recognition motifs within promoter and enhancer DNA sequences. These binding motifs typically contain 6 to 12 only partially conserved bases [1–3]. Learning quantitative models from experimental data that allow us to accurately predict the binding affinities of TFs to any given sequence is important for quantitatively predicting transcription rates from regulatory sequences and TF concentrations.

The task of *de-novo* motif discovery is to infer from experimental data a statistical or, equivalently, thermodynamic model that can then predict the binding affinity of a TF of interest for any sequence. Motif models can be inferred from numerous types of experiments [4]. Common *in vivo* techniques are ChIP-seq [5] and bacterial-one-hybrid [6], while most modern *in vitro* approaches are SELEX-based [7–9]. These measurements result in sets of hundreds to millions of bound sequences from which the binding motif model is deduced based on the statistical enrichment of binding sites compared to a background set of unbound sequences or a background model for random sequences.

The dominant model for describing the binding affinity of transcription factors to DNA target sequences has been the position weight matrix (PWM). This model assumes that the binding energy can be decomposed into a sum of contributions from each of the nucleotides in the binding site. By Boltzmann's law, this is equivalent to assuming statistical independence between nucleotides at different positions of the binding site. The PWM model has been enormously successful because with its only $3W$ parameters for a binding site of W nucleotides, it achieves high accuracy for predicting the binding affinity of high-affinity binding sites. However, modeling the nucleotide inter-dependency often yields better motif predictions than PWMs [10–12]. One reason is that the stacked, neighboring bases largely determine the physical properties of DNA, such as their equilibrium bending angle, minor groove width, propeller twist, or helical twist. The information on the geometric orientation of the bases propagates within the DNA for several positions before fading out, creating a dependence of the DNA physical properties on nucleotide pairs, triplets and longer k-mers. Since TFs recognize their target sites not only using hydrogen bonds but also using their structural fit, TF binding motifs show preferences depending on k-mer words [13], particularly in the flanking regions outside the hydrogen bonding core region [14]. Furthermore, alternative binding modes of TFs [15, 16] can lead to poor performance of PWMs.

During the past decade, it has become increasingly evident that weak binding sites in enhancers and promoters play an important role in determining transcriptional activity [17–21], and PWMs have shown limitations to describe the affinities for weak binding sites accurately. Therefore, various more refined models have been developed that depart from the simplifying assumption of independence of motif positions [22–24]. Prime among them are inhomogeneous Markov models of order k, in which the probability to observe a certain nucleotide at position i depends on the previous k nucleotides at $i-k$ to $i-1$. A zeroth order Markov model is therefore equivalent to a PWM. Dinucleotide weight matrices (DWMs), which are equivalent to first-order models, in which the probability of a nucleotide depends on its direct predecessor, have shown improved accuracy over PWMs [25–27].

For Markov models of higher order k, the large number of $W \times (4^{k+1}–1)$ parameters can lead to overfitting on the training data and hence bad predictive performance. To address this

limitation, our group had proposed a special type of Markov models, the Bayesian Markov models (BaMMs), in which the probability for a nucleotide at position i of the motif, for example the last nucleotide in `ACTCG`, is estimated by adding to the actual counts of `ACTCG` pseudo counts based on how often the shorter $(k-1)$-mer `CTCG` has been observed in the binding sites. The probability for `CTCG` in turn is estimated by adding its counts to pseudo counts based on how often the word of length $k-1$, `TCG`, has been observed, and so forth. This procedure can be derived formally in a Bayesian framework with Dirichlet priors. Our software BaMMmotif indeed improved on previous PWM-based methods for *de novo* motif discovery and binding site prediction on *in vivo* data [28].

Here we present BaMMmotif2, an open-source software written entirely from scratch in C++. It contains a novel algorithm for its seed finding stage, which gives it greatly improved speed and slightly improved sensitivity in comparison to BaMMmotif. We improved the robustness of the BaMM-based motif refinement stage using sequence masking. BaMMmotif2 can also learn positional preference profiles for binding site locations from the training data.

Higher-order models have the ability to learn several low-order motifs overlaid on top of each other [29]. It was therefore surmised that at least a part of the improvements of higher-order models on cross-validation benchmarks using ChIP-seq sequences could stem from learning not only the main binding motif of the ChIPped factor but also, overlaid, the binding motifs of cooperating factors whose binding sites tended to co-occur with it [18]. This would of course defeat the purpose of learning the binding affinity of the ChIPped factor. In a different cell type, for instance, in which different co-binding factors are expressed, such a mixed motif might perform badly. It was also surmised that more complex models could learn complex, nonspecific sequence biases characteristic of the measurement technique that would allow them to be distinguished from the background sequences. These platform-dependent biases could result from the library preparation, amplification, and ligation biases [30].

We therefore designed a set of benchmark experiments with a focus on detecting such overfitting (Fig. 1): (I) 5-fold cross-validation on ChIP-seq and HT-SELEX data; (II) cross-cell-line validation on ChIP-seq data for the same TFs; (III) model training on ChIP-seq data and testing on HT-SELEX data for the same TFs and (IV) vice versa. Scheme (I) examines how the models generalize to unseen data, especially when data is limited.

Our results demonstrate that BaMMmotif2 does not show signs of overfitting but rather learns the binding affinity of only the factor of interest, and that BaMMmotif2 is the most sensitive and fastest tool among the ones tested here. Furthermore, BaMMmotif2 keeps improving the performance with increasing model orders and scales better with larger datasets. Finally, we demonstrate the power of BaMMmotif2 models by predicting 1.8 million CTCF binding sites genome-wide with a false discovery rate below 0.5%.

MATERIALS AND METHODS

The BaMMmotif2 algorithm

BaMMmotif2 consists of a seeding stage and a motif refinement stage. The purpose of the seeding stage is to exhaustively identify motifs enriched in the input sequences in comparison to a second-order Markov background model trained also on the input sequences. Each of

the motifs below a P-value cut-off is refined by the BaMM-based refinement stage.

The fast seeding stage. This method is described in detail in the Supplementary Section 1.1. Briefly, we first count the number of occurrences of each non-degenerate W-mer word in $\{\mathrm{A,C,G,T}\}^W$ ($W = 8$ in this study) in the input sequences. From here on, we only inspect the count array and not the sequences anymore, making the runtime of the seeding stage almost independent of the input set size. By default, reverse complements are mapped to the alphabetically lower of the two W-mers.

For each W-mer, an enrichment z-value is calculated, which is the number of standard deviations with which the observed W-mer count surpasses its expected count. The expected count is calculated using a second-order homogeneous Markov model as a background sequence model, trained on the input sequences. We determine all locally optimal w-mers. These are the W-mers with a better enrichment z-value than any of its direct neighbors one substitution away. We use each of the locally optimal w-mers to initialize a search for locally optimal W-mer patterns in the 10-letter IUPAC alphabet $\{\mathrm{A,C,G,T,S,W,R,Y,M,K,N}\}$, where the last six letters stand for C or G, A or T, A or G, C or T, A or C, and G or T, respectively. For each such locally optimal IUPAC pattern, a PWM is derived from all matches in the input sequences to the degenerate pattern. The PWMs are then refined by applying the expectation-maximization (EM) algorithm in the multiple-occurrences-per-sequence (MOPS) model. We merge PWMs together that overlap by at least $W - 2$ highly similar matrix columns. Finally, the PWMs are reranked by their AvRec scores (explained in the next section) and written into an output file in MEME format [31], which is passed to the refinement stage.

The refinement stage. The refinement stage is initialized with the motif occurrences found by the PWMs passed to it from the seeding stage. The length of the motif is extended by 2 bp on both ends by default to ensure that we do not miss information in the flanking regions. Each seed model is refined into an inhomogeneous Bayesian Markov model (BaMM) of order K using the EM algorithm (see Supplementary Section 1.5). A BaMM is an interpolated Markov model in which the conditional probability of base $x_i \in \{\mathrm{A,C,G,T}\}$ at position i is calculated by combining the counts $n_i(x_{i-k:\,i})$ of k-mer $x_{i-k:\,i}$ with pseudo counts estimated from lower-order probabilities $p_i^{\mathrm{BaMM}}(x_i|x_{i-k+1:\,i-1})$:

$$p_i^{\mathrm{BaMM}}(x_i|x_{i-k:\,i-1}) = \frac{n_i(x_{i-k:\,i}) + \alpha_k \times p_i^{\mathrm{BaMM}}(x_i|x_{i-k+1:\,i-1})}{n_{i-1}(x_{i-k:\,i-1}) + \alpha_k}. \tag{1}$$

Here, the hyper-parameter α_k determines how much weight to give to the lower-order. The probabilities for order $k-1$ are again obtained by adding to the k-mer count the pseudo counts from order $k-2$, and so on down to order 0. In this way, when the number of occurrences observed for $(k+1)$-mer $x_{i-k:i}$ is much smaller than the number of pseudo counts $\alpha_k \times p_i^{\mathrm{BaMM}}(x_i|x_{i-k+1:\,i-1})$, the higher order falls back to the lower order: $p_i^{\mathrm{BaMM}}(x_i|x_{i-k:\,i-1}) \approx p_i^{\mathrm{BaMM}}(x_i|x_{i-k+1:\,i-1})$. In this way, BaMMs adapt the order that is learned in a data- and motif position-specific fashion to the amount of data (k-mer counts) available, up to the highest order K learned. We assume that the correlation between nearby bases declines with their distance. This is reflected in the pseudo-parameters α_k increasing with order k. For BaMMmotif2, we kept the same setting as in BaMMmotif, $\alpha_k = 7 \times 3^k$.

The motif model is optimized with the EM algorithm by maximizing the likelihood of the input sequences assuming zero or one motif occurrence per sequence (the ZOOPS model). It models the bound sequence using a Kth-order inhomogeneous BaMM $p^K_{\text{motif}}(\mathbf{x})$ (where $\mathbf{x} = x_{1:W}$ is the binding site), and models the other unbound sequence regions using a K'th-order homogeneous BaMM $p^{K'}_{\text{bg}}(\mathbf{x})$ (K' is 2 by default). This background sequence model is trained by default on the input sequences. Potential binding site sequences $\mathbf{x}$ are ranked by their score $S(\mathbf{x}) = \log(p^K_{\text{motif}}(\mathbf{x})/p^{K'}_{\text{bg}}(\mathbf{x}))$. In the weak binding limit, this score is proportional to the Gibbs free energy ΔG of binding (Supplementary Section 1.5).

Learning positional binding preferences. BaMMmotif2 can learn the positional binding preferences for enriched motifs with respect to the center of the input sequences. By aligning the sequences around some anchor feature, such as a transcriptional start site, a 3' splice site, or a binding site of some other transcription factor, the distance preference between enriched motifs and the reference feature can be learned. We parameterize the positional probability distribution with one parameter per position and ensure smoothness by adding L_2 penalties for the differences between successive sequence positions (Supplementary Section 1.5).

Masking sequences during the motif refinement stage. Sequences from *in vivo* experiments such as ChIP-seq commonly contain several distinct motifs from other TFs that together co-regulate their target genes. This can create two types of problems during the refinement stage. First, instead of refining the motif from the seeding stage, the model in some cases tends to learn two or even more motifs in the same higher-order model, as this often improves the likelihood on the training data. Second, if the seed motif is less enriched or less informative than other motifs in the positive sequence set, the model can switch from the seed motif to these other motifs. In this way, the weaker motif is not discovered at all. To avoid these two problems, we introduced a masking step in the EM optimization. We score all possible motif start positions in the input sequences using the PWM passed from the seeding stage to the refinement stage. We mask out all but the top $X\%$ of positions ($X = 5$ in this study) and ignore these positions in the EM iterations of the refinement stage.

Motif assessment using average recall (AvRec)

To assess the performance of a classifier such as a motif model, one often plots the true positive predictions (TP) versus the false positive predictions (FP) over all score thresholds. Normalizing FPs and TPs to a maximum of 1 by plotting the true positive rate TPR = TP/Positives versus the false positive rate FPR = FP/Negatives yields the receiver operating curve (ROC). The often-used area under the ROC curve (AUC) is not a good quality measure for a motif model because in many applications the fraction of positive sequences (those carrying the motif) is much smaller than the number of negative sequences. When scanning the human genome for CTCF binding motifs in windows of 100 bp, for example, the ratio is about 1:30 (Fig. 6). At this ratio, a false discovery rate FDR = TP/(TP + FP) below 50% requires a ratio FPR/TPR $< 1/30$. So $29/30 = 97\%$ of the ROC plot, the part with FDR $> 50\%$, would be irrelevant. A predictor could have an AUC of 95% and never reach an FDR below 50%.

We therefore previously developed the Average Recall (AvRec) score [32], which averages

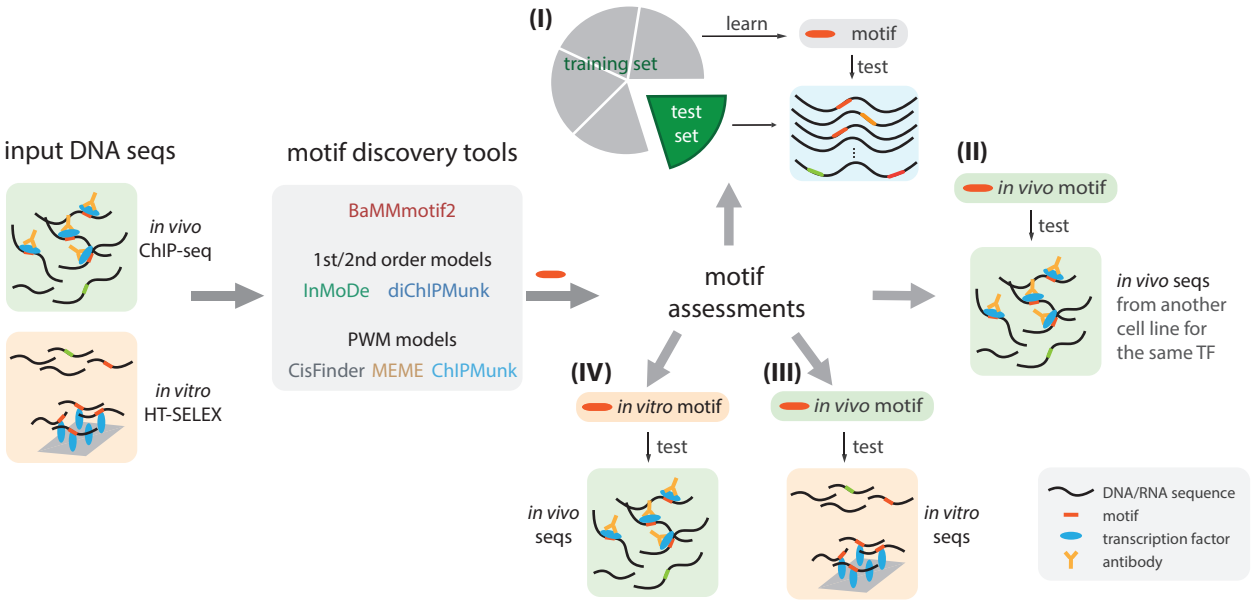

Figure 1. Benchmark pipeline for *de novo* motif discovery. Five state-of-the-art motif discovery tools and BaMMmotif2 learned motif models on *in vivo* and *in vitro* transcription factor binding datasets. The learned models were then assessed (I) by 5-fold cross-validation on the same type of data, (II) by cross-cell-line validation, and (III, IV) by cross-platform validations.

the recall (the same as true positive rate and sensitivity) over a range of TP:FP ratios from 1:1, corresponding to FDR = 0.5, to 1:100, corresponding to FDR = 1/101 (Fig. 2 A). The AvRec score therefore considers the range of FDR most relevant in practice and has the additional benefit that a different positive-to-negative ratio than 1 simply results in a vertical shift of the AvRec curve on the logarithmic y axis.

To calculate the AvRec score, we first simulated 10-fold more negative than positive sequences using a second-order Markov background model learned on the positive set. We computed the motif scores $S(x_{i:i+W-1}) = \log\left(p^{K}_{\text{motif}}(x_{i:i+W-1})/p^{K'}_{\text{bg}}(x_{i:i+W-1})\right)$ for all possible binding positions i (excluding the masked positions) and took the best score for each sequence. All sequences are sorted by descending score. The false positive count FP is the cumulative number of sequences from the negative set above the score cut-off, TP is the cumulative number of positive sequences above the score cut-off.

Benchmark design

The performance of BaMMmotif2 was evaluated together with five state-of-the-art motif discovery tools, MEME [31] as the most cited tool, CisFinder [33] for its speed and ability to run on large datasets, ChIPMunk [34] and diChIPMunk [27], which are used for generating the PWMs and dinucleotide PWMs in the HOCOMOCO database [35], and InMoDe [24], which can learn inhomogeneous Markov models or order 2 and beyond.

The processing of the ChIP-seq and HT-SELEX data is described in detail in section 2 of the Supplemental Material. The motif discovery tools were run on the input sequence sets with default parameters, and four CPU cores were used for tools that could be parallelized (CisFinder, MEME, and BaMMmotif2). For tools that learn multiple motif models from one dataset, the motif models ranked top by the tools were benchmarked.

To assess the model performance over the given sequences, we first performed the benchmark on both human ChIP-seq [36] and HT-SELEX datasets [37] using 5-fold cross-validation (Fig. 1 I). The cross-cell-line validation was applied to ChIP-seq data (Fig. 1 II) and the cross-platform validations were applied to both ChIP-seq data HT-SELEX data (Fig. 1 III and IV). A more detailed description including tool settings can be found in Supplementary Section II.

RESULTS

Model performance on *in vivo* and *in vitro* data

We learned *de novo* motifs with each of the six tools on 427 ChIP-seq datasets for 93 transcription factors from the ENCODE project [36] and evaluated their performance using 5-fold cross-validation (Fig. 1I).

As an example, we compare in Fig. 2A and B the AvRec plot of a fifth-order BaMM with a second-order InMoDe model for the Elf2 motif, trained and tested on 5000 sequences of length 208 bp via 5-fold cross-validation. At a positives-to-negatives ratio of 1:1 (bold blue line) and a TP:FP-ratio of 10:1 (see y axis, corresponding to an FDR of 1/11), the BaMMmotif2 model achieves a recall of 0.81 and the InMoDe model achieves 0.69. At a positives-to-negatives ratio of 1:10 and a TP:FP-ratio of 10:1 (broken blue line), or, equivalently, at a positives-to-negatives ratio of 1:1 and a TP:FP-ratio of 100:1, the models achieve recalls of 0.12 and 0.13, respectively. When comparing AvRec scores between fifth-order BaMMs with second-order models from InMoDe across all 427 ChIP-seq datasets, BaMMs attain higher AvRec scores for 415 (97%) of the datasets, and the median AvRec of BaMMs is 13.6% higher than the one of InMoDe models (Fig. 2C).

