Supplemental Information

P-value based regulatory motif discovery using positional weight matrices

Holger Hartmann, Eckhart W. Guthöhrlein,
Matthias Siebert, Sebastian Luehr, and Johannes Söding

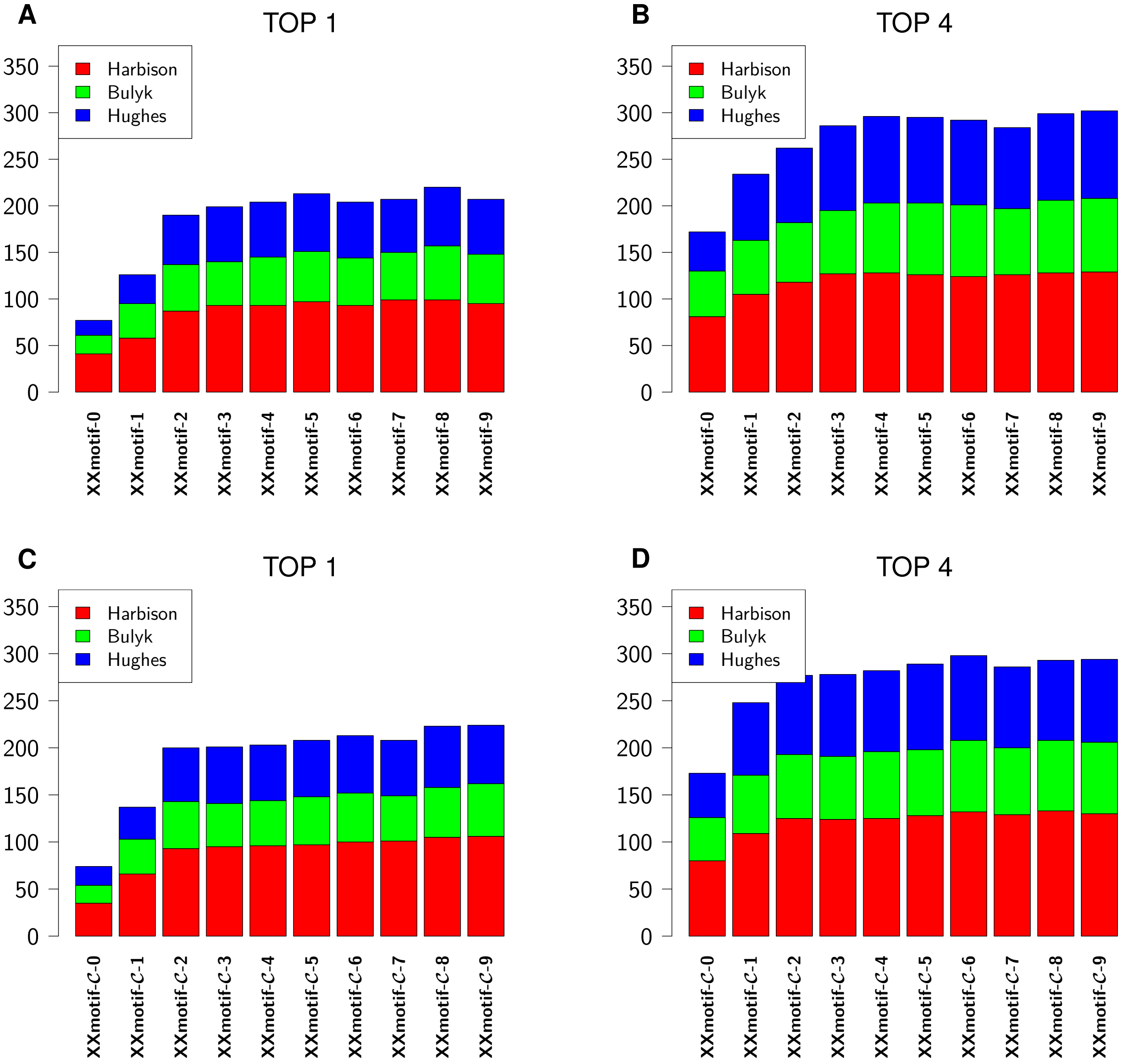
September 13, 2012

Contents

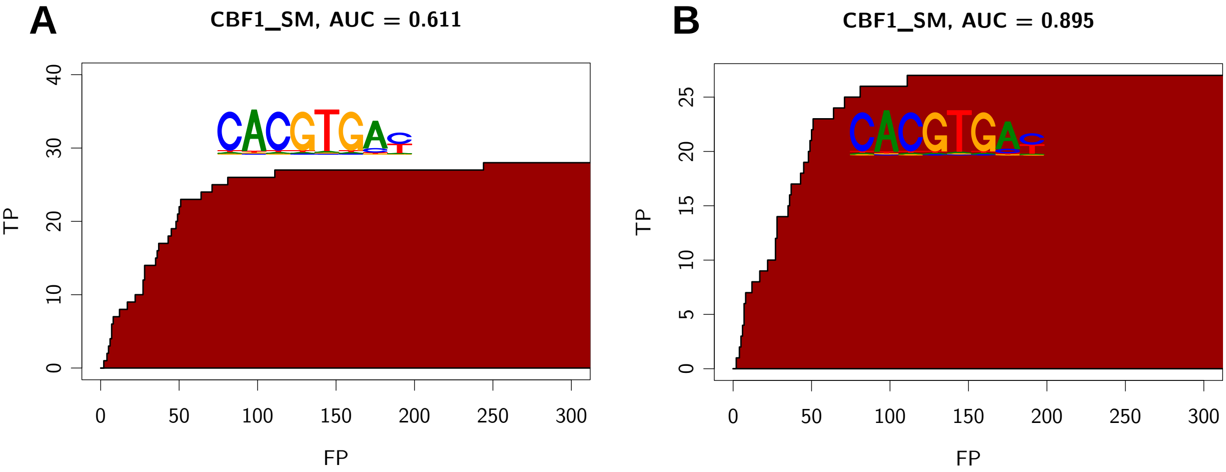
List of Figures

List of Tables

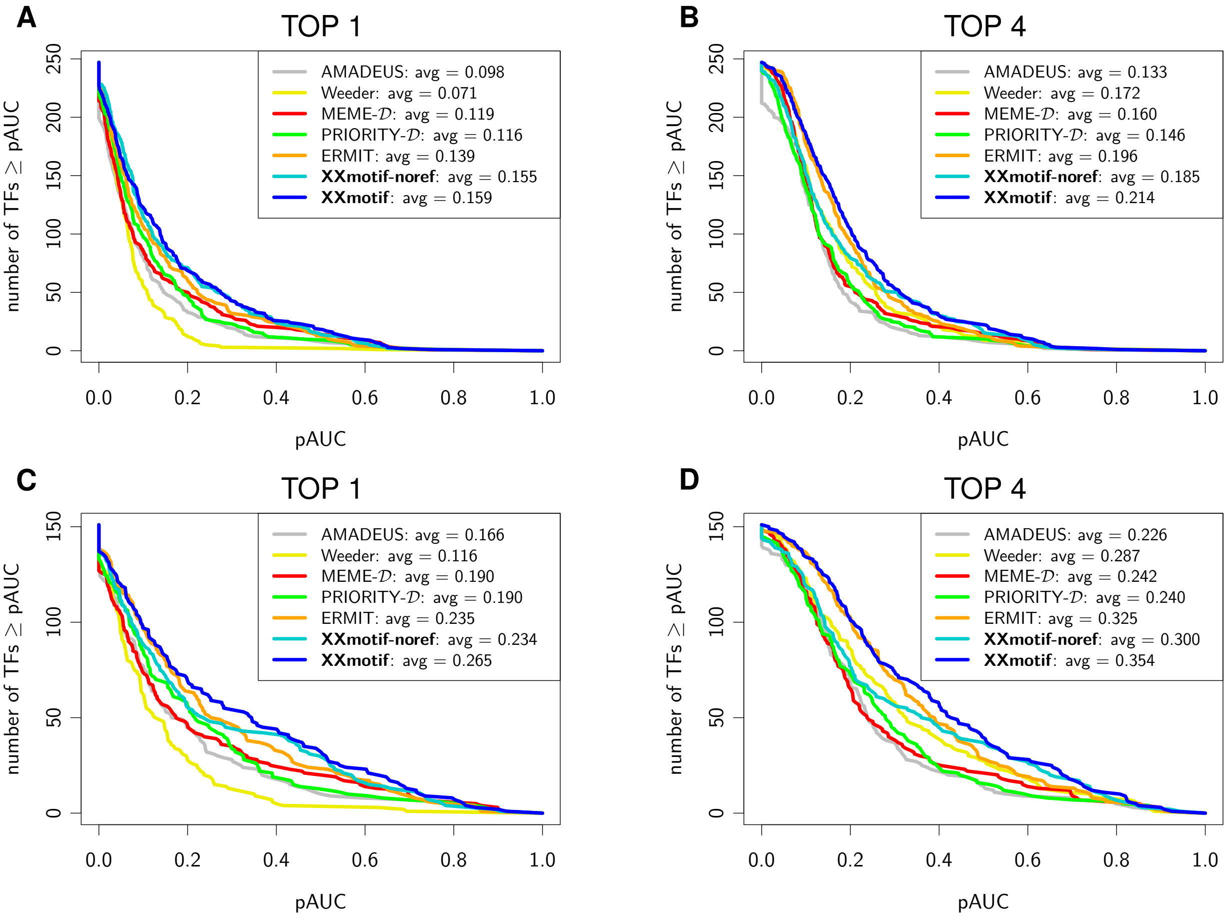
Part I.
Supplemental Figures



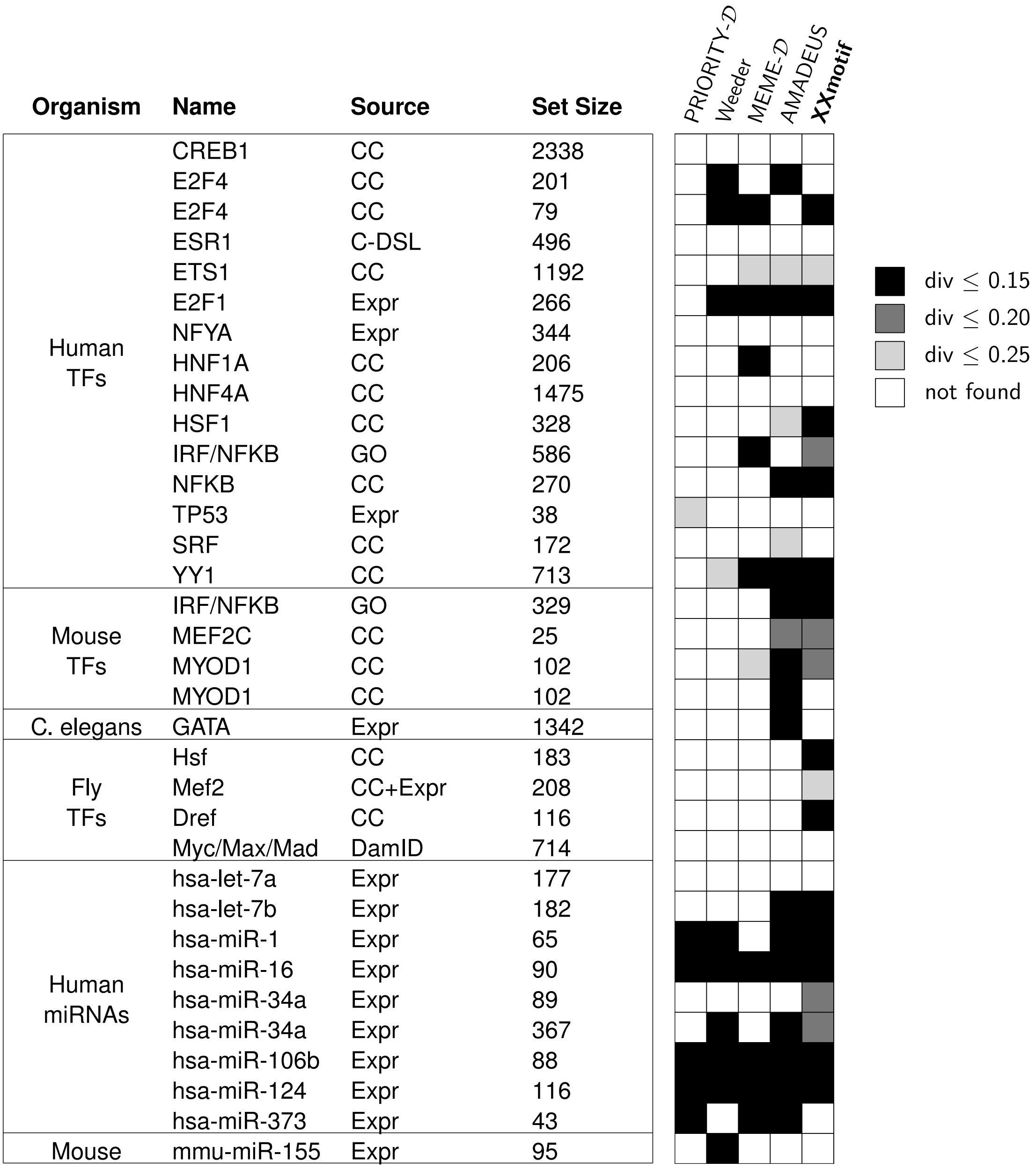
Supplementary Figure 1: Number of correctly identified motifs of XXmotif on the ChIP-chip data set of Harbison et al. (2004), depending on the order of the background model ranging from zero to nine. Three experimental reference sets are used to judge the correctness of motifs (red, green, blue). (A) Top 1 prediction without conservation, (B) Top 4 prediction without conservation, (C) Top 1 prediction with conservation, (D) Top 4 prediction with conservation.



Supplementary Figure 2: ROC curve of the PWM found by XXmotif in the CBF1_SM ChIP-chip data set and the corresponding partial area under curve (pAUC) value calculated from it. (A) All intergenic regions having a ChIP-chip P-value < 0.001 are listed as true positives (TPs). (B) Only those TPs are listed that have a binding site in the region that matches to at least one of the CBF1 PWMs from the "Bulyk", "Hughes", or "Harbison set".



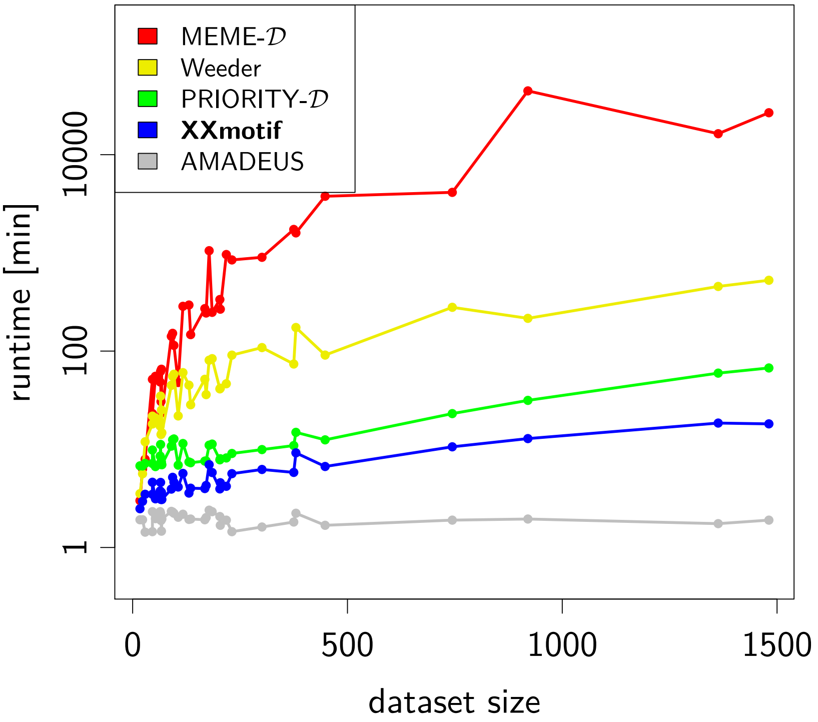
Supplementary Figure 3: PWM quality assessment on yeast ChIP-chip data from Harbison et al. (2004). The curves quantify how well the scores of the reported PWMs can predict the ChIP enrichment of the sequences. Each PWM is used to rank the intergenic regions by their maximum PWM score. For each predicted PWM, a receiver operator characteristic (ROC) curve with the number of correct predictions over the number of false predictions is computed, and the partial area under the ROC curve (pAUC) deduced from it. The pAUC is the fractional area under the ROC curve within the 5% best-ranked false predictions. For an ideal predictor, pAUC=1. The average pAUC scores are listed in the figure legends. (A, B) cumulative distribution of the pAUC over all 247 ChIP-chip datasets that had at least ten significantly enriched regions (P-value < 0.001). Regions with ChIP enrichment P-value < 0.001 are defined as correct predictions, all other regions as false predictions. (C, D) As in A, B but using only datasets that have at least five significantly ChIP-enriched regions with matches to the literature motif, and considering only sequences that contain a match to the literature motif.



