Supplement

Disentangling transcription factor binding site complexity

Ralf Eggeling

Supplement 1: Methodology

The input to Disentangler is always a set of N sequences of fixed length L over alphabet $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$.

We denote the ℓ-th symbol in the i-th sequence by $x_{i,\ell}$, the i-th sequence by $x_i = (x_{i,1}, \ldots, x_{i,L})$ and the entire data set by $\mathbf{x} = (x_1, \ldots, x_N)$.

We treat all sequences in our data set to be independent and identically distributed (i.i.d.), so we can view a single sequence of length L as random variables $X = (X_1, \ldots, X_L)$. From modeling perspective, the goal is to find a probability distribution over X, which we denote by $P(X)$.

Supplement 1.1: Graphical Models

A Bayesian network [1] is a general probabilistic graphical model that consists of a directed acyclic graph (DAG) $\mathcal{G}$ and a set of associated conditional probability parameters $\theta^{\mathcal{G}}$. Each node in the graph represents exactly one random variable of interest, and $\mathcal{G}$ implies

$$P(X|\mathcal{G}) = \prod_{\ell=1}^{L} P_{\mathcal{G}}(X_\ell|\mathsf{Pa}(X_\ell)), \tag{1}$$

where $\mathsf{Pa}(X_\ell)$ returns the parents of node X_ℓ in $\mathcal{G}$. Different DAGs thus encode different models assumption that often differ in their model complexity. Bayesian network structure learning, i.e., inferring a DAG directly from the data that that gives the optimal tradeoff between model complexity and data fit is an NP-hard problem. The problem can be simplified by introducing domain-specific knowledge that puts restrictions on the set of admissable DAGs.

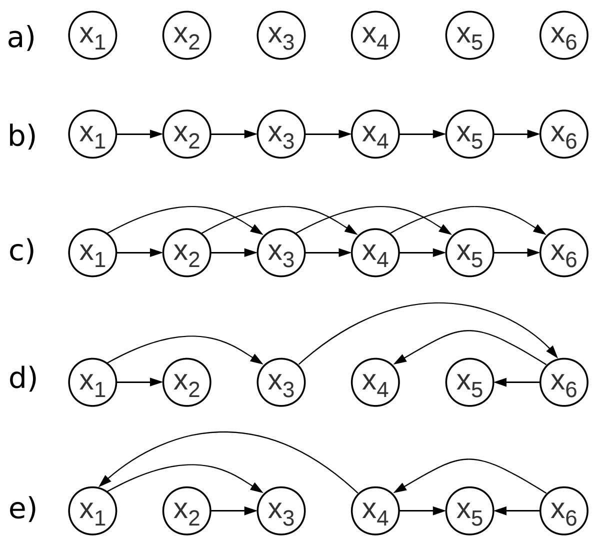


Figure 1: Probabilistic graphical models for transcription factor binding sites. Specifying the maximal model complexity allows different degrees of proximal and distal dependency. a) PWM model as graphical model b) First-order inhomogeneous Markov model (WAM) c) Second-order inhomogeneous Markov model d) tree-structured Bayesian network e) general Bayesian network.

Some of these are of particular importance for the modeling of biological sequence motifs (Figure 1). First, an empty DAG represents total statistical independence and is thus equivalent to a PWM model [2]. Second, setting $\mathsf{Pa}(X_\ell) = \{X_{\ell-d}, \ldots, X_{\ell-1}\}$ yields d-th order inhomogeneous Markov models, where the case of $d = 1$ is also known as weight array model [3]. Third, limiting the indegree $|\mathsf{Pa}(X_\ell)| \leq 1$ yields a forest structure, which becomes a tree-structured network when $|\mathsf{Pa}(X_\ell)| = 1$ holds for all but one sequence positions. Finally, limiting the indegree $|\mathsf{Pa}(X_\ell)| \leq d$ yields Bayesian network of (maximal) order d.

Supplement 1.2: Parsimonious Context Trees

Learning conditional probability distributions in graphical models has the drawback that the number of parameters grows exponentially with the number of parent variable. Hence, dependencies to multiple conditioning variables can only be justified by a large amount of data. One way to circumvent this problem, is to give structure to a conditional distribution, for instance via a context tree [4]. While being effective to learn, context trees are still somewhat limited in their expressiveness, since they arise from pruning the tree of context words, which corresponds to the parameter space of the model.

A parsimonious context trees [5], abbreviated PCT, allows to merge parameters based on the context and thus allows a higher degree of flexibility. A PCT of depth d is formally defined as a rooted, balanced tree that organizes the set of all possible realizations of the parent variables in different groups, represented by the leaves in the tree. Except for the root, each node in PCT is labeled by a non-empty subset of $\mathcal{A}$, while labels of all children of an arbitrary inner node forming a partition of $\mathcal{A}$. The cross product of the labels on each path from a leaf to the root defines a non-empty subset of $\mathcal{A}^d$ and thus the set cross products from of all leaves of a PCT forms a partition of $\mathcal{A}^d$. Examples for PCT and CT, which show the two structural differences PCTs allow, are given in Figure 2.

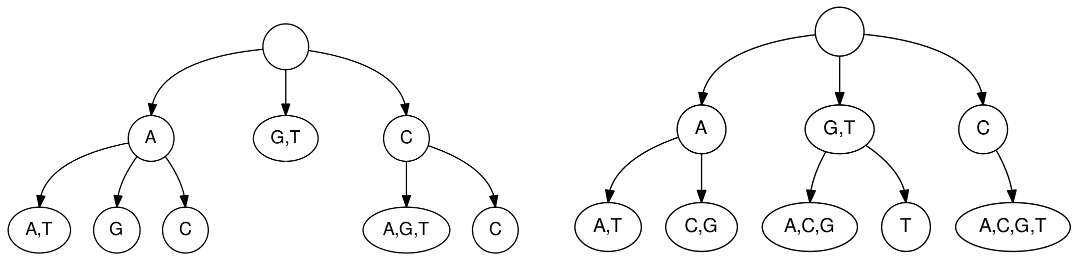


Figure 2: Examples for context tree (left) and Parsimonious context tree (right) of depth 2 over DNA alphabet. PCTs allow, in contrast to CTs, subtrees below nodes with more than one symbol, which allows context specific-skipping of a variable. They also allow variable grouping of symbols into child nodes of a common parent, whereas CTs allow at most one child node per parent labeled with more than one symbol.

Learning PCTs with a given scoring function in reasonable time is challenging, but can, especially for TFBS data, be solved with recent additions [6] to the original dynamic programming algorithm [5].

Supplement 1.3: Sparse graphical models for modeling TFBS

A PWM model ignores statistical dependencies among sequence positions entirely. Hence, the DAG is fixed and no conditional probability distributions exist, so learning the model requires only fitting the $3 * L$ free parameters.

As discussed in the previous section, using graphical models such inhomogeneous Markov models or Bayesian networks entails the problem of either over- or underfitting. Hence, we use PCT-augmented variants of them, that allow to effectively take into account proximal and distal dependencies within transcription factor binding motifs.

Supplement 1.3.1: Proximal dependence model

We use a dth-order inhomogeneous Markov model that is equipped with a PCTs at each position, which can reduce the actual order well below d. Here, the DAG (Figure 1) is fixed once d is selected, but there is, in addition to fitting the parameters, the structure learning task of selecting the optimal PCTs at each position based on the data. This inhomogeneous parsimonious Markov model (iPMM) is core of

InMoDe [7]. In this work, we refer to it as *proximal dependence* model (of order d) in order to emphasize the features that are modeled instead of the technical components that it uses.

Supplement 1.3.2: Distal dependence model

In analogy, a distal dependence model (of order d) is essentially a Bayesian network with $|\mathsf{Pa}(X_\ell)| \leq d$ that is also equipped with a PCT for each conditional distribution, hereby assuming that parent variables are ordered according to their position in the sequence. This model resembles variable-order Bayesian networks [8], which use traditional context trees [4] instead of PCTs. Here, PCTs for all potential pairs of variable and parent-variables are to be learned from the data. In addition the optimal DAG based on the PCT-based local scores needs to be found. For this task, we use Edmonds' algorithm for $d = 1$ [9, 10] and dynamic programming [11] otherwise. Learning additionally requires selecting the optimal network structure based on learned PCTs,

Supplement 1.4: Mixture models

The key idea behind a mixture model is to model $P(X)$ not by a single distribution, but to assume the existence of K component models. However, since we do not know which of the K models a particular sequence is associated with, we model this as a latent variable $U \in \{1, \ldots, K\}$. Hence,

$$P(X|\Theta) = \sum_{k=1}^{K} \alpha_k P_k(X|S_k, \theta_k) \tag{2}$$

where $\alpha_k = P(U = k)$ denotes the probability of latent variable U assuming component k, and S_k and θ_k denote the structure and all parameters in the k-th component model.

While dependency type and order could differ among the K mixture components in principle, we focus on the simple special case where all components are from the same model class. We thus consider mixtures of PWM models, mixture of dth-order proximal dependence models, and mixtures of dth-order distal dependence models.

In Equation 2, which defines the probability of an observed data point given the model, the number of mixture components K is given. When learning a mixture model, K can also be a user-defined input parameter [7]. A more sophisticated approach is to estimate $\hat{K}$ directly from data, typically from a small selection of candidate values, so that $\hat{K} \in \{K_{\min}, \ldots, K_{\max}\}$ [12]. In this work, we always consider $K_{\min} = 1$, so that only K_{max} remains as user-input.

Supplement 1.5: FAB algorithm

The most flexible model combination under consideration is mixture model of up to $K_{\max}$ components, each of which can be a d-th order distal dependence models. For each component, we denote the model structure, which includes DAG and PCTs, by S_k, and the set of required conditional probability parameters θ_k.

The learning task is thus to find (i) the number of mixture components $\hat{K}$, (ii) the model structures $\hat{S}_k$ for each of the $\hat{K}$ components, and (iii) the conditional probability parameters $\hat{\theta}_k$. Here, we use the factorized information criterion (FIC) as objective function [13], which is an approximation of the marginal log-likelihood of the mixture model. In practice, FIC cannot be computed exactly, since latent variables in $\mathbf{u}$, indicating the mixture component assignment, are involved. However, assuming a variational distribution $\mathbf{q}$ over the latent variables $\mathbf{u}$, a lower-bound $\mathcal{F}(\mathbf{q}, \Theta|\mathbf{x}) \leq \text{FIC}(\Theta|\mathbf{x})$ can be computed. It is given by

$$\mathcal{F}(\mathbf{q}, \Theta|\mathbf{x}) = \sum_{k=1}^{K} \sum_{i=1}^{N} q_{i,k} \log P_k(x_i|\theta_k) - \sum_{k=1}^{K} \frac{\mathcal{D}_k}{2} \log N - \frac{\mathcal{D}_\alpha}{2} \log N - H(\mathbf{q}) \tag{3}$$

Here, $\mathcal{D} = |\alpha - 1|$, $\mathcal{D}_k$ denotes the number of free parameters in θ_k, and $H(\mathbf{q})$ denotes the entropy of $\mathbf{q}$.

For the special case of $K = 1$, the variational distribution $\mathbf{q} = \mathbf{1}$ is fixed, so Equation 3 yields FIC exactly which turns out to coincide with the BIC score [14] for a single-component model. For $K > 1$, Equation 3 can be evaluated in closed form for arbitrary but fixed choices of $\mathbf{q}$, but it may be a loose lower bound to FIC if $\mathbf{q}$ is far from the true distribution. However, given some initial guess for it, we can iteratively update model parameters and variational distribution. Hereby, we monotonically improve $\mathcal{F}(\mathbf{q}^{(t)}, \Theta|\mathbf{x})$ for increasing t until it convergences to a local maximum.

Assuming an initial variational distribution $\mathbf{q}^{(0)}$ over $\mathbf{u}$, we iteratively execute the following four steps:

$$q_{ik}^{(t)} \quad \propto \quad \alpha_k^{(t-1)} P(x_i|\theta_k^{(t-1)}) \exp \frac{-\mathcal{D}_k}{2\alpha_k^{(t-1)} N} \tag{4}$$

$$\alpha_k^{(t)} = \sum_{i=1}^{N} q_{i,k}^{(t)}/N \tag{5}$$

$$S_k^{(t)} = \arg\max_{S_k} \sum_{i=1}^{N} q_{i,k}^{(t)} \ln P(x_i|\hat{\theta}_k(\mathbf{x}, \mathbf{q}^{(t)}) - \frac{\mathcal{D}_k}{2} \ln \sum_{i=1}^{N} q_{i,k}^{(t)} \tag{6}$$

[where $\hat{\theta}_k(\mathbf{x}, \mathbf{q}^{(t)})$ is the Maximum Likelihood estimate of θ_k on data $\mathbf{x}$, weighted by $\mathbf{q}$]

$$\theta_k^{(t)} = \arg\max_{\theta_k} \sum_{i=1}^{N} q_{i,k}^{(t)} \ln P(x_i|\theta_k^{(t)}) \tag{7}$$

This factorized asymptotic Bayesian (FAB) inference algorithm [13] is very similar in spirit to the EM algorithm [15], which it actually contains as special case when all model structures $S_1, \ldots, S_{K_{\max}}$ are fixed. Such a practically relevant special case within this work is when all component models are PWM models. As the fourth step of the FAB algorithm shows, parameter estimation with given structure is essentially a weighted maximum likelihood estimation.

