

# BayCis: A Bayesian Hierarchical HMM for Cis-regulatory Module Decoding in Metazoan Genomes

Tien-ho Lin<sup>1</sup>, Pradipta Ray<sup>1</sup>, Geir K. Sandve<sup>2</sup>, Selen Uguroglu<sup>3</sup>, Eric P. Xing<sup>1\*</sup>

<sup>1</sup> School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

<sup>2</sup> Dept of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway

<sup>3</sup> Dept of Computer Science and Engineering, Sabanci University, Istanbul, Turkey

## APPENDIX: SUPPLEMENTARY MATERIALS

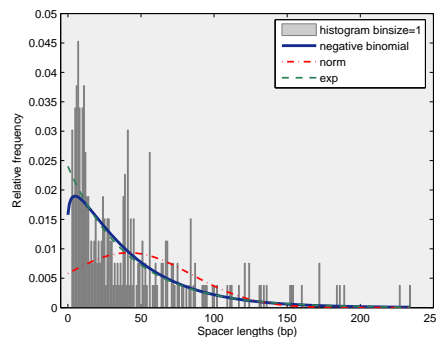
### Keywords

Motif, transcription factor, Bayesian model, *cis*-regulatory module, hierarchical hidden Markov model, generalized hidden Markov model, *Drosophila melanogaster*

## A Details on the BayCis model and algorithm

### A.1 Modeling spacer length distribution via GhHMM

Consider the actual spacer length histogram in *D. melanogaster* in Figure 1. Smoothed distribution fitted by maximum likelihood estimation according to geometric, normal, and negative binomial distribution are also shown. The normal distribution is definitely a very poor approximation. In the tail, the exponential and the negative binomial is not very different but in the shorter region, the negative binomial provides a better fit to the distribution. Furthermore, the peak lies between 5 and 10, not lying between 0 and 5.



**Fig. 1.** The histogram of spacer length distribution with known standard distributions superimposed.

Generalized hidden Markov models (GHMM) have been proposed for the explicit modeling of the state durations in an HMM [10, 3, 7]. A state in a GHMM does not generate one character at a time but instead a region of arbitrary length. The length of the regions is determined according to an explicit duration distribution

The explicit duration models accurately models the state durations at the cost of computation. Alternatively, the negative binomial distributions can be modeled by using instead of one self-transiting state, several externally indistinguishable but internally distinguishable states joined together, as shown in Figure 2. This allows approximation of the GHMM functionality in a HMM [2], where the efficient forward-backward and posterior decoding algorithms can be reused.

\* Correspondence should be addressed to [epxing@cs.cmu.edu](mailto:epxing@cs.cmu.edu). This material is based on work supported by the Pennsylvania Dept of Health's Health Research Program under Grant No. 2001NF-Cancer Health Research Grant ME-01-739, and by an NSF CAREER Award under Grant No. DBI-054659. The authors thank Wenjie Fu for result analysis, Jostein Johansen for help with evaluating CRM predictions, and Ozgur Tastan for investigating the spacer length distributions.

In the GhHMM version of BayCis, we model the cluster background as negative binomial distribution, but leave the global, proximal and distal background as geometric distribution. Unlike the Poisson distribution, the negative binomial distribution can model different mean and variance, allowing a better fit to the empirical distribution shown in Figure 1. This scenario has been used to model exon length distribution by EasyGene to achieve better accuracy [6]. To control computation cost, we approximate the negative binomial distribution by joining several geometrically distributed states. This also makes assigning conjugate priors possible, which will be explained in detail shortly. For the global background, the length distribution has a heavy tail, and in practical usage of BayCis system its length is dependent on how the user cuts the upstream sequence. For the proximal and distal background, the lengths tend to be very short, and the joining of a distal and then a proximal background already provides better expressive power.

## A.2 Details on Flattening hHMM and the modified FB-algorithm

When a hHMM is flattened to a HMM, if there are re-used models in the hHMM, these models must be duplicated, and the hierarchical structure will be lost under unsupervised learning of the parameters [8]. If the hierarchy is a tree, as in BayCis hHMM, the hHMM can be converted to a HMM without losing the hierarchical structure. The HMM state space is exactly the production states in the hHMM, denoted as  $\mathbb{Q} = \{b_g, b_c\} \cup \mathbb{B} \cup (\cup_k \mathbb{M}_k)$ .

