

Supplementary Materials for DECOD: Fast and Accurate Discriminative DNA Motif Finding

Peter Huggins^{^1}, Shan Zhong^{^1}, Idit Shiff^{^2}, Rachel Beckerman³, Oleg Laptenko³, Carol Prives³, Marcel H. Schulz¹, Itamar Simon², Ziv Bar-Joseph^{1,*}

¹Lane Center for Computational Biology, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

²Department of Molecular Biology, Hebrew University-Hadassah Medical School, Jerusalem 91120, Israel

³Department of Biological Sciences, Columbia University, New York., N.Y. 10027

[^]Equal contributors.

* Corresponding author: zivbj@cs.cmu.edu

1 Discriminative Motif Finding Methods	1
1.1 DME	1
1.2 DEME	1
1.3 DIPS	2
1.4 CMF	2
1.5 ALSE	2
1.6 Seeder	3
2 Supplementary Methods	3
2.1 Searching for the discriminative PWM that optimizes the target function	3
2.2 Computation of partial derivatives	3
2.3 Speeding up the optimization process	4
2.4 Identifying multiple PWMs representing combinatorial regulation	4
2.5 Simulated data and comparison	5
2.6 Motif discovery on the yeast dataset	6
2.7 Motif discovery on eukaryotic benchmark dataset	7
2.8 ChIP-chip experiment of the p53 binding targets	7
3 Supplementary Results	9
3.1 Robustness of DECOD to parameters	9
3.2 Performance comparison on simulated dataset with longer motifs and recovering motif locations	9
3.3 Performance comparison on benchmark dataset from higher eukaryotes	10
3.4 Results from other discriminative motif finding methods on the P53 dataset	11
3.5 Using DECOD to find motifs from ChIP-seq dataset	12
4 Supplementary Website	12
5 Supplementary References	12
Supplementary Tables	14
Supplementary Figures	19
Appendix	25

1 Discriminative Motif Finding Methods

In contrast to traditional motif finding methods, discriminative motif finding methods requires a negative set of sequences to be supplied and compared against. Several discriminative motif finding methods have been developed. Below we provide a detailed description of the methods that we compared in this work, highlighting the similarities and differences between DECOD and these methods when possible.

1.1 DME (*discriminative matrix enumerator*)

Assuming that each k-mer in the sequences is either motif or background, DME (Smith et al., 2005) uses a likelihood model to score for motif overrepresentation in the positive sequences relative to background sequences. It aims to maximize a target function that represents a modified version of the log ratio of the likelihood of the motif and background models given the positive set to that of the motif and background models given the negative set. And DME uses exhaustive search to find the motif model that maximizes this target function. To improve search efficiency, DME first only searches over a very sparse discrete PWM space in which the columns of the PWMs are only of several representative types. Then DME uses a refinement step to extend the search by including matrices in the neighborhood of the matrix found in the global search. Moreover, instead of using the original log likelihood ratio as the target function, DME uses a modified version which can be calculated very fast with the assumptions that the base frequencies in the positive and negative sequences are close, and that the motif occurrences in the positive sequences are not dense. Our method is similar to DME in that both assumes that each k-mer comes from either a motif or a 0th-order background model, and that both employs global and local searches to improve search speed. However, our method explicitly assumes the probability of the motif and background models being used while DME does not model these. Moreover, our target function is not based on likelihood models but instead based on the expected number of times that the PWM is used in generating the sequences. Furthermore, our method does not make the assumption as made by DME that the base frequencies are similar in the positive and negative sequences, and our method uses deconvolution to take into account the k-mer contexts that DME ignores. Also our method does not search over the PWM space directly but instead searches over the k-mers from which the PWM model is constructed.

1.2 DEME (*discriminatively enhanced motif elicitation*)

DEME (Redhead and Bailey, 2007) also uses a probabilistic approach to model the sequences. Given labeled sequences, DEME aims to find a set of parameters for the data model, including the motif model, the background model, the probability of a positive sequence containing a motif and the prior probability of a sequence being labeled positive, that maximizes a target function describing the conditional log likelihood of the sequence labels given the sequences themselves and the above parameters. DEME also uses a combination of global and local searches in the optimization process. In global search, DEME performs substring search and branch search to find strings from positive sequences, allowing mutations, whose corresponding motif model has the best objective function score. It then uses conjugate gradient to perform local search to further refine the model parameters. One unique feature of DEME is that it is able to work on protein sequences, and it can incorporate prior knowledge about protein residue characteristics by using a Bayesian prior on motif columns. DECOD is similar to DEME in that both uses 0th order background model, and both involves global and local searches in the optimization of the

target function. DECOD takes the probability of motif occurrence in positive sequences as a user-input parameter (by default assuming it to be once per positive sequence), while DEME tries to learn it automatically. Notably, DEME can only find one discriminative motif from an input dataset, whereas DECOD is able to probabilistically remove the signals of a previously found discriminative motif in order to find the next one.

1.3 DIPS (*discriminative PWM search*)

DIPS (Sinha, 2006) assumes that sequences are generated by a 0th order HMM, and it uses a probabilistic score (w-score) to count the occurrences of a PWM in each sequence, which accounts for both the number and strengths of motif occurrences. The w-score of a PWM in a sequence is the sum of the number of times that the PWM is used in all possible parses of the sequence in the HMM model, weighted by the probability of each parse. DIPS then uses the difference in the average w-score of a PWM between sets of positive and negative sequences as the target function. DIPS employs heuristic hill climbing to search for a PWM that maximizes the target function. Our method is very similar to DIPS since both model the sequences by 0th order HMMs, and both aim to maximize the differences in the expected number of times that a PWM is used in generating the positive and negative sequences respectively. Also the search strategy that we used is inspired by DIPS. However, the actual target functions used by the two methods are different. Our method does not calculate the w-scores of a PWM by working on the sequences and calculating the probability of each possible parse according to the HMM model. Instead we work on the k-mers extracted from the sequences directly (Equation (5) in the main text). This makes DECOD much faster than DIPS. Moreover, in the search process we both reduce the search space and reduce the amount of computation involved in calculating the target function score in order to speed up the optimization process particularly for longer motifs, while DIPS did not make these attempts.

1.4 CMF (*contrast motif finder*)

CMF (Mason et al., 2010) is a word-enumeration based method for discriminative motif finding. Given sets of positive and negative sequences, CMF first calculates a z-score of all k-mers to find the k-mers and its neighborhood that are most enriched in the positive compared to the negative set. It then uses the k-mers found to create two count matrices in the positive and negative sequences (representing false positives) respectively to be used as seed, and a PWM is generated by taking their differences. After that, CMF scans k-mers in all sequences in the positive and negative set using the PWM and a 1st-order Markov background model, and calculate a likelihood ratio score for each k-mer. It then applies a threshold on the likelihood ratios by controlling FDR, and all k-mers that pass the threshold are used to create a new PWM. The process is iterated until convergence.

1.5 ALSE (*all sequences*)

ALSE (Leung and Chin, 2006) aims to find a motif model that maximizes a target function that represents the likelihood of the motif being the true one given the input positive and negative sequences. The target function is calculated based on hypergeometric distribution. ALSE first

finds a set of seed matrices using a Voting algorithm, and then iteratively refines the matrices in EM-like iterations until no further improvement can be made in the likelihood function.

1.6 Seeder

Seeder (Fauteux et al., 2008) is a word-enumeration based method that starts by enumerating all k -mers of a given seed length. For each k -mer, the Hamming distance between the k -mer and its best matching subsequence (called ‘substring minimal distance’, SMD) in the positive and negative sets are then calculated respectively. The latter is used to calculate a word-specific background probability distribution, which is in turn used to evaluate the significance of the enrichment of each k -mer in the positive sequences based on the sum of its SMDs to positive sequences. Seed PWMs are generated from matches to the most enriched k -mers in each positive sequence, and the seeds are iteratively extended to form new PWMs of the desired motif width, and then the seeds are updated. The entire process is repeated until convergence.

2 Supplementary Methods

2.1 Searching for the discriminative PWM that optimizes the target function

In order to optimize $F(\theta)$, we adopt a discretized hill climbing approach very similar to DIPS (Sinha, 2006). The search space for θ is restricted to empirical PWMs of the form $\theta(T)$, where T is a subset of m k -mers in S_+ called site set (Sinha, 2006). The subset size m is a parameter called motif cardinality (Sinha, 2006).

Heuristic hill climbing is used to find a local subset T that maximizes $F(\theta(T))$. Each hill climbing step is composed of (i) delete: remove one k -mer from T that contributes the least to the score, and (ii) add: add one k -mer from $S_+ \setminus T$ to T that contributes the most to the score. In the delete step, every possible $t \in T$ is tested to find the t_i that maximizes $F(\theta(T \setminus t_i))$. Then T is updated by setting $T \leftarrow T \setminus t_i$. This step is fast since the size of the PWM set is small (usually 20). In the add step, every k -mer $s \in S_+ \setminus T$ is considered for being added to T . This step is slow since it requires us to loop over all k -mers. To make the calculations faster, partial derivatives are used to estimate $F(\theta(T \cup s))$ as follows:

$$F(\theta(T \cup s)) - F(\theta(T)) \approx \nabla F(\theta(T)) \cdot \delta \quad (9)$$

where $\delta = \theta(T \cup s) - \theta(T)$. Detailed derivation of the partial derivatives is provided below. Choices of s are sorted according to their estimated values of $F(\theta(T \cup s))$. Then $F(\theta(T \cup s))$ is computed exactly for each choice of s in sorted order, until an s is found that satisfies $F(\theta(T \cup s)) > F(\theta(T))$ and T is updated by setting $T \leftarrow T \cup s$. In practice, we terminate the hill climbing search if the top 500 k -mers on the ranked list do not lead to an improvement to the discriminative score.