The FAB algorithm terminates when the objective function (Equation 3) improves, within two iterations steps, by less than a user-specified threshold (in all case studies 10^{-6} is used). Since the algorithm finds only a local maximum of the target function, it needs to be started multiple times with different initializations. However, depending on the size and the nature of the data set as well as the number of component models and their maximal complexity, different amounts of restarts are needed to approximate the global optimum with some confidence.

IMD learns at every step only a two-component PWM mixture model, which optimizes quickly. Here, we use constantly 100 restarts, but terminate a single restarts that does not reach the the threshold of 10^{-6} after 60 seconds, so that a single recursion of IMD takes 6,000 second in the absolute worst case. For MCA, we use the following criteria, which turned out to give reasonable compromise between invested time and quality of the solution in preliminary studies:

- Three restarts at least for every model.
- As many additional restarts as possible within a given time limit.
- Terminate earlier when the best optimum was found within the first 1% of iteration steps.

The FAB algorithm is started with a maximal value $K_{\max}$ of mixture components and has a shrinking function [13] that reduces the number of components if they are not supported by the data. To be precise, we remove a component k, if $\sum_{i=1}^{N} q_{i,k} < \delta$. In the current implementation, we use $\delta = \max(3, 0.01 * N)$. However, shrinking is some effective only after an excessive of running time due to the complexity of the component models. In order to select number of mixture components and component complexity simultaneously, it is thus needed to start the optimization with all possible values of $K \leq K_{\max}$ and pick the best result. Since the running time is dominated by the run of $K_{\max}$, and most applications, such as IMD and MCA need different values of $K_{\max}$ anyway, this does not constitute a lot additional effort.

Supplement 2: Tools used in case studies

Supplement 2.1: NPLB

No Promoter Left Behind (NPLB) is a tool for characterizing promoter architectures [16], but can be also be used to cluster a set of pre-aligned binding sites, including a model selection step to infer the optimal the number of clusters.

We use the latest version available at
`https://github.com/NarlikarLab/NPLB`
for building a command line version of the tool.

Supplement 2.2: DIVERSITY

DIVERSITY [17] is a motif discovery tool that attempts to find multiple modes, which are different PWM models, of TFBS binding from ChIP-seq data. A webserver is available at:
`http://diversity.ncl.res.in`

We use the latest version available at
`https://narlikarlab.github.io/DIVERSITY`
for building a command line version of the tool.

Supplement 2.3: InMoDe

InMoDe [7] is a collection of seven tools for learning, leveraging, and visualizing intra-motif dependencies within DNA binding sites and similar functional nucleotide sequences.

We use the command line application, which is available as runnable .jar at:
`http://jstacs.de/index.php/InMoDe`

For the motif discovery studies, we use the subtool “FlexibleMoDe” with default parameters, which always returns two motifs. We consider the one that represents the majority of sequences as primary motif of interest for further analysis with Disentangler.

Supplement 2.4: Slim-Dimont

Sparse local inhomogeneous mixture (Slim) models [18] are statistical models for discrete sequences (as for instance DNA sequences) that allow for simultanous discriminative learning of features and model parameters. Slim models can be used in combination with Dimont for de-novo motif discovery.

We use the command line application, which is available as runnable .jar at:
`http://jstacs.de/index.php/Slim`

If the tool returns more than one motif, we consider the first motif as primary motif of interest for further analysis with Disentangler.

Supplement 2.5: BaMMmotif

Bayesian Markov Model motif discovery tool (BaMMmotif) [19] is an expectation maximization algorithm for the de novo discovery of enriched motifs as modelled by higher-order Markov models.

Following the suggestions from the authors, we use the latest version BaMMmotif2, available from
`https://github.com/soedinglab/BaMMmotif2`

It requires initial seeds to start motif discovery, which can be, according to suggestions from the authors, be obtained by first running another motif discovery software called PenGmotif, available from
`https://github.com/soedinglab/PEnG-motif`

If the sequential application of these tools returns more than one motif, we consider the first motif as primary motif of interest for further analysis with Disentangler.

Supplement 3: Data extraction

Supplement 3.1: JASPAR

When extracting data from the 2016 Jaspar release [20], we pick all TF data sets that have actual sequence data, as opposed to sole weight matrices, available, and download the sites from
`http://jaspar2016.genereg.net/html/DOWNLOAD/sites.tar.gz`
For each data set, we extract the motif alignment proposed by the database, which is indicated by upper-case letters in the sequence files. We further process the data by two simple steps for removing artifacts. First, we discard all binding sites that contain ambiguous nucleotides. Second, we verify for each data set that position-specific mononucleotide counts in the aligned binding sites indeed reproduce the weight matrix given in the database. This is not true for a few data sets from *C. elegans*, where all sites that were originally located on the forward strand are shifted into 3' direction by 2bp in relation to those that were originally located on the negative strand. We thus correct for this shift during the data extraction in order to avoid artificial, shift-induced intra-motif dependencies. After having completed these pre-processing steps, we discard all data sets that contain less than 100 sequences and finally retain the following 158 data sets. Data sets highlighted in boldface are discussed in the main manuscript in particular detail.

JASPAR ID	Sequence length	Sample size	TF Name	TFclass ID
MA0003.2	15	5098	TFAP2A	1.3.1.0.1
MA0007.2	15	11206	AR	2.1.1.1.4
MA0014.2	19	896	PAX5	3.2.2.2.2
MA0024.2	11	1059	E2F1	3.3.2.1.1
MA0035.3	11	17955	Gata1	2.2.1.1.1
MA0036.2	14	4380	GATA2	2.2.1.1.2
MA0037.2	8	4628	GATA3	2.2.1.1.3
MA0039.2	10	4311	Klf4	2.3.1.2.4
MA0047.2	12	800	Foxa2	3.3.1.1.2
MA0050.2	21	1362	IRF1	3.5.3.0.1
MA0052.2	15	1473	MEF2A	5.1.1.1.1
MA0058.2	10	24565	MAX	1.2.6.5.5
MA0060.1	16	116	NFYA	
MA0060.2	18	8768	NFYA	4.2.1.0.1
MA0062.2	11	987	Gabpa	3.5.2.1.4
MA0065.2	15	855	Pparg::Rxra	
MA0076.2	11	3427	ELK4	
MA0079.3	11	8734	SP1	2.3.1.1.1
MA0080.3	15	63715	Spi1	3.5.2.5.1
MA0083.2	18	2277	SRF	5.1.2.0.1
MA0093.2	11	16842	Lin-14	
MA0095.2	12	7171	YY1	
MA0098.2	15	1868	Ets1	3.5.2.1.1
MA0100.2	10	979	Myb	3.5.1.1.1
MA0102.3	11	15318	CEBPA	1.1.8.1.1
MA0103.2	9	3555	ZEB1	3.1.8.3.1
MA0104.3	8	1403	Mycn	1.2.6.5.2
MA0105.3	11	5112	NFKB1	6.1.1.1.1
MA0106.2	15	1231	TP53	
MA0112.2	20	467	ESR1	2.1.1.2.1
MA0114.2	15	16768	HNF4A	2.1.3.2.1
MA0137.2	15	2069	STAT1	6.2.1.0.1
MA0137.3	11	3629	STAT1	6.2.1.0.1
MA0138.2	11	867	REST	2.3.4.0.27
MA0139.1	11	943	CTCF	2.3.3.50.1
MA0140.2	18	4955	GATA1::TAL1	
MA0141.1	12	3605	Esrrb	2.1.1.2.4
MA0142.1	15	1356	Pou5f1::Sox2	
MA0143.1	15	662	Sox2	4.1.1.2.2
MA0143.3	8	1476	Sox2	4.1.1.2.2
MA0144.1	19	821	Stat3	6.2.1.0.3
MA0144.2	11	21620	STAT3	6.2.1.0.3
MA0145.1	14	4039	Tcfcp2l1	
MA0146.1	20	468	Zfx	2.3.3.65.1
MA0147.1	10	681	Myc	1.2.6.5.1
MA0147.2	10	5335	Myc	1.2.6.5.1
MA0148.1	11	888	FOXA1	3.3.1.1.1
MA0148.3	15	22008	FOXA1	3.3.1.1.1
MA0149.1	18	105	EWSR1-FLI1	
MA0150.2	15	726	Nfe2l2	
MA0154.2	11	33855	EBF1	6.1.5.0.1
MA0161.1	6	6912	NFIC	
MA0162.2	14	12256	EGR1	2.3.1.3.1
MA0216.2	11	2303	cad	
MA0247.2	10	515	tin	
MA0258.1	18	357	ESR2	2.3.1.3.2
MA0258.2	15	8243	ESR2	2.3.1.3.2
MA0259.1	8	104	ARNT::HIF1A	
MA0261.1	6	158	lin-14	
MA0452.2	14	1296	Kr	
MA0461.1	8	7714	Atoh1	1.2.3.4.8
MA0462.1	11	10522	BATF::JUN	
MA0463.1	14	956	Bcl6	2.3.3.22.2
MA0464.1	11	15804	Bhlhe40	
MA0465.1	11	1597	CDX2	3.1.1.9.2
MA0466.1	11	99494	CEBPB	1.1.8.1.2
MA0467.1	11	2097	Crx	3.1.3.17.3

MA0468.1	11	38217	DUX4	3.1.3.7.5
MA0469.1	11	2549	E2F3	3.3.2.1.3
MA0470.1	11	1878	E2F4	3.3.2.1.4
MA0471.1	11	2757	E2F6	3.3.2.1.6
MA0472.1	15	1246	EGR2	2.3.1.3.2
MA0473.1	13	13518	ELF1	3.5.2.3.1
MA0474.1	11	16727	Erg	3.5.2.1.6
MA0475.1	11	3667	FLI1	3.5.2.1.5
MA0476.1	11	29396	FOS	1.1.2.1.1
MA0477.1	11	5272	FOSL1	1.1.2.1.3
MA0478.1	11	5318	FOSL2	1.1.2.1.4
MA0479.1	11	8211	FOXH1	3.3.1.8.1
MA0480.1	11	2490	Foxo1	3.3.1.15.1
MA0481.1	15	311	FOXP1	3.3.1.16.1
MA0482.1	11	2746	Gata4	2.2.1.1.4
MA0483.1	11	1761	Gfi1b	2.3.3.21.2
MA0484.1	15	9452	HNF4G	2.1.3.2.2
MA0485.1	13	885	Hoxc9	
MA0486.1	15	225	HSF1	3.4.1.0.1
MA0488.1	13	20968	JUN	1.1.1.1.1
MA0489.1	14	10956	JUN(var.2)	1.1.1.1.1
MA0490.1	11	16992	JUNB	1.1.1.1.2
MA0491.1	11	38710	JUND	1.1.1.1.3
MA0492.1	15	33631	JUND(var.2)	1.1.1.1.3
MA0493.1	11	526	Klf1	2.3.1.2.1
MA0494.1	19	1269	Nr1h3::Rxra	
MA0495.1	18	53758	MAFF	1.1.3.2.1
MA0496.1	15	60790	MAFK	1.1.3.2.3
MA0497.1	15	2209	MEF2C	5.1.1.1.3
MA0498.1	15	2607	Meis1	3.1.4.2.1
MA0499.1	13	24514	Myod1	1.2.2.1.1
MA0500.1	11	19356	Myog	1.2.2.1.2
MA0501.1	15	1090	MAF::NFE2	
MA0502.1	15	7020	NFYB	4.2.1.0.2
MA0503.1	11	3429	Nkx2-5(var.2)	3.1.2.17.22
MA0504.1	15	395	NR2C2	2.1.3.4.2
MA0505.1	15	1702	Nr5a2	2.1.5.0.2
MA0506.1	11	4624	NRF1	1.1.1.2.2
MA0507.1	13	2287	POU2F2	3.1.10.2.22
MA0508.1	15	4603	PRDM1	2.3.3.12.1
MA0509.1	14	2138	Rfx1	3.3.3.0.1
MA0510.1	15	3868	RFX5	3.3.3.0.5
MA0511.1	15	1062	RUNX2	6.4.1.0.1
MA0512.1	11	5348	Rxra	2.1.3.1.1
MA0513.1	13	899	SMAD2::SMAD3::SMAD4	
MA0514.1	10	2067	Sox3	4.1.1.2.3
MA0515.1	10	249	Sox6	4.1.1.4.2
MA0516.1	15	1686	SP2	2.3.1.1.2
MA0517.1	15	620	STAT1::STAT2	
MA0518.1	14	2873	Stat4	6.2.1.0.4
MA0519.1	11	16507	Stat5a::Stat5b	
MA0520.1	15	1852	Stat6	6.2.1.0.7
MA0521.1	11	12895	Tcf12	1.2.1.0.3
MA0522.1	11	17261	Tcf3	4.1.3.0.2
MA0523.1	14	4188	TCF7L2	4.1.3.0.3
MA0524.1	15	18426	TFAP2C	1.3.1.0.3
MA0525.1	20	9632	TP63	
MA0526.1	11	13819	USF2	1.2.6.2.2
MA0527.1	15	705	ZBTB33	2.3.2.1.122
MA0528.1	21	15235	ZNF263	2.3.3.0.799
MA0529.1	15	3985	BEAF-32	
MA0530.1	15	474	cnc::maf-S	
MA0531.1	15	1902	CTCF	
MA0532.1	15	118	Stat92E	
MA0533.1	21	4737	su(Hw)	
MA0534.1	15	104	EcR::usp	
MA0535.1	15	102	Mad	
MA0536.1	11	869	pnr	
MA0537.1	11	3368	blmp-1	
MA0538.1	15	156	daf-12	
MA0541.1	15	2206	efl-1	
MA0542.1	8	722	elt-3	
MA0543.1	15	1253	eor-1	
MA0544.1	12	310	snpc-4	
MA0545.1	11	381	hlh-1	
MA0546.1	10	1010	pha-4	
MA0547.1	15	318	skn-1	
MA0548.1	15	150	AGL15	
MA0550.1	14	153	BZR1	
MA0552.1	14	114	PIF1	
MA0553.1	8	147	SMZ	
MA0554.1	15	888	SOC1	
MA0556.1	15	291	AP3	
MA0558.1	21	275	FLC	
MA0559.1	14	558	PI	
MA0560.1	10	527	PIF3	
MA0561.1	8	335	PIF4	
MA0562.1	8	286	PIF5	
MA0563.1	11	150	SEP3	
MA0940.1	13	1622	AP1	
MA1012.1	14	142	AGL27	