Due to the sparsity of our transition probability matrix, as shown in Figure 2, we can further reduce the time complexity of inference for obtaining the probability of a hidden state given the sequence, i.e. the forward-backward algorithm, which is a subroutine in the Bayesian learning algorithm. For notational simplicity, we assume the number of cluster background states is 3. The state space consists of a global background, 3 cluster backgrounds,  $K$  proximal and distal backgrounds, and  $2L_k$  motif states for each motif  $k$  (including sense and anti-sense), so the total size of the state space  $N$  is

$$N = 4 + 2K + 2 \sum_{k=1}^K L_k.$$

Following Rabiner's notation [10], let  $\alpha_t(j)$  be the probability of the partial sequence  $Y_1 \cdots Y_t$  and state  $s_j$  at location  $t$ , or  $\alpha_t(j) = p(Y_1 \cdots Y_t, X_t = s_j)$ . Let  $\beta_t(j)$  be the probability of the partial sequence  $Y_{t+1} \cdots Y_T$  given the state  $s_j$  at location  $t$ , or  $\beta_t(j) = p(Y_{t+1} \cdots Y_T | X_t = s_j)$  (in this section the term  $\beta_t(j)$  is used in backward algorithm for convention, not to be confused with the parameters  $\beta_{g,k}, \beta_{c,k}$ , etc.) The induction step in the forward and backward algorithm are thus

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) A_{ij} \right] B_j(Y_{t+1}), \quad t = 1, 2, \dots, T-1, 1 \leq j \leq N, \quad (1)$$

$$\beta_t(i) = \sum_{j=1}^N A_{ij} B_j(Y_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, 1 \leq j \leq N, \quad (2)$$

It is known that the standard forward and backward algorithm both take  $O(N^2T) = O(K^2 \bar{L}^2 T)$ , where  $\bar{L}$  is the averaged motif length,  $\bar{L} = \frac{1}{K} \sum_{k=1}^K L_k$ . If there are many motifs, the amount of calculations in the forward algorithm may still be large. Our modified forward-backward algorithm further reduces the amount of calculations in the matrix multiplication in (2), based on the fact that "non-trivial" transitions, i.e. transitions whose probability is not 0 nor 1, are restricted to transitions from any of the background states going to either any background state or to the first sense/ last antisense motif position. These transitions correspond to a smaller block of size  $(4 + 2K)$  by  $(4 + 4K)$  in the transition probability matrix, marked as "non-trivial transitions" in Figure 2. With this observation, the modified induction step in the forward algorithm is described here. The vector  $\tilde{\alpha}$  is a holder for temporary values.

1. Let  $\tilde{\mathbb{Q}}_1$  and  $\tilde{\mathbb{Q}}_2$  be the sets of source and target states of the non-trivial transitions, respectively. Formally speaking, if  $0 < A_{ij} < 1$ , we know  $i \in \tilde{\mathbb{Q}}_1$  and  $j \in \tilde{\mathbb{Q}}_2$ , where

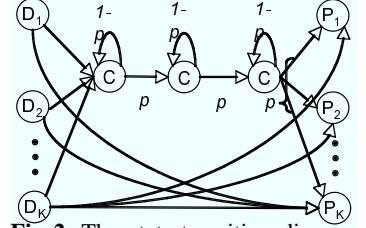


Fig. 2. The state-transition diagram of a gHMM.

$$\begin{aligned}\tilde{\mathbb{Q}}_1 &= \{b_g, b_c, b_p^{(1)}, \dots, b_p^{(K)}, b_d^{(1)}, \dots, b_d^{(K)}\}, \\ \tilde{\mathbb{Q}}_2 &= \tilde{\mathbb{Q}}_1 \cup \{1^{(1)}, 1^{(2)}, \dots, 1^{(K)}, L^{(1')}, L^{(2')}, \dots, L^{(K')}\}\end{aligned}$$

