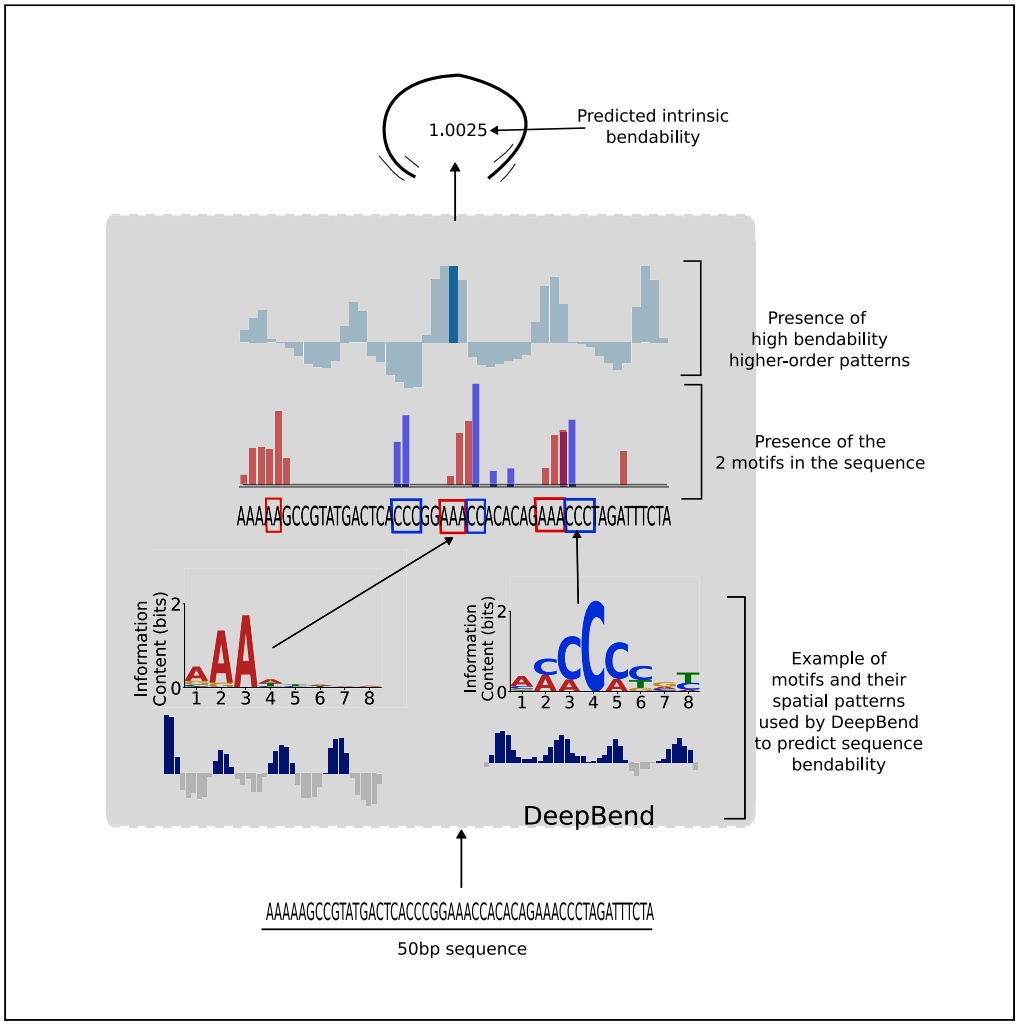
iScience

CellPress
OPEN ACCESS

Article

DeepBend: An interpretable model of DNA bendability



Samin Rahman Khan, Sadman Sakib, M. Sohel Rahman, Md. Abul Hassan Samee

msrahman@cse.buet.ac.bd (M.S.R.)
samee@bcm.edu (M.A.H.S.)

Highlights

DeepBend provides the motifs and higher-order patterns responsible for DNA bendability

Novel motifs produced by the model include the highly bendable GAAGAGC 7-mer

Motifs containing poly (dA:dT) highly influence the bendability at TAD boundaries

Khan et al., iScience 26, 105945
February 17, 2023
https://doi.org/10.1016/j.isci.2023.105945

iScience

CellPress
OPEN ACCESS

Article

DeepBend: An interpretable model of DNA bendability

Samin Rahman Khan,[1] Sadman Sakib,[1] M. Sohel Rahman,[1,*] and Md. Abul Hassan Samee[2,3,*]

SUMMARY

The bendability of genomic DNA impacts chromatin packaging and protein-DNA binding. However, we do not have a comprehensive understanding of the motifs influencing DNA bendability. Recent high-throughput technologies such as Loop-Seq offer an opportunity to address this gap but the lack of accurate and interpretable machine learning models still remains. Here we introduce DeepBend, a convolutional neural network model with convolutions designed to directly capture the motifs underlying DNA bendability and their periodic occurrences or relative arrangements that modulate bendability. DeepBend consistently performs on par with alternative models while giving an extra edge through mechanistic interpretations. Besides confirming the known motifs of DNA bendability, DeepBend also revealed several novel motifs and showed how the spatial patterns of motif occurrences influence bendability. DeepBend's genome-wide prediction of bendability further showed how bendability is linked to chromatin conformation and revealed the motifs controlling the bendability of topologically associated domains and their boundaries.

INTRODUCTION

Bendability is a critical mechanical property of genomic DNA with implications for its structure,[1] chromosomal packaging,[2] and interactions with DNA-binding molecules.[3] However, sequence signatures underlying DNA bendability are poorly understood at best. Advances in machine learning and Loop-Seq,[4] a SELEX-based assay of DNA bendability, offer an opportunity to address this gap. The recent Loop-Seq dataset, for example, quantified bendability across the entire yeast genome at 50 bps resolution (nearly 200,000 sequences). Loop-Seq starts with an initial library of PCR-amplified sequences. Each sequence has two flanking single-stranded overhangs that should attach to each other and form a loop if the sequence is naturally bendable. After leaving the library in a chemical solution for a specified time, the unlooped sequences are digested and the looped sequences are amplified for a second time. The experiment is then repeated using an identical library but omitting the digestion step. Finally, the bendability of each sequence is quantified as the logarithm of the ratio of its relative abundance in the two libraries.[4]

The first models of Loop-Seq data have been built from handcrafted features, such as, the frequencies and periodicities of dinucleotides[5] and AT- or GC-rich sequences up to 6 bps in length.[6] These features are based on DNA biophysics and are easy to interpret, but the models can predict bendability with a Pearson's correlation r of ~60% with the true value, leaving approximately ~60% ($1-0.6^2 = 0.64$) of the variance in data unexplained. Furthermore, it remains unclear whether sequence patterns beyond these conventionally known dinucleotides and AT-/GC-tracts are important for DNA bendability.

To improve the state-of-the-art models' performance in predicting bendability and discover the relevant sequence features *de novo*, here we introduce a convolutional neural network (CNN), DeepBend. Importantly, CNNs are known to suffer from a lack of interpretability.[7] To alleviate this issue, we designed DeepBend as a *visible neural network*[8] with mechanistically grounded kernels that directly reveal: (a) the sequence patterns, also known as *motifs*, underlying DNA bendability and (b) how periodic occurrences or relative arrangements of motifs influence bendability. Li et al.'s DNACycP model,[9] which is a hybrid of CNNs and RNNs, for modeling Loop-Seq data have recently shown that neural networks can model bendability data with high accuracy. DeepBend's novel architectural choices aim to push this progress further by accomplishing both accuracy and interpretability. Both DeepBend and DNACycP are similar in design principles, where they first capture patterns using the primary layers and then use secondary layers to detect

[1]Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

[2]Baylor College of Medicine, Houston, TX, USA

[3]Lead contact

*Correspondence: msrahman@cse.buet.ac.bd (M.S.R.), samee@bcm.edu (M.A.H.S.)

https://doi.org/10.1016/j.isci.2023.105945

long-range spatial relationships between these patterns. The DNACycP model uses Inception-Resnet convolution layers as their primary layers and an LSTM layer as a part of their secondary layers. As such, DNACycP, like the alternative models we have benchmarked DeepBend against, is difficult to interpret without post hoc analysis. It is not easy to use post hoc analysis for identifying motifs and their spatial patterns. Post hoc model interpretation techniques, like importance or relevance backpropagation, can identify motifs that the model deems significant. Unfortunately, deriving these motifs involve choosing thresholds for several parameters, but systematic approaches for these steps are still unclear. Furthermore, the efficacy of these discovered motifs in modeling the data has not been rigorously assessed. Finally, automatically detecting spatial patterns of the post hoc motifs is still a challenge. In contrast, we built DeepBend in the spirit of *model-based interpretation*,[10] which is based on the principle of designing individual components of a model to reflect domain knowledge. DeepBend uses a multinomial convolution layer to discover directly interpretable motifs and wide convolutions in the secondary layers that directly reveal spatial patterns influencing bendability. In general, model-based interpretation comes at the expense of lower performance than alternative "black-box" models.[10] Thus, we extensively benchmarked DeepBend on Loop-Seq data against alternative machine learning models, such as DNACycP, support vector machines and random forests, and deep neural network architectures, such as CNNs and their hybrids with recurrent neural networks (RNNs).

DeepBend consistently showed the superior predictive performance to simpler explainable models. It also showed performance on par with the state-of-the-art DNACycP model, while retaining an extra edge in interpretability. Applying DeepBend on Loop-Seq datasets revealed both known and novel motifs and their relative arrangements important for bendability. The model also revealed the 7-mer GAAGAGC as a novel motif and its significant role in determining bendability beyond the conventionally known dinucleotides and AT-/GC-tracts. The model also shows that the relative arrangements of these motifs are important in determining the bendability of sequences. Finally, DeepBend revealed how sequence motifs influence chromatin conformation through DNA bendability.

RESULTS

DeepBend: A deep convolutional neural network model of DNA bendability

DeepBend is a 3-layered CNN that takes in a one-hot encoded DNA sequence as input and predicts its bendability as output (Figure 1). The first two layers are applied parallelly to both the forward and the reverse complement strand of the input sequence, allowing the model to detect patterns in both DNA strands. The first layer is a multinomial convolution layer with ReLU activation.[11] Filters in this layer detect motifs and the convolution operation computes matching scores of the motifs (log likelihood ratios, see Methods) at each position of the input sequence. The second convolutional layer learns spatial patterns in the motif-matching scores. Up to this layer, all operations are applied to both the forward and reverse complement strands, thus producing two matrices. The element-wise maximum of the two outputs is passed to the next layer through a ReLU activation. This produces an output matrix that has matching scores of different spatial patterns of motif occurrences at different positions from both the forward and the reverse complement strands. In a post-processing step, we identify the motifs with periodicity in their spatial patterns. The third layer is a single-filtered convolutional layer with linear activation. It takes in all the matching scores from the previous layer as input and outputs the predicted bendability of the sequence.