Supplement 3.2: GTRD

We use ChIP-seq metaclusters for human and mouse from GTRD [21], available from:
`http://gtrd.biouml.org/downloads/current/human_meta_clusters.interval.gz`
`http://gtrd.biouml.org/downloads/current/mouse_meta_clusters.interval.gz`

Metaclusters aggregate multiple ChIP-seq experiments and evaluation pipelines, yielding a unique data set for each TF, which is identified by its TFclass [22] ID. We associate TFclass IDs with JASPAR IDs according to the transcription factor name or variants thereof.

However, not all TF names in JASPAR have a unique TFclass ID, and not all TFs that have a TFclass ID are represented have meta clusters in GTRD. For all JASPAR data sets, where both TFclass ID can be identified, and a metacluster is available, the TFclass ID is given in the table in the previous section. These are the datasets-pairs that are used in the validation study (Figure 6C in the manuscript).

For each match, we extract the sequences for the metacluster from the human/mouse genome, and treat it as positive data set. For each positive data set, we generate control data by learning a second-order homogeneous Markov chain, and sampling 100,000 sequences of length $\bar{L}$ from it, where $\bar{L}$ is the mean sequence length in the positive data set.

Supplement 3.3: ENCODE

we use all data sets in the Uniform TFBS track of the ENCODE project [23], which are available at
`http://hgdownload.cse.ucsc.edu/goldenPath/hg19/encodeDCC/wgEncodeAwgTfbsUniform`
as input data for motif discovery. The data sets differ in TF (antibody), cell line, treatment, or producing lab, but have been processed with a uniform pipeline, yielding a ranked peak list with corresponding enrichment scores. For each data set, we pick the top $5,000$ peaks and extract, for each peak, a 500bp sequence fragment (250bp upstream/downstream from the peak center) the human genome, version hg19. Extraction is done with a perl script
`http://www.jstacs.de/downloads/extract_data.pl`
available with documentation at
`http://jstacs.de/index.php/Slim`
in order to ensure that the sequence files have a .fasta annotation that compatible with Slim-Dimont [18]. This information, which contains sequence weights, is ignored by the other motif discovery tools.

Supplement 4: Additional results

This section presents (i) additional figures that could not be included in the main manuscript for space constraints, and (ii) some additional studies that address secondary aspects of the topic.

Supplement 4.1: Preliminary study: Finding optimal number of PWMs

We consider the following simple model selection task: Given a set of pre-aligned and strand-oriented TFBS, select the optimal number PWMs for modeling the data when choosing from a pre-defined range $(1, \ldots, K_{\max})$.

We compare the model selection results from Disentangler to two other sequence analysis tools that, albeit originally proposed for slightly different use cases, contain the given task as special case. NPLB [16] can be constrained to the given task by setting its λ-parameter to 0 and the minimal and maximal number of architectures to 1 and $K_{\max}$. DIVERSITY [17] is directly applicable to the given task when the motifs widths are fixed to the length of the input sequences, and the minimal maximal number of modes to 1 and $K_{\max}$.

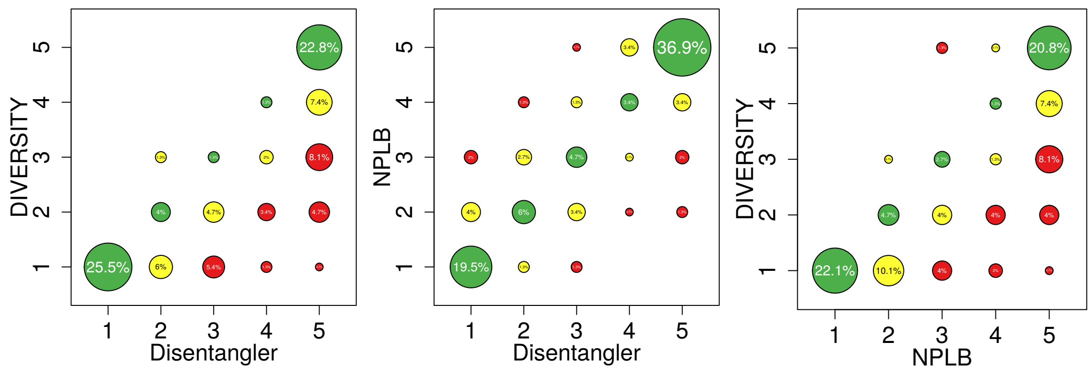


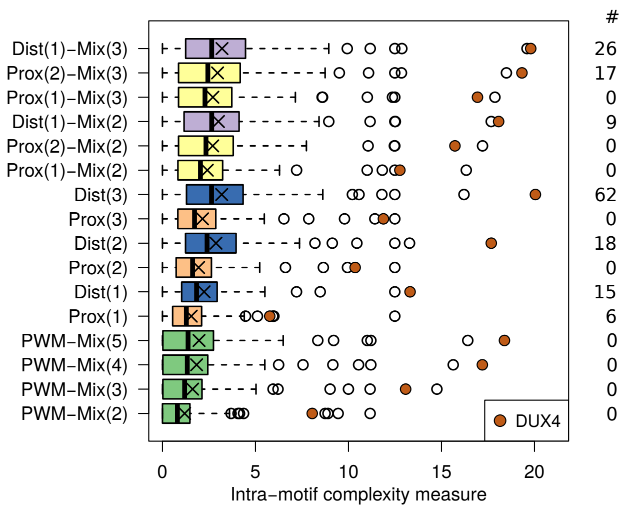
Figure 3: Cross-comparison of the predictions from Disentangler, NPLB, and DIVERSITY for determining the optimal among up to $K_{\max} = 5$ PWMs. The area of each circle is proportional to number of data sets.

We apply these three tools to find the optimal number of PWM-motifs for 158 JASPAR data sets. While Disentangler and NLPB give a solution even for very large data sets in a few hours at most, DIVERSITY did not finish computations for nine data sets, all with $N > 25,000$, within seven days. For these nine data sets, both Disentangler and NPLB output the maximal number of five motifs to be optimal. For the remaining 148 data sets, we plot the cross-comparisons among the predictions in Figure 3. NPLB and Disentangler agree in their prediction for the majority of data sets, and in the case of disagreement, the number of predicted motifs varies only slightly. DIVERSITY, however, appears to be more conservative in this setting, predicting in many cases substantially less motifs than the two other tools, but still more than one for the majority of data sets.

The observations are relevant in two aspects. First, the FIC-based learning approach used in Disentangler is – when being limited to PWM models as mixture components – at least as liberal as these of comparable tools, so it does not systematically under-estimate the number of motifs. Second, limiting the models to PWMs leads in many cases to the prediction of multiple models due to additional statistical features in the data beyond PWM assumptions. Since each data set can be assumed to contain binding sites of only one TF and little evidence for intermixing exists here, selecting the optimal number of PWM models alone cannot accurately detect inter-motif heterogeneity, which justifying the need for a new method like IMD.

Supplement 4.2: Intra-motif complexity measure on JASPAR data

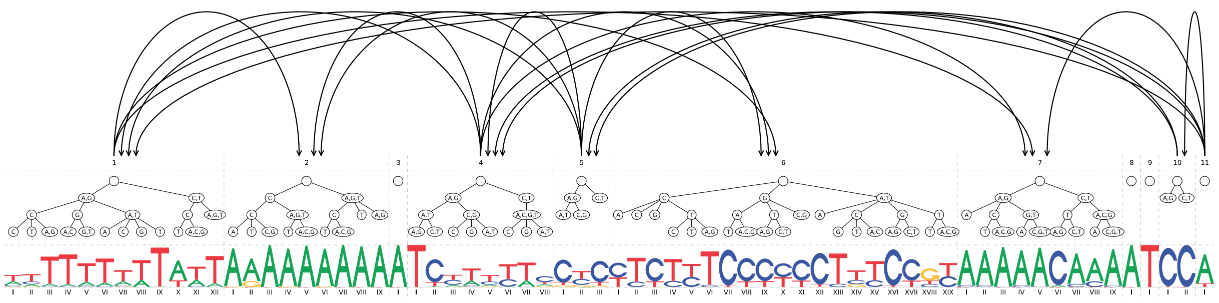
Distribution of the intra-motif complexity measure over 158 JASPAR data sets for different models. The rightmost column shows how many times a particular model is optimal. These values sum up to 153, the PWM model is optimal for the remaining five data sets.



According to MCA, mixtures of PWM models are a surprisingly poor representation of intra-motif complexity average, even a five-component mixture hardly outperforms simple proximal dependency. A likely explanation is that weak dependencies cannot be effectively represented with a mixture of PWMs. Even the two-component mixture model requires, compared to a single PWM model, twice as many model parameters to be fitted, whereas dependency models that contain a local structure such as PCTs can spend additional parameters only when needed. For some data sets, PWM mixtures are a better representation than proximal dependency, though. One example are motifs with a conserved core of three nucleotides or more, where a mixture model can take into account these features through multiple components. Proximal dependency cannot model correlations among both flanks surrounding the core, whereas a mixture model can take into account these features through multiple components. However, distal dependency captures the same features often in a more effective way. Compared to proximal dependency of the same order, distal dependency never yields a lower intra-motif complexity, as the model classes are nested. The more relevant comparison between proximal and distal dependency concerns the magnitude of improvement achieved by the latter. Here, we observe an increase in intra-motif complexity of about 33% on average, so only 2/3 of dependencies can be utilized by Markov models are variants thereof. Third-order distal dependency performs overall best, with a median intra-motif complexity of 2.5. Mixtures with first-order distal dependency components fail to improve on that, but they are faster to learn when the motif is long.