Overall, the PWM-based tools, CisFinder, MEME, and ChIPMunk, are outperformed by the tools using higher-order models. BaMMmotif2 with first-order models performs on par with InMoDe and better than the first-order tools diChIPMunk. Fifth-order BaMMs achieve even better AvRec scores, as seen in the box plots and AvRec cumulative distributions of Fig. 2D,E, and in one-on-one comparisons in Fig. S2A. We also compared BaMMmotif2 with our previous tool BaMMmotif [28]. BaMMmotif2 is 10 times faster while being slightly more sensitive (Fig. S 3).

Tools that learn higher-order Markov models can learn several motifs in one model, profiting from signals that are merely correlated with the real binding sites [29,38]. To find out whether BaMMs are affected or not, we introduced a masking step in the initial iteration of the EM algorithm (see Materials and Methods). We restrict the model refinement with the higher order BaMM to the 5 % potential motif positions with the highest scores scanned by the seeding PWM. In this way, we avoid overfitting and also speed up the refinement by a factor 10. However, this robustness is paid by a loss in motif model performance (Fig. S4 and S5). The performance decrease could be caused in part by the limitation of not being able to select better sites during the refinement that were too different from the seeding motif, and in part because sometimes the BaMMs would otherwise have learned more than one distinct motif in a single model. To be on the conservative and robust side, we adopted the masking step in BaMMmotif2 for all our benchmarks throughout this study, unless explicitly stated

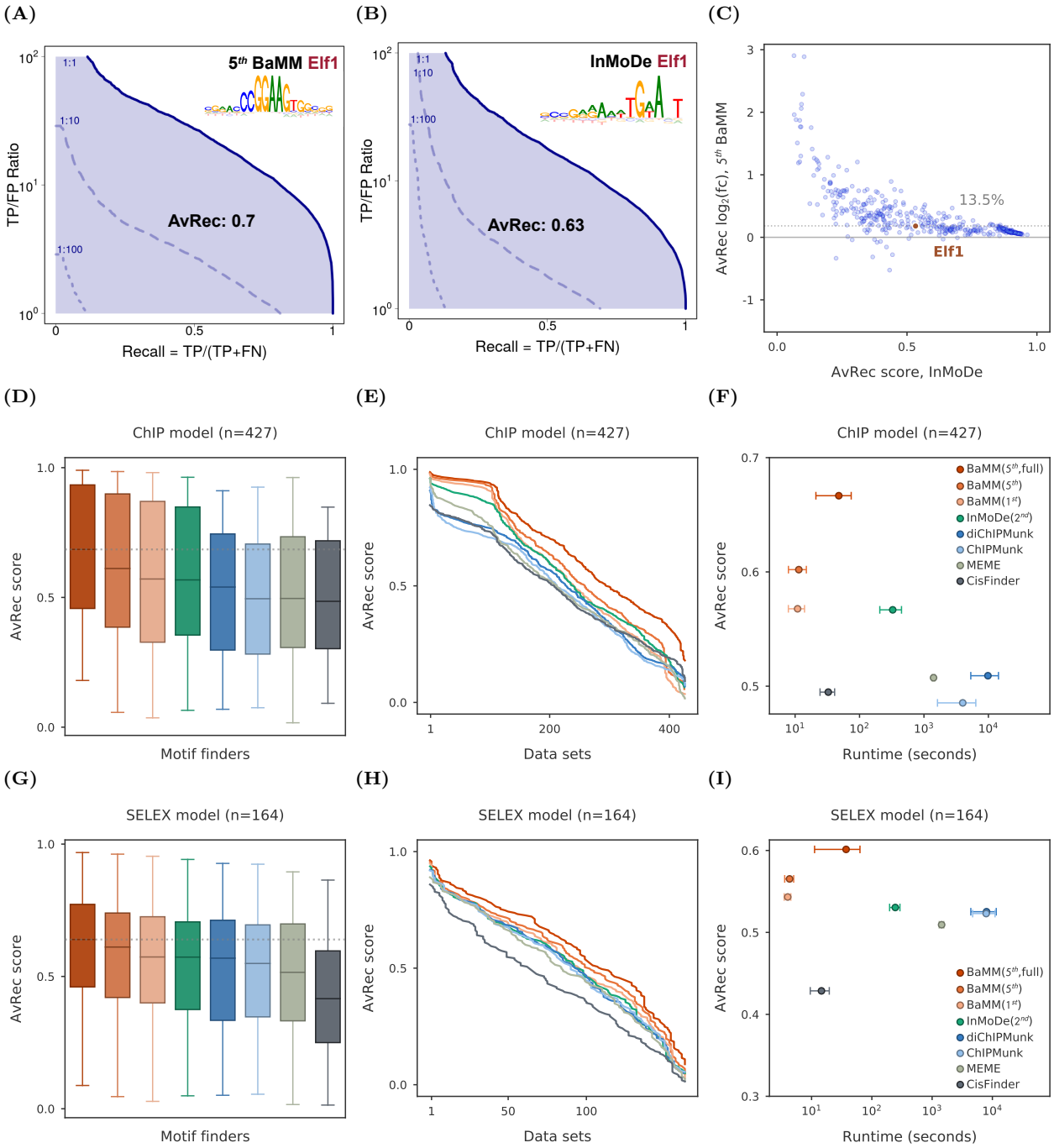

Figure 2. Performance of *de novo* motif discovery tools on *in vivo* and *in vitro* datasets. (**A**) AvRec analysis for fifth-order BaMM on the Elf1 ENCODE dataset. The AvRec is the recall (= sensitivity) averaged in log space over TP-to-FP ratios between 10^0 and 10^2. This ratio range corresponds to a precision between $1/(1+1)$ and $100/(1+100) = 0.99$. Bold line: 1:1 ratio of positives to negatives. At 1:10 ratio (dashed) and 1:100 (dotted), the curves are shifted down by a factor 10 and 100, respectively. Inset: motif logo of Elf1. (**B**) Same as (**A**) for the InMoDe model of Elf1. (**C**) $\log_2$ of AvRec fold change between fifth-order BaMMmotif2 and InMoDe models versus the AvRec of InMoDe. Each dot represents one dataset. Elf1 is highlighted in red. The median $\log_2$ fold change is 13.6%. (**D**) AvRec distributions as box plot, with boxes indicating 25%/75% quantiles and whiskers 95%/5% quantiles. Color code: see the legend in (**F**). (**E**) Cumulative distribution of AvRec scores on the 427 datasets. (**F**) Average runtime per dataset on four cores versus the median AvRec score. InMoDe, and (di)ChIPMunk are not parallelized and ran on a single core. Whiskers: ± 1 standard deviation. BaMM (5th, full): no masking step. (**G-I**) Analogous to (**D-F**) but for 164 HT-SELEX datasets from the Taipale lab [37].

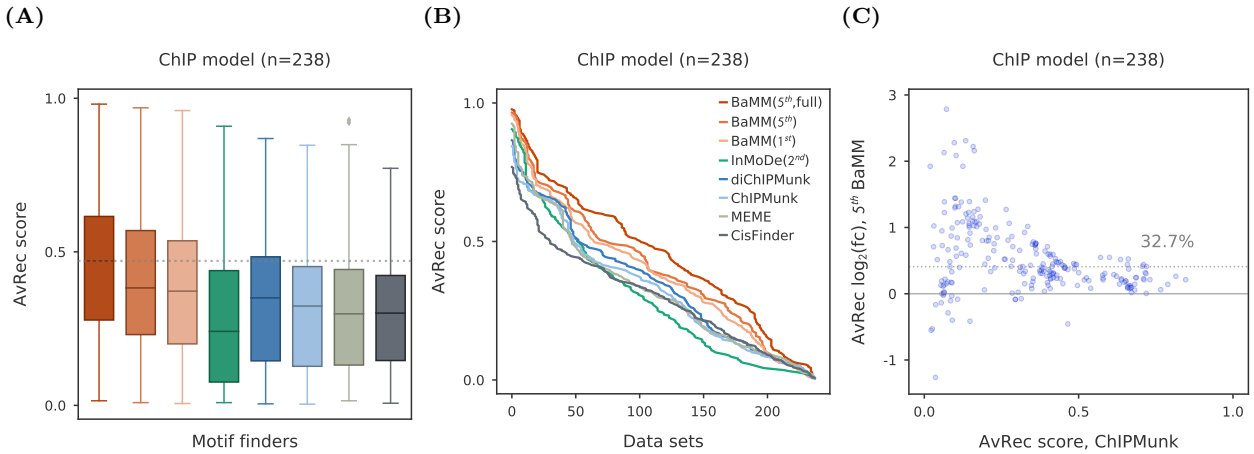

Figure 3. Cross-cell-line validation. 119 pairs of ENCODE datasets in which the same TF had been ChIPped in different cell lines were used in this benchmark. (**A**) AvRec distributions for 2×119 models that were tested on a ChIP-seq dataset from a different cell line than they were trained on. (**B**) Cumulative distributions of AvRec scores. (**C**) Log_2 fold change in AvRec between fifth-order BaMMmotif2 models and ChIPMunk for each of the 238 datasets. Median improvement is 32.7%.

otherwise.

Next, we assessed the performances of selected tools on 164 *in vitro* HT-SELEX datasets for 164 TFs [37]. Each dataset contains long oligomers of 200 bp. We also sampled 10-fold background sequences using the trimer frequencies from the same input set for estimating true negatives.

We conducted the *in vitro* benchmark and observed overall similar trends as on the ChIP-seq data (Fig. 2G-I). CisFinder tends to learn longer motifs than the other tools, which probably helped it on the ChIP-seq data but hurt its performance on the HT-SELEX data. The BaMMs learned without masking (BaMM 5th, full; red) gained only 5% on the masked version (BaMM 5th; orange), whereas the gain had been 12% on the ChIP-seq data. This comparison shows that, on the ChIP-seq data, the fifth-order BaMMs trained without masking indeed tend to learn also motifs of co-occuring TFs that help to distinguish positive from negative sequences. If the goal is to learn the pure binding affinity of the ChIPped TF, masking should therefore be turned on for *in vivo* data.

Assessing consistency of motif models across cell lines

ChIP-seq measurements have cell-type-specific biases associated with difference in chromatin accessibility, in particular of enhancers and promoters, and differences in TF concentrations [39]. A motif model that predicts only the binding affinity of the ChIPped TF should also perform well in predicting binding sites of the factor in other cell lines, whereas a motif model that has learned also motifs of co-occurring TFs and other sequence features with no direct effect on the binding affinity of the main TF should generalize badly to other cell lines in which different TFs will often co-occur with the ChIPped TF.

We therefore conducted a cross-cell-line benchmark on *in vivo* data. We assessed the performance of models learned on ChIP-seq data from one cell line and tested on ChIP-seq

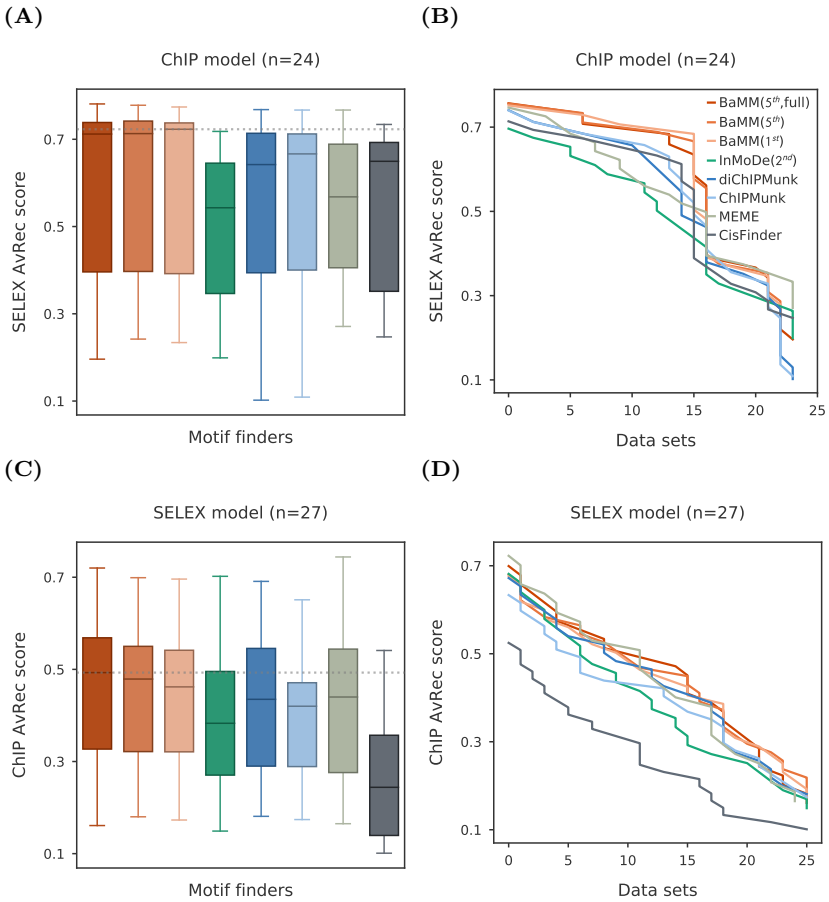

Figure 4. Cross-platform validation. (**A,B**) AvRec distributions and cumulative distributions for 24 models trained on ChIP-seq datasets and tested 7 on HT-SELEX datasets. (**C,D**) Same as A,B, for 8 motif models trained on HT-SELEX datasets and tested on 27 ChIP-seq datasets for the same TFs.

data of the same TF from another cell line. We found 119 pairs of ChIP-seq datasets in the ENCODE database in which the same TF had been ChIPped in two different cell lines. We trained the model on one dataset and tested it on the other, and vice versa, resulting in 238 AvRec scores (Fig. 3).

Remarkably, the AvRec scores are around 0.2 lower for all tools than the AvRec scores in Fig. 2D obtained when training and testing in the same cell lines, with the PWM-based tools going from AvRec 0.5 to as low as 0.3. This quite dramatic decrease indicates that all models, even the simple PWMs, do not perform well for predicting bound sequences in another cell line. Remarkably, except for InMoDe, the predictive power of the higher-order models does not suffer more than that of the PWMs. This indicates that the higher-order models (except InMoDe) do not tend to overfit to sequence features that are specific to one cell line, such as co-occurring TFs. It is surprising that the fifth-order BaMMs trained without masking maintain or even improve their edge on the other models, despite our expectation that they would be the most prone to overfit on cell type-specific features.

In vitro models predict *in vivo* binding and vice versa

Each measurement for detecting TF-DNA interactions has its own biases. ChIP-seq has biases from sequence-dependent PCR amplification, cell-type-specific sonication bias, and chromatin structure [40–42], while HT-SELEX has biased nucleotide compositions and depleted palindromes as a result of the library preparation, as well as sequence bias carry-over in selection cycles [43, 44]. To assess how much models base their predictions on technical biases that would improve their performance when tested on the same platform but decrease their performance when tested on a different platform, we performed two cross-platform benchmarks.

First, for each of the tools we trained a motif model on each of the 66 ChIP-seq datasets for which a HT-SELEX dataset with the same TF, but not necessarily from the same cell line, was available. We discovered that several datasets were of too low quality to give reliable models, and some HT-SELEX datasets showed signs of having had the identity of the TF switched. We therefore selected the 27 ChIP-seq datasets for which all 8 tools achieved at least an AvRec score of 0.1. The first- and fifth-order BaMMs achieve better accuracies than the PWM-based models (Fig. 4A and 4B). However, the fifth-order models do not improve over the first-order model.

Second, for each of the tools we trained a motif model on each of the 15 HT-SELEX datasets for which a ChIP-seq dataset with the same TF was available. We selected the HT-SELEX datasets for which all 8 tools achieved at least an AvRec score of 0.1. Again, BaMMs achieved the best AvRec scores with no major improvements from first to fifth order (Fig. 4C and 4D). This time, the improvements over PWM-based models are minor. CisFinder fared badly because it learned longer, more specific motifs than the other tools that were too specific to match well in the flanking regions of motifs in the positive sequences from the other platform (See Fig. S 7 for examples).

BaMMs learned similar information in first order on ChIP-seq and on HT-SELEX data while showing no tendency to learn systematic biases of these platforms (Fig. S 7). This demonstrates how the information in the first-order can help to improve cross-platform predictions. As a caveat to these cross-platform comparisons, the low number of comparisons results in high statistical noise in the AvRec scores.

Extended flanking regions increase motif prediction accuracy

Various studies have shown that the flanking regions outside of the core binding sites affect TF binding, by affecting DNA shape preferences or by harboring binding sites of co-cooperatively binding TFs at variable spacings [14, 45–47]. Therefore, we investigated the impact of extending the core motifs, by adding two or four nucleotides on each side in the seeding motifs and refining the extended motifs with BaMMmotif2.

We find that for BaMMs trained on ChIP-seq datasets, extending the models by 2×2 or 2×4 positions indeed improves the motif model performance across all orders, and more so with increasing order (Fig. 5A). The improvement from no added positions to 2×4 bp added is by 3% for zeroth order BaMMs (PWMs) and by 11% for fifth-order BaMMs (Fig. 5B and Fig. S 8A). This indicates that flanking regions carry information mostly in the higher orders and not much in preferences for specific nucleotides.

