PTF-Vāc: *Ab-initio* discovery of plant transcription factors binding sites using deep co-learning encoders-decoders

Sagar Gupta[1,2], Jyoti[1,2], Umesh Bhati[1,2], Veerbhan Kesarwani[1,2], Ravi Shankar*[1,2]

[1] Studio of Computational Biology & Bioinformatics,
The Himalayan Centre for High-throughput Computational Biology,
(HiCHiCoB, A BIC supported by DBT, India), Biotechnology Division,
CSIR-Institute of Himalayan Bioresource Technology (CSIR-IHBT),
Palampur (HP), 176061, India.

[2] Academy of Scientific and Innovative Research (AcSIR),
Ghaziabad, Uttar Pradesh- 201002

Authors' email addresses:
SG: sagargupta47321@gmail.com
Jyoti: dshwljyoti@gmail.com
UB: bhatiumesh007@gmail.com
VK: veerbhan8586@gmail.com
RS: ravish@ihbt.res.in

*To whom correspondence should be addressed- Email-ravish@ihbt.res.in

Abstract

Discovery of transcription factors (TFs) binding sites (TFBS) and their motifs in plants pose significant challenges due to high cross-species variability. The interaction between TFs and their binding sites is highly specific and context dependent. Most of the existing TFBS finding software tools are not accurate enough to discover these binding sites in plants. They fail to capture the cross-species variability, interdependence between TF structure and its TFBS, and context specificity of binding. Since they are coupled to predefined TF specific model/matrix, they are highly vulnerable towards the volume and quality of data provided to build the motifs. All these software make a presumption or expect that the user input would be specific to any particular TF which renders them of very limited uses. This all makes them hardly of any use for purposes like genomic annotations of newly sequenced species. Here, we report a Deep Encoders-Decoders system, PTF-Vāc, founded on a universal model of deep co-learning on variability in binding sites and TF structure, making it completely free from the bottlenecks mentioned above. It has successfully decoupled the process of TFBS discovery from the prior step of motif finding and requirement of TF specific motif models. Due to a universal model for TF:DNA interactions as its guide, it can discover the binding sites in total independence from data volume, species and TF specific models. It can accurately detect even the binding sites of never seen before TFs families and species and can be used to define credible motifs from its TFBS report.

Introduction

Transcription factors (TFs) play a crucial function in regulating gene expression at the initial stage of transcription by binding specifically to short DNA sequences across the promoter regions. The DNA binding regions, which usually consist of approximately 5 to 20 base pairs (bp) **[1, 2]**, act as the recognition sites for the TFs, allowing them to control the transcription machinery **[3, 4]**. Across the genome, these short, repeating and over-represented sequences define the binding motif for any given transcription factor. Thus, in order to decipher the control of gene expression, it is necessary to identify these binding spots and motifs. Recent technologies like chromatin immunoprecipitation (ChIP) **[5]**, tiling arrays (ChIP on Chip) **[6]**, and next-generation sequencing (ChIP-Seq) **[7]** have enabled direct genome-wide identification of regions bound by any given TF. ChIP experiments allow for motif finding for any particular TF, as the regions obtained from ChIPs are much larger than the actual TFBS while their TF specificity work as the guide to decode such binding motifs for any given TF **[8]**. However, it is not feasible to perform ChIP experiments for all TFs, for every possible conditions and species, as the cost and possible search space shoot exponentially and almost infinitely. Thus the computational models for TF:DNA interactions become indispensable need.

At present, almost every TFBS discovery problem is preceded by motif discovery problem, and this has many issues. In the current landscape of motif search, numerous challenges persist, reflecting the intricate nature of deciphering regulatory elements in genomic sequences. TF specific profiles and position weight matrices (PWM) have been age old and preferred approach where the binding regions are excavated for possible alignments based on which a scoring matrix. The problem can be formalized as “combinatorial optimization”, which involves finding the combination of oligos that build the highest-scoring profile by exploring the search space of all possible combinations with a heuristic and avoiding exhaustive enumeration. Nearly all combinatorial optimization techniques, such as greedy, local search, stochastic search, and genetic algorithms, have been tried and applied over the years **[9, 10]**. The introduction of PWMs, as articulated by Stormo and colleagues, brought a quantitative dimension to motif representation, allowing for probabilistic assessments of TF binding **[11]**. Subsequent milestones included the application of Gibbs sampling and information theory **[12]**. The creation of the databases like TRANSFAC and JASPAR offered the prime repositories of motifs from the experimental binding data **[13, 14]**.

A direct approach is an exhaustive search in which all possible motif candidates are scanned and guaranteed to find the most desirable one. However, such a direct approach frequently becomes unfeasible due to the exponential nature of the problem, resulting in unachievable processing needs **[15-17]**. On the other hand, statistical models, with tools like Consensus **[18]**, MEME **[19]**, and AlignACE **[20]** as the prominent examples, use a direct approach. They make use of probabilistic approach for motif identification in DNA sequences. However, these algorithms have some drawbacks despite their effectiveness. Their sensitivity to user defined parameters is one big drawback, since poor decisions might affect the precision and effectiveness of motif recognition **[19]**. Furthermore, as demonstrated by the exploration-exploitation trade-off difficulties, these algorithms may have trouble with local optima, producing results that are not ideal **[21]**. An additional issue is that the solutions have limited biological interpretability **[22]**.

Later on with rise of machine learning (ML) revolutions, motif discovery also witnessed its application with tools like SVMotif **[23]** which was developed to distinguish between transcription factor binding sites (TFBSs) and non-binding regions by employing support vectors. Because of its adaptability, it was able to incorporate a variety of biological information and anticipate binding sites in sequences that were not yet been identified. However, the shortcomings of such ML approached include their reliance on labeled data, sensitivity to hyper-parameters, restricted biological interpretability, possible scaling problems on large genomic datasets, and their inability to read more complex and hidden features. Advanced ML approaches, notably exemplified by the advent of DeepBind in 2015 **[15]**, signified a paradigm shift towards leveraging Deep learning (DL) for motif finding. DeepBind overlooks more comprehensive sequence contexts due to its emphasis on sequence information. It may need to be trained for use with data of different organisms due to its innate dependency upon species-specific training. Despite this, today DeepBind stands as the backbone of most of the TFBS discovery tools employing deep-learning.