DeepBend models loop-seq data with high accuracy and interpretability

We applied DeepBend to model four Loop-Seq libraries,[4] each quantifying the bendability of 50 bps long sequences: (i) The *Random library* has 12,472 randomly generated DNA sequences, (ii) the *ChrV library* comprises 82,404 sequences from yeast (*Saccharomyces cerevisiae*) chromosome V (tiled across the chromosome with 7 bps shift), (iii) the *Nucleosomal library* contains 19,907 sequences centered at high nucleosome occupancy locations in the yeast genome, and (iv) the *Tiling library* consists of 82,368 sequences (tiled across the chromosome with 7 bps shift) from 2001 bp regions centered around +1 nucleosome around 576 selected genes. Importantly, in the ChrV and Tiling libraries, each sequence overlaps with seven upstream and seven downstream sequences. For a fair and objective evaluation of the model performance, it is necessary to make sure that sequences in the test and the training sets do not overlap as this could otherwise inflate performance. Thus, while benchmarking on these two libraries, we have fit all models on specially designed datasets (see STAR Methods for details). The training (test) sets obtained from this separation will be referred to as the *ChrV training (test) set* and *Tiling training (test) set*.

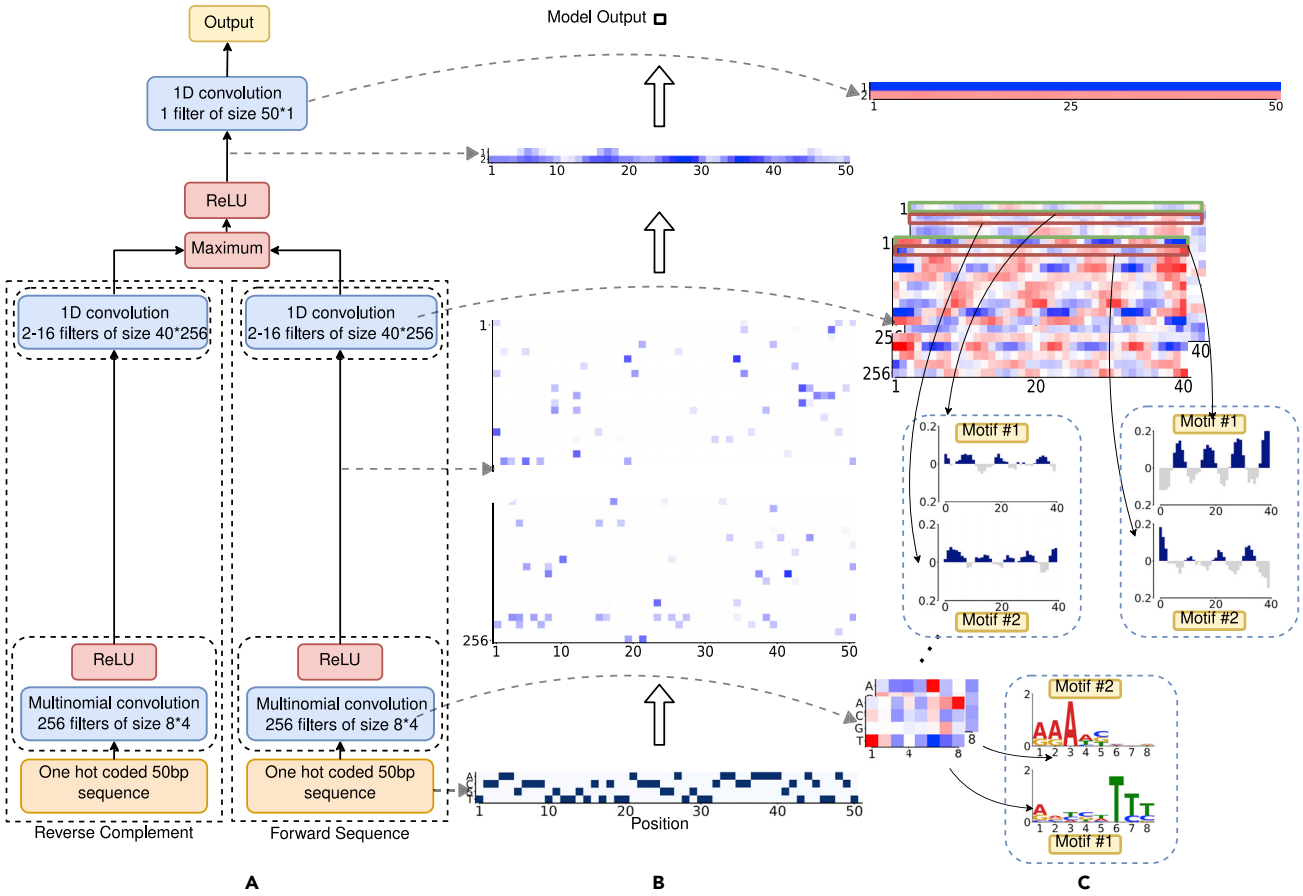


Figure 1. DeepBend model architecture and the inputs and outputs to different layers along with the convolution layers

(A–C) One hot-encoded forward and reverse complement sequences are taken as input into the model. The first multinomial layer acts as a motif detector. After the first convolution and ReLU operation, we get a matrix containing the matching scores of the motifs at each position for both the forward and reverse complement sequences. The next convolution layer is designed to detect spatial patterns of motif occurrences and produces an output matrix for each sequence which contains the matching scores of these patterns at each position. The element-wise maximum of the results from the second layer is taken and then after a ReLU operation, we get as output the most prominent matches of bendability patterns at each position from both the sequences (forward and reverse complement). This is fed into the final convolution layer which gives the output bendability value. The first convolution layer provides motif patterns using transformation discussed in the Methods section, and their spatial patterns are obtained from the second convolution layer. The extraction of motifs and their patterns are shown in (C).

For comparing DeepBend with Basu et al.'s model,[5] we trained DeepBend on the Tiling library and tested it on the Random, Nucleosomal, and ChrV libraries. The Pearson's correlation coefficient (r) between the true bendability of these three libraries and DeepBend's predictions were 0.895, 0.931, and 0.774, respectively. Compared to Basu et al.'s models, the improvements were 59.82% in the Random Library, 55.16% in the Nucleosomal Library, and 29% in the ChrV library. The performance comparison between DeepBend and Basu et al.'s models[5] has been shown in Figure 2A. DeepBend also outperformed other machine learning models in terms of r: SVMs (by 118.29%), Random Forests (by 208.62%), and a series of deep neural network models (by 5.29-118.29%) (see STAR Methods, Tables S1 and S2). All these models were trained on the Tiling library and tested on the Random library. We chose the Tiling library for training since it provides a comprehensive set of sequences spanning a whole chromosome, while the Random library was chosen for testing since the sequences are randomly generated and are equally likely to contain any 50 bps sequence.

Although DeepBend outperformed RNN models in our benchmarking (Table S2), we still performed a rigorous comparison between DeepBend and DNACycP as follows. For the Random and Nucleosomal libraries, we compared the models using 10-fold cross-validation. For the ChrV and Tiling library, we trained

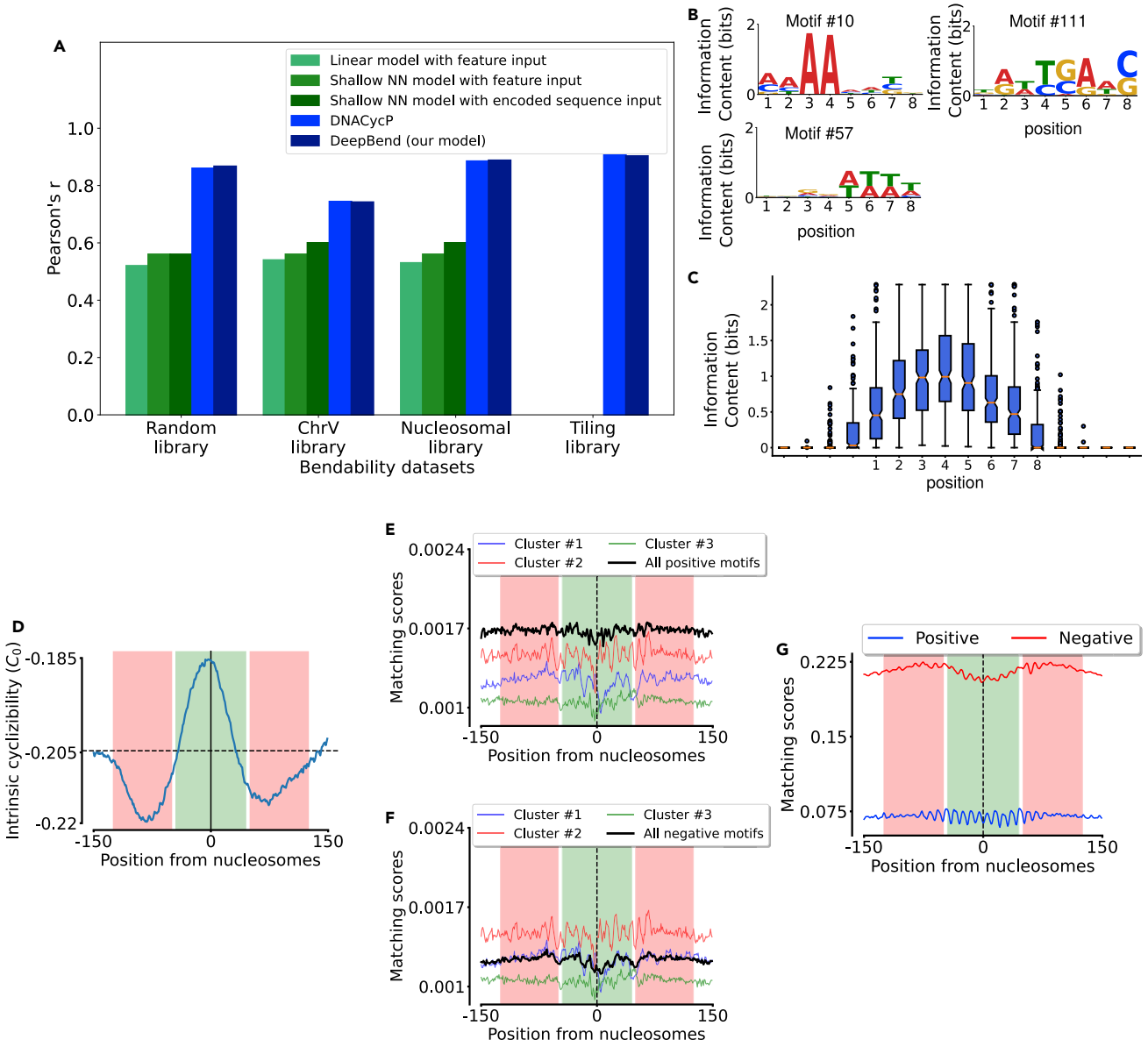


Figure 2. Model performance, motifs, and their presence around nucleosomal regions

(A) Comparison of model performance. Pearson's correlation coefficient (r) between true and predicted results from different models. For DNACycP[9] and DeepBend, 10-fold cross-validation results are shown for Random and Nucleosomal libraries. For ChrV and Tiling libraries, the results of the *ChrV test set* and *Tiling test set* are respectively shown. For the other three models, we have shown Basu et al.'s[5] results of testing their model on the Random, ChrV, and Nucleosomal libraries. Since they trained models on the Tiling library, we could not show their performance considering the Tiling library as test data.

(B) Some motifs from DeepBend. Motif #10 and #57 are previously known bendability motifs and motif #111 is a novel motif.

(C) Information content of motifs from DeepBend. Distribution of information content of the motifs at each position, showing that motifs are small in size, of length 3-5. The motifs were centered.