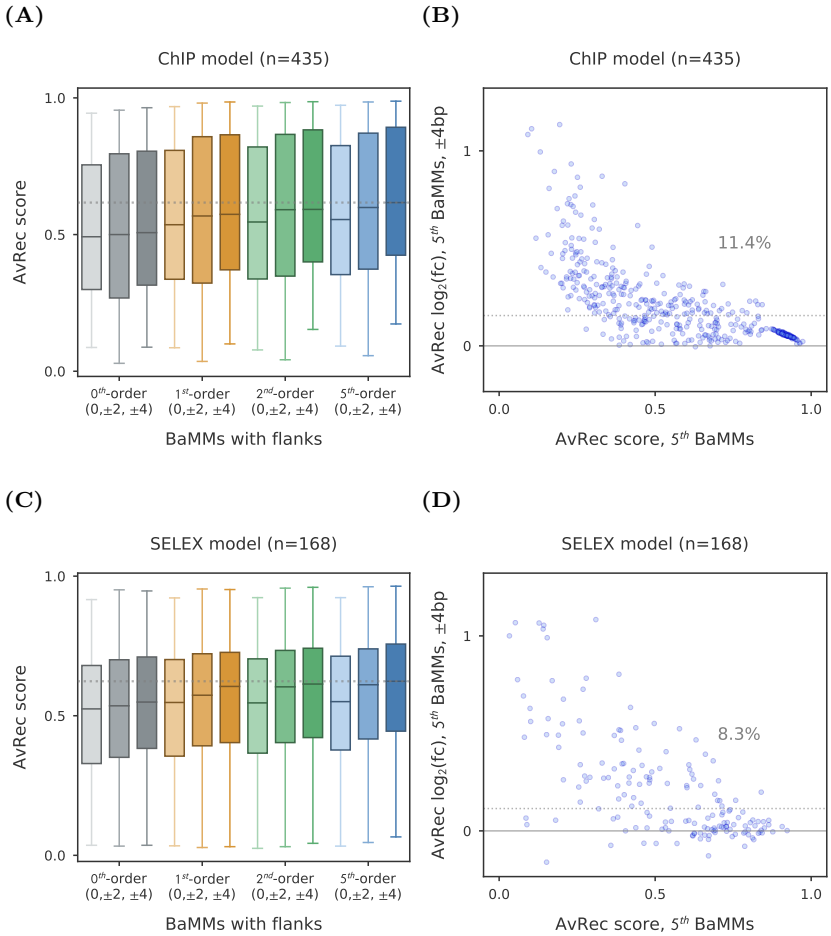

Figure 5. Extending the core motif by flanking positions improves motif performance. AvRec of BaMMs with different numbers of flanking positions added to the core motif, tested by 5-fold cross-validation. (**A**) AvRec distribution on 435 ChIP-seq datasets for models of order 0 , 1, 2, and 5, each for three sizes of flanking regions: 0 bp, ± 2 bp , and ± 4 bp. (**B**) Log_2 of fold change between fifth-order BaMMmotif2 models with ± 4 bp flanking positions and no added flanking positions. Median AvRec increase is 11.4 %. (**C,D**) Same as (A,B) for 168 HT-SELEX datasets.

It is not clear, however, if these improvements are due to DNA shape preferences that are reflected by preferences for certain di- and tri-nucleotides or by other sequence features of the genomic sequences such as motifs of co-occurring TFs. We therefore repeated the same analysis on HT-SELEX data. We restricted ourselves to long oligonucleotides of 200 bp because short oligonucleotides of 20 bp to 40 bp might not reflect well enough the physical properties of genomic DNA.

The results on the HT-SELEX data are very similar to those on ChIP-seq data (Fig. 5B, D). Again, PWMs gain much less AvRec score through 2 × 4bp extensions than fifth-order BaMMs (1.3% versus 8%, shown in Fig. S 8B and Fig. 5D). This result confirms that the features picked up by the higher orders are not chiefly ones that are specific to genomic sequences but are also learned on *in vitro*-selected sequences and are therefore likely to be associated with DNA structural preferences.

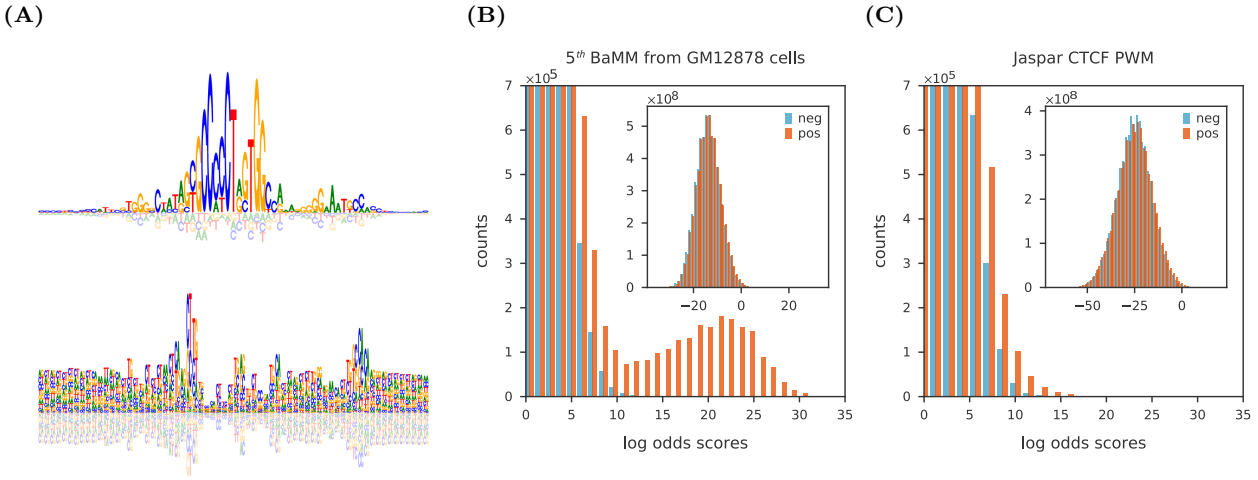

Figure 6. A fifth-order BaMM predicts 1.8 million high-confidence CTCF sites in the human genome. (**A**) Information content of CTCF motif model in zeroth- (top) and first-order (bottom). Distributions of motif log-odds scores on the hg19 genome (orange) and the reverse genome (blue), scanned with a fifth-order BaMM learned from the GM12878 cells (**B**) and a PWM (JASPAR ID: MA0139.1) (**C**). Inset: the full distributions of the predicted CTCF sites on the whole genome (orange) and the reverse genome (blue).

Learning positional binding preferences

Motifs often have certain positional preferences with regard to other motifs or genomic landmarks such as transcription start sites. Therefore, we introduced the possibility to learn the probability distribution of motif positions from the input data (Fig. S 1A). Learning the positional distribution of motifs around ChIP-seq peak positions did not improve the median motif performances (Fig. S 1B and 1C), probably because the information content of the positional distribution is very low when the the distribution is not much narrower than the window size. (The information content can be calculated as the difference between the entropies of the two positional distributions.) The positional preference is likely to have a positive impact when positioning effects are stronger, such as for splicing motifs around splice sites, core promoter motifs around transcription start sites, or TF binding sites or cooperatively binding TFs.

Fifth-order BaMM predicts 1.8 million high-confidence human CTCF sites

The CCCTC binding factor CTCF plays a key role in transcriptional regulation through the formation of chromatin loops and topologically associated domains (TADs). Chromatin loops and TADs are believed to be formed by DNA getting extruded through a cohesin ring [48]. The extrusion an loop growth is halted when the cohesin encounters CTCF bound to each of the two strands of the DNA in the preferred orientation.

Zhang et al. [49] identified 112 thousand CTCF binding sites from the GM12878 cell line, using a PWM of CTCF motif from the JASPAR database (ID: MA0139.1) [50]. We took the ENCODE dataset for CTCF from the same cell line, trained a fifth-order motif with the length of 67 bp in total, a core of 27 bp with 2×20 bp flanks (Fig. 6A) and scanned

the human genome (hg19) with it using the script BaMMScan in the BaMMmotif2 software. Fig. 6B shows a histogram of the BaMMmotif2 score distribution obtained on the human genome (orange). Remarkably, the motif score distribution on the human genome displays a two-peaked distributions with a minimum around score 11 separating a small peak from the bulk of the distribution. This shape is suggestive of a mixture of two somewhat overlapping but separated components, a huge component comprising the non-CTCF sites and a small component with true CTCF sites.

If this was true, we would expect few of the sites with scores above 11 to be false positive predictions. To verify this, we estimated the number of false positive predictions per score bin by scanning the reversed human genome sequence with the CTCF BaMM. The score distribution is shown with blue bars in Fig. 6B. The distribution of negatives show only 7542 positions with scores above 11, while the distribution of scores on the human genome has 1.8 million positions. This confirms our hypothesis that these 1.8 million positions in the human genome are high-confidence CTCF sites (FDR < 0.005).

To compare these predictions with a state-of-the-art prediction using the CTCF PWM in the JASPAR database (ID: MA0139.1) [50], we have plotted in Fig. 6C the scores distributions of the PWM (obtained with BaMMscan with second-order background Markov model) on the human genome (orange) and the reversed human genome. While the negatives look very similar to the one for the BaMM, the score distribution on the human genome has no separate peak at high scores, and only 67 thousand positions have a score larger than 11 (FDR < 0.15).

These results indicate that higher-order BaMMs can predict many more binding sites for CTCF and other factors with high confidence than PWM-based methods.

DISCUSSION

We presented BaMMmotif2, a fast and accurate *de novo* motif discovery algorithm for large-scale transcriptomic data. BaMMmotif2 builds on our earlier theory of Bayesian Markov models (BaMMs) implemented in BaMMmotif. BaMMs employ pseudocounts from model order $k-1$ to stabilize the estimation of the conditional probabilities for order k, for all orders k from 1 to the maximum order (five in this study). In this way, they can learn higher orders if a sufficient number of k-mer counts was observed to estimate them but otherwise fall back to a lower order that can still be estimated safely.

BaMMmotif2 was written from scratch in C++ using explicit AVX2 vectorization and multi-core parallelization. We developed a novel, fast seeding method to find enriched patterns that scales almost independently of the input set size. We also added a masking step to force the refinement stage to only refine the seed motifs and prevent it from learning in addition other predictive features such as co-occurring motifs of other TFs or experimental sequence biases. We also developed a Bayesian approach to learn positions binding preferences from the input data.

By their sheer number, ChIP-seq datasets are the dominant source of information for TF binding affinities. Therefore, most benchmark comparisons of *de novo* motif discovery tools have been performed exclusively or predominantly on ChIP-seq data. However, for assessing the quality of models more complex and informative than PWMs, such as higher-order

Markov models and mixture models, ChIP-seq data are problematic for several reasons. First, they often have complex sequence biases [39], which higher-order models can learn to distinguish from negative sequences generated with random background models. To alleviate this problem, second order background models should be used, but even this might be insufficient to eliminate learning generic sequence biases of the ChIPped versus random sequences. Second, sequences in ChIP-seq peaks usually contain in addition to the motif of the ChIPped TF the binding motifs of co-binding factors [38]. Complex models can improve their predictive performance by scoring sequences highly that contain any of these co-occuring motifs. This is possible even within a short motif length by learning the motifs superposed with each other, with the higher orders preventing mixing and blurring of motifs [29]. Although improving the apparent model performance, such models do not describe faithfully the binding affinity of the ChIPped factor.

Our goal was to compare PWM-based motif discovery tools with tools employing more complex models: dinucleotide weight matrices, parsimonious context trees, and BaMMs. We therefore set up a cross-cell line benchmark to assess how well the motif models learned in one cell line can predict binding in another cell line. Furthermore, we conducted a cross-platform benchmark, in which we trained the models on ChIP-seq data and tested them on HT-SELEX data, and vice versa. The results show that among the tested tools, those with more complex models still tend to perform better in these benchmarks, albeit with smaller improvements over the PWM-based tools. The improvements from higher orders were particularly marked for the BaMMs. So, most of the information in higher orders seems to be transferable between cell lines and measurement platforms.

Even though we did not see clear signs of overfitting in our BaMMs, we introduced sequence masking as a precaution against overfitting to other motifs and technology- or cell line-dependent sequence biases. We use the seed PWM to mask out all but the top-scoring 5% of positions, and we train the higher-order BaMM only on the remaining 5%. We thereby ensure that only sequence regions that actually carry the seed motif can be learned by the BaMM. The performance drop between training fifth-order BaMMs with and without masking was 8% on HT-SELEX data and 12% on ChIP-seq data (Fig. 2D,G, Fig. S 4). This indicates that if higher-order BaMMs profit from learning co-occurring motifs at all, the effect on their performance is quite limited. Still, if the goal is to learn binding affinities and not just predict motifs from *in-vivo* sequence data, we recommend to run BaMMmotif2 with the masking. On *in-vitro* data, however, masking is not necessary and in order to make use of the 5% improvement we recommend to run BaMMmotif2 without masking. However, even with masking the fifth-order BaMMs still perform competitively with the state-of-the-art tools while being significantly faster.

Transcription factors combine base- with DNA shape readout [13]. Instead of studying the TF-DNA binding using only the sequence features, some models utilize DNA shape features predicted from the sequence to enhance motif models [51–53]. The shape descriptors these tools use, like minor groove width, helical tilt and bent, or propeller tilt, are predicted from five-mer tables computed using molecular dynamics calculations. Given enough data, it is therefore evident that higher-order models such as BaMMs can learn these DNA structural preferences implicitly, yet are not limited to the pre-defined shape descriptors.

In recent years, deep learning approaches have become popular for learning motif models with very good predictive performance [51, 54, 55]. Such models usually take advantage

of contextual information such as co-occurring motifs etc, which increases their predictive power but serves a different purpose than the models we discuss here: learning a model for the sequence dependence of the binding affinity of a factor. In addition, BaMMs have the advantage or being conceptionally simple and interpretable in terms of k-mer dependent energy terms.

In conclusion, we have shown that higher order models for binding motifs improved binding site predictions on a large collection of ChIP-seq and HT-SELEX datasets, both in cross-validated setting and when training and testing on different experimental platforms and cell lines. Importantly, clear improvements in predictive performance are even seen beyond first order models: BaMMs of fifth order show a solidly improved performance across the bench over the tested state of the art tools, while being significantly faster.

AVAILABILITY

Data

ENCODE database

We evaluated the performance of selected algorithms on human ChIP-seq datasets from the ENCODE portal [36] until March 2012. In total, there are 435 datasets for 93 distinct transcription factors. The top 5000 peak regions sorted by their signal value are selected for each dataset when peaks are more than 5000, and all peaks are chosen if there are fewer than 5000 peaks. Positive sequences are extracted ± 104 bp around the peak summits. Background sequences are sampled by the trimer frequencies from positive sequences, with the same lengths as positive sequences and 10 times the amount of positive sequences. 8 datasets are excluded from all the results because diChIPMunk fails to learn models within 3 hours.

HT-SELEX datasets

For HT-SELEX data, we downloaded 164 datasets with 200 bp-long oligomers from Zhu et al. [37], which are deposited in the European Nucleotide Archive (ENA) under the accession PRJEB22684. Each dataset represents one non-redundant human transcription factor. For each dataset, we selected the first 5000 sequences from each selection round without any sorting.

The HT-SELEX data contain reads from at least four selection cycles, and the measured binding affinity iteratively increases with the cycles. Thus, we chose the sequences from the fourth selection rounds with detected high affinities for motif training and testing in the main paper. Since ChIPMunk and diChIPMunk took longer than 2 hours to run on the full datasets, we selected 5000 sequences out of the millions of reads as training and test sequences. To examine the power of BaMMs in learning the weak binding sites, we also used sequences from the second and third selection rounds. Background sequences are sampled in the same way as described previously.

Software and parameters

The new version of BaMMmotif2 software is implemented in C++ and Python3. The code is licensed under GPLv3 and freely accessible without registration at

github.com/soedinglab/PEnG-motif, and github.com/soedinglab/BaMMmotif2, and supported on Linux and MacOS. They are also integrated into our webserver [32].

Results and analysis scripts

The analysis scripts are available in Jupyter Notebook format at github.com/soedinglab/bamm-benchmark.

FUNDING

This work was supported by the DFG SPP1935 grant CR 117/6–1 and the International Max Planck Research School for Genome Science (IMPRS-GS). Funding for open access charge: institutional.

ACKNOWLEDGMENTS

We thank Matthias Siebert and Anja Kiesel for help with designing the code structure of BaMMmotif2, the members of the Söding lab, especially Eli Levy Karin, Saikat Banerjee, Milot Mirdita and Ruoshi Zhang for discussions, and the scientific community for sharing their data.

Author Contribution: W.G. developed the BaMMmotif2 software. M.M., C.R. and J.S. developed the seeding stage software (PEnG). W.G. implemented the statistical approach and conducted all the benchmarks. W.G. and J.S. wrote the manuscript. J.S. supervised the research.

Conflict of interest statement. None declared.

References

1. Serfling, E., Jasin, M., and Schaffner, W. (1985) Enhancers and eukaryotic gene transcription. *Trends Genet.,* **1**, 224–230.

2. Argos, P. (1988) A sequence motif in many polymerases. *Nucleic Acids Res.,* **16**(21), 9909–9916.

3. Mitchell, P. J. and Tjian, R. (1989) Transcriptional regulation in mammalian cells by sequence-specific DNA binding proteins. *Science,* **245**(4916), 371–378.

4. Jolma, A. and Taipale, J. (2011) Methods for analysis of transcription factor DNA-binding specificity in vitro. In *A Handbook of Transcription Factors* pp. 155–173 Springer.

5. Robertson, G., Hirst, M., Bainbridge, M., Bilenky, M., Zhao, Y., Zeng, T., Euskirchen, G., Bernier, B., Varhol, R., Delaney, A., et al. (2007) Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing. *Nat. Methods,* **4**(8), 651–657.

6. Meng, X., Brodsky, M. H., and Wolfe, S. A. (2005) A bacterial one-hybrid system for determining the DNA-binding specificity of transcription factors. *Nat. Biotechnol.,* **23**(8), 988–994.

7. Jolma, A., Yan, J., Whitington, T., Toivonen, J., Nitta, K. R., Rastas, P., Morgunova, E., Enge, M., Taipale, M., Wei, G., et al. (2013) DNA-binding specificities of human transcription factors. *Cell,* **152**(1-2), 327–339.

8. Riley, T. R., Slattery, M., Abe, N., Rastogi, C., Liu, D., Mann, R. S., and Bussemaker, H. J. (2014) SELEX-seq: a method for characterizing the complete repertoire of binding site preferences for transcription factor complexes. In *Hox Genes* pp. 255–278 Springer.

9. Isakova, A., Groux, R., Imbeault, M., Rainer, P., Alpern, D., Dainese, R., Ambrosini, G., Trono, D., Bucher, P., and Deplancke, B. (2017) SMiLE-seq identifies binding motifs of single and dimeric transcription factors. *Nat. Methods,* **14**(3), 316–322.

10. Man, T.-K. and Stormo, G. D. (2001) Non-independence of Mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay. *Nucleic Acids Res.,* **29**(12), 2471–2478.

11. Bulyk, M. L., Johnson, P. L., and Church, G. M. (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.,* **30**(5), 1255–1261.

12. Benos, P. V., Lapedes, A. S., and Stormo, G. D. (2002) Probabilistic code for DNA recognition by proteins of the EGR family. *J. Mol. Biol.,* **323**(4), 701–727.

13. Rohs, R., West, S. M., Sosinsky, A., Liu, P., Mann, R. S., and Honig, B. (2009) The role of DNA shape in protein–DNA recognition. *Nature,* **461**(7268), 1248–1253.

14. Gordân, R., Shen, N., Dror, I., Zhou, T., Horton, J., Rohs, R., and Bulyk, M. L. (2013) Genomic regions flanking E-box binding sites influence DNA binding specificity of bHLH transcription factors through DNA shape. *Cell Rep.,* **3**(4), 1093–1104.

