Machine learning methods to predict transcription factor family from binding specificity

Andrew B. Wilk

The Rohs Lab, Department of Quantitative and Computational Biology

University of Southern California, Los Angeles, CA, USA

Quantitative Biology Honors Thesis

May 2025

Introduction

Transcription factor proteins (TFs) directly interpret the genome, performing the first step in decoding the DNA sequence[1]. They bind to DNA in a sequence-specific manner to regulate processes that are essential to all life, including transcription and genome organization. Moreover, mutations in TFs and TF binding sites underlie many human diseases[2]. Therefore, determining how TFs identify binding sites and modulate transcriptional networks is key to understanding the finely tuned expression patterns of complex life[3].

TFs cannot be understood functionally without knowledge of the DNA sequences they bind, as identification of potential binding sites provides a gateway to further analyses[2]. TF DNA-binding specificities are commonly represented as "motifs"—models representing the set of related DNA sequences preferred by a given TF. Motifs are usually visualized as sequence logos, which are generated from position weight matrices (PWMs) encoding the relative likelihood of each nucleotide at every binding site position.

A motif is generated from quantitative affinity measurements for a large number of sequences. Modern *in vitro* methods for this purpose include protein-binding microarray[4], system evolution of ligands by exponential enrichment combined with high-throughput sequencing (SELEX-seq)[5], or high-throughput SELEX[6], while *in vivo* methods such as chromatin immunoprecipitation followed by sequencing[7] are used as well. There are various databases such as JASPAR[8], TRANSFAC[9], and CisBP[10], which catalog the known TFs and their motifs.

Another critical lens through which TF function may be examined is structure[11]. The determination of the three-dimensional structures of protein–DNA complexes has provided a detailed picture of binding, revealing a structurally diverse set of protein families that exploit a wide repertoire of interactions to recognize the double-helix[12]. The diversity of recognition mechanisms employed by TFs render a structural classification essential to unravelling the complexity of protein-DNA interactions. TFClass[13] is one such system, hierarchically classifying TFs based on structure according to the following schema: superclass, class, family, and subfamily.

Although binding specificity and structure are derived from vastly different experimental sources, their inexorable relationship lies at the heart of TF-DNA interactions and remains incompletely understood[14]. Recent studies have demonstrated the immense potential of machine learning methods to model complex relationships in genomics[15–17] and structural biology[18–20]. Our previous work presents DeepPBS[21], a geometric deep learning model of protein-DNA interactions, capable of predicting binding specificity from protein-DNA co-crystal structures. This study aims to extend this analysis by investigating the ability of machine learning to model the relationship in the reverse direction of DeepPBS: predicting structural information from binding specificity.

We demonstrate that transcription factor structural families can be predicted directly from motifs encoding binding specificity. We begin by benchmarking the task with a simple k-nearest neighbors classifier, then introduce a deep neural network. Both approaches achieve comparably high accuracy, with the neural network delivering predictions roughly a thousand times faster. Feature importance analysis shows that the deep model learns true family-specific binding modes. Finally, a comprehensive motif-similarity analysis reveals an inherent performance limit for predicting TF family from binding specificity—one that our classifiers appear to be approaching.