(D–G) Presence of motifs around nucleosomal regions. (D) Average bendability around nucleosomes (the green region from ± 45 is more bendable compared to ± 50 to ± 125 region) and (E). the strong/weak matching scores (see STAR Methods) from the first layer of the 79 bendability causing motifs can be clustered into three patterns. The averages of these 3 patterns and all the motifs together are shown. (F) A similar plot done for the 177 rigidity-causing motifs. (G) The average second layer matching scores for positive bendability and negative bendability for all 256 motifs.

and tested the models on the specially designed datasets noted above ensuring that sequences in the training and the test sets do not overlap in the yeast genome. In all these comparisons, DNACycP and DeepBend showed nearly identical performance (Table 1). The performance comparison of DeepBend and DNACycP is also shown in Figure 2A. The Pearson's correlation coefficient (r) between the true

Table 1. Performance comparison between DeepBend and DNACycP

Test libraries	DeepBend	DNACycP
Random Library[a]	0.87	0.86*
Nucleosomal Library[a]	0.89	0.89*
ChrV Test Set[b]	0.74	0.74
Tiling Test Set[b]	0.90	0.91

Shown are the Pearson's correlation coefficients (r) between true and predicted values.
*Results from original paper.[9]
[a]Results from testing during 10-fold cross-validation.
[b]Results from testing on the specially designed test set.

bendability and DeepBend's predictions for models trained and tested on different libraries are provided in Table S3.

Interpreting the DeepBend model is straightforward. Since each row of a first layer filter is a multinomial distribution over the four nucleotides, these filters are directly interpretable as biophysical models of sequence motifs.[11] Regularising variance in the last layer separates out the relative spatial patterns of motifs significant for different ranges of bendability into different filters of the second convolution layer. These patterns can be easily identified from the weights of the second layer filters. Further details on the model architecture and the rationale thereof have been presented in the STAR Methods section.

DeepBend confirms known motifs and discovers new motifs influencing bendability

Visualizing DeepBend's first layer kernels we found that the model was able to learn the motifs that are conventionally known to influence bendability.[5,6,12–14] Interestingly, DeepBend also found some new motifs and predicted a significant role for them in determining bendability. We ranked DeepBend's motifs according to their contribution to positive and negative bendability, as we quantified the change in model's prediction after deactivating one motif at a time (see STAR Methods). From all the DeepBend models we have trained, for interpretation purposes, we have selected the model that has been trained on the ChrV library and has a reduced number of filters (two) in the second convolution layer. All the motifs from this model along with their patterns have been provided in Note S3. Among the known motifs are the A/T dinucleotides (Motif #3, #10, #178, #232), A/T regions (Motif #8, 57, 150, 166, 213, 250), C/G regions (Motif #34, #24) which contribute to bendability positive or negatively depending on their relative arrangements[5,6,12–14] (discussed later in discussion). Although the visually discernable "core" signals from these motifs are similar, they differ in the flanking regions around the core signals. Likewise, Poly A tracts (Motif #11, #51, #117),[15] CG (Motif #28) are negatively contributing to bendability in general. Among the novel motifs, there are motifs such as GAAGAGC (Motif #40) and Motif #47 which contribute positively, and motifs such as Motif #78, Motif #111, Motif #77, and Motif #99 contribute negatively. The most prominent is motif #40, heptamer GAAGAGC, which is the most informative motif found by our model and is strongly indicative of positive bendability. Some examples are shown in Figure 2B.

Expectedly, most of the specific motifs are small, usually dinucleotides, as the bendability of the sequence is fundamentally determined by the bendability between adjacent bases. From the distribution of information content at each position of the motifs after centering them, we find that the average information content is much less than the maximum value and that most motifs are small in size, of length 3-5 (Figure 2C). So, in effect, DeepBend is suggesting that bendability of a sequence is principally determined by short motifs and arrangements thereof.

As has been shown in,[4] nucleosome regions have a peak in bendability near the dyad (Figure 2D). This definite pattern occurs due to the presence of bendability motifs around these regions. In nucleosomes, generally, A/T dinucleotides occur at a 10 bp period in antiphase with G/C dinucleotides. Most of the motifs identified by DeepBend are distributed at different distances from the center of the nucleosome. These motif distributions around nucleosomes conform to prior knowledge.[16,17] Motifs having dA:dT (motif #49, motif #50) are found less in the nucleosomal region. Motifs have more consistent periodic arrangements near the central nucleosomal regions ($\pm$45) as compared to the surrounding region ($\pm$50 to $\pm$125). This can be seen from the increased rise and fall in the average of the motif-matching scores of bendable motifs

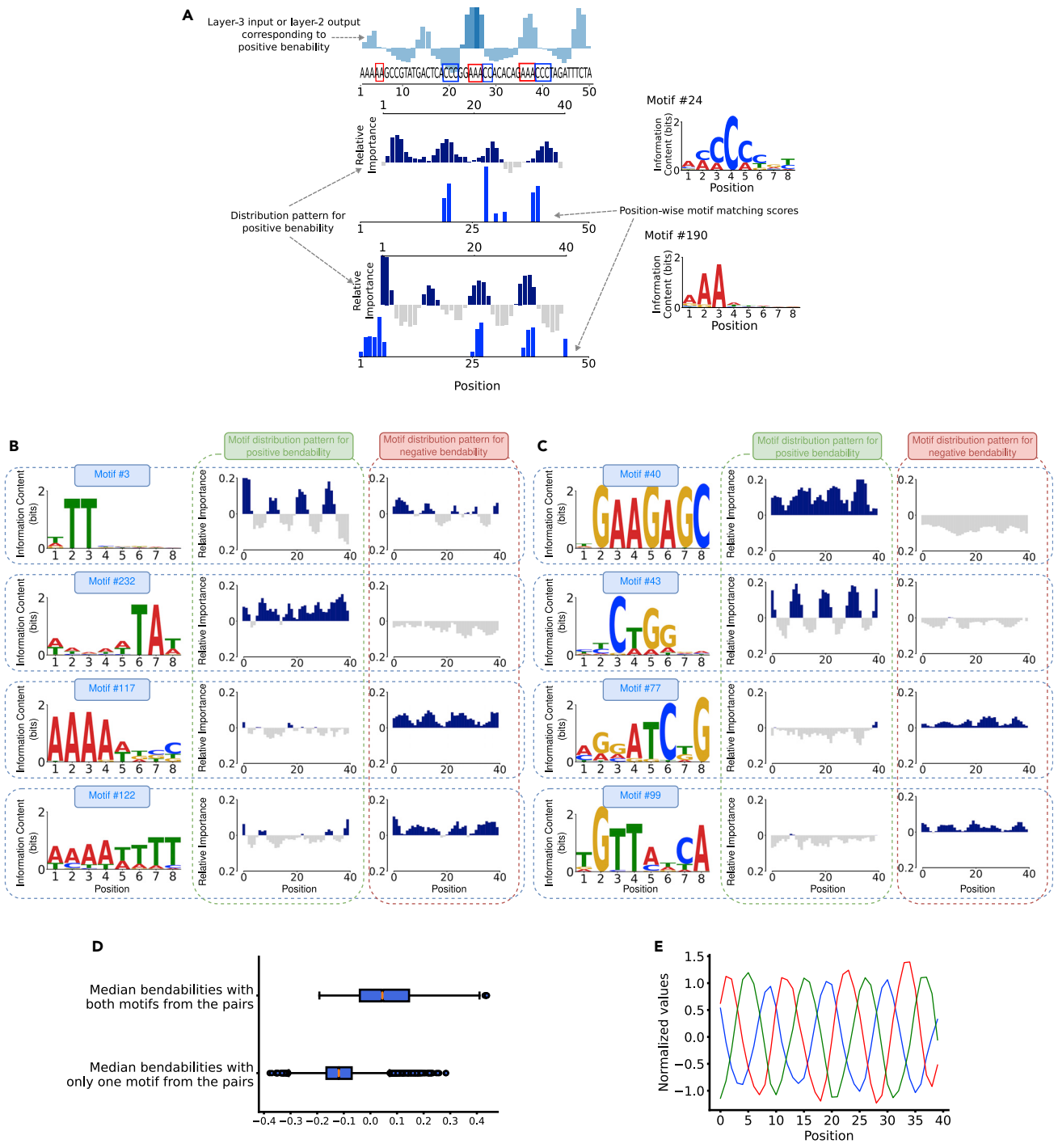


Figure 3. Model detecting spatial patterns

(A) ChrV:33,804-33853 is a highly bendable sequence. The input corresponding to the row with positive weights in the third layer shows the window positions where the second convolution layer has detected patterns of high bendability. When the convolution overlaps at a position where there is a good match between the spatial patterns and the matching scores of the motifs, as shown in the figure there is a high output from the second layer at that position for that filter. The position highlighted in darker blue shows a high output due to the presence of positively contributing distributions of motifs in this region at close proximity. Motif #24 and motif #190 are distributed periodically, which is an indicator of positive bendability matching the first-order patterns of the filter. And also these distributions are interleaved at the same region making that region more bendable, matching the higher-order relative arrangement of the motifs as well, which further intensifies the signal for positive bendability.

Figure 3. *Continued*

(B and C) Shows some motifs found by our model and their relative distribution for positive and negative bendability. (B) DeepBend has confirmed that simple motifs such as TT(#3) and TA(#232) contribute positively when they are at periodic helical length distances of 10 bases. Periodic TAs (#232) also contribute positively throughout the sequence. Long A(#117), T and dA:dT(#122) contributes negatively. (C) In addition to more traditional motifs, DeepBend is able to capture more elaborate motifs such as #40, #43, #77, and #99. All motifs and their spatial patterns are provided in Note S3 (Figures S2–S33).

(D and E) Higher-order relation between first-order spatial patterns of motifs. (D) Top: median bendabilities of sequences (from Random Library) with the highest positive contributions from the two motifs together for every positively contributing motif pair. Bottom: median bendabilities of sequences with the highest positive contribution from only the first motif and with no presence of the other motif in the same region for every positively contributing motif pair. This shows that there tends to be an additive effect when more motifs contribute positively in a region (see STAR Methods). (E) The normalized first-order patterns for high bendability can be clustered into periodic patterns of ~10bp. These first-order patterns from different clusters are interleaved with one other.

near the central nucleosomal region, Figures 2E and 2F. Less bendable motifs have a reduced presence in the central nucleosomal region, Figure 2F. Such arrangement of motifs leads to a periodic presence of positive bendability patterns and a general dip in negative bendability patterns in that region as can be seen from the second layer matching scores, Figure 2G.

DeepBend reveals spatial and periodic patterns of motif occurrence

DeepBend can learn how the motifs act throughout the sequence without requiring any prior information. DeepBend captures important relative spatial patterns of small motifs in sequence segments rather than fixed features such as k-gapped dinucleotides or k-mer counts.

The second layer filters capture spatial biases or periodic patterns in motif occurrences. Each of the rows of the filter corresponds to the relative spatial pattern of a motif from the first layer. We call each such pattern a first-order pattern. The periodic presence of a motif is an example of a first-order pattern. Due to the property of convolution, we can know how the motifs should be distributed for a particular pattern to be present from the weights of the filters of this convolution layer. Figure 3 presents a pictorial description of how DeepBend detects spatial patterns. Here, the filter row or kernel corresponding to a motif from the previous layer for positive bendability is shown. The regions where the motifs should be (relative to each other) have positive values and are shown in blue; the regions where the motifs should not be (relative to each other) have negative values and are shown in gray (Figures 3A-3C), and the magnitudes of the weights provide the significance of the presence or absence of the motif at that position.