Organism	Name	Source	Set Size	PRIORITY-D	Weeder	MEME-D	AMADEUS	XXmotif
Human TFs	CREB1	CC	2338	□	□	□	□	□
	E2F4	CC	201	□	■	□	■	□
	E2F4	CC	79	□	■	■	□	■
	ESR1	C-DSL	496	□	□	□	□	□
	ETS1	CC	1192	□	□	░	░	░
	E2F1	Expr	266	□	■	■	■	■
	NFYA	Expr	344	□	□	□	□	□
	HNF1A	CC	206	□	□	■	□	□
	HNF4A	CC	1475	□	□	□	□	□
	HSF1	CC	328	□	□	□	░	■
	IRF/NFKB	GO	586	□	□	■	□	▦
	NFKB	CC	270	□	□	□	■	■
	TP53	Expr	38	░	□	□	□	□
	SRF	CC	172	□	□	□	░	□
	YY1	CC	713	□	░	■	■	■
Mouse TFs	IRF/NFKB	GO	329	□	□	□	■	■
	MEF2C	CC	25	□	□	□	▦	▦
	MYOD1	CC	102	□	□	░	■	▦
	MYOD1	CC	102	□	□	□	■	□
C. elegans	GATA	Expr	1342	□	□	□	■	□
Fly TFs	Hsf	CC	183	□	□	□	□	■
	Mef2	CC+Expr	208	□	□	□	□	░
	Dref	CC	116	□	□	□	□	■
	Myc/Max/Mad	DamID	714	□	□	□	□	□
Human miRNAs	hsa-let-7a	Expr	177	□	□	□	□	□
	hsa-let-7b	Expr	182	□	□	□	■	■
	hsa-miR-1	Expr	65	■	■	□	■	■
	hsa-miR-16	Expr	90	■	■	■	■	■
	hsa-miR-34a	Expr	89	□	□	□	□	▦
	hsa-miR-34a	Expr	367	□	■	□	■	▦
	hsa-miR-106b	Expr	88	■	■	■	■	■
	hsa-miR-124	Expr	116	■	■	■	■	■
	hsa-miR-373	Expr	43	■	□	■	■	□
Mouse	mmu-miR-155	Expr	95	□	■	□	□	□

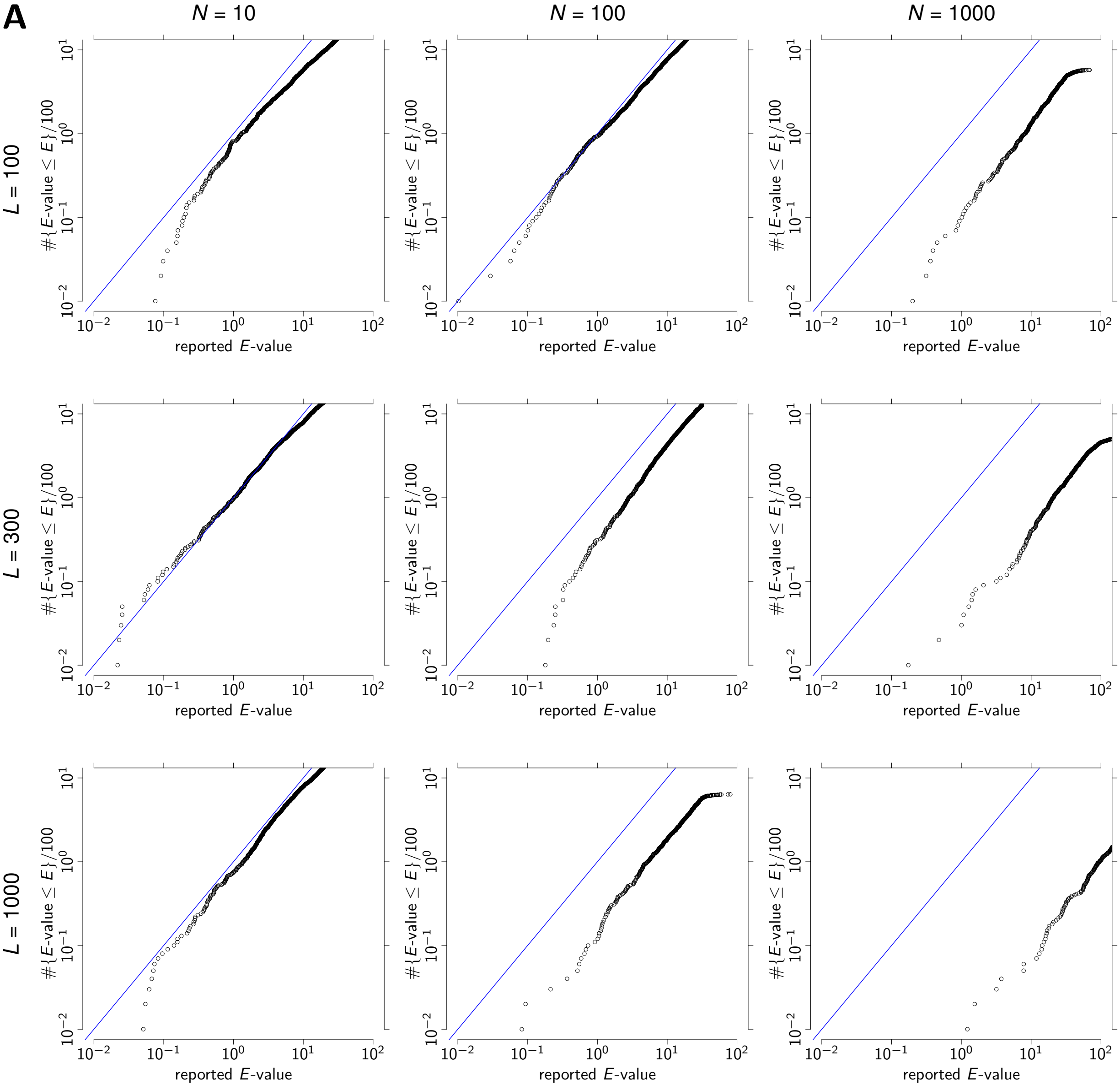
■ div ≤ 0.15
▦ div ≤ 0.20
░ div ≤ 0.25
□ not found

Supplementary Figure 4: Top 1 benchmark results on 24 target sets for transcription factors from human, mouse, worm, and fly, as well as 10 target sets of microRNAs from human and mouse from the metazoan target set compendium (Linhart et al., 2008). The plot is adapted from Linhart et al. (2008): The "Source" column indicates the experimental procedure or database from which the target set was derived: Gene expression microarrays (Expr), ChIP-chip (CC), ChIP-DSL (C-DSL), DamID (van Steensel et al., 2001), or Gene Ontology (GO) database (Ashburner et al., 2000). The black and gray boxes indicate the similarity of the predicted PWM to the reference motif in TRANSFAC of miRBase. Darker shades indicate closer similarity. "Set Size": number of sequences within the input set.

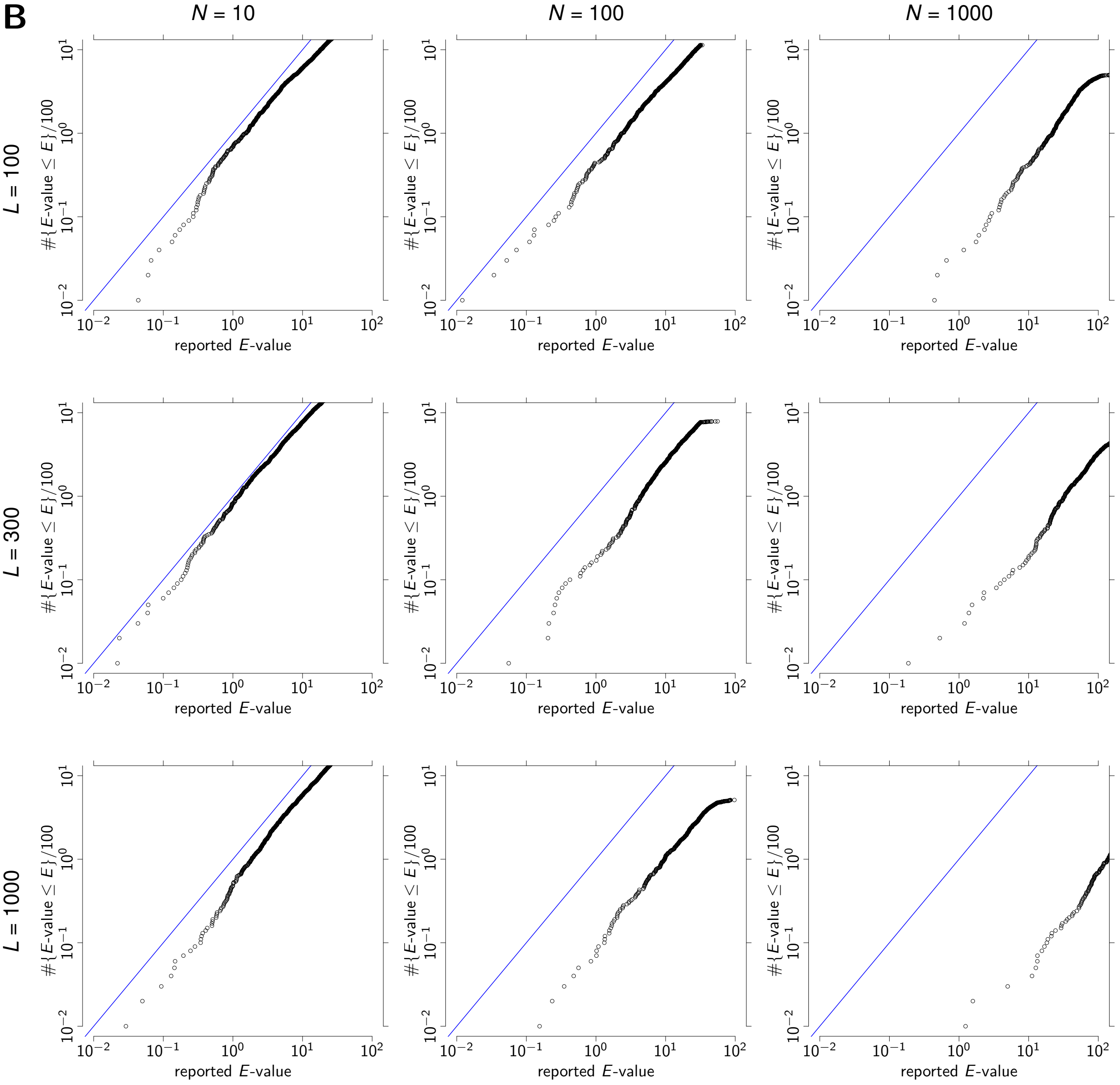


Supplementary Figure 5: Runtime of tested tools on the metazoan benchmark. All jobs were run on a single core Xeon 2.9 GHz CPU.

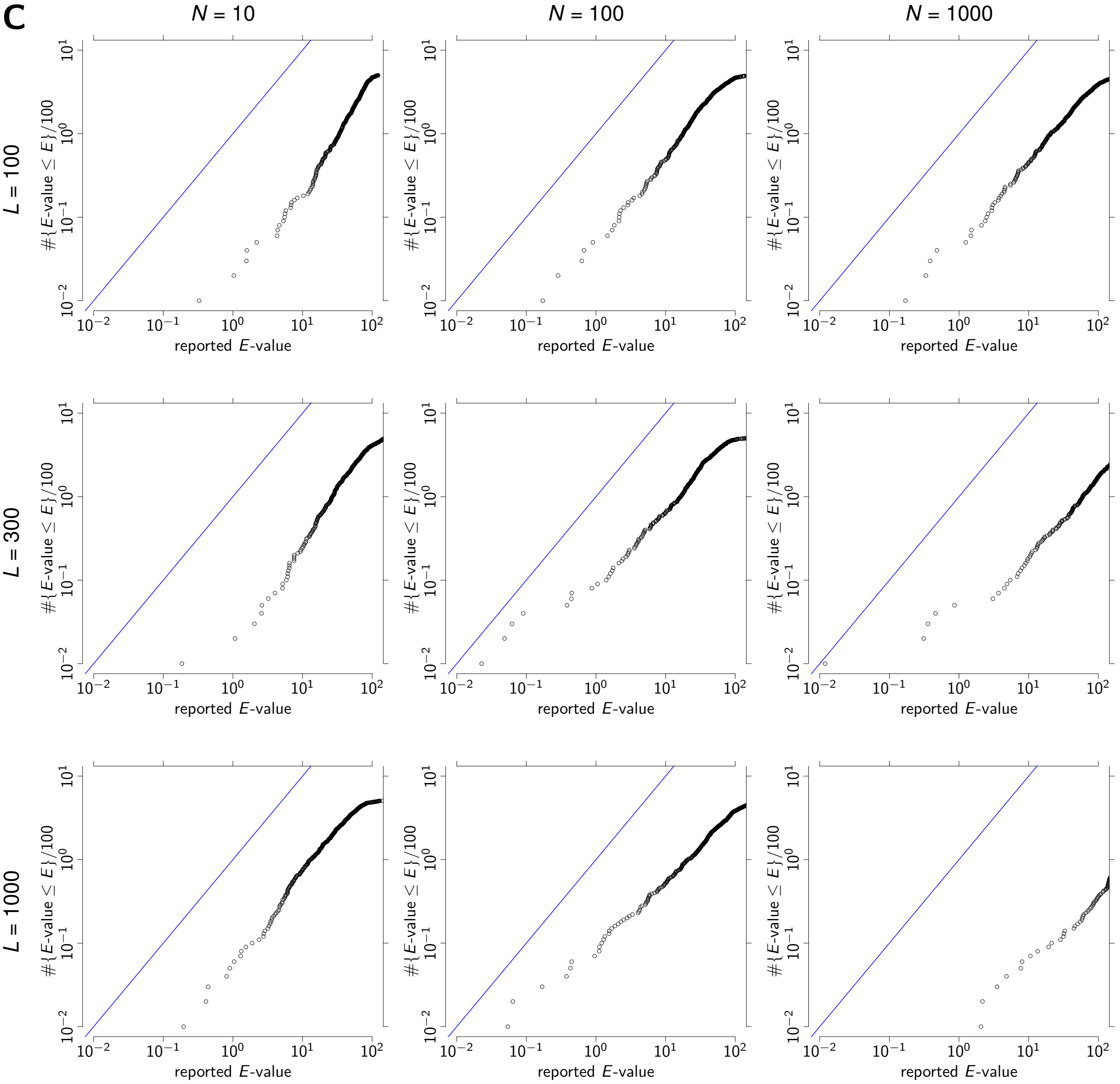
Second order background model, multiple occurrences per sequence.



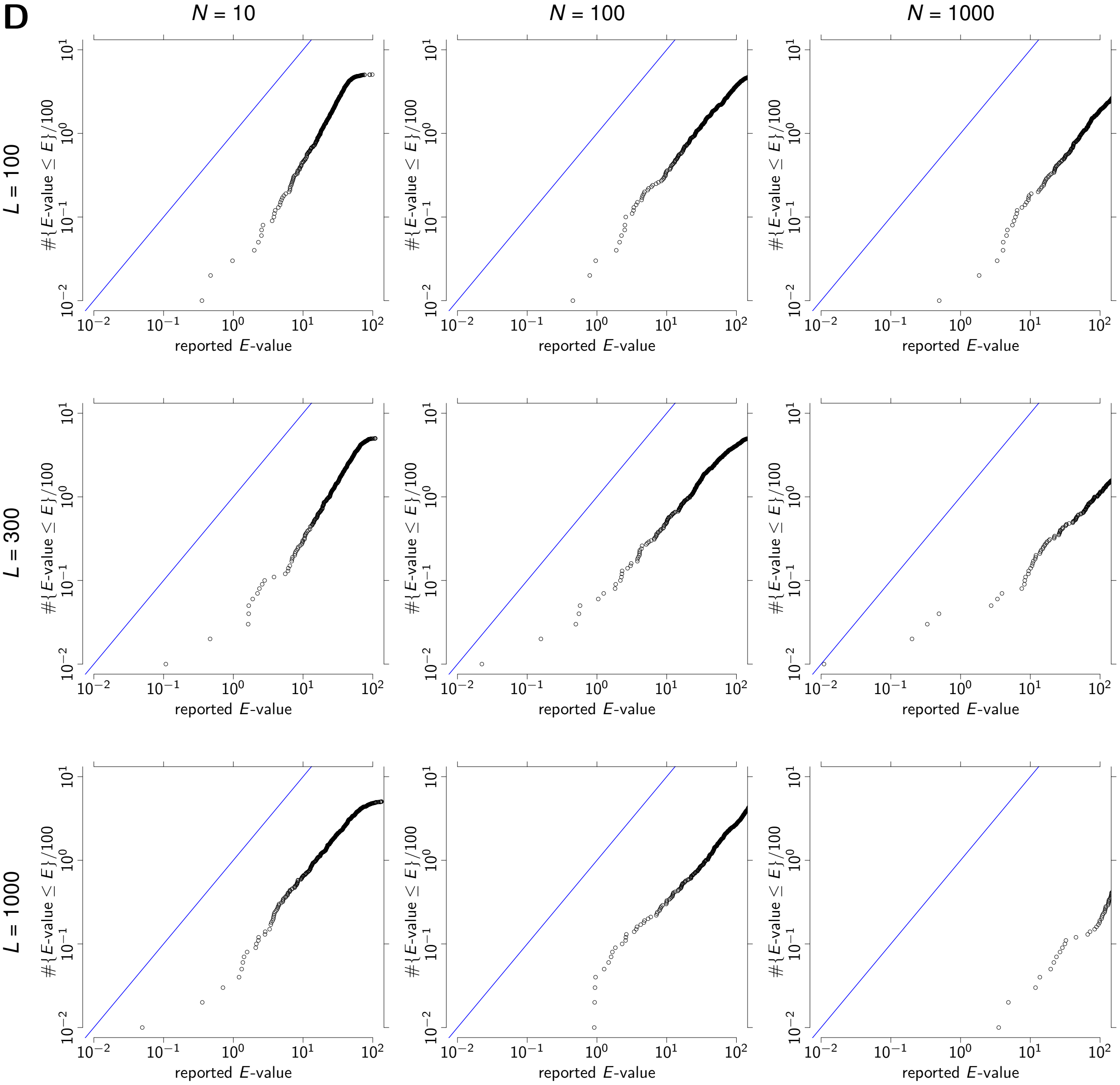
Second order background model, zero or one occurrence per sequence.



Eighth order background model, multiple occurrences per sequence.

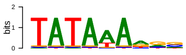
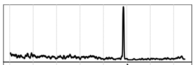
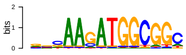
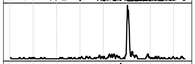
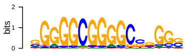
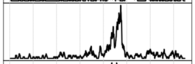
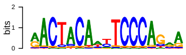
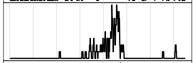
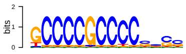
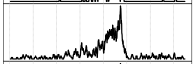
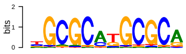
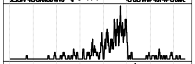
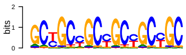
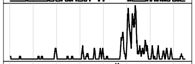
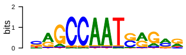
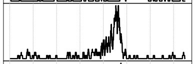
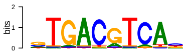
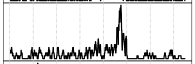
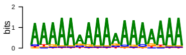
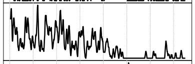
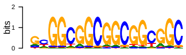
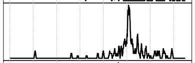
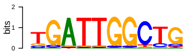
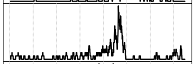
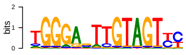
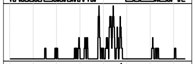
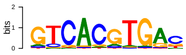
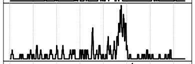
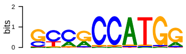
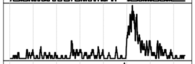
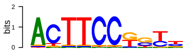
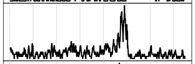
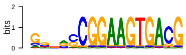
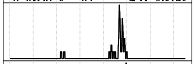
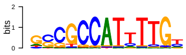
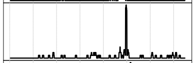
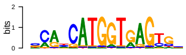
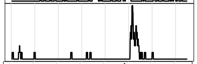
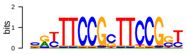
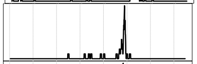
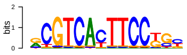
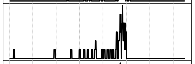
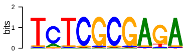
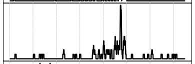
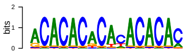
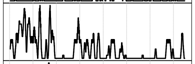
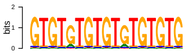
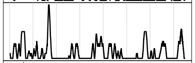
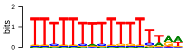
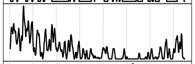
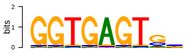
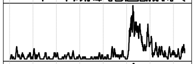
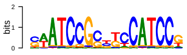
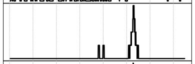
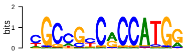
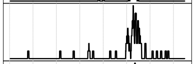
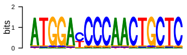
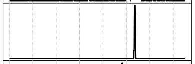
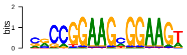
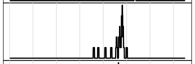
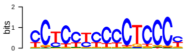
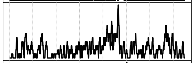
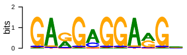
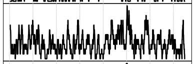
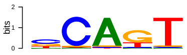
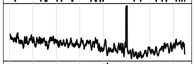
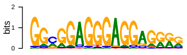
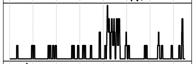
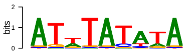
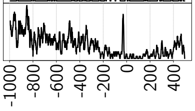


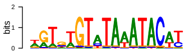
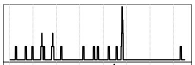
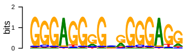
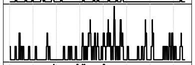
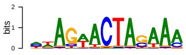
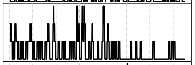
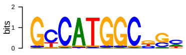
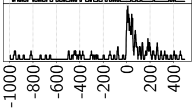
Eighth order background model, zero or one occurrence per sequence.