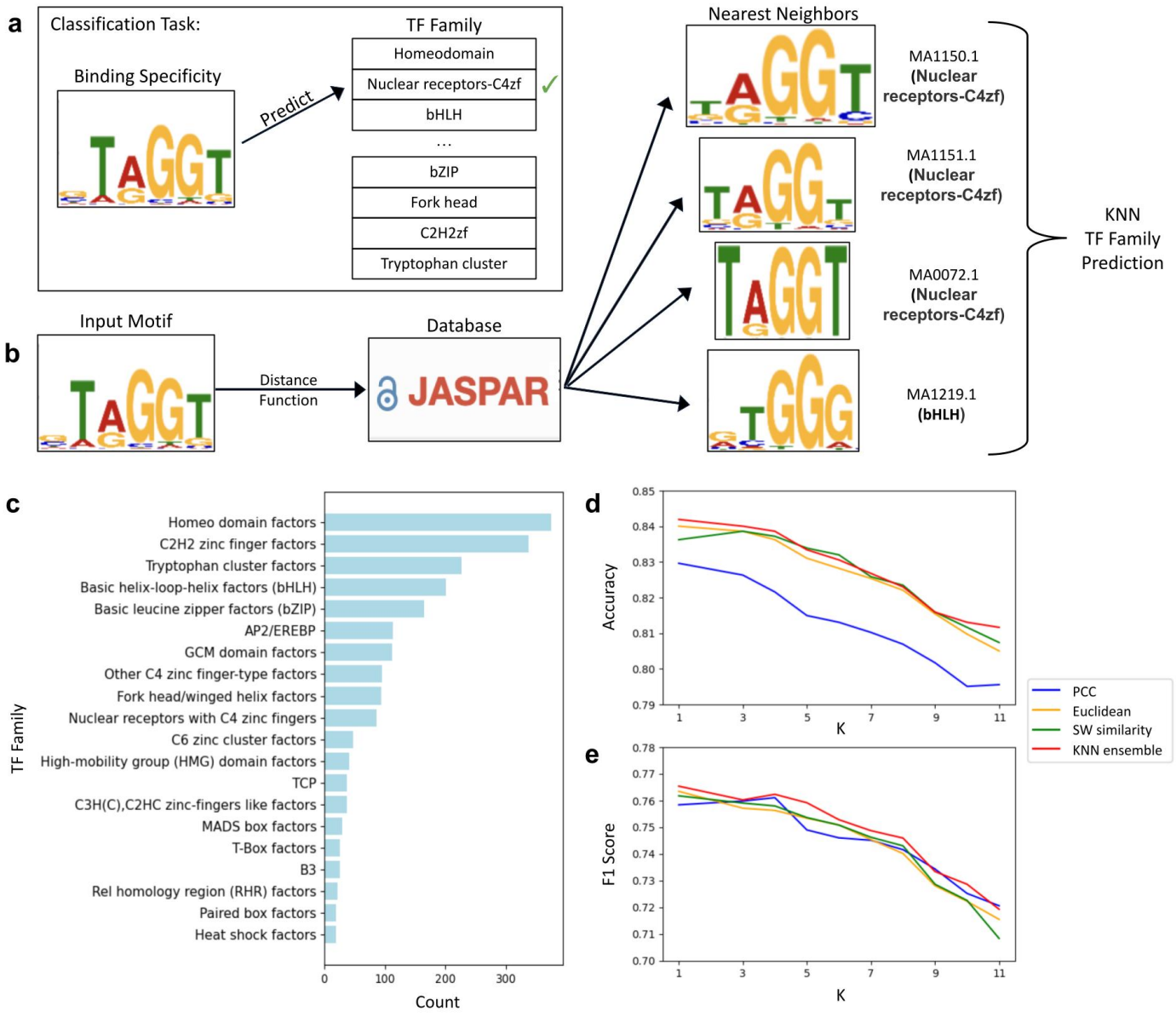

Figure 1 | Benchmarking TF family prediction with KNN. **a,** Illustration of classification task: predict TF family from motif encoding binding specificity. **b,** Illustration of k-nearest neighbors classifier, using Tomtom Motif Comparison Tool to calculate nearest neighbors. **c,** Evaluation dataset (from JASPAR), breakdown by TF family (n = 2113). Old versions of updated motifs, dimers, and motifs belonging to rare families (<20 motifs) are excluded. **d,e,** Accuracy and macro-F1 score for various KNN implementations on entire dataset. K is varied over [1...11], and 4 distance function are tested (Pearson correlation, Euclidean distance, Sandelin-Wasserman similarity, and a majority vote (ensemble) of the three functions).

Results

TF family prediction using KNN

To benchmark the task of predicting transcription factor family from binding specificity (Fig.1a), we implemented a k-nearest neighbors (KNN) classifier. KNN classification is a simple, non-parametric machine learning method that classifies data points based on the majority class of their K most similar data points in the dataset. To obtain distances between motifs for nearest neighbor calculations, we used the Tomtom Motif Comparison Tool[22] (Fig.1b). Tomtom compares query motif(s) against a dataset of known motifs and returns a ranked list of the significant matches. The Tomtom web server[23] offers three statistical measures for computing motif-motif similarity, or equivalently, distance: Pearson correlation coefficient (PCC), Euclidean distance, and Sandelin-Wasserman (SW) similarity. More information regarding how Tomtom calculates motif-motif similarity/distance can be found in the original report[22].

To evaluate the performance of KNN for TF family prediction, we used motifs from the JASPAR (2024) database[8]. Previous versions of motifs and those belonging to dimer or rare (< 20 motifs in JASPAR) families were excluded from the evaluation dataset, bringing its size to 2113 motifs across 20 TF families (Fig.1c). Using total accuracy and macro-F1 score as performance metrics, we evaluated various KNN implementations (Fig.1d,e). We tested K (number of neighbors considered) over the range [1, 2,...,11] and all previously mentioned distance metrics. We also defined and tested a "KNN ensemble" model, which uses the majority vote of the three distance functions for its prediction. Figure 1d,e show that lowering K tends to increase performance; as well, the various distance functions performed similarly, aside from PCC having marginally lower accuracy despite competitive F1 scores. Overall, the results show that the KNN ensemble with K = 1 is the most effective model by a small margin, with accuracy = 0.842 and F1 score = 0.765. This performance is a useful baseline, against which more complex models for TF family prediction can be tested.

Deep learning provides competitive performance and interpretable predictions

Next, we sought to evaluate the utility of deep learning for predicting TF family from binding specificity. This section explores model architecture, evaluation and training, and interpretability.