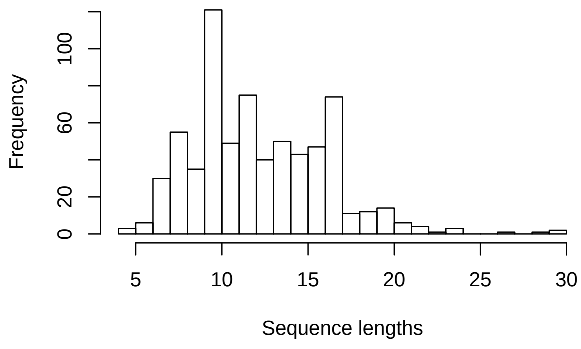
Supplement 4.3: Dist(3) visualization for DUX4

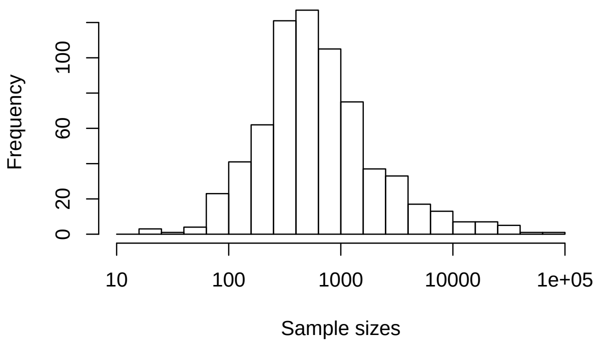
The following figure is an addition to Figure 7 of the main manuscript. It visualizes the third-order distal dependence model for the DUX4 data set from JASPAR. It achieves the highest intra-motif complexity measure as well as the highest TPR for classifying the corresponding GTRD data set.



Supplement 4.4: Benchmark validation on SwissRegulon TFBS

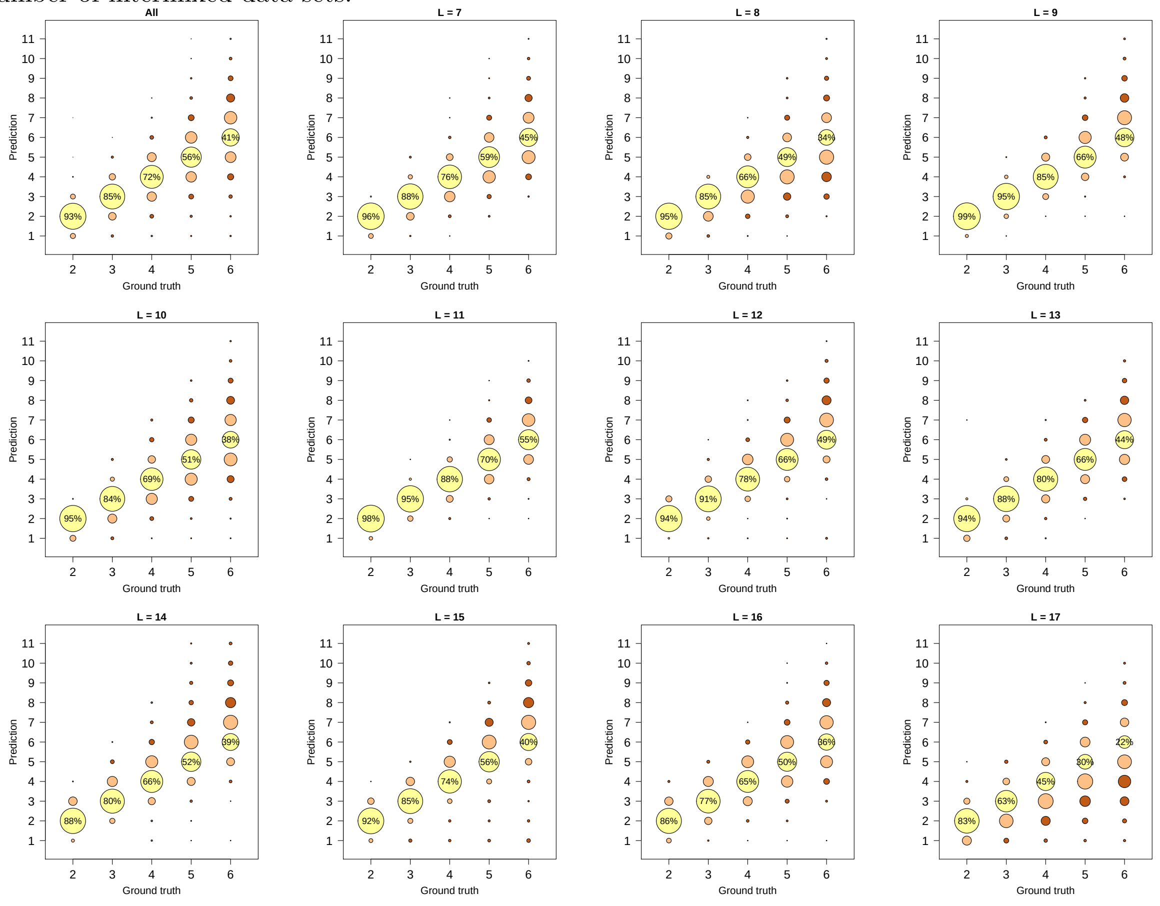
The main benchmark studies rely on data sets of pre-aligned TFBS from JASPAR. For further validation, we repeat the benchmark with pre-aligned TFBS from SwissRegulon [24]. `http://swissregulon.unibas.ch/data/hg19/hg19_sites.gff.gz`
The alignments following the tag “Sequence” were extracted and compiled into individual data sets according to the tag “Motif”. This preprocessing yields 683 data sets with the following distributions of sequence length L and sample size N:





Supplement 4.4.1: Benchmarking IMD

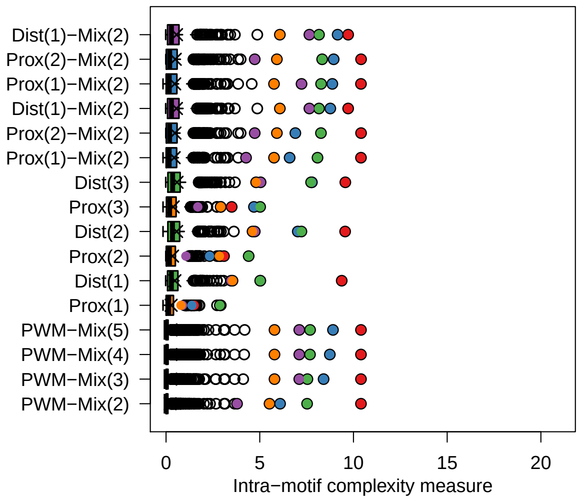
We follow the exact experimental setup that produces Figure 5A in the main manuscript. For this purpose we construct intermixtures from data sets of same length L for all possible $7 \leq L \leq 17$ (there are only few data sets for lengths longer or shorter than that). Using the intermixture threshold $T = 0.19$, we obtain the following results. The first plot below is an average over all sequence lengths, the remaining plots are specific for each data set group. The results demonstrate that IMD also works on data from SwissRegulon and that the threshold T is a robust choice that neither systematically over- nor underestimates the number of intermixed data sets.

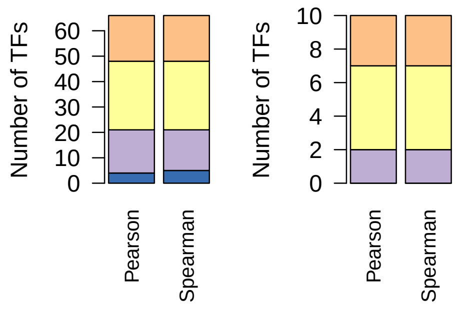


Supplement 4.4.2: Benchmarking MCA

Next, we apply MCA and compute the intra-motif complexity measure for each model under consideration and each data set: In contrast to data sets extracted from JASPAR, the vast majority of data sets in SwissRegulon show only a very small amount of intra-motif complexity, which is likely due to the particular computational pipeline that did predict them. For further benchmarking, we thus focus on the 66 data which have (i) $\Delta > 1$ and (ii) an associated data set in GTRD (identified by TFClass ID, in the same way as for JASPAR data). For each of these data sets, we compute the correlation between Δ_m and the predictive performance on GTRD meta peaks.

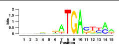
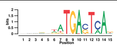
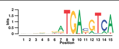
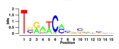
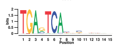
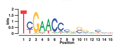
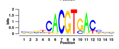
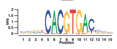
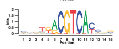
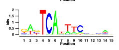
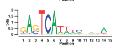
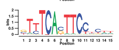
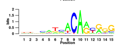
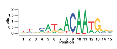
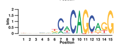
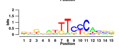
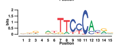
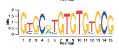
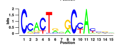
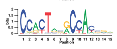
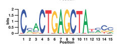
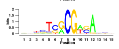
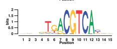
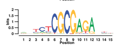
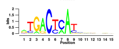
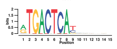
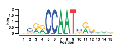
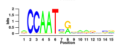
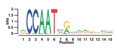
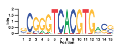
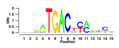
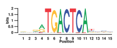
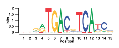
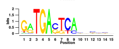
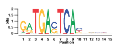
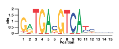
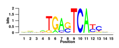
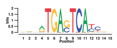
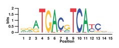
Due to the small intra-motif complexities, the models differ only slightly. As a consequence, there is a large number of insignificant correlations. The positive correlations outnumber the negative ones, but only by a relative small margin, even when limiting the data sets only to the 10 data sets with the highest Δ, Hence, very strong conclusions cannot be drawn based on the available data.

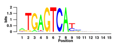
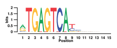
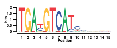
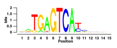
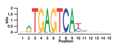
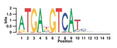
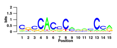
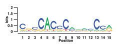
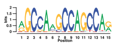
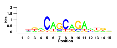
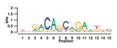
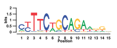
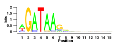
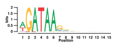
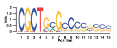
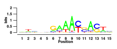
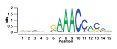
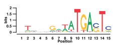
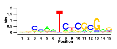
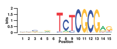
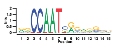
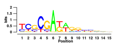
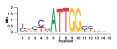
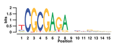
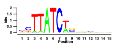
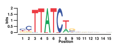
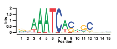
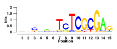
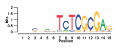
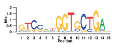
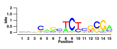
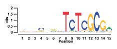
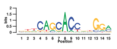
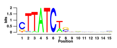
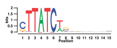
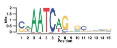
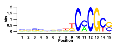
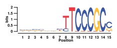
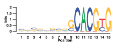
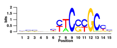
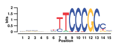
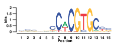
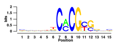
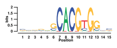
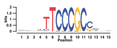
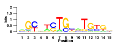
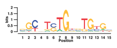
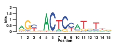
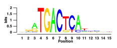
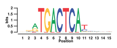
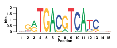
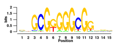
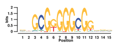
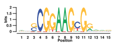
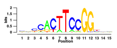
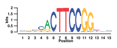
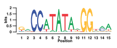
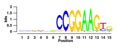
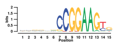
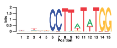
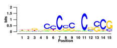
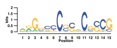
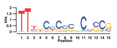
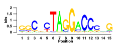
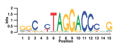
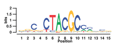
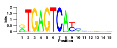
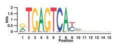
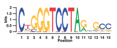
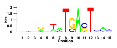
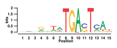
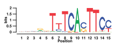
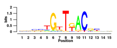
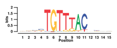
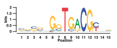
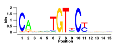
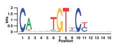
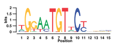
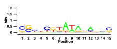
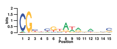
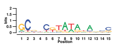
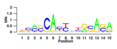
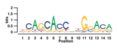
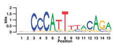
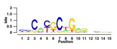
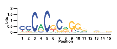
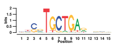
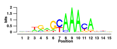
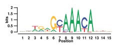
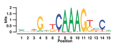
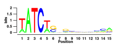
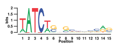
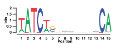
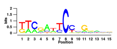
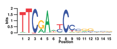
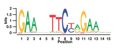
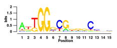
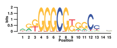
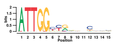
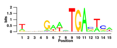
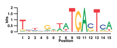
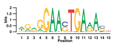
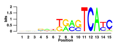
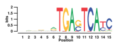
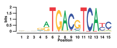
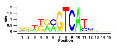
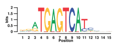
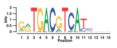
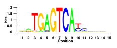
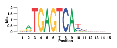
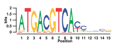
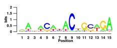
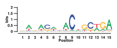
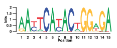
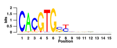
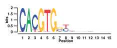
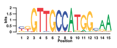
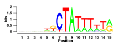
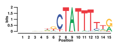
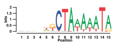
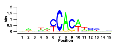
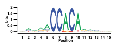
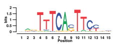
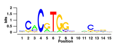
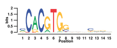
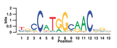
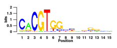
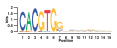
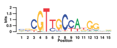
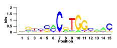
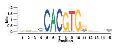
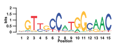
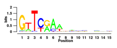
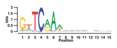
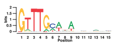