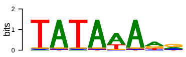
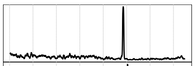
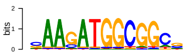
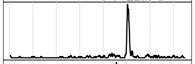
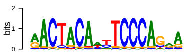
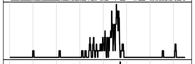
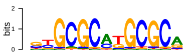
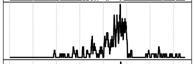
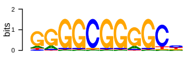
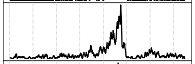
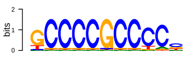
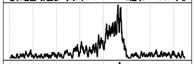
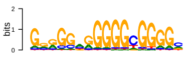
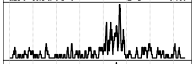
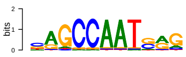
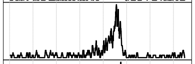
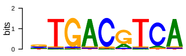
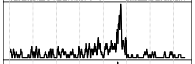
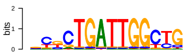
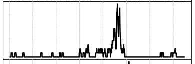
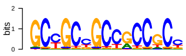
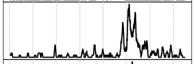
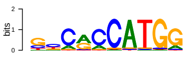
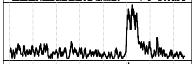
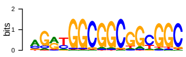
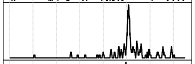
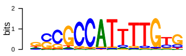
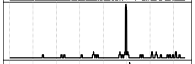
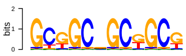
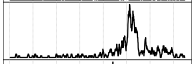
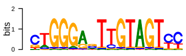
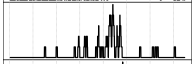
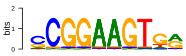
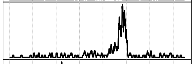
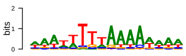
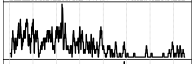
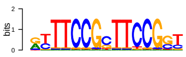
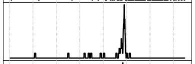
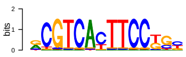
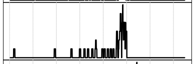
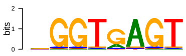
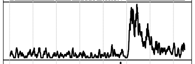
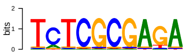
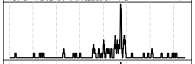
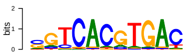
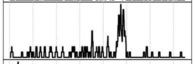
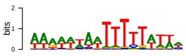
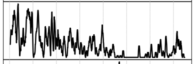
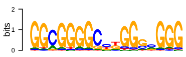
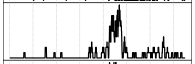
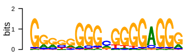
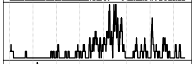
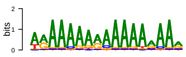
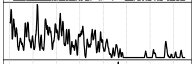
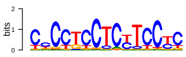
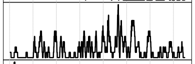
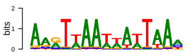
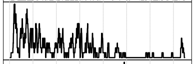
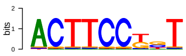
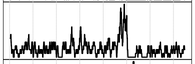
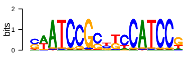
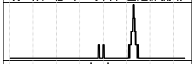
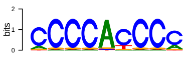
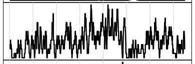
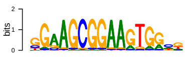
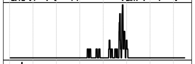
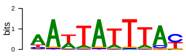
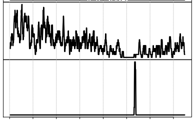
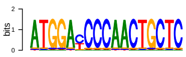
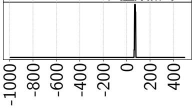
Supplementary Figure 6: Cumulative distribution of E-values reported by XXmotif on sets of random sequences with different numbers of sequences (N = 10, 100, 1000 sequences) and lengths (L = 100, 300, 1000). (A) Second order background model, multiple occurrences per sequence. (B) Second order background model, zero or one occurrence per sequence. (C) Eighth order background model, multiple occurrences per sequence. (D) Eighth order background model, zero or one occurrence per sequence.

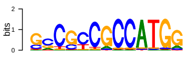
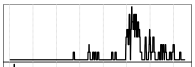
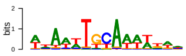
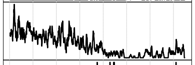
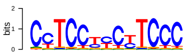
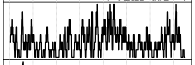
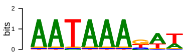
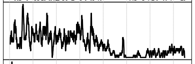
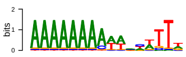
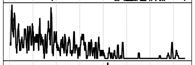
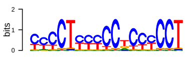
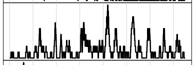
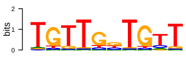
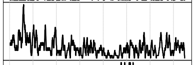
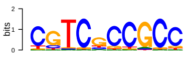
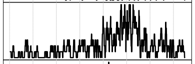
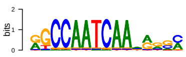
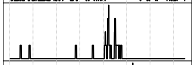
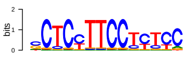
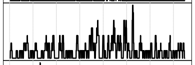
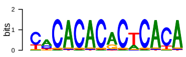
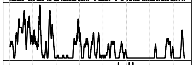
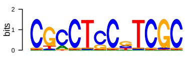
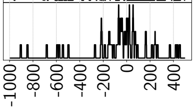
A

Motif	Logo	Distribution	occ [%]	*E*-value	Name	TOMTOM *E*-value
1			9.29	2×10^{-63}	TATA-box	7.24×10^{-8}
2			5.80	4×10^{-63}	YY1	–
3			9.88	2×10^{-59}	SP1 (rev)	2.48×10^{-4}
4			2.84	2×10^{-58}	SREBF1	–
5			9.98	9×10^{-57}	SP1	3.41×10^{-4}
6			6.60	7×10^{-44}	NRF1	–
7			2.42	3×10^{-36}	motif8 (rev) [1]	–
8			6.33	4×10^{-30}	NFYA	2.89×10^{-10}
9			5.26	5×10^{-29}	CREB1	1.82×10^{-4}
10			1.72	1×10^{-28}	–	–
11			3.11	2×10^{-27}	motif8 [1]	–
12			3.92	6×10^{-22}	NFYA (rev)	9.56×10^{-4}
13			1.13	2×10^{-16}	SREBF1 (rev)	–
14			3.86	4×10^{-15}	MYCN	8.43×10^{-4}
15			4.88	2×10^{-13}	Kozak sequence	
16			7.51	3×10^{-13}	SPI1 (rev)	1.12×10^{-3}
17			1.18	4×10^{-13}	XX1 (rev)	–
18			1.13	2×10^{-12}	YY1 (rev)	–
19			1.40	2×10^{-12}	XX2 (Kozak-like)	–
20			1.18	3×10^{-12}	XX3	–
21			1.29	1×10^{-10}	XX1	–
22			1.83	3×10^{-10}	CLUS1 element [2]	–
23			0.54	9×10^{-9}	–	–
24			0.43	1×10^{-8}	–	–
25			0.86	3×10^{-8}	–	–
26			5.74	6×10^{-8}	XX4	–
27			0.64	8×10^{-7}	XX5	–
28			1.18	9×10^{-6}	Kozak sequence	–
29			0.27	1×10^{-5}	coding, MT genes	–
30			0.81	4×10^{-5}	XX3 (rev)	–
31			3.27	5×10^{-5}	–	–
32			3.86	7×10^{-5}	–	–
33			6.55	3×10^{-4}	XX6 (Inr)	–
34			1.07	4×10^{-4}	- -	–
35			2.15	9×10^{-4}	–	–

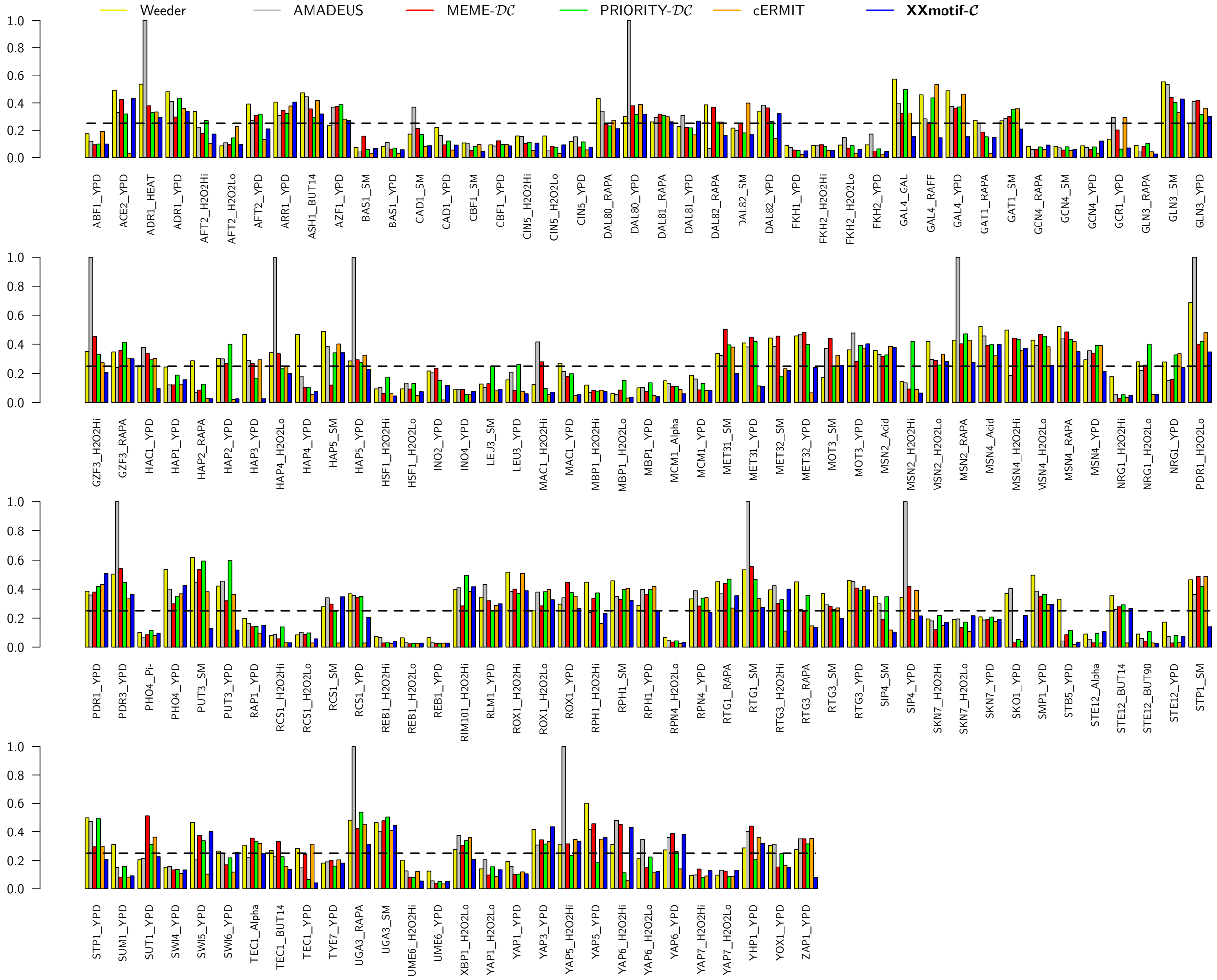
Motif	Logo	Distribution	occ [%]	*E*-value	Name	TOMTOM *E*-value
36			0.38	2×10^{-3}	TATA-like	–
37			1.02	2×10^{-3}	–	–
38			0.70	4×10^{-2}	–	–
39			2.63	4×10^{-2}	Kozak sequence	–

B

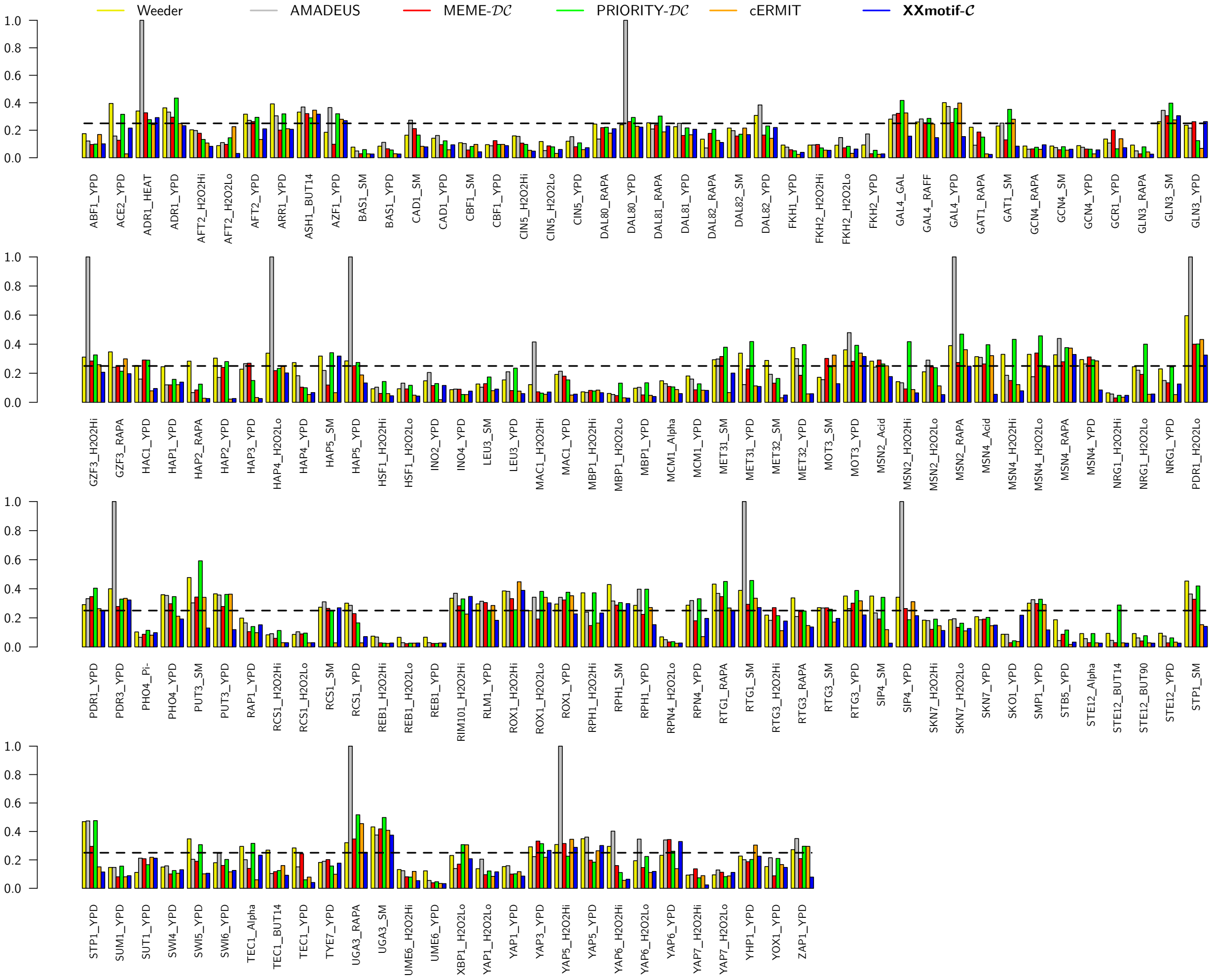
Motif	Logo	Distribution	occ [%]	*E*-value	Name	TOMTOM *E*-value
1			8.75	8×10^{-69}	TATA-box	2.51×10^{-6}
2			4.99	2×10^{-63}	YY1	–
3			2.74	4×10^{-57}	SREBF1	–
4			6.44	7×10^{-47}	NRF1	–
5			11.92	1×10^{-43}	SP1 (rev)	1.69×10^{-3}
6			10.04	1×10^{-41}	SP1	1.48×10^{-4}
7			5.32	9×10^{-27}	SP1 (rev)	1.33×10^{-5}
8			6.23	1×10^{-26}	NFYA	1.32×10^{-6}
9			4.67	2×10^{-22}	CREB1	2.98×10^{-4}
10			2.58	3×10^{-22}	NFYA (rev)	8.08×10^{-7}
11			2.74	8×10^{-22}	motif8 (rev) 1)	–
12			8.00	1×10^{-21}	Kozak sequence	–
13			2.15	2×10^{-18}	motif8 1)	–
14			1.18	5×10^{-16}	YY1 (rev)	–
15			5.48	1×10^{-15}	motif8 1)	–
16			0.97	5×10^{-14}	SREBF1 (rev)	–
17			4.83	1×10^{-12}	GABPA	2.43×10^{-6}
18			3.76	5×10^{-12}	–	–
19			1.18	5×10^{-12}	XX3	–
20			1.24	8×10^{-11}	XX1	–
21			9.13	2×10^{-10}	XX4	–
22			1.83	3×10^{-10}	CLUS1 element 2)	–
23			2.36	2×10^{-9}	MYCN	6.61×10^{-4}
24			1.61	2×10^{-9}	–	–
25			2.36	8×10^{-9}	SP1 (rev)	5.36×10^{-2}
26			1.93	4×10^{-8}	–	–
27			1.13	2×10^{-7}	–	–
28			3.01	2×10^{-7}	–	–
29			0.91	6×10^{-7}	HNF1B (rev)	3.25×10^{-4}
30			4.30	7×10^{-7}	SPI1 (rev)	4.34×10^{-4}
31			0.64	1×10^{-6}	XX5	–
32			6.77	2×10^{-6}	KLF4 (rev)	4.62×10^{-3}
33			0.97	3×10^{-6}	GABPA	8.44×10^{-3}
34			1.77	6×10^{-6}	FOXA2	2.40×10^{-3}
35			0.27	2×10^{-5}	coding, MT genes	–

Motif	Logo	Distribution	occ [%]	*E*-value	Name	TOMTOM *E*-value
36			1.61	3×10^{-5}	Kozak sequence	–
37			2.74	7×10^{-5}	–	–
38			2.47	2×10^{-4}	–	–
39			2.15	3×10^{-4}	–	–
40			0.81	5×10^{-4}	–	–
41			1.13	2×10^{-2}	–	–
42			1.88	2×10^{-2}	–	–
43			4.35	3×10^{-2}	–	–
44			0.70	4×10^{-2}	NFYA	1.69×10^{-3}
45			1.29	6×10^{-2}	KLF4	6.09×10^{-2}
46			0.43	8×10^{-2}	–	–
47			0.91	9×10^{-2}	–	–

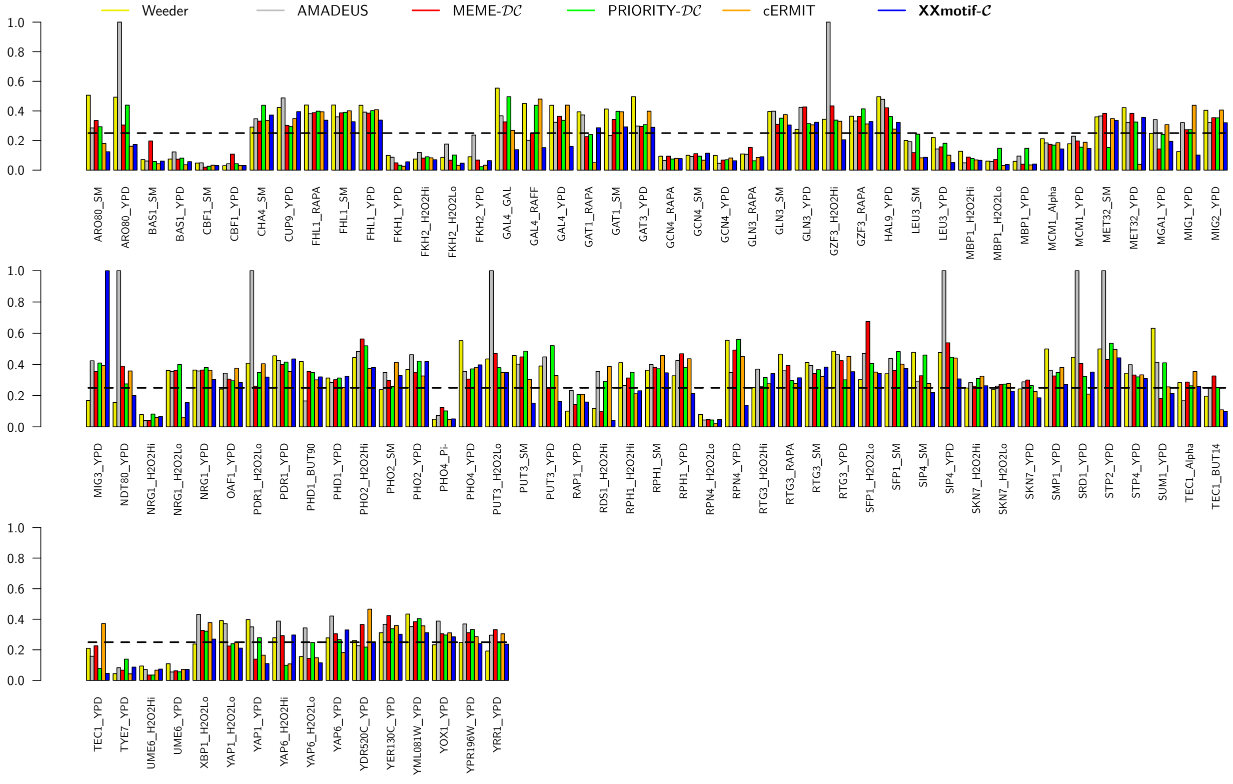
Supplementary Figure 7: Full list of human core promoter motifs discovered by XXmotif up to *E*-value 0.1 on 1871 human core promoter regions (300 bp to +100 bp around TSS) from the eukaryotic promoter database (A) without masking and (B) with masking of core promoter sequences using RepeatMasker (www.repeatmasker.org) prior to analysis.