Model Architecture

Figure 2 shows the architecture of our deep learning approach in detail, highlighting its input preprocessing, feature learning, and output transformation stages. A key feature of the network is its lack of dense/fully connected layers, relying solely on convolutional layers for feature learning. Convolutional layers were chosen because of their demonstrated efficacy in genomics applications[15,16] and their compatibility with the varied dimensions of the input motifs (overall shape: 4xN). Another key feature of the model is its reverse complement (RC) invariance. It achieves this by passing both the input and the input RC through the feature learning layers separately, before averaging the outputs to obtain the final prediction. This way, the model learns one set of weights to handle the forward or reverse complement, agnostic to the true direction. A more detailed description of the model architecture is provided in Methods.

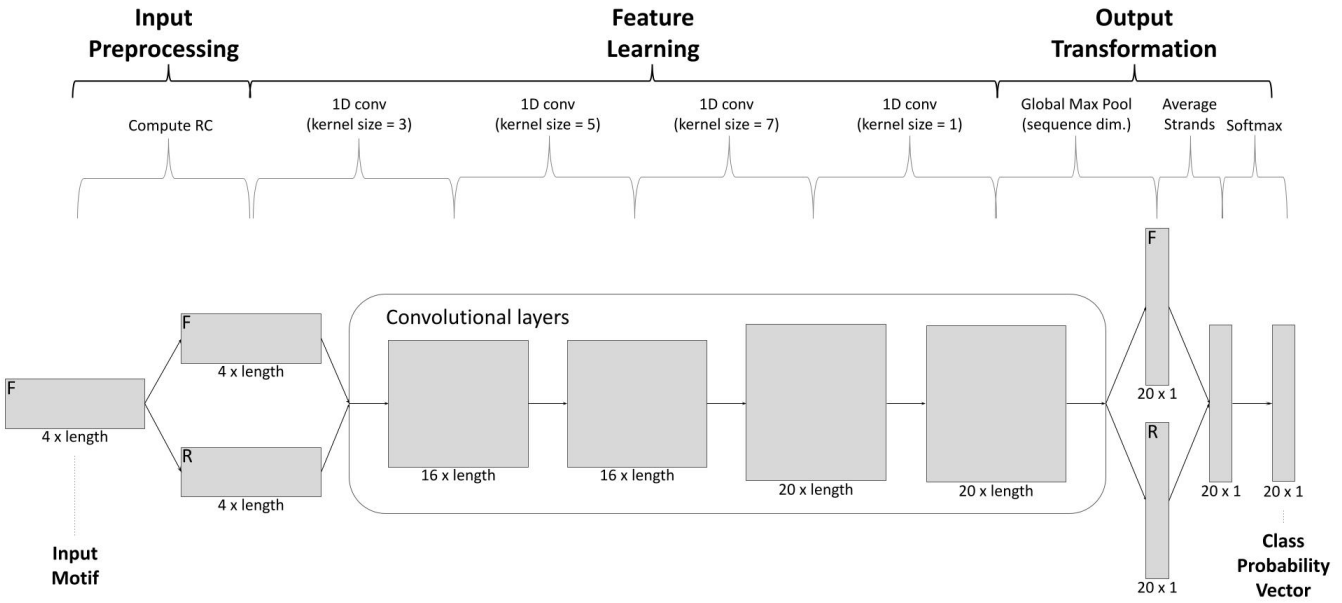

Figure 2 | Fully convolutional neural network for TF family prediction. Input motif is assumed to be forward strand. Compute reverse complement and separately feed both through convolutional layers. Number of channels, type of layer, and kernel size are specified in diagram. Each layer uses ReLU activation. After convolutional layers, a global max pool operation along sequence dimension reduces both feature maps, one for each strand, to 20x1 vectors. Average the results and apply softmax to obtain the final model output.

Evaluation and Training

We evaluated our deep learning model using stratified K-fold cross-validation on the same dataset as the KNN model (n = 2113; described above). We compared K = 5, 10, and 20 folds. For each choice of K, we trained K separate models—each on K–1 folds—and report the mean ± SD of each metric on the held-out fold. Training was stopped based on validation loss not improving for 7 epochs in a row. All runs used a learning rate of 1×10^{-3}, weight decay of 1×10^{-3}, batch size = 1, and cross-entropy loss. Full training and hyperparameter details are in Methods.

Figure 3a,b compare accuracy and macro-F1 score of the deep learning and highest performing KNN implementations. For accuracy, which can be dominated by majority classes, deep learning was outperformed by KNN, with KNN ensemble having an accuracy of 0.842 compared to a mean accuracy of 0.822 for the 20-fold CV deep model (Fig.3a). For F1 score, which balances precision and recall across all classes, deep learning outperformed KNN, with 10-fold CV having a mean macro-F1 of 0.785 compared to 0.766 for KNN ensemble (K = 1) (Fig.3b). Figure 3c compares the amount of time it takes to predict on the full dataset, showing that deep learning is one thousand times faster than KNN approaches. Figure 3e displays the per-family accuracies of the deep learning and KNN models, showing varied performance but a strong monotonic correlation between the two models. (Spearman ρ = 0.40; p = 0.081). Figure 3d plots class size against class accuracies for deep learning and KNN, revealing nearly equal linear associations (r = 0.30). In summary, relative to KNN, deep learning shows roughly equal accuracy and macro-F1, vastly improved speed, and correlated performance across TF families.