2.2 Computation of partial derivatives

For a simple motif component \mathbf{Z} , partial derivatives of F can be written succinctly:

$$\frac{\partial F}{\partial \theta_{ij}} = p \cdot (1-p) \cdot \sum_{a \in \Sigma^k} (X_a - Y_a) \frac{a_{ij} \theta_{ij}^{-1} \theta^a B^a}{(p \theta^a + (1-p) B^a)^2} \quad (10)$$

For a convolved motif component \mathbf{Z} , partial derivative can be written as a sum of $2k-1$ terms which are similar to the above. Details are in the Appendix.

2.3 Speeding up the optimization process

Although the running time of DECOD does not depend on the size of the input dataset, it grows exponentially with k , the length of the motif, since the target function (Equation (8), main text) includes a summation over all possible k -mers. Moreover, the search space for the k -mers from which a motif is constructed also grows exponentially with k . In the software implementation, the followings are implemented as an option to speed up the optimization process.

To speed up the calculation of the target function particularly for larger k s, we alternatively first calculate the frequencies of all k -mers in the positive and negative sets, and then the summation in Equation (8) (main text) is calculated only over those k -mers whose frequency differences are more than 2 (if $k < 10$) or 3 (if $k \geq 10$) standard deviations (sd) away (both sides) from the mean of all k -mers, with the underlying assumption that those k -mers whose frequency differences are small are likely to contribute little to the calculation of the target function.

To speed up the search process, we also limit the initial search space to those k -mers whose frequency differences are more than 1 sd away from the mean. Moreover, we perform two rounds of searches in each iteration. The first round is crude search in which we only use the partial derivatives in (9) to estimate the change brought about by adding a k -mer to or removing a k -mer from the motif without doing exact calculation of the target function at all. After a set of m k -mers (m is the motif cardinality) are obtained from the crude search that leads to a motif θ with the maximum target function score at this stage, we expand this set by including all other k -mers that are similar to θ (i.e. in the "neighborhood" of θ). Specifically, the probability of each k -mer given θ is calculated and all k -mers whose probabilities are higher than $0.5^{k/2} \cdot 0.1^{k/2}$ are added to this set. Then a second round of refined search is performed according the optimization process for exact calculations as described in Section 2.4.1, using this set as the new search space. The final motif found by this second round of search is reported.

2.4 Identifying multiple PWMs representing combinatorial regulation

Our hill climbing algorithm assumes (an estimate of) the plant probability p is given. Initially p is estimated by assuming that the motif occurs once per sequence in the positive sequences. In many cases we are interested in finding multiple PWMs in one dataset. For this we need to remove the first PWM identified. To accurately remove a PWM signal from the data, we need to re-estimate p for the estimated mixture component $Z(\hat{\theta})$. Following Equation (1) in the main text we have:

$$X - B \approx p(Z(\hat{\theta}) - B) \quad (11)$$

(with convergence as $n \rightarrow \infty$, if $\hat{\theta}$ is a consistent estimator) since we assume the difference between the observed k -mer counts and the background model results from the PWM. Given our current estimate of $\hat{\theta}$ we can recover p from the above equation. To increase signal, we only use the top 500 k -mers predicted by $Z(\hat{\theta})$ for this computation. Once estimates $\hat{\theta}$ and \hat{p} are determined we assign new values to the number of observed k -mers in the positive and negative sets by setting $X' = X - pZ(\hat{\theta})$ and then all entries are rescaled to sum to 1.

2.5 Simulated data and comparison

For each simulated study, 100 simulated datasets were generated and results were averaged. In each dataset, two groups of positive and negative sequences of length 400bp each were first generated using a multinomial background distribution with equal probabilities for A, C, G and T respectively. Then, in the positive set, palindrome motif(s) of the specified width were planted at randomly chosen positions. The information content of a column (column IC) in the PWM is defined as

$$IC = \sum_{i \in \Sigma} f_i \log_2 \frac{f_i}{b_i} \quad (12)$$

in which $\Sigma = \{A, C, G, T\}$, f_i is the base frequency of nucleotide i in that column of the motif, and b_i is the base frequency of nucleotide i in the background which is always 0.25 in our case. We compared our method with other popular software specifically designed for discriminative motif finding including: ALSE (v1.07, Leung and Chin, 2006), Seeder (v0.01, Fauteux et al., 2008), DME (v2 beta 2008.08.30, Smith et al., 2005), DEME (v1.0, Redhead and Bailey, 2007), DIPS (v1.1, Sinha, 2006) and CMF (Mason et al., 2010), in terms of the accuracy of the recovered motif and the running time needed. For all cases, the accuracy was measured by the average Kullback-Leibler (K-L) divergence per column (AKLD) between the recovered motif and the known planted motif defined as

$$d = \frac{1}{k} \sum_{i=1}^k \sum_{j \in \Sigma} (M_{ij} - M'_{ij}) \log_2 (M_{ij} / M'_{ij}) \quad (13)$$

in which k is the motif length, $\Sigma = \{A, C, G, T\}$, and M_{ij} and M'_{ij} are the corresponding positions in the two motifs being compared (Smith et al., 2005). One position shifting was allowed in calculating AKLD, i.e. when comparing two motifs A and B of length k , three AKLDs were calculated over (i) the full length, (ii) the first $k-1$ columns of A with the last $k-1$ columns of B and (iii) the last $k-1$ columns of A with the first $k-1$ columns of B. The lowest among the three was reported. All methods were run on both strands of the input sequences. For DECOD, on each dataset, both exact and speedup calculations (referred to as "DECOD-exact" and "DECOD-speedup") were run for 50 iterations (the default value) respectively, and the motif with the best discriminative score was reported. The motif cardinality was set to 20 and the probability of motif occurrence was set to once per positive sequence (the default values) for all analyses, unless otherwise noted. For DME, an '-n 200' option was used to allow the program to return many motifs as suggested in its documentation, and for finding bimodal motifs an '-i 0.5' option was used to allow the program to search for column types with information content as low as 0.5. For ALSE, an '-b' option was used to specify the number of motifs to be 1 or 2 accordingly for

each comparison. For CMF, an '-d 1' option was used to set the motif enrichment to be only in the positive sequences and an '-w 6 -l 6 -u 6' option was used to set the length of the motif to be 6. For Seeder, the seed width was set to be 6. For DIPS, we compared running it for both 5 iterations (by default) and 20 iterations (by using '-niter 20') (referred to as "DIPS-5iters" and "DIPS-20iters"). Default values were used for the other parameters for all methods. Running times were measured on a computer cluster with 2x Intel Xeon E5620 CPUs at 2.40Ghz and 24GB RAM.

Detailed descriptions about the motifs planted in generating each simulated dataset as discussed in the main text is given below.

2.5.1 Single unimodal motifs

One motif with a dominating nucleotide at each position was planted. To more closely mimic real cases, noise was added to each position of the motif so that the information contents (IC) of each column of the motif ranges from 2 bits (corresponding to a completely deterministic motif) to 0.64 bits (corresponding to a probability of 0.70 for the dominating nucleotide and 0.10 for each of the other three nucleotides).

2.5.2 Single bimodal motifs

In this case, the IC for the unimodal positions in the planted motif was 1.15, and the IC for the bimodal positions was 0.53 (the dominating two nucleotides had a probability of 0.45 each and the other two nucleotides had a probability of 0.05 each). We generated 1,000 positive and negative sequences respectively, and one instance of the motif was planted in each positive sequence.

2.5.3 Two motifs

Two different motifs having from 0 to 6 bimodal positions (column IC 0.53) and the rest positions being unimodal (column IC 1.15) were planted in each of the positive sequences at different positions. When carrying out the comparisons, each method was set to report the top 2 motifs. The AKLDs of both recovered motifs to the two planted motifs were calculated, and the recovered motif that had the smallest AKLD to either of the planted motifs was reported as Motif 1 (Figure 4 in the main text, upward), and the other was reported as Motif 2 (Figure 4 in the main text, downward).