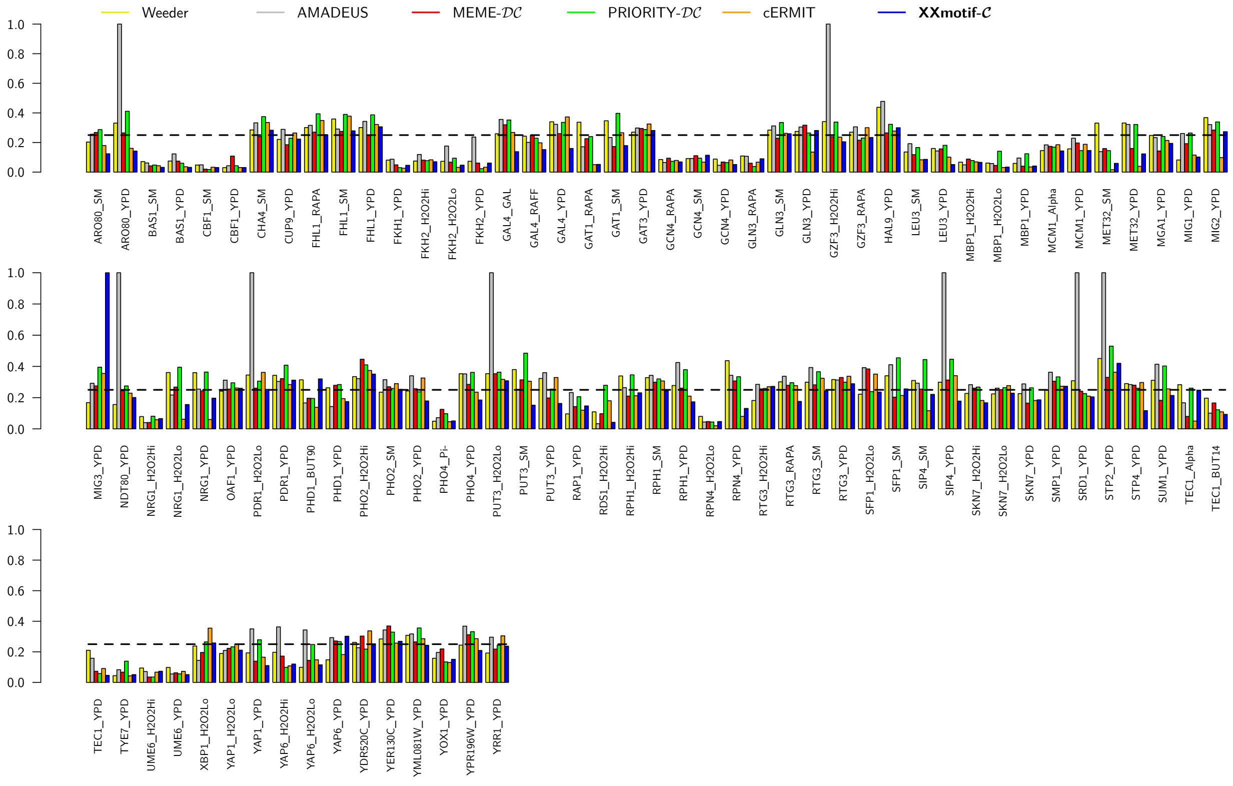
Supplementary Figure 9: Detailed top 1 results of the sensitivity benchmark (“Harbison set”). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



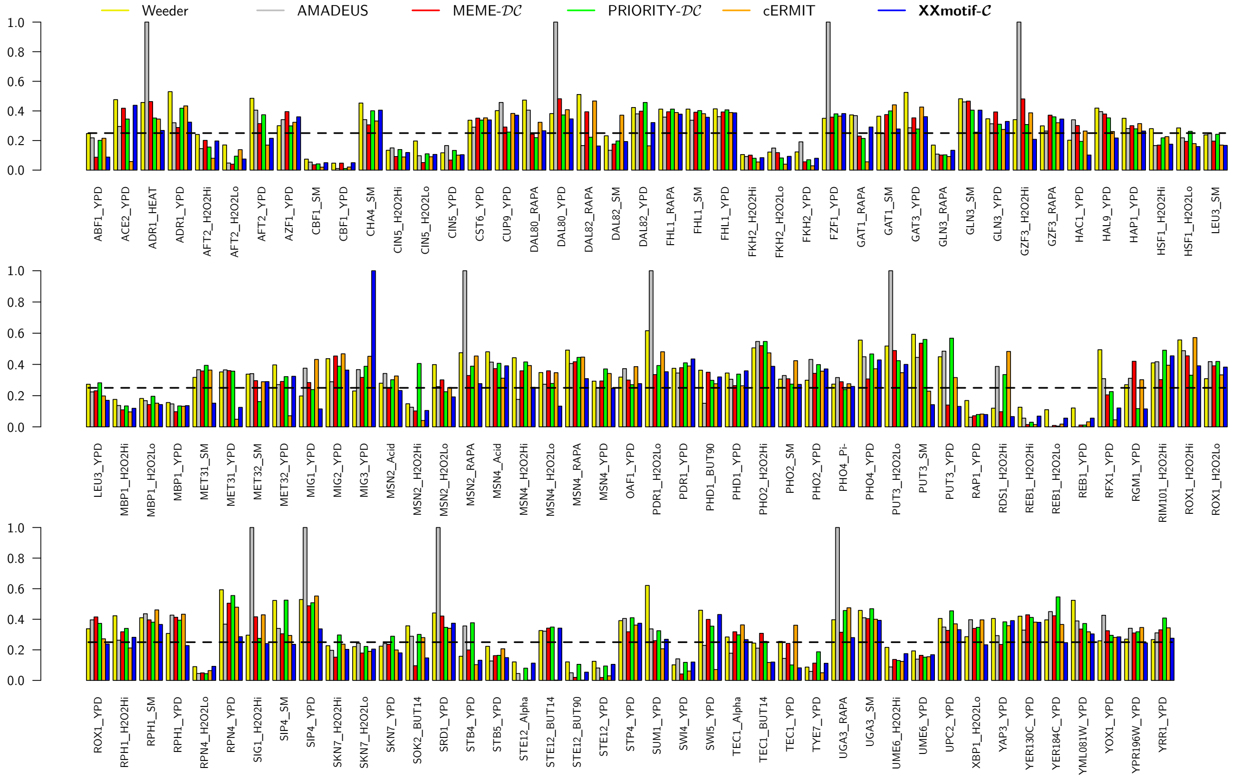
Supplementary Figure 10: Detailed top 4 results of the sensitivity benchmark ("Harbison set"). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



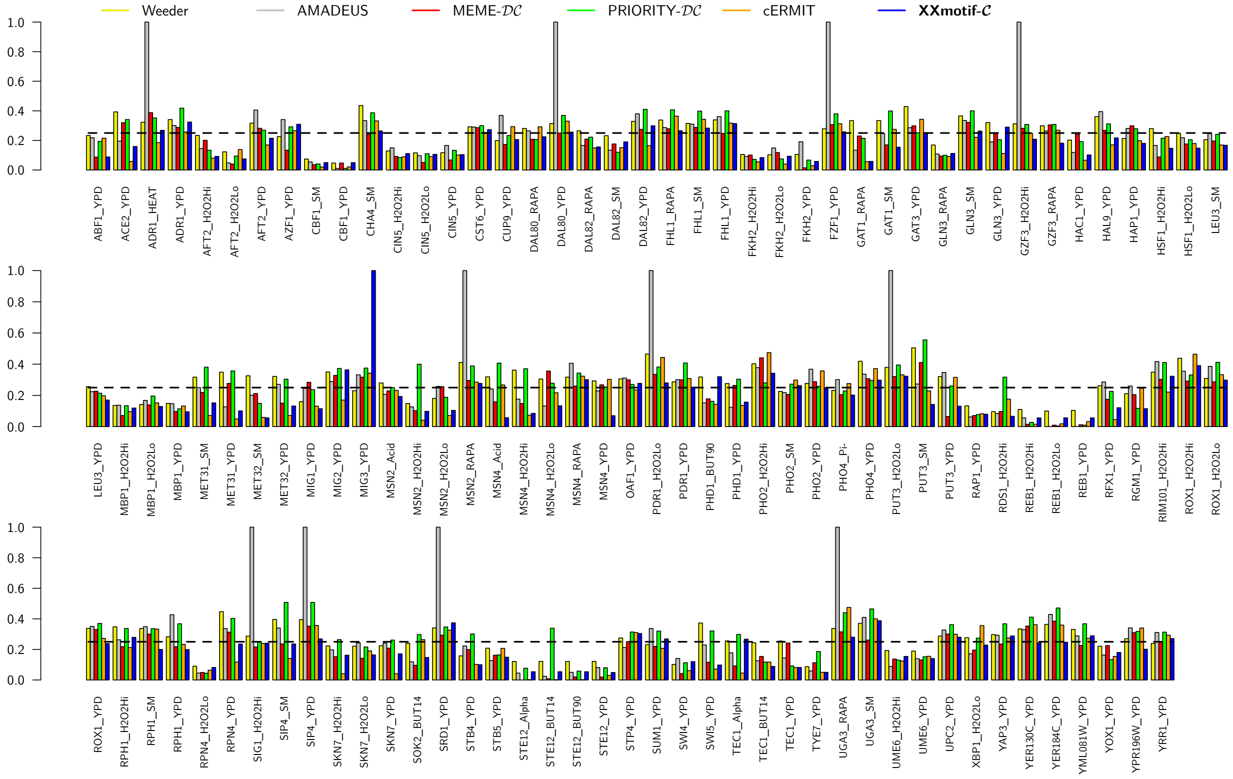
Supplementary Figure 11: Detailed top 1 results of the sensitivity benchmark ("Bulyk set"). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



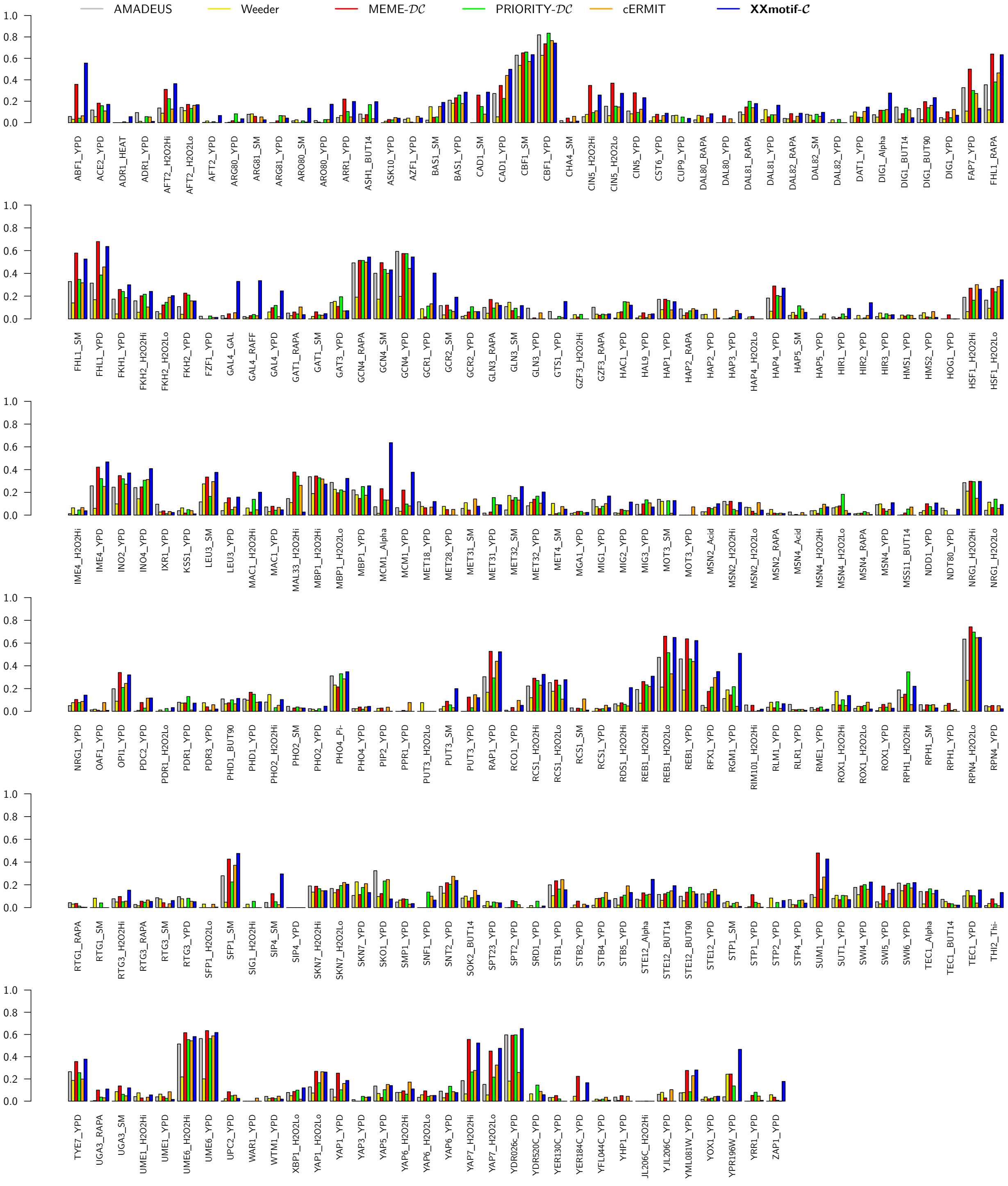
Supplementary Figure 12: Detailed top 4 results of the sensitivity benchmark (“Bulyk set”). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



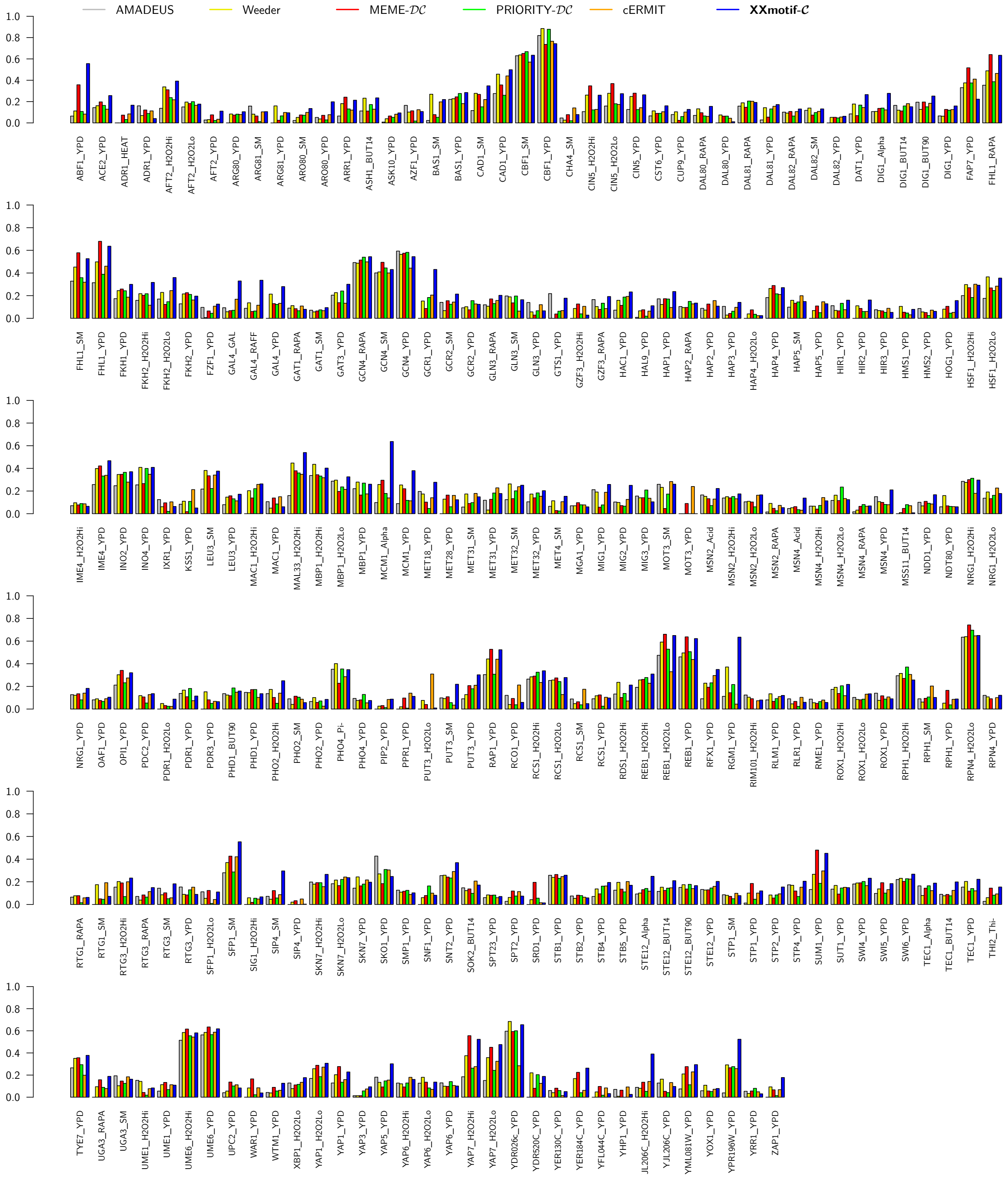
Supplementary Figure 13: Detailed top 1 results of the sensitivity benchmark ("Hughes set"). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