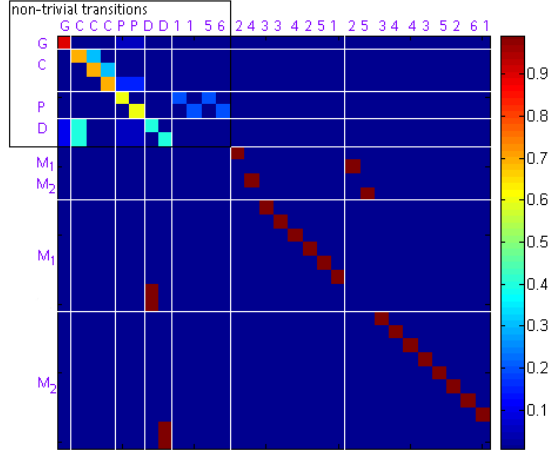
2. Forward induction: for each  $t = 1, 2, \dots, T - 1$ ,

$$\begin{aligned}\tilde{\alpha}(j) &\leftarrow \sum_{i \in \tilde{\mathbb{Q}}_1} \alpha_t(i) A_{ij}, \quad j \in \tilde{\mathbb{Q}}_2, \\ \tilde{\alpha}(l^{(k)}) &\leftarrow \alpha_t((l-1)^{(k)}), \quad 2 \leq l \leq L_k, 1 \leq k \leq K, \\ \tilde{\alpha}(l^{(k')}) &\leftarrow \alpha_t((l+1)^{(k')}), \quad 1 \leq l \leq L_k - 1, 1 \leq k \leq K, \\ \tilde{\alpha}(b_d^k) &\leftarrow \tilde{\alpha}(b_d^k) + \alpha_t(L_k^{(k)}) + \alpha_t(1^{(k')}), \quad 1 \leq k \leq K, \\ \alpha_{t+1}(j) &\leftarrow \tilde{\alpha}(j) B_j(Y_{t+1}), \quad j \in \mathbb{Q}\end{aligned}$$

3. Backward induction: for each  $t = T - 1, T - 2, \dots, 1$ ,

$$\begin{aligned}\beta_t(i) &\leftarrow \sum_{j=1}^N A_{ij} B_j(Y_{t+1}) \beta_{t+1}(j), \quad i \in \tilde{\mathbb{Q}}_1, j \in \tilde{\mathbb{Q}}_2 \\ \beta_t(l^{(k)}) &\leftarrow B_{(l+1)^{(k)}}(Y_{t+1}) \beta_{t+1}((l+1)^{(k)}), \quad 1 \leq l \leq L_k - 1, 1 \leq k \leq K, \\ \beta_t(l^{(k')}) &\leftarrow B_{(l-1)^{(k')}}(Y_{t+1}) \beta_{t+1}((l-1)^{(k')}), \quad 2 \leq l \leq L_k, 1 \leq k \leq K, \\ \beta_t(L_k^{(k)}) &\leftarrow B_{b_d^k}(Y_{t+1}) \beta_{t+1}(b_d^k), \quad 1 \leq k \leq K, \\ \beta_t(1^{(k')}) &\leftarrow B_{b_d^k}(Y_{t+1}) \beta_{t+1}(b_d^k), \quad 1 \leq k \leq K,\end{aligned}$$

The time complexity of the modified forward-backward algorithm is  $O((K^2 + K\bar{L})T)$ . Since the motif length is typically short, we can assume  $\bar{L} < K$  and the time complexity of the modified forward-backward algorithm will be  $O(K^2T)$ , instead of  $O(K^2\bar{L}^2T)$  of the standard forward-backward algorithm.



**Fig.3.** The transition probability matrix of the flattened HMM, shown as a heat map. G, C, P, D, and the numbers correspond to global, cluster, proximal and distal background, and the motif states. The motif states are ordered as:  $1^{(1)}, 1^{(2)}, \dots, 1^{(K)}, L_1^{(1')}, L_2^{(2')}, \dots, L_K^{(K')}, 2^{(1)}, (L_1 - 1)^{(1')}, 3^{(1)}, (L_1 - 2)^{(1')}, \dots, L_1^{(1)}, 1^{(1')}, \dots, 2^{(K)}, (L_K - 1)^{(K')}, 3^{(K)}, (L_K - 2)^{(K')}, \dots, L_K^{(K)}, 1^{(K')}$ .

### A.3 Posterior decoding of DNA binding sites

We can read off the functional annotation (or segmentation) of the input sequences from the posterior probability distribution of the functional states at each position of the sequences according to a *maximal a posteriori* (MAP) scheme. In this scheme, the predicted functional state  $X_t^*$  of position  $t$  is:  $X_t^* = \arg \max_{s \in \mathbb{S}} p(X_t = s|Y)$ , where  $S$  is the set of functional states (motifs and different kinds of background) and  $Y$  is the observed (genomic) sequence.