2.6 Motif discovery on the yeast dataset

For this analysis, probe sequences experimentally determined to be bound by each of the 65 yeast TFs tested in a ChIP-chip assay (Harbison et al., 2004) were downloaded from http://fraenkel.mit.edu/Harbison/release_v24/final_set/Final_Motifs/ and used as the positive dataset for each TF. The numbers of bound sequences for each TF range from 14 to 195 with a median of 56. A consensus motif for each of the 65 TFs was inferred systematically in (Harbison et al., 2004) and they were used as a gold standard to compare against in our analysis. The widths of these motifs range from 6 to 18 with a median of 9. Note that not all bound sequences contained the motif for the corresponding TF (Supplementary Table 2). The probes with highest

binding p-values for each TF as reported in (Harbison et al., 2004) were collected and used as a negative dataset. The number of probe sequences in the negative set is twice the number of those in the positive set for each TF. Then each method studied was run on both strands of the input sequences to search for one motif of the known width for each dataset.

2.7 Motif discovery on eukaryotic benchmark dataset

For this analysis, a benchmark dataset (Tompa et al., 2005) was downloaded which contains binding sites for 52 TFs from yeast, fly, mouse and human as well as negative control sets in which no true TFBSs exist (Matys et al., 2006). We did not include the yeast data in this dataset in our study since we already performed the comparison on Harbison's dataset. For each TF, the sequences containing the motif within the original genomic context (the “real” background type) were used as the positive sequences, and twice as many randomly selected sequences from the other TFs in the same species were used as negative sequences.

The motif width given as input to each motif finding method was specified to be the minimum width of the true binding sites of that TF, as this should be the most informative part of the motif. All methods were set to search for motifs on both strands of the input sequences. For DECOD, only speedup calculation was used. After a motif was found, it was converted to a log-odd scoring matrix using the background frequencies from the positive sequences, and then used to scan both strands of each positive sequence. To allow for flexibility, all k-mers with a score higher than 70% of the maximum possible score for the log-odd scoring matrix were reported to be motif instances (Harbison et al., 2004). For the other methods, motif instances reported in their output files were used directly. The prediction results for all methods were formatted as required and submitted to the server at <http://bio.cs.washington.edu/assessment/> for evaluation. We focused on two metrics: the nucleotide level sensitivity (nSn) and the nucleotide level positive prediction (nPPV). They are defined as follows:

$$nSn = nTP / (nTP + nFN) \quad (14)$$

$$nPPV = nTP / (nTP + nFP) \quad (15)$$

in which nTP is the number of true positive predictions, nFN is the number of false negative predictions and nFP is the number of false positive predictions (all at nucleotide level). See (Tompa et al., 2005) for detailed explanations.

2.8 ChIP-chip experiment of the p53 binding targets

2.8.1 P53 array design

The p53-focused array was designed as previously described (Shaked et al., 2008). The array includes 540 p53-PET sites, 62 additional previously described p53 target regions and 846 randomly chosen promoter regions. Each spot contains PCR product of the designated region with an average length of ~800 bps.

2.8.2 Cell growth and treatments

H1299 tet-off inducible cell lines were created as previously described (Chen et al., 1996). The cells were grown in DME-M (Sigma) supplemented with 10% FCS, 2.5ug/ml tetracycline (Teva), 300 ug/ml G418 (Mercury). The wild-type p53 expressing cells had 2 ug/ml puromycin in the culture medium and the 6KQ/6KR cells were cultured with 100 ug/ml hygromycin (Roche) in the medium. p53 induction was achieved by omitting tetracycline from the medium for 24 hours followed by three washes with PBS and either incubation of (6KQ) cells for 24 hours with 2.5 ng/ml tetracycline or wild-type and 6KR cells with 5 ng/ml tetracycline. The levels of p53 in these three clones were similar to each other as determined by Western blotting (see below) and were also similar to the amount of p53 in HCT116 cells treated with 375 uM 5-fluorouracil for 6 hours.

2.8.3 Western analysis

The cellular lysates were separated on 10% polyacrylamide gel, with equal protein amounts loaded on the gel for each sample, then transferred to a nitrocellulose membrane and incubated with mouse anti-p53 (DO-I; Santa Cruz), goat anti- β -actin (I-19; Santa Cruz) antibodies and horseradish peroxidase-conjugated secondary antibodies. The signal was visualized via enhanced chemiluminescence reaction and exposure to film. See Supplementary Figure 1 for details.

2.8.4 Chromatin immunoprecipitation-on-chip

Chromatin immunoprecipitation (ChIP)-on-chip analysis was performed essentially, as previously described (Lee et al., 2006), using 10 μ g anti-p53 antibody DO-1 (Santa Cruz). Approximately 5×10^7 cells were used. The array was scanned and analyzed with GenePix Pro software, and the fluorescence intensity in both channels was obtained for each spot. As the array is spotted four times, median Cy3 and Cy5 intensities were calculated for each spot. The two channels were normalized according to the median intensity of the random human promoter spots, and the Cy5/Cy3 ratio of each spot was calculated. The experiment was performed in duplicate, and the average binding ratio for each spot was calculated. The significance of the enrichment observed in each spot was determined by calculating the deviation of each ratio from the mean of the random promoters control spots (Z score). Only ~1% of the random promoters obtained Z of >2.5; thus, this cutoff is equivalent to an FDR of 0.01. For gene-specific validation (data not shown), the ChIP assay was performed as described above and the nonamplified immunoprecipitation and input fractions were subjected to 36 cycles of semiquantitative PCR.

For each comparison, sequences identified to be bound by both factors are only put into the negative set. In the WTP53-6KR comparison, there are 81 sequences in the positive set and 255 in the negative set. In the 6KR-6KQ comparison, there are 110 sequences in the positive set and 158 in the negative set. In the 6KQ-control comparison, 147 sequences in the positive set and 36 in the negative set. For motif finding, both strands of the repeat-masked sequences were searched.

3 Supplementary Results

3.1 Robustness of DECOD to parameters

To investigate whether DECOD is sensitive to the choice of the motif occurrence probability (p) and cardinality (C) parameters, we ran DECOD on simulated data using a range of different values for these parameters. We also tested the ability of DECOD to predict motifs longer than 6 ($k > 6$). Similar as before, 1,000 simulated positive and negative sequences, length 400bp each, were generated. One motif with column IC 1.15 was planted once in each positive sequence. DECOD was able to successfully recover the planted motif starting with a p as low as 0.25 or as high as 5 times per positive sequences (Supplementary Table 1A). In reality, some motifs may be more likely to occur more than once in a positive sequence. The insensitivity of DECOD to the value of p suggests that DECOD has the advantage of still being able to correctly recover the motif in such cases. We suggest assuming one occurrence per positive sequence as a starting point. Second, motif cardinality might affect the resolution of the recovered motif. However, for strong motifs as used in our experiments, DECOD works well for C ranging between 5 and 100 (Supplementary Table 1B). Increasing it to 100 does not affect the result much, though it does increase the run time (not shown) since the search process will necessarily take longer time to converge. On the other hand, with a small C the method is more likely to be stuck in a local optima due to the reduced resolution. Therefore we used $C = 20$ in all further analyses which is also the default choice of Sinha in DIPS (Sinha, 2006). In the command-line version of the program (downloadable from the Supplementary Website), both of the above parameters (the probability of motif occurrence and motif cardinality) can be user-specified.

DECOD also works well with longer motifs (Supplementary Table 1C, see also the Result section on yeast data in the main text, Section 3.2 in Supplementary Results on simulated data and Section 3.5 in Supplementary Results on ChIP-seq data). Since the exact calculation includes a summation over all k -mers, the running time using exact calculation increases exponentially with k , and therefore when k is too large (e.g. longer than 10), exact calculation is impractical. However, the speedup calculation does not suffer from this since it only makes use of those k -mers that show the most frequency difference between the positive and negative set (Supplementary Methods and Supplementary Table 1C), and the accuracy of the speedup calculation is comparable in almost all cases to the results from exact calculations (Supplementary Table 1C, Results in the main text [and Supplementary Results.](#))

3.2 Performance comparison on simulated dataset with longer motifs and recovering motif locations

In order to better evaluate the ability of DECOD to find longer motifs, we further generated simulated datasets in which a palindrome motif of length 8 is planted in some (not all) of the positive sequences. There are 100 positive and negative sequences in each dataset, and a known motif of width 8 with column IC 1.15 (the dominating nucleotide had a probability of 0.85 and the rest 0.05) was planted in some (percentage q , ranging from 50% to 90%) of the 100 positive sequences. This setting is basically the same as described in the last paragraph of Section 3.1.1 in the main text except that the motif width is 8 instead of 6. We compared DECOD with the other methods including DME, DEME, CMF, ALSE and Seeder on these datasets. For DECOD since

the motif width is 8, we only tested the performance of DECOD-speedup. All methods were set to search for motifs of width 8 (unless otherwise noted) on both strands of the input sequences. In addition the following parameters were used: For DME, “-n 200 -w 8” was used to allow it to report many motifs of width 8 as suggested in the documentation. For CMF, “-w 7 -l 7 -u 8 -d 1” was used to set the seed length to be 7 and allow the program to find motifs of lengths either 7 or 8; also the motifs were specified to be enriched only in the positive set. For Seeder, the seed width was set to 6 and motif width was set to 8. For ALSE, “-b 1” was used to specify the number of motifs to be 1. Default values were used for other parameters for all methods. We did not include DIPS in this comparison due to its excessive running time (>2hrs for each run). 100 simulated datasets were generated and results were averaged.