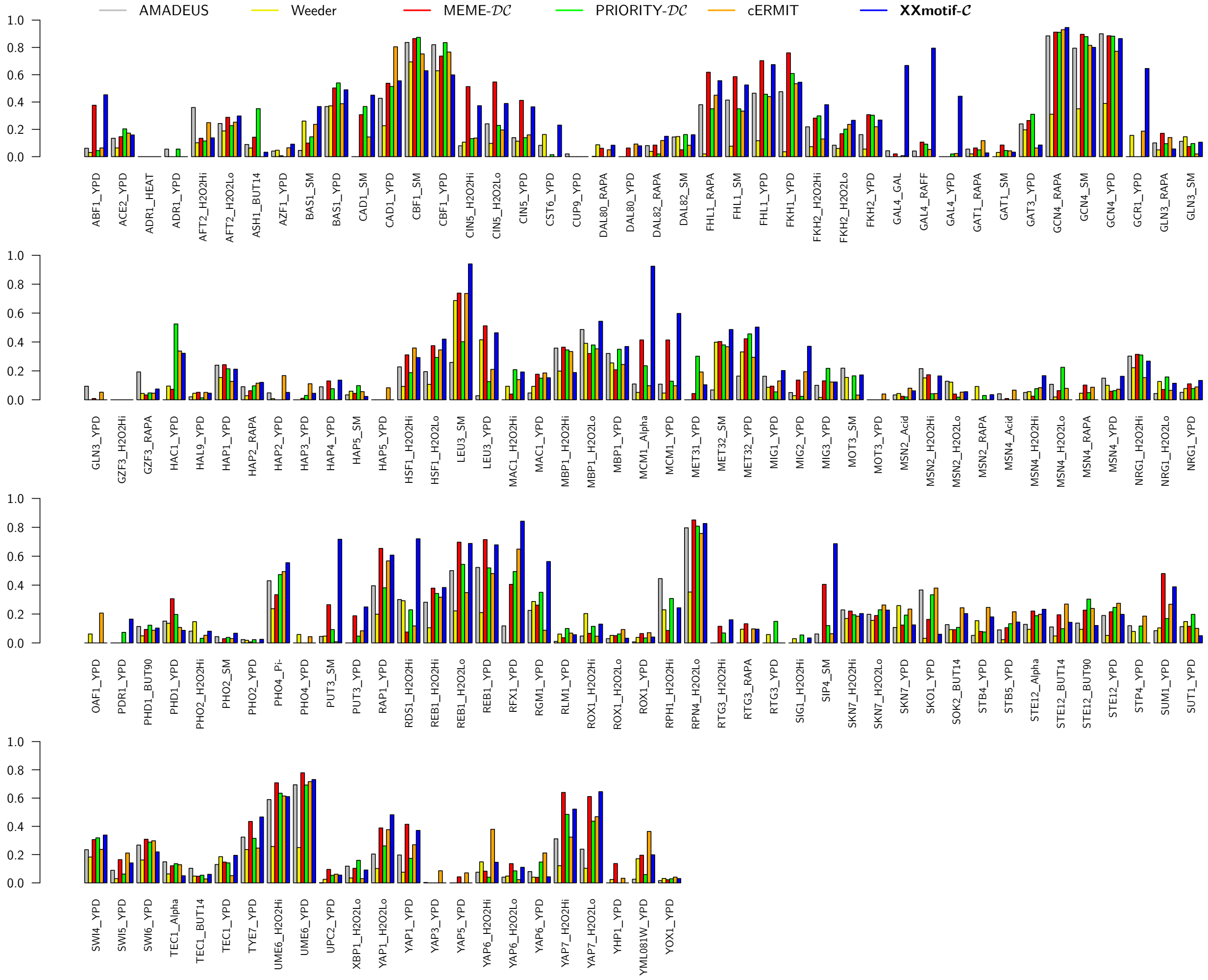
Supplementary Figure 14: Detailed top 4 results of the sensitivity benchmark ("Hughes set"). The dashed line marks the maximum euclidian distance of 0.25 until which the motif is counted as correctly identified.



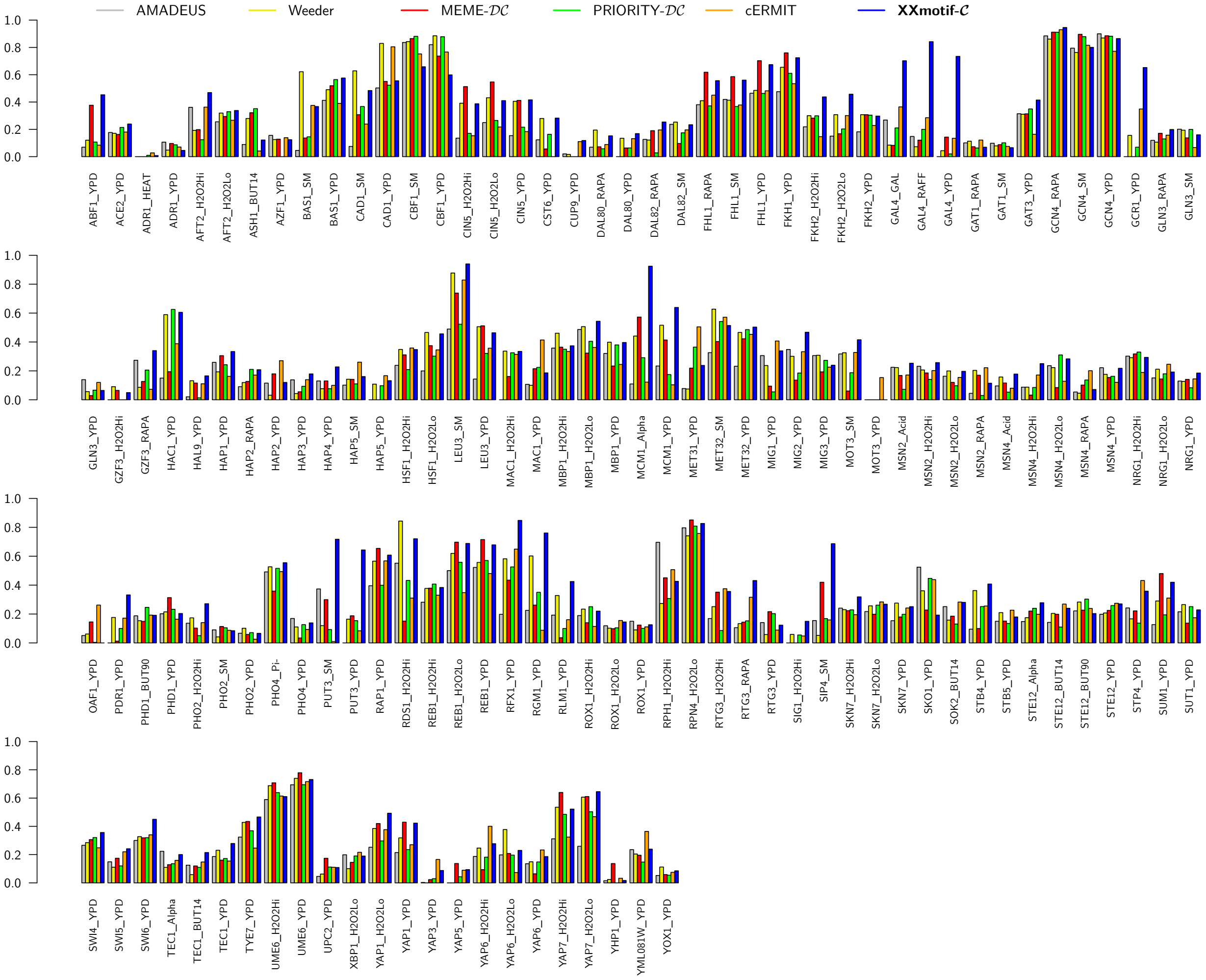
Supplementary Figure 15: Detailed top 1 results of PWM quality benchmark. The height of the bars corresponds to the pAUC value for the specific dataset.



Supplementary Figure 16: Detailed top 4 results of PWM quality benchmark. The height of the bars corresponds to the pAUC value for the specific dataset.



Supplementary Figure 17: Detailed top 1 results of PWM quality benchmark filtered to sequences containing a literature motif. The height of the bars corresponds to the pAUC value for the specific dataset.



Supplementary Figure 18: Detailed top 4 results of PWM quality benchmark filtered to sequences containing a literature motif. The height of the bars corresponds to the pAUC value for the specific dataset.

Part II. Supplemental Tables

A

	Harbison		Bulyk		Hughes		Sum	
	TOP1	TOP4	TOP1	TOP4	TOP1	TOP4	TOP1	TOP4
MEME	35	98	18	57	19	65	72	220
PRIORITY	70	92	33	43	36	53	139	188
MEME-$\mathcal{M}$	67	97	35	53	39	70	141	220
Weeder	65	86	43	54	40	53	148	193
AMADEUS	74	96	32	42	45	65	151	203
MEME-$\mathcal{D}$	74	105	34	57	45	74	153	236
MEME-$\mathcal{DC}$	74	106	35	59	46	76	155	241
PRIORITY-$\mathcal{D}$	79	93	36	45	41	58	156	196
PRIORITY-$\mathcal{DC}$	79	93	34	44	43	51	156	188
XXmotif-5-noref	91	104	43	51	54	61	188	216
XXmotif-noref	92	109	42	61	56	70	190	240
XXmotif	99	128	**58**	**78**	63	**93**	220	**299**
XXmotif-$\mathcal{C}$	**105**	**133**	53	75	**65**	85	**223**	293
ERMIT	88	115	36	51	56	77	180	243
cERMIT	88	119	39	60	50	80	177	259

B

	Harbison		Bulyk		Hughes		Sum	
	TOP1	TOP4	TOP1	TOP4	TOP1	TOP4	TOP1	TOP4
MEME	33	99	18	54	20	65	71	218
PRIORITY	68	90	32	44	37	51	137	185
MEME-$\mathcal{D}$	68	98	31	55	44	67	143	220
PRIORITY-$\mathcal{D}$	73	87	32	38	40	48	145	173
MEME-$\mathcal{M}$	69	103	38	58	41	72	148	233
AMADEUS	73	97	31	46	46	67	150	210
Weeder	70	87	40	55	41	52	151	194
MEME-$\mathcal{DC}$	75	105	38	64	49	80	162	249
PRIORITY-$\mathcal{DC}$	82	95	37	44	47	55	166	194
XXmotif-5-noref	91	104	43	51	54	61	188	216
XXmotif-noref	92	109	42	61	56	70	190	240
XXmotif	99	128	**58**	**78**	63	**93**	220	**299**
XXmotif-$\mathcal{C}$	**105**	**133**	53	75	**65**	85	**223**	293
ERMIT	91	117	37	63	48	81	176	261
cERMIT	94	117	43	63	53	83	190	263

Supplementary Table 1: Detailed results of the motif sensitivity benchmark. The tools are sorted by the sum of the top 1 predictions. Highest number per benchmark set is given in bold face. Methods above the separator take only intergenic regions with a ChIP-chip P-value $< 10^{-3}$ as input, methods below the separator take all intergenic regions and require the associated P-value as additional information. (A) XXmasker is applied only to the input sequences of XXmotif. (B) XXmasker is applied to the input sequences of all tools.

Motif	Count	Total	Bonferroni	Category	Term
XX1	4	22	1.8×10^{-3}	CC	GO:0022627 cytosolic small ribosomal subunit
	6	22	4.2×10^{-2}	CC	GO:0030529 ribonucleoprotein complex
	4	22	1.2×10^{-1}	BP	GO:0006414 translational elongation
XX1rev	—				
XX2	11	25	7.7×10^{-13}	CC	GO:0044445 cytosolic part
	10	23	6.2×10^{-12}	BP	GO:0006414 translational elongation
	10	23	6.5×10^{-11}	MF	GO:0003735 structural constituent of ribosome
XX3	8	20	5.3×10^{-5}	BP	GO:0006412 translation
	14	20	2.7×10^{-4}	BP	GO:0044267 cellular protein metabolic process
	6	21	5.8×10^{-4}	CC	GO:0005840 ribosome
	5	20	5.1×10^{-3}	MF	GO:0003735 structural constituent of ribosome
XX3rev	—				
XX4	62	97	7.3×10^{-9}	CC	GO:0044444 cytoplasmic part
	9	91	2.7×10^{-4}	BP	GO:0006414 translational elongation
	10	94	2.7×10^{-4}	MF	GO:0003735 structural constituent of ribosome
XX5	10	11	4.5×10^{-17}	BP	GO:0006414 translational elongation
	10	12	2.3×10^{-15}	MF	GO:0003735 structural constituent of ribosome
XX6	14	98	7.0×10^{-5}	BP	GO:0008380 RNA splicing
	83	106	6.3×10^{-4}	CC	GO:0043226 organelle
	35	108	2.3×10^{-3}	MF	GO:0000166 nucleotide binding

Supplementary Table 2: Gene ontology analysis (Huang et al., 2009) on genes in the Eukaryotic Promoter Database in which XXmotif detected one of the novel motifs.

Motif		Quantile		
	(number)	10%	50%	90%
XX1	(21)	8.70	19.80	83.49
XX2(Kozak-long)	(19)	0.28	29.02	182.78
XX3	(20)	9.30	31.13	152.53
XX4	(26)	0.13	16.37	117.80
XX5	(27)	7.60	36.58	258.9
XX6(Inr)	(33)	1.96	26.32	85.55
XX1(rev)	(17)	7.34	18.68	46.16
XX3(rev)	(30)	4.28	19.17	63.27
All EPD	(1871)	0.07	15.84	83.30

Supplementary Table 3: Expression levels of genes (Lundberg et al., 2010) carrying one of the motifs discovered by XXmotif. The genes were identified by XXmotif together with the motif in core promoter regions given by the eukaryotic promoter database (EPD). Quantiles of expression levels of all genes in EPD found to carry the motif are given in units of RPKM (reads per kilobase gene model and million reads). Genes with motifs XX1 to XX4 are strongly expressed. Note that genes XX1 and XX3 have a much higher expression than their reverse complements. Therefore, the binding orientation of the associated, unknown factors seems important for the assembly of the transcription initiation complex, which underscores the importance of XX1 and XX3 as true core promoter motifs despite their relatively infrequent occurrence.

Part III. Supplemental Methods

1. Introduction

These supplemental methods descriptions provide further details of the theoretical basis and the efficient realization of significance calculations within XXmotif. Furthermore, it provides all the parameters used for the tested motif finding tools within the benchmarks. This should support repeatability of the results within the paper and point out the main ideas that allow an efficient calculation of P-values from PWMs.

The remainder of this document is organized as follows: Section two describes from where XXmotif can be obtained. In the third section, we provide the theoretical basis for calculating match a P-values for a site and a motif enrichment P-value for a set of sequences. The fourth section gives a more detailed description of the XXmotif workflow. Finally, section five gives an overview about the tools used in the benchmarks and the chosen parameters.

2. Availability

A command line version of XXmotif can be obtained as source code or binaries (64 Bit and 32 Bit versions for UNIX systems) from `ftp://toolkit.lmb.uni-muenchen.de/xxmotif`.

A web server is available at `xxmotif.genzentrum.lmu.de`.

3. Theory

In order to optimize the enrichment P-value for a motif PWM, it has to be possible to calculate the significance of a specific site given the PWM (“match P-value”). This section provides the theoretical basis to efficiently calculate P-values for PWMs.

3.1. Calculating the background model probability for an l-mer

To calculate the probability to find a given l-mer x by chance, a background model is used. This model should be calibrated on a set having the same DNA properties as the input set, but no enriched motifs (negative set). The simplest background model assumes no correlations between the positions of the l-mer (0^{th}-order background model), and hence, utilizes only monomer probabilities of the nucleotides $f(x_i)$ on the negative set. According to this background model, the probability to find an l-mer x is:

$$P_{\text{bg}}(x) = \prod_{i=1}^{l} f(x_i) \tag{1}$$

However, the independence assumption underlying this model is very inaccurate and leads to an overestimation of the significance of poly A/T stretches or dinucleotide repeats, which are very frequent in non-coding DNA.

Therefore, many motif finding tools use higher-order background models to capture these dependencies. E. g., for a k^{th}-order background model all $(k+1)$-mers within the negative set are counted and probabilities $f(x_1 \dots x_{k+1})$ and conditional probabilities $f(x_{k+1}|x_1 \dots x_k)$ are calculated. With these, the probability to find an l-mer x can be calculated as follows:

$$P_{\text{bg}}(x) = f(x_1 \dots x_{k+1}) \prod_{i=k+2}^{l} f(x_i|x_{i-k} \dots x_{i-1}) \tag{2}$$

The main drawback of this method is the huge amount of possible $(k+1)$-mers for large k's necessary to estimate from usually limited data. As poly A/T stretches and many dinucleotide repeats of six or more nucleotides are overrepresented in genomic sequences, a k of at least 8 is still useful. However, this leads to very few counts that are indistinguishable from noise for many of the 262144 different 9-mers even for large negative sets.

To overcome this problem, XXmotif uses interpolated Markov models (Salzberg et al., 1998) which automatically use lower-order probabilities if the negative set does not provide enough counts for higher-order k-mers. Given a pseudocount factor α and the number of occurrences of a k-mer $n(x_1 \dots x_k)$, the conditional probability $f(x_{k+1}|x_1 \dots x_k)$ can be calculated as follows:

$$f(x_{k+1}|x_1 \dots x_k) = \frac{n(x_1 \dots x_{k+1}) + 4\alpha f(x_{k+1}|x_2 \dots x_k)}{n(x_1 \dots x_k) + 4\alpha} \tag{3}$$

In case of few k-mer counts, i. e., $n(x_1 \dots x_{k+1}) \approx 0$, the formula simplifies to the result for order $k-1$. However, if the k-mer counts are high, i. e., $n(x_1 \dots x_{k+1}) \gg 4\alpha$, the formula corresponds to the one for order k.

We used $\alpha = 10$ as default value for XXmotif as it seems to be a good trade-off between noise reduction and utilization of the counts of higher orders.

3.2. Calculating the match P-value measuring the significance of a binding site

To calculate the P-value of a specific site x of interest with length l given a PWM, the probabilities according to the background model $P_{\text{bg}}(z)$ of all l-mers z that have a better or equal log-odds score $S(z)$ than the site of interest x have to be summed up:

$$P\text{-value}(x) = \sum_{z \in \{z : S(z) \geq S(x)\}} P_{\text{bg}}(z)\,, \tag{4}$$

where the log-odds score $S(x)$ is calculated by summing up the logarithm of the probability $\text{PWM}(i, x_i)$ to have nucleotide x_i at position i within the PWM divided by the background probability $f(x_i)$ of this nucleotide:

$$S(x) = \sum_{i=1}^{l} \log\left(\frac{\text{PWM}(i, x_i)}{f(x_i)}\right) \tag{5}$$

Since it is very time consuming to generate all 4^l l-mers, P-value(x) cannot be efficiently obtained by exhaustive enumeration of all l-mers with score $S(z) \geq S(x)$. However, by using a branch-and-bound technique, it is possible to generate exactly these high-scoring l-mers in linear time with respect to the output size, i. e., the number of l-mers generated.

3.2.1. Branch-and-bound algorithm

Every PWM column contributes independently to the log-odds score (see Equation 5). Therefore, given the prefix of length m of an l-mer, the maximum log-odds score of the remaining suffix $S_{\max,m+1}$ is easily calculated by summing up the maximum log-odds value of the corresponding columns:

$$S_{\max,m+1} = \sum_{i=m+1}^{l} \max_{j \in \{A,C,G,T\}} \left\{ \log \left(\frac{\mathrm{PWM}(i,j)}{f(j)} \right) \right\} \tag{6}$$

If the maximum score of the suffix is not high enough to reach the threshold $S(x)$, it is not necessary to enumerate the suffixes and the current path can be abandoned. All paths reaching the l-th cumn correspond to l-mers that are ‘similar enough’ to the PWM. Pseudocode for the procedure is given in Algorithm 1.

Algorithm 1: CreateSimilarKmers(i, S_{i-1}, z_{i-1})
Recursive generation of l-mers similar to a PWM with branch-and-bound. The initial call is CreateSimilarKmers$(1, 0, \varepsilon)$, i. e., the algorithm starts in the first column with the neutral element of addition for the score so far and the empty word for the preceding l-mer.