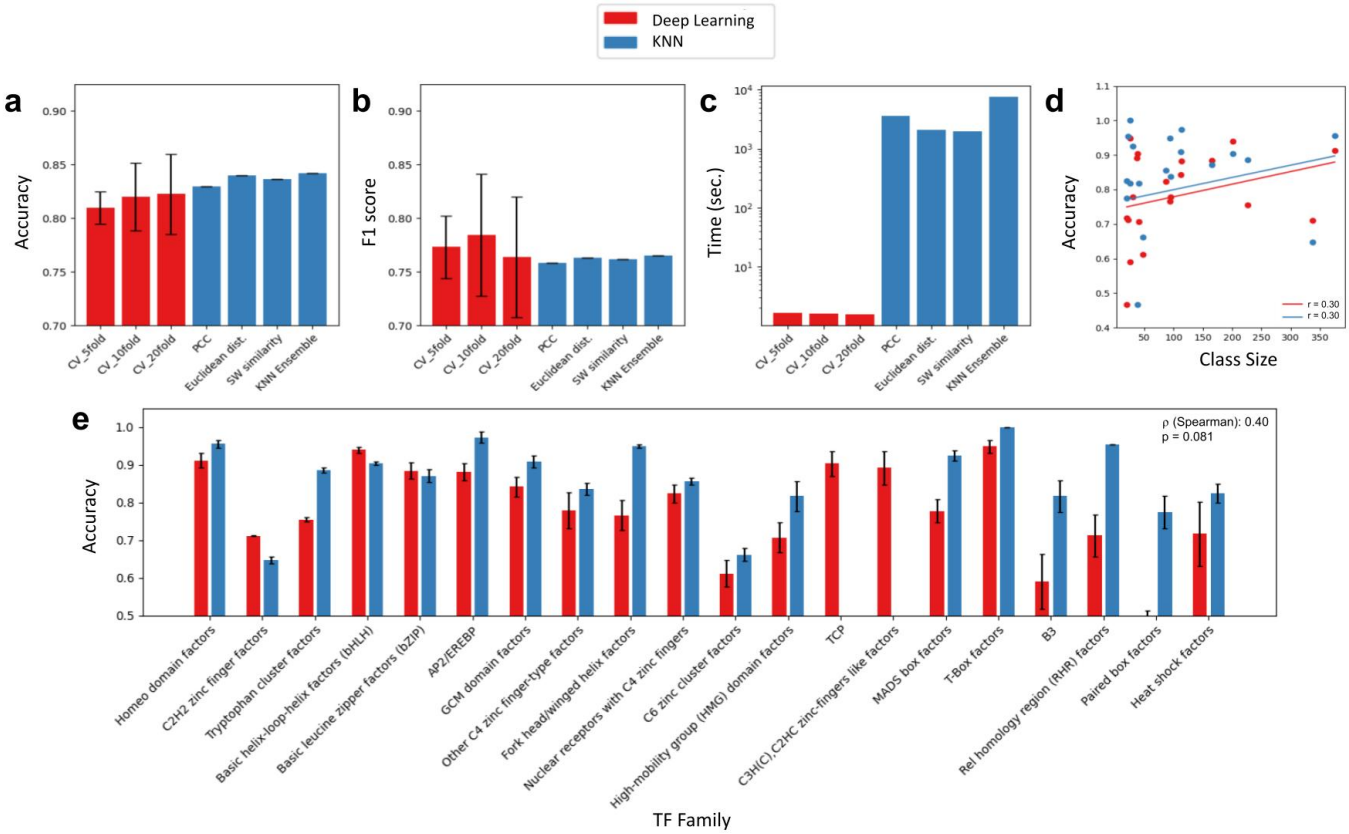

Figure 3 | Deep learning and KNN performance comparison. a, Accuracy on whole dataset, displaying different CV splits for deep learning (Red), and all KNN (Blue) distance functions. Error bars represent variance in model performance during CV. **b,** Macro-F1 on whole dataset, showing same models as **a**. **c,** Time to obtain predictions on whole dataset (seconds, log scale). For deep learning, timed forward passes through model. For KNN, timed (n^2) distance calculations. **d,** Plotting per class accuracy versus class size, for deep learning (Red) and KNN (Blue). Both have Pearson correlation r = 0.30. **e,** Comparing per-class accuracy for deep learning and KNN. Obtained mean accuracy for each CV regimen or distance function, then calculated total mean and standard deviation, as shown on plot. Spearman correlation between per class accuracies ρ = 0.40, p = 0.081.