We compared the top 1 motif recovered by each method in terms of the AKLD to the known motif (see Equation (13) in Section 2.5 for definition) and each method’s running time. As shown in Supplementary Figure 4A, in all ranges of q tested, the ALKD of the motif that DEME recovered is slightly better than DECOD, which is in turn slightly better than DME. All the above three methods perform much better than the other methods we tested. However, in terms of running time, DEME took about 6 times longer than DECOD (Supplementary Figure 4B).

Moreover, we also compared the ability of each method to recover the known motif locations. For DECOD the motif locations were determined in the way described in Section 2.7. For the other methods we used their predicted list directly. For each q (from 50% to 90%), we looked at the top $100*q$ motif occurrences (since 100 is the number of positive sequences, this is the true number of motif instances planted) predicted by each method, and in accordance with (Tompa et al., 2005) we computed the site-level sensitivities (Sn), positive prediction value (PPV) and the average site performance (ASP) as below:

$$sSn = sTP / (sTP + sFN) \quad (16)$$

$$sPPV = sTP / (sTP + sFP) \quad (17)$$

$$sASP = (sSn + sPPV) / 2 \quad (18)$$

in which sTP is the site-level true positive, sFN is the site-level false negative and sFP is the site-level false positive. In calculating the above measures, one position shift compared with the true location in either direction is allowed. The result is shown in Supplementary Figure 5. Although DECOD has lower sensitivities than DEME and Seeder (when q is high) (Supplementary Figure 5A), it (together with DME) always outperformed the other methods in terms of specificity (Supplementary Figure 5B), and its average site performance (ASP) is comparable with DME and DEME (Supplementary Figure 5C). In conclusion, DECOD works well with longer motifs even when the motif does not exist in all positive sequences, and it is able to accurately recover the correct motif as well as find the occurrences of the motif in the positive sequences very fast.

3.3 Performance comparison on benchmark dataset from higher eukaryotes

To further examine the ability of DECOD to discover motifs in more complex organisms, we tested its performance and compared with the other methods using another benchmark dataset (Tompa et al., 2005). This dataset contains the binding sites for 52 TFs from yeast, fly, mouse

and human as well as negative control sets in which no true TFBSs exist. This is a challenging dataset due to the small number of sequences for each TF. Since we already performed a comprehensive comparison on yeast, we only used the 46 TFs from the other three species in this comparison. For consistency with previous analysis of this data, and because unlike the previous data we used in this case the input data contain motif occurrence information, we evaluate the results for each method in terms of the metrics used by (Tompa et al., 2005), particularly the sensitivity (nSn) and positive prediction values (nPPV), at the nucleotide level (Section 2.7 in Supplementary Methods).

Although the performance for all methods was not great for this subset of the data, at the nucleotide level, DECOD performs better than most of the other methods including DEME, DME and CMF (Supplementary Figure 6). Moreover, the sensitivity (nSn) of DECOD was slightly higher than DEME, DME and Seeder (Supplementary Figure 6). Therefore, DECOD is also competitive with the other methods on recovering TFBS in higher eukaryotes from this dataset. Interestingly, although ALSE performed poorly on the simulated and yeast dataset, it outperforms the other methods on this dataset. Notice that here the nSn and nPPV values we obtained for all methods were generally lower than the results reported in (Tompa et al., 2005) for general (not discriminative) motif finders. This is because in (Tompa et al., 2005) the dataset was sent to the authors of each motif finder software, and many authors performed additional filtering steps, many including inspection by eyes, to improve their predictions. In our scenario we did not attempt to do these because our purpose was only to perform a fair comparison of the methods being evaluated using a consistent standard. We expect that after performing careful post-processing of the motifs reported by each software, the results can be further improved.

3.4 Results from the other discriminative motif finding methods on the P53 dataset

We also run DME and CMF on the P53 dataset and checked whether they could recover the interesting motifs that DECOD found. Each method was set to search for 10 motifs of width 8 in each comparison, and all methods were run on both strands of the repeat-masked input sequences (same as used for DECOD). The results were compared to known motifs in TRANSFAC (Matys et al., 2006) using STAMP (Mahony and Benos, 2007) in the same way we did for DECOD. We did not test DEME and DIPS due to their excessive running time and the fact that DEME is able to find only one discriminative motif. Also ALSE and Seeder were not included since they could not work properly with repeat-masked sequences as in our input dataset.

Both DME and CMF were able to find the motif matching IRF-1 in the 6KR-6KQ comparison (for DME, E-value = $2.59e-6$; for CMF, E-value = $1.46e-7$; Supplementary Table 3). However, DME was not able to recover the known p53 motif from the 6KQ-control comparison. Although CMF was able to find a motif similar to that for p53 in the 6KQ-control comparison, the match was very weak (E-value = $1.30e-4$, Supplementary Table 3) compared with the one recovered by DECOD (E-value = $1.09e-11$, Figure 5D in main text). Neither methods were able to find the motif matching Sox4 that DECOD identified from the WTP53-6KR comparison. The full list of all motifs identified by each method and their matches to known motifs using STAMP is available on the Supplementary Website.

3.5 Using DECOD to find motifs from ChIP-seq dataset

The running time of DECOD does not depend on the size of the input sequences, therefore DECOD is particularly suited for motif finding from data generated by large scale sequencing efforts such as ChIP-seq experiments. Here we demonstrate the ability of DECOD to recover known motifs for transcription factors from several published ChIP-seq dataset from the ENCODE project (Birney et al., 2007). Genomic sequences (peaks) determined to be bound in ChIP-seq studies by the following five TFs that have known motifs in JASPAR (Bryne et al., 2008) were downloaded and used as the positive set for each TF: c-Jun (K562 cells), c-Myc (K562 cells), Max (K562 cells), Egr-1 (K562 cells) and NFkB (GM12878 cells). Each set contains tens of thousands of sequences, and the lengths of the peak regions are typically a few hundreds (see Supplementary Table 4 for details). For negative sequences, we used both the upstream and downstream sequences flanking the peak regions, and the length of each flanking sequence was chosen to be the same as the corresponding peak region. Thus the negative set contains twice as many sequences as the corresponding positive set for each TF. We used DECOD to search for motifs of about the width of the known motif (see details in Supplementary Table 4) on both strands of the input sequences. As shown in Supplementary Table 4, DECOD was able to correctly recover all 5 motifs even for relatively longer and more complex motifs like CTCF. For 4 of the 5 motifs (c-Jun, Max, CTCF and NFkB), the correct motif that DECOD recovered was also the first one reported. For c-Myc, the correct motif was the second motif that DECOD recovered. Therefore, DECOD works well on finding motifs from ChIP-seq datasets in which the number of input sequences can be too big for other motif finding software to handle.

4 Supplementary Website

Software implementation of DECOD and detailed results for each comparison on the p53 dataset is available online at <http://www.sb.cs.cmu.edu/DECOD>

5 Supplementary References

- Birney, E. et al. (2007) Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, **447**, 799-816.
- Bryne, J.C. et al. (2008) JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. *Nucleic Acids Res*, **36**, D102-D106.
- Chen, X. et al. (1996) p53 levels, functional domains, and DNA damage determine the extent of the apoptotic response of tumor cells. *Genes Dev*, **10**, 2438-2451.
- Fauteux, F. et al. (2008) Seeder: discriminative seeding DNA motif discovery. *Bioinformatics*, **24**, 2303-2307.
- Harbison, C.T. et al. (2004) Transcriptional regulatory code of a eukaryotic genome. *Nature*, **431**, 99-104.

- Lee, T.I. et al. (2006) Chromatin immunoprecipitation and microarray-based analysis of protein location. *Nat Protoc*, **1**, 729-748.
- Leung, H.C.M. et al. (2006) Finding motifs from all sequences with and without binding sites. *Bioinformatics*, **22**, 2217-2223.
- Mahony, S. et al. (2007) STAMP: a web tool for exploring DNA-binding motif similarities. *Nucleic Acids Res*, **35**, W253-W258.
- Mason, M.J. et al. (2010) Identification of context-dependent motifs by contrasting ChIP binding data. *Bioinformatics*, **26**, 2826-2832.
- Matys, V. et al. (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res*, **34**, D108-D110.
- Redhead, E. et al. (2007) Discriminative motif discovery in DNA and protein sequences using the DEME algorithm. *BMC Bioinformatics*, **8**, 385.
- Shaked, H. et al. (2008) Chromatin immunoprecipitation-on-chip reveals stress-dependent p53 occupancy in primary normal cells but not in established cell lines. *Cancer Res*, **68**, 9671-9677.
- Sinha, S. (2006) On counting position weight matrix matches in a sequence, with application to discriminative motif finding. *Bioinformatics*, **22**, e454-e463.
- Smith, A.D. et al. (2005) Identifying tissue-selective transcription factor binding sites in vertebrate promoters. *Proc Natl Acad Sci U S A*, **102**, 1560-1565.
- Tompa, M. et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat Biotechnol*, **23**, 137-144.

Supplementary Table 1 Robustness of DECOD to various parameters.