If there is a spatial bias or periodicity between the occurrences of several first-layer filters, the filters in the second layer can also learn that. Figure 3A shows that the periodic patterns of motifs #24 and #192 are interleaved with one another with an offset. Such an interleaving periodic presence of the two motifs increases the bendability of the region. We refer to such composites of first-order patterns as higher-order patterns. Each of the filters from the second layer is actually a higher-order pattern. Thus, the number of filters in the second layer determines the number of different higher-order patterns the model can learn. By hyperparameter tuning, we have noted that two filters in the second layer are sufficient. This is because the first-order patterns of high and low bendability can co-exist separately in their respective single filter. We have seen only two main types of first-order relations: (1) in-phase and out-of-phase periodic patterns and (2) patterns that remain consistently high or low throughout a sequence. To make each of these filters in the second layer capture patterns that are representative of a particular bendability throughout the sequence, we have added position-wide variance regularisation in the third layer corresponding to each filter in the previous layer. This allows us to easily separate out the patterns for high and low bendability.

Instead of capturing long motif patterns, the model is able to learn spatial patterns through which it can find larger patterns with smaller motifs. Because of this, it is more expressive than models that take fixed gapped motifs as features. The periodic patterned distributions from the second layer, seen more often for positive bendability patterns, are useful for finding the presence of gapped motifs. An example is shown in Figure 3A. The pair of motifs #24 and #190 occurs at a period of ~10bp at the center of the example sequence. These periodic patterns are captured by the second layer convolution when the window passes over this region, producing a high output there.

The motifs and their first-order patterns learned by our model can be summarised in a table like that shown in Figure 3B and 3C. This model has been trained on the ChrV Library and has 2 filters in the second

convolutional layer. The motifs and their relative arrangements that are important for positive and negative bendability are shown here. The motifs on the left are some confirmatory motifs that show that from motif #3, the model learns that periodic occurrence of TT dinucleotide at a helical distance of 10bp can make a sequence more bendable, which was also reported by.[5] Periodicity of T/A dinucleotides at ~10bp is also considered as a preferred sequence for nucleosome positioning[16] and this high bendability explains how these sequences loop around nucleosomes. Motif #232 from our model tells us that the presence of a short sequence of alternating Ts and As makes sequences more bendable regardless of their distribution.[5] also reported TA dinucleotide to have the highest correlation with positive bendability among all dinucleotides. Periodic presence of A/T regions and C/G regions makes sequences more bendable.[5,6,14] The first-order periodic patterns for high bendability for motif #0, #8, #23, #34, #150, #183, and #229 from our model also show this. Observing the higher-order relations between these regions shows that an interleaving periodic presence of these A/T and C/G regions makes the sequence even more bendable. Interestingly, the same motif can contribute both positively and negatively depending on how they are arranged. For example, Motif #196 contributes positively when periodically arranged but contributes negatively when arranged more consistently.

In addition to the confirmatory motifs discussed above, DeepBend has also found some novel motifs. The distribution of motif #43 reveals that the CTGG sequence is positively contributing when positioned at helical distances. In general, positively influencing motifs mostly act at periodicities of the helical length of DNA and are small and specific. Negatively influencing motifs do not show these periodic patterns and are mostly non-specific/diverse.

We found the regions of motif presence using FIMO[18] (see STAR Methods). The 8 bp regions where our motifs are present, have a higher tendency of being in the conserved regions of the yeast genome (hypergeometric p value $<10^{-18}$), suggesting that the bendability-related motifs of the entire yeast genome are well-represented in the conserved regions and models such as DeepBend should be able to capture such motifs.

When different motifs contribute positively together it tends to increase the bendability of sequences. By comparing the bendabilities of sequences where the first-order pattern of only one motif from a pair is present versus when the first-order pattern from both of the motifs is present. We have found an increase in the median of bendabilities when the pair occur together (see Figure 3D and details in STAR Methods). We found that the first-order patterns for higher bendability for positively contributing motifs cluster into three interleaving periodic groups, Figure 3E. Such higher-order patterns of motifs reveal how these motifs are arranged relative to one another. The positive motifs occur in interleaving periodic patterns in order to increase the bendability of the sequences. So, we see that from the second layer of DeepBend, it is possible to easily determine both significant first-order and high-order relative arrangement of motifs.

DeepBend revealed the novel GAAGAGC motif and its strong role in determining bendability

As noted above, DeepBend has learned several novel motifs beyond the conventionally known poly-A or dA:dT patterns. Of these, the GAAGAGC motif (motif #40) (Figure 3C) is particularly interesting. The 7-mer GAAGAGC is present in many viral genomes, sometimes highly conserved within a hyper-variable region and potentially has a role in RNA folding and viral pathogenicity.[19,20] However, this motif's connection with DNA mechanical properties like bendability has never been discussed.

To substantiate the GAAGAGC motif's role irrespective of our model, we checked the *bendability quotient* of the 7-mer GAAGAGC and its reverse complement, GCTCTTC. Basu et al. defined and used this statistic to denote the frequency of a k-mer's occurrence in sequences with high bendability compared to those with low bendability[5] (see STAR Methods). Importantly, GAAGAGC and GCTCTTC showed the highest bendability quotient of all 7-mers (Figure 4A), indicating a clear role for this motif in making the DNA more bendable. Furthermore, GAAGAGC is much more bendable than the 7-mers that differ from it by even a single nucleotide. The bendability values of ChrV sequences containing GAAGAGC are significantly more positive than those containing 7-mers mismatching GAAGAGC (Figure 4B). Interestingly, these shape profiles (see STAR Methods) readily revealed some consistent patterns of DNA structural properties at the GAAGAGC motif. As we showed now in Figure 4E, the HelT values follow a periodic pattern of high and low values at the central five positions. Similarly, these sequences have a low MGW and high ProT at the central positions.

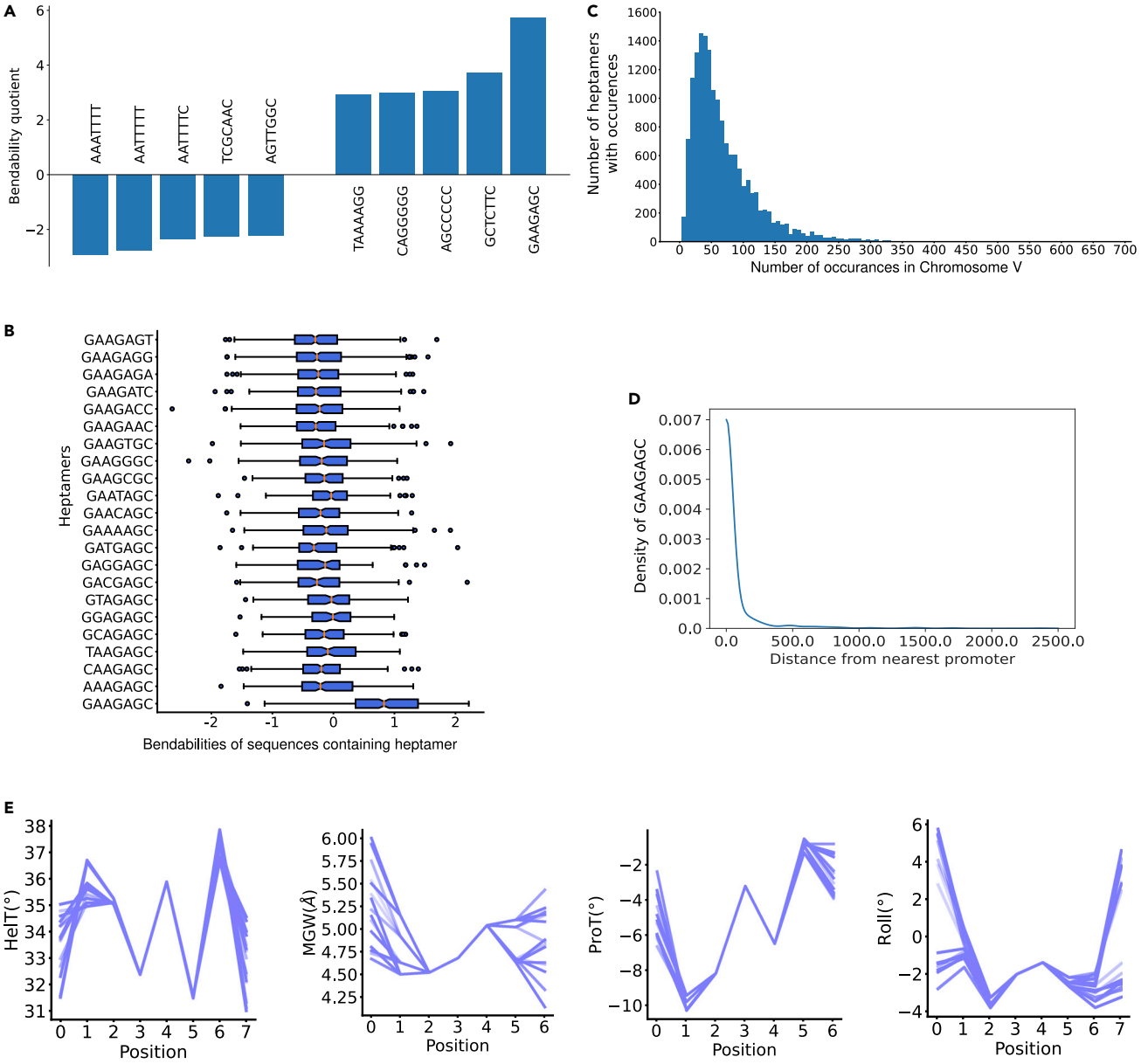


Figure 4. GAAGAGC motif

(A) Bendability quotient of heptamers with highest 5 and lowest 5 values.

(B) Boxplot showing the distribution of the bendability scores of the sequences from the ChrV library containing GAAGAGC and its neighbors (which differ from GAAGAGC by only 1bp).

(C) Shows the distribution of heptamers occurring at different counts in ChrV.

(D) The density plot of the distance of GAAGAGC 7-mers from their nearest promoters. 778 of the 960 occurrences are found within promoter regions (hypergeometric p value $< 10^{-9}$).