15. Fordyce, P. M., Pincus, D., Kimmig, P., Nelson, C. S., El-Samad, H., Walter, P., and DeRisi, J. L. (2012) Basic leucine zipper transcription factor Hac1 binds DNA in two distinct modes as revealed by microfluidic analyses. *Proc. Natl. Acad. Sci. U.S.A.,* **109**(45), E3084–E3093.

16. Zuo, Z. and Stormo, G. D. (2014) High-resolution specificity from DNA sequencing highlights alternative modes of Lac repressor binding. *Genetics,* **198**(3), 1329–1343.

17. Halazonetis, T. D., Georgopoulos, K., Greenberg, M. E., and Leder, P. (1988) c-Jun dimerizes with itself and with c-Fos, forming complexes of different DNA binding affinities. *Cell,* **55**(5), 917–924.

18. Slattery, M., Riley, T., Liu, P., Abe, N., Gomez-Alcala, P., Dror, I., Zhou, T., Rohs, R., Honig, B., Bussemaker, H. J., et al. (2011) Cofactor binding evokes latent differences in DNA binding specificity between Hox proteins. *Cell,* **147**(6), 1270–1282.

19. Crocker, J., Noon, E. P.-B., and Stern, D. L. (2016) The soft touch: low-affinity transcription factor binding sites in development and evolution. In *Curr. Top. Dev. Biol.* Vol. 117, pp. 455–469 Elsevier.

20. Kribelbauer, J. F., Rastogi, C., Bussemaker, H. J., and Mann, R. S. (2019) Low-Affinity Binding Sites and the Transcription Factor Specificity Paradox in Eukaryotes. *Annu. Rev. Cell Dev. Biol.,* **35**, 357–379.

21. Jiang, J. and Levine, M. (1993) Binding affinities and cooperative interactions with bHLH activators delimit threshold responses to the dorsal gradient morphogen. *Cell,* **72**(5), 741–752.

22. Rastogi, C., Rube, H. T., Kribelbauer, J. F., Crocker, J., Loker, R. E., Martini, G. D., Laptenko, O., Freed-Pastor, W. A., Prives, C., Stern, D. L., et al. (2018) Accurate and sensitive quantification of protein-DNA binding affinity. *Proc. Natl. Acad. Sci. U.S.A.,* **115**(16), E3692–E3701.

23. Mathelier, A. and Wasserman, W. W. (2013) The next generation of transcription factor binding site prediction. *PLoS Comput. Biol.,* **9**(9), e1003214.

24. Eggeling, R., Grosse, I., and Grau, J. (2017) InMoDe: tools for learning and visualizing intra-motif dependencies of DNA binding sites. *Bioinformatics,* **33**(4), 580–582.

25. Gershenzon, N. I., Stormo, G. D., and Ioshikhes, I. P. (2005) Computational technique for improvement of the position-weight matrices for the DNA/protein binding sites. *Nucleic Acids Res.,* **33**(7), 2290–2301.

26. Siddharthan, R. (2010) Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PloS One,* **5**(3), e9722.

27. Kulakovskiy, I., Levitsky, V., Oshchepkov, D., Bryzgalov, L., Vorontsov, I., and Makeev, V. (2013) From binding motifs in ChIP-Seq data to improved models of transcription factor binding sites. *J. Bioinform. Comput. Biol.,* **11**(01), 1340004.

28. Siebert, M. and Söding, J. (2016) Bayesian Markov models consistently outperform PWMs at predicting motifs in nucleotide sequences. *Nucleic Acids Res.,* **44**(13), 6055–6069.

29. Eggeling, R. (2018) Disentangling transcription factor binding site complexity. *Nucleic Acids Res.,* **46**(20), e121.

30. Orenstein, Y. and Shamir, R. (2014) A comparative analysis of transcription factor binding models learned from PBM, HT-SELEX and ChIP data. *Nucleic Acids Res.,* **42**(8), e63.

31. Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. (2009) MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res.,* **37**(suppl_2), W202–W208.

32. Kiesel, A., Roth, C., Ge, W., Wess, M., Meier, M., and Söding, J. (2018) The BaMM web server for de-novo motif discovery and regulatory sequence analysis. *Nucleic Acids Res.,* **46**(W1), W215–W220.

33. Sharov, A. A. and Ko, M. S. (2009) Exhaustive search for over-represented DNA sequence motifs with CisFinder. *DNA Res.,* **16**(5), 261–273.

34. Kulakovskiy, I. V., Boeva, V., Favorov, A. V., and Makeev, V. J. (2010) Deep and wide digging for binding motifs in ChIP-Seq data. *Bioinformatics,* **26**(20), 2622–2623.

35. Kulakovskiy, I. V., Vorontsov, I. E., Yevshin, I. S., Sharipov, R. N., Fedorova, A. D., Rumynskiy, E. I., Medvedeva, Y. A., Magana-Mora, A., Bajic, V. B., Papatsenko, D. A., et al. (2018) HOCOMOCO: towards a complete collection of transcription factor binding models for human and mouse via large-scale ChIP-Seq analysis. *Nucleic Acids Res.,* **46**(D1), D252–D259.

36. ENCODE Project Consortium and others (2012) An integrated encyclopedia of DNA elements in the human genome. *Nature,* **489**(7414), 57–74.

37. Zhu, F., Farnung, L., Kaasinen, E., Sahu, B., Yin, Y., Wei, B., Dodonova, S. O., Nitta, K. R., Morgunova, E., Taipale, M., et al. (2018) The interaction landscape between transcription factors and the nucleosome. *Nature,* **562**(7725), 76–81.

38. Hunt, R. W. and Wasserman, W. W. (2014) Non-targeted transcription factors motifs are a systemic component of ChIP-seq datasets. *Genome Biol.,* **15**(7), 412.

39. Chen, Y., Negre, N., Li, Q., Mieczkowska, J. O., Slattery, M., Liu, T., Zhang, Y., Kim, T.-K., He, H. H., Zieba, J., et al. (2012) Systematic evaluation of factors influencing ChIP-seq fidelity. *Nat. Methods,* **9**(6), 609–614.

40. Aird, D., Ross, M. G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., Jaffe, D. B., Nusbaum, C., and Gnirke, A. (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.,* **12**(2), R18.

41. Diaz, A., Park, K., Lim, D. A., and Song, J. S. (2012) Normalization, bias correction, and peak calling for ChIP-seq. *Stat. Appl. Genet. Mol. Biol,* **11**(3).

42. Teytelman, L., Özaydın, B., Zill, O., Lefrançois, P., Snyder, M., Rine, J., and Eisen, M. B. (2009) Impact of chromatin structures on DNA processing for genomic analyses. *PloS One,* **4**(8), e6700.

43. Zhao, Y., Granas, D., and Stormo, G. D. (2009) Inferring binding energies from selected binding sites. *PLoS Comput. Biol.,* **5**(12), e1000590.

44. Jolma, A., Kivioja, T., Toivonen, J., Cheng, L., Wei, G., Enge, M., Taipale, M., Vaquerizas, J. M., Yan, J., Sillanpää, M. J., et al. (2010) Multiplexed massively parallel SELEX for characterization of human transcription factor binding specificities. *Genome Res.,* **20**(6), 861–873.

45. Levo, M., Zalckvar, E., Sharon, E., Machado, A. C. D., Kalma, Y., Lotam-Pompan, M., Weinberger, A., Yakhini, Z., Rohs, R., and Segal, E. (2015) Unraveling determinants of transcription factor binding outside the core binding site. *Genome Res.,* **25**(7), 1018–1029.

46. Schöne, S., Jurk, M., Helabad, M. B., Dror, I., Lebars, I., Kieffer, B., Imhof, P., Rohs, R., Vingron, M., Thomas-Chollier, M., et al. (2016) Sequences flanking the core-binding site modulate glucocorticoid receptor structure and activity. *Nat. Commun.,* **7**(1), 12621.

47. Yella, V. R., Bhimsaria, D., Ghoshdastidar, D., Rodríguez-Martínez, J. A., Ansari, A. Z., and Bansal, M. (2018) Flexibility and structure of flanking DNA impact transcription factor affinity for its core motif. *Nucleic Acids Res.,* **46**(22), 11883–11897.

48. Merkenschlager, M. and Nora, E. P. (2016) CTCF and cohesin in genome folding and transcriptional gene regulation. *Annu. Rev. Genomics Hum. Genet.,* **17**(1), 17–43.

49. Zhang, R., Wang, Y., Yang, Y., Zhang, Y., and Ma, J. (2018) Predicting CTCF-mediated chromatin loops using CTCF-MP. *Bioinformatics,* **34**(13), i133–i141.

50. Khan, A., Fornes, O., Stigliani, A., Gheorghe, M., Castro-Mondragon, J. A., van der Lee, R., Bessy, A., Cheneby, J., Kulkarni, S. R., Tan, G., et al. (2018) JASPAR 2018: update of the open-access database of transcription factor binding profiles and its web framework. *Nucleic Acids Res.,* **46**(D1), D260–D266.

51. Mathelier, A., Xin, B., Chiu, T.-P., Yang, L., Rohs, R., and Wasserman, W. W. (2016) DNA shape features improve transcription factor binding site predictions in vivo. *Cell Syst.,* **3**(3), 278–286.

52. Peng, P.-C. and Sinha, S. (2016) Quantitative modeling of gene expression using DNA shape features of binding sites. *Nucleic Acids Res.,* **44**(13), e120–e120.

53. Samee, M. A. H., Bruneau, B. G., and Pollard, K. S. (2019) A de novo shape motif discovery algorithm reveals preferences of transcription factors for DNA shape beyond sequence motifs. *Cell Syst.,* **8**(1), 27–42.

54. Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015) Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat. Biotechnol.,* **33**(8), 831–838.

55. Kelley, D. R., Snoek, J., and Rinn, J. L. (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.,* **26**(7), 990–999.

56. Yevshin, I., Sharipov, R., Kolmykov, S., Kondrakhin, Y., and Kolpakov, F. (2019) GTRD: a database on gene transcription regulation—2019 update. *Nucleic Acids Res.,* **47**(D1), D100–D105.

57. Strimmer, K. (2008) fdrtool: a versatile R package for estimating local and tail area-based false discovery rates. *Bioinformatics,* **24**(12), 1461–1462.

58. Fordyce, P. M., Gerber, D., Tran, D., Zheng, J., Li, H., DeRisi, J. L., and Quake, S. R. (2010) De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis. *Nat. Biotechnol.,* **28**(9), 970–975.

Part I

Supplemental Figures

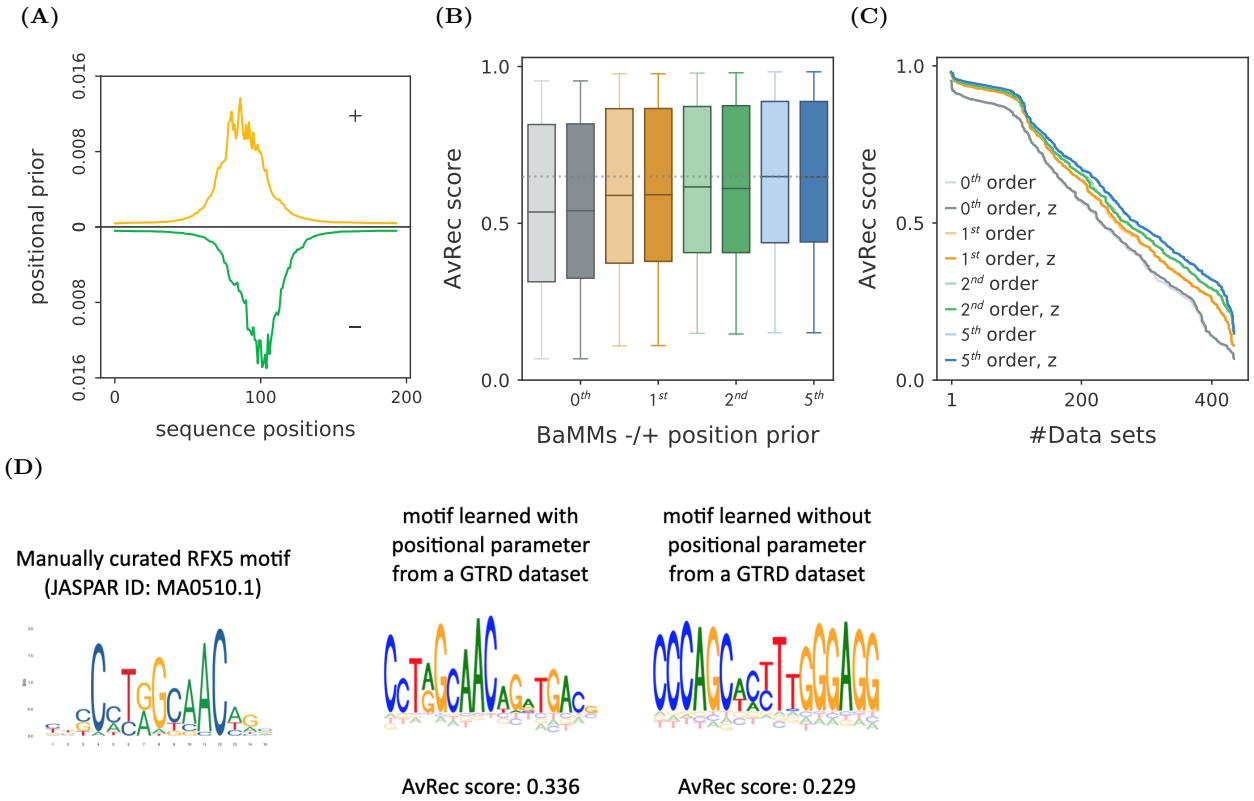

Figure S 1. Optimization of positional prior on *in vivo* data. BaMMs with zeroth- (grey), first- (orange), second- (green) and fifth-orders (blue) are trained and tested on 435 ENCODE datasets using 5-fold cross-validation. Panel (**A**) shows the distribution of optimized positional priors over the positions on both sequence strands that are center around ChIP-seq summits for GABPα motif. Panel (**B**) shows the AvRec distributions as box plot, with boxes indicating 25%/75% quantiles and whiskers 5%/95% quantiles. The colors are for different orders with- (dark colors) and without (light colors) optimization of positional prior in the motif training. Panel (**C**) shows the cumulative distributions of AvRec scores on 435 datasets. There is no major difference before and after the positional prior optimization. Panel (**D**) shows the motif for RFX5 factor learned from a GTRD dataset [56] with (middle) and without (right) the optimized positional parameter, compared to the reference motif reported in the JASPAR database [50] (left).

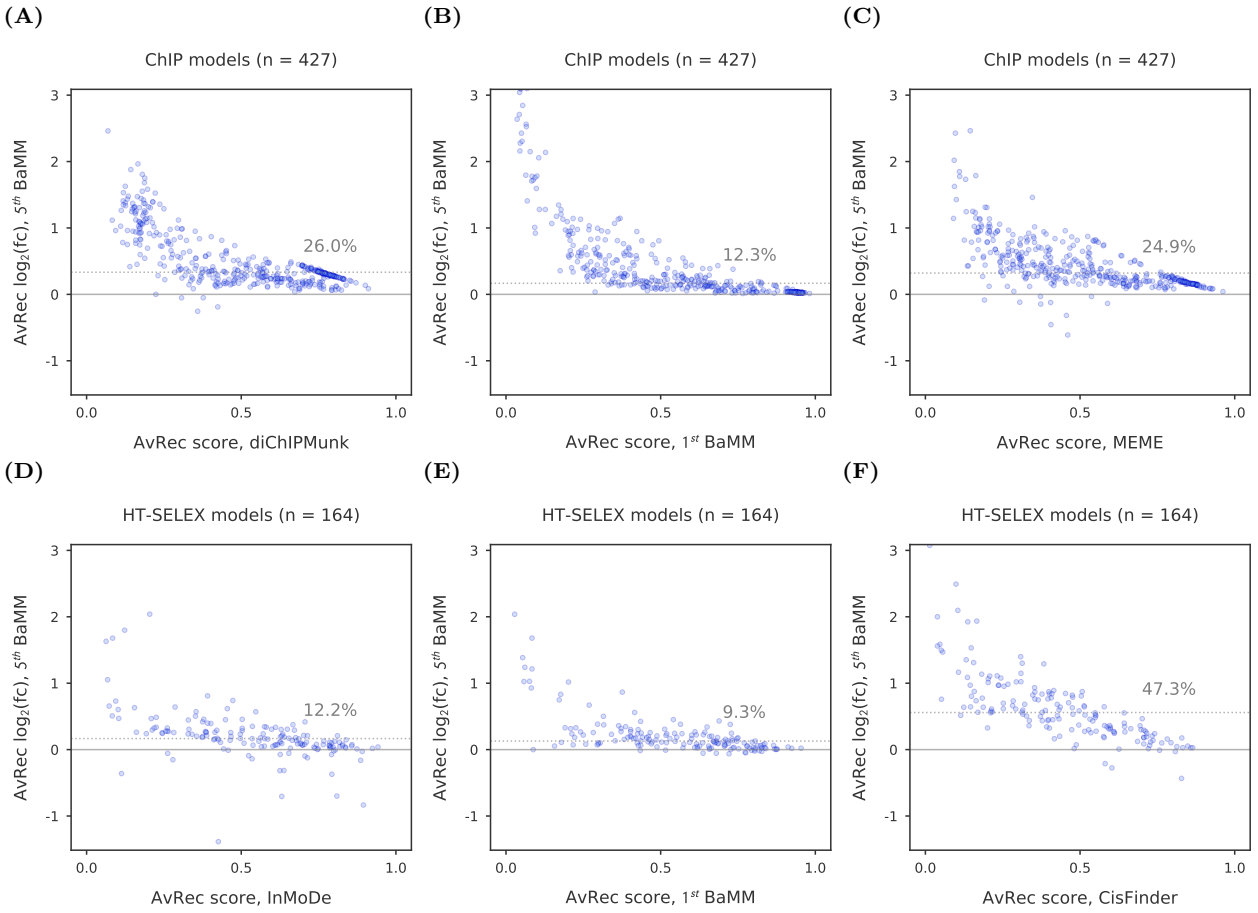

Figure S 2. Performance comparison of motif discovery tools on *in vivo* and *in vitro* data. $\log_2$ of fold change in AvRec between fifth-order BaMMmotif2 and diChIPMunk models versus AvRec of diChIPMunk models (**A**), first-order BaMMmotif2 (**B**) and MEME PWMs (**C**) on 427 ChIP-seq datasets. Each dot represents the test on one dataset from either ChIP-seq or HT-SELEX. The grey dashed lines indicate the median $\log_2$ fold change is 26%, 12.4% and 24.9% respectively. (**D-F**) Similar comparisons as (**A-C**) but on 164 HT-SELEX datasets.

