Supplementary material for

"Evaluation of methods for modeling transcription-factor sequence specificity"

(Weirauch *et al.* 2013, Nature Biotech)

Additional supplementary files, including best-performing PWMs for each TF, source code for select algorithms, PBM experimental data, and large supplementary tables are available at

http://hugheslab.ccbr.utoronto.ca/supplementary-data/DREAM5/

Table of Contents

Supplementary Note 1. DREAM5 challenge evaluation criteria.....3

Supplementary Note 2. 35-mer vs 8-mer scoring.....4

Supplementary Note 3. Effect of array data pre-processing steps on algorithms.....9

Supplementary Note 4. Examination of the relative difficulty of each TF for each algorithm.....15

Supplementary Note 5. Evaluation of sequence scanning methods using PWMs.....21

Supplementary Note 6. Secondary motifs, by category.....24

Supplementary Note 7. Evaluation of PBM-trained models on other experimental platforms.....34

Supplementary Note 8. Description of biophysical models, and comparison to log-odds scoring.....39

Supplementary Note 9. Full descriptions of algorithms.....43

Supplementary Table 1. Information on transcription factors and associated experiments.....75

Supplementary Table 2. Results of the original DREAM5 challenge.....75

Supplementary Table 3. Full evaluations for all algorithms, by TF.....75

Supplementary Table 4. Improvement of dinucleotide model over PWM model, for each TF.....76

Supplementary Table 5. Summary of evaluation of secondary motifs, compared to only using primary PWMs.....79

Supplementary Table 6. Improvement of secondary motifs over primary motifs, for each TF.....79

Supplementary Table 7. Full Comparison to ChIP-seq and ChIP-exo data.....79

Supplementary Table 8. Information on plasmids used for PBMs in this study.....79

Supplementary Figure 1. The effect of using different combinations of evaluation schemes on the final scores of the algorithms.....80

Supplementary Figure 2. Correlation of algorithm predictions.....81

Supplementary Figure 3. Comparison of dinucleotide and secondary motif improvement.....82

Supplementary Figure 4. PWM sequence logo comparisons.....83

Supplementary References.....86

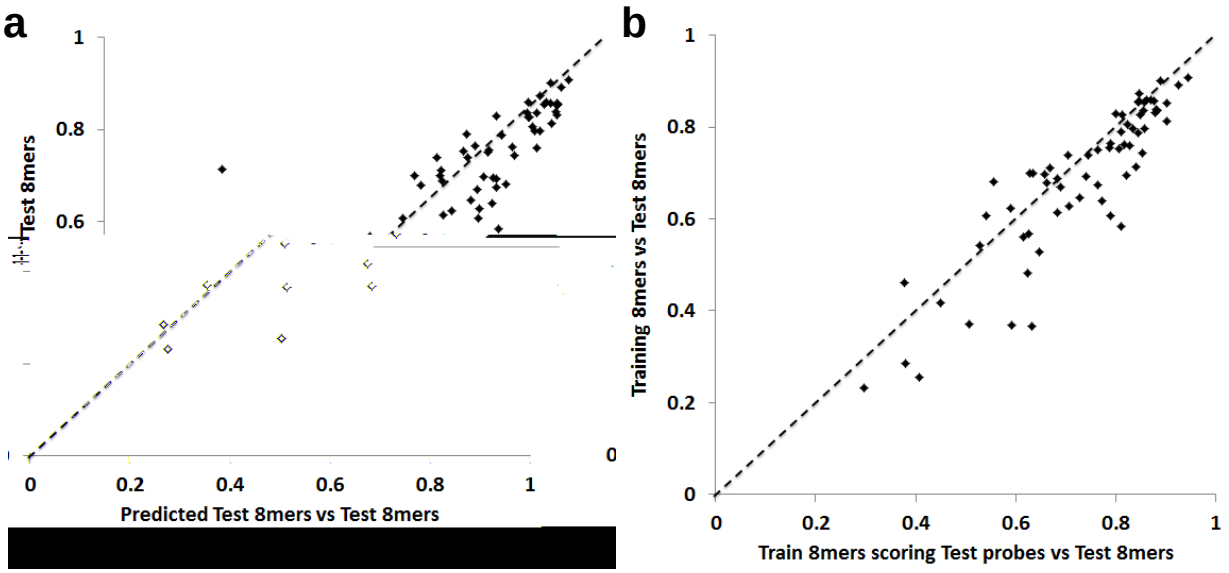
Supplementary Note 1. DREAM5 challenge evaluation criteria

For the initial DREAM5 challenge, we used a different set of evaluation criteria. Three of the criteria measured the average similarity of the predicted probe intensities (i.e. scanner measurements) to the actual intensities for each experiment, where similarity was measured using (i) the Pearson correlation of the raw probe intensities, (ii) the Pearson correlation of the log values of the intensities, or the (iii) Spearman rank correlation of the probe intensities. The other two criteria first transformed predicted probe intensities for each experiment into 8-mer intensities by calculating the median predicted intensity of all 32,896 8 base sequences on the array. All 8-mers were then ranked by their median predicted intensity, and (iv) the area under the receiver operating characteristic curve (AUROC) or the (v) the area under the precision-recall curve (AUPR) was calculated using high-scoring 8-mers as positives. We defined high-scoring 8-mers as those with "E-scores" (Berger *et al.* 2006) (modified AUC values, which score how well the presence of an 8-mer within a probe sequence predicts the ranking of the probe intensity) exceeding 0.45. It was previously established that E = 0.45 can be used as a cutoff for high-confidence in binding of a TF to the given 8-mer (Berger *et al.* 2008). Since the E-scores for each 8-mer are also derived on the basis of the intensity of the probes they reside on (Berger *et al.* 2006), this last criterion is essentially a measurement of consistency with the E-score.

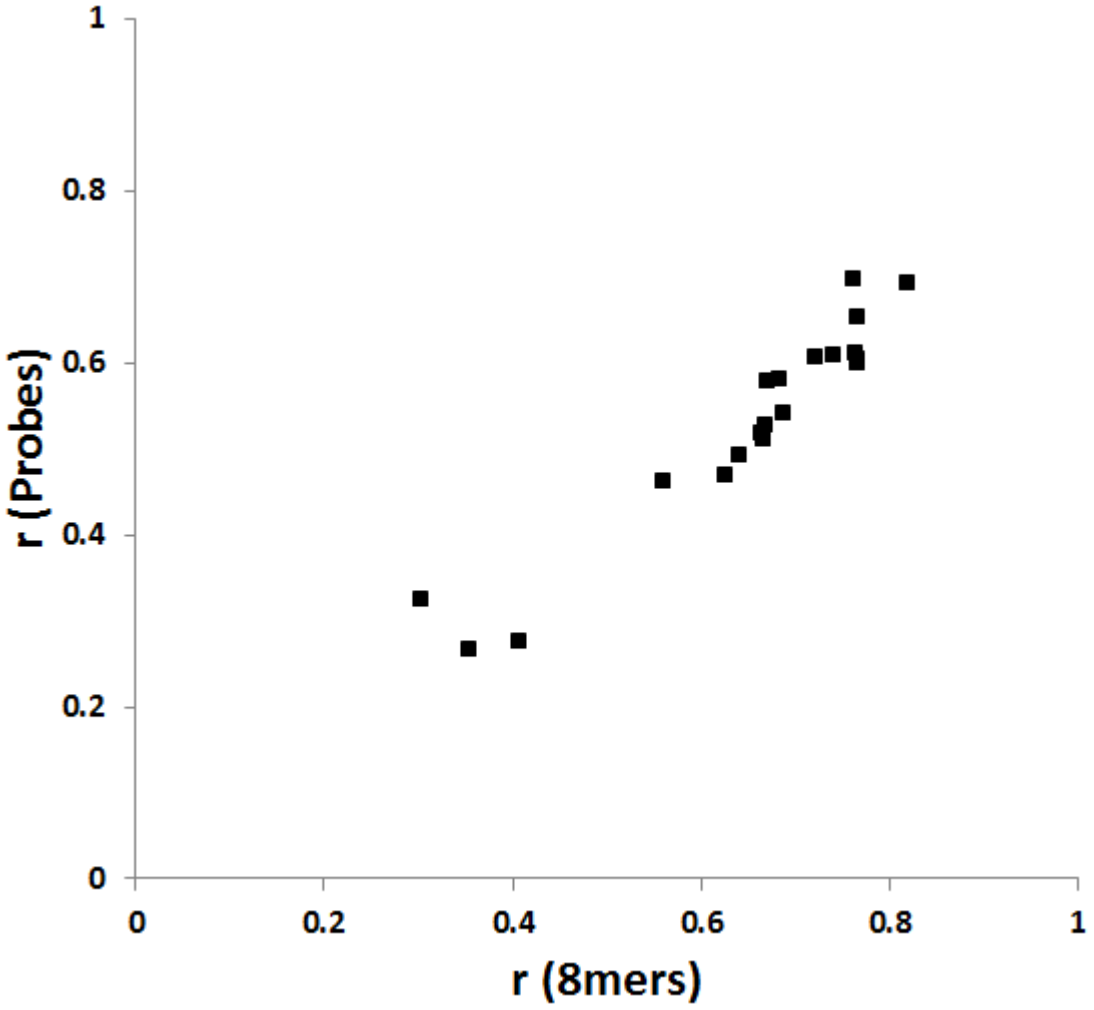
We calculated a single score for each of the five evaluation criteria for each team by averaging across all 66 experiments. A final team score was then calculated by ranking all teams within each criterion and calculating the average rank across all five criteria. This final score is slightly biased towards probe-based evaluations, since three criteria are based on probes, and only two are based on 8-mers.

scores from the same training PBM, and concluded that PWMs often do not completely capture sequence specificity of TFs (Badis *et al.* 2009). Zhao and Stormo (Zhao and Stormo 2011) contested this conclusion, claiming that a superior motif discovery tool, BEEML-PBM, is better suited to the task than the tools we used. But, Zhao and Stormo also introduced a new scoring procedure: on the test data, they first scored 35-mers, and then calculated 8-mer Z-scores. We tested this scoring system, and found that it results in dramatically improved correlations to the measured test 8-mer Z-scores. In fact, on the new 66 TF data set, motifs produced by BEEML-PBM and scored using this system outperform the 8-mer Z-scores (**Supplementary Note 2 Figure 1a**). Even the 8-mer Z-scores from the training data yield higher correlations if the test data is first scored as 35-mers, and then converted to 8-mers (**Supplementary Note 2 Figure 1b**). Thus, whether the test criteria involve scoring 35-mers or 8-mers, both PWMs and k-mer models benefit from first scoring the 35-mers. This finding suggests that our previous conclusions regarding secondary motifs should be revisited (see main text). We also observed that, using this procedure, the correlations obtained for 8-mers and for 35-mers on the same array scale with each other almost perfectly, whether the 35-mers are scored with PWMs or with 8-mers (**Supplementary Note 2 Figure 2**). The only significant difference we have observed between scoring 35-mers or 8-mers is that “secondary motifs” appear to confer a slight advantage when scoring 8-mers, but not 35-mers (see main text).

TF	Data type	PMID	35m PA	35m BEEML	35m FR	35m Zscores	8m PA	8m BEEML	8m FR	8m Zscores
Ace2	MITOMI	20802496	0.783	0.203	0.032	0.644	0.827	0.221	0.031	0.682
Aft1	MITOMI	20802496	0.232	0.039	0.241	0.194	0.327	0.034	0.285	0.230
Aft2	MITOMI	20802496	0.727	0.782	0.639	0.495	0.762	0.821	0.692	0.500
Atf4	160K PBM	(vinson)	0.084	0.159	0.536	0.207	0.134	0.072	0.557	0.439
Bas1	MITOMI	20802496	0.546	0.059	0.308	0.102	0.583	0.080	0.356	0.131
Cbf1	MITOMI	20802496	0.498	0.303	0.754	0.712	0.505	0.306	0.793	0.750
Cbf1	MITOMI	17218526	0.459	0.460	0.045	0.851	0.419	0.420	0.033	0.842
Cbf1	PBMs (vc)	22146299	0.359	0.383	0.138	0.641	0.339	0.366	0.182	0.673
Cebpb	160K PBM	(vinson)	0.543	0.838	NaN	0.538	0.354	0.482	NaN	0.445
Cin5	MITOMI	20802496	0.254	0.884	0.857	0.433	0.271	0.894	0.874	0.446
Cup9	MITOMI	20802496	0.200	0.276	0.298	0.211	0.232	0.324	0.346	0.233
Dal80	MITOMI	20802496	0.164	0.081	0.228	0.251	0.180	0.125	0.262	0.271
Gat1	MITOMI	20802496	0.331	0.655	0.636	0.606	0.373	0.726	0.704	0.657
Gcn4	MITOMI	20802496	0.509	0.754	0.734	0.488	0.511	0.760	0.742	0.484
Max	MITOMI	17218526	0.589	0.627	0.557	0.862	0.508	0.550	0.623	0.860
Mcm1	MITOMI	20802496	0.006	0.170	0.227	0.296	0.004	0.277	0.257	0.337
Met31	MITOMI	20802496	0.345	0.052	0.069	0.497	0.391	0.056	0.076	0.519
Met32	MITOMI	20802496	0.420	0.597	0.573	0.377	0.468	0.646	0.609	0.368
Met32	PBMs (vc)	22146299								



Supplementary Note 2 Figure 1. Evaluation of Zhao and Stormo's 8-mer scoring system Comparison of the accuracy of Zhao and Stormo's 8-mer scoring scheme predictions and test array Z-scores. **a.** Correlation across all 8-mers between the test array Z-scores and (1) BEEML-PBM's probe predictions, converted to 8-mer median intensities (X axis) and (2) the training array Z-scores (Y axis). **b.** Correlation across all 8-mers between the test array Z-scores and (1) the training array Z-scores summed across each probe sequence, converted to 8-mer median intensities (X axis) and (2) the training array Z-scores (Y axis).



Supplementary Note 2 Figure 2. Comparison of 8- mer correlation scoring to probe correlation scoring

We evaluated each of the algorithms by calculating the mean correlation of probes or 8-mers across all 66 TFs. For the probe-based evaluation, we calculated the Pearson correlation between the predicted probe intensities and the test intensities. For the 8-mer-based calculation, we converted the predicted and test probe intensities to median 8-mer intensities, and then calculated the Pearson correlation across all 32,896 8-mers. The final probe (Y axis) and 8-mer-based correlation (X axis) for each algorithm is plotted.

Supplementary Note 3. Effect of array data pre-processing steps on algorithms

For all published algorithms and the three algorithms that finished in the top four in the DREAM challenge that take less than 24 CPU hours to run per experiment, we determined the effect on performance of a panel of nine commonly used microarray pre-processing steps (see **Supplemental Note 3 Methods**). Using our final evaluation criteria, we compared the scores of the predictions produced by each algorithm when given input data with and without each pre-processing step. We found that spatial detrending invariably improves performance, regardless of algorithm (**Supplementary Note 3 Figure 1**). Other pre-processing steps improve some algorithms, while decreasing the performance of others. For example, quantile normalization substantially improves the performance of MatrixREDUCE and Team_E, but adversely affects the performance of Seed-and-Wobble and PWM_align. We found that most algorithms are robust to the presence of bad spots on the arrays, as only RankMotif++ and MatrixREDUCE displayed substantially increased performance when removing manually flagged spots from the training data set (**Supplementary Note 3 Figure 1**).

For each algorithm, we used the combination of pre-processing steps that resulted in the best final score to produce a final pre-processed data set specific to the given algorithm (see **Supplemental Note 3 Methods**). The degree to which data pre-processing improves algorithm performance varies substantially. Although all algorithms show increased performance when using their respective pre-processed data as input, score improvements ranged from 0.013 (8mer_pos) to 0.076 (MatrixREDUCE) (**Supplementary Note 3 Figure 1**). The average improvement across all 12 algorithms was 0.03, a difference nearly as great as that separating the final scores of the top four algorithms (see **Table 2**). Given the fact that many algorithms performed similarly to each other in the final evaluations, the inclusion of relevant data pre-

processing steps appears to be an important consideration when developing and implementing an algorithm.

Supplementary Note 3 Methods

Evaluated data pre-processing methods

We evaluated the effect of a panel of nine data pre-processing methods on the final performance of each algorithm. For each pre-processing method, we compared the final score achieved by the given algorithm upon performing pre-processing to the final score when performing no pre-processing. Each algorithm was trained on training array data created using the given pre-processing method, and tested on test array data pre-processed with the same pre-processing method.

Inclusion of linker sequence (Include linker)

Each probe sequence has a 35 base unique sequence, and a 25 base non-unique primer sequence. It is possible that a TF might bind specifically to a portion of the non-unique sequence. To gauge the effect of allowing for this possibility on each algorithm, we created a dataset that includes the first five bases of the probe linker sequence (and hence is 40 bases long, instead of only including the 35 unique bases).

Removal of bad array spots (No bad spots)

A well-documented artifact of microarray data analysis is that some probes will inevitably be unusable for a variety of reasons, including smudges on the slide, edge effects, or scratches on the surface. To correct for these effects, we created a dataset that does not include any probe whose spot was manually flagged as either bad or suspect. The minimum number of bad spots on an array was 25, with a maximum of 3044 and a mean of 678 (out of ~40,000 total probes).

Removal of low intensity probes (No low intensity)

Extremely low intensity probes are often caused by artifacts such as edge effects. To correct for this effect, this pre-processing dataset discards all probes with extreme low intensities using a threshold derived from the probe intensity histogram. We calculated the threshold by taking the mode intensity of the histogram, and then moving toward lower intensity bins until *f (k)* < 0:005 *f (m),* where *f (k)* is the frequency in bin *k*, and *f (m)* is the frequency at the mode.

Use of median instead of mean pixel intensity (Use median)

Most available algorithms use the mean pixel intensity of a probe as its score. As an alternative, this dataset instead provides the median pixel intensity.

Subtraction of background pixel intensity (Minus background)

Microarray spot pixel intensities might be influenced by factors other than hybridization of TFs to probes, including dust particles and stray fluorescent molecules. Such local background pixel intensities are often quantified by calculating the mean pixel intensity around (outside of) each spot. This dataset corrects for background effects by subtracting the mean local background pixel intensity from the mean pixel intensity of the probe.

Normalization by median probe intensity (Norm by median)

It is possible that the overall intensity of a probe might be influenced by its sequence, or by its physical location on the array. To address these possibilities, we created two datasets that account for the expected intensity of a probe. Expected intensities were determined by first calculating for each probe its median intensity rank across all 66 experiments. The expected intensity of a given probe in a given experiment was then estimated as the intensity of the probe with the associated rank in the given experiment. This procedure accounts for the fact that the

distribution of the probe intensities varies across experiments. For each probe in each experiment, we then either divide or subtract its actual intensity by its expected intensity. Normalized intensities less than 0 were set to 1.

Quantile normalization

Quantile normalization is a common practice in microarray data pre-processing, and is often used to help account for the fact that systematic variations at low intensities can differ from variations seen at medium and high intensities. This procedure forces one distribution (the probe intensities of a given experiment) to fit another distribution (here, a consensus distribution of the probe intensities across all 66 experiments). Hence, for a given experiment, the probe with the highest value will assume the value of the brightest probe in the consensus distribution, the 2nd brightest will receive the 2nd higher score, and so on.