Note that by using such a posterior decoding scheme (rather than a Viterbi), we integrate the contributions of all possible functional-state-paths for the input sequence (rather than a single “most probable” path), into the posterior probability of each position. Therefore, although in the HMM architecture we do not explicitly model overlapping motifs, our inference procedure does take into account possible contributions of DNA binding sites interacts with competing TFs.

### A.4 Bayesian inference and learning

Under the Bayesian framework described in the main paper, the parameters in the HMM are treated as continuous random variables (collectively referred as  $\Xi$ ) with a prior distribution. Now to compute the posterior probability of functional states, we need to marginalize out these parameter variables:

$$p(X_t|Y) = \int p(X_t = s|Y, \Xi)p(\Xi|Y)d\Xi \quad (3)$$

This computation is intractable in closed form. One approach to obtain an approximate solution is to use Markov chain Monte Carlo methods (e.g., a Gibbs sampling scheme). Here we use a more efficient, deterministic approximation scheme based on *Generalized Mean Field* inference [12], also referred to as *variational Bayesian learning* [5] in the special scenario applied to our problem setting. Omitting theoretical and technical details, our algorithm can be understood as replacing the single-round posterior decoding with an iterative procedure consisting of the following two step:

- Compute the expected counts for all state-transition events (formally called sufficient statistics) using the forward-background algorithm, using **current** values of the HMM parameters.
- Compute the Bayesian estimation (to be detailed shortly) of the HMM parameters based on its prior distribution and the expected sufficient statistics from last step. **Update** the HMM parameters with these estimations.

This procedure is different from the standard EM algorithm which alternates between inference about the hidden variables (the E step) and maximal likelihood estimation of the model parameters (the M step). In our algorithm, the “M” step is a Bayesian estimation step, in which we compute the posterior expectation of the HMM parameters.

Now we outline the formulas for Bayesian estimation of the HMM parameters. Note that since the state-transition probability distributions (which are multinomial) and the prior distributions (which are either beta or gamma) of the transitioning parameters are conjugate-exponential [1]<sup>4</sup>, we have to compute the Bayesian estimation of the logarithm of the transitioning parameters (referred to as the *natural parameterizations*) rather than of the parameters themselves. For example, for the state-transitioning parameter  $\beta_{g,g}$ , we have:

$$E[\ln(\beta_{g,g})] = \int_{\beta_{g,g}} \ln \beta_{g,g} p(\beta_{g,g} | \xi_{g,1}, \xi_{g,2}, E[n_{g,g}]) d\beta_{g,g} \\ = \Psi(\xi_{g,1} + E[n_{g,g}]) - \Psi\left(\sum_j \xi_{g,j} + \sum_{k \in \mathbb{B}_p} E[n_{g,k}]\right), \quad (4)$$

<sup>4</sup> Strictly speaking, this claim is only partially true. Because the conjugacy only applies to the transition probability between a pair of states, but not to the total transition probability mass from a state of interest to all motif-buffer states,  $\sum_{k \in \mathbb{B}_p} \beta_{[.,k]}$ , which is treated as a single “motif-buffer-going” probability in our beta or gamma prior models. (Defining priors for each individual  $\beta_{[.,k]}$ ,  $k \in \mathbb{B}_p$  would require too many hyper-parameters.) As a heuristic surrogate, in certain computational step, we split the *prior mass* (total pseudocounts) corresponding to the total “motif-buffer-going” probability equally among all individual “motif-buffer-going” probabilities as if each has its own pseudocounts, and install strict conjugacy. Since each prior distribution involves at most one such “motif-buffer-going” probability, and that the state-transition probabilities are multinomial parameters subject to a normalization constrain, we only need to use the installed conjugate-exponential property for Bayesian parameter estimation for each “non-motif-going” transition probability, and then obtain the Bayesian estimation of the total “motif-buffer-going” probability indirectly, by subtracting all newly estimated “non-motif-going” transition probabilities from 1.

where  $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x} \frac{\Gamma'(x)}{\Gamma(x)}$  is the digamma function;  $E[\cdot]$  denotes the expectation with respect to the posterior distribution of the argument; and  $n_{g,g}$  refers to the sufficient statistic of parameter  $\beta_{g,g}$  (i.e., counts of transitioning event  $g \rightarrow g$ ). The Bayesian estimate of the original parameter is simply  $\beta_{g,g}^* = \exp(E[\ln(\beta_{g,g})])$ . (In fact we will keep using the natural parameterization in the actual forward-background inference algorithm to avoid numerical underflow caused by long products of probability terms.)