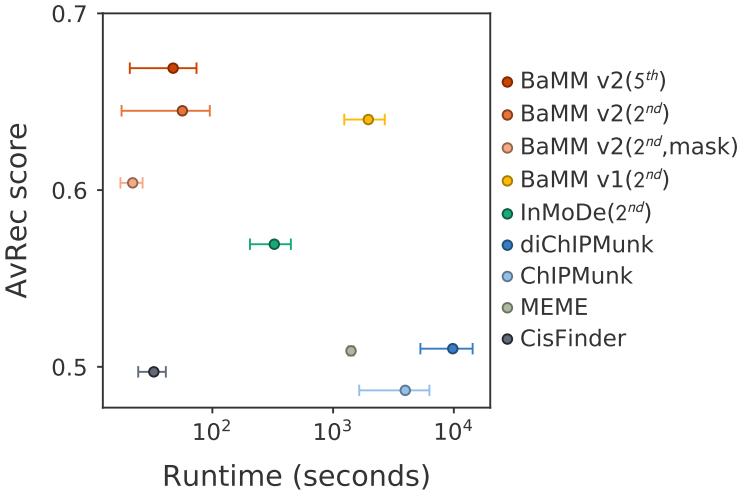

Figure S 3. Benchmark on *in vivo* data. Average runtime per dataset on a server with 4 cores versus the median AvRec score of several *de novo* motif discovery tools, including the previous version of BaMMmotif, validated on 419 datasets with 5-fold cross-validation, with MEME, CisFinder, BaMMmotif and BaMMmotif2 running on 4 computation cores. Whiskers indicate the standard deviation of AvRec score. 2nd-order models trained using BaMMmotif and BaMMmotif2 have similar average AvRec scores, yet BaMMmotif2 is $>\tilde{1}0$ times faster than BaMMmotif.

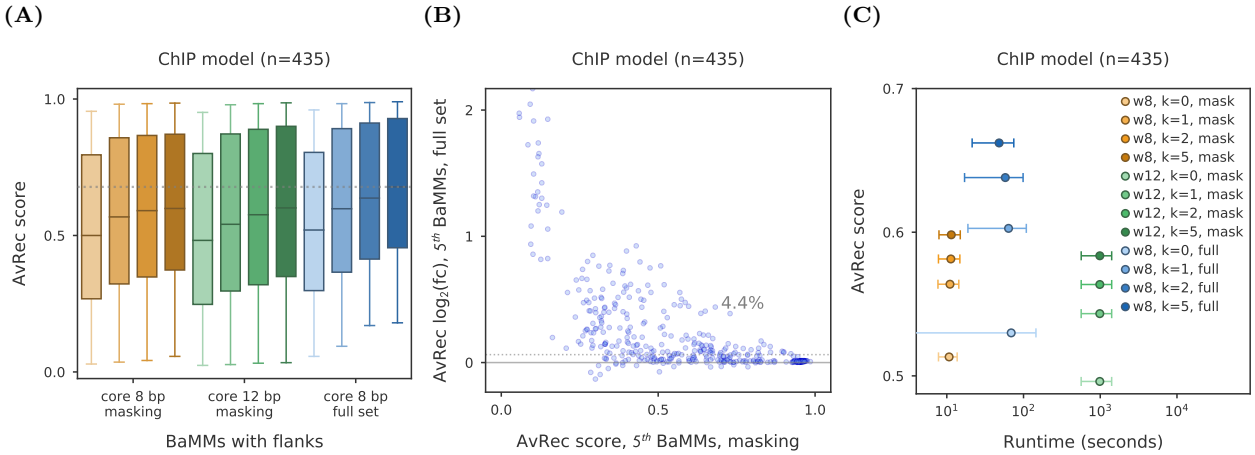

Figure S 4. EM optimization using the full set compared to masking 95% sequences on ChIP-seq datasets. (**A**) Using the full set of sequences for the EM optimization (blue) improves the performance of higher-order models, while extending the core regions for searching the enriched patterns (green) does not contribute to motif discovery, in comparison to that with 8 bp for seeding and masking 95% sequences for the EM optimization (yellow). All box-plot whiskers show 95th/5th percentile. Each cluster contains models with different orders (zeroth-, first-, second- and fifth-order). (**B**) Fifth-order BaMMs optimized on the full sequences set have a 4.4% AvRec fold increase compared to those trained only 5% sequences. (**C**) Using a masking step improves the speed by 10-fold, in comparison to using the full set for learning motif model.

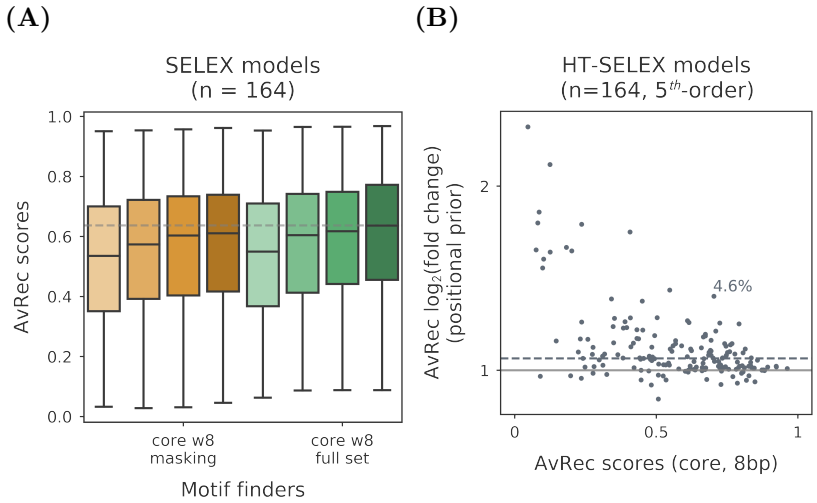

Figure S 5. EM optimization using the full set compared to masking 95% sequences from HT-SELEX datasets. (**A**) Using the full set for motif refinement (green) improves the performance of higher-order models over that using only 5% sequences (yellow). All box-plot whiskers show 95th/5th percentile. Each cluster contains models with different orders (zeroth-, first-, second- and fifth-order). (**B**) Fifth-order BaMMs with full set of sequences for optimization has a 4.6% AvRec fold increase compared to it with only 5% sequences for optimization.

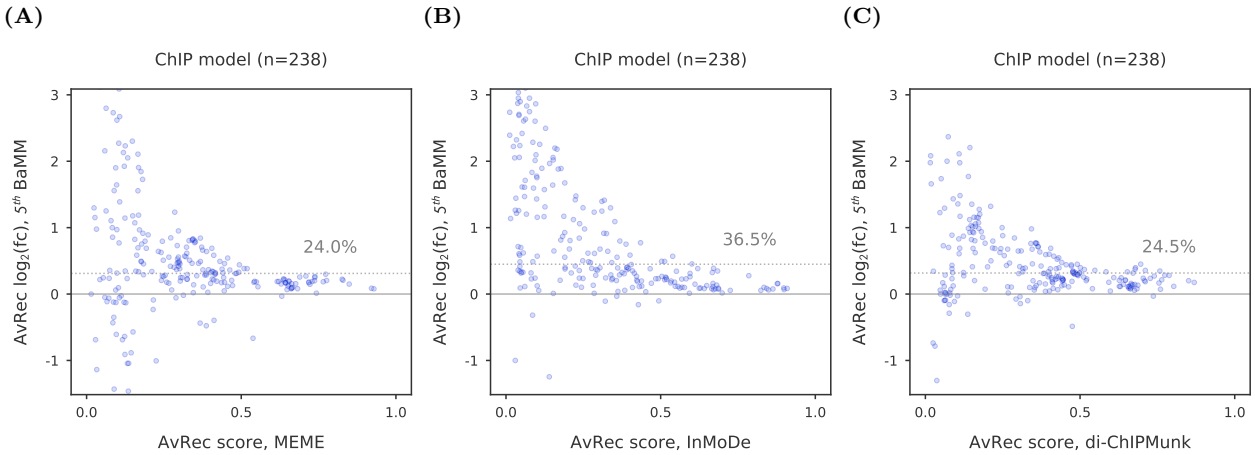

Figure S 6. Cross-cell-line validation. $\log_2$ of fold change in AvRec between fifth-order BaMMmotif2 and PWMs from MEME (**A**), second-order models from InMoDe (**B**), and first-order models from diChIPMunk (**C**), when comparing to AvRec scores of the latter models in 238 paired ENCODE datasets. Each dot represents one test. The range of AvRec scores is chosen from 0.5 to 8 and the outliers are not shown in these plots.

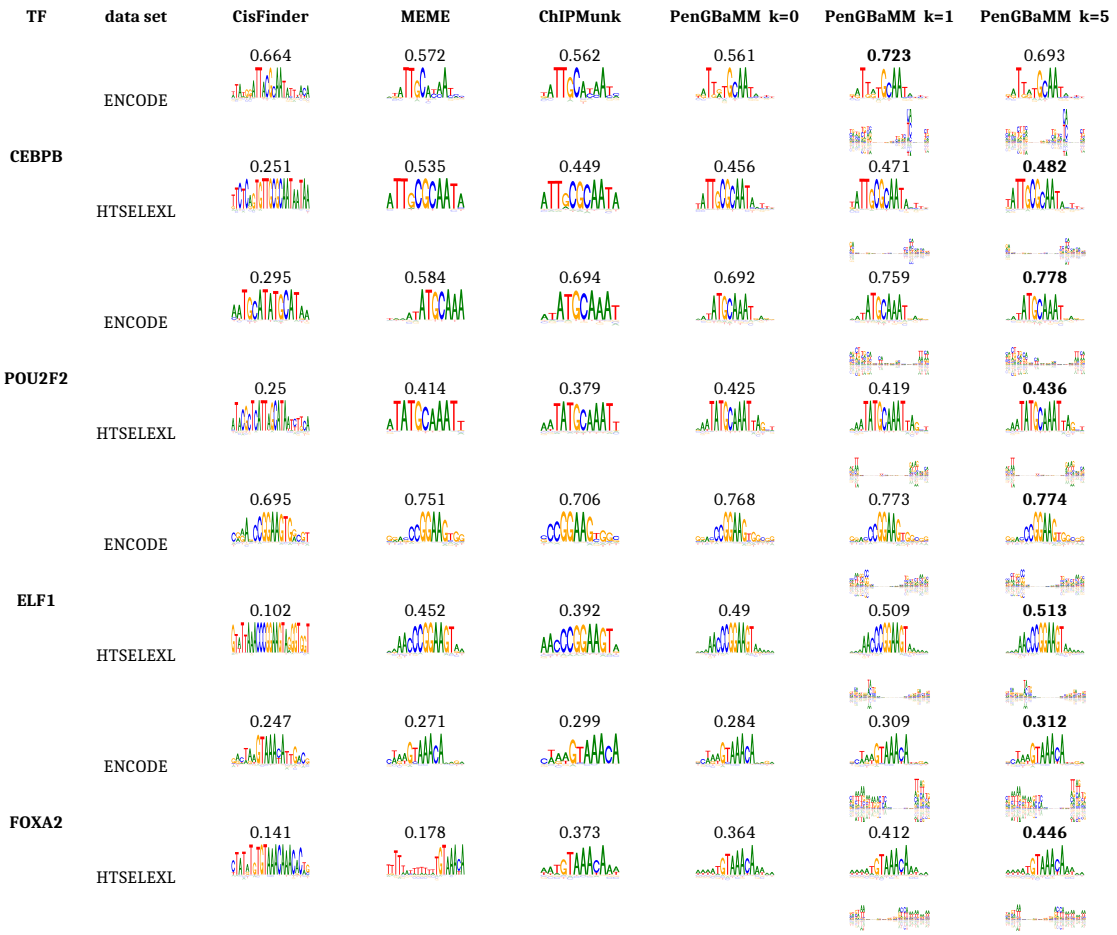

Figure S 7. Sequence logos and AvRec scores of motifs models from cross-platform validation. Motif models are trained by different models for four transcription factors: CEBPB, POU2F2, ELF1, and FOXA2. For each transcription factor, the first row shows models learned on ENCODE data by applying different tools. The number above each logo represents the AvRec score when testing the model on the corresponding HT-SELEX data. The second row shows the models learned on HT-SELEX data and AvRec scores when testing models on ENCODE data. For BaMM models, both zeroth- and first-order logos are plotted.

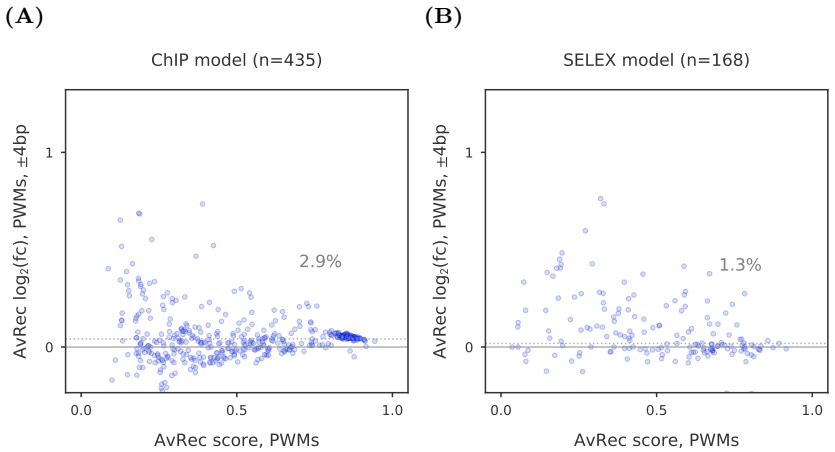

Figure S 8. Impact of extending core motif regions on PWMs. (**A**) Log_2 of fold change between PWM models with ± 4 bp flanking positions and no added flanking positions, using 435 datasets. Median AvRec change is 2.9 %. (**B**) Same as (**A**) but on 168 HT-SELEX datasets.

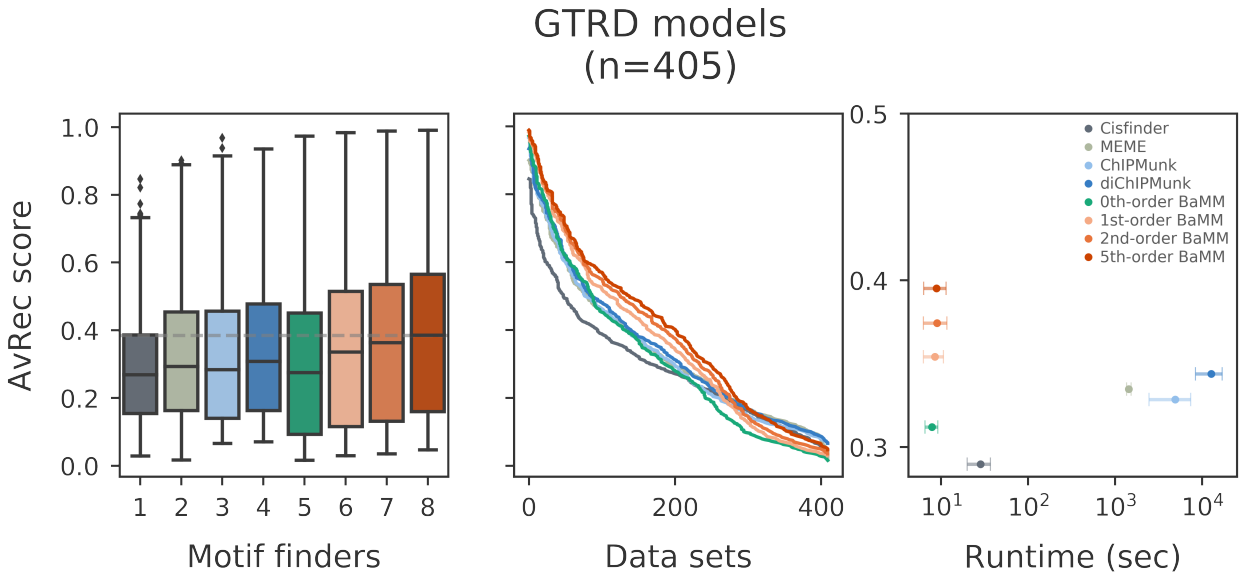

Figure S 9. Quantitative performance on *in vivo* GTRD datasets. The selected tools are applied to 405 GTRD datasets [56] and their Average Recall were calculated by 5-fold cross-validation, similar to 2. (**A**) AvRec distributions as box plot. All box-plot whiskers show 95th/5th percentile. (**B**) The cumulative of AvRec scores on 405 datasets. (**C**) Average runtime per dataset on a server with 4 cores versus the median AvRec score. Whiskers: standard deviation.

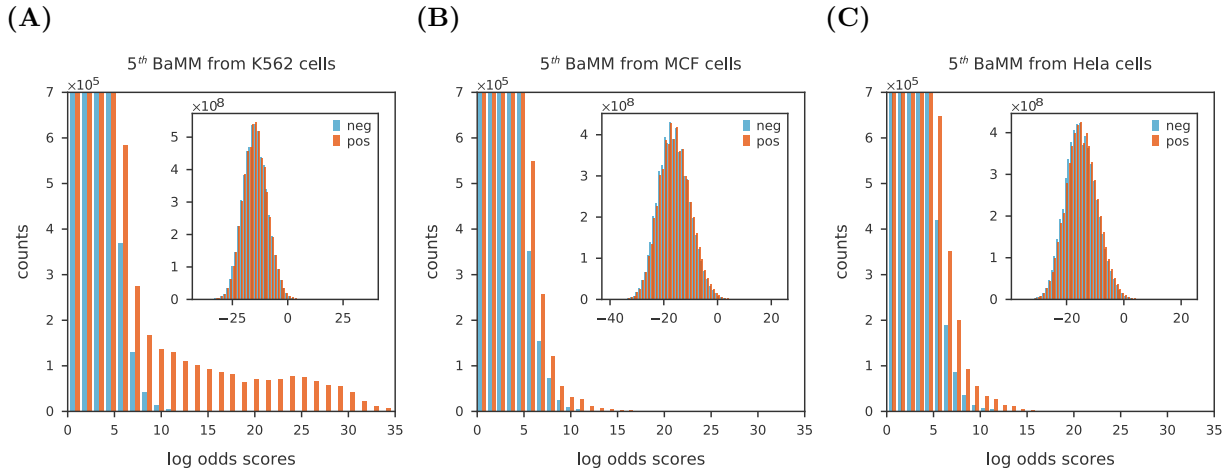

Figure S 10. Distributions of CTCF motif scores across the whole human genome (hg19), scanned by different BaMMs The distribution of log odds scores that were calculated on all the binding sites with a 5th-order BaMM of length 67 bp trained on ChIP-seq datasets either on K562 (**A**), MCF (**B**) or HeLa (**C**) cell lines on the whole genome (orange) and reverse genome (blue). The background model is a second-order Markov model. Inset: the full distributions of on the whole genome (orange) and reverse genome (blue). Note that the scores are the only 1% of the total that are randomly drawn from the full distribution of the whole genome.