The variability in motif length and composition, coupled with context-dependent functionality, presents challenges in accurately identifying motifs and binding sites. Addressing noisy data and distinguishing genuine motifs from false positives is essential, particularly in the presence of experimental noise and sequencing errors **[24]**. Furthermore, *de novo* motif discovery remains a complex problem, requiring algorithms capable of uncovering unknown patterns without prior knowledge **[25]**. Amid this all, there stands a big lacunae: a total ignorance towards stake of TF structure and its co-variability with the binding site. Very recently our group had conducted a comprehensive review and assessment of these software tools, addressing their shortcomings and proposing ways for improvement **[26]**. The most common and gravest issue was their inability to address cross-species variation rendering them useful only for model species.

Very recently, we identified the issues and developed a software tool, PTFSpot, which learns from the interdependent variability of TF's structures and their binding sites to identify the most potential regions for any TF's binding in total independence of any TF and species specific model **[27]**. The universal TF:DNA interaction model of PTFSpot makes it highly capable to determine the most potential binding regions for even unseen TFs and species. However, PTFSpot proposes a region of ~150-200 bases and mimics ChIP-seq like signals for most potential regions to look for the precise binding sites for any given TF. Unlike ChIP-seq, PTFSpot is not driven by limitations of experimental conditions. In turn, such proposed regions become highly reliable guide on which precise location of TFBS can be identified as well as can be used to build highly reliable motifs for any given TF, if the same structural and binding sites variation co-learning is carried forwards to deep-learning encoder-decoder system. The very same has been implemented in the present work through the developed software tool, PTF-Vāc. PTF-Vāc leverages from the highly refined information generated by PTFSpot to translate the longer binding regions proposed by PTFSpot into the most informative sites, the potential binding sequence/motif components using Transformer encoder-decoder for speech translation. Unlike any existing software, PTF-Vāc is a first of its kind auto-generative deep-learning system to decode precise binding sites of any transcription factor. The outcomes are groundbreaking, as PTF-Vāc consistently delivered best performance, whether given extensive data or just a single sequence, putting it far ahead of the existing software pool for the same purpose. PTF-Vāc marks a new era in plant regulatory research by enabling accurate identification of TFBS in sequences. It has successfully decoupled the problem of motif finding and binding site discovery while working on the plant specific system.

Materials and Methods

Dataset retrieval

For the development of universal and generalized model for the identification of TF binding sites, we retrieved data for *Arabidopsis thaliana* from our previous work, where we created six different datasets using ChIP/DAP-seq **[27]**. The positive instances of binding regions for any TF and its corresponding 3D protein structure associated with each TF were extracted from Dataset “E” while maintaining the methodology introduced by PTFSpot **[27]**. A total of 40 TFs representing 40 different families were considered for model training. By considering this subset of TFs, we could effectively represent the entire set of 325 TFs in Dataset “E” without imposing excessive computational demands. In this dataset, each TF represented a unique TF family, and we selected TFs with the highest binding data within their respective families. This focused selection aimed to overcome the challenges associated with data handling and computational resource requirements. The TF structures were derived using AlphaFold2 **[28]** as experimental 3D structure of most of the plants TFs are unavailable and would be so for long time. Also, for relating structural variability, all TFs structures derived from a common approach solves the purpose of learning the relative changes. The dataset was split in a 70:30 ratio for training and testing purposes.

Transformer-DenseNet system

The 3D structure of a protein plays a crucial role in determining its DNA binding sites like lock and key. In our approach, the structural information of TFs for each of the corresponding binding region extracted from DAP-seq data were integrated. A DenseNet model based on 3D protein structure was raised with a Transformer encoder which takes sequences as input. This integration was achieved through a bi-modal architecture, creating a universal model capable of identifying TF binding sites in a global and cross-species manner. Previously, we had developed a 3D protein structure-based DenseNet which utilized the spatial coordinates of all the atoms in the protein structure as input. In the cases where the length of a protein was shorter, padding with zero values was done for the empty columns. A window size of 300 covered the amino acid positions, with each position having a vector with elements corresponding to the 24 atoms, each of which is mapped to three-dimensional space using normalized X, Y, Z coordinates. Each vector position signifies a specific atom number.

Constructing the DenseNet architecture

The DenseNet model processes a 3D tensor representing protein structure 300×24×3 (300 amino acids, 24 atoms per amino acid, three x, y, z space coordinates for each atom). This architecture, consisted a single convolution layer with 32 convolution filters (kernel size = 3), batch normalization, and 2D max-pooling (stride = 2), and incorporates four dense blocks and three transition layers, totaling 121 layers. Dense blocks, fostering feature reuse through dense connectivity, play a crucial role. In a dense block, each layer ”*l*” receives direct input from all preceding layers, producing output through operations like convolution and normalization. Transition layers effectively reduce the dimensions before the next dense block.

Expressed mathematically, let “ H_l " represent the output feature maps of layer “*I*”. The output of layer “*I*” is computed as:

$$H_l = H_{l-1} \oplus f\left(H_{l-1}, W_l\right) \qquad (1)$$

Here, “ $\oplus$ ” signifies concatenation, “ W_l ” denotes the layer's weights, and “***f***” incorporates batch normalization (BN), rectified linear units (ReLU), and convolution operations. The unique dense connectivity enhances feature reuse, contributing to the model's ability to learn complex hierarchical features effectively.

The growth rate, denoted by the hyperparameter “*k*” in DenseNet, emerges as a pivotal determinant in the architecture's extraordinary performance. DenseNet's innovative design treats feature maps as a global network state, enabling impressive results even with a smaller growth rate. At each layer, “*k*” feature maps contribute to this global state, where the total number of input feature maps $'F_m'$ at the ' l^{th} ' layer is calculated as

$$F_m = k_c + k * (l-1) \qquad (2)$$

The term' k_c ' denotes the total number of channels in the input layer. To enhance computational efficiency, a 1x1 convolution layer precedes each 3x3 convolution layer. This design, known as a bottleneck layer, reduces the number of input feature maps, typically exceeding the output feature maps "*k*", by generating 4k feature maps. This architecture's output is flattened and concatenated in a feed-forward layer within the transformers encoder to capture TFs' relationship with input data. For a more in-depth understanding of the implementation of DenseNet architecture, additional details can be found in our previous study **[27]**.