Interpretability

A common critique of deep neural networks is their lack of interpretability—their internal decision structure often likened to a "black box"[24]. To ameliorate this limitation, we calculated feature importance scores using the Integrated Gradients algorithm[25]. By summing importance along the columns and normalizing across positions, we implemented relative importance scores for each nucleotide position.

Figure 4 presents the feature importance profiles for eight correctly predicted motifs, drawn equally from the bHLH and homeodomain families. For each motif, we plot the relative importance at each nucleotide position. In the bHLH examples (Fig.4a), which conform to the family-specific CANNTG[26] consensus, importance peaks around the conserved CA and TG dinucleotides, while the variable N positions carry minimal weight. In the homeodomain examples (Fig.4b), which conform to the family-specific NTAATNN[27] consensus, importance peaks in the conserved TAAT core. In both cases, the model captures each family's canonical recognition pattern, demonstrating that its internal attributions reflect true family-specific binding modes.

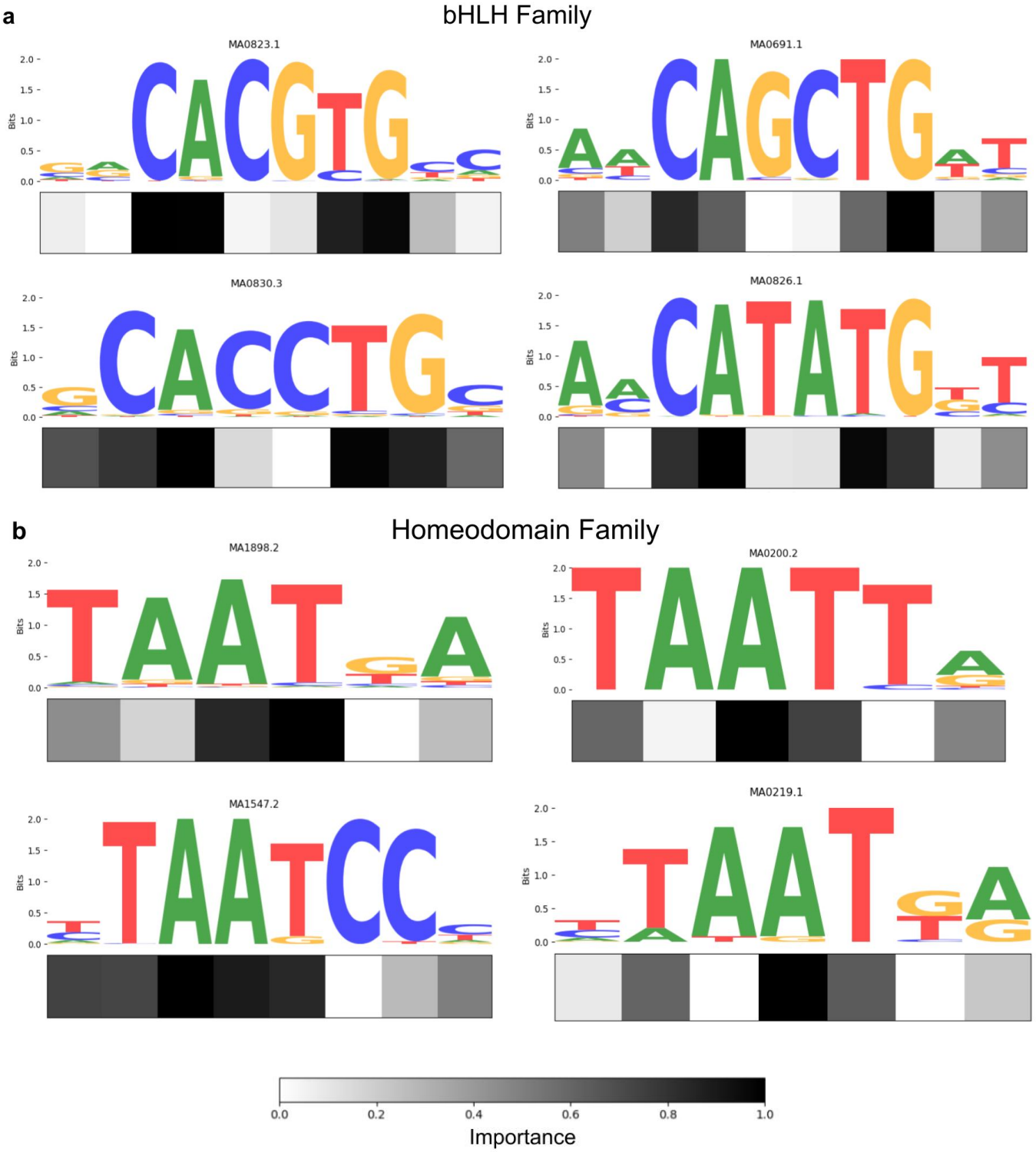

Figure 4 | Feature importance analysis reveals family-specific binding modes. a, Importance profiles for four representative bHLH motifs (CANNTG consensus). **b,** Importance profiles for four representative homeodomain motifs (NTAATNN consensus). Importance scores were obtained by summing Integrated Gradients attributions along columns and normalizing across positions.