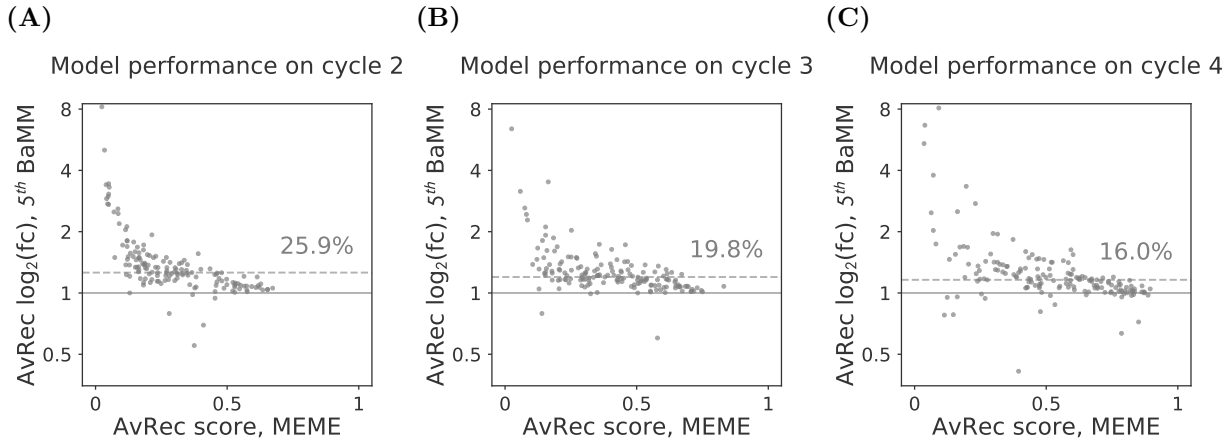

Figure S 11. Performance comparison of BaMMmotif versus MEME on weak binding prediction. $\log_2$ fold change between fifth-order BaMMmotif2 models and MEME models versus AvRec of MEME, with AvRec analyzed by 5-fold cross-validation on sequences from the 2nd- (**A**), 3rd- (**B**) or 4th- (**C**) selection cycle of 164 HT-SELEX datasets. The median fold change increases are 25.9%, 19.8% and 16%, respectively (grey dashed lines). Each dot represents one data set.

Part II

Supplemental Methods

The supplemental material provides further details of the theoretical basis, the implementation of the BaMMmotif2 package, and the processing procedure of the datasets that are used for this benchmark. It also documents the parameters used for testing the motif discovery tools in the benchmark. It ensures the reproducibility of the results in this paper.

1 *De novo* motif discovery and refinement

1.1 The fast seeding phase: PEnGmotif

We describe PEnGmotif (Pattern-based discovery of enriched genomic or transcriptomic sequence motifs), an efficient method for discovering sequence patterns enriched in a set of nucleotide sequences over random expectation sampled from a second-order background model. The enriched patterns found by PEnGmotif are optimized to PWMs and serve as seeds to initialise the refinement stage by BaMMmotif2.

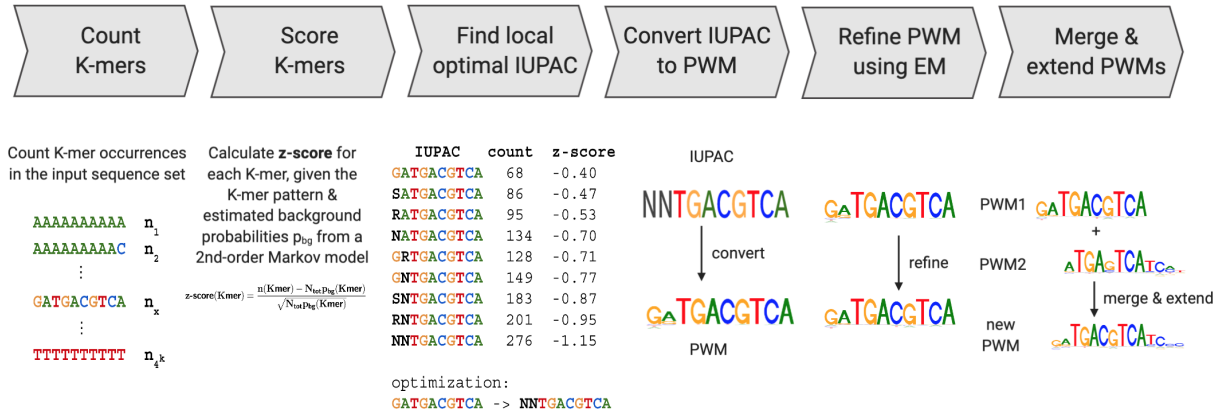

Figure S 12. Workflow of the fast seeding stage. Sequences from high-throughput assays, such as ChIP-seq, SELEX and PBM, are provided as input data. (i) Occurrences of all K-mers of a fixed specified length (default 10) are counted. (ii) An enrichment z-score is calculated for each K-mer based on a Poisson model. (iii) High-scored K-mers are optimized from the nucleotide alphabet (ACGT) to a degenerate IUPAC alphabet with 11 letters (ACGTRYSWSKN). (iv) The locally optimal IUPAC patterns are converted to PWMs. (v) PWMs are refined using the Expectation Maximisation algorithm. (vi) PWMs with similar overlapping regions are merged and extended.

Let K be the length of patterns that will be analysed (e.g. $K = 8$ used in the study). First, the number of occurrences of each of the 4^K non-degenerate seed patterns of length K are counted in a 4^K-dimensional array with $\mathbf{x} \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^K$. $p_{\mathrm{bg}}(\mathbf{x})$ denotes the probability of observing K-mer $\mathbf{x}$ in absence of specific binding. $p_{\mathrm{bg}}(\mathbf{x})$ can be directly counted from large background sequence sets or modelled as a homogeneous Markov model on a background data set or the dataset itself. For example, $p_{\mathrm{bg}}(\mathbf{x})$ is learned from the genomic input, a mock

immunoprecipitation or the input sequence library prior to the selection in HT-SELEX. We model the background probability using a homogeneous Markov model of order K' ($K' = 2$ by default):

$$p_{\mathrm{bg}}(x_{i_0:i_1}) = \prod_{i=i_0}^{i_1} p_{\mathrm{bg}}(x_i | x_{i-K':\,i-1}). \tag{2}$$

We assume the number of occurrences in absence of specific binding to follow a Poisson distribution: $\mu = L_{\mathrm{tot}} p_{\mathrm{bg}}(\mathbf{y})$, where $L_{\mathrm{tot}} = \sum_{n=1}^{N}(L_n - K + 1)$ is the total number of all counted patterns in the input sequences (N is the total sequence number and L_n is the length of n'th sequence).

z-score We compute Z-scores for all non-degenerate K-mer patterns. The Z-score is the deviation from expectation divided by the standard deviation. As for the Poisson distribution the variance equals the mean, the Z-score is:

$$Z(\mathbf{y}) = \frac{n(\mathbf{y}) - L_{\mathrm{tot}} p_{\mathrm{bg}}(\mathbf{y})}{\sqrt{L_{\mathrm{tot}} p_{\mathrm{bg}}(\mathbf{y})}}\,. \tag{3}$$

The z-score can be used to pre-filter what K-mers should enter the optimization routine.

p-value As we are also interested in highly enriched sequences (x) and (y) are fulfilled and we can use the Stirling approximation to calculate the p-value:

$$\begin{aligned}
\text{p-value}(\mathbf{y}) &= \sum_{k=n}^{\infty} \frac{\mu^k}{k!} e^{-\mu} \\
&= \frac{\mu^n}{n!} e^{-\mu} \sum_{k=0}^{\infty} \frac{\mu^k}{(n+1)\cdots(n+k)} \\
&\lessapprox \frac{\mu^n}{n!} e^{-\mu} \sum_{k=0}^{\infty} \frac{\mu^k}{(n+1)^k} \\
&\approx \frac{\mu^{n(\mathbf{y})}}{n(\mathbf{y})!} e^{-\mu} \frac{1}{1 - \mu/(n(\mathbf{y})+1)} \\
\log \text{p-value}(\mathbf{y}) &\approx n(\mathbf{y}) \log \frac{\mu}{n(\mathbf{y})} + n(\mathbf{y}) - \mu - \frac{1}{2}\log(2\pi n(\mathbf{y})) - \log\left(1 - \frac{\mu}{n(\mathbf{y})+1}\right)\,.
\end{aligned} \tag{4}$$

Mutual information We optimize the mutual information (MI) between two random variables,

$$\mathrm{MI}(q) = -qH(p_{\mathrm{obs}}) - (1-q)H(p_{\mathrm{exp}}) + H(p), \tag{5}$$

with $H(x) := -x\log x - (1-x)\log(1-x)$.

We then find locally optimal non-degenerate patterns with a recursive function which takes a K-mer $\mathbf{y}$ and checks for all its neighbouring K-mers, i.e. those that are at most one substitution away. If it finds a neighbouring $\mathbf{y}_{\mathrm{neigh}}$ with a better mutual information, the

function is called recursively with $\mathbf{y}_{\text{neigh}}$ as an argument. If no neighbour of $\mathbf{y}$ has better mutual information than $\mathbf{y}$, $\mathbf{y}$ is appended to the list of locally optimal K-mers. Similarly, we optimized the high-scored K-mers from the nucleotide alphabet (ACGT) to a degenerate IUPAC alphabet with 11 letters (ACGTRYSWSKN).

The IUPAC patterns can be transformed to PWMs based on the combined occurrences of all non-degenerated K-mers that match the degenerate IUPAC pattern in the input sequences. Alternatively, there is a faster approach based on the insight that if we allow any of the four nucleotides $a \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ at position j, the vast majority of motif matches will still be true positives due to the descriptive power of the other $K-1$ IUPAC letters. Therefore, we count the four nucleotides at motif position j for matches to the pattern $y_{0:j-1}\mathrm{N}y_{j+1:K-1}$ in which we replaced the jth IUPAC letter by an N:

$$p_{ja} = \frac{n(y_{0:j-1}\, a\, y_{j+1:K-1})}{n(y_{0:j-1}\, \mathrm{N}\, y_{j+1:K-1})}\,, \tag{6}$$

where we have called $n(\mathbf{y})$ the number of occurrences of K-mer $\mathbf{y}$ in the input set. Note that these PWM probabilities can be computed solely from the K-mer counts in a time $O(W \times D)$ that is independent of the size of the input dataset L_{tot}, and only depends on the degeneracy $D = \left|\{\mathbf{x} \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^W : \mathbf{x} \text{ matches } \mathbf{y}\}\right|$ of the motif $\mathbf{y}$, i.e., the number of different K-mers it matches.

We then refine the obtained PWMs by learning a multiple-occurrence-per-sequence model (MOPS) directly on the K-mer counts. The likelihood of a K-mer $\mathbf{x} \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}^K$ given a position weight matrix model with probabilities $\mathbf{p} = (p_j(\mathbf{A})$ is

$$\frac{p(\mathbf{x}|\mathbf{p}_{\text{motif}})}{p(\mathbf{x}|\mathbf{p}_{\text{bg}})} = \prod_{j=0}^{K-1} \frac{p_j(x_j)}{p_{\text{bg}}(x_j)}\,. \tag{7}$$

Expectation step: Compute the responsibilities $r(\mathbf{x})$, i.e., the probability that the factor will bind to K-mer $\mathbf{x}$.

$$r(\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{p}_{\text{motif}})/p(\mathbf{x}|\mathbf{p}_{\text{bg}})}{\sum_{\mathbf{x}' \in \{\mathrm{A,C,G,T}\}^K} n(\mathbf{x}')p(\mathbf{x}'|\mathbf{p}_{\text{motif}})/p(\mathbf{x}'|\mathbf{p}_{\text{bg}})} \tag{8}$$

Maximization step: Update the probabilities of the position weight matrix model.

$$p_j(\mathbf{A}) = \sum_{\mathbf{x} \in \{\mathrm{A,C,G,T}\}^K} I(x_j = a)\, n(\mathbf{x})\, r(\mathbf{x}) \tag{9}$$

By inserting the E-step equation into the M-step, we obtain

$$p_j^{(t)}(\mathbf{A}) \propto \sum_{\mathbf{x} \in \{\mathrm{A,C,G,T}\}^K} I(x_j = a)\, n(\mathbf{x})\, \frac{p(\mathbf{x}|\mathbf{p}_{\text{motif}}^{(t-1)})}{p(\mathbf{x}|\mathbf{p}_{\text{bg}})} \tag{10}$$

and subsequent normalisation for each j over $a \in \{\mathrm{A,C,G,T}\}$ yields the updated motif matrix probabilities.

To model saturation effects at the motifs with high affinities, we can use a saturation function that will limit the weight of the odds ratios to a maximum value A, e.g. $A = 1000$:

$$p_j^{(t)}(\mathbf{A}) \propto \sum_{\mathbf{x}\in\{\mathrm{A,C,G,T}\}^K} I(x_j{=}a)\, n(\mathbf{x}) \left(A^{-1} + \frac{p(\mathbf{x}|\mathbf{p}_{\mathrm{bg}})}{p(\mathbf{x}|\mathbf{p}_{\mathrm{motif}}^{(t-1)})} \right)^{-1} \tag{11}$$

In a thermodynamic interpretation, A is the odds ratio of sites that have an occupancy of 50% at the assumed concentration of the transcription factor in the nucleus.

Merging and extending PWMs. We can reduce the redundancy of the PEnG!motif output and more importantly, generate more specific and sensitive motifs by merging sub-motifs that describe parts of the same underlying biological motif. For that, we first compute a list of pairwise similarity scores between all PWMs $\{p^{(1)}, \ldots, p^{(M)}\}$ with P-values above a user-specified cutoff obtained in the last step. Here, $p_{ja}^{(m)}$ is the probability of observing a nucleotide a at the j'th position of that PWM. The similarity score $S(p^{(m)}, p^{(m')})$ is defined by the maximum similarity score $s(\cdot,\cdot)$ evaluated in the overlapping regions when the two patterns of length l and l' are shifted by $d = -2, -1, \ldots, l' - l + 2$ to each other:

$$S\Big(p^{(m)}, p^{(m')}\Big) = \max_{-2\le d\le l'-l+2} \left\{ s\Big(p_{j_1:j_2}^{(m)}, p_{j_1':j_2'}^{(m')}\Big) \right\} . \tag{12}$$

The indices defining the overlap region in the two PWMs are $j_1 = \max\{0, d\}, j_2 = \min\{l - 1, l' - 1 + d\}$ and $j_1' = \max\{0, -d\}, j_2' = \min\{l' - 1, l - 1 - d\}$. The similarity score between the PWMs in the overlap region is computed using

$$s(p, p') = \frac{1}{2}\left(d(p, p^{(\mathrm{bg})}) + d(p', p^{(\mathrm{bg})})\right) - d(p, p'), \tag{13}$$

The distance $d(p, p')$ between two PWMs p and p' of length l is the sum over the PWM columns of the relative entropies of each with their average distribution $\bar{p} := (p + p')/2$,

$$d(p, p') = \sum_{j=0}^{l-1} \left(H(p||\bar{p}) + H(p'||\bar{p})\right) = \sum_{j=0}^{l-1} \sum_{a\in\mathrm{A,C,G,T}} \left(p_{ja}\log_2 p_{ja} + p'_{ja}\log_2 p'_{ja} - 2\bar{p}_{ja}\log_2 \bar{p}_{ja}\right). \tag{14}$$

The pair with the highest score will be merged using the positional offset d that yielded the maximum similarity score. The pair of PWMs $(p^{(m)}, p^{(m')})$ has a score above a user-specified threshold ($0.4 \times W$ bits by default) are merged together using the positional offset d that yielded the maximum similarity score. In the overlapping regions, the nucleotide probabilities of merged PWM will be the weighted sum of the nucleotide probabilities of the two merged PWMs, where the weights are the numbers of matches of the associated IUPAC patterns. The new weights of the columns of merged PWM will be the sum of these numbers of matches. In the non-overlapping regions, the probabilities and weights are simply copied over from the one PWM.

1.2 Higher-order inhomogeneous Markov models

BaMMmotif [28] refines the pre-aligned short patterns or position-weight-matrices (PWMs) to higher-order Bayesian Markov models for the enriched motifs.

According to Boltzmann's law, the probability of a genomic site with sequence $\mathbf{x}$ to be bound by the transcription factor divided by the probability of $\mathbf{x}$ not to be bound is

$$\exp\left(-\frac{\Delta G(\mathbf{x})-\mu}{k_B T}\right) = \frac{p(\text{bound}|\mathbf{x})}{p(\text{not bound}|\mathbf{x})} = \frac{p(\text{bound}|\mathbf{x})}{1-p(\text{bound}|\mathbf{x})}, \tag{15}$$

with the chemical potential μ that depends on the factor concentration but not on $\mathbf{x}$. Solving for $p(\text{bound}|\mathbf{x})$ yields the well-known behaviour for saturated binding,

$$p(\text{bound}|\mathbf{x}) = \left(1+\exp\left(\frac{\Delta G(\mathbf{x})-\mu}{k_B T}\right)\right)^{-1}. \tag{16}$$

We parameterise the dependence of $\Delta G(\mathbf{x})$ on the binding site sequence $\mathbf{x}$ by a probability distribution $p_{\text{motif}}(\mathbf{x})$ which is defined by

$$p_{\text{motif}}(\mathbf{x})/p_{\text{bg}}(\mathbf{x}) \propto \exp(\Delta G(\mathbf{x})/k_B T). \tag{17}$$

The proportionality constant is determined by the normalization. Solving for $p_{\text{motif}}(\mathbf{x})$ and normalising yields

$$p_{\text{motif}}(\mathbf{x}) := \frac{p_{\text{bg}}(\mathbf{x})\exp(-\Delta G(\mathbf{x})/k_B T)}{\sum_{\mathbf{y}} p_{\text{bg}}(\mathbf{y})\exp(-\Delta G(\mathbf{y})/k_B T)}, \tag{18}$$

where the sum in the normalisation constant runs over all possible binding site sequences $\mathbf{y} \in \{\text{A},\text{C},\text{G},\text{T}\}^W$.