The total ‘‘motif-buffer-going’’ probability is estimated as described in footnote 4, e.g.,  $\beta_{g,g}^* = \sum_{k \in \mathbb{B}_p} \beta_{g,k}^* = 1 - \beta_{g,g}^*$ . To estimate each individual ‘‘motif-buffer-going’’ probability, we use the standard Baum-Welch update based on expected sufficient statistics computed from the matrix of co-occurrence probabilities  $p(X_t, X_{t+1}|Y)$ , scaled by the Bayesian estimation of the total ‘‘motif-buffer-going’’ probability, for example:

$$\beta_{g,i} = \beta_{g,g}^* \frac{\sum_t p(X_t = g, X_{t+1} = i|Y)}{\sum_{t,k} p(X_t = g, X_{t+1} = k|Y)} \quad (5)$$

The initial state probability of the the *BayCis* HMM is not important for CRM prediction as it only directly determine the functional state of the first position of the input sequences and its influence diminishes quickly along the sequence. We simply fix the initial state to be a global background with probability 1.

## A.5 Bayesian learning of the GHMM parameters

The Bayesian estimation of the GHMM parameters is similar to the estimation of the HMM parameters, with some modifications. Note that although we use HMM state space to simulate a negative binomial duration distribution, the self-transition probability of all the cluster background state must remain the same. Otherwise, the duration distribution will no longer be negative binomial. Hence the averaged number of self-transitions and transitions to the next state is used.

Let  $c^j$  denotes the  $j$ -th cluster background states,  $n_{c^j, c^j}$  denotes the number of self transition on state  $c^j$ ,  $n_{c^j, c^{j+1}}$  denotes the number of transition from state  $c^j$  to  $c^{j+1}$ . Let  $E[n_{c,c}]$  denotes the average of expected number of self-transitions from every cluster background states, and  $E[n_{c,c1}]$  denotes the average of expected number of transitions out of every cluster background states, defined as:

$$E[n_{c,c}] = \frac{1}{\xi_{cr}} \sum_{j=1}^{\xi_{cr}} E[n_{c^j, c^j}], \quad (6)$$

$$E[n_{c,c1}] = \frac{1}{\xi_{cr}} \left( \sum_{j=1}^{\xi_{cr}-1} E[n_{c^j, c^{j+1}}] + \sum_{k \in \mathbb{B}_p} E[n_{c^{\xi_{cr}}, k}] \right) \quad (7)$$

Bayesian estimation of the expected value of (log) self-transition probability, with respect to the posterior distribution, would be

$$E[\ln(\beta_{c^j, c^j})] \Psi(\xi_{c,1} + E[n_{c,c}]) - \Psi(\xi_{c,1} + \xi_{c,2} + E[n_{c,c}] + E[n_{c,c1}]) \quad 1 \leq j \leq \xi_{cr}. \quad (8)$$

As in other parameters, the natural parameterization  $\ln(\beta_{c^j, c^j})$  is used, but when the Bayesian estimation of the original parameter is preferred, we use  $\beta_{c^j, c^j}^* = \exp(E[\ln(\beta_{c^j, c^j})])$ .

## B Additional details on experiments

### B.1 The *Drosophila* TRS dataset

We tested our model on a selective dataset consisting of transcriptional regulatory regions regulating the *Drosophila melanogaster* developmental genes. Each TRS in the dataset consists of the CRMs pertinent to a particular gene, any intra-CRM background inbetween, with flanking regions on either side of the extremally located CRMs such that the entire sequence is at least 10K bp long, and the boundaries of the dataset are at least 2K bp from the extremal

CRMs. We included the exonic regions of the genes only when they fell in the aforementioned selected region, and not otherwise.