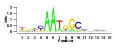
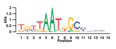
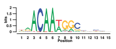
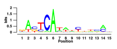
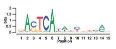
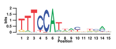
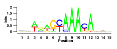
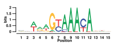
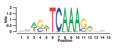
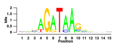
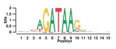
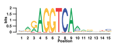
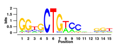
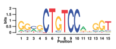
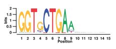
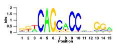
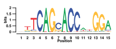
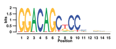
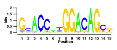
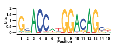
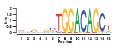
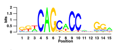
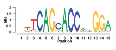
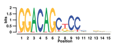
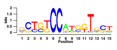
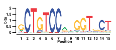
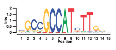
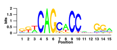
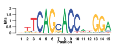
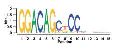
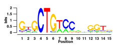
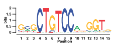
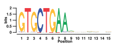
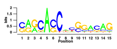
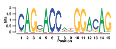
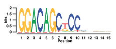
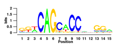
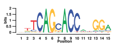
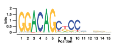
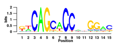
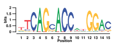
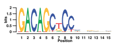
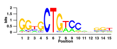
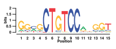
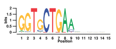
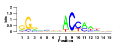
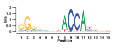
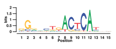
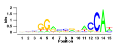
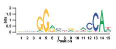
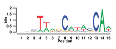
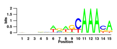
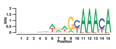
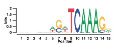
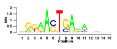
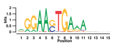
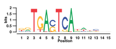
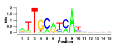
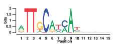
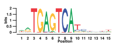
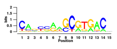
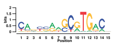
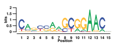
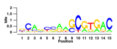
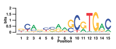
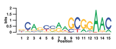
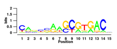
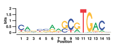
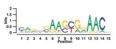
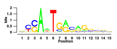
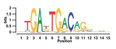
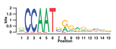
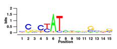
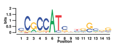
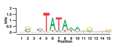
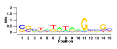
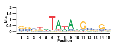
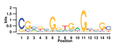
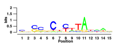
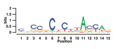
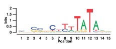
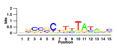
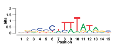
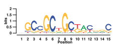
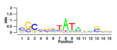
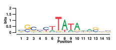
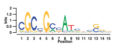
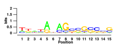
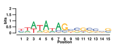
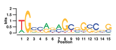
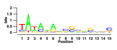
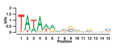
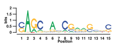
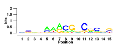
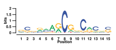
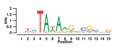
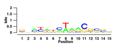
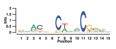
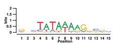
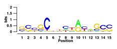
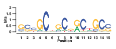
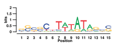
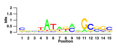
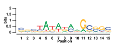
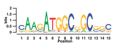
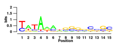
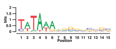
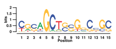
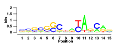
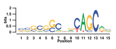
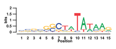
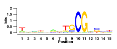
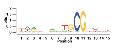
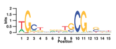
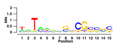
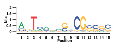
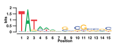
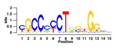
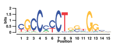
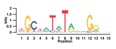
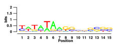
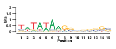
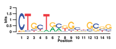
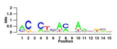
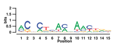
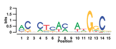
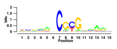
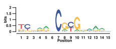
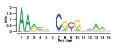
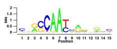
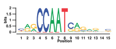
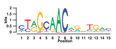
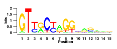
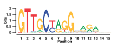
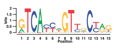


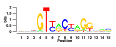
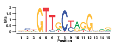
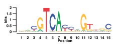
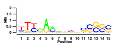
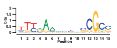
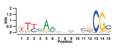
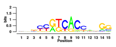
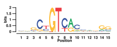
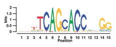
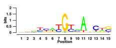
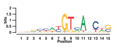
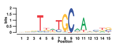
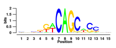
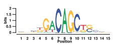
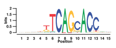
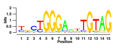
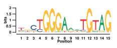
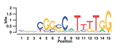
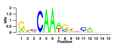
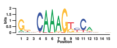
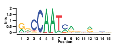
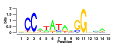
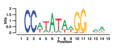
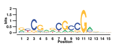
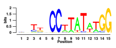
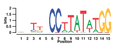
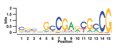
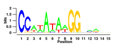
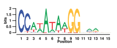
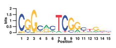
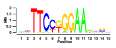
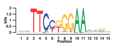
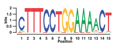
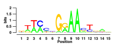
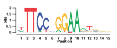
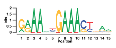
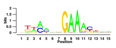
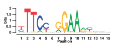
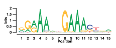
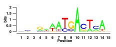
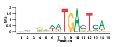
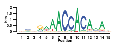
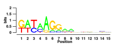
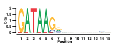
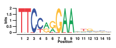
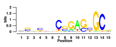
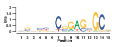
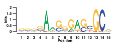
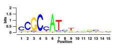
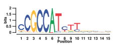
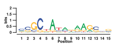
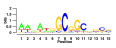
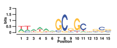
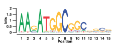
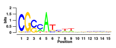
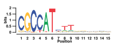
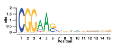
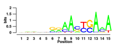
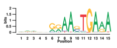
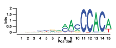
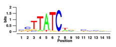
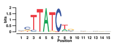
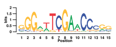
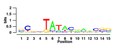
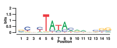
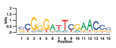
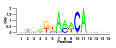
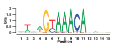
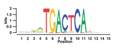
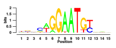
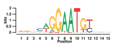
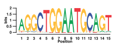
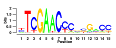
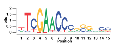
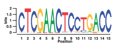
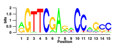
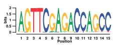
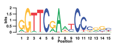
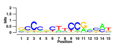
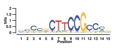
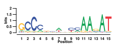
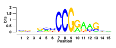
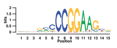
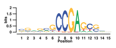
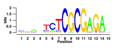
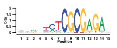
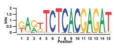
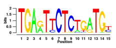
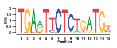
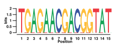
Supplement 4.5: Disentangler applied to output of Slim-Dimont

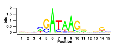
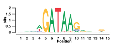
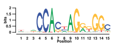
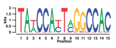
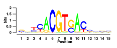
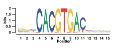
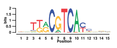
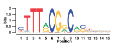
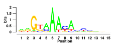
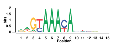
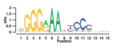
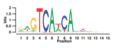
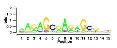
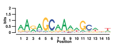
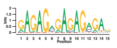
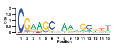
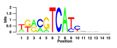
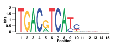
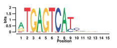
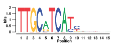
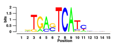
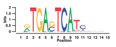
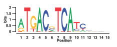
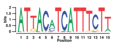
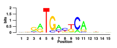
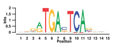
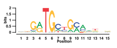
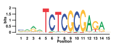
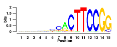
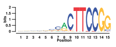
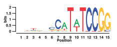
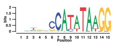
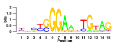
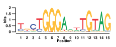
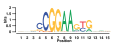
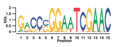
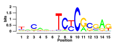
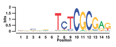
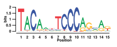
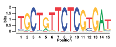
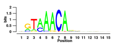
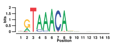
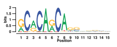
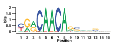
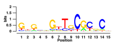
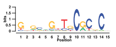
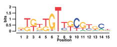
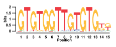
The following tables shows sequence logos of the primary motif reported by Slim-Dimont with default parameters on ENCODE ChIP-seq data, together with the IMD clusters, restricted to these cases where IMD return $\hat{M} \in (2,3)$. The ID given the in the table is the part of the filename that uniquely identifies a given dataset within the Uniform TFBS track. In other words, the filename of a data set is given by wgEncodeAwgTfbs**ID**UniPk.narrowPeak.gz
Data sets highlighted in boldface are the examples that are discussed in detail.

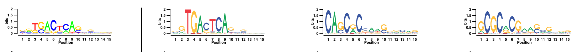
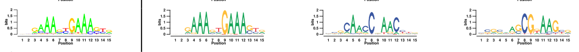
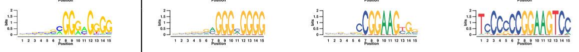
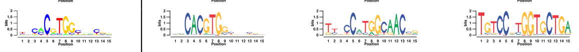
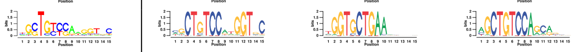
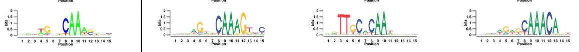
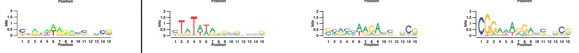
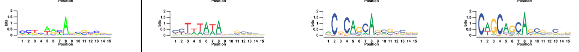
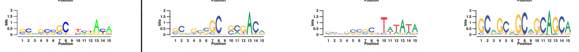
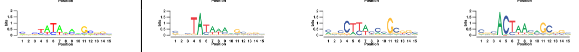
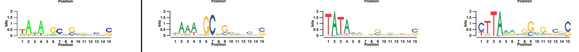
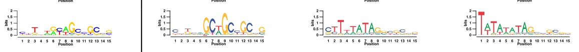
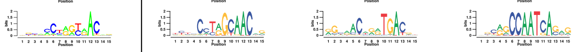
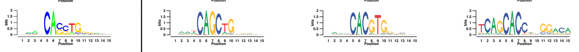
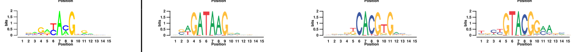
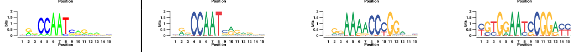
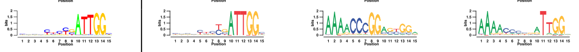
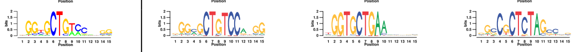
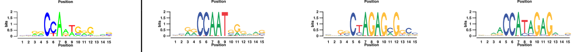
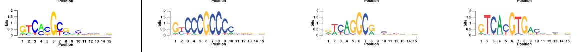
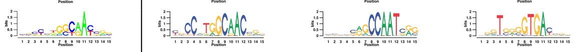
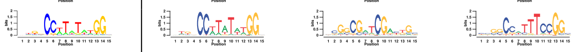
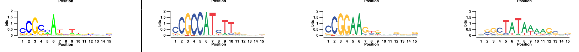
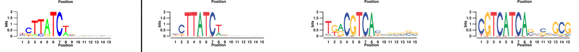
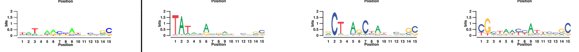
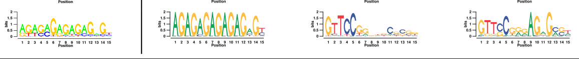
TF name	ID	Slim-Dimont prediction	IMD cluster 1	IMD cluster 2
ATF2	HaibGm12878Atf2sc81188V0422111			
ATF3	HaibA549Atf3V0422111Etoh02			
ATF3	HaibGm12878Atf3Pcr1x			
BCL11A	HaibGm12878Bcl11aPcr1x			
BCL11A	HaibH1hescBcl11aPcr1x			
BCL3	HaibGm12878Bcl3V0416101			
BDP1	SydhK562Bdp1			
BRCA	SydhHelas3Brca1a300Iggrab			
c-Fos	**SydhHelas3Cfos**			
c-Fos	SydhGm12878Cfos			
c-Jun	SydhH1hescCjunIggrab			
c-Jun	SydhK562Cjun			
c-Jun	SydhK562CjunIfna30			