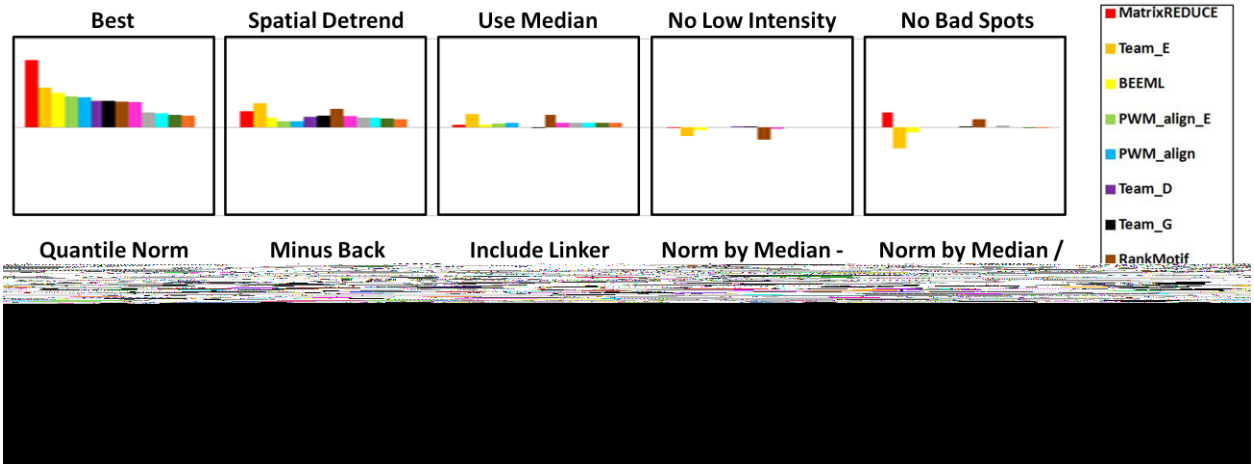
Spatial detrending

Spatial detrending accounts for the non-uniform distribution of probe intensities that is often observed across a microarray slide. For example, it is common for certain regions of the array to be darker or lighter than other regions, due to unwanted effects such as smudges or fingerprints. Spatial detrending accounts for such effects by rescaling the intensity of each spot by the ratio of the global median and the median calculated within an N x N window centered on the spot. Here, we tried windows of varying sizes by setting N equal to 5, 7, 9, or 12.

Creation of a final pre-processed dataset for each algorithm

For each algorithm, we sought to determine the single dataset whose pre-processing steps resulted in the best performance (measured as the final score achieved by the algorithm, as described above). In theory, the combination of pre-processing steps that individually improved the performance of a given algorithm should result in the best performance when combined into

a single pre-processed dataset. In practice, we found that certain pre-processing steps negatively affected other pre-processing steps when used in combination, and that different results were obtained depending on the order that some pre-processing steps were performed. To circumvent these issues, we adopted a greedy method where for each algorithm, we first chose the pre-processing method that resulted in the largest increase in performance, performed that method on the data, and subsequently performed the 2nd best method. At each step, we evaluated the final score of the algorithm, removing any pre-processing step that resulted in a loss of performance. This procedure was iteratively repeated until all pre-processing steps were tested that individually resulted in an increase in performance for the given algorithm. Certain pre-processing steps that are highly related (e.g. spatial detrending with window length of 7 vs. 9) were not permitted to both be used for a given dataset- in such cases, the single pre-processing method amongst the related ones was chosen that individually resulted in the largest increase in performance. In all cases, the final combination of pre-processing steps resulted in a higher final score than any of the individual pre-processing steps used on its own (**Supplementary Note 3 Figure 1**).



Supplementary Note 3 Figure 1. Effect of array data pre-processing steps on algorithm performance

The final performance of each algorithm was determined when using one of nine microarray data pre-processing steps, and compared to the final performance when using non-processed data. Shown here is the difference between these two values for each pre-processing step, for each algorithm. For “Spatial Detrend”, the results for the best-performing window size are used for each algorithm (see **Online Methods**). The scale for all plots ranges from +0.10 to -0.10. A score of +0.10 indicates that the given pre-processing step improves the final score of the given algorithm from e.g 0.60 to 0.70. The plot labeled “Best” shows the performance of the best combination of pre-processing steps for the given algorithm. Pre-processing steps are sorted in decreasing order of mean improvement across all algorithms. Algorithm key is indicated at the right.

Supplementary Note 4. Examination of the relative difficulty of each TF for each algorithm.

Overall, there was high variability in the final scores of the various algorithms across the 66 TFs we analyzed. We therefore conducted a series of analyses aiming to assess which TF sequence specificities were hardest to model, and to understand what made them challenging. First, we re-clustered **Figure 2** so that TFs with related DNA binding domain sequences are near each other (**Supplementary Note 4 Figure 1**). The results of this analysis did not indicate any clear tendency for TFs from certain structural classes to be harder to model than others.

As an alternative approach, we next devised a scoring method to quantify the relative difficulty for an algorithm to accurately capture a TF's binding preferences. We calculated a single score for each TF/algorithm pair that we refer to as the relative prediction accuracy (RPA). RPAs take the difference in quality of the predictions of the various algorithms into account by performing a Z transformation of each TF's final evaluation scores such that they are relative to each algorithm, and hence can be interpreted as how well the given algorithm performed for a given TF, relative to the other TFs (see **Supplementary Note 4 Methods**). A higher RPA value thus indicates that the given algorithm achieved a higher final score than its average final score across all TFs.

We found that there are clearly certain TFs whose sequence preferences are harder (or easier) to predict than others, regardless of the algorithm or type of model used (**Supplementary Note 4 Figure 2,** panel a). For most TFs, at least one algorithm has a positive RPA, indicating that at least one algorithm performs better than its average for most TFs. Likewise, the minimum RPA is less than zero for most TFs (**Supplementary Note 4 Figure 2,** panel b), indicating that most TFs are a challenge to predict for at least one algorithm.

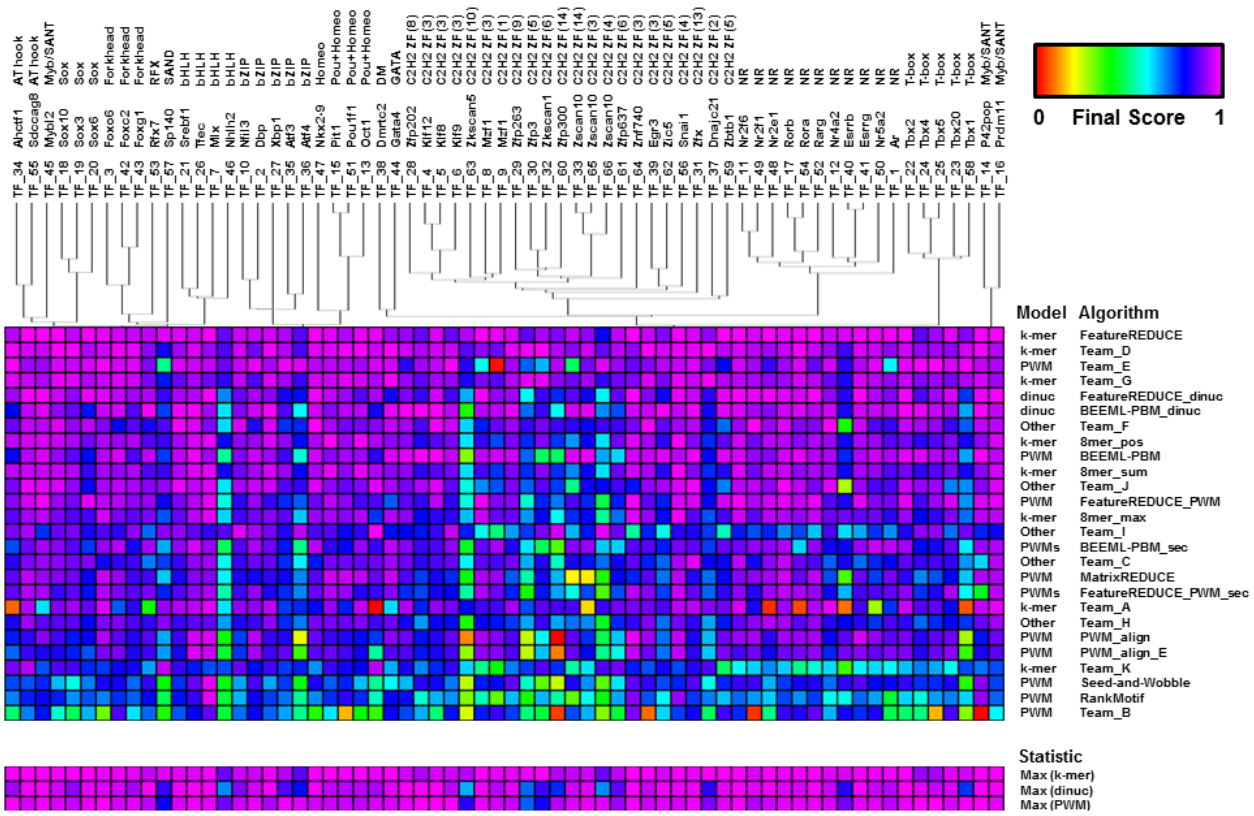
A major factor in the overall ease of predicting probe intensities for a given TF appears to be the quality of the underlying experimental data: if there are a large number of high-scoring 8-mers, and if there is a high correlation between the 8-mer transformed data from the two arrays, then the data is easier to model (**Supplementary Note 4 Figure 2**, panel c). Indeed, the three TFs for which it is hardest to predict sequence preferences in the PBM data (the C_2H_2 proteins Zkscan5, Zfp3, and Zfp300) have only between one and three 8-mer E-scores exceeding 0.45 in both their training and test data. Such cases appear to be particularly difficult for regression-based algorithms such as BEEML-PBM and FeatureREDUCE, because they will only work if there is enough variation in the data to parameterize the model and thus fit a PWM properly. Although many of the hardest to model TFs are C_2H_2 zinc fingers, several non-C_2H_2 TFs, including Nhlh2, Sp140, Tbx1, and Atf4, present a similar modeling challenge due to their small number of strongly bound 8-mers, indicating that this phenomenon is not exclusive to the C_2H_2 class. Nor is it a general property of C_2H_2 zinc fingers: when only considering C_2H_2 TFs with at least 10 8-mers with E-scores exceeding the 0.45 threshold, there is virtually no difference from the majority of other TFs (data not shown). In summary, we do not find any clear tendency for specific families of TFs to be harder to model. Instead, algorithm performance is largely dictated by properties of the underlying array data.

Supplementary Note 4 Methods

Calculation of Relative Prediction Accuracies (RPAs)

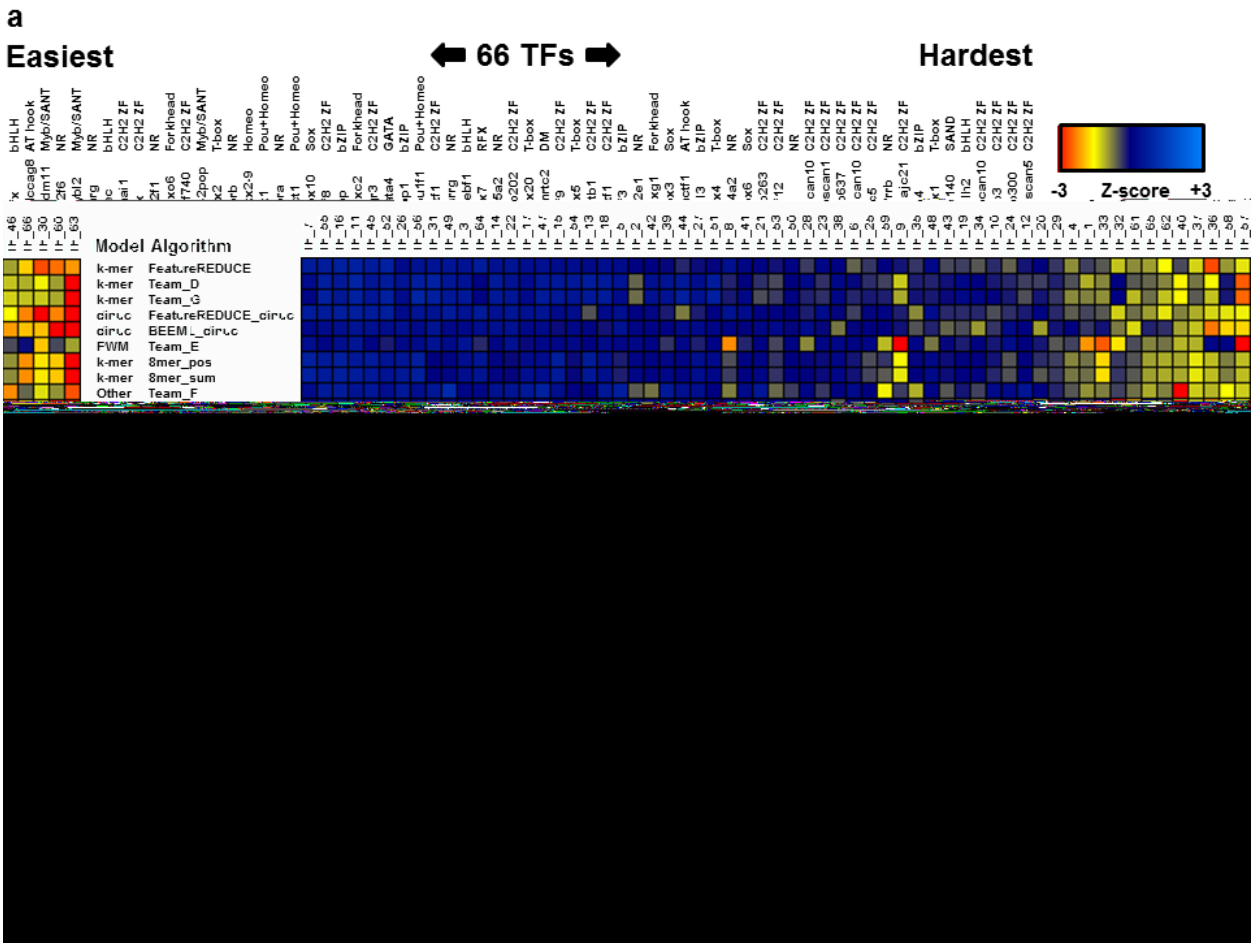
We found that different algorithms performed better (or worse) on different subsets of TFs. In order to quantify the difficulty a given algorithm had in accurately capturing a given TF's binding preferences relative to the other TFs, we calculated a single score for each algorithm/TF pair

that we refer to as the relative prediction accuracy (RPA). For a given evaluation scheme, the 66 evaluation scores of an algorithm are first transformed into Z-scores (relative to their distribution across all 66 experiments). The Z-scores are then summed up across all three evaluation schemes, yielding a final score for each TF/algorithm pair quantifying how much better the given algorithm performs on the given TF, relative to its performance on all of the other TFs.



Supplementary Note 4 Figure 1. Algorithm performance, clustered by TF family

Same data as in Figure 2, with the TFs clustered by DNA binding domain amino acid sequences. The tree was created using the ClustalW web server, with default settings.



Supplementary Note 4 Figure 2. Relative ease of modeling each transcription factor

(a) Comparison of the relative performance of each algorithm on each TF. Each entry depicts the relative performance accuracy (RPA) of the given algorithm/TF pair, which is essentially a Z transformation of an algorithm's final score, relative to its final score on for al TFs. TFs are sorted in decreasing order of their mean RPA, and hence the "easiest" TFs to predict are on the left. (b) Summary statistics of RPA values across all algorithms. The mean, maximum, and minimum RPAs achieved by any algorithm. (c) Potential causes of difficulty in TF modeling. Each value is Z-transformed as for the RPA values. The Pearson correlation of each statistic with the RPA values is indicated on the right. Key: Corr (E), correlation of 8-mer E-scores between experimental replicates; Corr (Z), correlation of 8-mer Z-scores between replicates;

Train E > 0.45, number of 8-mer E-scores exceeding 0.45 in the training data; Test E > 0.45, number of 8-mer E-scores exceeding 0.45 in the test data; Train Max Z, value of the highest Z-score in the training data; Test Max Z, value of the highest Z-score in the test data; Non-specific, performance of an algorithm that simply predicts the median intensity of the g 13( )-4(i)5(n)13(t)-4(e181

outperformed the 8mer_max algorithm in our evaluations (**Table 2**). We also found that, in general, the strategy of only allowing a TF to bind to one strand at a time had negligible effect on the performance of most algorithms (**Supplementary Note 5 Table 1**). Likewise, little to no improvement was seen when BEEML-PBM's position-specific effect was included in the probe scoring scheme (**Supplementary Note 5 Table 1**).

In contrast, the incorporation of the effect of non-specific sequence binding (the μ parameter) results in a relatively large improvement of performance for the BEEML-PBM method, raising its final score from 0.889 to 0.914 (**Supplementary Note 5 Table 1**). A similar improvement is not exhibited for most other algorithms, likely because the estimation of μ is dependent on the PWM parameters themselves, and hence the value of μ estimated by BEEML-PBM does not necessarily transfer to the PWMs produced by the other algorithms. Interestingly, we found the greatest improvement in BEEML-PBM's performance when the μ and position effects were included, but not the strand effect, suggesting that the various strategies employed by BEEML-PBM have interdependent effects.

Supplementary Note 5 Methods

Comparison of methods for scanning sequences using PWMs

Several scoring systems have been proposed for scoring a sequence using a PWM, including log-odds-based and energy-based scoring systems (see **Supplementary Note 8**). Further, additional scoring schemes specific to PBMs are utilized by the BEEML-PBM method in order to produce its final probe intensity predictions (see Zhao *et al.* 2009 for full descriptions). In order to directly compare the performance of the various scoring methods, we scored the final PWMs of each PWM-based method using each scoring scheme. Scoring systems evaluated here

include the log-odds framework (either summing or taking the maximum score across the probe sequence), and the energy-based scoring system, using all possible combinations of the three scoring strategies utilized by BEEML-PBM. For each algorithm/scoring system pair, the final score (as described above) is reported in **Supplementary Note 5 Table 1**.

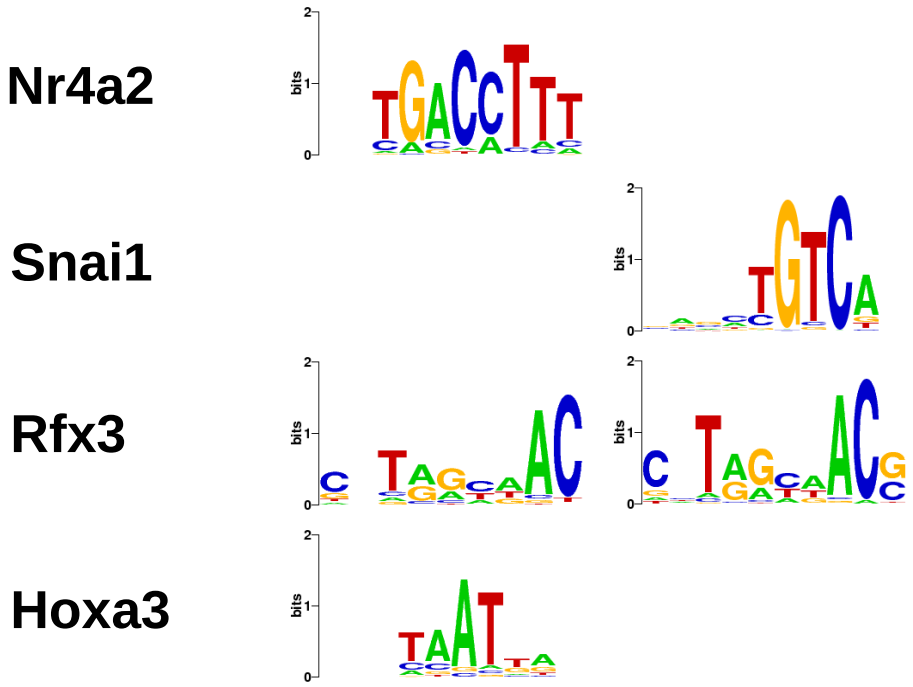
PWM[1]	**PWM Score**[2]	**FR**	**TM_E**	**BML**	**MR**	**PWM (E)**	**PWM**	**SW**	**RM**
Prob	Log Sum (max)	0.827	0.806	0.856	0.762	0.834	0.822	0.719	0.677
Prob	Log Sum (sum)	0.858	0.855	0.888	0.842	0.843	0.827	0.723	0.694
Energy	Boltz	0.858	0.858	0.889	0.843	0.846	0.828	0.724	0.697
Energy	Boltz (str)	0.858	0.862	0.889	0.841	0.846	0.827	0.725	0.699
Energy	Boltz (pos)	0.861	0.861	0.890	0.848	0.838	0.838	0.729	0.705
Energy									

Supplementary Note 6. Secondary motifs, by category

We identified cases in where secondary motifs increase predictive performance, and grouped them into one of seven categories. Here, we provide examples and discuss each category.