Selection of the datasets was based on the REDfly CRM database and the Drosophila Cis-regulatory Database at the National University of Singapore [4, 9]. We initially chose 89 CRMs pertaining to 34 early developmental genes. This selection was based on a filtering of CRMs, through which we only chose CRMs which were at least 200 bp long, and contained at least 5 motif instances (2 CRMs with a borderline count of 4 motif instances were also included).

All motif instances used were based on biological curation, and motif instances of the same type in the database often correspond to varying lengths of nucleotide sequences. This is at odds with most computational models of the motifs, which assume a fixed length of the motif in terms of nucleotides. We overcome this issue by searching a 10 bp neighborhood of the annotated location for a fixed width nucleotide sequence which has a high log odds probability of being a motif over background (based on the PWM counts of the motif). Since both our motif algorithm and most competing motif search algorithms assume a PWM based model of the motif, this curation provides more accurate annotation data without placing any competing algorithm at a disadvantage. A short summary of our input sequences is provided in Table 1.

<i>Gene/Length</i>	<i>CRM/Length</i>	<i>Motif</i>	<i>Gene/Length</i>	<i>CRM(Length)</i>	<i>Motif</i>
l.28 (10072)	l.28.DRE / 664	DEAF1 / 8 DFD / 4	lbd-a (10045)	lbd-A)ab-2(1.7) / 1745	EVE / 4 KR / 1 GT / 1 HB / 5
alphaTub84B (10055)	alphaTub84B_alpha1-tubulin_promoter / 855	TRL / 5	tp (10050)	tp_ApME680 / 680	ANTP / 5
bap (10000)	bap_baplac4.5 / 4957	MAD / 4	betatub60D (10181)	betaTub60D_beta3-14/vm1 / 524	BAP / 1 UBX / 2
ct (10068)	ct_wingmargin_enhancer / 2692 wingmargin_Guss / 668	SD / 7	hfd (11658)	Dfd_EAE / 2658 Dfd_EAE-D / 833 Dfd_EAE-F9 / 329 EAE-F2 / 392	DEAF1 / 2 DFD / 13 EXD / 1
dpp (30199)	dpp_dpp813 / 812 dpp_dpp261 / 256 dpp_dpp419 / 419 dpp_intron2 / 1983 dpp_dLmel / 539 dpp_BS1.0 / 8801 dpp_BS1.1 / 1738	ABD-A / 9 BIN / 3 DL / 14 EN / 5 EXD / 5 GRH / 1 UBX / 13	en (11004)	en_stripe_enhancer_intron_1 / 900 en_intron / 720 en_upstream_enhancer / 2401	EN / 6 EVE / 3 FTZ / 12 FTZ-F1 / 2 HB / 2 KR / 1 ZEN / 3
ems (10304)	ems_elementV / 304 ems_ARFE / 1244	ABD-B / 7 TLL / 2 BCD / 2 EMS / 3	twi (10415)	twi_dLmel / 1415	DL / 7