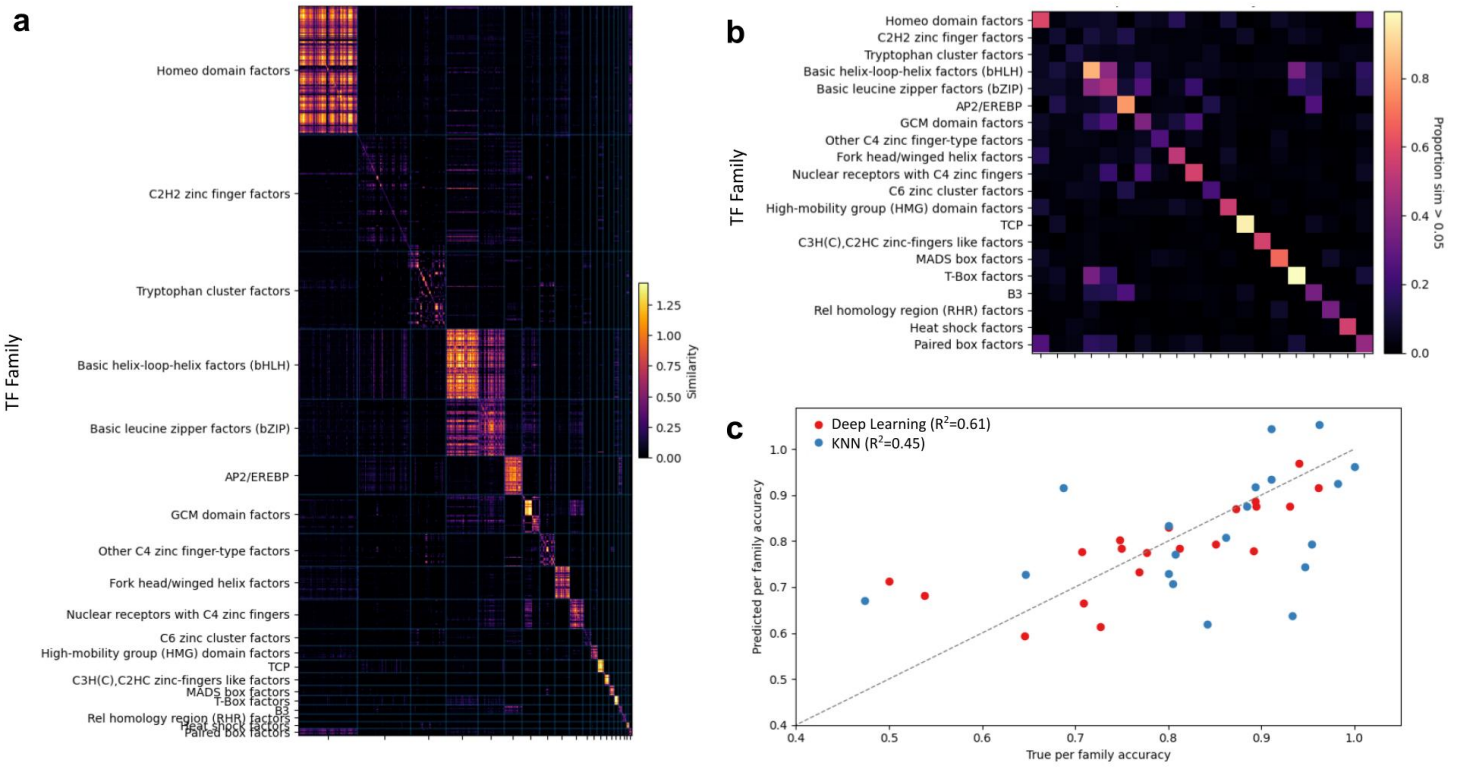

Figure 5 | Motif similarity analysis reveals inherent performance limit. a, Pairwise similarity matrix of all motifs in the dataset (n = 2113), sorted by family. Similarity scores were calculated by taking the negative logarithm of Tomtom's Euclidean distance q-values. To visualize the range of maximum variation, the 99th percentile of similarity was clipped. **b,** 20x20 pairwise family overlap matrix. Values are the proportion of comparisons between or within families that had similarity score > 0.05. Empirically, similarity scores < 0.05 were noise. **c,** Multiple linear regression of deep learning and KNN per-class accuracies, using family similarity metrics as predictors. Predictors were intra-family proportion, inter-family proportion mean, median, maximum, standard deviation, IQR, skewness, and kurtosis. Deep learning fit achieved R^2=0.61, and KNN fit achieved R^2=0.45, revealing family overlap metrics could explain considerable variation in classifier per-class performance.