We derive a model for the Gibbs binding energy $\Delta G(\mathbf{x})$ for any potential binding site sequence $\mathbf{x} = x_{1:K} \in \{\text{A},\text{C},\text{G},\text{T}\}^K$ by computing a motif score $S(\mathbf{x})$:

$$S(\mathbf{x}) = -\frac{\Delta G(\mathbf{x})}{k_B T} + \text{const.} := \log\frac{p_{\text{motif}}(\mathbf{x})}{p_{\text{bg}}(\mathbf{x})} = \sum_{j=0}^{K-1}\log\frac{p_j^K(x_j|x_{j-K:j-1})}{p_{\text{bg}}^{K'}(x_j|x_{j-K':j-1})}. \tag{19}$$

where we model the background probability using a homogeneous Markov model of order K':

$$p_{\text{bg}}(x_{i_0:i_1}) = \prod_{i=i_0}^{i_1} p_{\text{bg}}(x_i|x_{i-K':\,i-1}). \tag{20}$$

We model the motif using an inhomogeneous Markov model of order K:

$$p_{\text{motif}}(x_{0:K-1}) = \prod_{j=0}^{K-1} p_j(x_j|x_{j-K:\,j-1}). \tag{21}$$

We learn the parameters of the inhomogeneous Markov model by maximising the posterior probability. A natural prior is a product of Dirichlet distributions with pseudo-count parameters proportional to the lower-order model probabilities, with proportionality constants α_{kj} for $k = 1, \ldots, K$, whose size determines the strength of the prior. Maximizing the posterior probability yields

$$p_j^k(x_{k+1}|x_{1:k}) = \frac{n_j(x_{1:k+1}|\mathbf{r}) + \alpha_{kj} p_j^{k-1}(x_{k+1}|x_{2:k})}{n_{j-1}(x_{1:k}|\mathbf{r}) + \alpha_{kj}}. \tag{22}$$

1.3 Masking in the motif refinement step

We train BaMMs using the expectation-maximization (EM) algorithm. In the E-step, we (re-) estimate the responsibilities r for a motif to be present at position i of sequence n,

$$r_{ni} := p(z_n = i|\mathbf{x}_n, p^K_{\text{motif}}(\mathbf{x})) = \frac{p(\mathbf{x}_n|z_n = i, p^K_{\text{motif}}(\mathbf{x}))\, p(z_n = i)}{\sum_{i'=0}^{L_n-W+1} p(\mathbf{x}_n|z_n = i', p^K_{\text{motif}}(\mathbf{x}))\, p(z_n = i')} \tag{23}$$

In the M-step, we use the new r_{ni} to update the model parameters $p_{\text{motif}}(\mathbf{x})^K$ for all orders $k = 0, ..., K$. This update equation looks exactly the same as the previous equation for known motifs locations, except that now the counts $n_j(x_{1:k+1})$ are interpreted as fractional counts computed according to

$$n_j(\mathbf{x}, x_{k+1}|\mathbf{r}) := \sum_{n=1}^{N} \sum_{i=1}^{L_n-W+1} r_{ni} I\left(x_{n,\, i+j-k\,:\,i+j} = (\mathbf{x}, x_{k+1})\right). \tag{24}$$

The indicator function I returns 1 if the logical expression is true and 0 otherwise. The parameter updates are done for all orders from 0 to K.

Here we introduce a masking step between the E- and M-step by masking out the first N% of r_{ni} after re-ranking increasingly (N is 90 by default) in the first iteration of the EM. By doing this, we learn the model only on the strong binding sites and thus eliminate the effect of unrelated motifs. We then iterate the EM algorithm until convergence.

1.4 Optimization of order- and position-specific hyperparameters α

In the previous version of BaMMmotif [28], the hyperparameters α_{kj} were empirically chosen. Here in this project, we try to learn the position-specific α_{kj} from the data.

We choose as prior on the hyperparameters α_{kj} (for $1 \leq k \leq K$) an inverse Gamma distribution with parameters 1 and $(\beta\gamma^k)$,

$$p(\alpha_{kj}|\beta, \gamma) = \frac{\beta\,\gamma^k}{\alpha_{kj}^2} e^{-\beta\,\gamma^k/\alpha_{kj}} \tag{25}$$

where $\beta \approx 5$ and $\gamma = 3$ corresponds roughly to the previous choice $\alpha_{kj} = \beta\,\gamma^k = 20 \times 3^{k-1}$ that worked for all of the datasets in the previous study [28].

According to Bayes' theorem, the conditional probability of $\boldsymbol{\alpha}$ given motif positions $\mathbf{z}$ can be written as:

$$
\begin{aligned}
p(\boldsymbol{\alpha}_k|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) &\underset{\boldsymbol{\alpha}_k}{\propto} p(\mathbf{X}|\boldsymbol{\alpha},\mathbf{z},p^{k-1}_{\text{motif}})\, p(\boldsymbol{\alpha}|\mathbf{z},p^{k-1}_{\text{motif}}) \\
p(\boldsymbol{\alpha}_k|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) &\underset{\boldsymbol{\alpha}_k}{\propto} p(\mathbf{X}|\boldsymbol{\alpha},\mathbf{z},p^{k-1}_{\text{motif}})\, p(\boldsymbol{\alpha})
\end{aligned}
\tag{26}
$$

where

$$
\begin{aligned}
&p(\mathbf{X}|\mathbf{z},\boldsymbol{\alpha},p^{k-1}_{\text{motif}}) \\
&\propto \prod_{j=0}^{W-1}\prod_{\mathbf{y}} \frac{\Gamma(\alpha_{kj})}{\prod_a \Gamma(\alpha_{kj} v^*_j(a|\mathbf{y}'))} \frac{\prod_{a=1}^4 \Gamma(n^{\mathbf{z}}_j(\mathbf{y},a)+\alpha_{kj} v^*_j(a|\mathbf{y}'))}{\Gamma(n^{\mathbf{z}}_{j-1}(\mathbf{y})+\alpha_{kj})} \cancel{\prod_{a=1}^4 \frac{1}{v_{\text{bg}}(a|\mathbf{y})^{n^{\mathbf{z}}_j(\mathbf{y},a)}}}.
\end{aligned}
\tag{27}
$$

Inserting (25) and (27) yields for the conditional probability

$$
\begin{aligned}
p(\boldsymbol{\alpha}_k|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) &= \sum_{j=0}^{W-1}\left(\prod_{\mathbf{y}} \frac{\beta\,\gamma^k}{\alpha^2_{kj}} e^{-\frac{\beta\,\gamma^k}{\alpha_{kj}}} \frac{\Gamma(\alpha_{kj})}{\prod_a \Gamma(\alpha_{kj} v^*_j(a|\mathbf{y}'))} \frac{\prod_{a=1}^4 \Gamma(n^{\mathbf{z}}_j(\mathbf{y},a)+\alpha_{kj} v^*_j(a|\mathbf{y}'))}{\Gamma(n^{\mathbf{z}}_{j-1}(\mathbf{y})+\alpha_{kj})}\right) \\
&= \prod_{j=0}^{W-1} p(\alpha_{kj}|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}})\,,
\end{aligned}
\tag{28}
$$

which factorizes over the α_{kj}. We could therefore use Gibbs sampling to draw each new value of α_{kj} from its probability distribution independent of the others.

But for an efficient optimisation we need to reparameterise α_{kj} as

$$
\alpha_{kj} = e^{a_{kj}} \tag{29}
$$

and sample a_{kj} instead of α_{kj}, because otherwise it would take too long to explore the entire probability distribution by small steps in α_{kj}. If we went in steps of 0.5, for example, it would take almost 20000 directed steps to move from $\alpha_{kj} = 1$ to 10000. With steps of size 0.5, it would take only $2log20000 = 18.4$ directed steps to reach 10000. The probability density also needs to be transformed with the variable:

$$
\begin{aligned}
p(a_{kj}|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) &= \left|\frac{d\,\alpha_{kj}}{d\,a_{kj}}\right| p(\alpha_{kj}|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) && (30)\\
&= \alpha_{kj}\, p(\alpha_{kj}|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) && (31)
\end{aligned}
$$

The log conditional probability for a_{kl} is

$$
\begin{aligned}
&\log p(a_{kl}|\mathbf{X},\mathbf{z},p^{k-1}_{\text{motif}}) = \text{const.} - \log\alpha_{kj} - \beta\,\gamma^k/\alpha_{kj} + 4^k \log\Gamma(\alpha_{kj}) \qquad (32)\\
&+ \sum_{\mathbf{y}=y_{1:k}} \left(\sum_{a=1}^4 \left[\log\Gamma(n^{\mathbf{z}}_j(\mathbf{y},a)+\alpha_{kj} p^{k-1}_{\text{motif},j}(a|\mathbf{y}')) - \log\Gamma(\alpha_{kj} v^{k-1}_j(a|\mathbf{y}'))\right] - \log\Gamma(n^{\mathbf{z}}_{j-1}(\mathbf{y})+\alpha_{kj})\right)
\end{aligned}
$$

We can sample from this distribution using the Metropolis-Hastings algorithm. We draw a new $a_{kl}^{\text{try}} \sim \mathcal{N}(a_{kl}, 1)$ and accept this trial sample with a probability

$$\frac{p(a_{kl}^{\text{try}}|\mathbf{X}, \mathbf{z}, p_{\text{motif}}^{k-1})}{p(a_{kl}|\mathbf{X}, \mathbf{z}, p_{\text{motif}}^{k-1})} \text{ if } p(a_{kl}^{\text{try}}|\mathbf{X}, \mathbf{z}, p_{\text{motif}}^{k-1}) < p(a_{kl}|\mathbf{X}, \mathbf{z}, p_{\text{motif}}^{k-1})$$
$$1 \text{ if otherwise .} \tag{33}$$

Because it is fast to sample a_{kl} in this way, we draw 10 or times in a row and only take record the last accepted sample of a_{kl}. This 10-fold repetition ensures that we can explore almost the entire range of relevant values of a_{kl} within these 10 steps.

At the start of the sampling, the a_{kj} will move in the direction of the medians of their probability distribution in relatively directed steps until the changes to the a_{kj} become non-directional and begin to fluctuate. We can then fix the a_{kj} to the average of the last 20 or so samples and perform a few (e.g. 5) iterations of the EM algorithm (described in section 1.2) to find the optimum model parameters $v_j^K(a|\mathbf{y})$ given the fixed a_{kj}.

1.5 Learning positional preferences of motifs

Thermodynamic treatment of positional preference

We proceed analogously to section 1.2 but introduce a positional preference as an additive term ΔG_i in the binding energy. The probability of a factor to bind a binding site consisting of W nucleotides between i and $i + W - 1$ in a sequence $\mathbf{x} = x_{1:L}$ then becomes

$$p(\text{factor bound at position } i|\mathbf{x}) = \left(1 + \exp\left(\frac{\Delta G(x_{i:i+W-1}) + \Delta G_i - \mu}{k_B T}\right)\right)^{-1} . \tag{34}$$

We define $p_{\text{motif}}(x_{0:W-1})$ as in eq. (18) and we further define a positional distribution

$$p(z{=}i|\text{factor bound to } \mathbf{x}) = \frac{\exp(-\Delta G_i/k_B T)}{\sum_{i'=1}^{L} \exp(-\Delta G_{i'}/k_B T)} . \tag{35}$$

We abbreviate the denominator as const. gives

$$-\frac{\Delta G_i}{k_B T} + \text{const.} = \log p(z{=}i|\text{factor bound to } \mathbf{x}) =: s_i. \tag{36}$$

Once we know $p_{\text{motif}}(\cdot)$ and $p(z{=}i|\text{factor bound to } \mathbf{x})$, we can compute $S(x_{i:i+W-1})$ and s_i and the relative binding strength $(\Delta G(x_{i:i+W-1}) + \Delta G_i)/k_B T$ for any potential binding site position i in any sequence $\mathbf{x} = (x_1 \dots x_L)$.

If we again assume to be in a regime of unsaturated binding, $p(\text{bound}|\mathbf{x}) \lesssim 0.1$ we can approximate the probability $p(\mathbf{x}_n|\text{bound}, p_{\text{motif}}^k)$ for pulling out a sequence $\mathbf{x}_n$ from an underlying

distribution of possible sequences $p_{\text{bg}}(\mathbf{x})$ as

$$
\begin{aligned}
p(\mathbf{x}_n|\text{bound}, p^k_{\text{motif}}) &\propto p(\text{factor bound}|\mathbf{x}_n, p^k_{\text{motif}})\, p_{\text{bg}}(\mathbf{x}_n) \\
&= p_{\text{bg}}(\mathbf{x}_n) \sum_{i=1}^{L-W+1} p(\text{factor bound at } i|\mathbf{x}_n, p^k_{\text{motif}}) \\
&= p_{\text{bg}}(\mathbf{x}_n) \sum_{i=1}^{L-W+1} \left(1 + \exp\left(\frac{\Delta G(x_{i:i+W-1}) + \Delta G_i - \mu}{k_B T}\right)\right)^{-1} \\
&\approx p_{\text{bg}}(\mathbf{x}_n) \sum_{i=1}^{L-W+1} \exp\left(-\frac{\Delta G(x_{i:i+W-1}) + \Delta G_i - \mu}{k_B T}\right) \\
&\propto p_{\text{bg}}(\mathbf{x}_n) \sum_{i=1}^{L-W+1} \exp\left(S(x_{i:i+W-1}) + s_i\right). \qquad (37)
\end{aligned}
$$

To find the model parameters $\boldsymbol{\theta}$ consisting of $\mathbf{s} = (s_1, \ldots, s_{L-W+1})$ and of p^k_{motif} specifying $p_{\text{motif}}(\cdot)$, we need to optimise the log likelihood function of these parameters:

$$
LL(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(\mathbf{x}_n|\text{bound}, p^k_{\text{motif}}, \mathbf{s}) \qquad (38)
$$

Flat Bayesian prior on positional preference

Let us define parameters $\boldsymbol{\pi}$ with $\pi_i = p(z{=}i|z_i{\neq}0) = e^{s_i}$ the probability of a motif to start at position i of a sequence. The M-step will then be given again by equation (23) but this time using the positional preferences π_i instead of the flat positional distribution. We will use a flat prior distribution,

$$
p(\boldsymbol{\pi}|\beta) = \text{Dir}(\boldsymbol{\pi}|\beta\mathbb{1}), \qquad (39)
$$

and we will choose a value around $\beta = 2 \ldots 10$.

The auxiliary function becomes

$$
\begin{aligned}
&Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q|\mathbf{r}, p^{k-1}_{\text{motif}}) \\
&= \sum_{n=1}^{N} \left[\sum_{i=0}^{L_n-W+1} r_{ni} \log \left(p(\mathbf{x}_n|z_n = i, p^k_{\text{motif}})\, p(z_n = i|q) \right) \right] + \log p(p^k_{\text{motif}}|p^{k-1}_{\text{motif}}, \boldsymbol{\alpha}) + \log p(\boldsymbol{\pi}|\beta) \\
&= \sum_{n=1}^{N} \sum_{i=0}^{L_n-W+1} r_{ni} \log p(\mathbf{x}_n|z_n = i, p^k_{\text{motif}}) + \log p(p^k_{\text{motif}}|p^{k-1}_{\text{motif}}, \boldsymbol{\alpha}) \\
&\quad + \sum_{n=1}^{N} \left(r_{n,0} \log(1-q) + \sum_{i=1}^{L_n-W+1} r_{ni} \log(q\pi_i) \right) + \log \mathrm{Dir}(\boldsymbol{\pi}|\beta\mathbb{1}) \\
&= \sum_{n=1}^{N} \sum_{i=0}^{L_n-W+1} r_{ni} \log p(\mathbf{x}_n|z_n = i, p^k_{\text{motif}}) + \log p(p^k_{\text{motif}}|p^{k-1}_{\text{motif}}, \boldsymbol{\alpha}) \qquad (40) \\
&\quad + \sum_{n=1}^{N} \left(r_{n,0} \log(1-q) + (1-r_{n,0}) \log q + \sum_{i=1}^{L_n-W+1} r_{ni} \log \pi_i \right) + \sum_{i=1}^{L_n-W+1} (\beta-1) \log \pi_i .
\end{aligned}
$$

We use the method or Lagrange multipliers again to find the optimum of $Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q|\mathbf{r}, p^{k-1}_{\text{motif}})$ under the constraint $\sum_{i=1}^{L-W+1} \pi_i = 1$:

$$
\frac{\partial}{\partial \pi_i} \left(Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q|\mathbf{r}, p^{k-1}_{\text{motif}}) - \lambda \left(\sum_{i=1}^{L-W+1} \pi_i - 1 \right) \right) = \sum_{n=1}^{N} \frac{r_{ni}}{\pi_i} + \frac{\beta-1}{\pi_i} - \lambda = 0 \qquad (41)
$$

Solving for π_i, normalising the distribution and defining $N_i := \sum_{n=1}^{N} r_{ni}$ yields

$$
\boxed{\pi_i = \frac{N_i + \beta - 1}{N + (L-W+1)(\beta-1)} .} \qquad (42)
$$

Prior penalising jumps in the positional preference profile

For many applications it might be more appropriate to limit the complexity of the positional preference profile by imposing a smoothness on the $p(z = i)$. For example, transcription factor binding sites will be more frequent near the center of ChIP-seq peaks than farther away; factors bind more strongly to the outer parts of probes on protein binding microarrays than to the parts near the glass slide; transcription factors in HT-SELEX experiments might prefer the center of probes over the ends. In the following we assume that all training and test sequences have the same length L.

Because the smoothness prior couples neighbouring positional probabilities with each other, there is no closed-form solution for the parameters anymore. We have to use a gradient-based optimisation such as conjugate gradients to minimise Q with respect to the positional parameters. We therefore parameterise the positional distribution in such a way that the normalisation condition $\sum_i \pi_i = 1$ and the limits $0 \leq \pi_i \leq 1$ automatically hold true during the numerical optimisation,

$$
p(z_n = i|z_n \neq 0) = \frac{e^{s_i}}{\sum_{i'=1}^{L-W+1} e^{s_{i'}}} . \qquad (43)
$$

We impose a smoothness prior on the π_i, that encourages the point-wise estimated first derivative to stay small,

$$p(\boldsymbol{\pi}|\beta) = \prod_{i=2}^{L-W+1} \mathcal{N}\left(s_i - s_{i-1}|\, 0, \beta^{-1}\right) , \tag{44}$$

with precision (= inverse variance) β.