A. Robustness to p (probability of motif occurrence)

Assumed motif occurrence per positive sequence	DECOD-exact	DECOD-speedup
0.25	0.484 \pm 0.000	0.484 \pm 0.000
0.5	0.484 \pm 0.000	0.484 \pm 0.000
1 (truth)	0.484 \pm 0.000	0.484 \pm 0.000
2	0.472 \pm 0.019	0.470 \pm 0.023
5	0.382 \pm 0.031	0.391 \pm 0.041

B. Robustness to C (cardinality, the number of k-mers in the site-set from which the PWM is constructed)

Cardinality (C)	DECOD-exact	DECOD-speedup
5	0.484 \pm 0.000	0.484 \pm 0.000
10	0.483 \pm 0.004	0.481 \pm 0.008
20	0.484 \pm 0.000	0.484 \pm 0.000
50	0.484 \pm 0.000	0.484 \pm 0.000
100	0.484 \pm 0.000	0.484 \pm 0.000

C. Robustness to w , the width of the motif

Motif width (w)	DECOD-exact		DECOD-speedup	
	AKLD	Time(s)	AKLD	Time(s)
6	0.484 \pm 0.000	113.078 \pm 6.896	0.484 \pm 0.000	33.259 \pm 6.442
7	0.474 \pm 0.023	785.812 \pm 184.128	0.469 \pm 0.016	31.597 \pm 2.986
8	0.206 \pm 0.019	8909.048 \pm 1180.864	0.306 \pm 0.026	268.254 \pm 64.414

Supplementary Table 2 Detailed results of each method on the yeast dataset (Harbison et al., 2004).

	DECOD	DME	DEME	CMF	Seeder	ALSE	N_BOUND	%Contain	W	Enrichment
ABF1	+	+	+	+	+		178	86.50%	13	99
CBF1	+	+	+	+		+	195	68.70%	7	99
FHL1	+	+	+	+	+		131	73.30%	10	99
RAP1	+	+	+	+			109	74.30%	10	79.92
REB1	+	+	+	+	+		99	86.90%	7	77.93
UME6	+	+	+	+	+		93	68.80%	8	72.32
RPN4							70	80.00%	9	72.02
GCN4	+	+	+	+	+		143	65.00%	7	64.62
YAP7							101	80.20%	8	62.65
MCM1		+	+				77	66.20%	11	55.28
NRG1				+			108	59.30%	7	45.42
MBP1	+	+	+	+	+		92	47.80%	7	40
SKN7						+	148	37.20%	9	38.79
CIN5	+		+	+			118	42.40%	8	38.36
SUM1	+	+	+		+		51	88.20%	10	36.47
SWI6	+	+	+	+	+		121	51.20%	7	33.62
HSF1	+						74	67.60%	13	32.96
SWI4	+	+	+	+	+		130	49.20%	7	31.96
TYE7	+	+	+	+	+	+	56	53.60%	8	30.56
SFP1							37	73.00%	9	26.64
FKH2	+		+	+	+		91	58.20%	7	26.62
HAP1			+		+		116	28.80%	11	24.72
INO4	+	+	+	+		+	32	68.80%	8	24.15
FKH1	+	+	+	+	+		104	76.90%	8	23.43
CAD1		+	+				29	65.50%	10	21.69
SNT2	+	+	+				20	70.00%	9	21.64
SUT1			+		+		67	37.30%	10	21.01
STE12	+		+		+		142	88.00%	7	20.86
NDD1							94	28.70%	11	20.74
LEU3		+	+				32	40.60%	10	20.45
HAP4	+	+	+	+			54	50.00%	7	20.32
AFT2							76	63.20%	6	19.4
MSN2							74	40.50%	9	18.81
PHD1				+			103	57.30%	8	17.93
YDR026c	+	+	+	+		+	15	86.70%	9	17.26
YAP1		+	+				37	51.40%	9	15.55
THI2							49	28.60%	12	15.38
INO2	+	+	+	+			35	71.40%	7	14.97
SPT2	+						36	63.90%	11	14.4

SIP4							24	37.50%	13	14.36
SIG1			+				16	87.50%	12	13.48
STB5		+	+	+			44	40.90%	9	13.44
GAL4							37	32.40%	18	13.42
RDS1						+	49	24.50%	6	12.65
ZAP1							18	27.80%	14	12.35
SOK2	+						73	64.40%	6	12.28
MET4		+					37	21.60%	15	12.26
STB1	+	+	+	+		+	23	47.80%	9	11.95
GLN3							79	55.70%	7	11.65
RFX1			+			+	25	28.00%	13	11.48
AZF1			+				24	54.20%	18	10.85
RCS1		+					41	46.30%	7	10.44
PHO2							14	50.00%	11	10.2
IME1							36	61.10%	11	9.92
RLR1	+	+					25	64.00%	12	9.76
PDR1		+					68	22.10%	11	9.21
PHO4							24	45.80%	7	9.17
DIG1							66	54.50%	7	8.74
TEC1	+	+	+	+		+	37	78.40%	7	6.4
BAS1	+	+	+	+		+	17	52.90%	6	4.99
SPT23							45	91.10%	8	4.79
ACE2		+				+	71	28.20%	7	4.78
STB4							28	21.40%	9	3.69
DAL82							62	41.90%	6	3.33
GAT1							49	36.70%	6	2.25
Total(Top)	15	13	15	14	11	3				
Total(All)	28	31	34	24	17	9				



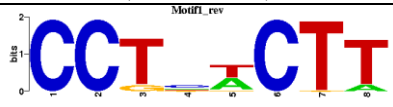



N_BOUND: Number of probes bound by the TF in the ChIP-chip experiment

%Contain: The percentage of the bound probes containing the motif of the TF

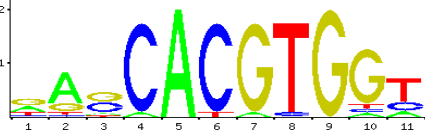
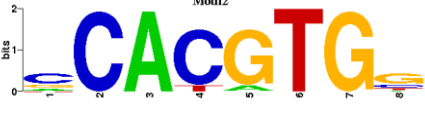

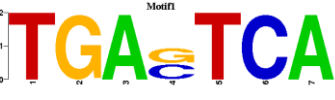
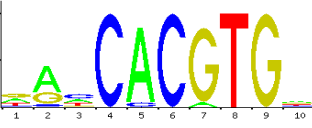

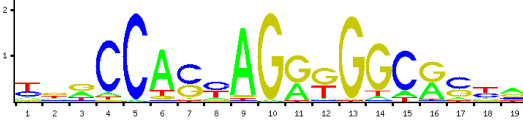



W: The width of the motif

Enrichment: The enrichment score of the motif (Harbison et al., 2004)

Supplementary Table 3 Motifs recovered by DECOD and CMF on the P53 dataset (see Supplementary Website for full list)

Method	Comparison	Motif recovered by the method	Known motifs in TRANSFAC	Match E-value
DME	6KR-6KQ	 <p>DME10_rev CTTCA_TTT (10th motif)</p>	 <p>IRF1_M00747 TCATT IRF1_M00747</p>	2.59e-6
CMF	6KR-6KQ	 <p>Motif1_rev CCTCTTT (1st motif)</p>	 <p>IRF1_M00747 TCATT IRF1_M00747</p>	1.46e-7
CMF	6KQ-control	 <p>Motif4 AATTT (4th motif)</p>	 <p>p53_M00272 GACTGTC P53_M00272</p>	1.30e-4

Supplementary Table 4 Using DECOD to find motifs from ChIP-seq datasets

TF	#Seqs ¹	Median length	Known motif (JASPAR)	Recovered motif (DECOD)
c-Myc (K562)	15479	599	 <p>MA0059.1</p>	 <p>(Search for width 8, 2nd)</p>
c-Jun (K562)	26920	384	 <p>MA0099.2</p>	 <p>(Searching for width 7, 1st)</p>
Max (K562)	10480	395	 <p>MA0058.1</p>	 <p>(Searching for width 8, 1st)</p>
CTCF (K562)	64387	151	 <p>MA0139.1</p>	 <p>(Searching for width 12, 1st)</p>
NFkB (GM12878)	38559	493	 <p>MA0105.1</p>	 <p>(Searching for width 11, 1st)</p>

¹ ChIP-seq peak regions were downloaded from the following URLs:

c-Myc:

<http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeYaleChIPseq/wgEncodeYaleChIPseqPeaksK562CmycV2.narrowPeak.gz>

c-Jun:

<http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeYaleChIPseq/wgEncodeYaleChIPseqPeaksK562CjunV2.narrowPeak.gz>

Max:

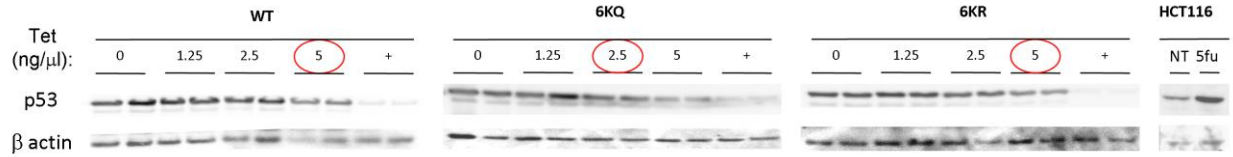
<http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeYaleChIPseq/wgEncodeYaleChIPseqPeaksK562MaxV2.narrowPeak.gz>