Word representations of sequence data for Transformer

Each sequence can be considered as a collection of independent words. An alphabet of the four bases allows creating 1024 unique 5-mer words, 4096 unique 6-mer words, and 16,384 unique 7-mer words. Every occurrence of the pentamer, hexamer, and heptamer sequences was transformed for the verbal representation of the sequences in an overlapping window. Pentamers, hexamers, and heptamers provide information on shape and regional motif information **[29-31]**. Transformers takes two input i.e., source and target input. The source input vector had a maximum length of 465 words and a sequence length of 159-162 bases. The target input vector which was motif for that particular sequence had a maximum length of 12 words and a sequence length of 9-12 bases. Also, target sequences contains two flanking words situated adjacent to the binding sites. Distinct integers were used as a token for each unique word comprises of pentameric, hexameric, and heptameric sequences. The tokenized version of the sequence were converted further into numeric vectors and matrices, known as embedding. The transformer encoder fed the tokenized and embedded sequences as input to train and assess the models. The TensorFlow (Keras) tokenizer class was used to implement the tokenization procedure.

Implementation of the Transformers Encoders-Decoders

The encoder-decoder architecture in Transformers is crucial for handling sequential data. The encoder processes the input sequence, extracting hierarchical features, while the decoder generates the output sequence by attending to relevant parts of the input sequence. This bidirectional flow of information ensures the model's ability to capture dependencies and relationships within the input and generate contextually rich sequences as output. The integration of attention mechanisms in both the encoder and decoder enhances the model's capacity to handle transcription factor motif translation.

The Encoder

Encoder comprises of the multi-headed attention mechanism which in itself comprises several self-attention layers, which allow the encoder to attend to different words of the input and consider their relationships when making decisions. The self-attention layers use dot-product attention, allowing the model to weigh the importance of different input elements based on their relationships. This enables the transformer to capture the input data's long-range dependencies and contextual relationships. The sequences from the previous step's tokenized representation were fed into the input layer of the Transformer system. To create an output matrix formatted as samples, sequence length, and embedding size, embedding converted each word's token into a word vector whose length equals the embedded size.

Let's denote the input sequence of input word tokens ***Z*** of length ***I***. The embedding layer projects these discrete input tokens into continuous vector embeddings. If "***v***" is the size of the vocabulary and "***d***" is the dimensionality of the word embeddings , the embedding layer is represented as a matrix $E_{embed} \in \mathbb{R}^{v \times d}$.

$$E_{embed} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdots \\ x_v \end{bmatrix} \begin{bmatrix} W_{11}, W_{12}, \ldots, W_{1d} \\ W_{21}, W_{22}, \ldots, W_{2d} \\ W_{31}, W_{32}, \ldots, W_{3d} \\ \ldots\ldots\ldots\ldots\ldots \\ W_{v1}, W_{v2}, \ldots, W_{vd} \end{bmatrix} \qquad (3)$$

Each word in matrix "E_{embed}" combines with its positional embedding "P", where "P" shares dimension "d" with the word embedding vector. The resulting matrix, E'_{embed}, is obtained through $E'_{embed}=E_{embed}+P$, where P is calculated using sinusoidal positional embedding equations.

$$P_{i,2j}=\sin\left(\frac{i}{10000^{2j/d_{model}}}\right) \tag{4}$$

$$P_{i,2j+1}=\cos\left(\frac{i}{10000^{2j/d_{model}}}\right) \tag{5}$$

Here, i represents the position of the token, and d_{model} is the embedding dimension. E'_{embed} matrix enters the transformer encoder block, which processes it through a multi-head attention layer. This module involves multiple heads, each dividing its query, key, and value parameters N-ways and independently processing the splits. The output from each head produces an attention score derived from the key, query, and value computations.

The process involved the repetition of the same steps, where individual attention vectors were generated and then concatenated. These concatenated vectors were subsequently forwarded to the feed-forward network block within the encoder for further processing. For further insights into the details of the encoder section and the mathematical equations governing the self-attention mechanism, please refer to the PTFSpot work **[27]**. As a measure to mitigate overfitting, the output from the multi-head attention layer was supplied to a dropout layer, succeeded by a normalization layer. Following normalization, the output was directed to fully connected feed-forward layers that converges in the subsequent dropout layer, succeeded once again by a normalization layer. After normalization, the output is seamlessly fused with the DenseNet output, encapsulating vital structural information from the 3D protein structure. This amalgamated result is channelled into fully connected feed-forward layers, converging within the subsequent dropout layer and followed by another normalization layer. Subsequently, this enhanced output from the encoder integrates into the decoder block of the transformer.

Decoder block of the transformer

The decoder block mirrors the structure of the encoder, comprising three sub-layers: Masked Multi-Head Attention, Multi-Head Attention, and Feed-forward Network. Notably, there are two Multi-Head Attention sub-layers in the decoder, with one of them being masked. Each decoder being supplied with two inputs: one originating from target sequence, and the other derived from the encoder's representation. This intricate interplay significantly enhances the decoder's capacity to assimilate information from both its preceding counterpart and the comprehensive encoding of the input sentence by the encoder. The Masked Multi-Head Attention mechanism in the decoder plays a pivotal role in capturing dependencies within the target sequence while preventing information leakage from future positions during training. This attention mechanism allows the decoder to attend to different parts of the input sequence, aiding in the generation of contextually rich representations for each position in the target sequence.

In the contemporary paradigm of Transformer architecture, the mathematical abstraction of the Masked Multi-Head Attention mechanism within the decoder unfolds as an intricate orchestration of operations. Let's consider the target input sequence as X_T and its embedding as $\boldsymbol{E}$, where $\boldsymbol{E}$ is the output embedding matrix obtained after converting the input into word embedding and adding positional encoding, similar to encoder block before being fed into the Masked Multi-Head Attention layer.