c-Jun	SydhK562CjunIfna6h			
c-Jun	SydhK562CjunIfng30			
c-Myc	UtaH1hescCmyc			
CBX3	HaibK562Cbx3sc101004V0422111			
CCNT2	SydhK562Ccnt2			
CEBPB	HaibGm12878Cebpbsc150V0422111			
CHD2	**SydhH1hescChd2Iggrab**			
CHD2	SydhHelas3Chd2Iggrab			
COREST	SydhK562Corestab24166Iggrab			
COREST	SydhGm12878Corestsc30189Iggmus			
COREST	SydhHepg2Corestsc30189Iggrab			
COREST	SydhK562Corestsc30189Iggrab			
E2F6	HaibK562E2f6V0416102			
E2F6	SydhHelas3E2f6			
E2F6	SydhK562E2f6Ucd			
eGFP-HDAC8	UchicagoK562Ehdac8			
eGFP-JunD	UchicagoK562Ejund			
Egr-1	HaibH1hescEgr1V0416102			
ELK1	**SydhK562Elk112771Iggrab**			
ELK1	SydhHelas3Elk112771Iggrab			
EZH2	BroadHsmmtEzh239875			
FOSL1	HaibH1hescFosl1sc183V0416102			
FOSL1	HaibK562Fosl1sc183V0416101			
FOXM1	HaibGm12878Foxm1sc502V0422111			
FOXP2	HaibPfsk1Foxp2Pcr2x			
GR	HaibEcc1GrV0416102Dex100nm			
GTF2F1	SydhHelas3Gtf2f1ab28179Iggrab			
HDAC2	BroadK562Hdac2a300705a			
HDAC2	HaibH1hescHdac2sc6296V0416102			
HDAC2	HaibHepg2Hdac2sc6296V0416101			
HDAC2	HaibK562Hdac2sc6296V0416102			
HSF1	SydhHepg2Hsf1Forskln			
IRF1	SydhK562Irf1Ifng6h			
IRF4	HaibGm12878Irf4sc6059Pcr1x			
JunD	**SydhH1hescJundIggrab**			
JunD	SydhHepg2JundIggrab			
JunD	SydhK562JundIggrab			
KAP1	SydhU2osKap1Ucd			
Max	SydhH1hescMaxUcd			
MEF2C	**HaibGm12878Mef2csc13268V0416101**			
MTA3	HaibGm12878Mta3sc81325V0422111			
Mxi1	SydhGm12878Mxi1Iggmus			
Mxi1	SydhHepg2Mxi1			
Mxi1	SydhK562Mxi1af4185Iggrab			
MYBL2	HaibHepg2Mybl2sc81192V0422111			

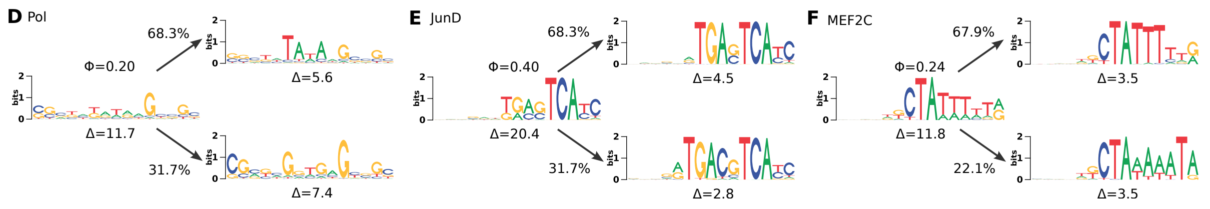
NANOG	HaibH1hescNanogsc33759V0416102			
NFATC1	HaibGm12878Nfatc1sc17834V0422111			
NFIC	HaibHepg2Nficsc81335V0422111			
NR2F2	HaibK562Nr2f2sc271940V0422111			
NRSF	HaibA549NrsfV0422111Etoh02			
NRSF	HaibGm12878NrsfPcr1x			
NRSF	HaibH1hescNrsfV0416102			
NRSF	HaibHepg2NrsfPcr2x			
NRSF	HaibHepg2NrsfV0416101			
NRSF	HaibK562NrsfV0416102			
NRSF	HaibPanc1NrsfPcr2x			
NRSF	HaibPfsk1NrsfPcr2x			
NRSF	HaibSknshNrsfPcr2x			
NRSF	HaibSknshNrsfV0416101			
NRSF	HaibU87NrsfPcr2x			
p300	HaibGm12878P300Pcr1x			
p300	HaibH1hescP300V0416102			
p300	HaibHepg2P300V0416101			
p300	SydhGm12878P300b			
p300	SydhHelas3P300sc584sc584Iggrab			
PAX5-C20	HaibGm12878Pax5c20Pcr1x			
PAX5-C20	HaibGm12891Pax5c20V0416101			
PAX5-N19	HaibGm12878Pax5n19Pcr1x			
Pbx3	HaibGm12878Pbx3Pcr1x			
PLU1	BroadK562Plu1			
Pol2	**HaibHepg2Pol2Pcr2x**			
Pol2	SydhGm12878Pol2			
Pol2	SydhGm12892Pol2Iggmus			
Pol2	SydhGm18505Pol2Iggmus			
Pol2	SydhGm18951Pol2Iggmus			
Pol2	SydhHuvecPol2			
Pol2	SydhK562Pol2			
Pol2	SydhK562Pol2Ifng6h			
Pol2	UtaGlioblaPol2			
Pol2	UtaH1hescPol2			
Pol2	UtaHuvecPol2			
Pol2	UtaMcf7Pol2Serumstim			
Pol2-4H8	HaibH1hescPol24h8V0416102			
Pol2-4H8	HaibHuvecPol24h8V0416101			
Pol2-4H8	HaibSknmcPol24h8V0416101			
Pol2-4H8	HaibU87Pol24h8V0416101			
Pol2(b)	BroadNhekPol2b			
Pol2(phosphoS2)	SydhK562Pol2s2Iggrab			
RFX5	SydhGm12878Rfx5200401194Iggmus			
RFX5	SydhH1hescRfx5200401194Iggrab			

RFX5	SydhK562Rfx5Iggrab			
SIN3A	SydhGm12878Sin3anb6001263Iggmus			
SIN3A	SydhH1hescSin3anb6001263Iggrab			
Sin3Ak-20	HaibA549Sin3ak20V0422111Etoh02			
Sin3Ak-20	HaibH1hescSin3ak20Pcr1x			
SIX5	HaibGm12878Six5Pcr1x			
SP1	HaibHepg2Sp1Pcr1x			
SRF	**HaibK562SrfV0416101**			
SRF	HaibH1hescSrfPcr1x			
SRF	HaibHepg2SrfV0416101			
STAT1	SydhHelas3Stat1Ifng30			
STAT1	SydhK562Stat1Ifna30			
STAT2	SydhK562Stat2Ifna30			
STAT5A	HaibGm12878Stat5asc74442V0422111			
STAT5A	HaibK562Stat5asc74442V0422111			
SUZ12	SydhH1hescSuz12Ucd			
TAF1	HaibGm12878Taf1Pcr1x			
TAF1	HaibH1hescTaf1V0416102			
TAF1	HaibSknshTaf1V0416101			
TBLR1	SydhGm12878Tblr1ab24550Iggmus			
TBLR1	SydhK562Tblr1ab24550Iggrab			
TBP	SydhGm12878TbpIggmus			
TCF12	HaibA549Tcf12V0422111Etoh02			
TEAD4	HaibHepg2Tead4sc101184V0422111			
TFIIIC	**SydhK562Tf3c110**			
TFIIIC	SydhHelas3Tf3c110			
THAP1	HaibK562Thap1sc98174V0416101			
TR4	SydhHelas3Tr4			
ZBTB33	HaibGm12878Zbtb33Pcr1x			
ZNF274	SydhNt2d1Znf274Ucd			

TF name	ID	Slim-Dimont prediction	IMD cluster 1	IMD cluster 2	IMD cluster 3
ARID3A	SydhK562Arid3asc8821Iggrab				
ATF3	HaibHepg2Atf3V0416101				
BCL3	HaibA549Bcl3V0422111Etoh02				
BRF2	SydhHelas3Brf2				
c-Jun	SydhHepg2CjunIggrab				
c-Jun	SydhK562CjunIfng6h				
COREST	SydhHelas3Corestsc30189Iggrab				
ELK1	SydhGm12878Elk112771Iggmus				
ETS1	HaibA549Ets1V0422111Etoh02				
ETS1	HaibGm12878Ets1Pcr1x				
FOXP2	HaibSknmcFoxp2Pcr2x				
HDAC6	BroadK562Hdac6a301341a				

Ini1	SydhHelas3Ini1Iggmus	
IRF1	SydhK562Irf1Ifng30	
MAZ	SydhHelas3Mazab85725Iggrab	
Mxi1	SydhH1hescMxi1Iggrab	
NRSF	HaibHelas3NrsfPcr1x	
p300	SydhHepg2P300sc582Iggrab	
Pol2	HaibHelas3Pol2Pcr1x	
Pol2	SydhGm18526Pol2Iggmus	
Pol2	SydhHepg2Pol2Forskln	
Pol2	SydhK562Pol2Ifng30	
Pol2	SydhMcf10aesPol2Tam	
Pol2	UtaProgfibPol2	
RFX5	SydhHepg2Rfx5200401194Iggrab	
Sin3Ak-20	HaibPfsk1Sin3ak20V0416101	
SIRT6	SydhK562Sirt6	
SP1	HaibH1hescSp1Pcr1x	
SP1	HaibK562Sp1Pcr1x	
SP2	HaibHepg2Sp2V0422111	
SP2	HaibK562Sp2sc643V0416102	
SP4	HaibH1hescSp4v20V0422111	
SREBP1	SydhHepg2Srebp1Insln	
SRF	HaibGm12878SrfPcr2x	
TAF1	HaibK562Taf1V0416101	
TBLR1	SydhK562Tblr1nb600270Iggrab	
TBP	SydhHepg2TbpIggrab	
ZZZ3	SydhGm12878Zzz3	

Supplement 4.6: Additional intermixture types



There are three further noteworthy intermixture types not mentioned in the main manuscript.

Some data sets stem from ChIP-experiments against non-specific TFs or even the RNA polymerase for different cell lines and conditions. In such cases, we may not expect to find an overrepresented motif at all, even though motif discovery algorithms often report one. For Pol, we obtain indeed a fairly uninformative sequence logo. IMD separates a weak TATA motif from the remaining sequences, although the decision is close (Φ is only marginally above T).

For JunD the two clusters are variants of the same motif that differ by inclusion/exclusion of a central cytosine. Such a variable spacer has been used as a motivation for incorporating intra-motif dependencies into motif discovery [18]. Provided JunD indeed binds to both motifs alike, IMD makes here an incorrect prediction. These cases are rare, all but one are either Jun or JunD data sets.

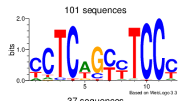
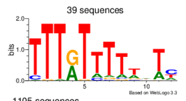
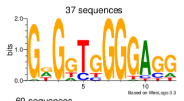
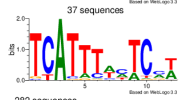
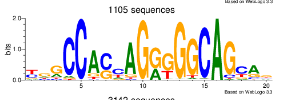
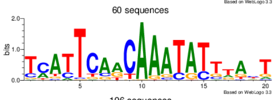
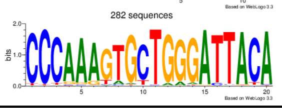
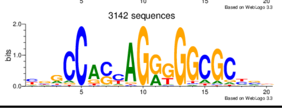
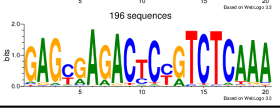
There are also some cases in which the predicted binding sites are an intermix of similar sites in different shifts or strand orientations, such as MEF2C. Such purely computational artifacts can be caused by imperfect motif discovery (local optima) or by inappropriate input parameters. One example for the latter are all 12 data sets for NRSF (see table above). This TF can recognize an exceptionally long motif of more than 20bp that can not be fully captured with motif width 15.