CTCF:

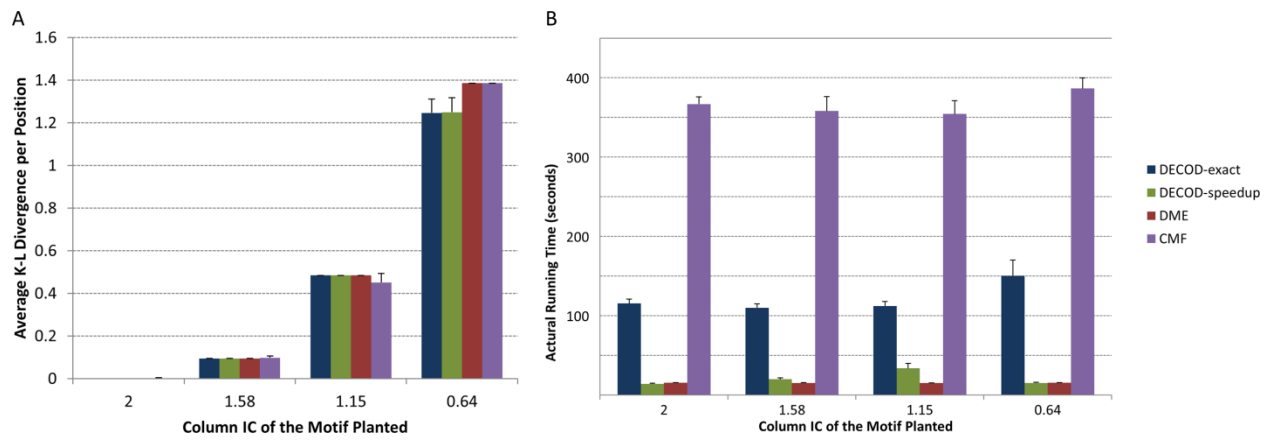
<http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeUwChIPSeq/wgEncodeUwChIPSeqPeaksRep1K562Ctcf.narrowPeak.gz>

NFkB:

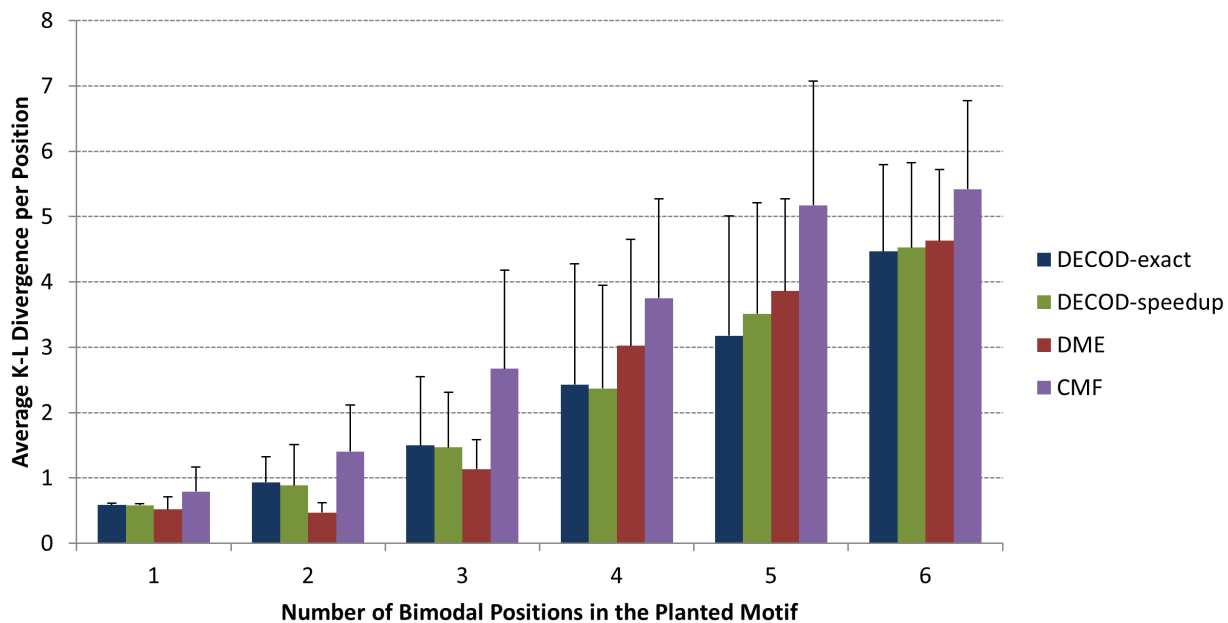
<http://hgdownload.cse.ucsc.edu/goldenPath/hg18/encodeDCC/wgEncodeYaleChIPseq/wgEncodeYaleChIPseqPeaksGm12878NfkbTnfa.narrowPeak.gz>



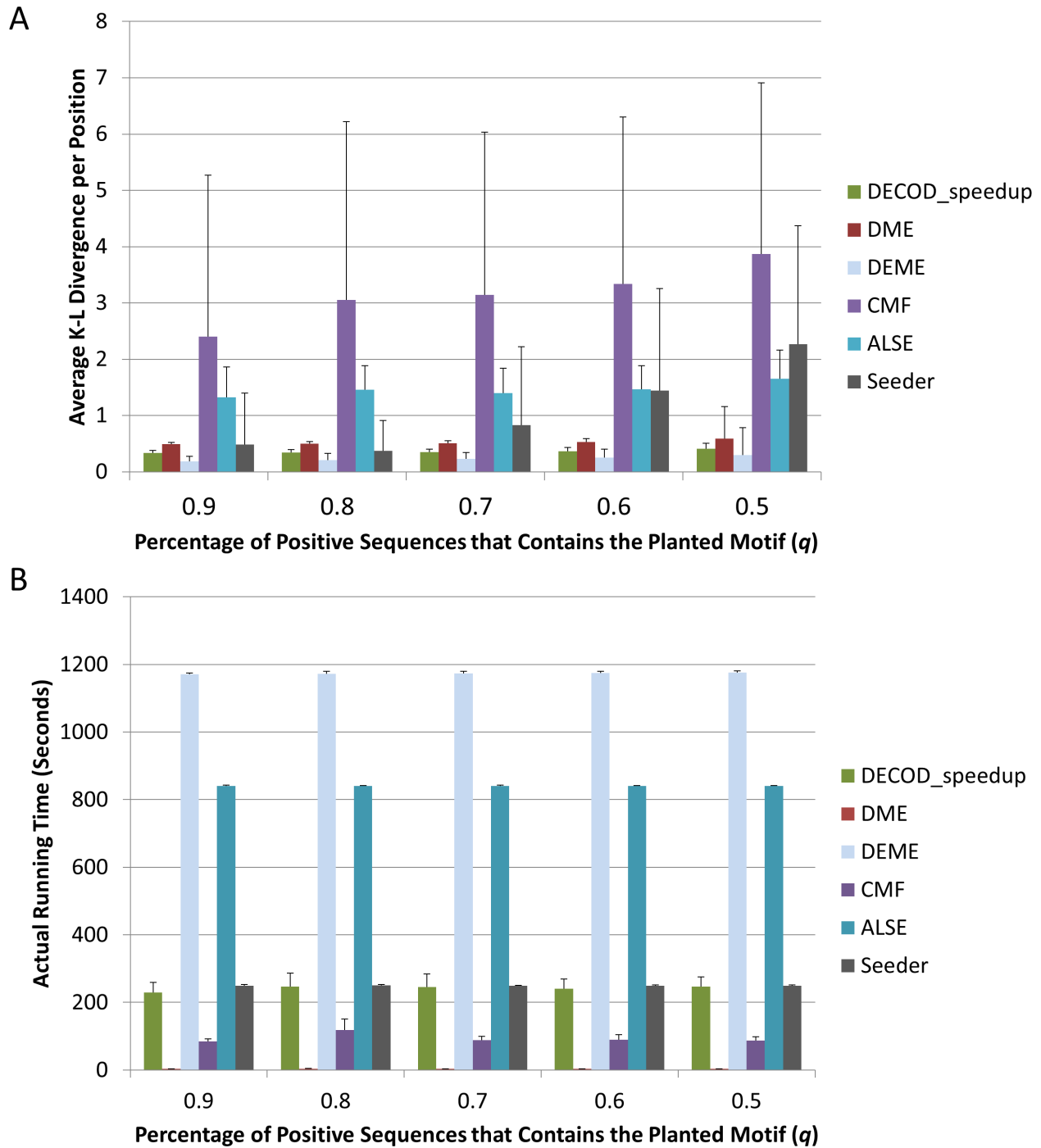
Supplementary Figure 1: H1299 cells with inducible wild type and mutant p53. Western analysis of p53 and beta actin protein levels in WT p53, 6KR p53 and 6KQ p53 H1299 cell clones induced with different concentrations of tetracycline (0-5 ng/ml tet). Cells treated with high amounts of tetracycline (2500ng/ml) to shut off p53 expressions are designated by a plus sign. The final concentration of tet used for induction of p53 in the CHIP analysis is 5 ng/ml for wt and 6KR mutant containing H1299 cells and 2.5 ng/ml for 6KQ mutant containing cells. HCT116 cells without treatment (NT), or treatment with 5-urouracil (5FU) are shown for comparison of p53 protein levels.