Category 1. Minor variations on the primary motif (fine tuning)

The majority of secondary motif improvements fall into this category. In each case, the final score increases with the addition of a secondary motif (see "Impr" col) , but the primary and secondary motifs are nearly identical (see "r(P,S)" column, and logos).



Name	TF_ID	Study	Alg.	r(P,T)[1]	r(S,T)[2]	r(P+S,T)[3]	Impr[4]	r(P,S)[5]
Nr4a2	TF_12	DREAM	BEEML	0.437	0.624	0.648	0.212	0.855
Snai1	TF_56	DREAM	FR	0.868	0.885	0.917	0.048	0.818
Rfx3	3961.1	Badis09	BEEML	0.693	0.707	0.747	0.054	0.755
Hoxa3	2783.2	Badis09	FR	0.906	0.844	0.941	0.036	0.736

[1] Correlation between primary motif 8-mer predictions and test 8-mer scores

[2] Correlation between secondary motif 8-mer predictions and test 8-mer scores

[3] Correlation between combined (regressed) 8-mer predictions and test 8-mer scores

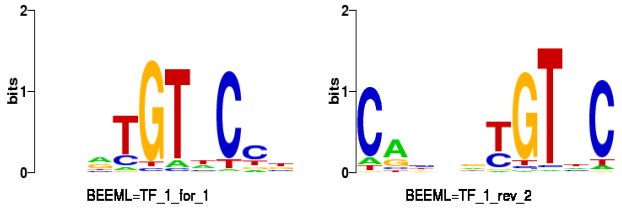
[4] Improvement of combined predictions over primary predictions

[5] Correlation between primary motif 8-mer score predictions and secondary motif 8-mer score predictions

Category 4. Half sites and dimers

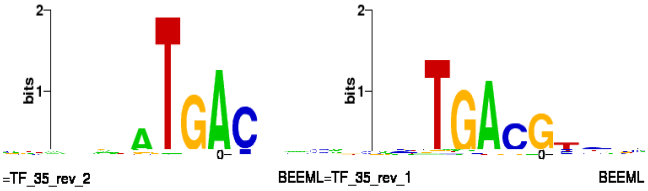
This category contains TFs that can bind DNA as homodimers. In each case, one motif includes a single half site, and the other contains all or a portion of the other half site. The portion of the half sites present in each motif are indicated on the right.

Ar



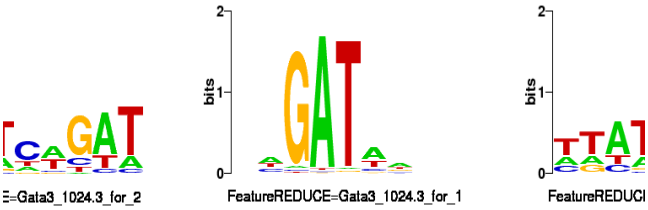
P[1]: GGAACANNNTGTTCC
S[2]: GGAACANNNTGTTCC

Atf3



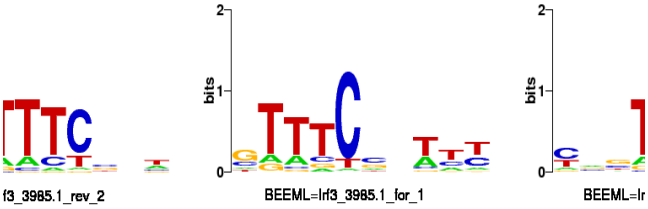
P: TGACGTCA
S: TGACGTCA

Gata3



P: TTATCWGATAA
S: TTATCWGATAA

Irf3



P: TTTCNNTTTC
S: TTTCNNTTTC

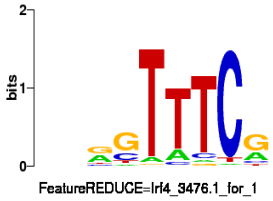
Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Ar	TF_1	DREAM	BEEML	0.739	0. 493	0.746	0.006	0.547
Atf3	TF_35	DREAM	BEEML	0.655	0.536	0.676	0.021	0.577
Gata3	1024.3	Badis09	FR	0.873	0.495	0.882	0.008	0.421
Irf3	3985.1	Badis09	BEEML	0.615	0.635	0.661	0.046	0.788

[1] Primary motif consensus sequence. Half sites of full binding site are colored in red and blue. The portion of the full binding site included in the motif is underlined.

[2] Secondary motif, with underlining and coloring as for primary motif.

Category 4. Half sites and dimers (cont'd)

Irf4

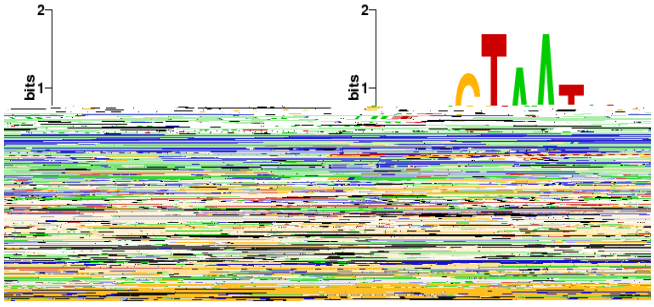


P: TTTCNNTTTC
S: TTTCNNTTTC

Irf6

P: TTTCNNTTTC
S: TTTCNNTTTC

Nfil3



P: TTACGTAA
S: TTACGTAA

Nfi

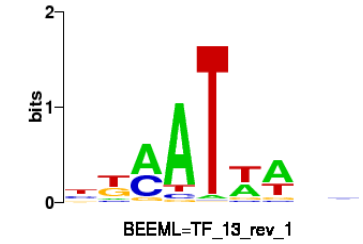
P: TTACGTAA
S: TTACGTAA

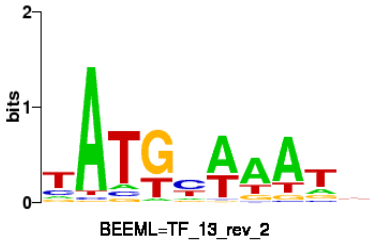
Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Irf4	3476.1	Badis09	FR	0.807	0.694	0.843	0.036	0.622
Irf6	3803.1	Badis09	FR	0.789	0.716	0.829	0.040	0.668
Nfil3	TF_10	DREAM	FR	0.681	0.476	0.724	0.044	0.345
Nfil3	TF_10	DREAM	BEEML	0.728	0.522	0.737	0.009	0.518

Category 5. POU+Homeodomains

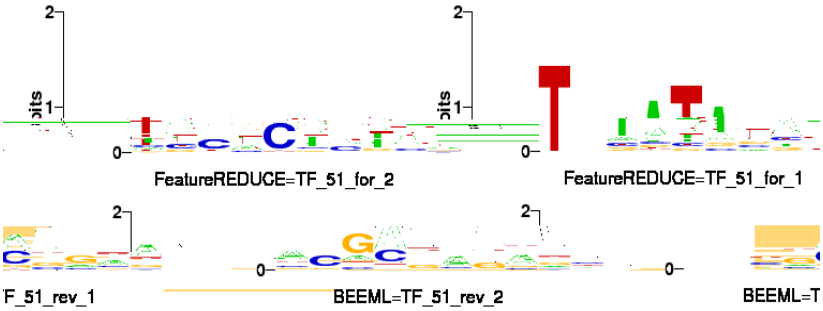
In this category, all three examples have the "classic" primary and secondary POU+Homeodomain motifs of TAAT and ATGCWWW.

Oct1





Pou1f1

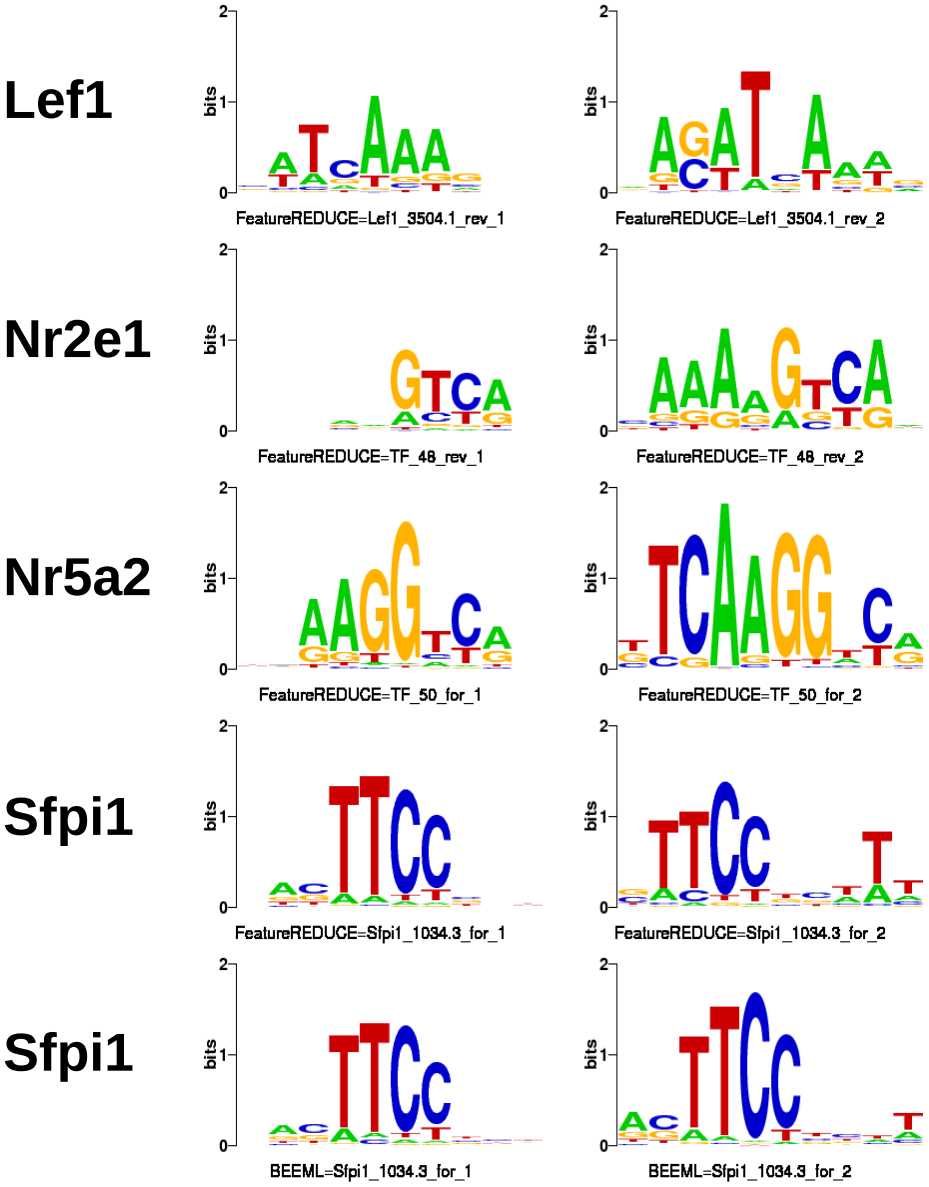


Pou1f1

Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Oct-1	TF_13	DREAM	BEEML	0.839	0.541	0.856	0.017	0.465
Pou1f1	TF_51	DREAM	FR	0.871	0.493	0.890	0.019	0.370
Pou1f1	TF_51	DREAM	BEEML	0.841	0.626	0.849	0.008	0.635

Category 6. Motif “extensions”

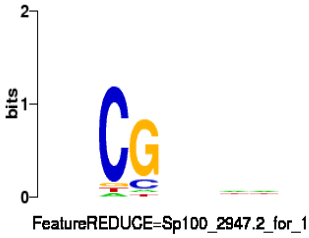
In this category, the secondary motif is visually similar to the primary motif, but it includes additional bases.

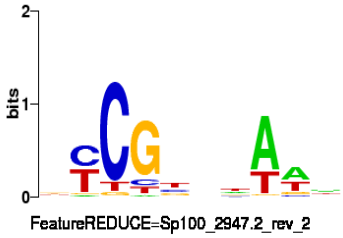


Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Lef1	3504.1	Badis09	FR	0.691	0.506	0.724	0.033	0.445
Nr2e1	TF_48	DREAM	FR	0.850	0.361	0.863	0.014	0.246
Nr5a2	TF_50	DREAM	FR	0.509	0.221	0.526	0.017	0.173
Sfpi1	1034.3	Badis09	BEEML	0.661	0.656	0.673	0.012	0.902
Sfpi1	1034.3	Badis09	FR	0.666	0.479	0.683	0.016	0.528

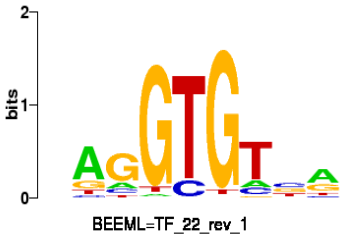
Category 6. Motif “extensions” (cont’d)

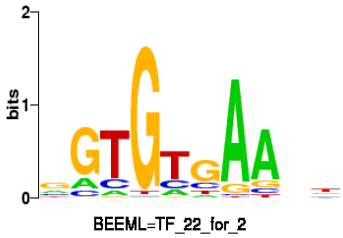
Sp100



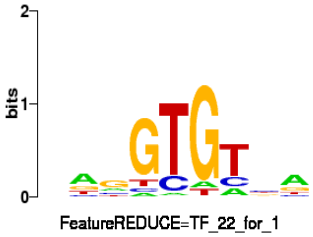


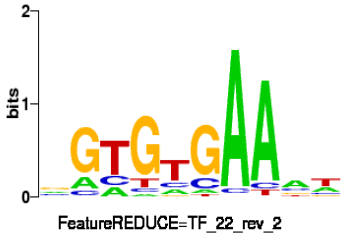
Tbx2



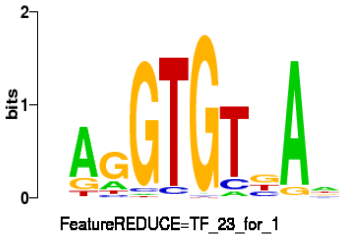


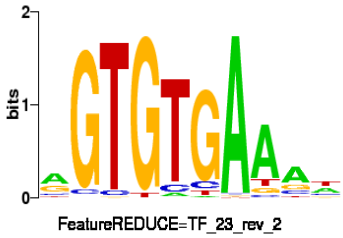
Tbx2



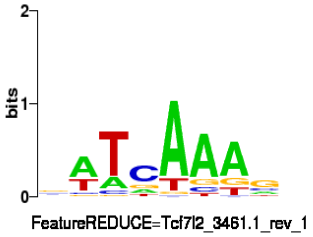


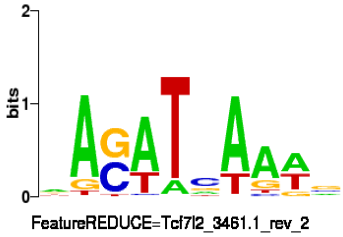
Tbx20





Tcf7l2



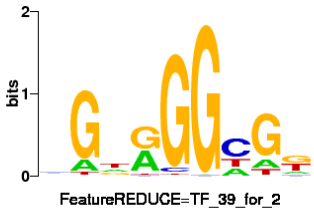


Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Sp100	2947.2	Badis09	FR	0.818	0.605	0.854	0.036	0.450
Tbx2	TF_22	DREAM	BEEML	0.841	0.480	0.872	0.032	0.307
Tbx2	TF_22	DREAM	FR	0.814	0.324	0.827	0.013	0.224
Tbx20	TF_23	DREAM	FR	0.832	0.431	0.841	0.008	0.351
Tcf7l2	3461.1	Badis09	FR	0.836	0.559	0.866	0.030	0.424

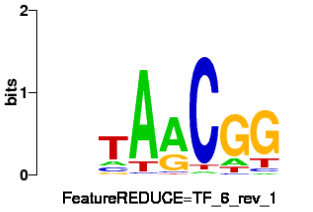
Category 7. C_2H_2 Zinc Fingers

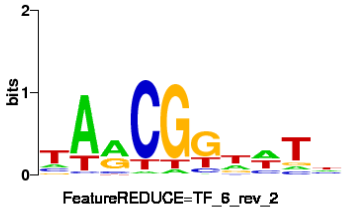
The majority of "effective" secondary motifs that we identified are for C2H2 zinc fingers. Many include "extensions" similar to the previous category, which might be examples of additional zinc finger arrays being utilized. In other cases, the motifs are entirely unrelated. In such cases, a different set of zinc finger array might be utilized for DNA binding.