(E) Shape features (Helical twist, Minor Groove Width, Propeller Twist, Roll) of GAAGAGC 7-mer found using DNAShapeR.[21]

Besides, GAAGAGC and GCTCTTC occur 120 times in ChrV, which makes them among the top 14% 7-mers found in ChrV (Figure 4C). Given such frequent occurrence of the GAAGAGC motif and its potentially strong connection to bendability, we asked whether this motif plays a role in biologically important regions. We searched for GAAGAGC and each possible substring of GAAGAGC in PDB (we generated the substrings by using 'X' in one position at a time). Although We could not find the exact sequence GAAGAGC, we found occurrences of the following five substrings, XAAGAGC, GXAGAGC, GAAXAGC, GAAGXGC, and GAAGAGX, in 24 structures. 16 of these 24 structures correspond to "RNA polymerase

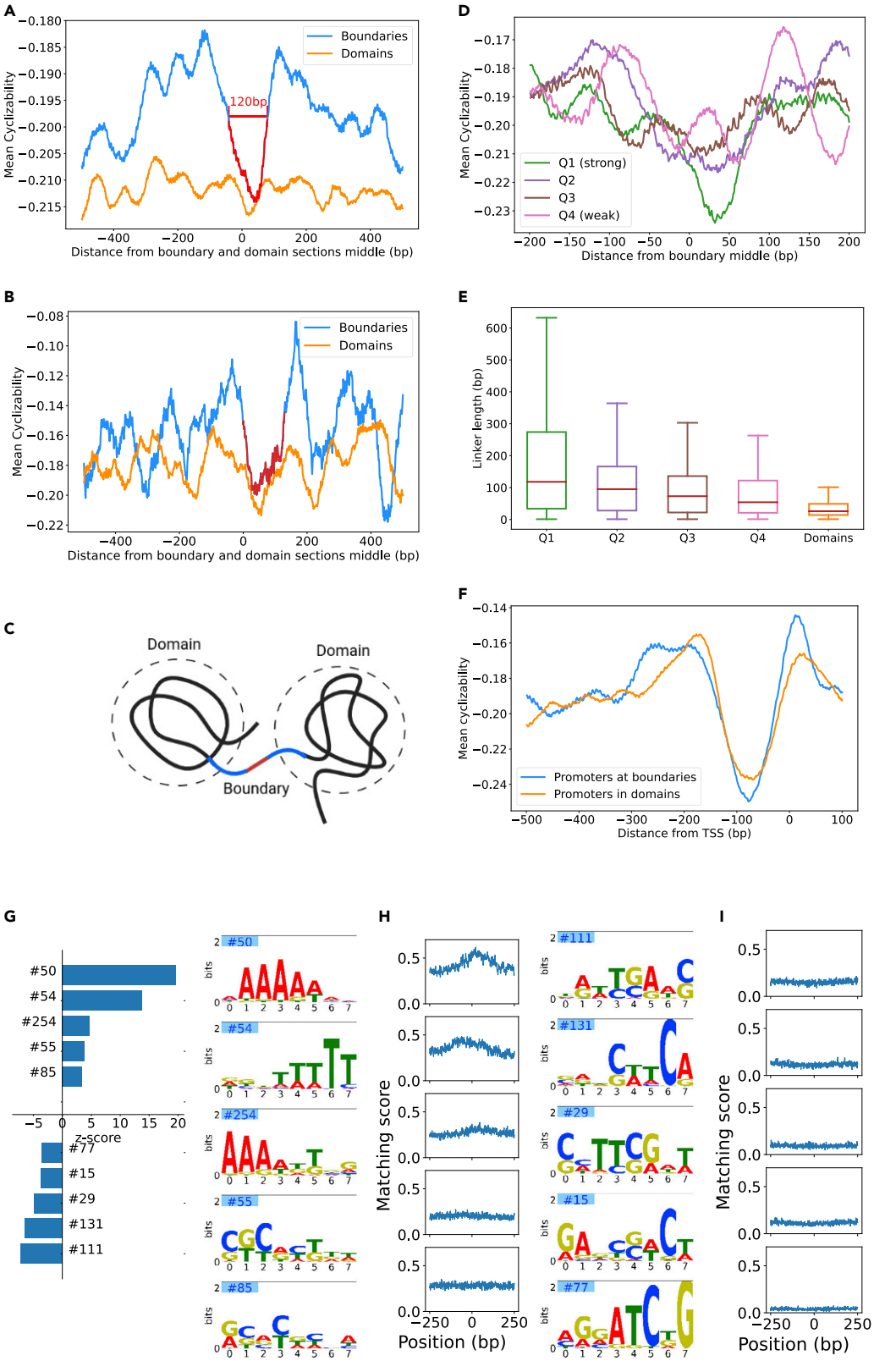

A
Mean Cyclizability
Boundaries
Domains
120bp
Distance from boundary and domain sections middle (bp)
B
Mean Cyclizability
Boundaries
Domains
Distance from boundary and domain sections middle (bp)
C
Domain
Domain
Boundary
D
Mean Cyclizability
Q1 (strong)
Q2
Q3
Q4 (weak)
Distance from boundary middle (bp)
E
Linker length (bp)
Q1
Q2
Q3
Q4
Domains
F
Mean cyclizability
Promoters at boundaries
Promoters in domains
Distance from TSS (bp)
G
#50
#54
#254
#55
#85
z-score
#77
#15
#29
#131
#111
bits
H
Matching score
Position (bp)
I
Matching score
Position (bp)

Figure 5. Stronger boundaries are more rigid

(A) Mean bendability of 1000 bp regions at boundaries and 1000 bp sections of self-interacting domains of all 16 chromosomes where bendability is predicted by DeepBend.

(B) Mean bendability of 1000 bp regions at boundaries and 1000 bp sections of self-interacting domains in chromosome V where bendability data is obtained from Chromosome V library.

(C) Formation of boundaries in yeast. A stiff region (red) forms the main part of the boundaries which is flanked by two highly bendable regions (blue).

(D) Boundaries in all of the 16 chromosomes in yeast are split into quartiles (Q1-Q4) based on their TAD separation score. Then, the bendability of all boundary regions in a quartile was averaged.

(E) Distribution of lengths of nearest linkers from boundary middle positions of quartile boundaries and rest of the linkers that are considered in domains. Boundaries from all 16 chromosomes were considered.

(F). Promoters at boundaries contain rigid NDRs flanked by highly bendable regions compared to promoters at domains. Regions between 500 bp upstream and 100 bp downstream of the transcription start site were considered promoter regions. Bendability of promoter regions from all 16 chromosomes was averaged. A promoter was considered at a boundary if the middle of any boundary falls in this promoter.

(G) *Z* score of some motifs that influence bendability more in boundaries compared to domains (top) and in domains compared to boundaries (bottom). Table S6 lists the z-scores for all the motifs.

(H and I) Motif logos and mean matching score of motifs with the most positive (H) and most negative (I) *Z* score in G. These are similar to motifs found by using streme[26] in boundary and domain regions (see Note S1 and Figure S1).

II elongation complex," "Protein bound to termination sequence," "RNA polymerase bound to promoter DNA," and "transcription pre-initiation complex." Thus, we hypothesized that the sequence GAAGAGC predominantly occurs at promoter-proximal regions. This was indeed the case, as we showed in Figure 4D. We also found the association to be statistically significant. ~72.4% of the yeast genome are promoter regions. ~81% (778 of 960) GAAGAGC occurrences are found within the promoter regions (hypergeometric p value $<10^{-9}$). This suggests a potential role for the GAAGAGC motif in regulating gene transcription or other processes downstream of transcription.

DeepBend reveals bendability property and sequence motifs associated with chromatin conformation

Eukaryotic chromosomes are organized into Topologically Associated Domains (TADs) where sequences within a TAD are more likely to interact with each other compared to sequences across TADs. As we reviewed later in discussion, recent studies found several molecular covariates of TAD formation in yeast. However, it has not been reported whether specific sequence motifs influence TAD formation in yeast through controlling DNA bendability. CTCF binding sites are found to form boundaries between TADs in metazoans, and CTCF binding sites were found to have higher DNA bendability than surrounding DNA in five different species.[9] Notably, CTCF is not present in yeast, and the location of TAD in yeast has been explained with nucleosome positioning and histone marks.[22] Nucleosome-resolution mapping of chromosome folding revealed that boundaries separating domains in yeast are strongly enriched for the nucleosome-depleted regions (NDRs) or long linkers (the sequences between consecutive nucleosomes) that are often found in yeast promoters.[22,23] However, not all NDRs form boundaries and strong boundaries tend to occur at promoters of highly transcribed genes. Some other features of yeast boundaries include the enrichment of a variety of histone marks such as H3K4me3 and H3K18ac, high levels of the RSC ATP-dependent chromatin remodeling complex, and high levels of the cohesin loading factor Scc2.[22]

To determine if bendability plays a role in TAD formation and identify the motifs that influence TAD formation, we predicted the bendability of all 16 yeast chromosomes using DeepBend and analyzed a Hi-C matrix of yeast chromosomal contacts at 200bp resolution[24] (see STAR Methods). Our analysis revealed a clear connection between boundary formation and bendability. Comparing the bendability of $\pm$500 bp regions at boundaries against $\pm$500 bp regions at domain sections (see STAR Methods), we found that boundaries have a 120 bp rigid region at the center flanked by two ~400 bp highly bendable regions (Figures 5A and 5B). To further corroborate this finding, we sorted the boundaries into quartiles of their TAD separation score[25] and analyzed DeepBend's predicted bendability values of $\pm$200 bp at the boundaries. Indeed, as we consider boundaries with weaker separation scores, bendability increases at central regions while the linkers become shorter (Figures 5D and 5E). Finally, promoters at boundaries are more bendable on both sides compared to promoters at domains (Figure 5F).

Next, we investigated which sequence motifs are relatively more important for determining bendability in domains and boundaries. We first calculated the motif-matching score, a score derived from

DeepBend denoting the relative match of the learned 256 motifs at some position of a DNA sequence, in all 16 chromosomes of yeast (see STAR Methods). For each motif, we conducted a two-sample z-test[27] for each motif to compare its presence in domains and boundaries in terms of the distribution of its matching scores. We considered matching scores in −250 bp to +250 bp around each boundary middle as the distribution of motifs in boundaries and scores in the remaining chromosomal region as distribution in domains. A positive *Z* score indicates both higher presence and influence in determining bendability in boundaries compared to domains. *Z* score also allowed us to create a ranking for all 256 learned motifs (Table S6). In general, sequence motifs containing poly (dA:dT) showed higher z-scores denoting that they highly influence bendability in boundaries (Figure 5H). Besides these classes of sequences, motifs with lower, but still significant positive *Z* score values contain poly (dC:dG) sequences that are likely to influence bendability at boundaries, albeit to a lesser extent than poly (dA:dT) sequences. Motifs that have the lowest negative z-scores, denoting higher influence on bendability in domains, consist of shorter sequences of A/T and C/G, such as TT and TC dinucleotides (Figure 5I). Our finding conforms with the fact that homopolymeric sequences, poly (dA:dT) and poly (dG:dC) are prevalent in promoters of *S. cerevisiae*[16] where boundaries are more likely to occur and short A/T and G/C sequences are more likely to occur in periodic nucleosomal sequences which are less found in boundaries. Among the motifs with significant z-scores, we show some motifs in Figure 5 that represent the sequence patterns that influence bendability in boundaries and domains. We also ran a motif-matching tool, streme,[26] to find out relatively enriched motifs in domains and boundaries. The motifs found by streme are similar to our motifs with significant z-scores.

From the mean matching score patterns of these motifs in −250 bp to +250 bp around boundaries, we found that prevalent motifs in boundaries, like those containing long A/T sequences, tend to occur more at the boundary centers and gradually less at positions farther from boundaries. Less prevalent motifs in boundaries appear less throughout the surrounding region around boundaries. Prevalence of poly (dA:dT) sequences might create the sharp rigidity in boundaries shown in Figure 5A.[16]

DISCUSSION

DeepBend matched the predictive performance of the best models of large-scale Loop-Seq datasets and provided a unified explanation for this performance. Interpretability has spurred the development of post hoc interpretation methods to infer the sequence motifs potentially driving CNN's predictions. However, such post hoc methods are complicated to use. They involve choosing thresholds for several parameters, but systematic approaches for these steps are still unclear. Furthermore, the efficacy of their discovered motifs in modeling the data has not been rigorously assessed. Finally, it is complicated to check higher-order sequence patterns if they are spread over multiple layers in the model. DeepBend is able to provide the first-order and higher-order spatial patterns underlying high and low bendability thanks to design choices like using a wider second convolution layer and adding positional variance regularisation in the final convolution layer.