Masked Multi-Head Attention

For each distinctive head i, symbolizing $i=1,2,\ldots.,h,$ the calculation of three pivotal matrices ensues: Q_i (Query), K_i (Key), and V_i (Value). This is accomplished through the matrix multiplication of E with the corresponding weight matrices W_{q_i}, W_{k_i}, and W_{v_i}. The formulation of attention scores Z_i transpires through the application of the softmax function to the scaled dot-product attention mechanism, adhering to the expression:

$$Z_i = softmax\left(\frac{Q_i K_i}{\sqrt{d_k}}\right) \qquad (6)$$

To preserve causality during training, a masking mechanism is applied to the attention scores, specifically designed to impede the model from attending to future positions, encapsulating the essence of masked self-attention. The ensuing phase involves the computation of the weighted sum, where each head i contributes to the formation of $Head_i$ via the formula:

$$Head_i = Z_i V_i \qquad (7)$$

A concatenation of the outputs from all heads transpires, culminating in a subsequent multiplication with the output weight matrix W_o, yielding the definitive output of the Masked Multi-Head Attention mechanism in the decoder:

$$MaskedMultiHead(E) = \left(head_1 \oplus head_{2\ldots\ldots} \oplus head_h\right) W_o \quad (8)$$

This comprehensive exposition encapsulates the complex procedures involved in applying masked self-attention across multiple heads within the decoder framework. In the decoding phase of a Transformer model, after the Masked Multi-Head Attention mechanism processes the input sequence, the encoder output is incorporated as the second input for the decoder. The Decoder Multi-Head Attention mechanism is then employed, similar to the encoder, enabling the model to selectively focus on relevant parts of the input sequence and the encoded information. Subsequently, the output from the decoder Multi-Head Attention mechanism undergoes further refinement through the Feed-forward layer. In decoder processing, layer normalization and residual connection enhance information flow, stabilizing the network. Subsequently, the final output undergoes conditional probability estimation, unfolding as a crucial aspect of the decoding mechanism. Building upon the output after layer normalization, the model proceeds to compute the conditional probability distribution over the vocabulary for the subsequent token **[32]**. This involves employing the softmax function, transforming raw scores into a probability distribution. The resultant distribution guides the selection of the next token in the sequence, a pivotal step in generating contextually meaningful and coherent outputs. The conditional probability $P\langle y_t|y_1,\ldots,y_{t-1},x\rangle$ for the next token “ y_t ” is computed using the following equation:

$$P\langle y_t|y_1,\ldots,y_{t-1},x\rangle=softmax\left(w_o h_t\right) \qquad (9)$$

where, “ w_o ” is the weight matrix, and “h_t ” is the hidden state at position “t ". This process repeats iteratively for each time step until the entire sequence is decoded. The model uses the generated tokens as input for subsequent time steps, allowing for autoregressive decoding. In the context of evaluating the Transformer model's decoder performance, the BLEU score serves as a crucial metric **[33]**. The BLEU score measures the similarity between the model-generated sequences and target sequences, providing a quantitative assessment of the quality of the generated output.

The BLEU score is calculated based on the precision of 5-grams in the predicted sequence compared to the reference sequence. The formula for BLEU score is given by:

$$BLEU\ Score=BP\times\exp\left(\sum_{n=1}^{N} w_n\times\log\left(P_n\right)\right) \qquad (10)$$

where, BP is the Brevity Penalty, accounting for the length of the generated sequence relative to the reference. N is the maximum order of 5-grams considered. w_n is the weight assigned to each n-gram precision. P_n is the precision of 5-grams in the predicted sequence. This formula quantifies the precision of the model's output against reference sequences, considering 5-gram orders and assigning appropriate weights.

Generating binding sites from Transformer-Decoder output

The decoder segment of the transformer is designed to generate multiple words (5-mer, 6-mer, 7-mer) as part of its output sequence comprising both the motif and flanking bases for each provided sequence. These multiple words are supposed to map to the target (the binding site) in an overlapping manner where each such word aggregating to a common region in overlaps add to the confidence. The same is repeated for the 5’ and 3’ flanking regions for a single 7-mer words to act as the anchors for binding region and boost the collective confidence value for the detected target region where three such different regions come in support, collectively for a single region and with multiple spoken words. The crucial step post-decoding involves matching these generated words with target sequences. The goal is to identify the most suitable sequence region based on a similarity threshold. For this task, the Jaro similarity algorithm **[34]** was applied. Jaro similarity is a string matching algorithm that give a similarity value between 0 and 1. A value of 0 indicates no match, while 1 signifies a perfect match. The words generated by the decoder are individually matched with the target sequence for its every position in a sliding window manner, and their Jaro similarity scores are calculated. The region with the highest aggregated Jaro score with highest aggregations of the words is identified as a potential binding site component. The Jaro Similarity “$J_{\left(S_i S_j\right)}$ ” between two strings “ S_i ” and “S_j is computed using the following equation:

$$J_{\left(S_i S_j\right)} = \frac{1}{3}\left(\frac{m}{|S_i|} + \frac{m}{|S_j|} + \frac{m-t}{m}\right) \qquad (11)$$

where, “m " is the number of matching character, t is half the number of transpositions (characters that match but are in the wrong order), S_i and S_j are the lengths of the two strings.

Likewise, the anchoring word or the flanking bases, is mapped back to the sequence to determine its anchoring position while calculating Jaro score. In cases where multiple positions were identified within the sequences, preference was given to the position where the flanking anchor words mapped besides the identified binding site as well as the Jaro similarity was calculated max. This intricate process enhances the model's ability to generate meaningful and contextually aligned binding site sequences for transcription factor, showcasing the power of Encoder-Decoder deep-learning system for speech translation from a long DNA sequence to sequence generation of a precise binding site. **Figure 1** depicts the operation of the implemented deep-learning Encoder-Decoder system.

Results and Discussion

Deep co-learning by Transformer-DenseNet system on the words and structure performs consistently

Unlike animals, where TFBSs have been mainly studied around model organisms like human and mouse with not much worries about variability, the scenario in plants is starkly different. Plant genomes are extremely complex and variable, which reflects upon the variability of the binding sites for their TFs. The binding site of a TF on one species may be largely different from the one in another species for the same TF, as has been highlighted by recent studies **[27, 35, 36]**. Recent reports have brought attention to the prevalent under performance in the majority of existing TFBS discovery categories during TFBS annotations, often resulting in a high incidence of false positives **[26]**. A potential solution lies in understanding the variations in TF structure and their corresponding binding sites. Recently, the same was implemented and has been achieved by our group through the software PTFSpot **[27]**. PTFSpot builds upon a novel motif discovery algorithm to define the motifs of different TFs, but the same time very unlike others, downplays them and instead uses them as anchors to retrieve suitable context to learn upon with respect to associated TF and their corresponding variability. PTFSpot employed a Transformer-DenseNet system to perform deep co-learning to propose the most potential binding regions, which can be seen like a virtual ChIP-seq signal without any conditional specificity. This way it solves a much bigger problem of identifying the most potential harboring regions in a sequence for any TF in any given species without relying upon any species and TF specific model.