Egr3

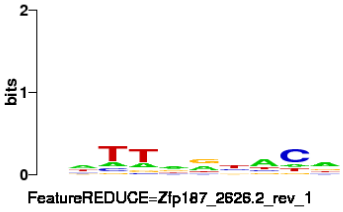


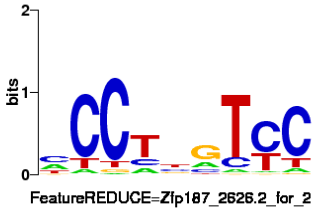
Klf9



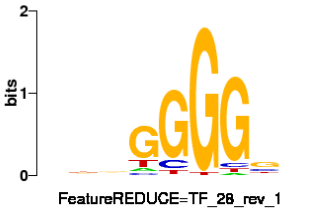


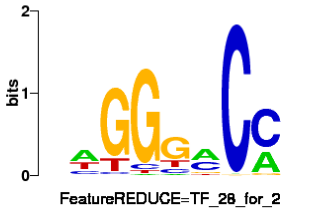
Zfp187



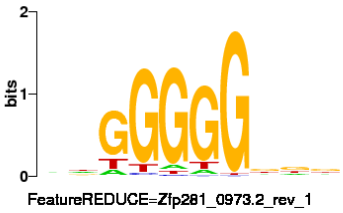


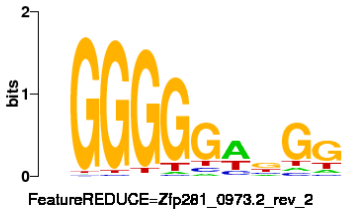
Zfp202





Zfp281

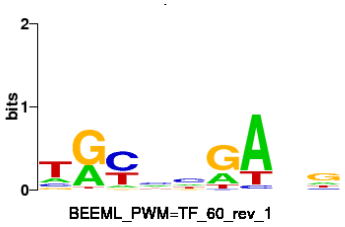


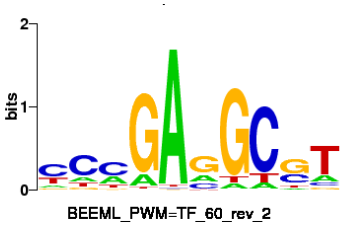


Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Egr3	TF_39	DREAM	FR	0.562	0.520	0.639	0.077	0.395
Klf9	TF_6	DREAM	FR	0.357	0.388	0.377	0.020	0.683
Zfp187	2626.2	Badis09	FR	0.376	0.353	0.452	0.076	-0.140
Zfp202	TF_28	DREAM	FR	0.708	0.244	0.717	0.009	0.186
Zfp281	0973.2	Badis09	FR	0.806	0.657	0.809	0.003	0.694

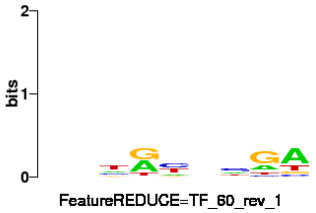
Category 7. C_2H_2 Zinc Fingers (cont'd)

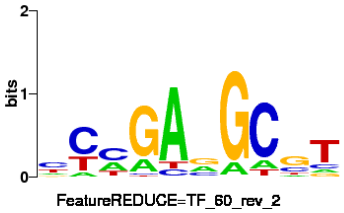
Zfp300



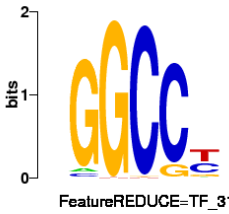


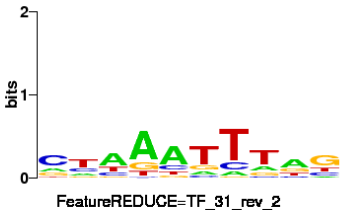
Zfp300



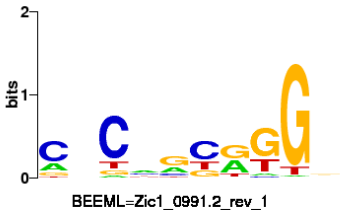


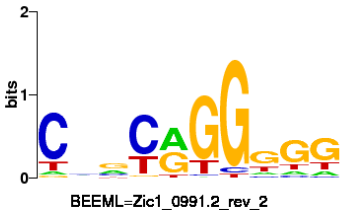
Zfx



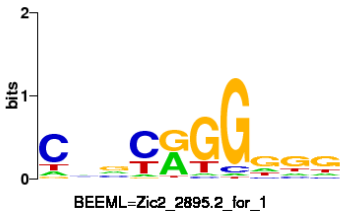


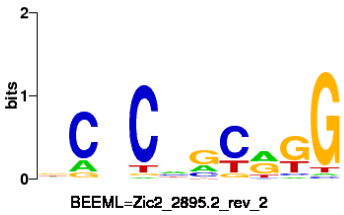
Zic1





Zic2

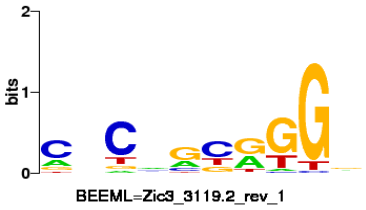




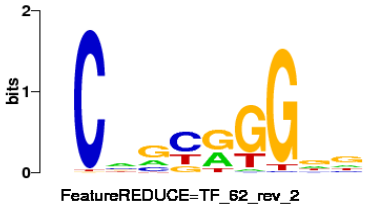
Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Zfp300	TF_60	DREAM	BEEML	0.314	0.148	0.342	0.028	0.038
Zfp300	TF_60	DREAM	FR	0.569	0.024	0.576	0.007	-0.119
Zfx	TF_31	DREAM	FR	0.784	0.226	0.828	0.044	-0.077
Zic1	0991.2	Badis09	BEEML	0.754	0.805	0.818	0.064	0.800
Zic2	2895.2	Badis09	BEEML	0.803	0.614	0.804	0.001	0.708

Category 7. C_2H_2 Zinc Fingers (cont'd)

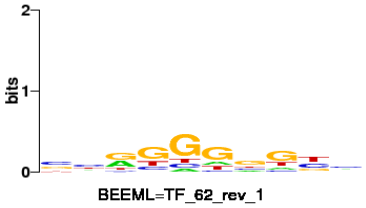
Zic3



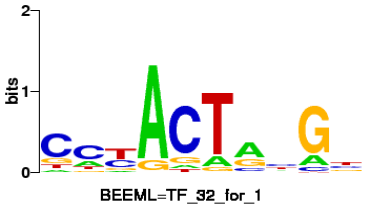
Zic5

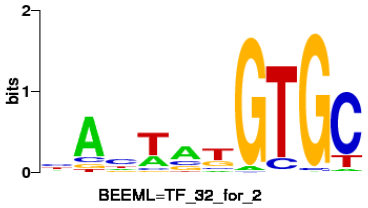


Zic5

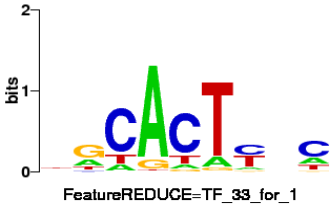


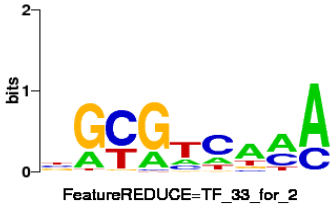
Zkscan1



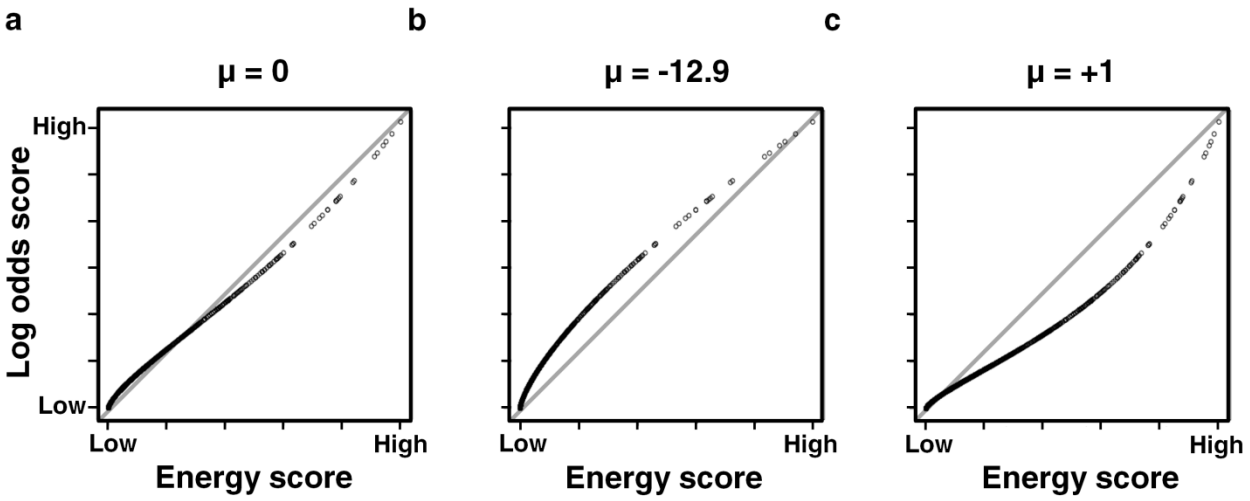


Zscan10





Name	TF_ID	Study	Alg.	r(P,T)	r(S,T)	r(P+S,T)	Impr	r(P,S)
Zic3	3119.2	Badis09	BEEML	0.672	0.767	0.773	0.100	0.804
Zic5	TF_62	DREAM	FR	0.226	0.258	0.275	0.049	0.436
Zic5	TF_62	DREAM	BEEML	0.613	0.571	0.631	0.017	0.722
Zkscan1	TF_32	DREAM	BEEML	0.241	0.523	0.549	0.309	0.142
Zscan10	TF_33	DREAM	FR	0.467	0.162	0.492	0.025	-0.002



Supplementary Note 8 Figure 1. Comparison of energy and log-odds scoring systems

Illustration of the effect of the μ parameter on the relationship between the scores produced by the log-odds and energy scoring systems. Each panel depicts the energy and log-odds score for all 32,896 unique 8 base sequences using a single PWM produced by the BEEML-PBM algorithm for Sox10 (TF_18). The value of the μ parameter, which is used only in the energy scoring system, is varied across the three panels (value indicated at top).

Supplementary Note 9. Full descriptions of algorithms

I. Brief description of published and novel algorithms

BEEML-PBM

We ran two versions of BEEML-PBM: one using PWMs, one using dinucleotides. For the PWM version, we ran BEEML-PBM on each training experiment using PWMs of width 7 to 10. The width producing the highest correlation between the predicted intensities and the training intensities was chosen as the final PWM for each experiment. The dinucleotide models take substantially longer to run, so we only used widths of 10 because, in general, this size produced the best results using the PWMs. For both versions, we used the following parameters, per the author's suggestions: num.trials = 10 (number of random starts for optimization); max.iter = 20 (max number of iterations in levenberg-marquardt optimation); lambda = 0.1 (regularization control).

FeatureREDUCE

We ran FeatureREDUCE using its default settings (available upon request). FeatureREDUCE estimates multiplicative affinity parameters associated with both mononucleotides and dinucleotides. For the full model, we included the '-kmer' option, which additively models PBM-specific biases using k-mers of length 4 to 8. For FeatureREDUCE_dinuc, we did not include the '-kmer' option. For FeatureREDUCE_PWM, we used the '-nodinuc' option, and did not include the '-kmer' option. A more detailed description of the FeatureREDUCE algorithm is provided at the end of this document.

MatrixREDUCE

We obtained MatrixREDUCE v1.0 and ran it using the recommended settings: -dyad_length=3 -flank=3 -min_gap=0 -max_gap=20 -max_motif=1 -motif=7-10.

RankMotif++

We ran RankMotif++ using the settings recommended by the authors (Badis et al. 2009). Namely, the following options were used: -u 1 (log transform the probe intensities); -p 3 (positive probe threshold); -c 1.5 (confidence interval scale); -n 400 (400 negative sequences); -s 5 (five restarts); -a ACGT; -r TGCA. We varied the PWM width from 6 to 13 using the "-w" option, and used the single best PFM (as determined by RankMotif++'s internal criteria). Multiple attempts at variations on parameter settings produced similar results.

Seed-and-Wobble

Seed-and-Wobble can be run in three different modes: continuous (which only considers continuous motif patterns), gapped (which allows gaps in the motif), and symmetric gapped (which allows a single gap between the two half sites, where the half sites are of the same length). For the gapped version, we tried all 330 possible gapped patterns of length 8 with gaps up to length 5. For the symmetric version, we tried nine possible patterns, from a gap of length 0 to a gap of length 8. For the continuous version, we only used continuous 8-mers. For all three modes, we used the default parameters: startposition=2 (position from end of probe to consider); Escore_cutoff=0.25 (to store for integrated list of top k-mers from all seeds); topN = 1; (number of top k-mer seeds to use for seed-and-wobble PWM construction). Each Seed-and-Wobble PFM was "trimmed" based on the information content at each position, as described in (Badis et al. 2009). We found that the continuous version performed best in our evaluation scheme, so we only include results from the continuous version.

8mer_max

This algorithm first converts the predicted probe intensities to 8-mer scores by calculating the median probe intensity of each 8-mer. Each test array probe is then scored by taking the maximum scoring 8-mer that occurs in its probe sequence.

8mer_sum

Same as 8mer_max, but the sum of all 8-mer sequences is used to score each test probe.

8mer_pos

Same as 8mer_sum, but takes into account the position within the probe sequence of each 8-mer, using a scheme similar to that of BEEML-PBM (Zhao and Stormo 2011). In brief, the algorithm identifies for each experiment the top 25 8-mers (based on median intensities), and calculates the mean intensity of these top-scoring 8-mers at each probe position. This results in a distribution containing the mean intensity of high-scoring 8-mers at each probe position (across all 66 experiments). This distribution is then used to correct for position effects by essentially dividing the observed frequency of each 8-mer at each position by its expected intensity. Full details of the algorithm can be found in (Zhao and Stormo 2011).

PWM_align

Calculates 8-mer E-scores as described in (Berger and Bulyk 2006). Aligns all 8-mers with E-score > 0.45 using ClustalW, then trims the resulting alignment by restricting to positions

present in at least half of the sequences in the alignment. The resulting alignment is then converted to a position frequency matrix.

PWM_align_E

Aligns 8-mers with E-scores > 0.45, as described for PWM_align. Before converting to a PFM, each sequence in the alignment is first weighted by the E-score of the corresponding sequence. For example, if the top-scoring 8-mer is 0.49 for the sequence GATGTTCC, this sequence gets counted 49 times in the alignment. If the lowest scoring 8-mer is 0.45 for the sequence TGTGTTCT, this sequence counts 45 times in the alignment. Thus, higher-scoring 8-mer receive higher weights in the final PFM frequencies.

II. Full description of algorithms from the DREAM challenge

Team_A: Reconstructing binding site motifs from PBM data – the Amadeus approach

Introduction

Our group is developing methods for motif finding based on co-regulated gene sets (Linhart, Halperin et al. 2008) or using raw expression data without a predefined gene set (Halperin, Linhart et al. 2009). We were therefore most interested in the power of these methods for detection of TFBS motifs (the Bonus Round). Towards this, we developed a fast and accurate method building on the capability of our Amadeus motif finder (Linhart, Halperin et al. 2008). As an aside, we also used established learning methods for the main challenge, the results of which are described in this manuscript.

Methods

Motif reconstruction

We developed a very simple, efficient and generic method for extracting binding site motifs (represented as a position weight matrix, PWM) from PBM data (Orenstein, Linhart et al. 2012). We first score each 9-mer based on the average binding intensity of the probes that contain it, and use the scores to rank the 9-mers. Note that we disregard the linker segment of the probes. We then provide the top 9-mers as a target set to our Amadeus motif finding algorithm (Linhart,

Halperin et al. 2008). The input to Amadeus is the 1000 top-ranked 9-mers, while the background set is all possible 9-mers, and its output is a PWM representing the motif. We had full success recovering correctly all 20 TFs in the training PBM data. Amadeus was optimized to work efficiently with large input sets and with the concise input here the algorithm works extremely fast, requiring 30-60 seconds per PBM dataset.

Binding intensity prediction

Our approach to the main challenge was to formulate the problem as a linear regression problem. We used two types of features: (1) 8-mers as Boolean features, indicating if an 8-mer appears in the probe sequence. The 1000 top and 250 bottom ranking 8-mers in terms of their average intensity were used. These features are used for identifying specific binding. (2) Integer variables indicating the number of occurrences of each of the 64 possible nucleotide triplets in the probe sequence. These variables are used for evaluating non-specific binding. Each probe was thus assigned a set of 1314 features and a linear regression between the features and the probe's intensity was computed using the Lasso algorithm (Tibshirani 1996). The resulting feature weights are then used to predict binding intensities of the test probes. The average correlation on the test examples was 0.66 with a standard deviation of 0.1.

Discussion

Our probe ranking method did not fare well in the main challenge, probably since its defining feature set was too simplistic and not informative enough. Possible directions for improvement include accounting for the effect of PBM array geometry on probe intensities and the use of more features for non-specific binding.

In contrast, the motif finding method, which was our main focus, is extremely fast and highly accurate, and was one of two best performers in the Bonus Round. We believe that the method captures well the information of specific binding, as opposed to non-specific binding. Our fast and simple pre-processing phase (selecting the top ranking 9-mers) has the advantage of discarding noise, which improves the accuracy of the PWM. A key advantage of our method is speed – running two orders of magnitude faster than other PBM-based motif finding methods. Moreover, Amadeus allows us to statistically integrate data from multiple PBMs, and even analyze together PBM and other data types (e.g., a target gene set from an expression/ChIP-chip experiment, or PBM for a related species) pertinent to the same TF. Another advantage is the generality of the approach, as any of the many available motif finders can be used in the second phase.

Acknowledgements

This study was funded by the European Community's Seventh Framework Programme under grant agreement no. HEALTH-F4-2009-223575 for the TRIREME project, and by the Israel Science Foundation (grant no. 802/08). YO was supported in part by a fellowship from the Edmond J. Safra Bioinformatics Program at Tel Aviv University.

Team_B: Incorporating motif discovery into feature extraction when predicting protein-DNA binding affinity

Introduction

The proposed method aims to extract distinguishing sequence features by motif discovery for predicting the binding specificity of transcription factors (TFs).

Methods

Given a protein binding microarray (PBM) as the training dataset, we extracted the top 1000 probe sequences with the highest signal intensities (each intensity value was subtracted by its corresponding background value) as the positive set and the 1000 probe sequences with the lowest signal intensities as the negative set. We then conducted motif discovery by using eTFBS (http://biominer.csie.cyu.edu.tw/etfbs/), a motif discovery tool that employs the algorithm proposed in (Chen, Tsai et al. 2008) to find over-represented subsequences in the positive set versus the negative set. Each group of similar over-represented subsequences is summarized as a position frequency matrix (PFM). For a TF, a number of such motifs are discovered by eTFBS using the default parameter settings. These PFMs were considered as the candidates of binding motifs for the TF of interest in the following analyses.

Next, we used the *mSS* scores employed in (Kel, Gößling et al. 2003) to calculate the similarity between a motif *M* and a probe sequence *S*:

$$mSS(M,S) = \max_{j=1}^{35-m+1} mSS(M, S_{j:j+m-1}),$$

where *m* is the length of the motif *M* and $S_{p1:p2}$ stands for the subsequence of *S* starting at the position *p*1 and ending at the position *p*2.

We expected that the *mSS* score of a probe sequence in a PBM should be highly correlated with the probe intensity measured by the PBM. That is, if the motif *M* is more similar to the real binding profile of the TF, the correlation value would be higher. Spearman's rank correlation coefficient was applied on the vector of the *mSS* scores against the vector of measured PBM signals based on the positive set of a training array. We defined the set of positive probes using a method similar to that employed by the RankMotif++ algorithm (Chen, Hughes et al. 2007). The only difference is that we required the number of positive probes in the set to be at least 1600. We calculated the Spearman's rank correlation coefficient for all the motif candidates and assigned the one with the highest value as the binding motif of the TF of interest.

For each probe sequence in the training array, we collected at least one motif (the one with the highest correlation score along with those with a correlation score > 0.8×the score of the top

motif) for calculating *mSS* scores and used them as the features for constructing regression models. In addition to the *mSS* scores, we also calculated the *g* scores (equation (9) in (Chen, Hughes et al. 2007)) for each motif and included them in the feature set for constructing the predictive model as well. Each probe sequence employs the normalized signal intensity (normalized by the method described in (Chen, Hughes et al. 2007)) as the target value when conducting SVR training (using nu-SVR of LIBSVM (Chang and Lin 2001)). Five-fold cross-validation was performed to find the best parameter settings based on only the instances in the positive set. After the regression model was constructed using the parameter combination that achieves the best performance on the training data, the prediction was made for the corresponding testing array.

Discussion

The proposed method did not perform as well as expected. Two potential reasons might have caused this result. First, the features adopted in building the regression models rely heavily on the discovered motifs. For better performance, probe-specific but TF-independent features might be necessary. Second, when constructing the regression models, only a small set of high-intensity probes were used for parameter tuning. The constructed regression models might have poor predictions on low-intensity probes. Although the performance of predicting binding affinity did not perform as well as expected, we observed that the motifs discovered by the proposed method are consistent between ME and HK arrays.

Team_C: Random Forests for predicting TF-DNA binding

Introduction

The method applied by our team blends two methods. One is based on random forests, a machine learning algorithm; the other is the more classical bioinformatical approach of motif finding. We begin modelling by selecting a sample of sequences with evenly distributed binding intensities that is subsequently divided randomly into two equal sets – the training set and validation set. Then we construct, for each sequence, several sets of descriptive variables that will be used by the machine learning. Next, the random forest classifier is trained on the training set data using these variables. The motif finding algorithm is also applied for a subset of sequences with high binding intensities. In the following step all algorithms are used to predict binding intensities of sequences from the validation set. Finally, a random forest model is constructed from the validation set with prediction of the individual algorithms used as descriptive variables. The classifier obtained in this way is then used for predicting the binding intensities of all probe sequences.