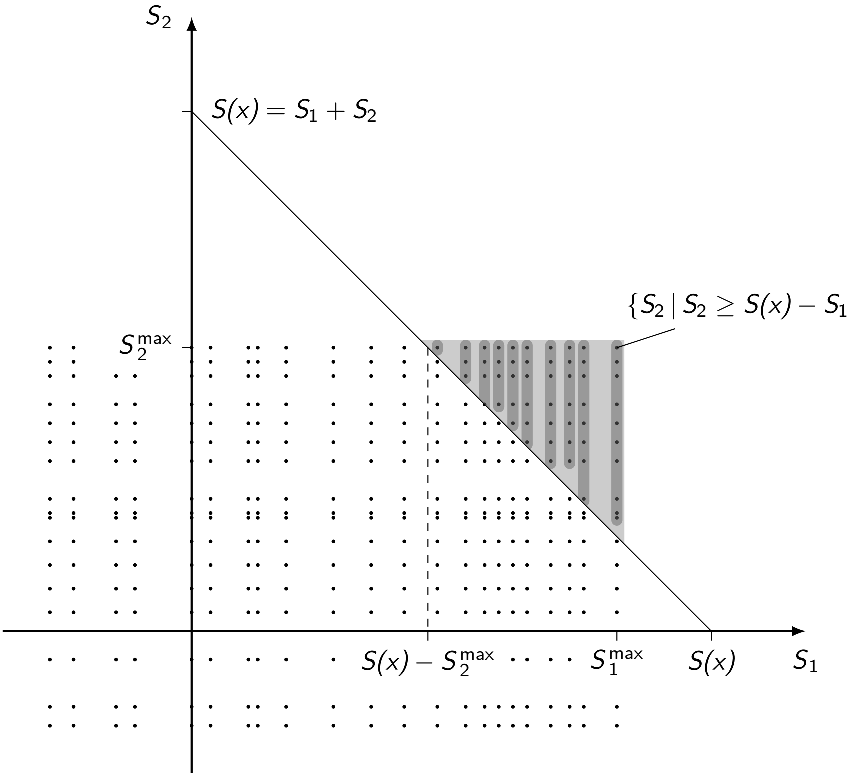
```
Data: PWM score matrix
      $S_{\max,j}$ maximum possible score for columns $j, \dots, l$
      $S(x)$ similarity threshold: score for site $x$
Input: $i$ current column
       $z_{i-1}$ generated $(i-1)$-mer
       $S_{i-1}$ score of $z_{i-1}$
foreach $j \in \{A, C, G, T\}$ do
    $S_i \leftarrow S_{i-1} + \mathrm{PWM}_{i,j}$
    if $S_i + S_{\max,i+1} < S(x)$ then continue
    $z_i \leftarrow z_{i-1} \cdot j$
    if $i < l$ then
        CreateSimilarKmers$(i+1, S_i, z_i)$
    else
        add $z_i$ to list of similar $l$-mers
```

The run time is approximately linear in the number of branches followed, that is, in the number of l-mers generated plus the number of dead-end paths. Two optimizations are used to reduce the number of futile bifurcations by trying highest scoring nucleotides first: (a) sort each column’s entries in order of descending score, (b) reorder columns according to their highest scoring entry in descending order.

Of course, the maximal number of similar l-mers is still 4^l. However, only high scoring matches are of interest during the search for significant motifs. This means that only a small fraction of l-mers has to be considered for stringent thresholds.

3.2.2. Splitting the match *P*-value calculation for long PWMs

For long l-mers, enumeration of high scoring matches can still be very time consuming, especially if the PWM has many degenerate columns. Therefore, we accelerate the calibration for $l > 8$ by splitting the motif into two parts, calculate P-values for both parts individually and combine them to yield the final P-value (see Fig. 19). The left part of the l-mer (x_l) is set to length 8, the right part (x_r) to the remaining nucleotides, allowing to calculate P-values for l-mers with up to 17 nucleotides.



Supplementary Figure 19: Splitting and recombination of the branch-and-bound P-value calculation for long l-mers (shown in log-space). The l-mer $z = z_1 \dots z_l$ is divided into a left and a right part, $z_\mathrm{l} = z_1 \dots z_8$ and $z_\mathrm{r} = z_9 \dots z_l$. The background model probabilities for z_l and z_r are calculated with the branch-and-bound algorithm CreateSimilarKmers. The P-value $P(x)$ for an l-mer x is the background model probability for a chance l-mer $Z = (z_\mathrm{l}, z_\mathrm{r})$ to reach at least a total score of $S(x)$. To calculate it, the background probabilities of all pairs $(z_\mathrm{l}, z_\mathrm{r})$ with $S_\mathrm{l}(z_\mathrm{l}) + S_\mathrm{r}(z_\mathrm{r}) \geq S(x)$ (dots in gray regions) have to be summed up.

The P-value $P(x)$ for an l-mer x with PWM score $S(x)$ is the background model probability for a chance l-mer $z = (z_\mathrm{l}, z_\mathrm{r})$ to obtain at least a total score of $S(x)$:

$$P(x) = P_\mathrm{bg}\left(S_\mathrm{l}(z_\mathrm{l}) + S_\mathrm{r}(z_\mathrm{r}) \geq S(x) \right) , \tag{7}$$

where, $P_\mathrm{bg}(\cdot)$ denotes the probability of a k-mer according to the background model. We can separate the contributions of the left and right part of the l-mer by writing

$$P(x) = \sum_{(z_\mathrm{l},z_\mathrm{r})\,:\,S_\mathrm{l}(z_\mathrm{l})+S_\mathrm{r}(z_\mathrm{r})\geq S(x)} P_\mathrm{bg}(z_\mathrm{l})\, P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l})\,. \tag{8}$$

or, equivalently,

$$P(x) = \sum_{z_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{l}) \sum_{z_\mathrm{r}\,:\,S_\mathrm{r}(z_\mathrm{r})\geq S(x)-S_\mathrm{l}(z_\mathrm{l})} P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}) \tag{9}$$

This calculation is illustrated in Figure 19.

If we neglected dependencies between both parts, we could simplify this expression to

$$P(x) \approx \sum_{z_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{l}) \sum_{z_\mathrm{r}\,:\,S_\mathrm{r}(z_\mathrm{r})\geq S(x)-S_\mathrm{l}(z_\mathrm{l})} P_\mathrm{bg}(z_\mathrm{r})\,. \tag{10}$$

The sums over z_r (dark grey region in the Figure 19) could be precalculated for every score threshold S with the branch-and-bound algorithm: $P_\mathrm{cum}(S) = \sum_{z_\mathrm{r}\,:\,S_\mathrm{r}(z_\mathrm{r})\geq S} P_\mathrm{bg}(z_\mathrm{r})$ (dark grey regions in the figure), which would reduce the computation of $P(x)$ to a single sum,

$$P(x) \approx \sum_{z_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{l}) P_\mathrm{cum}\left(S(x) - S_\mathrm{l}(z_\mathrm{l})\right)\,, \tag{11}$$

making the P-values very efficient to compute.

To include higher-order dependencies between the left and right part of the l-mer, the P-values of the right part have to be evaluated depending on the left part of the l-mer. The exact calculation would be too time-consuming since it would involve calling the branch-and-bound algorithm for each possible z_l. We seek an approximation to equation 9 that looks similar to equation 10,

$$P(x) \approx \sum_{z_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{l}) \sum_{z_\mathrm{r}\,:\,S_\mathrm{r}(z_\mathrm{r})\geq S(x)-S_\mathrm{l}(z_\mathrm{l})} P_\mathrm{bg}(z_\mathrm{r}|\mathrm{PWM})\,. \tag{12}$$

The crucial aspect is that the new term $P_\mathrm{bg}(z_\mathrm{r}|\mathrm{PWM})$ that replaces $P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l})$ does not depend on z_l, because we can then precalculate all sums on the right side as before.

The approximation $P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}) \approx P_\mathrm{bg}(z_\mathrm{r}|\mathrm{PWM})$ should err on the conservative side, overestimating the P-values in order not to report significant, false motifs. If we maximized over all possible z_l, we could get an exact upper bound: $P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}) \leq \max_{z_\mathrm{l}'\in\mathcal{Z}_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}')$. We obtain an approximate yet more stringent upper bound by noting that only l-mers whose left 8 nucleotides match very well to the PWM have a chance of achieving significant P-values at all. For these l-mers, our approximation must be as good as possible. We therefore maximize over the set of 8-mers whose probability according to the PWM is at least γ of the maximum:

$$\mathcal{Z}_\mathrm{l} = \{z_\mathrm{l} : \mathrm{PWM}(z_\mathrm{l}) \geq \gamma \times \max \mathrm{PWM}(z_\mathrm{l}')\}\,. \tag{13}$$

In summary, we use the approximation

$$P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}) \approx P_\mathrm{bg}(z_\mathrm{r}|\mathrm{PWM}) = \max_{z_\mathrm{l}'\in\mathcal{Z}_\mathrm{l}} P_\mathrm{bg}(z_\mathrm{r}|z_\mathrm{l}') \tag{14}$$

together with equation 12. The $P_\mathrm{bg}(z_\mathrm{r}|\mathrm{PWM})$ and their cumulated sums are precalculated, making the computation of P-values very efficient. We tried out values $0.1, 0.5, 0.8, 1.0$ for γ and obtained best results for $\gamma = 0.8$. Above this value, E-values may be unreliable in rare cases while below this value the sensitivity on our small benchmark set decreases slightly.

3.3. Calculating the enrichment *P*-value measuring the significance of a set of binding sites

After a match P-value is calculated for every site given the PWM, it is necessary to find the optimal P-value threshold for considering a site as significant, i. e., for accepting a motif instance as a true positive. This is determined using so-called order statistics. The possible sites are sorted by their match P-value in increasing order (i. e., in order of decreasing significance). Now, for a binomial distribution, the probability of finding exactly K sites in a set of N possible sites with P-values at least as small as p is $\binom{N}{K} p^K (1-p)^{N-K}$. Accordingly, the probability of finding at least K sites with P-values at least as good as the K-th best, P_K, is given by:

$$P^{(K)}_{\text{enrichment}} = \sum_{k=K}^{N} \binom{N}{k} (P_K)^k (1-P_K)^{N-k} \tag{15}$$

The optimal K^* is the one with minimal $P^{(K)}_{\text{enrichment}}$:

$$K^* = \underset{K \in \{1,\ldots,N\}}{\operatorname{argmin}} \{P_{\text{enrichment}}(K)\} \tag{16}$$

The P-value of the most significant set of binding sites is thus $P^{(K^*)}_{\text{enrichment}}$, and the K^* best sites are considered to be functional.

3.3.1. Multiple occurrence per sequence model

Using the multiple occurrence per sequence model (mops model) as the motif model of XXmotif allows in principle to find a motif at every position in the input set. Hence, N, the number of different binding sites, equals the number of nucleotides in the input set M, subtracted by the nucleotides at the sequence ends covered by the motif. Having L sequences and a PWM length W,

$$N = M - L\,(W-1) \tag{17}$$

However, overlapping binding sites are not allowed to be simultaneously occupied as this would give excessively significant P-values for repetitive motifs. Hence, the number of motifs K within the subset of functional motifs has to be smaller than N. Moreover, allowing XXmotif to find motifs on both strands of the DNA increases N by a factor of two, whereas it is prohibited to have a binding site at one strand that overlaps with a binding site of the same motif at the reverse strand. Otherwise, palindromic motifs get too significant P-values, as these overlapping sites would double the number of binding sites of palindromes.

3.3.2. Zero or one occurrence per sequence model

Using the zero or one occurrence per sequence model (zoops model) as the motif model of XXmotif allows at most one binding site per input sequence. Therefore, N is the number of sequences and K can have values between 0 and N, as overlaps are not possible in this setting. Hence, the P-value P_K of a site used in equation 15 now refers to a per sequence P-value, which can be calculated from the site's match P-value p by calculating the probability to find a binding site with length W at least as significant as p within sequence S:

$$P_K = 1 - (1-p)^{|S|-W+1}\,, \tag{18}$$

i. e., the probability of the complementary event of not finding it at any of the $|S|-W+1$ possible starting positions of the PWM in sequence S.

Instead of individual sequence lengths, the geometric mean length is used for all sequences. This avoids problems resulting from equally scoring matches becoming disproportionately significant (or insignificant) if found in very short (or very long) sequences. The geometric mean is adequate since lengths are scaling variables that are best compared in terms of factors, not absolute differences.

3.3.3. One occurrence per sequence model

XXmotif also provides a one occurrence per sequence model. It is implemented by using the same framework as for the zero or one occurrence per sequence model, however, the number of motifs K is not optimized using order statistics, but manually set to N. Subsequently, the final P-value is the likelihood to find N times an instance with the N'th best P-value. As only the N'th best P-value contributes to the final result, this option should only be used if it is known that all sequences contain the motif, otherwise the zero or one occurrence per sequence option is recommended.

3.4. Time complexity for computing a motif enrichment *P*-value

Let N be the number of sequences in the input set, L_{av} their average length, and l the length of the PWM. We first consider the simpler case $l \leq 8$ for which we do not split the l-mers into two parts. The branch-and-bound algorithm to generate the list of M l-mers with a score larger than the cut-off takes time $O(4M)$. The calculation of background probabilities $P_{\text{bg}}(z)$ for all M l-mers z with equation 2 takes time $O(Ml)$. Sorting the resulting list by $P_{\text{bg}}(z)$ and computing a cumulative version of the list takes $O(M \log M)$ and $O(M)$, respectively. Looking up in the cumulated list the P-values for all NL_{av} possible starting positions of the motif takes $O(NL_{\text{av}})$, and sorting the list by P-values takes $O(NL_{\text{av}} \log NL_{\text{av}})$. Computing the order statistics for a given number K of motif occurrences (eq. 15) takes time $O(NL_{\text{av}})$. (Computing the binomial coefficients takes constant time per coefficient, since it can be done iteratively from the coefficient of the previous term.) In summary, the time complexity is

$$O(M \times (l + \log M) + NL_{\text{av}} \log NL_{\text{av}})\,. \tag{19}$$

By splitting the l-mers into pieces of length 8 and $l-8$, we effectively limit the complexity to nearly the one at a PWM length of $\min\{8, l-8\}$.

For comparison, the method of Zhang *et al.* (2007) has a time complexity of $O(Ml + K\,NL_{\mathrm{av}} \times Ml)$. Here, the dominant second term describes the time complexity to compute the probability of observing at least K PWM matches with a score larger than the threshold in a set of input sequences of total length NL_{av} that are distributed according to a Markov background model, and where M is the number of l-mers scoring above the threshold. This is slower than our algorithm by a factor of almost KMl. The huge difference in efficiency to XXmotif's P-value calculation stems from the fact that the method of Zhang *et al.* treats the case of overlapping motif instances in an exact manner, whereas XXmotif simply forbids overlapping motif instances.

The most important advantage of XXmotif's P-value calculation is that for each K it only takes an additional time of order $O(NL_{\mathrm{av}})$ for calculating the new order statistics, since the previously calculated cumulated list of P-values can be used. In contrast, the exact method of Zhang et al. (2007) and approximation by Touzet and Varré (2007) need the same amount of time for each new score threshold and therefore each new value of K. This makes them unsuitable for optimizing PWMs.

3.5. Correcting *P*-values for multiple testing

Finally, multiple testing has to be taken into account: Any PWM in the whole motif space has the same chance to achieve a certain significance by coincidence. Hence, for calculating the E-value which corresponds to the expected number of motifs with a given P-value, a Bonferroni correction is applied. This is a conservative method that multiplies the calculated P-value by the number of different motif models that could be tested. Since in principle infinite slightly different PWMs exist in the motif space, it is necessary to define a parameter N_{eff} which defines the effective number of possible PWM columns. Hence, for a PWM with length W, the respective E-value is calculated as follows:

$$E\text{-value} = P\text{-value}\, N_{\mathrm{eff}}^{W} \tag{20}$$

In the seed enumeration stage of XXmotif, the motif model consists of one out of ten different IUPAC characters per position (A, C, G, T, M, R, W, S, Y, K). However, only the four nucleotides A, C, G, and T are independent, the remaining characters are partly similar to each other. Hence, N_{eff} should be set to a value between four and ten, with $N_{\mathrm{eff}} = 6$ is used as default. In the refinement stage of XXmotif, we use $N_{\mathrm{eff}} = 10$ to account for the strong similarities between different PWM columns, but still capture the higher number of different PWMs than IUPAC strings of the same length. In case of the seed enumeration stage, for which gaps are allowed in the IUPAC string, additionally to the factor N_{eff}^{W}, a factor of two per gap position is used to capture the higher amount of motifs to test if a certain number of gaps is present. To improve the agreement between empirical and reported E-values (Supplemental Fig. 6), we also scale the log E-value down by a factor of two.

3.6. Calculating conservation *P*-values

Like enrichment P-values, conservation P-values are calibrated on the negative set, if available. Otherwise, the input set is used. They are calculated as the probability to find at most m mutations from the first sequence to n other sequences within the alignment, given the frequency of every nucleotide within the site. As this nucleotide composition $c = f_{\text{site}}(A,C,G,T)$ is taken into account, different mutation rates within regions of different A/T content are included.

$$P_{\text{cons}}(m, n, c) = \sum_{i=0}^{m} \frac{f(i,n,c)}{\sum_{j=0}^{\infty} f(j,n,c)} \tag{21}$$

For each site it is tested how many sequences have no gaps in the alignment and the maximal n is used. To preclude that related informative sequences are lost if a closely related sequence was not alignable and therefore has only gaps at the site position, a preprocessing step is used to fill gaps in closely related species with the nucleotides of the a more distantly related species. This procedure can be considered as an upper bound estimation for the mutation within the site.

To calculate a combined conservation P-value for the K best sites according to the PWM, we use the formula for the distribution of the product of independent pairwise P-values given by Bailey and Gribskov (1998):

$$P_{\text{cons}}^{(K)} \approx p \frac{\sum_{i=0}^{K-1} (-\log p)^i}{i!} \tag{22}$$

where p is the product of conservation P-values of the K considered sites:

$$p = \prod_{i=1}^{K} P_{\text{cons},i} \tag{23}$$

3.7. Weighted combination of *P*-values

Given two independent P-values p_1 and p_2, these can be combined to a single P-value using the formula:

$$P_{\text{comb}} = \Pr[P_1 P_2 \leq p_1 p_2] = p_1 p_2 \left(1 - \log\left(p_1 p_2\right)\right) \tag{24}$$

However, this formula for independent P-value combination implicitly assumes that both P-values are similarly important. If the first source of information p_1 is much more important than the second source of information p_2, meaning that most non-random events (TPs) have much smaller p_1's than p_2's, combined P-values can be even worse in distinguishing non-null-model events than the single P-value of the more important source of information.

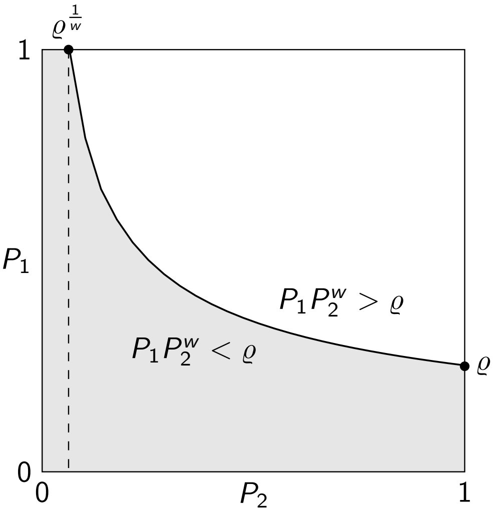
E. g., if $p_1 = 10^{-5}$ and $p_2 = 0.1$, $\quad P_{\text{comb}} = 10^{-6}(1 - \log(10^{-6}) = 1.5 \times 10^{-5} > p_1$

This scenario is very common for XXmotif. Here, enrichment P-values p_1 are combined with conservation P-values p_2 which often are only slightly significant. This low information stems from TF turnover events, which cancel out any conservation information even for functional binding sites, or, if the species are too closely related, even completely conserved binding sites are not significant.