Supplement 4.7: Disentangler applied to output of DIVERSITY

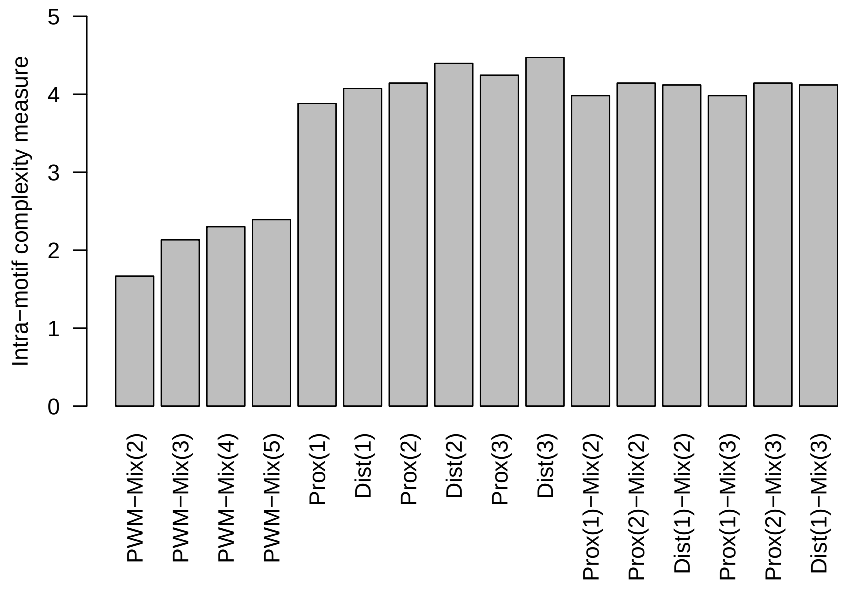
This section shows additional figures from the application of Disentangler on DIVERSITY output that were left out of the main manuscript for space constraints.

Supplement 4.7.1: CTCF

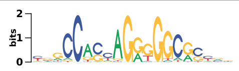
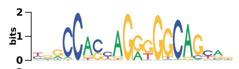
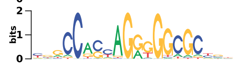
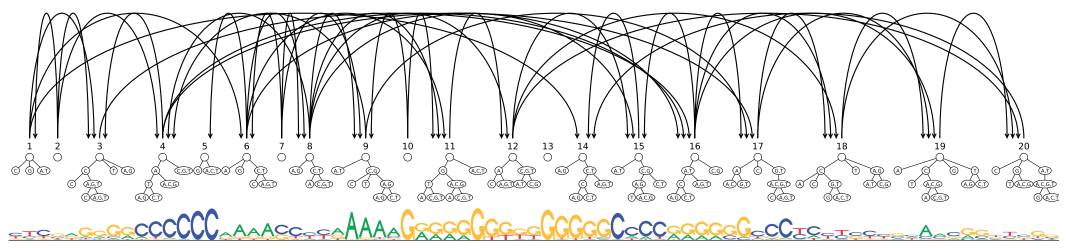
For the ChIP-seq data set of CTCF (K562 cell line, ID=UwK562Ctcf), the following nine modes are reported to be optimal according to DIVERSITY:

Mode	Sequence logo	Mode	Sequence logo	Mode	Sequence logo
1		4		7	
2		5		8	
3		6		9	

After joining the binding sites of mode 5 and 6, IMD reveals that all these sites are bound by the same factor ($\Phi = 0.079$). Applying MCA yields the following assessment of intra-motif complexity:



Visualizations of winning model (Dist(3)), baseline PWM, and two-component PWM mixture:

Model	Visualization
PWM	
PWM-Mix(2)	 $\alpha_1 = 0.3$  $\alpha_2 = 0.7$
Dist(3)	

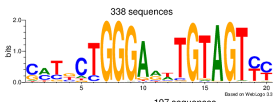
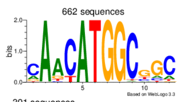
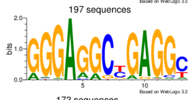
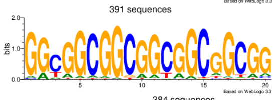
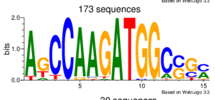
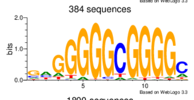
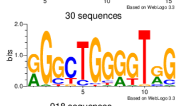
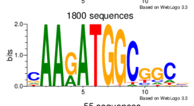
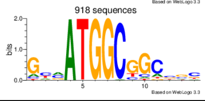
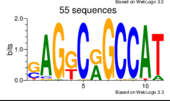
For studying the prediction performance in different cell lines, we use all ENCODE ChIP-seq data sets from the Uniform TFBS track that are annotated with Antibody=CTCF and Lab=UW.

The performance measure is the true positive rate under a false positive rate of 0.01:

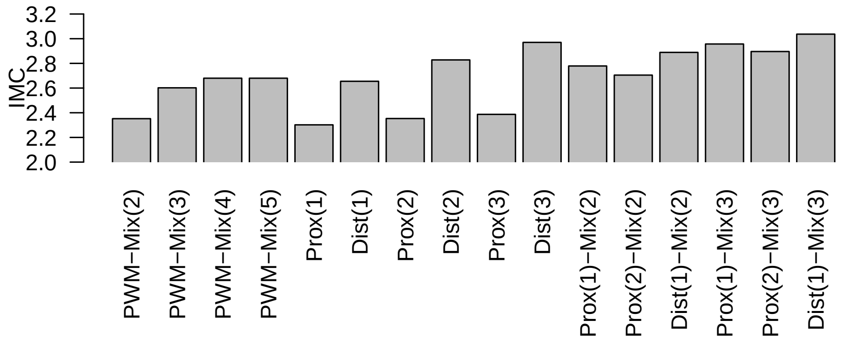
ID	PWM	PWM-Mix(5)	Dist(1)-Mix(3)
UwK562Ctcf	0.624	0.647	0.683
UwGm12878Ctcf	0.653	0.680	0.764
UwHepg2Ctcf	0.593	0.621	0.650
UwHelas3Ctcf	0.632	0.657	0.759
UwA549Ctcf	0.696	0.718	0.846
UwMcf7Ctcf	0.635	0.668	0.755
UwNhekCtcf	0.634	0.662	0.828
UwHuvecCtcf	0.638	0.660	0.614
UwNhlfCtcf	0.695	0.719	0.814
UwHvmfCtcf	0.675	0.695	0.632
UwAg04449Ctcf	0.646	0.677	0.674
UwAg04450Ctcf	0.699	0.721	0.718
UwAg09309Ctcf	0.676	0.705	0.761
UwAg09319Ctcf	0.692	0.713	0.752
UwAg10803Ctcf	0.734	0.742	0.777
UwAoafCtcf	0.655	0.678	0.738
UwBe2cCtcf	0.714	0.736	0.789
UwBjCtcf	0.669	0.692	0.719
UwCaco2Ctcf	0.640	0.659	0.809
UwGm06990Ctcf	0.662	0.685	0.642
UwGm12801Ctcf	0.691	0.715	0.650
UwGm12864Ctcf	0.658	0.679	0.804
UwGm12865Ctcf	0.641	0.673	0.718
UwGm12872Ctcf	0.658	0.679	0.617
UwGm12873Ctcf	0.663	0.691	0.812
UwGm12874Ctcf	0.624	0.650	0.704
UwGm12875Ctcf	0.614	0.649	0.727
UwHacCtcf	0.670	0.695	0.761
UwHaspCtcf	0.673	0.695	0.457
UwHbmecCtcf	0.685	0.705	0.731
UwHcfaaCtcf	0.660	0.676	0.767
UwHcmCtcf	0.684	0.705	0.667
UwHcpeCtcf	0.716	0.740	0.822
UwHct116Ctcf	0.663	0.692	0.725
UwHeeCtcf	0.708	0.724	0.771
UwHek293Ctcf	0.647	0.678	0.788
UwHffCtcf	0.681	0.705	0.628
UwHffmycCtcf	0.720	0.736	0.785
UwHl60Ctcf	0.658	0.682	0.620
UwHmecCtcf	0.664	0.696	0.838
UwHmfCtcf	0.730	0.740	0.681
UwHpafCtcf	0.713	0.726	0.726
UwHpfCtcf	0.677	0.691	0.807
UwHreCtcf	0.669	0.696	0.836
UwHrpeCtcf	0.692	0.711	0.640
UwNb4Ctcf	0.700	0.730	0.758
UwNhdfneoCtcf	0.718	0.732	0.797
UwRptecCtcf	0.737	0.761	0.782
UwSaecCtcf	0.635	0.657	0.661
UwSknshraCtcf	0.642	0.667	0.832
UwWerirb1Ctcf	0.569	0.592	0.667
UwWi38Ctcf	0.638	0.660	0.659

Supplement 4.7.2: YY1

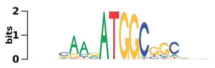
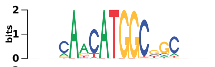
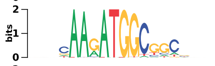
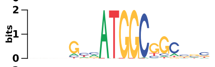
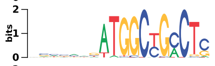
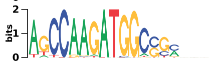
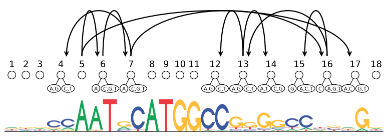
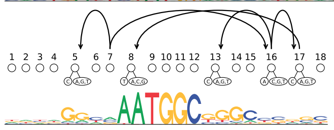
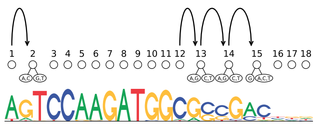
For the ChIP-seq data set of YY1 (K562 cell line, ID=SydhK562Yy1Ucd), the following nine modes are reported to be optimal according to DIVERSITY:

Mode	Sequence logo	Mode	Sequence logo
1		6	
2		7	
3		8	
4		9	
5		10	

After joining the binding sites of mode 3, 5, 6, 9, and 10 into one data set, IMD revals that all these sites are bound by the same factor ($\Phi = 0.13$).



Visualizations of winning model, baseline PWM, and five-component PWM mixture:

Model	Visualization	
PWM		
PWM-Mix(5)		$\alpha_1 = 0.154$
		$\alpha_1 = 0.531$
		$\alpha_1 = 0.260$
		$\alpha_1 = 0.011$
		$\alpha_1 = 0.044$
Dist(1)-Mix(3)		$\alpha_1 = 0.697$
		$\alpha_1 = 0.270$
		$\alpha_1 = 0.033$

For studying the prediction performance in different cell lines, we use all ENCODE ChIP-seq data sets from the Uniform TFBS track that are annotated with Antibody=YY1.

Due to the small number of data sets, we do not focus on a single lab. The performance measure is the true positive rate under a false positive rate of 0.01:

Cell line	ID	PWM	PWM-Mix(5)	Dist(1)-Mix(3)
K562	SydhK562Yy1Ucd	0.355	0.404	0.458
GM12892	HaibGm12892Yy1V0416101	0.310	0.330	0.441
K562	HaibK562Yy1V0416102	0.328	0.363	0.427
Gm12878	SydhGm12878Yy1	0.302	0.328	0.330
Nt2d1	SydhNt2d1Yy1Ucd	0.316	0.366	0.461

Supplement 4.8: Stability analysis

Another question worthwhile to investigate is how stable the solutions of IMD and MCA are for individual data sets. To address this issue, we pick nine sets of aligned TFBS that are discussed in the manuscript/supplement in more detail:

- DUX4 from JASPAR (manuscript Figure 5)
- the six intermixture examples: c-Fos, SRF, and TFIIIC (manuscript Figure 7), Pol2, JunD, and MEF2C (Supplement 4.6)
- the two merged datasets from DIVERSITY output: CTCF and YY1 (manuscript Figure 9)

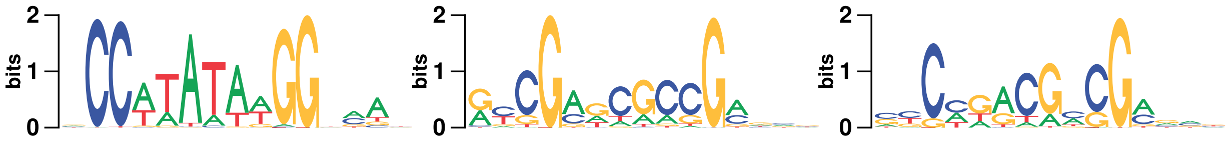
For each of these cases, we carry out bootstrapping by sampling (with replacement) ten data sets of same size from the original data set. For each of the resulting 10×9 resamples, we apply both subtools and discuss the results below.