Methods

No pre-processing of the signal intensities was performed, apart from a logarithmic transformation of the binding intensities. Although the distribution of the binding intensities is far from uniform (with very large bias towards low binding intensities) we equalize the distribution using the following procedure. We first draw 2000 random numbers from a uniform distribution covering the entire range of binding intensities. Then, for each number, the sequence with the closest binding intensity is recruited to the sample. Finally duplicates are removed. The size of the resulting sample varied between 807 sequences (TF 53) and 1685 (TF 6). The resulting sample is then evenly split between the training set and the validation set.

Machine learning (ML) algorithms deal with data in the form of an information system that is a table, where rows correspond to objects and columns to variables describing these objects, including one column for a decision variable. The algorithm then learns the associations that connect descriptive variables to particular values of the decision variable. Therefore, the first step towards an application of ML methods is converting a string representing a sequence to a form suitable for input. To this end, we used n-gram spectra. In this representation, one records for all possible substrings of length n whether they are present or not in a given string: the corresponding attribute takes values of 1 or 0, respectively. For an alphabet of size A, there are A^n possible substrings (and hence A^n descriptive variables in the information system). We used 4-, 5- and 6-gram spectra resulting in information systems with 256, 1024 and 4096 variables, respectively. The decision variable was the binding intensity. The random forest (Breiman 2001) in the regression variant was grown using the training set. Additionally, for each system, we found those n-grams that contribute significantly to the random forest outcome with the help of

the Boruta algorithm (Kursa, Jankowski et al. 2010; Kursa and Rudnicki 2010). Then we used these attributes to grow additional random forests.

An alternative method was based on a motif finding algorithm. RankMotif++(Chen, Hughes et al. 2007) was run 4 times for each TF using the subset of sequences with high binding intensities, and the motif with the highest likelihood was retained. Then the MAST program from the MEME suite of programs was used to find motifs in a data set. The result for each sequence was a p-value for finding the motif in a given sequence.

Finally all algorithms that were trained on the training set were used to predict binding intensities for the test set. These values were used as attributes for the random forest algorithm that was used for blending. Subsequently, we used contributing algorithms to predict the binding intensities for all sequences and used blender random forest to obtain the final results.

Discussion

We used fraction of explained variance, measured on the validation set, for internal evaluation of the performance of contributing algorithms. The best results were obtained for the random forest trained on 5-grams -- the average fraction of the explained variance was 55%.

The feature selection procedure gave significant reduction of the number of variables used for model building. The average number of important features was reduced to 62 for 5-gram representation. Unfortunately, models built on the reduced sets performed slightly worse, with average explained variance in the validation set equal to 51%.

The results of RankMotif++ were significantly worse than that of the random forest models, with 24% explained variance on the validation set. Despite the lower accuracy of RankMotif++ results, they were included into the final blend along with the results of 4 random forest classifiers. It is interesting to note that feature selection actually slightly decreased the accuracy of the model on the training set. Nevertheless, we decided to include models built on reduced sets in the blend as well.

Unfortunately, the results of the final classifier applied to the test set were significantly worse than the internal tests of the individual models on the validation set. This likely happened due to insufficient sampling of the sequence space both in the training and validation sets, resulting in models that were over trained for the particular subset of probe sequences.

Team_D: A linear model for predicting TF binding affinities based on protein binding microarray measurements

Introduction

Protein binding microarrays (PBMs) are a high throughput technology used for measuring a protein's binding affinity toward thousands of double-stranded DNA sequences at once. We present a linear model that uses PBM measurements to build a protein binding profile that can be used to predict a protein's binding affinity towards short DNA sequences. Rather than being based on position weight matrix (PWM) models where adjacent nucleotides are assumed independent, our model learns the protein's binding specificity towards a set of short nucleotide strings (k-mers). Full details of our method are provided in (Annala, Laurila et al. 2011).

Methods

Our method first constructs a spatial intensity map and intensity histogram for each PBM sample. Probes with very low intensities are discarded using a threshold derived from the intensity histogram. Next, spatial detrending is applied by rescaling the intensity of each microarray spot by the ratio of the global median and the median calculated within a 7x7 window centered on the spot. This step compensates for the spatial trends (light or dark blotches) often seen in microarray samples. After this, the samples used for learning the motif models are quantile normalized. The quantile normalization step is able to recover high intensity tails in saturated PBM samples. It is critical that we do not simply discard the saturated probes as we did with dark probes, because whereas dark probes can be considered non-informative, high intensity probes are the most informative features in terms of binding affinity.

After pre-processing the PBM samples, our algorithm constructs a design matrix $\boldsymbol{H}$ for each PBM array involved in the experiment, so that

$$h_{s,k} = \begin{cases} 1, \text{ if k-mer } k \text{ is found in probe sequence } s \\ 0, \text{ otherwise} \end{cases}$$

The design matrix is built in a strand specific manner, so that reverse complement k-mers are considered separately. An extra column of ones is also added to the design matrix in order to account for a constant background in the probe intensities.

Once a design matrix has been constructed, we solve the k-mer affinity contributions $\boldsymbol{\alpha}$ from the linear system

$$\boldsymbol{p} = \boldsymbol{H\alpha} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{p}$ represents the log-transformed probe intensities from a PBM experiment, and the error term $\boldsymbol{\varepsilon}$ accounts for noise in the measured probe intensities. If we include all 4-8 mers in the design matrix $\boldsymbol{H}$, the system can easily become underdetermined. For this reason, we regularize the system by only including those 7-8-mers with the highest median intensity across the probes that contain them. We also include all 4-6-mers, since they are critical for accurately predicting the intensities of low affinity probes. This regularization approach is based on the assumption that k-mers with the highest median intensity are the most informative in terms of protein binding.

This sparse but large linear system is solved for the affinity vector by applying the conjugate gradient method to the normal equations $\boldsymbol{H}^T \boldsymbol{H}\hat{\boldsymbol{\alpha}} = \boldsymbol{H}^T \boldsymbol{p}$. Once the affinity vector $\hat{\boldsymbol{\alpha}}$ of a protein has been estimated from the data, we use it to predict binding intensities of probes on another PBM array (or any DNA sequences) by constructing another design matrix $\boldsymbol{H}'$ for the given sequences, and calculating the predicted intensities $\boldsymbol{p}' = \boldsymbol{H}' \hat{\boldsymbol{\alpha}}$.

Discussion

Our method was ranked as the best performer in the DREAM5 challenge, and although its prediction accuracy did vary between TFs, we found our pre-processing steps to significantly improve the results for samples containing hybridization artifacts (Annala, Laurila et al. 2011). And while we here only used the model in predicting PBM probe intensities, the model can also be applied in less artificial contexts. One obvious application is to use our model for predicting genomic binding sites and their associated TF affinities. With some adjustment, we suspect that our model can also be applied to ChIP-seq data.

Team_E: Prediction of TF-DNA interactions with Protein Binding Microarrays using basic PWM models

Introduction

In this challenge, we used a position weight matrix (PWM) as binding site model for predicting the read-out of a PBM experiment. In a PWM, the columns represent the weights for each of the 4 bases at the corresponding position in the binding sequence. The basic assumption in the PWM model is that each position in a binding site contributes independently and additively to the binding energy. We used a standard Expectation-Maximization algorithm for inferring the models from the data. Our prediction method further takes a reproducible probe-specific but factor-independent bias into account. The latter may have improved the ranking of our team by the performance measures used, without actually contributing to an understanding of the binding specificities of the factors under investigation. Furthermore, our approach is not completely automatic. Some parameter choices for model training were based on intuitive judgments from exploratory analysis of the training data. In summary, the good performance of our team indicates that good binding site models can be derived from PBM data with a combination of common sense and existing, well established sequence analysis methods.

Methods

Data pre-processing

All computations were carried out in log-space. We first converted the signal mean values into logarithms of base 10. No other values were used for pre-processing. In particular we did not exclude measurements flagged as bad quality. Before submitting the predictions, we reconverted the log-values into mean signal values by exponentiation.

Description of the Method

Our prediction method is based on the following linear model,

$$y_i \approx a + bm_i + c \cdot \mathrm{Score}(S_i, M)$$

Here, y_i is the binding score (log-transformed signal mean) for sequence S_i, m_i is the mean of y_i over all experiments carried out with the same microarray (HK or ME), M is a binding site model for the factor under consideration, and “Score” is a scoring function that returns a predicted signal mean value for a given sequence and binding site model.

As binding site model we used a standard position weight matrix (PWM). The PWMs were derived in a semi-automatic fashion. For each training array, we first ranked the probe

Description of the method.

We build a probabilistic classifier based on a ZOOPS model in the foreground and a homogeneous Markov model in the background, and we denote the class posterior of class c given sequence $\underline{x}_n$ and parameters θ by $P(c|\underline{x}_n,\theta)$. Under the assumption that an input sequence contains exactly one occurrence of the motif, the likelihood of sequence $\underline{x}_n$ amounts to

$$P_{OOPS}(\underline{x}|\theta) = \sum_{l=1}^{L-w+1} P_{pos}(l|\theta) P_F(x_1,\ldots,x_{l-1}|\theta) P_M(x_l,\ldots,x_{l+w-1}|\theta) P_F(x_{l+w},\ldots,x_L|\theta),$$

where $P_{pos}(l|\theta)$ denotes the probability that a motif occurrence starts at position l, $P_F(\underline{x}|\theta)$ denotes the likelihood given the flanking model, and $P_M(\underline{x}|\theta)$ denotes the likelihood given the motif model. If an input sequence does not contain a motif occurrence, it is modeled by the flanking model alone.

As a motif model, we use a WAM model, i.e. an inhomogeneous Markov model of order 1, and as a flanking model and for the background class, we use homogeneous Markov models of order 2 or 3.

We learn the parameters θ by an extension of the MSP principle to weighted data, where the log class posterior of each class, i.e. foreground (fg) and background (bg), contributes with weight $w_n^c, c \in \{fg, bg\}$ to the objective function. The parameters are then numerically optimized with respect to

$$\hat{\theta} = \arg\max_{\theta} \left[\sum_{n=1}^{N} \sum_{c \in \{fg,bg\}} w_n^c \cdot \log P(c|\underline{x}_n,\theta) \right] + \log Q(\theta|\alpha),$$

where the first term is a weighted variant of conditional likelihood, and $Q(\theta|\alpha)$ denotes a prior on the parameters θ with hyper-parameters α. Since this numerical optimization may get stuck in local optima or saddle points, we start from 10 different initializations. After the parameters have been trained, they can be used to evaluate the class posterior for the probe sequences of the test array.

Finally, we map these predicted probabilities of binding back to signal intensities by ranking the obtained probabilities and reporting the mean intensity of the same rank as the weighted average over all arrays of the same type, i.e. ME or HK, from the 20 PBMs of the training data.

Discussion

The extension of Dispom to weighted data was one of the top-scoring approaches of the main challenge. We assume that the strengths of this approach are i) a reasonable mapping of the intensities to weights combined with a discriminative approach, which allows for the inclusion of

all probe sequences into the training, and ii) the utilization of a WAM for modeling the motif, which can capture statistical dependencies between adjacent positions. Further experiments showed that performance can even be increased by using more complex motif models, and by optimizing the parameter q of the mapping. However, this improved version of the algorithm has not been included in the re-evaluation in this manuscript.

Acknowledgements

This work was supported by grant XP3624HP/0606T by the Ministry of Culture of Saxony-Anhalt.

Team_G: Prediction of transcription factor-DNA binding affinities with Protein Binding Microarrays using a linear model for k-mers

Introduction

A Protein Binding Microarray (Berger and Bulyk 2006) provides detailed measurements of affinities of a transcription factor to a large number of short DNA probe sequences. In order to describe this interaction we assume that a probe intensity can be modeled as a sum of affinities of individual short k-mer motifs occurring in the probe sequence. Formally, we model the observed probe intensities as a product of an occurrence matrix of in-probe-motifs and of a vector of unknown motif affinities. We apply a multiple linear model to estimate the motif affinities. In order to provide the best predictions, we evaluate different sets of motifs: k-mers of different lengths (k=4..8) with or without a central gap. Moreover, we test whether predictions are improved by unification of k-mers with their reverse complements, or by ignoring of k-mers located close to the start of the probe sequence.

Methods

A PBM experiment for a transcription factor may be represented as a set of pairs (S_p, A_p) where S_p denotes a DNA sequence of a probe p and A_p is a corresponding binding affinity of the transcription factor calculated as logarithm of observed probe spot mean intensity.

For our model we assume that the probe affinity can be modeled as a composition of affinities of single motifs m present in the sequence S_p. Further, we assume that the contribution of individual motifs is additive.

This gives for all probes $p = 1,\ldots,P$ and all considered motifs $m = 1,\ldots,M$ a multiple linear model:

$$A_p = C_{p,m} \cdot a_m + \varepsilon_p,$$

where $C_{p,m}$ denotes the number of occurrences of motif m in the sequence of probe p and a_m denotes individual affinity of motif m. We estimate the motif affinities by minimizing the squared probe error ε_p so that $\widehat{a_m}$ correspond to the coefficients of the linear model.

In order to estimate the bias and standard error of the estimated coefficients $\widehat{a_m}$ we apply the delete-50% jackknife technique (Efron and Tibshirani 1994). We repeat the estimating procedure n times for randomly chosen subsets of 50% probe sequences and as a final affinity for a motif we choose the median of affinities estimated for the motif in different runs.

There is no reason to assume that the same motif model would provide the best description for different transcription factors. We therefore individually determined the best model for each PBM experiment. We tested k-mers of lengths from 4 to 7, with or without unifying sequences with their reverse complements. Additionally, for 4-mers and 6-mers we allowed a central gap of length from 1 to 6 nucleotides. Finally, we tested whether skipping of the nucleotides on the end of the probe influences the prediction performance.

Discussion

Using the PBM data of one array design (HK or ME) for training and the data for the other for evaluation, we were able to compare the results for different motif models. For 37 out of the 40 experiments in the training set, continuous *6*-mers were chosen as the best motif model. For the remaining 3 experiments, continuous 5-mers showed the best results. Interestingly, using k-mers with a central gap did not provide better results for any of the TFs from the training set. The second parameter which influenced the prediction was the choice of how many nucleotides from the beginning of the probe sequence should be skipped. We observed that exclusion of one or two flanking nucleotides increases the number of correctly predicted top-100 probe sequences but slightly decreases overall Spearman (and Pearson) correlation of all probe sequences. Depending on which aspect is of the optimization interest this model setting could be changed.

We have introduced a simple linear model for prediction of TF-DNA affinity which estimates binding affinities based on short motifs present in probe sequences. Finally, we have applied the continuous *6*-mer motif model without skipping the first nucleotides of the probe sequences to all 40 PBMs from the training set.

Team_H: High-resolution models of transcription factor-DNA affinities using discriminative learning

Introduction

PBM technology provides unprecedented high-resolution data on the subtle DNA sequence preferences of transcription factors (TFs). To exploit this rich binding data, we developed more general models of TF binding preferences based on inexact matches of k-length subsequences ("k-mers") rather than traditional position weight matrices (PWMs). We used a supervised learning strategy to train these TF binding preference models directly on PBM probe-level data. Therefore, in our approach, each (probe sequence, probe intensity) pair is a labeled training example, and we train support vector regression (SVR) models to directly learn the mapping from probe sequence to the measured binding intensity. We used a new k-mer based string kernel, called the di-mismatch kernel, for representing the similarity of double-stranded probe sequences on the PBM. This kernel is based on weighted counts of k-mer features, allowing up to *m* mismatches in the alphabet of dinucleotides, which favors mismatches that occur consecutively and better models preferred TF binding patterns. Full details on the kernel and SVR training procedure are described in (Agius, Arvey et al. 2010).

Methods

Data pre-processing

Examination of PBM probe intensity distributions suggests that only a few hundred of the ~40K PBM sequences in the positive tail actually contain the binding signal, while most of the intensity distribution comes from low-level non-specific binding and instrument noise. Therefore, we only used a few hundred high-intensity probe sequences as "positive" training data and a similar number of "negative" training sequences, rather than sampling from the full intensity distribution. More specifically, we ranked the normalized probe intensities and selected 500 sequences from the tail ends of the distribution as positive and negative training sequences.

Our approach requires the selection of appropriate parameters *k* and *m* for k-mer length and dinucleotide mismatches, respectively, correlating roughly to the length and degeneracy of the preferred motifs for each TF. In our previous work (Agius, Arvey et al. 2010), we used detection of the 100 highest-intensity probes within the top 100 scored probes as our measure of performance for model selection and for reporting results, reasoning that detection of the top probes was the most appropriate prediction task. However, since the original DREAM contest used correlation-based measures over the whole intensity distribution for three out of five performance metrics, here instead we used Spearman correlation for model selection based on 2-fold cross-validation on the training PBM arrays using k = 10,…,15 and *m* = 2….7. We then submitted the best-performing SVR models to DREAM.

Description of the method

In our approach, we directly learn the mapping from DNA probe sequences to intensity in PBM binding experiments by using support vector regression (SVR) together with a novel k-mer based string kernel called the di-mismatch kernel. For the kernel computation, each training sequence is decomposed into its constituent k-mers, and the weighted counts of k-mer features are computed based on matching in the dinucleotide alphabet, and allowing up to *m* dinucleotide mismatches in each feature count (Agius, Arvey et al. 2010). A normalized linear kernel is then computed on these features.

Careful feature selection can eliminate noisy features and reduces computational costs, both in the training and testing of the model. In particular, retaining very infrequent k-mers may add noise, and discriminative k-mer features should display an enrichment in the bound probes. Therefore, we selected the feature set to be those k-mers that are overrepresented in the ‘‘positive’’ probe class by computing the mean di-mismatch score for each k-mer in the “positive” class and the “negative” class and ranking features by the difference between these means. For this competition, we used the 4000 top-ranked k-mers without trying to optimize the number of retained features for each TF.

Discussion

Our training strategy was designed to learn models that accurately predict the high-affinity probe sequences, and in previous work we found that our k-mer based SVR models outperformed PWM-based approaches for detection of the top 100 probes within the top 100 predictions. Consistent with these published results, we were among the top three DREAM teams for the two ROC-based performance metrics, showing that our method is highly competitive for the task of discriminating high-affinity probes from “no-affinity” probes.

However, since we do not train on the “middle” of the probe intensity distribution – corresponding, we believe, to non-specific or very low-affinity binding – our method does not capture the probe sequence biases that correlate with lower measured intensities. Therefore, it is unsurprising that for the three correlation-based DREAM metrics, which evaluate correlation between real and predicted intensities over the whole probe intensity distribution, our DREAM performance was weaker. Although we did use Spearman correlation as our metric for model selection in our submitted results, we subsequently found that using a fixed reasonable parameter choice (e.g. (k,*m*) = (13,5)) gave similar overall performance for all metrics. Since we noticed that the other two teams with good ROC performance also had good correlation performance, we wondered whether we could adjust our training procedure to improve our performance on correlation-based metrics while retaining our advantage for discriminating high-affinity from no-affinity probes, for example by sampling probes from the full intensity distribution and adding these examples to the training data. After some unsuccessful experiments in this direction, we concluded that the model was not well-suited to learning to predict the full intensity

distribution, potentially because (i) SVR models, which use epsilon-insensitive loss rather than squared error, may not be appropriate regressors for the full probe distributions; and/or (ii) our feature selection procedure captures enrichment in the tails rather than correlation with the full signal. Potentially, a sparse square loss regression model based on a similar k-mer feature representation, such as lasso regression, might be more appropriate for this task. Finally, we did not perform any normalization of the input data for this contest – we suspect that doing so would have resulted in increased performance.