With this prior, the auxiliary function becomes

$$\begin{aligned} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}}) = & \sum_{n=1}^{N} \sum_{i=0}^{L-W+1} r_{ni} \log p(\mathbf{x}_n|z_n = i, p^k_{\text{motif}}) + \log p(p^k_{\text{motif}}|p^{k-1}_{\text{motif}}, \boldsymbol{\alpha}) \\ & + \sum_{n=1}^{N} \left(r_{n,0} \log(1-q) + (1-r_{n,0}) \log q + \sum_{i=1}^{L-W+1} r_{ni} \left(s_i - \log \left(\sum_{i'} e^{s_{i'}} \right) \right) \right) \\ & - \frac{\beta}{2} \sum_{i=2}^{L-W+1} (s_i - s_{i-1})^2 + \frac{L-W}{2} \log \beta + \text{const.} \end{aligned} \tag{45}$$

The partial derivatives of $Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}})$ are

$$\begin{aligned} \frac{\partial}{\partial s_i} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}}) = & \sum_{n=1}^{N} r_{ni} - \sum_{n=1}^{N} \sum_{i'=1}^{L-W+1} r_{ni'} \frac{e^{s_i}}{\sum_{i''} e^{s_{i''}}} \\ & - \beta\,(s_i - s_{i-1})\, I(2 \le i \le L-W+1) \\ & + \beta\,(s_{i+1} - s_i)\, I(1 \le i \le L-W) \end{aligned} \tag{46}$$

and

$$\boxed{\frac{\partial}{\partial s_i} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}}) = N_i - (N - N_0)\, p(z{=}i|z{\neq}0) - (\beta \mathbf{A}\boldsymbol{s})_i}$$

with the abbreviations $N_0 := \sum_{n=1}^{N} r_{n,0}$ and

$$\mathbf{A} := \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & -1 & 2 & -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & -1 & 2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 2 & -1 & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 & -1 & 2 & -1 \\ 0 & \dots & \dots & \dots & 0 & 0 & -1 & 1 \end{pmatrix} . \tag{47}$$

The partial derivative will adjust s_i such that $p(z{=}i|z{\neq}0) = e^{s_i} / \sum_{i'} e^{s_{i'}}$ equals $N_i/(N - N_0)$ plus a smoothness correction $\mathbf{A}\boldsymbol{s}$ that will pull s_i up or down in order to minimise the estimator of the second derivative of the profile at position i. We run a few iterations of conjugate gradients (e.g. 5 to 10) during each EM step to learn the positional preferences.

Learning the optimal smoothness parameter β from the data. We can regard Q also as a function of β,

$$Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}, \beta | \mathbf{r}, p^{k-1}_{\text{motif}}) = -\frac{\beta}{2} \sum_{i=2}^{L-W+1} (\pi_i - \pi_{i-1})^2 + \frac{L-W}{2} \log \beta + \text{const}_\beta \,, \tag{48}$$

and optimise is with respect to β:

$$0 = \frac{\partial}{\partial \beta} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}, \beta | \mathbf{r}, p^{k-1}_{\text{motif}}) = -\frac{1}{2} \sum_{i=2}^{L-W+1} (s_i - s_{i-1})^2 + \frac{L-W}{2\beta} \tag{49}$$

and therefore

$$\boxed{\beta = \left(\frac{1}{L-W} \sum_{i=2}^{L-W+1} (s_i - s_{i-1})^2 \right)^{-1}} \tag{50}$$

Instead of optimising β, we can again interpret Q as the likelihood of an ensemble of fractional motif instances with weights r_{ni} and compute the expectation value of β. If we assume a uniform prior on β, $p(\beta) = \text{const}$, the posterior distribution of β is proportional to the likelihood. We note that the functional form of $Q(\beta)$ is that of a Gamma distribution, $Q(\beta) = \log \text{Ga}(\beta | a, b) + \text{const} = (a-1) \log \beta - b\beta + \text{const}$, with $a - 1 = (L-W)/2$ and $b = (1/2) \sum_i (s_i - s_{i-1})^2$. Since the expectation value of a Gamma distribution is a/b, we can conclude for β

$$\boxed{\mathbb{E}[\beta] = \left(\frac{1}{L-W+2} \sum_{i=2}^{L-W+1} (s_i - s_{i-1})^2 \right)^{-1} .} \tag{51}$$

We can then update β by its expectation value instead of the mode of $Q(\beta)$. Alternatively, we could sample β from the Gamma distribution $\text{Ga}(\beta | (L-W+2)/2, (1/2) \sum_i (s_i - s_{i-1})^2)$.

Prior penalising kinks in the positional preference profile

For various applications such as PBMs and HT-SELEC, we might be interested in more smooth positional preferences. In these cases, it might be better to use a smoothness prior on the π_i that encourages the point wise estimated *third* derivative to stay small,

$$p(\boldsymbol{\pi} | \beta) = \prod_{i=2}^{L-W} \mathcal{N} \left(s_i - \frac{s_{i-1} + s_{i+1}}{2} \middle| 0, \beta^{-1} \right) , \tag{52}$$

with precision (= inverse variance) β. With this prior, the auxiliary function becomes

$$\begin{aligned} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi} | \mathbf{r}, p^{k-1}_{\text{motif}}) = & \sum_{n=1}^{N} \sum_{i=0}^{L-W+1} r_{ni} \log p(\mathbf{x}_n | z_n = i, p^k_{\text{motif}}) + \log p(p^k_{\text{motif}} | p^{k-1}_{\text{motif}}, \boldsymbol{\alpha}) \\ & + \sum_{n=1}^{N} \left(r_{n,0} \log(1-q) + (1-r_{n,0}) \log q + \sum_{i=1}^{L-W+1} r_{ni} \left(s_i - \log \left(\sum_{i'} e^{s_{i'}} \right) \right) \right) \\ & - \frac{\beta}{2} \sum_{i=2}^{L-W} \left(s_i - \frac{s_{i-1} + s_{i+1}}{2} \right)^2 + \frac{L-W-1}{2} \log \beta + \text{const.} \end{aligned} \tag{53}$$

The partial derivatives of $Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}})$ are

$$\begin{aligned}\frac{\partial}{\partial s_i} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}}) = &\sum_{n=1}^{N} r_{ni} - \sum_{n=1}^{N} \sum_{i'=1}^{L-W+1} r_{ni'} \frac{e^{s_i}}{\sum_{i''} e^{s_{i''}}} \\ &+ \frac{\beta}{2}\left(s_{i-1} - \frac{s_{i-2}+s_i}{2}\right) I(3 \le i \le L-W+1) \\ &- \beta\left(s_i - \frac{s_{i-1}+s_{i+1}}{2}\right) I(2 \le i \le L-W) \\ &+ \frac{\beta}{2}\left(s_{i+1} - \frac{s_i+s_{i+2}}{2}\right) I(1 \le i \le L-W-1) \end{aligned} \tag{54}$$

and

$$\boxed{\frac{\partial}{\partial s_i} Q(p^k_{\text{motif}}, \boldsymbol{\alpha}, q, \boldsymbol{\pi}|\mathbf{r}, p^{k-1}_{\text{motif}}) = N_i - (N - N_0)\, p(z{=}i|z{\neq}0) - \frac{\beta}{4}(\mathbf{B}\boldsymbol{s})_i}$$

with the abbreviations $N_0 := \sum_{n=1}^{N} r_{n,0}$ and

$$\mathbf{B} := \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ -2 & 5 & -4 & 1 & 0 & \ddots & \ddots & \ddots & \vdots \\ 1 & -4 & 6 & -4 & 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & 1 & -4 & 6 & -4 & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 1 & -4 & 6 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 6 & -4 & 1 \\ \vdots & \ddots & \ddots & \ddots & \ddots & 1 & -4 & 5 & -2 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 & -2 & 1 \end{pmatrix}. \tag{55}$$

The partial derivative will adjust s_i such that $p(z{=}i|z{\neq}0) = e^{s_i}/\sum_{i'} e^{s_{i'}}$ equals $N_i/(N-N_0)$ plus a smoothness correction $\mathbf{B}\boldsymbol{s}$ that will pull s_i up or down in order to minimise the estimator of the third derivative of the profile at position i.

Learning the optimal smoothness parameter β from the data. Analogously to the previous smoothness prior, we can learn β from the data using the update

$$\boxed{\beta = \left(\frac{1}{L{-}W-1} \sum_{i=2}^{L-W} \left(s_i - \frac{s_{i-1}+s_{i+1}}{2}\right)^2\right)^{-1}} \tag{56}$$

or

$$\boxed{\beta = \left(\frac{1}{L{-}W+1} \sum_{i=2}^{L-W} \left(s_i - \frac{s_{i-1}+s_{i+1}}{2}\right)^2\right)^{-1}.} \tag{57}$$

1.6 Scanning sequences for motif occurrences

To obtain the motif occurrences from the sequences, given a known or learned motif, we developed a motif scanning tool BaMMScan to evaluate the possible motif occurrences on the input sequences. The motif score $s_i(x_{1:K})$ is calculated for each position i on every sequence x for the order K. A background score distribution is created by generating M-fold background sequences from a second-order homogeneous Markov model from input set (M can be 10). We sort the list of $N^+ + N^-$ positive- and negative-set scores jointly in descending order. We denote the cumulative number of scores from the negative set up to rank l in this list by FP_l and then compute the P-value of entry l with score S_l in that list by

$$P\text{-value}(S_l) = \frac{1}{N^-}\left(\mathrm{FP}_l + \frac{S_l^{\text{higher}} - S_l}{S_l^{\text{higher}} - S_l^{\text{lower}} + \epsilon}\right). \tag{58}$$

and the E-values are obtained simply as

$$E\text{-value} = N^+ \times P\text{-value}. \tag{59}$$

The motif occurrences with a P-value smaller than certain cutoff (e.g. $1e^{-4}$) are reported.

1.7 Evaluation criteria using the average recall (AvRec) score

To assess the predictive performance of the motif finders, we first defined an average recall (AvRec) score (details also described in [32]). The AvRec score represents the averaged recall over the range of precision from 0 to 1. The advantage of AvRec score over commonly used p-value is that it covers the most relevant range of False-discovery-rates (FDR) in practical applications and allows the user to intuitively estimate the motif performance in her particular application.

We obtain a p-value for each sequence by

$$p\text{-value}_l = \frac{\mathrm{FP}_l + 0.5}{\mathrm{N}^- + 1} \tag{60}$$

After having a p-value for every motif occurrence (as described in eq.58), we obtain a list of corresponding local FDR values and an estimate of the weight of the null component η_0 by applying fdrtool [57] on the p-value distribution (Figure S 13A). We then calculate FDR and recall for each entry by

$$\mathrm{FDR}_l = \frac{\mathrm{FP}_l}{\mathrm{FP}_l + \mathrm{TP}_l} \tag{61}$$

$$\mathrm{recall}_l = (1 - \mathrm{FDR}_l)\frac{l}{(1-\eta_0)N} \tag{62}$$

The ratio between true positive (TP) and false positive (FP) is calculated by

$$R_{l[\mathrm{TP/FP}]} = \frac{1 - \mathrm{FDR}_l}{\mathrm{FDR}_l} \times M \tag{63}$$

with M as the ratio between negative and positive sequences.

We visualize the characteristics by plotting the TP/FP ratio $R_{\mathrm{TP/FP}}$ on the y-axis against the recall on the x-axis, and define the calculated area-under-the-curve as the AvRec score for motif evaluation (Figure S 13B).

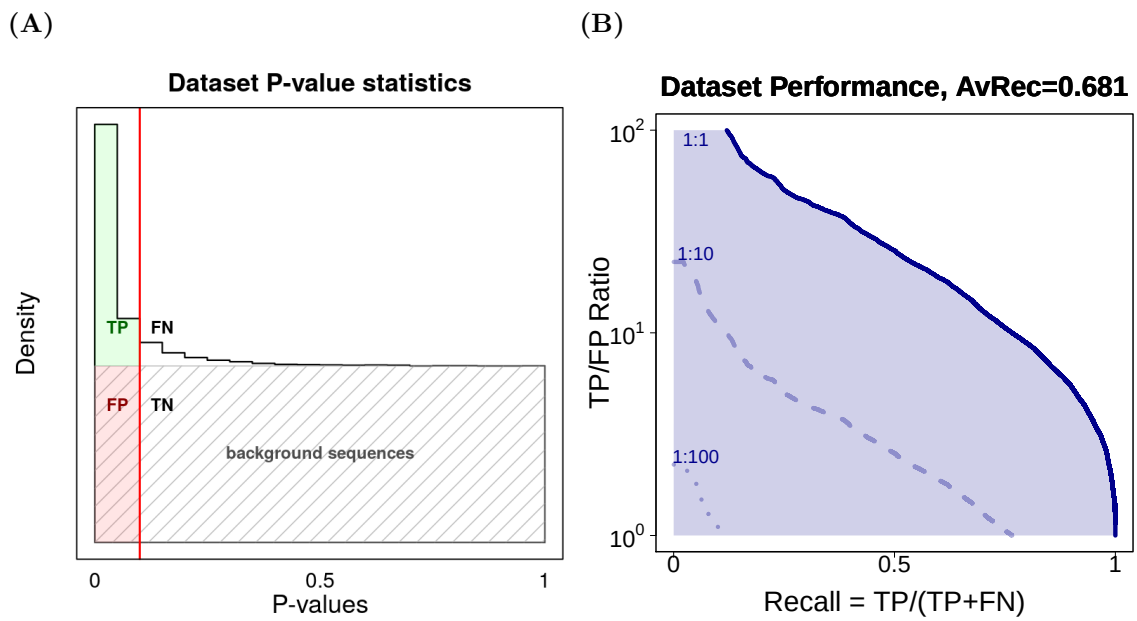

Figure S 13. Schemes of motif assessment on sequences. (**A**) We calculate p-values for the most likely positions on both positive and background sequences, given the motif and a second-order background model learned from the sequences. We plot the density of the p-values and choose a cutoff at 0.1 (the solid red line). The background sequences are mapped in the grey shadow, given the ratio between the background and positive sequences. The true positives (TP, in green), false negatives (FN, in white), false positives (FP, in red) and true negatives (TN) are visualized on the plot. (**B**) For each p-value for positive sequences, we calculate the recall and the ratio between TP and FP, and then plot the recall against the ratio of TP/FP. The solid dark blue line represent for the scenario when the ratio between positive and background sequences is 1:1, the dash lines under it are for the cases when the ratio is 1:10 and 1:100, respectively. An average recall (AvRec) score is calculated as the area under the curve for the 1:1 ratio scenario, and used as a measurement for motif quality on the positive sequence.

2 Datasets used for the benchmark

2.1 ENCODE database

We evaluated the performance of the selected algorithms on human ChIP-seq datasets from the ENCODE portal [36] till March 2012. In total, there are 435 datasets for 93 distinct transcription factors. The top 5000 peak regions, sorted by their signal value, were selected for each dataset. If fewer than 5000 peaks were contained in a dataset, all peaks were chosen. Positive sequences were extracted ±104 bp around the peak summits. Background sequences were sampled by trimer frequencies from positive sequences, with the same length as positive sequences and 10 times the amount of positive sequences. 8 datasets were excluded from all

the results because diChIPMunk failed to learn models within 3 hours.

2.2 HT-SELEX datasets

For HT-SELEX data, we downloaded 164 datasets with 200 bp-long oligomers from Zhu et al. [37], which are deposited in the European Nucleotide Archive (ENA) under accession PRJEB22684. Each dataset represents one non-redundant human transcription factor. For each dataset, we selected the top 5000 sequences from the 4th cycle without any sorting. Background sequences are sampled in the same way as described previously.

2.3 GTRD database

For the GTRD database, we obtained 405 *in vivo* datasets for 405 non-redundant human transcription factors from Yevshin et al. [56]. The top 5000 peak regions are selected after sorting by q-values. Positive sequences are extracted ±100 bp around the peak summits. Background sequences are sampled in the same way as described previously.

2.4 MITOMI datasets

MITOMI is a microfluidics-based approach for *de novo* discovery and quantitative biophysical characterization of DNA target sequences [58]. We downloaded the MITOMI data for 28 *Saccharomyces cerevisiae* transcription factors under the accession GPL10817. The 3 bp and 15 bp long adapters on both ends are truncated. We then downloaded yeast GTRD datasets for 8 transcription factors [56] for the motif discovery.

2.5 Cross-platform datasets

Out of 435 ENCODE datasets for 93 TFs and 164 HT-SELEX datasets for 164 non-redundant TFs, there are 66 TFs which have both *in vivo* and *in vitro* datasets. Out of 66 TFs, most of them have very low AvRec scores when performing the cross-platform validations. We investigated into details and found out that for most of them, the learned motifs were very distinct from the two platforms. This result confirms that TFs can bind to different motifs when experimenting either *in vivo* or *in vitro*. For the left 16 paired tests, they are motifs for 4 TFs, namely CEBPB, POU2F2, ELF1, and FOXA2, which were used in our benchmark.

3 Motif finders used in the benchmark

The source code is available for command-line versions of PEnGmotif and BaMMmotif2 and supported on Linux and MacOS:

3.1 PEnGmotif

PEnGmotif repository: github.com/soedinglab/PEnG-motif. For this study, we used parameters `--optimization_score MUTUAL_INFO -w 8 --threads 4`. The output is in MEME-like format. The motifs are sorted by their AvRec scores, and the best one was taken for the benchmark.

3.2 BaMMmotif2

BaMMmotif2 repository: github.com/soedinglab/BaMMmotif2. For this study, we seeded with the PWMs discovered by PEnGmotif and used parameters `--EM -k [k] --advanceEM --extend 2 2` for further optimization. `[k]` is chosen as 1 and 5 for the benchmark for this study. The output format is defined as BaMM format with extensions like `.ihbcp` and `.hbcp`.

3.3 BaMMmotif

BaMMmotif repository: github.com/soedinglab/BaMMmotif. For this study, we seeded with PWMs by triggering XXmotif internally and used parameters `--reverseComp --XX-localization --XX-localizationRanking --XX-K 2 --maxPValue 0.05 --maxPWMs 3 --extend 2 2` for further optimization. The output format is defined as BaMM format with extensions like `.ihbcp` and `.hbcp`.

3.4 CisFinder

CisFinder was downloaded. We ran patternFind for identifying motifs, patternCluster for clustering motifs, and patternTest for improving motifs. Default parameters were applied. The discovered motifs were converted to MEME-like output format and re-ranked by our motif sorting script, and only the best motif was taken for the benchmarks.

3.5 MEME

MEME version 5.1.1 was installed and applied with parameters `-dna -mod zoops -nmotifs 3 -revcomp -p 4 -V 2`. Maximum 3 motifs were saved in the output, and the best one according to the AvRec score was taken for the benchmarks.

3.6 ChIPMunk

ChIPMunk version v8 was downloaded and applied with parameters `ru.autosome.ChIPMunk 8 12 yes 1.0 100 10 1 4`. The discovered motifs were converted to MEME-like output format.

3.7 diChIPMunk

diChIPMunk was implemented in the same package as ChIPMunk. We ran it with parameters `ru.autosome.di.ChIPMunk 8 12 yes 1.0 200 20 1 4`. The discovered motifs were converted to BaMM-like output format for further comparison.

3.8 InMoDe

InMoDe was downloaded from http://www.jstacs.de/index.php/InMoDe. We applied the module `flexible`, which allows us to customize the learning task. The discovered motifs were converted to BaMM-like output format for further comparison.