DeepBend improves over previous interpretable feature-based models. Such improvement can be attributed to two aspects as follows. Firstly, DeepBend is capable of capturing continuous distributions—it captures the spatial patterns of small motifs in sequence segments rather than fixed features, such as k-gapped dinucleotides or k-mer counts. Convolution captures the spatial relations better. Secondly, DeepBend is capable of identifying more complex probabilistic motifs that are not possible to be captured in simple models.

The motifs found by DeepBend include both known and novel motifs. GAAGAGC and its reverse complement GCTCTTC are the most specific motifs found by our model; indeed, these are the most bendable 7-mers. The regions where our motifs are found are also more likely to be conserved. Regions important for chromatin conformation, such as TAD boundaries and nucleosomal regions show a significant variation in presence of motifs found by DeepBend, which hints that these bendability motifs play a role in the higher-order organization of chromosomes as well.

In conclusion, our work implies that the emerging ideas of visible neural networks and model-based interpretation,[10] where a model designer incorporates domain expertise in the model architecture, can be effectively used in genomics. DeepBend was inspired by these ideas, and we implemented it as a generic model for mapping sequences to quantitative phenotypes. We anticipate that models such as DeepBend

will open up new opportunities to easily identify and visualize the sequence patterns and their spacing rules underlying the mechanical and physical properties of DNA sequences.

Limitations of the study

The hyperparameters such as the number and length of motifs, number, and lengths of higher-order patterns used for our models were not exhaustively searched for finding the most optimal set of motif and pattern representations. In the case where the motifs and patterns are densely packed in small sequences, different sets of motifs can together represent the same sequence patterns. The problem of finding the best representations of motifs using an efficient search algorithm is left to a future study. These are some of the directions in which such interpretable models can be improved. Furthermore, the suitability of DeepBend for finding functional motif patterns for other functional properties also needs to be explored. Additional studies on the structural origin of the bendability of the motifs especially GAAGAGC could also be looked into.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Material availability
  - Data and code availability
- METHOD DETAILS
  - Bendability datasets
  - Preparation of test datasets from libraries with tiled overlapping sequences
  - DeepBend model architecture and rationale
  - Extracting motif pattern from trained model
  - Ranking motif by global importance analysis
  - Motif strong/weak matching score
  - Alternate models and their performances
  - Denoting bendability across chromosome
  - Boundary and nucleosome position determination
  - Comparing bendability of boundaries and domains
  - Bendability quotient
  - Contribution to bendability by motif pairs
  - Finding structural feature of GAAGAGC using DNAShapeR
  - Finding motif presence using FIMO
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2023.105945.

AUTHOR CONTRIBUTIONS

M.A.H.S. and M.S.R. proposed the concepts of the study and supervised the project. S.R.K. organized the datasets, designed and developed the DeepBend model, and did further model analyses. S.S. developed the alternate models and did the analysis of bendability and motifs in chromatin conformation regions. All authors wrote the article.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: August 19, 2022
Revised: December 5, 2022
Accepted: January 5, 2023
Published: February 17, 2023

REFERENCES

1. Harteis, S., and Schneider, S. (2014). Making the bend: DNA tertiary structure and protein-DNA interactions. Int. J. Mol. Sci. *15*, 12335–12363.

2. Peng, J., Yang, J., Anand, D.V., Shang, X., and Xia, K. (2021). Flexibility and rigidity index for chromosome packing, flexibility and dynamics analysis. Front. Comput. Sci. *16*, 164902.

3. Vámosi, G., and Rueda, D. (2018). DNA bends the knee to transcription factors. Biophys. J. *114*, 2253–2254.

4. Basu, A., Bobrovnikov, D.G., Qureshi, Z., Kayikcioglu, T., Ngo, T.T.M., Ranjan, A., Eustermann, S., Cieza, B., Morgan, M.T., Hejna, M., et al. (2021). Measuring DNA mechanics on the genome scale. Nature *589*, 462–467. https://doi.org/10.1038/s41586-020-03052-3.

5. Basu, A., Bobrovnikov, D.G., Cieza, B., Arcon, J.P., Qureshi, Z., Orozco, M., et al. (2022). Deciphering the mechanical code of the genome and epigenome. Nat. Struct. Mol. Biol. *29*, 1178–1187.

6. Zhang, Y., Basu, A., Ha, T., and Bialek, W. (2022). Searching for sequence features that control DNA flexibility. Biophys. J.

7. Lundberg, S., and Lee, S.I. (2017). A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. https://doi.org/10.48550/arXiv.1705.07874.

8. Yu, M.K., Ma, J., Fisher, J., Kreisberg, J.F., Raphael, B.J., and Ideker, T. (2018). Visible machine learning for biomedicine. Cell *173*. https://doi.org/10.1016/j.cell.2018.05.056.

9. Li, K., Carroll, M., Vafabakhsh, R., Wang, X.A., and Wang, J.-P.; 03 (2022). DNAcycP: a deep learning tool for DNA cyclizability prediction. Nucleic Acids Res. *50*, 3142–3154.

10. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. Proc. Natl. Acad. Sci. USA *116*, 22071–22080.

11. Park, M., Singh, S., Khan, S.R., Abrar, M.A., Grisanti, F., Rahman, M.S., and Samee, M.A.H. (2022). Multinomial convolutions for joint modeling of regulatory motifs and sequence activity readouts. Genes *13*, 1614. https://doi.org/10.3390/genes13091614.

12. Wu, H.-M., and Crothers, D.M. (1984). The locus of sequence-directed and protein-induced DNA bending. Nature *308*, 509–513. https://doi.org/10.1038/308509a0.

13. Stefl, R., Wu, H., Ravindranathan, S., Sklenár, V., and Feigon, J. (2004). DNA A-tract bending in three dimensions: solving the dA4T4 vs. dT4A4 conundrum. Proc. Natl. Acad. Sci. USA *101*, 1177–1182.

14. Rosanio, G., Widom, J., and Uhlenbeck, O.C. (2015). In vitro selection of DNAs with an increased propensity to form small circles. Biopolymers *103*, 303–320.

15. Segal, E., and Widom, J. (2009). Poly(dA:dT) tracts: major determinants of nucleosome organization. Curr. Opin. Struct. Biol. *19*, 65–71.

16. Struhl, K., and Segal, E. (2013). Determinants of nucleosome positioning. Nat. Struct. Mol. Biol. *20*, 267–273.

17. Jansen, A., and Verstrepen, K.J. (2011). Nucleosome positioning in Saccharomyces cerevisiae. Microbiol. Mol. Biol. Rev. *75*, 301–320.

18. Grant, C.E., Bailey, T.L., and Noble, W.S. (2011). FIMO: scanning for occurrences of a given motif. Bioinformatics *27*, 1017–1018.

19. Goebel, S.J., Miller, T.B., Bennett, C.J., Bernard, K.A., and Masters, P.S. (2007). A hypervariable region within the 3′ cis-acting element of the murine coronavirus genome is nonessential for RNA synthesis but affects pathogenesis. J. Virol. *81*, 1274–1287.

20. Patarca, R., and Haseltine, W.A. (2021). Structural flexibility of the SARS-CoV-2 genome relevant to variation, replication, pathogenicity, and immune evasion. Preprint at bioRxiv. https://doi.org/10.1101/2021.12.20.473542.

21. Chiu, T.-P., Comoglio, F., Zhou, T., Yang, L., Paro, R., and Rohs, R. (2016). DNAshapeR: an R/Bioconductor package for DNA shape prediction and feature encoding. Bioinformatics *32*, 1211–1213.

22. Hsieh, T.-H.S., Weiner, A., Lajoie, B., Dekker, J., Friedman, N., and Rando, O.J. (2015). Mapping nucleosome resolution chromosome folding in yeast by Micro-C. Cell *162*, 108–119.

23. Wiese, O., Marenduzzo, D., and Brackley, C.A. (2019). Nucleosome positions alone can be used to predict domains in yeast chromosomes. Proc. Natl. Acad. Sci. USA *116*, 17307–17315.

24. Hsieh, T.-H.S., Fudenberg, G., Goloborodko, A., and Rando, O.J. (2016). Micro-C XL: assaying chromosome conformation from the nucleosome to the entire genome. Nat. Methods *13*, 1009–1011.

25. Ramírez, F., Bhardwaj, V., Arrigoni, L., Lam, K.C., Grüning, B.A., Villaveces, J., Habermann, B., Akhtar, A., and Manke, T. (2018). High-resolution TADs reveal DNA sequences underlying genome organization in flies. Nat. Commun. *9*, 189.

26. Bailey, T.L. (2021). STREME: accurate and versatile sequence motif discovery. Bioinformatics *37*, 2834–2840. https://doi.org/10.1093/bioinformatics/btab203.

27. Illukkumbura, A. (2020). Introduction to Hypothesis Testing (Independently Published).

28. Costantino, L., Hsieh, T.-H.S., Lamothe, R., Darzacq, X., and Koshland, D. (2020). Cohesin residency determines chromatin loop patterns. Elife *9*, e59889. https://doi.org/10.7554/eLife.59889.

29. Brogaard, K., Xi, L., Wang, J.-P., and Widom, J. (2012). A map of nucleosome positions in yeast at base-pair resolution. Nature *486*, 496–501.

30. Engel, S.R., Dietrich, F.S., Fisk, D.G., Binkley, G., Balakrishnan, R., Costanzo, M.C., Dwight, S.S., Hitz, B.C., Karra, K., Nash, R.S., et al. (2014). The reference genome sequence of Saccharomyces cerevisiae: then and now. G3 Genes Genomes Genetics *4*, 389–398.

31. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arXiv [cs.DC].

32. Chollet, F. (2015). Keras. and Others. https://keras.io.

33. Koo, P.K., Majdandzic, A., Ploenzke, M., Anand, P., and Paul, S.B. (2021). Global importance analysis: an interpretability method to quantify importance of genomic features in deep neural networks. PLoS Comput. Biol. *17*, e1008925.

34. Huang, R., Sun, H., Liu, J., Tian, L., Wang, L., Shan, Y., and Wang, Y. (2020). Feature variance regularization: a simple way to improve the generalizability of neural networks. AAAI *34*, 4190–4197.

35. Avsec, Ž., Weilert, M., Shrikumar, A., Krueger, S., Alexandari, A., Dalal, K., Fropf, R., McAnany, C., Gagneur, J., Kundaje, A., and Zeitlinger, J. (2021). Base-resolution models of transcription-factor binding reveal soft motif syntax. Nat. Genet. *53*, 354–366.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
DNA bendability datasets	Basu et al. (2021)[4]	https://doi.org/10.1038/s41586-020-03052-3
Yeast chromatin interaction data	Costantino et al. (2020)[28]	https://doi.org/10.7554/eLife.59889
Nucleosome dyad positions	Brogaard et al. (2012)[29]	https://doi.org/10.1038/nature11142
Saccharomyces Genome Database	Engel et al. (2014)[30]	https://doi.org/10.1534/g3.113.008995
Software and algorithms		
Python	Python Software Foundation	https://www.python.org/
Tensorflow	Abadi et al. (2016)[31]	https://www.tensorflow.org/
Keras	Chollet et al. (2015)[32]	https://keras.io/
Multinomial Convolutions for Joint Modeling of Regulatory Motifs and Sequence Activity Readouts	Park et al. (2022)[11]	https://doi.org/10.3390/genes13091614
Global importance analysis to quantify importance of genomic features in deep neural networks	Koo et at. (2021)[33]	https://doi.org/10.1371/journal.pcbi.1008925
Micro-C XL	Hsieh et al. (2016)[24]	https://doi.org/10.1038/nmeth.4025
FIMO	Grant et al. (2011)[18]	https://doi.org/10.1093/bioinformatics/btr064
R	The R Foundation	https://www.r-project.org/
DNAShapeR	Chiu et al. (2015)[21]	https://doi.org/10.1093/bioinformatics/btv735
Code of DeepBend for model training, predicting bendability and interpreting the model.	This paper	https://github.com/SameeLab-BCM/DeepBend

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Md. Abul Hassan Samee (samee@bcm.edu).