Team_I: Prediction of TF-DNA interactions with Protein Binding Microarrays using Rank Optimization

Introduction

We describe a rank optimization-based method for modeling and predicting Protein Binding Microarray (PBM) probe intensities.

Methods

Data pre-processing

First, we ignore all probes flagged as bad in each dataset by replacing entries corresponding to a flagged probe with NA. After that, for each array, we obtain the signal rank and the background rank of each probe by sorting their raw probe signal and background signal, respectively. Then, we define the corrected signal rank of each probe as the signal rank minus the background rank, and average background rank as the mean of the background rank of the given probe among all the datasets.

Description of the method

We propose two models for predicting the corrected rank of each probe: the PWM model and the k-mer model. For the PWM model, we apply the de novo motif finding tool Amadeus (Linhart, Halperin et al. 2008) to predict the PWM motif; then, the probes are ranked according to the PWM scores. During training, we obtain a positive probe set and a negative probe set: the positive probe set is defined as the set of probes with corrected signal rank higher than (mean+2*std), and the negative probe set is defined as a set of 400 probes randomly selected from the probes with corrected signal rank lower than (mean-2*std).

20 de novo PWMs are produced as the output of Amadeus, providing us with 20 features representing the maximum PWM score (log likelihood) of each de novo PWM. We applied multi-linear regression on these features to fit the corrected signal rank.

For the k-mer model, we predict the rank of a probe based on the occurrences of k-mers. Due to computational constraints, we only consider 6-mers. For each 6-mer, the probes in the training set are partitioned into a positive set (containing the given 6-mer) and a negative set (not containing the given 6-mer). Then, we draw the ROC curve against the corrected signal rank of each probe and the AUC score of the given 6-mer is the area under that ROC curve. After we compute the AUC score for each 6-mer from the training set, the k-mer score of each testing probe is defined as the max AUC score among all 6-mers inside that probe. Similarly, linear regression is applied to transform the k-mer score to a corrected signal rank.

To combine the PWM model and the k-mer model, we fit the corrected rank with the PWM predicting rank and the k-mer predicted rank using linear regression again in the training set. The final predicted signal rank of each probe in the testing set is the sum of the predicted corrected rank and the average background rank of the probe.

Since the final evaluation of the DREAM5 challenge also considered the raw signal value correlation (Pearson correlation), we fit the predicted signal rank to raw signal values using 3-order polynomial regression on the raw signal value of the training data.

Discussion

Our team performance performed best among all methods in the Spearman correlation criterion, which was the main target of our algorithm; however the average rank when considering all evaluation criteria was only 6. Specially, the measurements based on 8-mer AUPR and AUROC were very bad for our predictions (10th and 9th). The exact reason for this is still unknown and it is quite surprising to see that we can attain such good results for the probe rank but bad results for the 8-mer rank. Our preliminary analysis suggest that the probe signal is highly correlated with the background signal, as we obtained an average 0.8 spearman rank correlation between the probe signal and background signal. The reason for this, we conjectured, is because the background contamination is so serious that nearly all probes with the motif have high background, and we guessed that some TFs might be left behind after washing the PBM if the PBM was reused.

Team_J: Thermodynamic models with dinucleotide contributions to binding energy inferred by maximizing mutual information

Introduction

We used a modified version of the information-theoretic inference technique described in (Kinney, Tkačik et al. 2007) and (Kinney, Murugan et al. 2010) to infer models for the sequence-dependent binding energies of the 86 assayed mouse transcription factors (TFs). Unlike most motif finding algorithms, this method allows models of arbitrary functional form to be fit to PBM data. Applying this capability to the 20 TFs for which two arrays were available, we found that models containing nearest-neighbor dinucleotide contributions to binding energy greatly outperformed models with only single nucleotide contributions. We also found that models predicting the total thermodynamic occupancy at all possible binding sites on each probe greatly outperformed models for which only the strongest binding site on each probe is considered. The latter improvement is achieved with only one additional parameter per TF, the determination of which puts predicted binding energies in physical units (i.e. kcal/mol or k_bT). For the results reported here, we also fit a heuristic model for the significant sequence-dependent bias observed in the provided PBM data. We have since found that using a direct estimate of bias substantially improves the predictive performance of our inferred models.

Methods

Data pre-processing

We used 'signal median' fluorescence minus 'background median' fluorescence as the measured fluorescence of each probe. We then binned these fluorescence values into 10 bins respectively containing 1.0%, 1.5%, 2.2%, 3.2%, 4.7%, 7%, 10%, 15%, 22%, and 32% of the probes (from most to least fluorescent); alternative binning schemes made little difference in our results. In what follows, we use the notation $\mu_\alpha(\sigma)$ to denote the bin assigned to probe σ in the experiment on TF α. This bin μ_α serves as the probe's "measurement" in the model fitting procedure described below.

Binding model structure

The models we fit consisted of three parts: a function $\varepsilon_\alpha(\rho)$ that predicts the binding energy of TF α to a DNA binding site ρ, a function $x_\alpha(\sigma)$ that converts the binding energies of every possible site ρ in probe σ to a predicted TF occupancy on that probe, and a function $\eta(\sigma)$ that predicts the sequence-dependent bias in this probe's fluorescence.

We used linear models with nearest-neighbor dinucleotide contributions to predict the sequence-dependent binding energies of all TFs. Explicitly we used

$$\varepsilon_\alpha(\rho) = \sum_{i=1}^{L} E^{\alpha}_{i\rho(i)} + \sum_{i=1}^{L-1} J^{\alpha}_{i\rho(i)\rho(i+1)} ,$$

which is defined by mononucleotide parameters E^{α}_{ib} and dinucleotide parameters J^{α}_{iab}. Here L is the length of the binding site, $a,b \in \{A,C,G,T\}$, and $\rho(i)$ denotes the base at position i within site ρ. We also fit mononucleotide models for comparison, i.e. ones for which all .

The predicted occupancy of TF α on probe σ was computed using the sum of Boltzmann weights for each binding site:

where ρ runs over all sites of length L on both strands of the probe σ. Note that this functional form assumes the energies ε_α are in units of k_bT. For comparison, we also fit models where occupancy at only the strongest (lowest energy) binding site was included in the sum.

Modeling sequence-dependent bias

The PBM data provided showed strong sequence-dependent bias in fluorescence signals; across the different TFs, the mean correlation between the AT content of microarray probes and their measured fluorescences was 0.3. We attempted to model this bias — assumed to be probe-dependent but not TF-dependent — with a function $\eta(\sigma)$ (one function η for HK arrays and one for ME arrays) described by a dinucleotide matrix covering the 35 bp variable region of each probe:

$$\eta(\sigma) = \sum_{i=1}^{35} \eta^{o}_{i\sigma(i)} + \sum_{i=1}^{34} \eta^{1}_{i\sigma(i)\sigma(i+1)} .$$

Maximizing predictive information

In what follows, we use 'predictive information' to mean the mutual information between fluorescence measurements and the corresponding predictions of a model (assessed over all microarray probes). In the case at hand, our models make two predictions for each probe σ: the transcription factor occupancy $x_\alpha(\sigma)$ and the probe-specific bias $\eta(\sigma)$. The mutual information between these predictions and the measurements $\mu_\alpha(\sigma)$ is written as $I(x_\alpha,\eta;\mu_\alpha)$.

Discussion

We tested the general importance of dinucleotide contributions to binding energy and of occupancy at multiple sites per probe. Models of the 20 training set TFs were fit to each of the two arrays (HK or ME) and the predictive information of the resulting models was evaluated on the other array; in most cases, the performance of HK-trained models on ME arrays very closely tracked the performance of ME-trained models on HK arrays.

Including nearest-neighbor dinucleotide contributions to binding energy increased the predictive information of models by 50% on average, with 9 out of the 20 training data TFs showing increases of 50-100%. We view this as strong evidence that multinucleotide contributions to binding energy are important for describing the sequence specificities of many mouse TFs.

Secondly, using the sum of binding probabilities for all sites on a microarray probe, rather than for a single most favored site, increases the predictive information of the models by 17% on average, with increases of 20-40% for 8 out of the 20 training data TFs. Note that this improvement comes at the cost of only one additional parameter per TF: the energy scale.

The inferred binding energy models for the 20 training set TFs show reasonable agreement between the HK and ME data sets: 16 out of these 20 TFs exhibit a correlation coefficient between inferred parameters of 0.4-0.8 with a mean of 0.6. Inference for the other four TFs (Sp1, JunB, Sox14, Foxp2) did not appear to have been successful.

We have identified two factors that negatively affected our predictions for the DREAM competition. We now believe it is best to estimate the probe-specific bias η directly from data, e.g. using the median fluorescence of a given probe across all assayed TFs. This obviates the need for the iterative algorithm described above. Furthermore, the use of least squares regression for fitting the 5 TF-specific parameters is not the same as maximizing the correlation between these predictions and the measurements. We believe we should have maximized correlation because this, and not the sum of square deviations, was used to judge predictions.

Acknowledgements

This project was supported by NSF grant PHY-1022140 (A.M.), NSF grant PHY-0957573 (C.C.), and the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory (J.B.K.).

Team_K: Predicting TF-DNA binding specificity with Protein Binding Microarrays using linear regression and feature selection

Introduction

Modeling TF-DNA interactions is an important problem in understanding transcriptional regulation. The basic idea of our method is to use a set of short DNA words (k-mers) as features, and apply linear regression to predict the TF-DNA binding intensities:

$$p_j = \sum_i c_i s_{ij} + c_0 + \varepsilon_j \quad (1)$$

where p_j is the log-transformed binding intensity of the j-th probe sequence, c_i is the regression coefficient for the i-th feature (k-mer) and s_{ij} is the matching score between the i-th feature and the j-th probe sequence. A similar method has been used to predict *cis*-regulatory elements from gene expression data (Bussemaker, Li et al. 2001). Initially we pooled all k-mers with k ranging from 1 to 8. As the number of k-mers is large, we used feature selection to obtain a small subset of the k-mers that contribute the most to the binding specificity, in order to both save computational cost and improve generalizability of the model.

Methods

The matching score between a k-mer a and a probe sequence of length L, p, is defined as the sum of squares of the number of matched characters between a and every length-k subsequence of p.

$$S(a, p) = \sum_j (\sum_i \delta(a_i, p_{j+i}))^2,$$

where a_i and p_i is the i-th character of a and p, respectively, while $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. This scoring method allows an inexact match between a k-mer and a probe sequence to be scored, and therefore takes into account degenerate binding sites. On the other hand, it gives relatively higher weights to perfect matches than to imperfect matches, thus preventing multiple inexact matches from overweighting an exact match.

In order to find the most relevant k-mers for predicting TF-DNA specificity, we ranked all k-mers by two methods and further conducted feature selection based on the ranks. First, for each k-mer, we identified the probe sequences that contain at least one exact match to the k-mer. We

To perform feature selection, we first combined the top 16 features (k-mers) from the correlation-based ranking, and the top 16 features from the affinity-based ranking, to obtain up to 32 core features. We then performed a linear regression using the core features, and calculated the Spearman rank correlation between the predicted signal values and the true signal values as the baseline prediction accuracy. The top 1024-16 = 1008 features from each of the two ranking methods were combined as candidate features to give approximately 2000 candidate features. Each candidate feature was used in turn with the core features to perform a linear regression and the prediction accuracy was calculated as above. The candidate features were ranked by the improvement of prediction accuracy compared to the baseline accuracy. The top 256 features were selected as the final set of candidate features, from which we further performed a stepwise forward feature selection (Keleş, van der Laan et al. 2002). Basically, starting from the 32 core features, at each step we try to select and add one feature that can result in the largest improvement of prediction accuracy when it is added in the regression. This procedure is repeated until no improvement can be obtained or all features in the final candidate pool have been used. Finally, a linear regression is performed using the final set of selected features (typically less than 200), and the learned linear function is applied to predict the signal intensities for the test probes.

Discussion

Our method performed reasonably well when compared to the other methods participating in the challenge. Among 14 teams, the overall ranking of our method is #11. Two interesting observations are worth mentioning. First, with the probe-level evaluation, our method works better under the Spearman rank correlation-based metric than under the Pearson correlation-based correlation (0.53 vs. 0.46). This can be explained by the fact that our method has used Spearman rank correlation as an internal accuracy measurement to guide feature ranking and feature selection; therefore, our method is probably slightly "over-tuned" towards the Spearman correlation-based evaluation metric. Second, the prediction accuracy of our method measured by the probe evaluation metric is significantly higher than that measured by the 8-mer evaluation metric. With the probe-level evaluation, the accuracy of our method is comparable to the median accuracy achieved by the teams (0.46 vs. 0.52, Pearson correlation, and 0.53 vs 0.55, Spearman correlation). In contrast, with the 8-mer-level evaluation, our method is apparently an outlier when compared to the other teams (0.16 ours vs. 0.52 median accuracy, AUPR). We hypothesized that 8-mers may have played a relatively less important role in our prediction model. To further investigate this, we examined the final set of features selected by our method for each TF. Indeed, we found that only 31.6% of selected features are 8-mers, significantly lower than the frequency (75%) that would be expected if the k-mers were randomly chosen. In contrast, 7-mers are significantly enriched in selected features (44.6% observed vs. 18.8% expected). Surprisingly, the feature set is mostly enriched with 3-mers and 4-mers (5.8 and 5.7-fold enrichment, respectively), which may represent half binding sites separated by gaps of variable lengths.

III. Detailed description of FeatureREDUCE

Introduction

FeatureREDUCE builds on the biophysical modeling framework of the MatrixREDUCE algorithm (Foat, Houshmandi et al. 2005; Foat, Morozov et al. 2006; Foat, Tepper et al. 2008). In MatrixREDUCE, the DNA sequence specificity of a given transcription factor is represented as a position-specific affinity matrix (PSAM; (Bussemaker, Foat et al. 2007)), which is directly related to the differences in binding free energy associated with point mutations in the DNA sequence. Under the assumption of independence between nucleotide positions, the PSAM coefficients are directly inferred from a set of high-throughput measurements (mRNA expression, ChIP fold-enrichment, PBM intensity, etc) and their associated cis-regulatory sequences, which can be much longer than the length of a single binding site. In the biophysical model underlying MatrixREDUCE, the affinities of all possible binding sites within the longer sequence are added up, under the assumption that saturation of binding is weak to moderate.

FeatureREDUCE extends MatrixREDUCE in three distinct ways. First, it uses a more refined representation of binding specificity, in which dependencies between nucleotides are detected and modeled explicitly using additional free energy parameters. The resulting FSAM (feature-specific affinity model) can be used to predict the relative binding affinity for any oligomer of a specified length. Second, FeatureREDUCE accounts for certain biases that are specific to the PBM technology. Finally, it employs robust regression techniques, which prevents over-fitting and allows for improved estimation of biophysical parameters.

Methods

FeatureREDUCE employs a biophysical model in a robust regression framework to produce different types of models from a-3(a)13(m)hodeled expli

Supplementary Tables

Supplementary Table 1. Information on transcription factors and associated experiments
(see excel spreadsheet "STab01_TFsUsedInThisStudy.xlsx")

Team[1]	Model type[2]	Final Rank[3]	Pearson[4]	Pearson (Log)[5]	Spear-man[6]	AUROC 8mer[7]	AUPR 8mer[8]
Team_D	k-mer	1 (2)	0.641 (1)	0.674 (2)	0.639 (4)	0.994 (2)	0.700 (1)
Team_F	Other	2 (3.8)	0.610 (4)	0.673 (3)	0.655 (3)	0.976 (4)	0.545 (5)
Team_E	PWM	3 (4)	0.637 (2)	0.694 (1)	0.673 (2)	0.952 (7)	0.522 (8)
Team_G	k-mer	4 (4.4)	0.573 (6)	0.621 (6)	0.574 (6)	0.994 (1)	0.674 (3)
Team_J	Other	5 (5)	0.612 (3)	0.650 (4)	0.623 (5)	0.965 (6)	0.524 (7)
Team_I	Other	6 (6)	0.581 (5)	0.647 (5)	0.692 (1)	0.940 (9)	0.306 (10)
Team_C	Other	7 (7.8)	0.518 (8)	0.523 (10)	0.484 (10)	0.975 (5)	0.530 (6)
Team_H	Other	7 (7.8)	0.469 (10)	0.417 (12)	0.367 (12)	0.991 (3)	0.676 (2)
Team_9	Other	9 (8.4)	0.497 (9)	0.575 (7)	0.562 (7)	0.941 (8)	0.248 (11)
Team_A	k-mer	10 (9)	0.533 (7)	0.461 (11)	0.431 (11)	0.925 (12)	0.584 (4)
Team_K	k-mer	11 (10.2)	0.461 (11)	0.540 (9)	0.531 (9)	0.930 (10)	0.156 (12)
Team_12	k-mer	12 (10.4)	0.461 (12)	0.544 (8)	0.538 (8)	0.929 (11)	0.150 (13)
Team_B	PWM	13 (12.2)	0.267 (13)	0.189 (13)	0.100 (13)	0.891 (13)	0.462 (9)
Team_14	PWM	14 (14)	0.000 (14)	0.000 (14)	0.000 (14)	0.487 (14)	0.003 (14)

Supplementary Table 2. Results of the original DREAM5 challenge.

[1] ID of contest participant

[2] Type of model employed by method

[3] Final rank of team. Mean rank across five scoring schemes is indicated in parentheses.

[4] Pearson correlation between predicted probe intensities and actual intensities (average across all 66 experiments). Rank indicated in parentheses.

[5] Pearson correlation between the log of the predicted probe intensities and the log of the actual intensities (average across all 66 experiments). Rank indicated in parentheses.

[6] Spearman rank correlation between predicted probe intensities and actual intensities (average across all 66 experiments). Rank indicated in parentheses.

[7] Area under the receiver operating characteristic curve (AUROC) of high-scoring 8-mers.

[8] Area under the precision-recall curve (AUPR) of high-scoring 8-mers.