The process of TFBSs discovery has been overtly intertwined with the problem of motif discovery, which in turn depends upon very precise TF specific information, derived from various binding experiments including ChIP-seq and its derivatives. They provide an assumption that the provided batch of input sequences bind to a specific TF. This has many limitations: 1) Entry of any nonspecific sequence would introduce noise in the data, 2) It is quite impractical to derive such directed TF specific sequences through experimental methods due to cost and possible search space size which could be infinitely large across the species where variability is persistent, and 3) Any such approach is bound to be strongly influenced by the volume of data provided to generate the model as it becomes sharply weaker model at lesser number of sequences and become too much noisy if sequence volume is increased as many other non-binding motifs are also proposed. 4) Above all, these approaches are absolutely of no use for direct genomic scale scanning where no guide exists to help.

In such scenario, the TF binding region proposed by PTFSpot becomes a highly accurate guide, on which suitable further learning can be performed and deep-learning decoders may be applied to speak out the precise TFBSs. In the PTF-Vāc implementation, as outlined in the methods section, a co-learning system was established using a combination of Transformers and DenseNet. This system simultaneously acquired knowledge about TFBS, considering both sequence contexts and the protein structure of the TF. The Transformers part takes two input i.e., source and target input.

The source input is the sequence consisting of contextual information and binding site for a particular TF. The target input is the corresponding binding site whose information was derived from the motifs which PTFSpot had initially developed to anchor and draw the contextual information to derive a universal model for variability and binding associations.

PTFSpot derived binding sites have been derived using well established motif discovery algorithm of RBPSpot **[32]**. Motifs derived this way were highly reliable and exhibited coverage of at least 75% across the bound data for any specific TF. Furthermore, these binding sites demonstrated high statistical significance (binomial test for motif enrichment with p-value <<0.01). Additionally, all matching instances were required to display at least 70% similarity among themselves. Taking these considerations into account, Dataset "E" from PTFSpot was retrieved, while focusing solely on positive instances as the problem in the present work was not classification but text generation (binding site generation). The associated 3D protein structure for each TF was also considered. As outlined in the methods section, the study narrowed its focus to 40 TFs from Dataset "E" effectively representing the entire set of 325 TFs without imposing excessive computational demands, as founded by PTFSpot. The model underwent training and testing on a dataset comprising 48,000 instances of binding regions for the selected 40 TFs, employing a 70:30 split ratio with no overlap. Each TF chosen represented a distinct TF family, providing comprehensive binding data for that specific family. This dataset was passed onto the raised Transformers-DenseNet system. This system was obtained after multiple steps of training and optimization.

An exploration into the performance of varying numbers of encoders and decoders for the Transformers-DenseNet system revealed that at the sixth layer of encoder-decoder pair sharpest rise of performance was noticed which remained at the similar level until the 8th layer addition. After that, performance dipped down again. Thus, the Transformer-DenseNet system in the present work was implemented with six encoder-decoder pairs. The input sequence was padded if the length was shorter or longer than 465 words to ensure a constant size of the input matrix. The output from the multi-head attention layer was passed into the dropout layer with a dropout fraction of 0.1. Its outcome was normalized by a normalization layer. This layer was followed by the third layer having a Feed-Forward network with 64 nodes, followed by another dropout layer with a dropout fraction of 0.1. A feed forward layer with 64 nodes, followed by another normalization layer, followed the above one. The output from the encoder part was transferred to the decoder partners through the decoder's masked multi-head attention unit while preventing information leakage during the training process. Final processing is done by the feed-forward layer of the decoder for the text generation. The cross-entropy loss function was employed for loss calculation with "Adam" optimizer. The learning rate of the optimizer was set to 0.0001 and the model underwent training for 50 epochs with a batch size of 6. The final hyperparameters of the output layer of the optimized model were: {"Activation function": softmax, "Loss function": cross-entropy, "Optimizer": Adam}. The resulting final model was saved in "*pt*" format. The whole Transformers-DenseNet system was implemented in pytorch except the tokenization procedure which was implemented in the TensorFlow (Keras) tokenizer class. The "*pt*" format provided the model graph definition and weights to the pytorch structure while saving the model for translation. **Figure 1** shows the detailed workflow and the implemented transformer architecture in it. On test set, the raised model performance exhibited an outstanding accuracy of approximately 93.2%. Thus, the developed approach's first performance measure demonstrated the raised software as a strong and reliable one for TFBS identification.

The next step was to assess how much the observed performance gained from consideration of the TF's structure while looking into the target binding region sequence. To assess that the deep-learning system was raised again without the 3D structural information of the TF and corresponding

DenseNet parallel system which handled the structural information part. The resulting accuracy for the developed model dropped sharply to 73.9% (**Figure 2a**), a significant drop. This provided strong support for the importance of simultaneously considering sequence and structural variability as previously demonstrated by PTFSpot too. The co-variability learned by the model enables it to accurately identify binding sites.

The identified binding sites displayed significant enrichment and matched the experimentally reported motifs

The next performance evaluation step was to determine the enrichment status of the reported binding sites by PTF-Vāc in the experimental data for any given TF as well as its closeness to the experimentally reported motifs. Experimental information on plant TFs, their binding sites, and interactions is far more limited than that of human TFs. Only for a few plant TFs experimentally validated motifs and matrices are available and for very limited number of species **[37]**. Due to such constraints for experimentally validated motifs availability for plants, the present comparison was restricted to 30 TFs with experimentally reported motifs at JASPAR for *A. thaliana* (**Supplementary Table S1 Sheet 1**). These reported motifs range from 8 to 10 bases. The complexity of plant genomes, species diversity, and the resource-intensive nature of experimental validation are just a few of the obstacles that might make data accessibility in the context of plant genomics a limiting factor. It has to be noted that experimentally retrieved motifs for 30 TFs were not part in either the training or testing datasets.

In order to ensure a uniform motif search approach for a fair comparison, these motifs underwent scanning using a methodology similar to that utilized in the PTFSpot approach **[27]**. Subsequently, the binding sites reported by PTF-Vāc were evaluated for statistically significant enrichment in the corresponding binding data for the given TF. It was discovered that for all the studied TFs their identified motifs were significantly enriched (binomical test p-value < 0.01) for their respective binding region data.