Material availability

This study did not generate new unique material.

Data and code availability

- All original code is publicly available at https://github.com/SameeLab-BCM/DeepBend as of the date of publication.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

METHOD DETAILS

Bendability datasets

Basu et al.[4] developed 'loop-seq' assay to measure looping rate of short DNA sequences. They defined the term intrinsic cyclizability which was proved to be highly correlated with DNA bendability. We used their measured values of intrinsic cyclizability as a measure of DNA bendability. These values were obtained from the five sequence libraries, which are Cerevisiae Nucleosomal Library, Random Library, Tiling Library, ChrV Library and Library L, provided as supplementary data with their paper.

Preparation of test datasets from libraries with tiled overlapping sequences

When we trained and tested our model on the same library (see Table S3), we avoided overlapping of sequences between training and test dataset. In the ChrV library, adjacent sequences are offset by 7bp. In the Tiling library, adjacent sequences taken from the same 2001 bp region are also offset by 7bp. The overlap between nearby sequences can range from 43bp to 1bp. There is also a correlation between bendability values of adjacent sequences. In ChrV library, adjacent sequences have a Pearson's correlation (r) of 0.467, 0.329, 0.220, 0.104 and so on. For Tiling library, adjacent sequences have a Pearson's correlation (r) of 0.666, 0.474, 0.318, 0.159 and so on. Training and testing are done carefully when these libraries are used as datasets. If a sequence is taken into the test set, 7 sequences upstream and downstream and the sequence itself is not taken into the training set. This ensures that there is no leaking of information from the test set into the training set. For the test set, we have taken 4% of the total sequences. The ChrV test set consists of 3,297 sequences from a total of 82,405 sequences. The ChrV training set consists of 44,604 sequences. The rest are lost for avoiding leakage. The Tiling library, its test and train set consist of 82,369, 3,205 and 44,657 sequences respectively.

DeepBend model architecture and rationale

The primary goal of using a deep learning model is to improve prediction accuracy and improve interpretability. With our model, we were able to separate out small motifs and the patterns of distribution of these motifs that correspond to bendability positively and negatively. The same motifs, depending on how they are arranged in the sequence, can contribute to positive or negative bendability, our model provides more insight into these motif arrangements.

Our model is a 3-layered Convolutional Neural Network (CNN) regression model that predicts the bendability (intrinsic cyclizibility) of a 50bp sequence. One-hot encoded forward and reverse-complement DNA sequences each of length 50bp and width 4 is given as input to the model. The first two layers are shared between the forward and reverse-complement input sequences so that both strands are considered by the model without compromising training time. The element-wise maximum of the outputs from these sequences are then passed to the next layers.

Layer 1: The first layer is the custom 1D convolution from the MuSeAM model[11] that learns the multinomial distribution of motif patterns. In this multinomial convolutional layer, 256 filters of size 8×4 have been used with "same" padding and stride 1. The filters are applied separately on one-hot encoded forms of both forward and reverse complement sequences. ReLU activation function is applied after convolution operation. The purpose of this layer is to capture small motif patterns. The advantage of using a multinomial convolutional layer is that it provides salient motif patterns without requiring additional post-processing. The outputs of the first multinomial convolution layer are the positive matching scores of the motifs at each position. Using specificity factors, alpha = 75.0 and beta = 1/75.0, provides confident looking motifs without degrading the accuracy of the model.

Layer 2: The second layer is a 1D convolution layer with 2–16 filters, each of length 40 and width 256 and applying "same" padding, stride 1. This layer is used to capture patterns of greater lengths. It is L2 regularised ($\lambda = 0.0005$). The second convolution convolves over the matching scores of all the motifs from the previous layer in segments of the sequence and produces output depending on the relative spatial positioning of these motifs. These convolutions are of length 40 and so they can capture patterns of motif distribution in larger areas. By hyperparameter tuning (see Note S2), we have found for the bendability datasets, increasing the length of filters beyond the input sequence length or using more than 16 filters does not increase performance any further. Furthermore, adding padding allows these convolutions to detect the patterns at any position across the sequence, even near the edges, without having to find multiple shifted versions of the same pattern. Each filter of the second layer F_i is a two-dimensional matrix of size $n_1 \times l_1$, where n_1 is the number of filters in the first layer and l_1 is the distance over which the patterns are assumed to be periodic. The j-th row of F_i corresponds to the j-th filter in the first layer and learns the i-th first-order relative spatial pattern of the j-th motif relevant to bendability. The weights of the filter F_i at each position represent the relative arrangements of the motifs. Each filter F_i represents higher-order distributive patterns of all the motifs together. The model captures the most prominent patterns that strongly influence model performance. Each output feature map of this layer is a vector of length 50 in which a value at a

position denotes how well the region around that position in the input sequence matches the arrangements of motifs represented by the corresponding filter.

Element-wise maximum score is taken between the feature maps of this layer generated from forward and reverse-complement input sequences. After that, the ReLU operation is performed. This propagates the best matching scores to the next layers and allows the model to learn to match motifs at a DNA sequence position irrespective of the strand.

Layer 3: The third layer is a 1D convolution layer with a single filter of width 50 that uses the matching scores with each distribution/arrangement pattern from the previous layer and provides a single floating-point output.

The output of the penultimate layer (second layer in our case) is a 2D matrix U of size $n_2 \times l_2$ where n_2 is the number of filters in the second layer and l_2 is the length of the input sequence. Because of using padding throughout the convolutional layers, the lengths of the inputs and outputs are kept the same. We use $U_{i,j}$ to denote the output for the filter i and position j. The final layer is a convolutional layer with a single filter with no padding and with no activation function at the end. So, the final convolution layer does only one convolutional operation to produce the final floating-point output Y and does not move across the input matrix on any axis. The input dimensions of the last layer weights W correspond to the output dimensions of the previous layer, i.e., $n_2 \times l_2$. The weight $W_{i,j}$ corresponds to or is applied to the output $U_{i,j}$. Our target is to interpret what influences (i.e., increases or decreases) the output Y, i.e., what makes the sequence more bendable (output more positive) or more rigid (output more negative). The outputs for each higher-order feature and input position from the previous layer, $U_{i,j}$, are non-negative numbers corresponding to how well the feature of filter i matched with the input at the corresponding position j. The sign of the weight $W_{i,j}$ of the last layer determines whether the output $U_{i,j}$ contributes positively or negatively to the final output, Y and its value or magnitude determines the contribution factor. If we look at it in another way, for each input position j, every filter from the second layer, F_i has a weight $W_{i,j}$ assigned to it in the last layer. An output/match for the filter F_i at a position j contributes to the final output Y by being factored with that weight $W_{i,j}$. For example, suppose the weight for filter #12 for input position 5, $W_{12,5} = 0.2$. In that case, we can assume that if there is a match of the filter #12 at position 5 then the final output will be moved to the positive direction in proportion to the weight, $W_{12,5}$ and the matching output. In the absence of careful regularization, a filter's contribution may fluctuate between positive and negative values (also of varying magnitudes) across input positions. To limit this, we have added variance regularization[34] along the positional axis of the last layer convolution. As a result, the weights for a particular filter would be of similar value and a filter would contribute similarly to the final score if it matches any of the positions. Thus, we can clearly say in which direction a match of the filter pattern F_i, i.e, the filter itself is contributing, from the corresponding weights in the last layer, $W_i = \frac{1}{l_2}\sum_j^l W_{i,j}$. Then, we can identify the distributive patterns of motifs that contribute positively and negatively to bendability and also their relative magnitude of contribution. The last layer is a convolutional layer instead of a fully connected layer so that we do not lose the positional and filter dimensions. Variance regularisation loss for final layer weights W is calculated as follows:

$$W_i = \frac{1}{l_2}\sum_j^l W_{i,j}$$

$$V_i = \frac{1}{l_2}\sum_j^l \left(W_{i,j} - W_i\right)^2$$

$$L = \frac{1}{n_2}\sum_i^{n_2} V_i$$

Since all the patterns are in the filters of a single layer, we can get the higher-order relations by observing filter weights. We do not need to do a post hoc analysis[35] on our model for finding out the variation of output due to the relative arrangement of motifs.

Extracting motif pattern from trained model

Multinomial convolution operation

The convolution layer in MuSeAM learns multinomial distributions. For this, the convolution matrix W is transformed into a multinomial distribution matrix T where:

$$m_i = max_{j=0}^{3}(W_{i,j})$$

$$T_{i,j} = \frac{e^{\alpha.(W_{i,j}-m_i)}}{\sum_{j=0}^{3} e^{\alpha.(W_{i,j}-m_i)}} \quad \text{(Equation 1)}$$

and α is a free parameter that determines the specificity of the motif. Here i and j are indices for the rows and the columns of T, respectively. For each multinomial convolution matrix of size T and a one-hot encoded nucleotide sequence of length L, a convolution operation computes the term

$$\sum_{i=0}^{L}\sum_{j=0}^{3} ln\left(\frac{T_{i,j}}{B_j}\right).s_{i,j}, \quad \text{(Equation 2)}$$

on each L-length sub-sequence s of S. where B is a background distribution over the four DNA nucleotides. The log likelihood vector is then passed through a ReLU operation which only passes those values which pass a certain threshold, i.e those that have a good enough match. The second convolution layer helps capture the distribution of the motifs in the sequence. The output of the first multinomial convolution layer is the matching score of the motifs at each position.

Extracting motif patterns from multinomial filters

For getting the motif patterns from the convolution matrix, transformation (1) is used to get the probability matrix for the motifs.

Ranking motif by global importance analysis

In order to understand the importance of each of the motifs obtained from our model toward bendability, we check the average change in predicted bendability score for sequences in a dataset when the motifs signals are turned off.[33] The first convolutional layer and the ReLU gives the matching scores of each motif. By turning the outputs of one of the motifs to zeros, the model provides output as if the motif were not present in the sequence. If X are the input sequences in the dataset $f : x \rightarrow y$ is the function of the original model and $f_{motif} : x \rightarrow y$ is the function of the model with the output of a motif turned off then the importance of that motif is, $I_{motif} = \frac{1}{N}\sum_{i}^{N} f(X_i) - f_{motif}(X_i)$. We used the sequences from the ChrV Library for determining the global importance of motifs (Table S5).

Motif strong/weak matching score

The matching scores of motifs are obtained from the output of the first layer of our model. The output of the first layer is the log likelihood of the sequence segment being the motif sequence with respect to some background distribution. If the log likelihood crosses a certain threshold we say that there is a match. That threshold is applied by the ReLU operation after the convolution. We call this output the matching score of a motif, which is obtained at each position for all input sequences. For finding the matching scores of a chromosome of S. Cerevisiae we used the 50bp sequences from that chromosome offset at 7bp from the adjacent ones. We ran the model on these sequences and obtained the output from the first layer, which are matching scores of the motifs of each 50bp sequence at each position. Then we applied ReLU activation function and averaged the overlapping scores to get the matching score of the entire chromosome.