Supplementary Table 3. Full evaluations for all algorithms, by TF
(see excel spreadsheet "STab03_FullEvaluationResultsByTF.xlsx")

			Feature-REDUCE			BEEML-PBM		
TF ID	**TF Name (DBD)**	**dinuc[1]**	**PWM[2]**	**Im-prove[3]**	**dinuc[1]**	**PWM[2]**	**Im-prove[3]**	**Homodimer-ization?[4]**
TF_66	Zscan10 (C2H2 ZF)	0.609	0.458	**0.151**	0.693	0.674	**0.019**	Direct (22735705)
TF_46	Nhlh2 (bHLH)	0.695	0.576	**0.119**	0.620	0.525	**0.095**	Direct (18356286)
TF_47	Nkx2-9 (Homeo)	0.989	0.886	**0.103**	0.968	0.961	0.007	None
TF_61	Zfp637 (C2H2 ZF)	0.878	0.781	**0.097**	0.744	0.632	**0.112**	None
TF_63	Zkscan5 (C2H2 ZF)	0.664	0.575	**0.089**	0.384	0.301	**0.083**	None
TF_44	Gata4 (GATA)	0.818	0.730	**0.088**	0.980	0.969	0.011	Family-based
TF_58	Tbx1 (T-box)	0.762	0.686	**0.076**	0.672	0.673	-0.001	None
TF_27	Xbp1 (bZIP)	0.910	0.840	**0.070**	0.998	0.927	**0.071**	Direct (17765680)
TF_38	Dmrtc2 (DM)	0.935	0.865	**0.070**	0.817	0.837	-0.020	Family-based
TF_36	Atf4 (bZIP)	0.796	0.731	**0.065**	0.602	0.571	**0.031**	Direct (1827203)
TF_32	Zkscan1 (C2H2 ZF)	0.844	0.779	**0.065**	0.849	0.476	**0.373**	None
TF_60	Zfp300 (C2H2 ZF)	0.770	0.709	**0.061**	0.592	0.479	**0.113**	None
TF_12	Nr4a2 (NR)	0.917	0.867	**0.050**	0.963	0.938	**0.025**	Direct (21316423)
TF_35	Atf3 (bZIP)	0.817	0.770	**0.047**	0.864	0.853	0.011	Direct (8622660)
TF_10	Nfil3 (bZIP)	0.885	0.839	**0.046**	0.909	0.903	0.006	Direct (16725346)
TF_49	Nr2f1 (NR)	0.984	0.940	**0.044**	0.932	0.919	0.013	Direct (10624948)
TF_30	Zfp3 (C2H2 ZF)	0.606	0.564	**0.042**	0.718	0.722	-0.004	Direct (9155026)
TF_21	Srebf1 (bHLH)	0.936	0.898	**0.038**	0.988	0.951	**0.037**	Direct (15550381)
TF_65	Zscan10 (C2H2 ZF)	0.813	0.776	**0.037**	0.978	0.982	-0.004	Direct (16767105)
TF_18	Sox10 (Sox)	0.961	0.925	**0.036**	0.914	0.919	-0.005	Direct (10931919)
TF_57	Sp140 (SAND)	0.844	0.810	**0.034**	0.756	0.758	-0.002	None
TF_50	Nr5a2 (NR)	0.941	0.908	**0.033**	0.957	0.957	0.000	Family-based
TF_37	Dnajc21 (C2H2 ZF)	0.838	0.808	**0.030**	0.818	0.798	**0.020**	None

TF_62	Zic5 (C2H2 ZF)	0.813	0.783	**0.030**	0.992	0.976	**0.016**	None
TF_48	Nr2e1 (NR)	0.991	0.963	**0.028**	0.910	0.908	0.002	Direct (8047143)
TF_3	Foxo6 (Frkhead)	0.975	0.949	**0.026**	0.957	0.947	0.010	None
TF_28	Zfp202 (C2H2 ZF)	0.901	0.876	**0.025**	0.999	0.980	**0.019**	None
TF_39	Egr3 (C2H2 ZF)	0.902	0.878	**0.024**	0.960	0.960	0.000	None
TF_55	Sdccag8 (AT hook)	0.991	0.968	**0.023**	0.950	0.944	0.006	None
TF_34	Ahctf1 (AT hook)	0.927	0.906	**0.021**	0.803	0.806	-0.003	None
TF_53	Rfx7 (RFX)	0.941	0.920	**0.021**	0.999	0.950	**0.049**	None
TF_41	Esrrg (NR)	0.997	0.977	**0.020**	0.963	0.941	**0.022**	Direct (12180985)
TF_19	Sox3 (Sox)	0.916	0.896	**0.020**	0.886	0.878	0.008	None
TF_29	Zfp263 (C2H2 ZF)	0.925	0.908	**0.017**	0.978	0.959	**0.019**	None
TF_7	Mlx (bHLH)	0.996	0.979	**0.017**	0.932	0.926	0.006	Direct (11230181)
TF_56	Snai1 (C2H2 ZF)	0.981	0.966	**0.015**	0.956	0.948	0.008	None
TF_16	Prdm11 (Myb)	0.995	0.980	**0.015**	0.956	0.952	0.004	None
TF_59	Zbtb1 (C2H2 ZF)	0.982	0.967	**0.015**	0.978	0.976	0.002	None
TF_54	Rora (NR)	0.967	0.952	**0.015**	0.970	0.978	-0.008	Direct (7935491)
TF_51	Pou1f1 (Pou+ Homeo)	0.925	0.911	0.014	0.947	0.932	**0.015**	Direct (10026784)
TF_43	Foxg1 (Frkhead)	0.930	0.916	0.014	0.870	0.860	0.010	None
TF_6	Klf9 (C2H2 ZF)	0.851	0.837	0.014	0.952	0.954		

Dataset	Algorithm	Probes 1[1]	Probes 2[2]	Probes 1+2[3]	8mers 1[4]	8mers 2	8mers 1+2
DREAM	BEEML-PBM	0.540	0.480	0.533	0.701	0.621	0.727
DREAM	FeatureREDUCE	0.504	0.338	0.491	0.641	0.390	0.654
Badis09	BEEML-PBM	0.573	0.485	0.556	0.742	0.619	0.760
Badis09	FeatureREDUCE	0.543	0.371	0.530	0.723	0.433	0.730

Supplementary Table 5. Summary of evaluation of secondary motifs, compared to only using primary PWMs.

[1] Correlation between primary motif probe intensity predictions and test probe intensities (mean across all TFs)

[2] Correlation between secondary motif probe intensity predictions and test probe intensities (mean across all TFs)

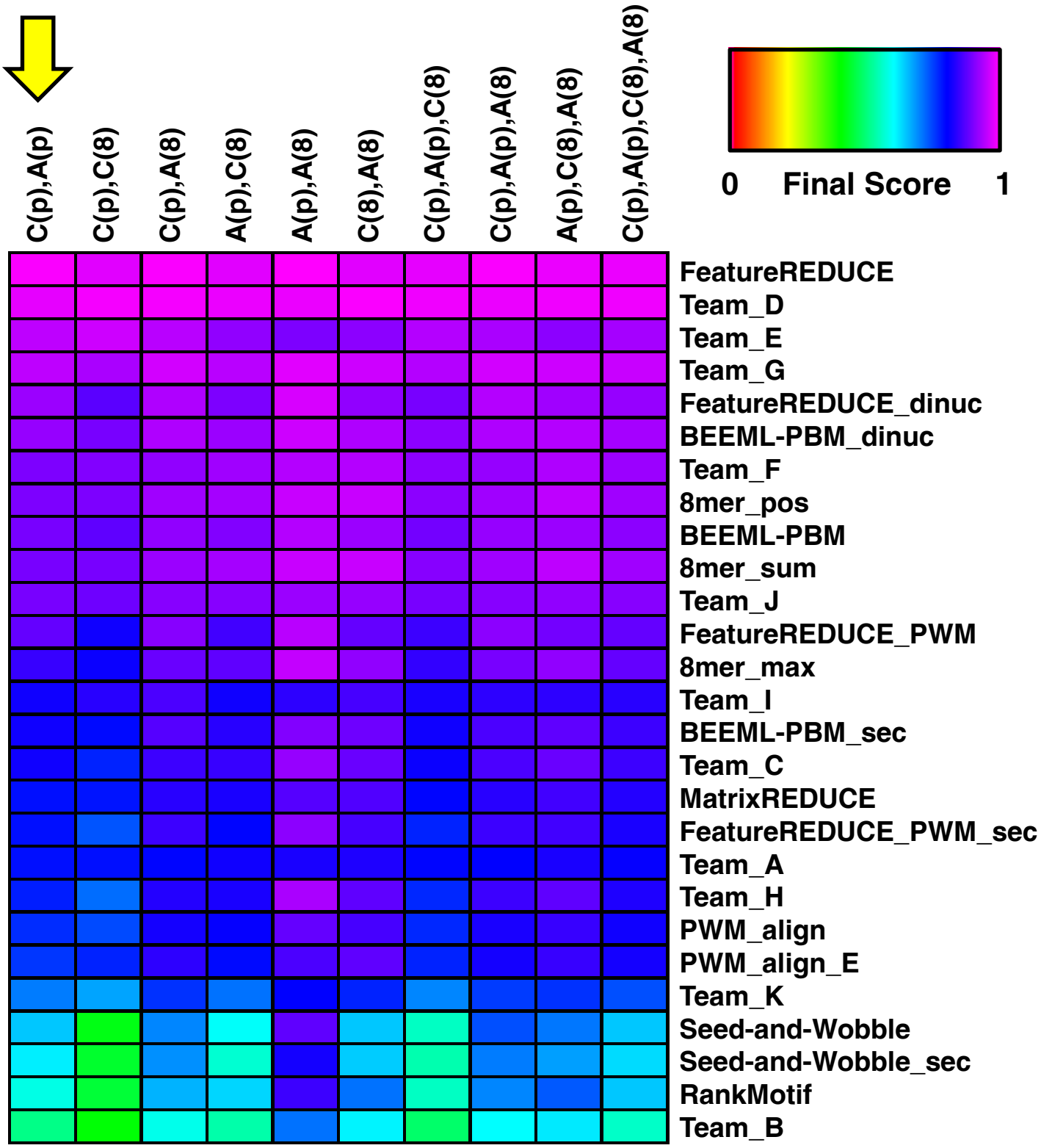
[3] Correlation between combined primary and secondary motif probe intensity predictions and test probe intensities (mean across all TFs)

[4] Same as for probe columns, but first converting the probe intensities to median 8-mer intensities (for both the predictions and the test arrays), and then calculating the correlations (see Methods)

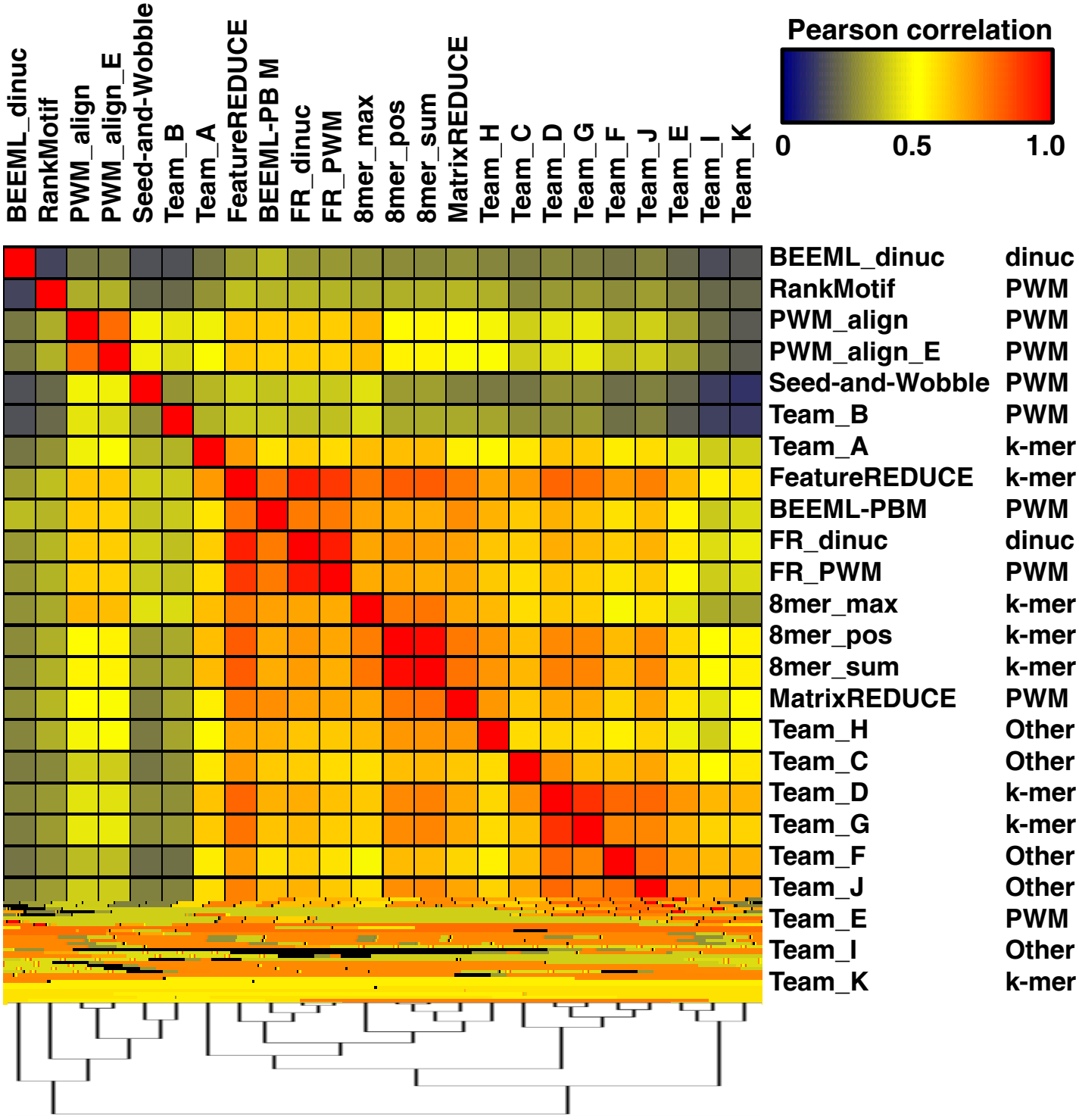
Supplementary Table 6. Improvement of secondary over primary motifs, for each TF
(see excel spreadsheet "STab06_SecondaryMotifsByTF.xlsx")

Supplementary Table 7. Full Comparison to ChIP-seq and ChIP-exo data
(see excel spreadsheet "STab07_FullComparisonToInVivoData.xlsx")

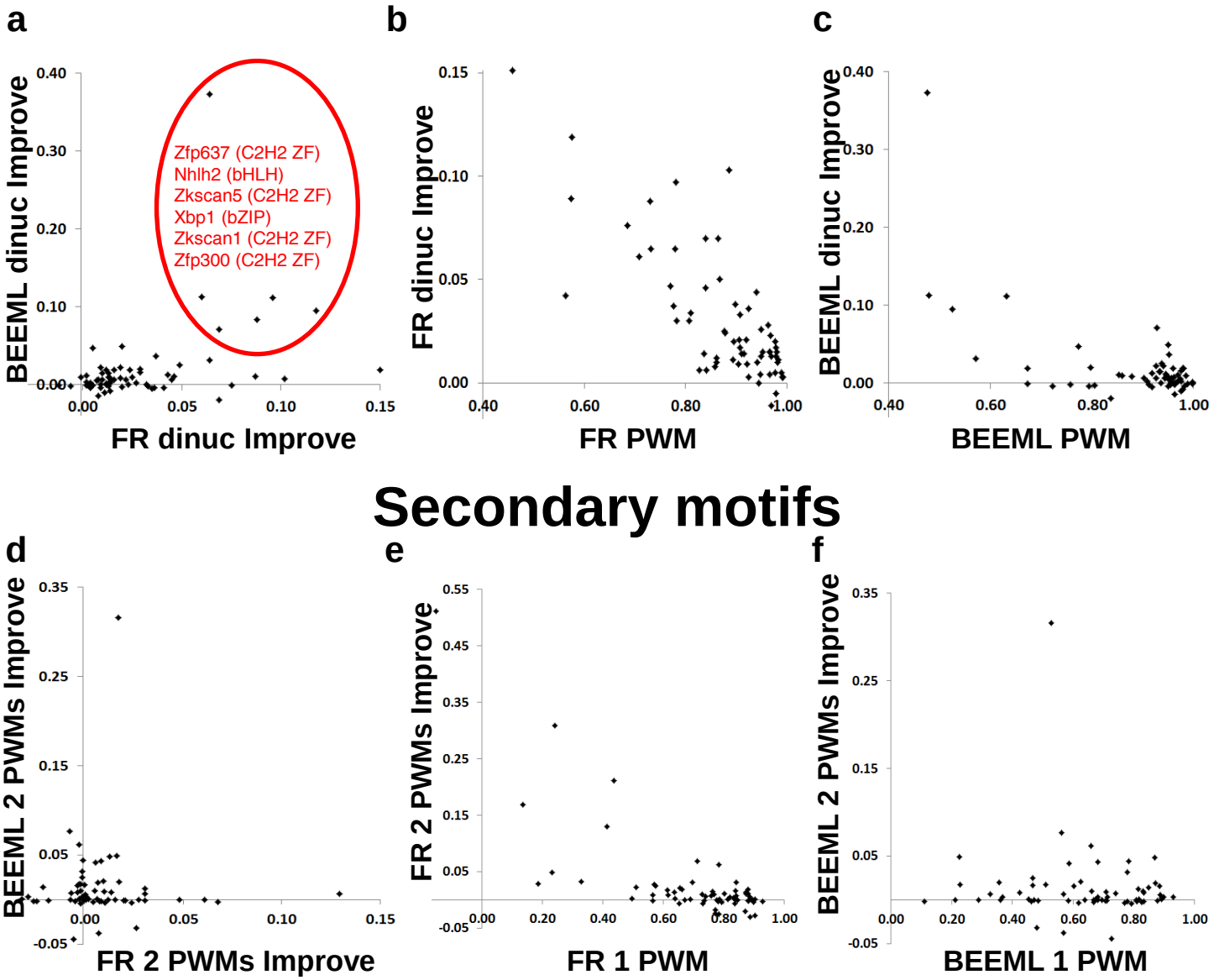
Supplementary Table 8. Information on plasmids used for PBMs in this study
(see excel spreadsheet "STab08_PlasmidInfo.xlsx")



Supplementary Figure 1. The effect of using different combinations of evaluation schemes on the final scores of the algorithms.
Final score of each algorithm, using all possible combinations of the four evaluation schemes. Final combination used is indicated by the yellow arrow. Abbreviations: C(p), probe correlation; C(8), 8-mer correlation; A(p), probe AUROC; A(8), 8-mer AUROC.

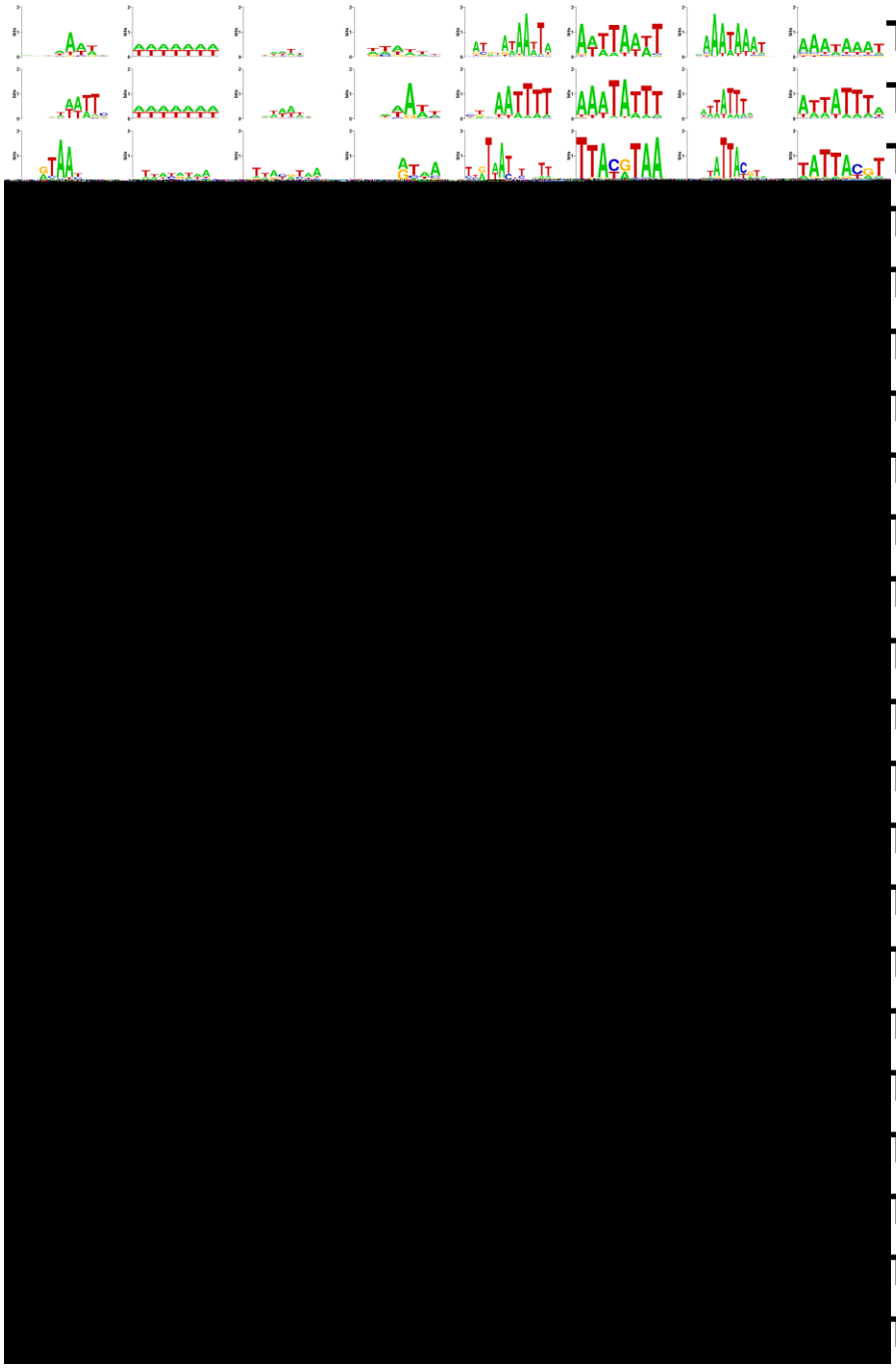


Supplementary Figure 2. Correlation of algorithm predictions
Heatmap depicting the overall similarity of the probe intensities produced by each pair of algorithms. Prediction similarity was calculated as the average across all 66 experiments of the pearson correlation between the probe intensity predictions of each algorithm pair. The tree at the bottom indicates the results of hierarchical clustering using average linkage clustering agglomeration.