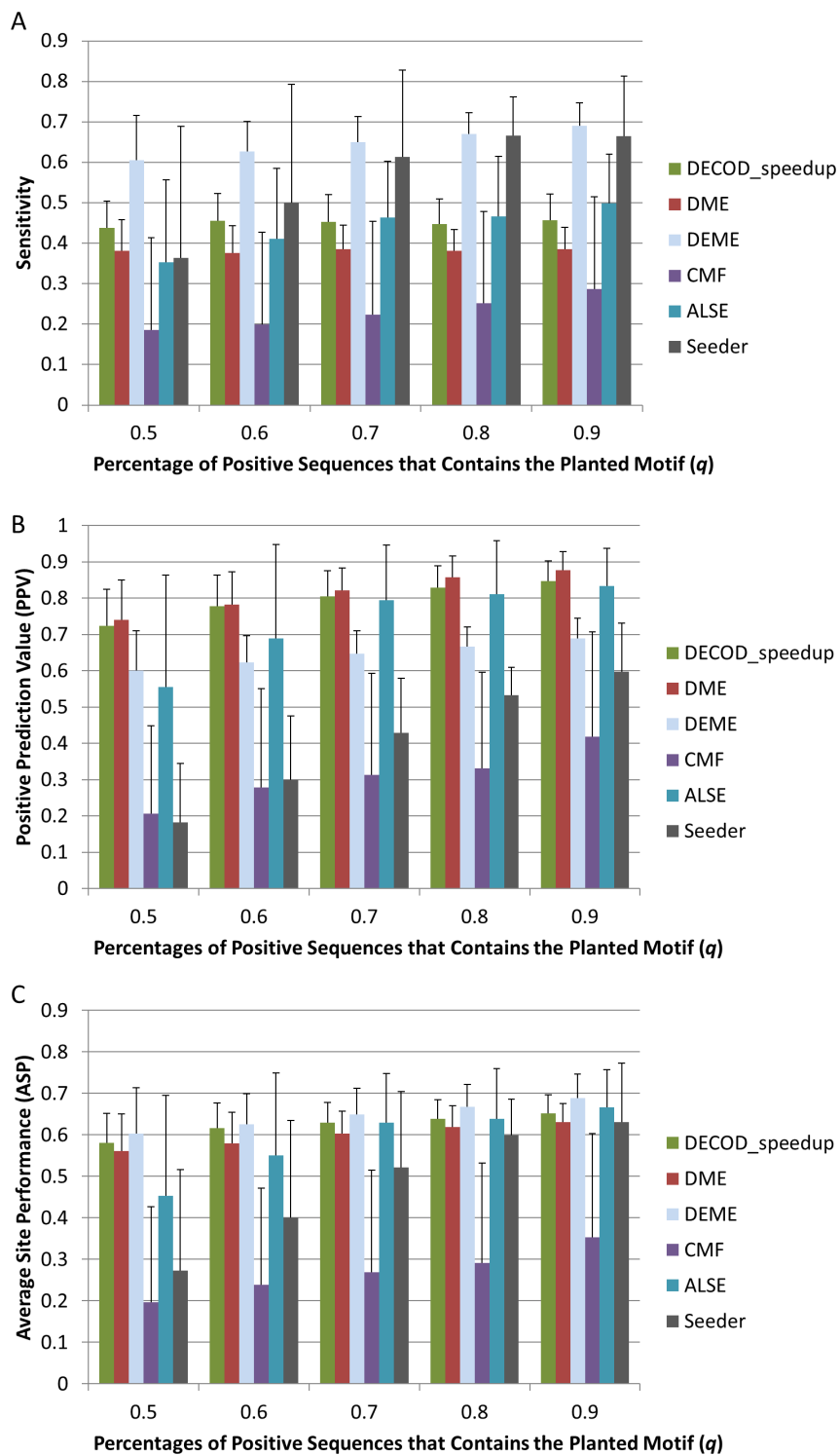
Supplementary Figure 2 Performance comparison on the simulated data planting one motif in each of the 1000 positive sequences. (A) Accuracy as measured by AKLD (B) Actual running time (seconds)



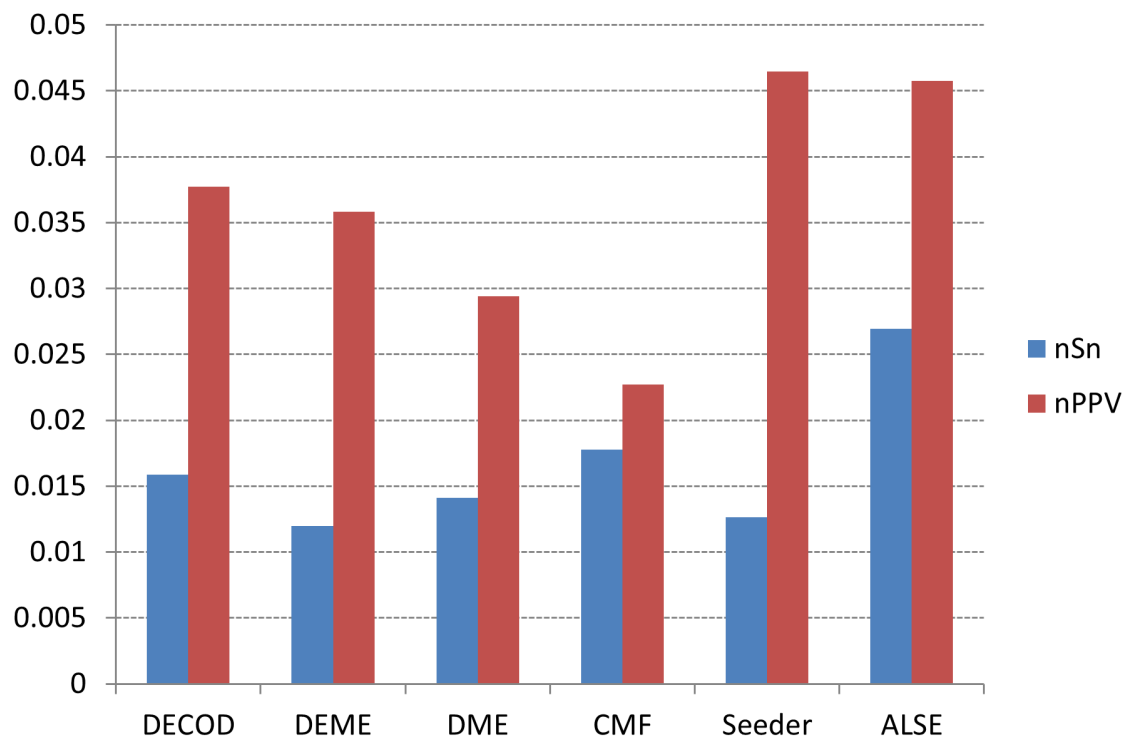
Supplementary Figure 3 Performance comparison on recovering bimodal motifs. A motif containing bimodal positions is planted in each of the 1000 positive sequences (see details in Supplementary Methods)



Supplementary Figure 4 Performance comparison on simulated dataset in which a motif of width 8 was planted in some (percentage q , ranging from 50% to 90%) of the 100 positive sequences. (A) Accuracy as measured by AKLD (B) Actual running time (seconds)



Supplementary Figure 5 Comparison of the (A) site-level sensitivity, (B) PPV and (C) ASP of all methods on identifying the true locations of the planted motifs in the same simulated study as Supplementary Figure 4.



Supplementary Figure 6 Sensitivity (nSn) and positive prediction value (nPPV) at the nucleotide level for each method on the fly, mouse and human TFBS benchmark dataset (Tompa et al., 2005)

Appendix: Calculating the derivative for the deconvolved mixture component

1. The definition of $F(\theta)$ in the convolved mixture model is

$$F(\theta) = \sum_{a \in \Sigma^k} c(a) \frac{pA}{pA + [1 - (2k - 1)p]B^a} \quad (1)$$

$$= \sum_{a \in \Sigma^k} c(a) \left[1 - \frac{[1 - (2k - 1)p]B^a}{pA + [1 - (2k - 1)p]B^a} \right] \quad (2)$$

in which Σ^k denotes all possible k -mers, $c(a) := X(a) - Y(a)$ is the frequency difference of the k -mer a in the positive and negative sequences, p is the probability of the motif occurrence, k is the motif length, B is the background model, and

$$A := \theta^a + ([\underline{B}_1 \bar{\theta}_{k-1}]^a + \cdots + [\underline{B}_{k-1} \bar{\theta}_1]^a) + ([\underline{\theta}_1 \bar{B}_{k-1}]^a + \cdots + [\underline{\theta}_{k-1} \bar{B}_1]^a) \quad (3)$$

$$:= X + Y + Z \quad (4)$$

in which θ is a $4 \times k$ PWM matrix of the motif with columns sum to 1. We represent the k -mer a also as a $4 \times k$ matrix, each element $a_{ij} \in \{0, 1\}$ and the columns sum to 1. $[\underline{P}_j \bar{Q}_{k-j}]$ denote the PWM obtained by taking the last j columns from the PWM P and the first $k - j$ columns from the PWM Q . We regard the background model B as a PWM also with all columns equal. We use θ^a as a shorthand for $\Pr(a|\theta)$.