Therefore, it is desirable to assign a weight $w \in]0,1[$ to the source of information which is less important and calculate a P-value for the weighted score $\varrho = p_1 p_2^w$ (see Fig. 20). This can be calculated analytically:

$$P_{\text{comb}} = \Pr[P_1 P_2^w \leq \overbrace{p_1 p_2^w}^{\varrho}] = \varrho^{\frac{1}{w}} + \int_{\varrho^{\frac{1}{w}}}^{1} \frac{\varrho}{P_2^w} \, \mathrm{d}P_2 = \frac{\varrho - w\varrho^{\frac{1}{w}}}{1-w} = \frac{p_1 p_2^w - p_1^{\frac{1}{w}} p_2 w}{1-w}, \tag{25}$$

For the weight $w = 1/3$, which is the default value used in XXmotif, the example calculation from above gives $P_{\text{comb}} = 6.96 \times 10^{-6}$, which is 1.44 times more significant than the single P-value.



Supplementary Figure 20: Weighted combination of P-values. The probability that the product of P_1 and P_2^w is less than ϱ is equal to the shaded area. This in turn is composed of a rectangle with area $\varrho^{1/w}$ and the area under $P_1 P_2^w$ over $[\varrho^{1/w}, 1]$.

3.8. Calculating localization *P*-values

If motif instances cluster together at a fixed distance relative to a specified fixed point, e. g., TSS or nucleosome, motif identification is facilitated by introducing a P-value that captures the differences of this clustering to a random distribution. To decide whether for a motif of size l the instances are significantly clustered within a region Δ, a cluster P-value P_{cl} is calculated using the formula

$$P_{\text{cl}} = \sum_{k=K}^{N} \binom{N}{k} (P_{\text{reg}})^k (1 - P_{\text{reg}})^{N-k} \quad \text{with } P_{\text{reg}} = \frac{|\Delta|}{L - l + 1}$$

where K is the number of motifs within the tested region, N is the number of all motif instances, $|\Delta|$ is the size of the region where the motifs are clustered and L is the length of the sequences. To find the most significant clustering all possible regions Δ are tested and the region with the best P_{cl} is selected if it is significant, i. e., $P_{cl} < 10^{-3}$.

3.8.1. Positional quasi *P*-value

Since P_{reg} cannot be smaller than $1/(L-l+1)$ it is only possible to calculate a quasi P-value, which cannot directly be combined with the enrichment P-value by simply using the formula for combining P-values shown in equation 25.

We define our positional quasi P-value for position z_k and a given cluster region Δ_0 ranging from positions z_s to z_e by

$$p(z_k) = \frac{|\{z : 1 \leq z \leq L-l+1 \wedge |z-\mu| \leq d\}|}{L-l+1}$$

where

$$d = \max\{|z_k - \mu|, D\}$$

and

$$\begin{aligned} \mu &= \frac{z_e + z_s}{2} \\ D &= \left\lceil \frac{\Delta_0}{2} \right\rceil = \left\lceil \frac{z_e - z_s + 1}{2} \right\rceil \end{aligned}$$

To simplify the expression for $p(z_k)$ and see how it depends on μ, D and $L-l+1$ one can first note that

$$p(z_k) = \frac{\lfloor \min\{L-l+1, \mu+d\} \rfloor - \lceil \max\{1, \mu-d\} \rceil + 1}{L-l+1}$$

and write this formula as

$$p(z_k) = \begin{cases} \dfrac{\Delta_0}{L-l+1} & \text{for } |z_k - \mu| \leq D \\ \dfrac{\Delta(z_k)}{L-l+1} & \text{for } |z_k - \mu| \leq D \end{cases} \tag{26}$$

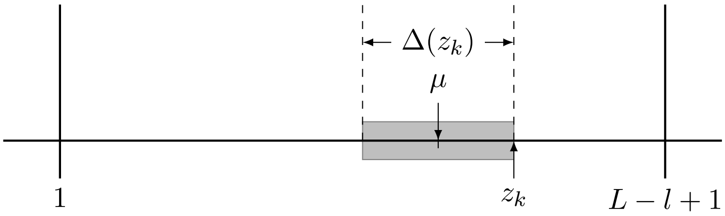
where

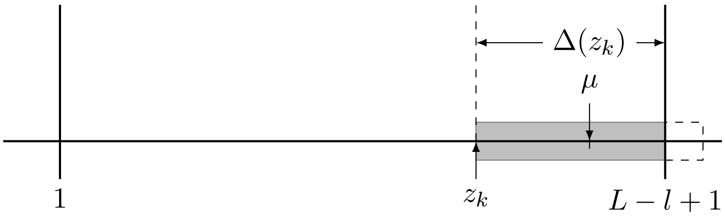
$$\Delta_0 = \lfloor \min\{L-l+1, \mu+D\} \rfloor - \lceil \max\{1, \mu-D\} \rceil + 1$$

and

$$\Delta(z_k) = \lfloor \min\{L-l+1, \mu+|z_k-\mu|\} \rfloor - \lceil \max\{1, \mu-|z_k-\mu|\} \rceil + 1$$

as can be seen for two different values for μ and z_k in the following figure:





Then for $z_k \geq \mu$:

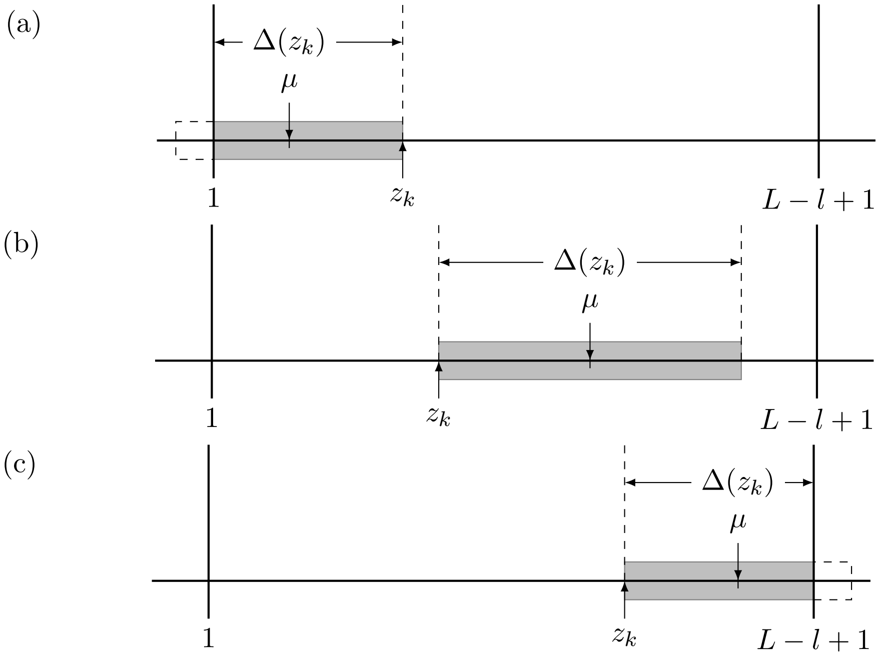
$$\begin{aligned}\Delta(z_k) &= \underbrace{\lfloor \min\{L-l+1, z_k\}\rfloor}_{z_k} - \lceil \max\{1, 2\mu - z_k\}\rceil + 1 \\ &= \begin{cases} \lfloor 2(z_k-\mu)+1\rfloor & \text{for } z_k < 2\mu - 1 \\ z_k & \text{for } z_k \geq 2\mu - 1 \end{cases}\end{aligned} \tag{27}$$

For $z_k \leq \mu$ we obtain:

$$\begin{aligned}\Delta(z_k) &= \lfloor \min\{L-l+1, 2\mu - z_k\}\rfloor - \underbrace{\lceil \max\{1, z_k\}\rceil + 1}_{z_k} \\ &= \begin{cases} \lfloor 2(\mu - z_k)+1\rfloor & \text{for } z_k > 2\mu - L + l - 1 \\ L-l+1-z_k+1 & \text{for } z_k \leq 2\mu - L + l - 1 \end{cases}\end{aligned} \tag{28}$$

We can summarize equations 27 and 28:

$$\Delta(z_k) = \begin{cases} z_k & \text{for } z_k \geq 2\mu - 1 & \text{(a)} \\ \lfloor 2|z_k - \mu| + 1\rfloor & \text{for } 2\mu - L + l - 1 < z_k < 2\mu - 1 & \text{(b)} \\ L-l+1-z_k+1 & \text{for } z_k \leq 2\mu - L + l - 1 & \text{(c)} \end{cases} \tag{29}$$



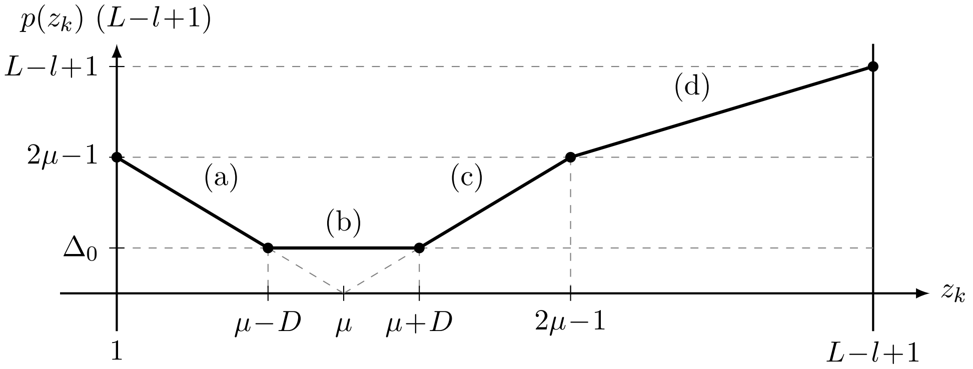
Case (a) can only occur when $L - l + 1 \geq 2\mu - 1$, i. e., $\mu \leq \frac{L-l+2}{2}$

Case (c) can only occur when $2\mu - L + l - 1 \geq 1$, i. e., $\mu \geq \frac{L-l+2}{2}$

To substitute equation 29 into equation 26, we can now distinguish two cases:

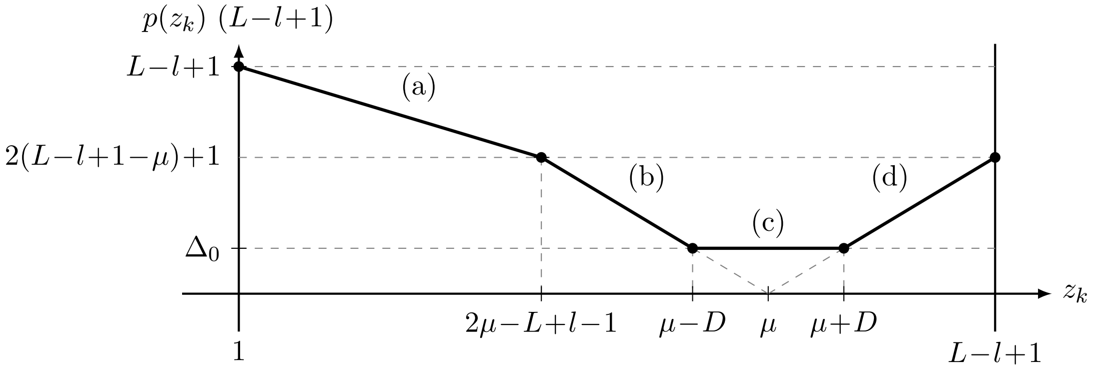
1. $\mu \leq \dfrac{L-l+2}{2}$:

$$p(z_k)\ (L{-}l{+}1) = \begin{cases} \Delta_0 & \text{for } \mu - D \leq z_k \leq \mu + D \quad \text{(b)} \\ z_k & \text{for } 2\mu - 1 \leq z_k \quad \text{(d)} \\ \lfloor 2|z_k - \mu| + 1 \rfloor & \text{for } 1 \leq z_k \leq \mu - D \vee \quad \text{(a)} \\ & \qquad \mu + D \leq z_k \leq 2\mu - 1 \quad \text{(c)} \end{cases}$$



2. $\mu \geq \dfrac{L-l+2}{2}$:

$$p(z_k)\ (L{-}l{+}1) = \begin{cases} \Delta_0 & \text{for } \mu - D \leq z_k \leq \mu + D \quad \text{(c)} \\ L{-}l{+}1{-}z_k{+}1 & \text{for } z_k \leq 2\mu{-}L{+}l{-}1 \quad \text{(a)} \\ \lfloor 2|z_k - \mu| + 1 \rfloor & \text{for } 2\mu{-}L{+}l{-}1 \leq z_k \leq \mu{-}D \vee \quad \text{(b)} \\ & \qquad \mu + D \leq z_k \leq L{-}l{+}1 \quad \text{(d)} \end{cases}$$

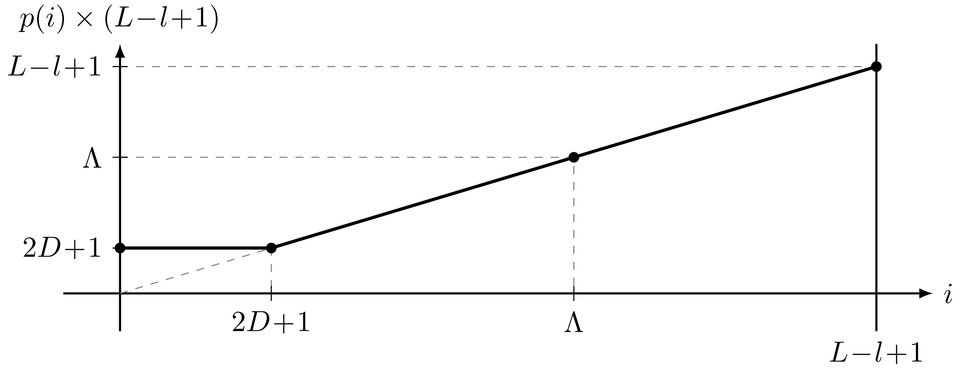


When we sort the values $p(z_k)$ in ascending order (indexed by i), we get

$$p(i) = \left\{\begin{array}{ll} 2D+1 & \text{for } 1 \leq i \leq 2D+1 \\ i & \text{for } 2D+1 \leq i \leq \Lambda \\ i & \text{for } \Lambda \leq L{-}l{+}1 \end{array}\right\} \times (L{-}l{+}1)^{-1}$$

where

$$\Lambda \;\; = \begin{cases} 2\mu - 1 & \text{for } \mu \leq \dfrac{L{-}l{+}2}{2} \\ 2(L{-}l{+}1{-}\mu) + 1 & \text{for } \mu > \dfrac{L{-}l{+}2}{2} \end{cases}$$



3.8.2. *P*-value combination of the positional quasi *P*-value

Suppose our positional P-value $p_2(i)$, which may not be uniformly distributed and therefore is no true P-value, is calculated according to

$$\Delta_0 \; = \lfloor \min\{L-l+1, \mu+D\} \rfloor - \lceil \max\{1, \mu-D\} \rceil + 1$$

$$p_2(i) = \begin{cases} \dfrac{\Delta_0}{L-l+1} & \text{for } |i-\mu| \leq D \\ \dfrac{i}{L-l+1} & \text{for } |i-\mu| > D \end{cases}$$

Suppose the match of the PWM at position $i \in \{1, \ldots, L-l+1\}$ is quantified by a true P-value $p_1(i)$. We would now like to calculate the true combined P-value for $p = p_1(i)\; p_2(i)$ at a specified position i, which is the probability that a better combined P-value could be achieved at any start position in a sequence of length $L-l+1$:

$$P\text{-value}(p) = P\Big(\min_i \{p_1(i)\; p_2(i)\} < p\Big)$$

We start by calculating $1-P$-value(p):

$$\begin{aligned}
P\left(\min_i\{p_1(i)\ p_2(i)\} \geq p\right) &= \prod_{i=1}^{L-l+1} P\left(p_1(i)\ p_2(i) \geq p\right) \\
&= \prod_{i=1}^{L-l+1} P\left(p_1(i)\ \max\left\{\frac{\Delta_0}{L-l+1}, \frac{i}{L-l+1}\right\} \geq p\right) \\
&= \prod_{i=1}^{\Delta_0} P\left(p_1(i) \geq p\ \frac{L-l+1}{\Delta_0}\right) \prod_{i=\Delta_0+1}^{L-l+1} P\left(p_1(i) \geq p\ \frac{L-l+1}{i}\right) \\
&= \left(1-p\ \frac{L-l+1}{\Delta_0}\right)^{\Delta_0} \prod_{i=\Delta_0+1}^{L-l+1}\left(1-p\ \frac{L-l+1}{i}\right)
\end{aligned}$$

⇒ P-value(p) can be calculated as follows:

$$P\text{-value}(p) = 1-\left(1-p\ \frac{L-l+1}{\Delta_0}\right)^{\Delta_0} \exp\left(\sum_{i=\Delta_0+1}^{L-l+1} \log\left(1-p\ \frac{L-l+1}{i}\right)\right)$$

if $p\ \dfrac{L-l+1}{\Delta_0} \leq 0.1$, we can expand the logarithm into a Taylor series:

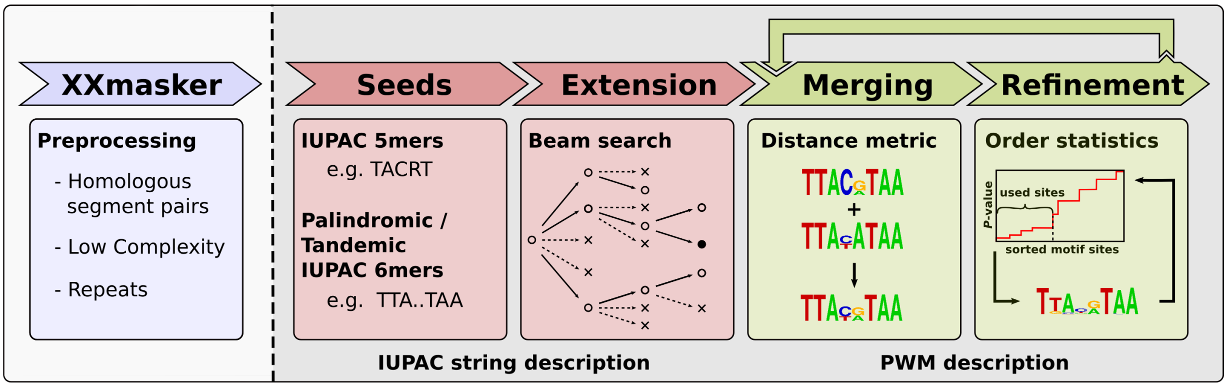
$$\log(1+x) = x+\frac{x^2}{2}+\frac{x^3}{6}+\frac{x^4}{24}+\ldots+\frac{x^k}{k!}+\ldots \quad \text{for } x=-p\ \frac{L-l+1}{i}$$

⇒ by using a first order approximation one finally gets:

$$\begin{aligned}
P\text{-value}(p) = 1 &- \left(1-p\ \frac{L-l+1}{\Delta_0}\right)^{\Delta_0} \\
&\exp\left(-p\ (L-l+1)\ \underbrace{\sum_{i=\Delta_0+1}^{L-l+1}\frac{1}{i}}_{I_1(\Delta_0+1)}\ +\frac{1}{2}p^2(L-l+1)^2\ \underbrace{\sum_{i=\Delta_0+1}^{L-l+1}\frac{1}{i^2}}_{I_2(\Delta_0+1)}\ +\ldots\right) \\
= 1 &- \left(1-p\ \frac{L-l+1}{\Delta_0}\right)^{\Delta_0} \exp\left(\sum_{k=1}^{\infty}\frac{1}{k!}\left(-p\ (L-l+1)\right)^k I_k(\Delta_0+1)\right) \\
\approx 1 &- \left(1-p\ \frac{L-l+1}{\Delta_0}\right)^{\Delta_0} \exp\left(-p\ (L-l+1) \sum_{i=\Delta_0+1}^{L-l+1}\frac{1}{i}\right)
\end{aligned}$$

4. Workflow of XXmotif

In this section we describe the workflow of XXmotif in more detail. In a nutshell, starting with an optional run of XXmasker, IUPAC strings are extended, merged together to obtain a PWM model and refined by optimizing the P-value (see Fig. 21).