Supplementary Figure 3. Comparison of dinucleotide and secondary motif improvement

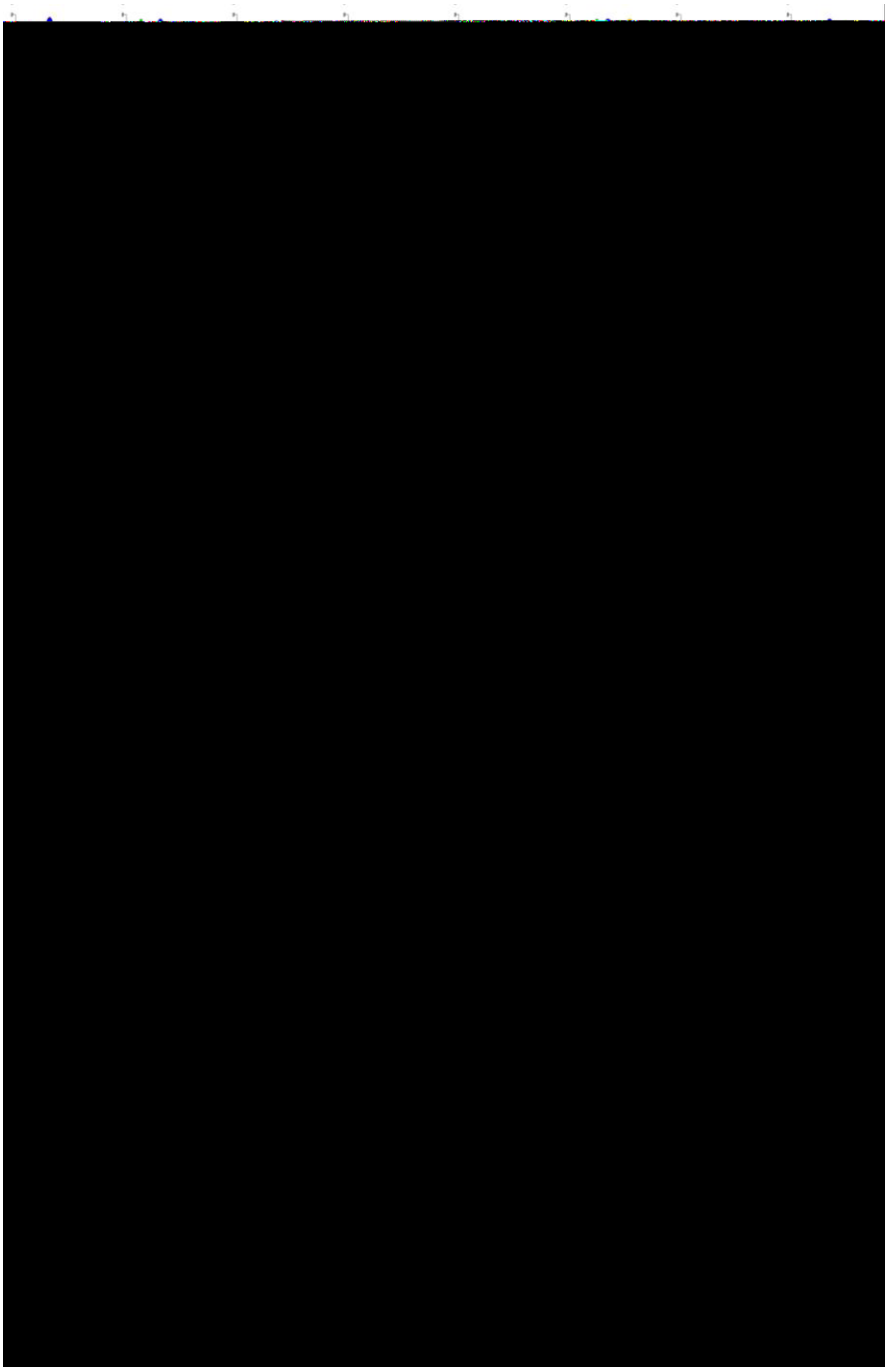
Improvement achieved for each TF for BEEML-PBM (BEEML) and FeatureREDUCE (FR), for dinucleotides (**a** though **c**) and secondary motifs (**d** through **f**). For dinucleotides, scores are based on the final evaluation score. For secondary motifs, scores are based on the Pearson correlation of 8-mer predictions (see Methods). **a.** Improvement of dinucleotides over a single PWM, for FR (x axis) and BEEML (y axis), for each TF. The six TFs with substantial improvement for both FR and BEEML are indicated. **b.** Performance of FR single PWM predictions (x axis) vs the improvement of FR dinucleotide predictions over predictions from a single FR PWM (y axis). **c.** Same as (**b**), but for BEEML. **d-f.** Same as **a-c**, but for secondary motifs.



ID	Name	Family
TF_34	Ahctf1	AT_hook
TF_55	Sdccag8	AT_hook
TF_10	Nfil3	bZIP_1
TF_27	Xbp1	bZIP_1
TF_2	Dbp	bZIP_1
TF_35	Atf3	bZIP_1
TF_36	Atf4	bZIP_1
TF_38	Dmrtc2	DM
TF_3	Foxo6	Fork_head
TF_42	Foxc2	Fork_head
TF_43	Foxg1	Fork_head
TF_44	Gata4	GATA
TF_21	Srebf1	HLH
TF_26	Tfec	HLH
TF_46	Nhlh2	HLH
TF_7	Mlx	HLH
TF_18	Sox10	HMG_box
TF_19	Sox3	HMG_box
TF_20	Sox6	HMG_box
TF_47	Nkx2-9	Homeobox
TF_14	P42pop	Myb
TF_16	Prdm11	Myb

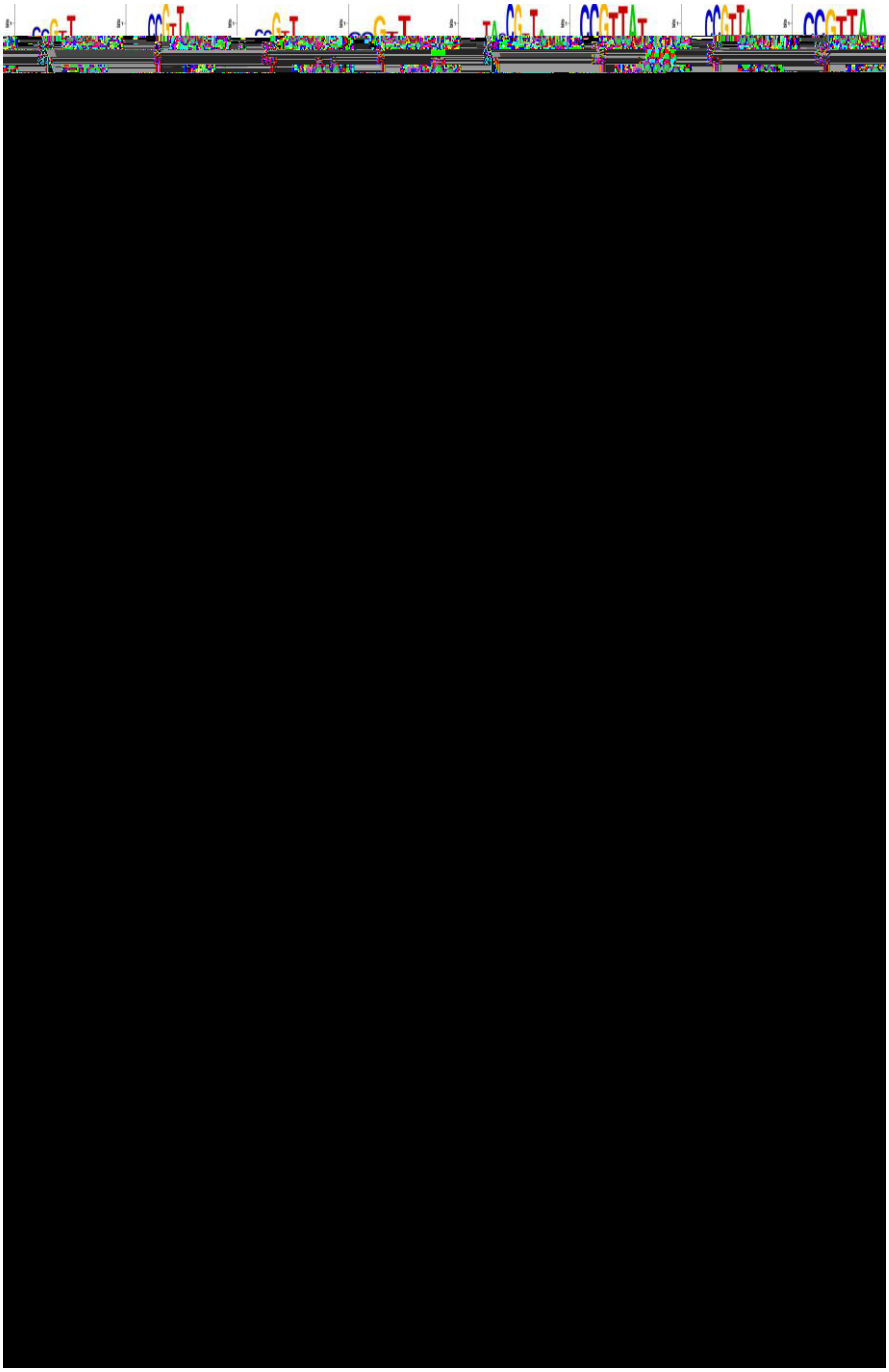
Supplementary Figure 4. PWM sequence logo comparisons (cont'd on subsequent pages)

Visual display of PWMs produced by each algorithm for each TF. A red 'X' indicates that the given algorithm did not produce a PWM for the given TF. Algorithm key is displayed at the top. Abbreviations: BML, BEEML-PBM; TM_E, Team_E; FR, FeatureREDUCE; MR, MatrixREDUCE; RM, RankMotif; SnW, Seed-and-Wobble; PA_E, PWM_align_E; PA, PWM_align.



ID	Name	Family
TF_45	Mybl2	Myb
TF_13	Oct1	Pou+Hom
TF_15	Pit1	Pou+Hom
TF_51	Pou1f1	Pou+Hom
TF_53	Rfx7	RFX
TF_57	Sp140	SAND
TF_22	Tbx2	T-box
TF_23	Tbx20	T-box
TF_24	Tbx4	T-box
TF_25	Tbx5	T-box
TF_58	Tbx1	T-box
TF_9	Mzf1	zf-C2H2
TF_63	Zkscan5	zf-C2H2
TF_31	Zfx	zf-C2H2
TF_33	Zscan10	zf-C2H2
TF_60	Zfp300	zf-C2H2
TF_37	Dnajc21	zf-C2H2
TF_39	Egr3	zf-C2H2
TF_4	Klf12	zf-C2H2
TF_5	Klf8	zf-C2H2
TF_64	Znf740	zf-C2H2
TF_65	Zscan10	zf-C2H2

Supplementary Figure 4. PWM sequence logo comparisons (cont'd)



Supplementary Figure 4. PWM sequence logo comparisons (cont'd)

References

Agius, P., A. Arvey, et al. (2010). "High Resolution Models of Transcription Factor-DNA Affinities Improve *In Vitro* and *In Vivo* Binding Predictions." PLoS Comput Biol 6(9): e1000916.

Annala, M., K. Laurila, et al. (2011). "A linear model for transcription factor binding affinity prediction in protein binding microarrays." PLoS ONE: in Review.

Badis, G. et al (2008). “A library of yeast transcription factor motifs reveals a widespread function for Rsc3 in targeting nucleosome exclusion at promoters.” Mol Cell 32, 878-887.

Badis, G. et al (2009). “Diversity and complexity in DNA recognition by transcription factors.” Science 324, 1720-1723.

Bailey, T. L. and C. Elkan (1994). "Fitting a mixture model by expectation maximization to discover motifs in biopolymers." Proceedings of the International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology 2: 28-36.

Berger, M.F. et al (2006). “Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities.” Nat Biotechnol 24, 1429-1435.

Berger, M. F. and M. L. Bulyk (2006). Protein Binding Microarrays (PBMs) for Rapid, High-Throughput Characterization of the Sequence Specificities of DNA Binding Proteins. Gene Mapping, Discovery, and Expression. M. Bina, Humana Press. 338: 245-260.

Berger, M.F. et al (2008). “Variation in homeodomain DNA binding revealed by high-resolution analysis of sequence preferences.” Cell 133, 1266-1276.

Breiman, L. (2001). "Random Forests." Machine Learning 45(1): 5-32.

Bussemaker, H. J., B. C. Foat, et al. (2007). "Predictive modeling of genome-wide mRNA expression: from modules to molecules." Annu Rev Biophys Biomol Struct 36: 329-347.

Bussemaker, H. J., H. Li, et al. (2001). "Regulatory element detection using correlation with expression." Nat Genet 27(2): 167-174.

Cerquides, J. and R. Mántaras (2005). Robust Bayesian Linear Classifier Ensembles. Machine Learning: ECML 2005. J. Gama, R. Camacho, P. Brazdil, A. Jorge and L. Torgo, Springer Berlin / Heidelberg. 3720: 72-83.

Chang, C.-c. and C.-j. Lin (2001). LIBSVM: a library for support vector machines.

Chen, C.-Y., H.-K. Tsai, et al. (2008). "Discovering gapped binding sites of yeast transcription factors." Proceedings of the National Academy of Sciences 105(7): 2527-2532.

Chen, X., T. R. Hughes, et al. (2007). "RankMotif++: a motif-search algorithm that accounts for relative ranks of K-mers in binding transcription factors." Bioinformatics 23(13): i72-i79.

Efron, B. and R. J. Tibshirani (1994). An Introduction to the Bootstrap.

Foat, B. C., S. S. Houshmandi, et al. (2005). "Profiling condition-specific, genome-wide regulation of mRNA stability in yeast." Proc Natl Acad Sci U S A 102(49): 17675-17680.

Foat, B. C., A. V. Morozov, et al. (2006). "Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE." Bioinformatics 22(14): e141-149.

Foat, B. C., R. G. Tepper, et al. (2008). "TransfactomeDB: a resource for exploring the nucleotide sequence specificity and condition-specific regulatory activity of trans-acting factors." Nucleic Acids Res 36(Database issue): D125-131.

Fordyce, P.M. et al (2010). “De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis.” Nat Biotechnol 28, 970-975.

Grau, J. (2010). Discriminative Bayesian principles for predicting sequence signals of gene regulation. Ph.D., Martin Luther University Halle-Wittenberg.

Halperin, Y., C. Linhart, et al. (2009). "Allegro: Analyzing expression and sequence in concert to discover regulatory programs." Nucleic Acids Research 37(5): 1566-1579.

Keilwagen, J., J. Grau, et al. (2011). "De-Novo Discovery of Differentially Abundant Transcription Factor Binding Sites Including Their Positional Preference." PLoS Comput Biol 7(2): e1001070.

Kel, A. E., E. Gößling, et al. (2003). "MATCHTM: a tool for searching transcription factor binding sites in DNA sequences." Nucleic Acids Research 31(13): 3576-3579.

Keleş, S., M. van der Laan, et al. (2002). "Identification of regulatory elements using a feature selection method." Bioinformatics 18(9): 1167-1175.

Kinney, J. B., A. Murugan, et al. (2010). "Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence." Proceedings of the National Academy of Sciences 107(20): 9158-9163.

Kinney, J. B., G. Tkačik, et al. (2007). "Precise physical models of protein–DNA interaction from high-throughput data." Proceedings of the National Academy of Sciences 104(2): 501-506.

Kursa, M. B., A. Jankowski, et al. (2010). "Boruta A System for Feature Selection." Fundamenta Informaticae 101(4): 271-185.

Kursa, M. B. and W. R. Rudnicki (2010). "Feature Selection with the Boruta Package." Journal Of Statistical Software 36(11).

Lavoie, H. et al (2010). “Evolutionary tinkering with conserved components of a transcriptional regulatory network.” PLoS Biol 8, e1000329.

Linhart, C., Y. Halperin, et al. (2008). "Transcription factor and microRNA motif discovery: The Amadeus platform and a compendium of metazoan target sets." Genome Research 18(7): 1180-1189.

MacIsaac, K.D. et al (2006). “An improved map of conserved regulatory sites for Saccharomyces cerevisiae.” BMC Bioinformatics 7, 113.

Maerkl, S. J. and S. R. Quake (2007). "A systems approach to measuring the binding energy landscapes of transcription factors." Science 315(5809): 233-237.

Morozov, A.V. & Siggia, E.D. (2007). “Connecting protein structure with predictions of regulatory sites.” Proc Natl Acad Sci U S A 104, 7068-7073.

Nutiu, R. et al (2011). “Direct measurement of DNA affinity landscapes on a high-throughput sequencing instrument.” Nat Biotechnol 29, 659-664.

Schütz, F. and M. Delorenzi (2008). "MAMOT: hidden Markov modeling tool." Bioinformatics 24(11): 1399-1400.

Siggers, T., Duyzend, M.H., Reddy, J., Khan, S. & Bulyk, M.L. (2011). “Non-DNA-binding cofactors enhance DNA-binding specificity of a transcriptional regulatory complex.” Molecular systems biology 7, 555.

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso." Journal of the Royal Statistical Society. Series B (Methodological) 58(1): 267-288.

Zhao, Y., Granas, D. & Stormo, G.D. (2009). “Inferring binding energies from selected binding sites.” PLoS Comput Biol 5, e1000590.

Zhao, Y. and G. D. Stormo (2011). "Quantitative analysis demonstrates most transcription factors require only simple models of specificity." Nat Biotechnol 29(6): 480-483.

Zhou, X. and E. K. O'Shea (2011). "Integrated approaches reveal determinants of genome-wide binding and function of the transcription factor Pho4." Mol Cell 42(6): 826-836.

Zhu, C. et al. (2009). “High-resolution DNA-binding specificity analysis of yeast transcription factors.” Genome Res 19, 556-566.