2.

$$\frac{\partial F(\theta)}{\partial \theta_{mn}} = \sum_{a \in \Sigma^k} c(a) \frac{[1 - (2k - 1)p]B^a \cdot p}{(pA + [1 - (2k - 1)p]B^a)^2} \cdot \frac{\partial A}{\partial \theta_{mn}} \quad (5)$$

$$= p \cdot [1 - (2k - 1)p] \cdot \sum_{a \in \Sigma^k} c(a) \frac{B^a}{(pA + [1 - (2k - 1)p]B^a)^2} \cdot \frac{\partial A}{\partial \theta_{mn}} \quad (6)$$

$$\frac{\partial A}{\partial \theta_{mn}} = \frac{\partial X}{\partial \theta_{mn}} + \frac{\partial Y}{\partial \theta_{mn}} + \frac{\partial Z}{\partial \theta_{mn}} \quad (7)$$

3. For X ,

$$X = \theta^a \quad (8)$$

$$= \prod_{i=1}^k \left(\sum_{j=1}^4 \theta_{ji} a_{ji} \right) \quad (9)$$

$$\frac{\partial X}{\partial \theta_{mn}} = \prod_{\substack{i=1 \\ i \neq n}}^k \left(\sum_{j=1}^4 \theta_{ji} a_{ji} \right) \cdot a_{mn} \quad (10)$$

$$(11)$$

4. For Y ,

$$Y = [B_1 \bar{\theta}_{k-1}]^a + \cdots + [B_{k-1} \bar{\theta}_1]^a \quad (12)$$

$$[B_1 \bar{\theta}_{k-1}]^a = \prod_{i=1}^1 \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=2}^k \left(\sum_{j=1}^4 \theta_{j,i-1} a_{ji} \right) \quad (13)$$

$$[B_2 \bar{\theta}_{k-2}]^a = \prod_{i=1}^2 \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=3}^k \left(\sum_{j=1}^4 \theta_{j,i-2} a_{ji} \right) \quad (14)$$

$$\vdots \quad (15)$$

$$[B_{k-1} \bar{\theta}_1]^a = \prod_{i=1}^{k-1} \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=k}^k \left(\sum_{j=1}^4 \theta_{j,i-(k-1)} a_{ji} \right) \quad (16)$$

There are $k - 1$ rows above. The last $n - 1$ rows do not contain θ_{mn} so the partial derivative of them with respect to θ_{mn} for these rows will be 0. For the first $k - n$ rows,

$$\frac{\partial [B_1 \bar{\theta}_{k-1}]^a}{\partial \theta_{mn}} = \prod_{i=1}^1 \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=2 \\ i \neq n+1}}^k \left(\sum_{j=1}^4 \theta_{j,i-1} a_{ji} \right) \cdot a_{m,n+1} \quad (17)$$

$$\frac{\partial [B_2 \bar{\theta}_{k-2}]^a}{\partial \theta_{mn}} = \prod_{i=1}^2 \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=3 \\ i \neq n+2}}^k \left(\sum_{j=1}^4 \theta_{j,i-2} a_{ji} \right) \cdot a_{m,n+2} \quad (18)$$

$$\vdots \quad (19)$$

$$\frac{\partial [B_{k-n} \bar{\theta}_n]^a}{\partial \theta_{mn}} = \prod_{i=1}^{k-n} \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=k-n+1 \\ i \neq k}}^k \left(\sum_{j=1}^4 \theta_{j,i-(k-n)} a_{ji} \right) \cdot a_{m,k} \quad (20)$$

Thus,

$$Y = \sum_{l=1}^{k-1} \left[\prod_{i=1}^l \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=l+1}^k \left(\sum_{j=1}^4 \theta_{j,i-l} a_{ji} \right) \right] \quad (21)$$

$$(22)$$

and

$$\frac{\partial Y}{\partial \theta_{mn}} = \sum_{l=1}^{k-n} \left[\prod_{i=1}^l \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=l+1 \\ i \neq l+n}}^k \left(\sum_{j=1}^4 \theta_{j,i-l} a_{ji} \right) \cdot a_{m,l+n} \right] \quad (23)$$

5. For Z ,

$$Z = [\theta_1 \bar{B}_{k-1}]^a + \cdots + [\theta_{k-1} \bar{B}_1]^a \quad (24)$$

$$[\theta_1 \bar{B}_{k-1}]^a = \prod_{i=1}^1 \left(\sum_{j=1}^4 \theta_{j,k+i-1} a_{ji} \right) \cdot \prod_{i=2}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \quad (25)$$

$$[\theta_2 \bar{B}_{k-2}]^a = \prod_{i=1}^2 \left(\sum_{j=1}^4 \theta_{j,k+i-2} a_{ji} \right) \cdot \prod_{i=3}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \quad (26)$$

$$\vdots \quad (27)$$

$$[\theta_{k-1} \bar{B}_1]^a = \prod_{i=1}^{k-1} \left(\sum_{j=1}^4 \theta_{j,k+i-(k-1)} a_{ji} \right) \cdot \prod_{i=k}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \quad (28)$$

There are $k - 1$ rows above. The first $k - n$ rows do not contain θ_{mn} so the partial derivative of them with respect to θ_{mn} for these rows will be 0. For the last $n - 1$ rows,

$$\frac{\partial [\theta_{k-n+1} \bar{B}_{n-1}]^a}{\partial \theta_{mn}} = \prod_{i=k-n+2}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=1 \\ i \neq 1}}^{k-n+1} \left(\sum_{j=1}^4 \theta_{j,k-(k-n+1)+i} a_{ji} \right) \cdot a_{m,1} \quad (29)$$

$$\frac{\partial [\theta_{k-n+2} \bar{B}_{n-2}]^a}{\partial \theta_{mn}} = \prod_{i=k-n+3}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=1 \\ i \neq 2}}^{k-n+2} \left(\sum_{j=1}^4 \theta_{j,k-(k-n+2)+i} a_{ji} \right) \cdot a_{m,2} \quad (30)$$

$$\vdots \quad (31)$$

$$\frac{\partial [\theta_{k-1} \bar{B}_1]^a}{\partial \theta_{mn}} = \prod_{i=k}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=1 \\ i \neq n-1}}^{k-1} \left(\sum_{j=1}^4 \theta_{j,k-(k-1)+i} a_{ji} \right) \cdot a_{m,n-1} \quad (32)$$

Thus,

$$Z = \sum_{l=1}^{k-1} \left[\prod_{i=l+1}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=1}^l \left(\sum_{j=1}^4 \theta_{j,k-l+i} a_{ji} \right) \right] \quad (33)$$

and

$$\frac{\partial Z}{\partial \theta_{mn}} = \sum_{l=k-n+1}^{k-1} \left[\prod_{i=l+1}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=1 \\ i \neq n-k+l}}^l \left(\sum_{j=1}^4 \theta_{j,k-l+i} a_{ji} \right) \cdot a_{m,n-k+l} \right] \quad (34)$$

6. In summary,

$$A = X + Y + Z \quad (35)$$

$$\begin{aligned} &= \prod_{i=1}^k \left(\sum_{j=1}^4 \theta_{ji} a_{ji} \right) + \\ &\sum_{l=1}^{k-1} \left[\prod_{i=1}^l \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=l+1}^k \left(\sum_{j=1}^r \theta_{j,i-l} a_{ji} \right) \right] + \\ &\sum_{l=1}^{k-1} \left[\prod_{i=l+1}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{i=1}^l \left(\sum_{j=1}^4 \theta_{j,k-l+i} a_{ji} \right) \right] \end{aligned} \quad (36)$$

and

$$\begin{aligned} \frac{\partial A}{\partial \theta_{mn}} &= \prod_{\substack{i=1 \\ i \neq n}}^k \left(\sum_{j=1}^4 \theta_{ji} a_{ji} \right) \cdot a_{mn} + \\ &\sum_{l=1}^{k-n} \left[\prod_{i=1}^l \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=l+1 \\ i \neq l+n}}^k \left(\sum_{j=1}^4 \theta_{j,i-l} a_{ji} \right) \cdot a_{m,l+n} \right] + \\ &\sum_{l=k-n+1}^{k-1} \left[\prod_{i=l+1}^k \left(\sum_{j=1}^4 b_j a_{ji} \right) \cdot \prod_{\substack{i=1 \\ i \neq n-k+l}}^l \left(\sum_{j=1}^4 \theta_{j,k-l+i} a_{ji} \right) \cdot a_{m,n-k+l} \right] \end{aligned} \quad (37)$$