ftz (10487)	ftz_upstream_enhancer / 2562 ftz_proxA / 580 ftz_Prox-323 / 324 ftz_neurogenic_enhancer / 2250 ftz_rebrn2_element / 745	CAD / 2 FTZ / 21 FTZ-F1 / 1 GRH / 4 TTK / 4 HR39 / 1 SLP1 / 1	salm (10144)	salm_salE/Pv / 1078 salm_wingpouch_Guss / 328 salm_blastoderm_early_enhancer / 512 salm_sal242S/P / 242 salm_sal272P/P / 276	BCD / 7 CAD / 4 HB / 1 HKB / 2 SD / 2 KR / 3 UBX / 5
h (10867)	h_stripe_3+4_ET22 / 1745 h_h7_element / 932 h_stripe_6+2 / 1081 h_stripe_6 / 547	BCD / 10 HB / 29 KNI / 22 KR / 13 TLL / 7	hb (12055)	hb_D.7 / 730 hb_anterior_activator / 245 hb_HZ1.4 / 1421 hb_upstream_enhancer / 1424 hb_HZ526 / 528	BCD / 8 HB / 1 TLL / 9
kni (15498)	kni_KD / 870 kni_L2.enhancer / 1360	BCD / 2 CAD / 1 GT / 2 TLL / 6 HB / 8 KR / 4 HIS2B / 5 SD / 5	Kr (11348)	Kr_CDI / 1159 Kr_SiB0.1.2HZ / 1130 Kr_SiH0.6HZ / 540 Kr_H1 / 950 Kr_KrF / 1587	BCD / 4 GT / 1 HB / 6 KNI / 1 TRL / 7 TLL / 7
otp (10000)	otp_C / 441	BYN / 4	tho (10589)	tho_NEE-600 / 590 tho_NEE-300 / 328 tho_NEE / 299	DL / 4 SNA / 4 TWI / 2
gsb (10916)	gsb_fragIV / 516	EVE / 3 FTZ / 3 PRD / 7	ser (10000)	Ser_minimal_wing_enhancer / 812	AP / 14 SUH / 2 PAN / 9
scr (13258)	Scr_5.HH / 5653 Scr_3.OXX / 2953 Scr_6.5KS / 6985	CAD / 2 SLP1 / 1 FTZ / 21 GRH / 4 FTZ-F1 / 1 HR39 / 1 ITK / 4	sh (11144)	sh_enhancer / 2144 sh_del-1-5 / 463 sh_220bp / 221	ABD-A / 4 ANTP / 4 FTZ / 4 UBX / 4
slp1 (10000)	slp1_5-2 / 1554	PAN / 9	sna (10013)	sna_2.8kb / 2913 sna_VA / 612	DL / 10 TWI / 2
so (10012)	so_so10 / 428 so_so7 / 1612	EY / 3 TOY / 5	lll (10063)	lll_P2 / 2764 lll_P3 / 1725	BCD / 8 TRL / 1 GRH / 1 TTK / 1
tin (10000)	tin_tinD / 350	MAD / 7 MED / 3 TIN / 2	kim (10065)	kim_mesectoderm / 631	SNA / 3 TWI / 2
eve (14256)	eve_stripe_3+7 / 511 eve_stripe_2 / 484 eve_MHE / 312 eve_EME-B / 395 eve_EME-B5 / 233 eve_eme2 / 300 eve_EME-B3 / 262	BCD / 5 GT / 3 HB / 12 KNI / 5 KR / 10 MED / 5 TIN / 4 PAN / 6 ZFH1 / 1	ubx (78414)	Ubx_bx1 / 1705 Ubx_BRE / 502 Ubx_basal_promoter / 1189 Ubx_PRE_polycomb_response_element / 1556 Ubx_PBX_enhancer / 1378 Ubx_pbxPB / 297 Ubx_pbxSB / 623 Ubx_pbxAS / 584	EN / 5 EVE / 2 ZEN / 2 FTZ / 10 TLL / 5 GRH / 1 TRL / 17 HB / 27 KNI / 3 TWI / 6 KR / 1 UBX / 2 PHO / 5 Z / 20
vg (12096)	vg_boundary_enhancer / 754 vg_minimal_boundary_enhancer / 360 vg_quadrant_enhancer / 798	MAD / 2 NUB / 4 SUH / 1 SD / 4 VVL / 1	w (11737)	w_Bmdel-W / 6628 w_HPst-W / 7737 w_H-del-BgRVdel-W / 770	Z / 11
zen (10662)	zen_D.7 / 726 zen_L.4 / 1513 zen_dorsal_ectoderm / 624	BRK / 6 DL / 3 GRH / 1 MAD / 10			