The resemblance between the experimentally reported motif and its corresponding matching motif from PTF-Vāc was assessed using TOMTOM, a part of MEME suite **[19]**. This comparison aimed to assess the degree of similarity and consistency between motifs identified through experimental techniques and those obtained through PTF-Vāc. All motifs retrieved from PTF-Vāc in this study for the 30 TFs matched with the experimentally reported motifs from JASPAR (TOMTOM p-value: <<0.01) (**Supplementary Table S2 Sheet 1**). 100% similarity to the experimentally reported motifs was observed for PTF-Vāc motifs obtained for ANAC046, ANAC055, ANAC083, ANAC092, BIM2, bZIP68, ERF13, GATA15, WRKY18, WRKY30, WRKY45, and WRKY75. PTF-Vāc reported motifs covered at least 83.26% binding data and went up to 100% coverage of the binding data. The consensus motif T[TC]GCCGAC (for RAP2-10) was the least abundant among the PTF-Vāc motifs, covering 83.26% of the bound data. Similarly, the consensus motif [AG][AG]AT[AG]CGCAA[TA], [TC]AAT[CA]ATT, [TC]CAACGC[TG] were the most abundant motif, covering 100% of the bound data for ANAC083, ATHB6, and WRKY30, respectively (**Supplementary Table S2 Sheet 2**). **Supplementary Table S2 Sheet 2** provides a detailed comparison between motifs reported in this study and their experimentally reported counterparts. This part of the study further validated the methodology of PTF-Vāc and its identified binding sites.

In-silico motif validation through molecular docking

We delved into examining the binding affinity between PTF-Vāc and the reported motifs associated with the same TFs using molecular docking. The next set of performance assessment of PTF-Vāc identified binding sites was done to evaluate the binding binding affinity of the identified binding sites with its corresponding TF. Randomly, five TFs were selected for the same: ABI5, GATA11, MYB55, WRKY45, and WRKY75. Due to the general lack of protein 3D structure for these plant TFs, AlphaFold2 **[28]** was used to model their 3D structure. Furthermore, the 3D structures of the identified binding sites were generated using the DNA sequence to structure web-server (http://www.scfbio-iitd.res.in/software/drugdesign/bdna.jsp). Molecular docking were executed employing pyDock **[38]**, and the obtained results were analyzed through Chimera **[39]** and PyMOL **[40]**. Molecular docking has been used analytically in several studies investigating the interaction between TFs and DNA **[41-45]**. The predicted model structures of the five TFs were docked with the binding sites identified by PTF-Vāc and as well as experimentally identified binding sites for the same TF. The docking outcomes revealed that the binding affinities between the identified PTF-Vāc binding sites and their corresponding experimentally reported motifs for each TF were nearly similar. For instance, the binding affinities of GATA11 TF exhibited similar values for PTF-Vāc identified binding sites and the corresponding experimentally reported motif. However, in the case of ABI5, the binding site derived from PTF-Vāc exhibited a higher affinity (docking score: -76.6 Kcal/mol) compared to the experimentally reported motif (docking score: -67.43 Kcal/mol). The details of docking results are given in **Table 1**. Thus, the molecular docking study also supported the identified binding sites from PTF-Vāc.

Comparative benchmarking and performance comparison

In order to get the final picture that how much performance leap PTF-Vāc approach offered over the existing state of the art methods for the same purpose, it was necessary to compare it with the most advanced software tools for objective benchmarking. All of the existing software tools identify motifs from the sequences obtained from the binding experiments for some TF or require some guide where assumption is made that the user provides all sequences coming from the binding data for some given TF. If any diversion from this assumption is made, all these tools fail to generate quality motif due to noise in their input data. As already mentioned above, all the existing methods have coupled motif discovery as prelude to binding sites discovery and for all of them the above mentioned issue prevails. The same factor also leads to their limitations for cross-species use in plants where binding sites vary a lot along with TF structure. This is also the cause that hardly any of the existing methods can credibly work for major jobs like genomic annotations for TFBS for any newly sequenced genome as none of them can work without a predefined motif matrix or PWM which they use for scanning. Expecting such motifs for novel genome wide scan itself is a point of mistake considering the variability across the plant species. The only solution is *ab-initio* binding site discovery which PTF-Vāc has achieved while basing upon the universal binding model of PTFSpot **[27]**. It has learned the relationship between the variability of binding sites, TF structure, and context, making it free from any dependence on predefined motifs and matrices for any possible TF and any possible species. Thus, PTF-Vāc unlike the existing tools is capable to speak out the binding sites *ab-initio* even for single input sequence with the same level as it can do for multiple sequences. PTF-Vāc was compared to four most advanced tools *viz.* deepRAM **[16]**, DESSO **[17]**, TF-MoDISco **[46]**, and SeqConv **[47]** for different performance measure parameters which included: 1) Influence of input sequence data on the binding sites discovery, 2) Overlap and similarity with experimentally reported motifs, and 3) Coverage of overall binding data given for any TF.

DeepRAM and SeqConv: The approach employed by deepRAM deploy convolutional filters for feature extraction in motif identification with a focus on local pattern recognition. On the other side, SeqConv learns motifs from a deep CNN model by extracting active sequences in the rectification layer, yielding a 24-bp subsequence with the maximum convolution value for each sequence. However, their emphasis on individual filters may lead to redundant motif sequences and a limited capture of distributed motif representations.

DESSO: It employs convolutional filters to identify low-level features, followed by max pooling and fully connected layers for high-level feature extraction. Motif detectors learn patterns from DNA sequences, and background sequences are selected to eliminate biases. However, the reliance on a binomial distribution for P-value calculations poses a potential limitation, impacting accuracy in representing motif occurrences, particularly in non-uniform distributions or cases with dependencies among motif instances.

TF-MoDISco: It leverages contribution scores for assessing positional importance, offering insights into the significance of individual bases. It identifies high-importance windows called "*seqlets*", clustering recurring similar “*seqlets*” into motifs after post-processing steps. While per-base importance scores aid in incorporating information from all network neurons, potential limitations arise when the model assign disproportionately high importance to specific positions based on prevalent motifs in the training data. This can lead to biases and potentially result in the overlooking of less frequent but biologically relevant motifs.