Supplement 4.8.1: Stability of IMD

We observe that IMD is very stable and yields the same intermixture number as the original data set for the majority of cases (90%):

Resample	DUX4	c-Fos	SRF	TFIIIC	Pol2	JunD	MEF2C	CTCF	YY1
1	1	2	2	2	2	2	2	1	1
2	1	2	3	2	2	2	2	1	1
3	1	2	3	2	2	2	2	1	1
4	1	2	3	1	2	2	2	1	1
5	1	2	3	2	2	2	2	1	1
6	1	2	3	2	4	2	2	1	1
7	1	2	3	2	2	2	2	1	1
8	1	2	2	1	2	2	2	1	1
9	1	2	2	2	2	2	2	1	1
10	1	2	3	2	2	2	2	1	1
original	1	2	2	2	2	2	2	1	1

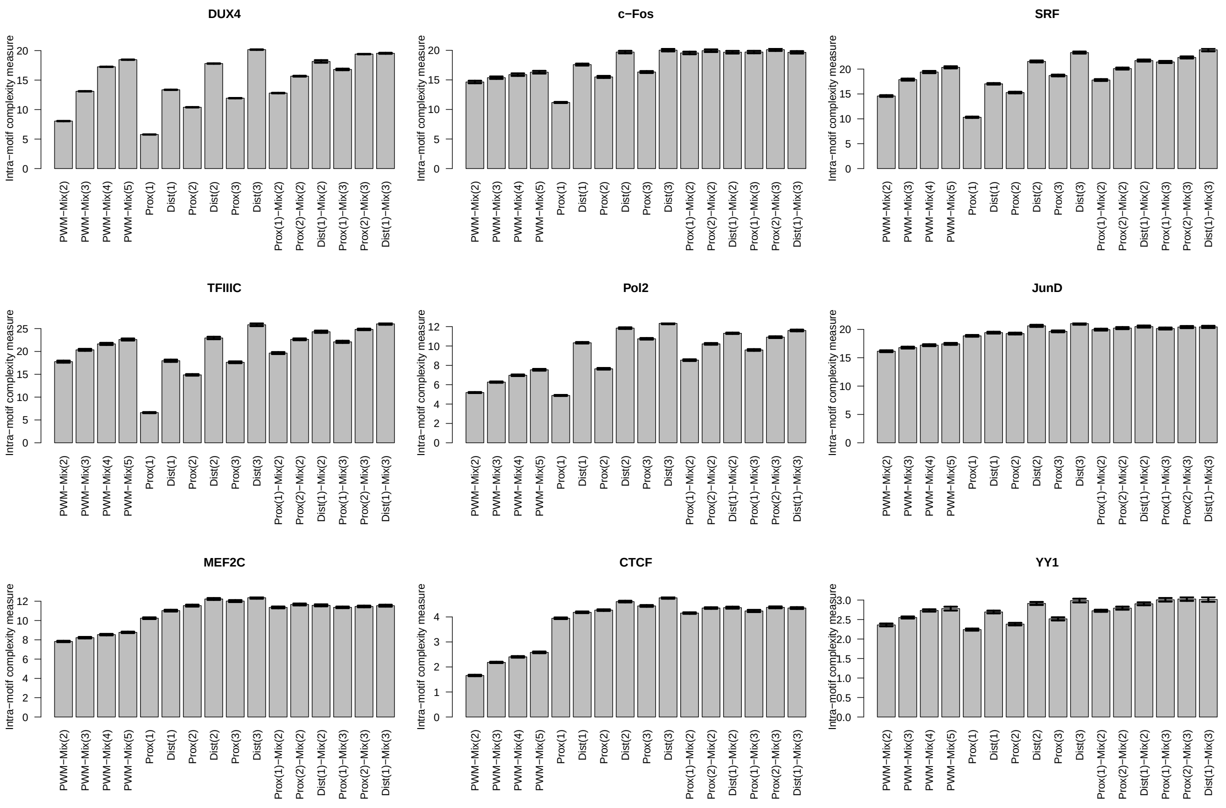
One exception is SRF, where in 7 of 10 resamples, the second cluster which contains putative nonfunctional sequences, is divided again in two different clusters. For example, SRF resample 2 yields the following three IMD-clusters:



Such a behavior is uncritical, as the purpose of IMD is to remove possible artifacts from the motif of interest, and that is accomplished for SRF in every case. For TFIIIC, we observe that no intermixture is detected in two of ten resamples, which can be attributed to the fact that the intermixture measure of the original data set is only marginally above the threshold of 0.19 (cf. manuscript Figure 7C). Small fluctuations in the composition of the data set may thus lead to a different decision, but except for this close case, intermixtures (or absence thereof) is always recognized as such.

Supplement 4.8.2: Stability of MCA

For MCA, we plot the intra-motif complexity for each TF and each model, averaged over the ten re-samples. The error bars on top of the barplots indicate double standard error. We observe that the assessment of different models by MCA is stable for each data set as the standard errors are small:



Supplement 5: Disentangler software

Disentangler is implemented in Java using the Jstacs library [25] and requires an existent Java installation (8u74 or later).

Supplement 5.1: Subtools

The software contains the two subtools of Disentangler, following the terminology of the manuscript they are called “Intermixture detection” and “Motif complexity analysis”. In addition, there is a tool “SequenceScan” that can be used to search for motif hits within target sequences based on models that are returned by “Motif complexity analysis”.

All tools expect a set of aligned, gapless, TFBS of the same length as input. If the content of the input file starts with '>', it is interpreted as FastA file. Otherwise it is interpreted as plain text, where every line contains a single sequence. The input expects upper- and lower case letters of the standard DNA alphabet A,C,G,T. If other symbols from the IUPAC code (such as N) are encountered, they are replaced by a random sample from the distribution of A,C,G,T in the data set.

Supplement 5.1.1: Intermixture detection

If ”JSD weights” is disabled, the intermixture measure is computed on a non-weighted Jenson-Shannon divergence. Option included for experimental purposes, for practical use keeping the default is strongly recommended. Smaller values for ”Restarts”, ”Time limit” and ”Termination threshold” can speed up every recursive step, which can be beneficial for testing purposes, but they may affect quality of the results. The tool returns a text file with the intermixture number and all clusters produced by IMD as text files of the binding sites and sequence logos of the mononucleotide statistics. The values for the intermixture measure at each recursive step can be found in the protocol.

Supplement 5.1.2: Motif complexity analysis

The tool allows to learn of proximal/distal dependency models and mixtures thereof. Note: Learning distal dependence models of order greater than one can be very time- and memory consuming if the input sequences are long. It is not recommended for motifs of length greater than 20. The tool returns a text-file containing the intra-motif complexity measure of the data set, a visualization of the learned model, and a storable (.xml) file that can be used as input to “Sequence scan”. The mixture weights and model complexities of each component can be found in the protocol.

Supplement 5.1.3: Sequence scan

This tool is a variant of the InMoDe ScanApp [7], with increased support for different types of models, that is, mixture models and distal dependence models. ”Input model” needs to be an model file (in .xml format) produced by “Motif complexity analysis”. The ”FPR” pertains here to the number of sequence that have at least one hit. The tool returns a list with coordinates of motif hits as well as the extracted binding sites.

Supplement 5.2: User interfaces

There are two versions of a runnable .jar that differ only in the user interface.

Supplement 5.2.1: GUI

The graphical user interface (GUI) `DisentanglerGUI.jar` allows us to run Disentangler in an interactive mode locally on a desktop computer. It is sufficient to explore the functionality for testing purposes and to perform single analyses. For larger studies involving more than a few data sets, it is recommended to use the command line interface (next section).

The GUI can be started by calling

```
java -jar DisentanglerGUI.jar
```

or by double-clicking on the .jar-file.

Supplement 5.2.2: CLI

The command line interface (CLI) is contained in `DisentanglerCLI.jar`. An overview over is obtained by calling

```
java -jar DisentanglerCLI.jar
```

which yields

```
Available tools:

        imd - Intermixture detection
        mca - Motif complexity analysis
        scan - Sequence scan

Syntax: java -jar DisentanglerCLI.jar <toolname> [<parameter=value> ...]

Further info about the tools is given with
        java -jar DisentanglerCLI.jar <toolname> info

Tool parameters are listed with
        java -jar DisentanglerCLI.jar <toolname>
```

Calling DisentanglerCLI.jar with the corresponding tool name, e.g.,

```
java -jar DisentanglerCLI.jar imd
```

yields a list of input parameters (as the mandatory input file is missing):

```
Parameters of tool "Intermixture detection" (imd, version: 1.0):
i - Input data (The file containing the input data.)    = null
  it - Intermixture threshold (Threshold on intermixture measure.,
valid range = [0.0, 1.0], default = 0.19)   = 0.19
w - Weighted JSD (If enabled, JSD is computed as weighted average.,
  default = true) = true
n - Number of restarts (The total number of restarts to be carried out.,
  valid range = [1, 10000], default = 100)     = 100
t - Time limit (Upper time limit for termination of an individual run in seconds.,
  valid range = [0.01, 100000.0], default = 60.0)   = 60.0
tt - Termination threshold (Stop FAB algorithm at this FIC-difference.,
  valid range = [1.0E-10, 1.0], default = 1.0E-6)      = 1.0E-6
outdir - The output directory, defaults to the current working directory (.)    = .

At least one parameter has not been set (correctly).
```

The command

```
java -jar DisentanglerCLI.jar imd i=test.txt
```

runs IMD with default threshold on the given input file.

References

[1] G. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20:197–243, 1995.

[2] G.D. Stormo, T.D Schneider, and L.M. Gold. Characterization of translational initiation sites in E.coli. *Nucleic Acids Res.*, 10(2):2971–2996, 1982.

[3] M.Q. Zhang and T.G. Marr. A weights array method for splicing signals analysis. *Comput. Appl. Biosci.*, 9:499–509, 1993.

[4] Jorma Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664, 1983.

[5] P.-Y. Bourguignon and D. Robelin. Modèles de Markov parcimonieux: sélection de modele et estimation. In *Proceedings of JOBIM*, 2004.

[6] R. Eggeling and M. Koivisto. Pruning rules for learning parsimonious context trees. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

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

[8] I. Ben-Gal, A. Shani, A. Gohr, J. Grau, S. Arviv, A. Shmilovici, S. Posch, and I. Grosse. Identification of transcription factor binding sites with variable-order Bayesian networks. *Bioinformatics*, 21:2657–2666, 2005.

[9] Y.J. Chu and T.H. Liu. On the shortest arborescence of a directed graph. *Scienca Sinica*, 14:1396–1400, 1965.

[10] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, 71B:233–240, 1967.

[11] T. Silander and P. Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[12] L. Narlikar. MuMoD: a Bayesian approach to detect multiple modes of protein-DNA binding from genome-wide ChIP data. *Nucleic Acids Res.*, 41(1):21–32, 2013.

[13] R. Fujimaki and S. Morinaga. Factorized asymptotic Bayesian inference for mixture modeling. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

[14] G.E. Schwarz. Estimating the dimension of a model. *Ann. Stat.*, 2:461–464, 1978.

[15] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39(1):1–38, 1977.

[16] S. Mitra and L. Narlikar. No Promoter Left Behind (NPLB): learn de novo promoter architectures from genome-wide transcription start sites. *Bioinformatics*, 32(5):779–781, 2016.

[17] S. Mitra, A. Biswas, and L. Narlikar. DIVERSITY in binding, regulation, and evolution revealed from high-throughput ChIP. *PLOS Comput. Biol.*, 14(4):e1006090, 2018.

[18] J. Keilwagen and J. Grau. Varying levels of complexity in transcription factor binding motifs. *Nucleic Acids Res.*, 43(18):e119, 2015.

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

[20] A. Mathelier, O. Fornes, D.J. Arenillas, C. Chen, G. Denay, J. Lee, W. Shi, C. Shyr, G. Tan, et al. JASPAR 2016: a major expansion and update of the open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, 44(D1):D110, 2016.

[21] I. Yevshin, R. Sharipov, T. Valeev, A. Kel, and F. Kolpakov. GTRD: a database of transcription factor binding sites identified by ChIP-seq experiments. *Nucleic Acids Res.*, 45:D61–D67, 2017.

[22] E. Wingender, T. Schoeps, M. Haubrock, and J. Dönitz. TFClass: a classification of human transcription factors and their rodent orthologs. *Nucleic Acids Res.*, 43:D97–D102, 2015.

[23] The ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):714657–74, 2012.

[24] M. Pachkov, P.J. Balwierz, P. Arnold, E. Ozonov, and E. van Nimwegen. SwissRegulon, a database of genome-wide annotations of regulatory sites: recent updates. *Nucleic Acids Res.*, 41:D214–20, 2013.

[25] J. Grau, J. Keilwagen, A. Gohr, B. Haldemann, S. Posch, and I. Grosse. Jstacs: A Java framework for statistical analysis and classification of biological sequences. *J. Mach. Learn. Res.*, 13:1967–1971, 2012.