Alternate models and their performances

MuSeAM model

MuSeAM model detects the presence of motifs in the sequence. But fails to capture how these motifs are spread in the sequence. Training with larger motifs in a single layer means that they have to capture combinations of motifs and their distributions. This makes the model larger and harder to interpret. Models we

have trained in this architecture have not reached the expected results. (r: 0.64). We have also tried using filters of different lengths in our MuSeAM model in order to allow capturing larger regions (r: 0.781).

Model with dinucleotide encoded input

To show that the distribution of only dinucleotides is not enough to accurately predict bendability we also trained a CNN model that takes dinucleotide one-hot encoding as input and learns dinucleotide occurrence patterns in the first 1D convolution layer and their distribution in the second convolution layer. This model achieves Pearson's r = 0.802 on the test dataset whereas a previous model which takes dinucleotide counts and gapped dinucleotide counts achieved Pearson's r = 0.6.

Multinomial CNN-RNN model

We have also tried increasing and decreasing model depth, width and changing hyperparameters. We also experimented with RNNs in our second layer. Using a model with multinomial CNN layer and then a bidirectional GRU layer resulted in Pearson's correlation (r) of 0.8452 in the test set. Although the models are very close in their performance, we decided to move forward with the DeepBend architecture as it provided the scope for a better understanding of what makes sequences more or less bendable. All the models have been trained on Tiling library, validated on ChrV test library (see STAR Methods) and tested on Random library. The results of these models are summarised in Table S2.

Machine learning models using sequence features

We used several machine learning models such as Linear Regression, Support Vector Machine and Random Forest to predict bendability of sequences from extracted features. As features, we used the number of times each of the 16 dinucleotides, 64 trinucleotides and 256 tetranucleotides occurred in a sequence and 136 helical separation extent values, which is a measure of tendency of a dinucleotide pair to be separated at helical distance.[16] Thus, each sequence was converted into a 152 feature vector. The models were trained on Tiling library and tested on Random library, Cerevisae nucleosomal library and chromosome V library. The Pearson's correlation between predicted and actual values for these models are shown in Table S1.

Denoting bendability across chromosome

We derived the mean bendability across chromosome V as follows: ChrV library contains bendability values of 50-bp sequences at 7 bp offset. We calculated bendability value at each bp by taking the average of the overlapping 50-bp sequences passing that location. When predicting cyclizability with DeepBend in a chromosome, we similarly used 50-bp DNA sequences at 7 bp offset as input to our model.

Boundary and nucleosome position determination

We obtained yeast chromatin interaction data found with Micro-C XL.[28] From this data, we determined domains and domain boundaries at 200-bp resolution with Hi-C Explorer (calling options: min_depth = 1000, max_depth = 10000, step = 1000, thres_comparison = 0.05, delta = 0.01, correct_for_multiple_testing = fdr). In 16 chromosomes of yeast, we found 2862 200-bp long boundaries in total (Table S4).

We obtained nucleosome dyad locations from.[29] A nucleosome was considered as the region from −73 bp to +73 bp from the dyad, totalling 147 bp. The rest of the DNA sequence was considered as linkers. We downloaded nucleotide sequences of all chromosomes of Yeast from the Saccharomyces Genome Database (SGD).[30] We also downloaded Yeast gene transcripted regions from YeastMine of SGD. We considered the whole transcripted regions as gene.

Comparing bendability of boundaries and domains

With Hi-C Explorer, we determined 200-bp long boundaries in all 16 chromosomes of Yeast and denoted the rest of the chromosome as domain regions. So, in a chromosome with N boundaries, we had N+1 domains. For each boundary, we determined its middle bp and took −499 bp to +500 bp from this middle bp to check bendability of boundary regions. To compare bendability of domains and boundaries, we also took 1000 bp sections in domains. When the length of a domain, L, was not a multiple of 1000, we excluded (L mod 1000)/2 bp from each flank of domain and sectioned the rest. We then took bendability value of boundaries and domain sections. (See how bendability is denoted across chromosome from previous sections).

Bendability quotient

The bendability quotient of a k-mer in a dataset is defined as the average bendability of all the sequences from a dataset that contains the k-mer.[5] We have calculated the bendability quotient using the Random Library as the number of k-mers should theoretically be most equally distributed here.

Contribution to bendability by motif pairs

To show that motif pairs contribute more toward bendability together, we compare sequences that have the greatest presence of the first-order pattern of one of the motifs from the pair and the absence of the other with the sequences with the greatest presence of both of the motifs together. We are considering the first-order patterns for positive bendability. We calculated the presence of the first-order pattern of a motif j in a region $[s, s+k]$ of the sequence as follows,

$$M_{j,[s,s+k]} = \sum F_j \odot m_{j,[s,s+k]}$$

where, F_j is the j-th row of F, the second layer filter for positive bendability. The j-th row corresponds to the first-order pattern for positive bendability for the j-th motif. m is the output of the first layer, which are the matching scores of the motifs. $m_{j,[s,s+k]}$ corresponds to the matching scores for motif j in the region $[s, s+k]$.

Let the indicator variable for the presence of first-order pattern of motif j in region $[s, s+k]$ be,

$$P_{j,[s,s+k]} = \begin{cases} 1 & , \quad M_{j,[s,s+k]} > 0 \\ 0 & , \quad \text{otherwise} \end{cases}$$

Likewise, let the indicator variable for the absence of first-order pattern of motif j in region $[s, s+k]$ be,

$$A_{j,[s,s+k]} = \begin{cases} 1 & , \quad M_{j[s,s+k]} = 0 \\ 0 & , \quad \text{otherwise} \end{cases}$$

Let, us consider a pair of motifs i and j. To find the contributions from only motif i with no interleaving j, we have used the following metric,

$$O_{i,j} = \sum_p \left(M_{i,[p,p+k]} \odot A_{j,[p,p+k]} \right) \qquad \text{(Equation 3)}$$

To find the contributions from both motifs i and j, where both are acting together, we have used the following metric.

$$B_{i,j} = \sum_p \left(\left(M_{i,[p,p+k]} + M_{j,[p,p+k]} \right)^2 \odot P_{i,[p,p+k]} \odot P_{j,[p,p+k]} \right) \qquad \text{(Equation 4)}$$

For each pair of motifs, we use the above metrics to find the contribution of the positive first-order patterns of a motif alone, using Equation (3) and working together, using Equation (4). We have ranked the sequences with the highest contributions for each case. Let us call the sequences the highest contribution from only a single motif from a pair, i.e, $O_{i,j}$, as sequences X. Similarly, let us call the sequences the highest contribution from both motifs from the pair, i.e, $B_{i,j}$, as sequences Y. We found that the median of sequences Y to be more than sequences X in 51% of the motif pairs. In Figure 3D we have shown that considering all motif pair cases, the medians of bendabilities for sequences X are distributed more positively than those of sequences Y.

Finding structural feature of GAAGAGC using DNAShapeR

We used the tool DNAShapeR[21] to reveal the structural features of the GAAGAGC motif. For an input sequence S, DNAShapeR scans S using a 5-bp sized window and for each 5-length sub-sequence M of S, it outputs the predicted value of different shape features (helical twist, minor groove width, etc.) at the central position of M. Thus, for a given sequence S, DNAShapeR predicts structural features for all but the first and the last two nucleotides of S. Accordingly, we identified all occurrences of the GAAGAGC 7-mer in the yeast genome (either in the forward or the reverse strand) and extracted the 11-bp sequences with the two nucleotides flanking each occurrence of GAAGAGC. Then, we used DNAShapeR to predict the values of four prominent DNA shape features, namely Helical Twist (HelT), Minor Groove Width (MGW), Propeller Twist (ProT), and Roll, for each 11-bp sequence extracted above.

Finding motif presence using FIMO

We used FIMO[18] to find the presence of motifs from our model in different sequences. For each 8bp motif, FIMO provides the list of 8bp regions with significant matches, by calculating p value from log likelihood score. We have considered matches with $p < 10^{-4}$ to be significant matches.

QUANTIFICATION AND STATISTICAL ANALYSIS

The models from[5,6] achieve a maximum Pearson's correlation coefficient, r of 0.6. This value translates to a coefficient of determination, $R^2 = 0.6^2 = 0.36$. Since R^2 also gives the fraction of explained variance, we estimated the fraction of unexplained variance to be: 1 - 0.36 = 0.64. Since, some of these models are not linear, hence we mentioned 60% as an approximation of the fraction of unexplained variance.

We used hypergeom.sf (hypergeometric survival function) from scipy.stats to find out p values. We get p value as the probability of getting X successes from N, when there are n successes in a population of M as, pval = hypergeom.sf(x-1, M, n, N). The formula is as follows:

$$SF(X, M, n, N) = 1 - \sum_{k=0}^{X} \frac{\binom{n}{k}\binom{M-n}{N-k}}{\binom{M}{N}}, \text{where, } \binom{n}{k} = \frac{n!}{k!(n-k)!}$$

The parameters for the 2 cases we have calculated hypergeometric p values are as follows.

1. Number of bps in S. Cerevisiae genome, M = 12,157,224, number of bps in regions where motifs are present, n = 3,663,408, number of bps in conserved region, N = 7381044, number of bps in regions where motifs are present and inside conserved region, X = 2,270,652. This gives p value $<10^{-18}$.
2. Number of bps in S. Cerevisiae genome, M = 12,157,224, number of GAAGAGC 7-mers, n = 960, number of bps in promoter region, N = 8,803,581, number of GAAGAGC 7-mers inside promoter regions, X = 778. This gives a p value = $3.42X10^{-10}$.

To compare motif enrichment between boundaries and domains, we employed two-sample Z-test. From the model, we get the motif matching score at each nucleotide position of a chromosome for each individual motif. We take the scores in −250bp to +250bp from boundary middle as the enrichment of motif in a boundary. The rest of the scores in that chromosome are considered to be of domains. For example, DeepBend predicts bendability of 576,871bp of Chromosome V. As we identified 134 boundaries in Chromosome V, the sample size of enrichment scores of a motif in Chromosome V boundaries is 134 × (250 X 2 + 1) or, 67,134 and the sample size of enrichment scores in domains in the same chromosome is (576,871 - 67,134) or, 509,737. To determine Z-test score within our computational limit, we calculated Z-test value in 16 chromosomes separately and then combined these values by taking their weighted average using square root of chromosome length as weight. For each motif m in chromosome c, the Z-test value from boundary and domain samples is:

$$Z_{mc} = \frac{(\mu_{b,mc} + \mu_{d,mc})}{\sqrt{\frac{\sigma^2_{b,mc}}{n_{b,c}} + \frac{\sigma^2_{d,mc}}{n_{d,c}}}}, m = 1, 2, \ldots, 256;\ c = 1, 2, \ldots, 16$$

Here, μ denotes mean of sample, σ denotes SD of sample and n denotes size of sample.

Then, we combined Z-test values from all chromosomes for each motif as:

$$Z_m = \frac{\sum_{c=1}^{16} Z_{mc}\sqrt{n_{b,c} + n_{d,c}}}{\sum_{c=1}^{16} \sqrt{n_{b,c} + n_{d,c}}}, m = 1, 2, \ldots, 256$$