All four tools discussed above generate a PWM to discover motifs based on the learned patterns in the input data. However, in instances where these tools encounter noisy data during the learning process, they tend to produce redundant and ineffective motifs. The reliance on PWMs for motif prediction can lead to challenges in accurately discerning meaningful motifs when faced with data complexities. This emphasizes the importance of robust algorithms and noise-resistant strategies in the motif prediction process. In order to offer an unbiased performance comparison for each method, we cross-referenced the identified motifs with experimentally validated motifs obtained from the JASPAR database. We utilized available DAP-seq dataset for 30 TFs to scan and report their binding sites across these sequences (**Supplementary Table S1 Sheet 1)**. As mentioned above, experimentally retrieved motifs for 30 TFs were not part in either the training or testing datasets.

Influence of input dataset sizes on TF binding site discovery

In first phase of our benchmarking, to measure the influence of input sequence data on the binding sites discovery, we conducted an in-depth assessment of the five software (deepRAM, DESSO, SeqConv, TF-MoDISco, and PTF-Vāc). Different input datasets based on number of sequences were constructed from the ChIP-seq dataset for every TF. The input this way ranged from one sequence to the whole dataset. It was observed for majority of TFs, deepRAM needed at last 50 sequences for motif identification whereas DESSO and SeqConv required at least 100 sequences for motif discovery. On the other note, TF-MoDISco demands more than 200 sequences. Only PTF-Vāc discovered the binding sites irrespective of number of input sequences. Moreover, it was observed that deepRAM, DESSO, and TF-MoDISco often produced multiple motifs for a given TF with a sufficient input sequences, whereas only SeqConv and PTF-Vak consistently identified a single motif. **Figure 2b** provides a snapshot of the performance on the WRKY45 TF dataset of varying sizes, clearly showcasing PTF-Vak's ability to discover motifs regardless of input data size, in contrast to other software influenced by varying sequence data sizes. Detailed results of this analysis are presented in **Supplementary Table S1 Sheet 2**.

Exploring similarity and overlap in TF binding site discovery

In the second phase of our benchmarking, we rigorously assessed the overlap and similarity between the motifs discovered by various software and experimentally reported motifs for all the 30 TFs. Using TOMTOM [19], we evaluated the resemblance between experimentally reported motifs and their corresponding motifs discovered by each software. Notably, all motifs identified by PTF-Vāc for the 30 TFs exhibited a significant match with experimentally reported motifs (TOMTOM p-value: <<0.01) **(Supplementary Table S2 Sheet 1).** Impressively, 100% similarity was observed between PTF-Vāc motifs and experimentally reported motifs for 12 TFs, including ANAC046, ANAC055, ANAC083, ANAC092, BIM2, bZIP68, ERF13, GATA15, WRKY18, WRKY30, WRKY45, and WRKY75. Moreover, a minimum overlap of 85% was consistently observed for all TFs between motifs discovered through PTF-Vāc and experimentally reported motifs. **Figure 2c** provide a snapshot of motif similarity between discovered motif and experimentally reported for different software on ten TF. For deepRAM, out of 30 TFs, only 22 TFs matched with the experimentally reported motifs whereas for DESSO, only 14 TFs matched with the experimentally reported motifs. deepRAM displayed a maximum overlap of 100% for four TFs (ABF2, ABI5, bZIP68, ERF4) and minimum overlap of only 20% for some of the TFs (CDF3, MYB55). Subsequently, DESSO displayed a maximum overlap of 100% for three TFs (ATHB7, bZIP68, WRKY75) and minimum overlap of less than 20% for one TF (MYB55). Only six TFs for TF-MoDISco and three TFs for SeqConv matched with the experimentally reported motifs. Both, TF-MoDISco and SeqConv displayed the maximum overlap of 90%.

Evaluating binding data coverage for each TF

In the third phase of our benchmarking, we conducted an exhaustive assessment of the overall binding data coverage for each TF. This investigation primarily focused on the data coverage of motifs discovered by the five software tools. The results revealed PTF-Vāc exhibited the highest data coverage for majority of TFs with at least 83.26% data coverage and went up to 100% coverage of the binding data for some of the TFs (ANAC083, ATHB6, and WRKY30). Additionally, PTF-Vāc identified motif which achieved data coverage of 99% for TFs ABF2, ANAC058, ATHB15, ATHB7, BIM2, CDF3, ERF11, ERF4, ERF8, GATA15, WRKY18, and WRKY25 (**Supplementary Table S2 Sheet 2-6**). **Figure 2d** provides an overview of data coverage for discovered motifs for each software for 10 TFs. DESSO successfully identified motifs for 29 out of 30 TFs, excluding ABI5, with a minimum data coverage of 70%. On the other hand, deepRAM discovered motifs for all 30 TFs, but 14 TFs exhibited a minimum data coverage of 75%. PTF-Vāc and deepRAM were the only tools successful in extracting motifs for all 30 TFs. In contrast, TF-MoDISco failed to extract motifs for several TFs, including ABF2, ANAC058, ANAC083, ANAC092, ATHB6, ATHB7, bZIP68, CDF3, CRF4, ERF4, ERF8, ERF13, ERF105, GATA11, GATA15, MYB55, RAP2-10, and WRKY75. The motifs discovered for the remaining 12 TFs covered less than 90% of the dataset **(Supplementary Table S2 Sheet 2-6)**. Another tool, SeqConv was only capable to identify the motifs for GATA11, WRKY45, and WRKY75 with the data coverage of 69.58%, 80.66%, and 81.68%, respectively (**Supplementary Table S2 Sheet 2-6**).

PTF-Vāc accurately makes cross-species TFBS identification

The raised model's final assessment was set to determine how well it performs to identify cross-species TFBS. In our investigation, we focused on *Zea mays* (seven TFs) and *Glycine max* (three TFs), for which ChIP-seq data and experimentally validated motifs were available. This set of TFs were never utilized before in this study and it was created solely for cross-species validation purpose for the model's performance. The binding sites and motifs identified by PTF-Vāc demonstrated highly significant match (TOMTOM p-value << 0.01) with the experimentally reported motifs for all the considered TFs, affirming its efficacy in handling cross-species conditions. For bHLH47, EREB172, HB34, GLYMA-06G314400, and GLYMA-08G357600 highest match was observed. For three TFs, namely GLK1, MYBR17, and GLYMA-13G317000, the similarity of 88.9% was observed, whereas 77.8% of similarity was observed for EREB97 and TCP23. **Figure 3** provides the comparison details for the identified binding sites by PTF-Vāc and corresponding experimentally validated binding sites for cross-species performance The identified motifs covered maximum data above 90% for bHLH47, EREB97, EREB172, GLK1, HB34, TCP23, GLYMA-06G314400, and GLYMA-13G317000 **(Supplementary Table S2 Sheet 7)**. All these series of validation and benchmarking studies reinforced that PTF-Vāc performed consistently in superlative manner and can be very useful for cross-species TFBS identification without any dependency on predefined models of TFBS.