Supplementary Figure 21: Overview of XXmotif with its main stages. After an optional step to mask confounding sequence regions (blue), P-values of all 5-mers and gapped palindromic 6-mer seed patterns are evaluated, and the best seeds are recursively extended by an optional gap and motif position (red). Patterns are converted into PWMs and fed to the PWM stage (green). Here, similar PWMs are merged and then iteratively refined by optimizing the motif enrichment P-value. Finally, merging and refinement stages are iterated till convergence.

4.1. XXmasker

XXmotif is more sensitive to reporting false motifs due to duplicated, homologous sequence stretches in the poisitive seuqence set, because XXmotif with its greedy motif extension strategy is able to detect a motif even if it occurrs in only two sequences if the motif is long enough to make it statistically significant. Most other motif discovery methods would not be able to elongate such a motif sufficiently to make it significant. can discover motifs even if they occur only twice in the positive sequences if they are long enough to produce a significant E-value. . However, this high sensitivity becomes problematical if the input sequences contain homologous parts, repeats or low complexity regions. These stretches of DNA are typically longer than 20 nucleotides and occur multiple times, typically as perfect repeats. In order to avoid an assignment of high P-values to these features, we have developed XXmasker, which is an optional tool that masks these sequence regions prior to the main algorithm of XXmotif.

Nucleotides are masked by XXmasker if at least one out of the following three conditions is satisfied:

1. **The nucleotide is within a homologous region:**
   To detect homologous regions, BLAST is used with E-value cutoff 10^{-10} and the soft masking option (`"-F m S"`). For this, a database is incrementally built with all input sequences, where the first input sequences is kept completely and parts of the remaining sequences are masked in all regions in which BLAST detects sufficient homology to any of the already considered sequences. The very stringent E-value cutoff assures that no informative regions are masked, the BLAST masking option avoids that low-complexity segments cause homologous parts to fall below the E-value threshold.

2. **The nucleotide is within a low complexity region:**
   We define a low complexity region to be a DNA stretch of at least 50 nucleotides consisting of at most two different nucleotides.

3. **The nucleotide is within a repeat:**

> We define a repeat region as a DNA stretch of at least 50 nucleotides consisting of perfect repeats with a repeat length in between 3 and 10.

Generally, we chose very strict parameters for all of these conditions to be satisfied. However, in some cases a relaxation of the E-value cutoff, or the masking of imperfect low complexity and repeat regions might be a possibility to further improve the performance.

4.2. Seeds enumeration stage

XXmotif starts by enumerating all 5-mers with at most two IUPAC characters (M, R, W, S, Y, K) as well as all palindromic and tandemic 6-mers with gaps of size $0 - 11$ between the first and last three positions (seeds stage; Fig. 21). For each of these seed patterns, a P-value is calculated using the binomial distribution which is corrected for multiple testing to obtain the corresponding E-value.

To calculate the P-value for an IUPAC string U, the probabilities of all l-mers x matching to U have to be summed up. Therefore, in case of gaps in the IUPAC string, probabilities of l-mers with any nucleotide at these positions have to be considered:

$$P(U) = \sum_{x \in U} P(x)$$

where $P(x)$ is calculated as shown in section 3.1. The probability to find K out of N possible binding sites matching U is calculated using the binomial distribution.

$$P_{\text{enrichment}} = \sum_{k=K}^{N} \binom{N}{k} (P(U))^{k} (1 - P(U))^{N-k} \tag{30}$$

To account for multiple testing, E-values are calculated as described in section 3.5

4.3. Seed extension stage

Seeds are extended using a beam search approach, i. e., not only the one most promising path is followed, but the B most promising paths are examined. This allows for a very efficient extension, while avoiding local minima which may arise more likely by using only the best path.

As all IUPAC degenerations of a non-degenerate seed are highly overlapping and would therefore extend to similar IUPAC strings, it is possible to reduce runtime by extending only a small subset of these. Therefore, we extend only the five most promising degenerate seeds per non-degenerate seed, giving a total of 5120 5-mer (5×4^5), 3840 gapped palindromic and 3840 gapped tandemic 6-mer degenerate seeds ($5 \times 12 \times 4^3$).

All of these seeds are extended individually as long as the E-value improves. Possible extensions are IUPAC characters (A, C, G, T, M, R, W, S, Y, K) at the beginning and the end of the current IUPAC string, allowing gaps of size zero to three. Larger gap sizes are not necessary as it is very unlikely that the extended IUPAC string is more significant than the unextended version (see section 3.5, multiple testing). Extensions having a lower E-value than the unextended version are sorted and the three most significant ones (circles in Fig. 21) are iteratively extended.

Afterwards, extended IUPAC strings are converted into PWMs by calculating the frequencies of every nucleotide within the matching sites.

Since identical IUPAC strings can be reached from different seeds, all extensions are stored in a hash allowing for a fast extraction of already calculated results.

4.4. PWM merging stage

Similar PWMs are merged in order to create a list of non-redundant motifs. First, all motifs are ranked by E-value, and, beginning with the motif having highest significance, similarity tests are performed. Therefore, all less significant motifs are compared to it, and, if similar enough, merged. Afterwards, this procedure is repeated for the second most significant motif and so on. The criteria for these similarity tests are the following:

1. The divergence between the two PWMs calculated with a normalized Euclidian distance is smaller than 0.25 in an overlapping region of at least length six. This criterion is frequently used to assign a motif to be correctly found in diverse benchmarks, e. g., Gordân et al. (2010), Georgiev et al. (2010):

   $$D(a,b) = \frac{1}{\sqrt{2}w} \sum_{i=1}^{w} \sqrt{\sum_{L \in \{A,C,G,T\}} (a_{i,L} - b_{i,L})^2} \tag{31}$$

   where a and b are the regions of both PWMs that are overlapping and w is the size of the overlap.

2. The overlapping region has an average entropy over the six positions with highest information content of at least 0.5 for both PWMs. This assures that the overlap is within important parts of both PWMs. The restriction to only six positions guarantees that uninformative positions within a binding site do not negatively influence the score:

   $$E(a) = \frac{1}{6} \sum_{j=1}^{6} \left(\sum_{L \in \{A,C,G,T\}} a_{j,L} \log_2 \frac{a_{j,L}}{0.25} \right) \tag{32}$$

   where a consists of the six PWM columns of the overlap with highest information content. The use of entropy as a criteria for measuring motif similarity was first described by Gordân et al. (2010).

3. At high merging threshold, the binding sites of both motifs have to overlap at least 40% with the binding sites of the other motif. When the mreging threshold is set to “medium”, the binding sites of the motif with less sites have to overlap at least 40% with the binding sites of the other motif. At low threshold, the fraction of overlaping sites is 20%. This criterion assures that no motifs are merged that are only similar in a small overlapping region but different in the surrounding.

Criteria 1 and 2 were also used in our motif sensitivity and metazoan benchmark in order to determine successful motif discovery.

The merged PWM is built from all binding sites of both PWMs and 10% pseudocounts (Durbin et al., 2006). If the length of both motifs is not the same, the length of the motif with the better E-value is chosen. Afterwards, an E-value is calculated for the merged motif. If this E-value is better than the E-values of both unmerged motifs, only the merged motif is kept. Otherwise, only the better of the original motifs is kept.

4.5. PWM refinement stage

The set of non-redundant motifs is now iteratively refined by selecting the most significant motif instances and motif lengths. To decide which sites are functional, putative binding sites are sorted by P-value. For each K, we calculate the probability for observing by chance at least K binding sites with a P-value equal to or better than the K-th best. The K that optimizes the P-value is used to select the sites contributing to the refined PWM (order statistics, see section 3.3). Afterwards, the PWM is updated using 10% pseudocounts (Durbin et al., 2006) and the refinement step is repeated.

To decide whether a different motif length is more significant, all PWMs including up to two more or fewer positions at both ends are tested. For every tested length, order statistics is used to select the most significant motif set. However, the refined PWM with this new length might influence the sorting of the putative binding sites and thus the P-value. Therefore, two iterations of motif set optimization and PWM creation are performed. Afterwards, the motif length having the best P-value is chosen for a new iteration of the refinement stage. To improve runtime, only sites of the unoptimized PWM with a log-odds score greater zero are tested for the most significant motif set.

The observed P-values are corrected for multiple testing, and the refinement step is repeated as long as the E-value improves. Finally, the merging and refinement stages are iterated until convergence.

5. Overview of the used published motif finders

In this section we give a short overview of the tools used and describe the parameters chosen for all benchmarks. Generally, we used the default parameters of each tool and added useful optional parameters if provided, e. g., the possibility to search on both strands or to use a multiple occurrence per sequence model. The only exception is MEME, for which we used additional arguments suggested within Bailey et al. (2010).

5.1. PRIORITY

PRIORITY (Narlikar et al., 2006) is a PWM-based method that refines the motif model using Gibbs-sampling, a Markov chain Monte Carlo (MCMC) method that approximates sampling from a joint posterior distribution by sampling iteratively from individual conditional distributions (Gelfand and Smith, 1990). PRIORITY uses informative priors based on common structural classes of transcription factors to improve the sampling and provides the opportunity to add more priors to the procedure if more information is available, e. g., a nucleosomal prior (PRIORITY-$\mathcal{N}$, Narlikar et al. (2007)), a discriminative prior (PRIORITY-$\mathcal{D}$, Gordân et al. (2010)), an alignment-free conservation prior (PRIORITY-$\mathcal{C}$, Gordân et al. (2010)), or a combination of these.

PRIORITY can only be run with a zero-or-one occurrence per sequence model and cannot optimize the motif length. As sampling is not deterministic, PRIORITY is run many times and the resulting motif is set to the best of these trials.

We used PRIORITY version 2.1.0 for all benchmarks.

5.1.1. Parameters PRIORITY

PRIORITY was run using default parameters, consisting of the supplied third order background model, the default motif length 8, and 50 trials. From the command line it is started using

```
java -jar priority.jar -nogui
```

5.1.2. Parameters PRIORITY-$\mathcal{D}$

To start PRIORITY using the discriminative prior (Gordân et al., 2010), the corresponding prior has to be created. This can be done using a Perl script supplied by the PRIORITY-package:

```
./discr_from_pos_and_neg.pl 8 input.fasta negset.fasta input.prior
```

Now, we added the directory containing the $\mathcal{D}$-prior to the PRIORITY `params` file and started the tool as before.

5.1.3. Parameters PRIORITY-$\mathcal{DC}$

To start PRIORITY using the discriminative conservation prior (Gordân et al., 2010), at first the sequences of all homologous regions have to be provided in a separate directory (`homologs/`). Afterwards, the $\mathcal{DC}$-prior is created using two Perl scripts supplied by the PRIORITY-package:

```
./generate_fastalike_cons_simple.pl input.fasta homologs/ 8 input.info
./generate_fastalike_cons_simple.pl negset.fasta homologs/ 8 negset.info
./discr_INFO_from_pos_and_neg.pl 8 input.fasta negset.fasta input.info \
         negset.info input.prior
```

PRIORITY is now started as before, by adding the directory containing the $\mathcal{DC}$-prior to the `params` file.

5.2. MEME

MEME (Multiple Em for Motif Elicitation, Bailey and Elkan (1994)) is a PWM-based motif finding tool that iteratively refines candidate PWMs by an expectation maximization (EM) algorithm. It allows to find the optimal motif length and provides the possibility to add higher-order background models and the priors used by PRIORITY to improve the motif search.

MEME was used in version 4.3.0 in our analysis.

5.2.1. Parameters MEME

As suggested by Bailey et al. (2010), depending on the complexity of the organism, two different parameter settings were used:

```
./meme input.fasta -dna -revcomp -mod zoops -minsites 20 -nmotifs 4
```

- Yeast data sets: `-minw 7 -maxw 12`
- Metazoan data sets: `-minw 8 -maxw 20`

5.2.2. Parameters MEME-$\mathcal{M}$

To test MEME with a higher order background model, we trained a fifth order background model using the script `fasta-get-markov` supplied with the MEME-package:

```
./fasta-get-markov -m 5 > background.b
```

Hence, MEME was run using an additional argument:

```
   -bfile background.b
```

5.2.3. Parameters MEME-$\mathcal{D}$

Bailey et al. (2010) demonstrated that the performance of MEME can be further improved by using the discriminative prior from the Hartemink lab (Gordân et al., 2010) as additional information. Therefore, we created the prior file `input.prior` using a Perl script from the Hartemink lab as shown in section 5.1.2.

Afterwards, we used the script `hartemink2psp` supplied with the MEME-package to translate this prior to the `psp` format, which can be used as input for MEME:

```
cat input.prior | hartemink2psp -mod zoops -revcomp -width 8 > input.psp
```

Now, MEME can be run with the discriminative prior using an additional argument:

```
   -psp input.psp
```

5.2.4. Parameters MEME-$\mathcal{DC}$

Furthermore, MEME can be run using conservation information by incorporating the discriminative conservation prior from the Hartemink lab (Gordân et al., 2010) as additional information. To create this prior again Perl scripts from the Hartemink lab have to be used as shown in section 5.1.3.

Now, as for the discriminative prior, the `hartemink2psp` script supplied by the MEME-package is used to translate the prior and MEME is started with the `-psp input.psp` argument.

5.3. Weeder

Weeder (Pavesi and Pesole, 2006) is a pattern-based motif finding tool that exhaustively enumerates the motif space. It tolerates mismatches within the patterns and does not need the exact pattern length as input. Internally, the sequences are represented as a suffix tree, which also allows to efficiently enumerate longer patterns. However, it is not possible to use conservation information.

Weeder was used in version 1.4.2 for our analysis.

5.3.1. Parameters

Weeder was started using the optional arguments `S`, to process both strands of DNA, and `M`, to use a multiple occurrence per sequence model:

```
./weederlauncher.out input.fasta speciescode medium S M
```

where `speciescode` was replaced by the respective two letter code.

5.4. ERMIT

ERMIT (Georgiev et al., 2010) is a pattern-based motif finding tool that incorporates quantitative experimental evidence to find a motif pattern that is enriched in sequences with high evidence values. It starts with IUPAC 5-mers and elongates them as long as their enrichment score improves. To incorporate conservation information, binding sites are filtered to the ones that fit to the pattern in all species.

ERMIT was used in version 1.01 for our analysis.

5.4.1. Parameters ERMIT

To run ERMIT, input files have to be parsed from FASTA format into a special format, and a summary file has to be created (`sequence_file`) that contains the location of the input file. Furthermore, an evidence file has to be created with the probabilities assigned to every sequence identifier, and a summary file is needed (`evidence_file`) containing the location of this evidence file. Now, ERMIT can be started using the command line statement:

```
./cERMIT evidence_file sequence_file output chip_chip
```

where `chip_chip` specifies the data type of the input sequences. As the required evidence values per sequence were only available for the yeast ChIP-chip experiments from the Harbison data set

(Harbison et al., 2004), this data type was always set to `chip_chip`.

5.4.2. Parameters cERMIT

To run ERMIT with conservation information (cERMIT), the homologous regions of an alignment have to be parsed to the file format required by ERMIT and stored separately for each species. Afterwards, the locations of each of these files have to be added to the summary file `sequence_file`. Now, cERMIT can be started as before:

```
./cERMIT evidence_file sequence_file output chip_chip
```

5.5. AMADEUS

AMADEUS (Linhart et al., 2008) is both a pattern- and a PWM-based motif finding tool that starts by enumerating all k-mers of a given length, which are in the following merged depending on their similarity and refined by an EM-algorithm. However, it is neither possible to optimize the motif length, nor to incorporate conservation information.

AMADEUS was used in version 1.0 for our analysis.

5.5.1. Parameters

To run AMADEUS from the command line, a parameter file (`params.txt`) has to be created. Therefore, paths for files with all sequences, input set identifiers and negative set identifiers have to be supplied. We used the default parameters, motif length `8`, and running mode `normal` for our analysis:

```
java -Xmx3000m -jar AmadeusPBM_v1.0.jar file params.txt
```

References

M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* **2000**;*25*(1):25–29.

T. L. Bailey, M. Bodén, T. Whitington, and P. Machanick. The value of position-specific priors in motif discovery using MEME. *BMC Bioinformatics* **2010**;*11*:179.

T. L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc Int Conf Intell Syst Mol Biol* **1994**;*2*:28–36.

T. L. Bailey and M. Gribskov. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics* **1998**;*14*(1):48–54.

R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. Biological sequence analysis. Cambridge University Press, eleventh edition, **2006**.

A. E. Gelfand and A. F. M. Smith. Sampling-Based Approaches to Calculating Marginal Densities. *Journal of the American Statistical Association* **1990**;*85*(410):pp. 398–409.

S. Georgiev, A. P. Boyle, K. Jayasurya, X. Ding, S. Mukherjee, and U. Ohler. Evidence-ranked motif identification. *Genome Biol* **2010**;*11*(2):R19.

R. Gordân, L. Narlikar, and A. J. Hartemink. Finding regulatory DNA motifs using alignment-free evolutionary conservation information. *Nucleic Acids Res* **2010**;*38*(6):e90.

C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J.-B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature* **2004**;*431*(7004):99–104.

D. W. Huang, B. T. Sherman, and R. A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res* **2009**;*37*(1):1–13.

C. Linhart, Y. Halperin, and R. Shamir. Transcription factor and microRNA motif discovery: the Amadeus platform and a compendium of metazoan target sets. *Genome Res* **2008**;*18*(7):1180–1189.

E. Lundberg, L. Fagerberg, D. Klevebring, I. Matic, T. Geiger, J. Cox, C. Algenäs, J. Lundeberg, M. Mann, and M. Uhlen. Defining the transcriptome and proteome in three functionally different human cell lines. *Mol Syst Biol* **2010**;*6*:450.

L. Narlikar, R. Gordân, and A. J. Hartemink. A nucleosome-guided map of transcription factor binding sites in yeast. *PLoS Comput Biol* **2007**;*3*(11):e215.

L. Narlikar, R. Gordân, U. Ohler, and A. J. Hartemink. Informative priors based on transcription factor structural class improve de novo motif discovery. *Bioinformatics* **2006**;*22*(14):e384–e392.

G. Pavesi and G. Pesole. Using Weeder for the discovery of conserved transcription factor binding sites. *Curr Protoc Bioinformatics* **2006**;*Chapter 2*:Unit 2.11.

S. L. Salzberg, A. L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated Markov models. *Nucleic Acids Res* **1998**;*26*(2):544–548.

H. Touzet and J.-S. Varré. Efficient and accurate P-value computation for Position Weight Matrices. *Algorithms Mol Biol* **2007**;*2*:15.

B. van Steensel, J. Delrow, and S. Henikoff. Chromatin profiling using targeted DNA adenine methyltransferase. *Nat Genet* **2001**;*27*(3):304–308.

J. Zhang, B. Jiang, M. Li, J. Tromp, X. Zhang, and M. Q. Zhang. Computing exact P-values for DNA motifs. *Bioinformatics* **2007**;*23*(5):531–537.