Motif similarity analysis reveals inherent performance limit

The nearly identical overall accuracy and correlated per-family accuracies for KNN and deep learning suggested a potential shared ceiling on performance. We hypothesized this limit could be a result of sparse binding specificity within families, or significant overlap between families.

To test this, we computed pairwise motif similarities across the full dataset using Tomtom's Euclidean distance q-values. We transformed the statistical q-values into similarity scores by taking the negative logarithm (base 10) of q. Figure 5a displays the resulting 2113 x 2113 matrix, sorted by family, with the 99th percentile clipped to emphasize the most meaningful range of variation. Then, for each of the 20 TF families, we calculated the proportion of motif-motif comparisons exceeding a similarity threshold (sim > 0.05; lower values were noise) against every other family, producing a 20 x 20 family-overlap matrix (Fig.5b). Visual inspection of these similarity

matrices reveals that some families (bHLH, homeodomain) form tight diagonal blocks with high intra-family similarity, whereas others (C2H2 zinc fingers, tryptophan cluster factors) are more heterogenous, with lower intra-family similarity. Off-diagonal hotspots, such as between bHLH and bZIP, indicate significant inter-family overlap, which could make discrimination more difficult. These intra and inter-family similarity patterns demonstrate that some motifs are inherently difficult to predict, imposing a limit on achievable accuracy.

From each family's inter-family similarity distribution, we calculated summary statistics—mean, median, maximum, standard deviation, IQR, skewness, and kurtosis—and used these, along with intra-family similarity, as predictors in a multiple linear regression of per-family accuracy for KNN and deep learning (Fig.5c). The resulting deep learning fit achieved $R^2 = 0.61$, and KNN achieved $R^2 = 0.45$, indicating that family similarity metrics explain a substantial portion of variance in the per-class performance of both models. This finding, along with their correlated total and per-class accuracies, suggest that both classifiers may be approaching the theoretical upper bound of predicting TF family from binding specificity.

Discussion

Our study demonstrates that transcription factor structural families can be accurately predicted directly from binding specificity motifs. The k-nearest neighbors baseline achieved an overall accuracy of 0.842 and macro-F1 of 0.765 (KNN ensemble), while the deep learning approach achieved comparable performance (accuracy = 0.822, macro-F1 = 0.785) (10-fold CV). The distinct approaches demonstrated correlated per-family accuracies (Spearman $\rho = 0.40$; $p = 0.081$). Moreover, the deep model provided its predictions roughly one thousand times faster than the KNN model, highlighting its practical utility for predicting on large-scale binding data. Feature importance analysis via Integrated Gradients revealed that the deep model learns family-specific recognition codes. It assigned the highest importance to conserved positions of bHLH (CANNTG) and homeodomain (NTAATNN) motifs, validating its biological interpretability.

A comprehensive motif-similarity analysis supported the hypothesis of a shared performance ceiling for both classifiers. We quantified binding specificity sparsity within families and overlap between them, which indicated that some motifs are inherently difficult to predict and there is a theoretical upper bound on performance. Linear regression of per-family accuracies against family-overlap metrics explained up 61% of the variance for deep learning and 45% for KNN, indicating that family-overlap statistics alone can predict which TF families are most or least distinguishable. This suggests that our models are approaching the inherent performance limit that is set by the underlying binding specificity overlap between families.

Together, these results imply that binding specificity encodes substantial, but not unlimited, structural information, and machine learning methods can capture the available signal. Future work could address some of the limitations of this study, such as the inability to predict on dimers or rare TF families. The analyses could also be extended to predict finer-grained structural groupings, such as sub-families, or to further characterize the binding specificity landscape across the diversity of transcription factor proteins.

Methods

Data collection and pre-processing

We downloaded non-redundant motifs from JASPAR 2024[8], then filtered out dimers and families with less than 20 motifs, leaving 2113 motifs across 20 TF families. Each motif was encoded as a 4xN position weight matrix.

K-nearest neighbors classifier

We implemented a KNN classifier using three distance metrics—Pearson correlation, Euclidean distance, and Sandelin-Wasserman Similarity—from the Tomtom Motif Comparison Tool[22] on the MEME Suite[23]. We also implemented a "KNN ensemble" which used the majority vote of the three distance functions. We tested K from 1 to 11.

Neural network architecture

Our model is a fully convolutional network built for reverse complement invariance. Both the forward and reverse complement of the input motif are fed through the following layers: {[1D-conv (16 channels, kernel=3), 1D-conv (16 channels, kernel=5), 1D-conv (20 channels, kernel=7), 1D-conv (20 channels, kernel=1)]}. All layers used ReLU activations, with padding equal to [1, 2, 3, 0], respectively. After the convolutional layers, A global max pool operation reduces the 20xN feature map to a 20x1 vector. The two 20x1 vectors (forward and reverse complement) are averaged, then a softmax function is applied to obtain a 20x1 probability vector for the final prediction.