**Table 1.** Summary of the Drosophila TRS dataset used for in performance comparison.

This database is available online at <http://www.sailing.cs.cmu.edu/BayCis>. Each TRS is graphically depicted with color coded CRM and motif regions, and is extensively hyperlinked so that the corresponding sequences

may be obtained by clicking on a relevant gene dataset or CRM. A snapshot of the front page of the online database is shown in Fig.3 in the main paper.

## B.2 Hyperparameter selection scheme

Choosing hyperparameters for transition probabilities can be a difficult problem and has significant impact on the performance of the model. As discussed in the Methods section, the hyperparameters of the BayCis model reflect prior beliefs about the architectural features of the CRM structure, such as rough spans of the inter- or intra-module background and distances between motif instances.

A standard way of specifying hyperparameters would be to see which parameter settings work best for datasets with known TFBS, and apply the same on all datasets on which TFBS discovery is to be performed. This is somewhat similar to the supervised learning setup of “training” and “test” sets. The basic assumption here is that in CRMs regulating genes of similar functionality, the CRM architecture would be somewhat similar causing the same set of hyperparameters to work well. More formally, the hyperparameters can be also estimated in the maximal likelihood fashion based on the empirical Bayes principle. We chose to use a representative dataset based on the CRMs of the *even-skipped* gene to choose our hyperparameters for the hHMM and GhHMM.

Based on our observations, the most important hyperparameters governing precision and recall are those regulating transition probabilities into and out of the CRM background state(s). The CRM background state(s) and motif specific states are the only states from where one can enter the motif specific states of the HMM. Hence, hyperparameters which cause the HMM to stay in the CRM background states more frequently than usual risk a low precision, high recall performance while hyperparameters which cause the CRM background states to be rarely visited risk a high precision, low recall scenario. Accurate prediction of CRMs cause the HMM to obtain acceptable values of precision and recall.

We specify the hyperparameters as follows: for the global background,  $\omega_g = 0.002$ ; for the inter-module background,  $\omega_c = 0.05$ ; for the proximal motif buffer,  $\omega_p = 0.25$ ; for the distal buffer hyperparameters,  $\omega_{d,1} = 0.125$  (distal to global background)  $\omega_{d,2} = 0.125$  (distal to clustal background) and  $\omega_{d,3} = 0.25$  (distal to proximal buffer), and the strength of the hyperparameters are set to 1/10 of the expected counts of the transitions on a 15 kbp dataset with the exception of  $\omega_g$  which is set to 10,000. The background probability of the nucleotide at each position was computed locally using a 2nd-order Markov model from a sliding window of 1100 bp centered at the corresponding position. For the GhHMM, based on visual inspection of spacer length distributions between motifs, we choose the parameter as  $r = 2$ .

## B.3 More on F1 and CC scores

The nucleotide-based prediction error is used in the Nature Biotechnology benchmark paper by Tompa et al. [11]. The formulas for the F1 and CC scores are as follows:

$$CC = \frac{nTP \times nTN - nFN \times nFP}{\sqrt{(nTP + nFN)(nTN + nFP)(nTP + nFP)(nTN + nFN)}}, \quad (9)$$

$$F1 = \frac{2 \times Pr \times Re}{Pr + Re}, \quad (10)$$

where  $Pr = \frac{nTP}{nTP+nFP}$  (Precision) and  $Re = \frac{nTP}{nTP+nFN}$  (Recall).

Both CC and F1 are calculated from the number of nucleotides (single positions) that are correctly/wrongly predicted as positives/negatives. The value range of CC is in principle between -1 and +1 (as it is a correlation), but in practice it would lie between 0 (random predictions) and 1 (perfect predictions). As F-1 measure is also a value between 0 and 1, we use the same numerical units in the plot.

## References

1. M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems 13*, 2001.
2. R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1988.

3. J. D. Ferguson. Variable duration models for speech. *Proc. of the Symposium on the Application of HMM to Text and Speech*, pages 143–179, 1980.
4. S. Gallo, L. Li, Z. Hu, and M. Halfon. Redfly:a regulatory element database for drosophila. *Bioinf*, 22(3):381–383, 2006.
5. Z. Ghahramani and M. Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems 13*, 2001.
6. T. S. Larsen and A. Krogh. Easygene—a prokaryotic gene finder that ranks orfs by statistical significance. *BMC Bioinformatics*, 4:21, Jun 2003.
7. S. E. Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Comput. Speech Lang.*, 1(1):29–45, 1986.
8. K. Murphy and M. Paskin. Linear time inference in hierarchical hmms. In *Adv in Neural Inf Proc Sys 14*, 2002.
9. V. Narang, W. K. Sung, and A. Mittal. Computational annotation of transcription factor binding sites in *D. melanogaster* developmental genes. In *Proceedings of The 17th International Conference on Genome Informatics*, 2006.
10. L. R. Rabiner and B. H. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, pages 4–15, January 1986.
11. M. Tompa, N. Li, T. Bailey, G. Church, B. DeMoor, E. Eskin, A. Favorov, M. Frith, Y. Fu, W. Kent, V. Makeev, A. Mironov, W. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu. Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotech*, 23(1):137–44, 2005.
12. E. P. Xing, M. I. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Proceedings of the 19th Annual Conference on Uncertainty in AI*, 2003.