Neural network training and evaluation

We ran stratified K-fold cross validation (K=5, 10, 20), training each fold with early stopping after 7 epochs without validation loss improvement. We optimized with Adam (learning rate = 1×10^{-3}, weight decay = 1×10^{-3}) and batch size = 1. We used cross-entropy loss. To obtain evaluation metrics, we averaged the performances on held-out folds. To compare predictions, we measured how long for deep learning to complete (n = 2113) forward passes, and how long for the Tomtom webserver to compute (2113 x 2113) distance calculations.

Feature importance scores

We applied Integrated Gradients[25] to trace model outputs back to input positions. Scores were summed in motif columns and normalized across positions, to produce relative importance profiles.

Motif similarity and performance ceiling

We used Tomtom's Euclidean distance q-value as a similarity metric for pairwise comparison of all motifs. We converted them to $-\log_{10}(q)$ similarity scores and clipped at the 99th percentile for visualization. We quantified intra- and inter-family overlap as the proportion of motif pairs with similarity > 0.05.

Linear regression

We fit linear regression models of per-family accuracy using family-overlap metrics (intra-family proportion; inter-family mean, median, max, SD, IQR, skewness, kurtosis) and reported R^2 to assess how well family-overlap predicts performance across families.

References

1. Spitz, F. & Furlong, E. E. M. Transcription factors: from enhancer binding to developmental control. Nat. Rev. Genet. 13, 613–626 (2012).
2. Lambert, S. A. et al. The human transcription factors. Cell. 172, 650–665 (2018).
3. Zhao, Y., Granas, D. & Stormo, G. D. Inferring binding energies from selected binding sites. PLoS Comput. Biol. 5, e1000590 (2009).
4. Berger, M. F. & Bulyk, M. L. Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors. Nat. Protoc. 4, 393–411 (2009).

5. Slattery, M. et al. Cofactor binding evokes latent diferences in DNA binding specificity between Hox proteins. Cell 147, 1270–1282 (2011).
6. Jolma, A. et al. DNA-binding specificities of human transcription factors. Cell 152, 327–339 (2013).
7. Park, P. J. ChIP-seq: advantages and challenges of a maturing technology. Nat. Rev. Genet. 10, 669–680 (2009).
8. Rauluseviciute, I. et al. JASPAR 2024: 20th anniversary of the open-access database of transcription factor binding profiles. Nucleic Acids Res. 52, D174–D182 (2024).
9. Wingender, E. et al. TRANSFAC: an integrated system for gene expression regulation. Nucleic Acids Res. 28, 316–319 (2000).
10. Weirauch, M. et al. Determination and inference of eukaryotic transcription factor sequence specificity. Cell 158, 1431–1443 (2014).
11. Rohs, R. et al. The role of DNA shape in protein–DNA recognition. Nature 461, 1248–1253 (2009).
12. Garvie, C. W. & Wolberger, C. Recognition of specific DNA sequences. Mol. Cell 8, 937–946 (2001).
13. Wingender, E. et al. TFClass: a classification of human transcription factors and their rodent orthologs. Nucleic Acids Res. 43, D97–D102 (2015).
14. Rohs, R. et al. Origins of specificity in protein–DNA recognition. Annu. Rev. Biochem. 79, 233–269 (2010).
15. Kelley, D. R. et al. Sequential regulatory activity prediction across chromosomes with convolutional neural networks. Genome Res. 28, 739–750 (2018).
16. Fudenberg, G., Kelley, D. R. & Pollard, K. S. Predicting 3D genome folding from DNA sequence with Akita. Nat. Methods 17, 1111–1117 (2020).
17. Alharbi, W. S. & Rashid, M. A review of deep learning applications in human genomics using next-generation sequencing data. Hum. Genomics 16, 26 (2022).
18. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589 (2021).
19. Baek, M., McHugh, R., Anishchenko, I. *et al.* Accurate prediction of protein–nucleic acid complexes using RoseTTAFoldNA. *Nat Methods* **21**, 117–121 (2024).
20. Li, J., Chiu, TP. & Rohs, R. Predicting DNA structure using a deep learning method. Nat Commun 15, 1243 (2024).
21. Mitra, R. et al. Geometric deep learning of protein–DNA binding specificity. Nat Methods 21, 1674–1683 (2024).
22. Gupta, S. et al. Quantifying similarity between motifs. Genome Biol 8 (2007).
23. Bailey, T. L. et al. The MEME suite. Nucleic Acids Res. 43, W39–W49 (2015).
24. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1, 206–215 (2019).
25. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. Proc. 34th Int. Conf. Mach. Learn. 70, 3319–3328 (2017).
26. Michael, A.K. et al. Cooperation between bHLH transcription factors and histones for DNA access. Nature 619, 385–393 (2023).
27. Chi, Y. I. Homeodomain revisited: a lesson from disease-causing mutations. Hum. Genet. 116, 433–444 (2005).