The curious case of splice variants of transcription factors and their differential binding preferences

This part of the study brings into the notice a very interesting condition of finding the binding sites of splice variants of transcription factors, a question vastly unaddressed in plants. ARF8 has two splice variants in *Arabidopsis thaliana* but the binding motif is defined only for one variant and one completely misses out the binding site detection for the second splice variant of ARF8. Since the existing tools rely upon motif detection approaches, they too are unable to identify the splice variants specific binding even within the same species. This means that with the existing software pool one is far away from properly understanding how various splice variants of any TF work and create spatio-temporal regulatory system for plant's growth and development. The case of ARF8 transcription factor very aptly underlines the same.

To evaluate the efficacy of the developed PTF-Vāc to help answer such situation, we retrieved the experimental data from Ghelli et al. in 2018 **[48]**. In their study, two splice variants of ARF8 exhibited distinct tissue-specific functional roles. ChIP-qPCR results demonstrated that in one scenario, ARF8.2 regulated the IAA19 gene, thereby influencing the auxin pathway. While in a different condition (later flower developmental stages), ARF8.4 regulated both the IAA19 and MYB26 genes, controlling tissue elongation and lignification. In order to validate PTF-Vāc's capability in detecting splice variants' differential binding sites, we initiated a protein structure similarity analysis by superimposing ARF8.2 and ARF8.4, revealing a significant structural difference with RMSD of 1.2 Å. Subsequently, PTF-Vāc was employed for these two variants (**Figure 4**). PTF-Vāc exactly identified the same two genes as the targets of these two variants of

ARF8 which were reported experimentally also. The analysis indicated that ARF8.2 specifically bound to the promoter region of the IAA19 gene. However, ARF8.4 exhibited binding to both IAA19 and MYB26 genes, and the bindings sites of these variants also differed, substantiating founding assumption of PTF-Vāc algorithm that structural changes in transcription factors lead to alterations in its DNA binding sites. This study also highlights the capability of PTF-Vāc to distinguish between splice variants differential binding for any given TF which most of the existing tools are unable to answer.

Web-server implementation

The PTF-Vāc server hosts the software in a user friendly active manner. To identify TFBS, users are required to upload their DNA sequences in FASTA format and its associated TF in "*pdb*" format, followed by clicking the submit button. The input data is then passed first through PTFSpot engine in the back-end which identifies most potential binding regions to be looked upon for the given TFs. Afterwards, these binding regions are passed to PTF-Vāc for TFBS identification. Subsequently, a results page featuring relevant information in an interactive format is presented. Users can download results in tabular format. If multiple sequences are provided, and binding sites of them are identified, a motif logo, a position weight matrix, and a consensus is also generated. User can also select to compare the generated motif's weight matrix with user's choice of weight matrix while using TOMTOM in the back-end. The server implementation has been done using PHP, JavaScript, HTML5, and the Apache-Linux platform. The majority of codes were developed in Shell and Python for jobs related to data curation and back-end processing. The project was executed within the Ubuntu Linux platform's Open Source OS environment. The PTF-Vāc system is freely accessible at https://scbb.ihbt.res.in/PTF-Vac/ .

Conclusion

Cross-species variability is a pronounced feature of plant genomes which is duly reflected across its various components including TFs and their binding sites. This means that the age old practice of using TFBS models and matrices developed for model species for TFBS finding in other species is flawed and a big source of misinformation. Most of the existing tools are coupled to the process of defining a TF specific binding matrix/motif/model using experimental binding data for the given TF for the selected species, as their first step. This way they don't address cross-species variability and are not even able to answer the differential binding preferences of the splice variants of any TF even within the same species. The present work has put forward a completely innovative approach, PTF-Vāc, which has learned the co-variabilities of TF structure and its associated binding sites preferences while utilizing the universal TF:DNA interaction model of PTFSpot. PTF-Vāc uses deep-learning encoder-decoder systems to speak out the binding sites in any given DNA sequence in completely *ab-initio* manner. This makes it completely free of shortcomings of the existing software pool and from any dependence on predefined TF specific models and motifs. Thus, it is capable to accurately identify the binding sites of even totally novel TFs and across even newly sequenced genomes even in total lack of TF information and any prior information. Thus, it is going to have a profound impact in the area of plant science and regulatory research as with PTF-Vāc, one would not need to carry out binding experiments to generate the binding data and identify the binding motifs, and then run them for whole genome scanning. Nor one would be anymore compelled to use TFBS models from other species to find the binding preferences in some other species, an age old wrong practice. And even the differential binding preferences of splice variants of transcription factors withing the same species can be detected accurately. It will need just a simple run of PTF-Vāc server with the user provided DNA sequences and the protein whose binding site the user wants to know.

References

1. Stewart AJ, Hannenhalli S, Plotkin JB. Why Transcription Factor Binding Sites Are Ten Nucleotides Long. Genetics 2012; 192:973–985

2. Mitchell PJ, Tjian R. Transcriptional regulation in mammalian cells by sequence-specific DNA binding proteins. Science 1989; 245:371–378

3. Davidson EH, McClay DR, Hood L: Regulatory gene networks and the properties of the developmental process. Proceedings of the National Academy of Sciences of the United States of America 2003, 100(4):1475–1480

4. Jones DM, Vandepoele K. Identification and evolution of gene regulatory networks: insights from comparative studies in plants. Curr Opin Plant Biol 2020; 54:42–48

5. Ho JW, Bishop E, Karchenko PV, et al. ChIP-chip versus ChIP-seq: Lessons for experimental design and data analysis. BMC Genomics 2011; 12:134

6. Waldminghaus T, Skarstad K. ChIP on Chip: surprising results are often artifacts. BMC Genomics 2010; 11:414

7. Johnson DS, Mortazavi A, Myers RM, et al. Genome-wide mapping of in vivo protein-DNA interactions. Science 2007; 316:1497–1502

8. Zambelli F, Pesole G, Pavesi G. Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. Briefings in Bioinformatics 2013; 14:225–237

9. Fan Y, Kon M, DeLisi C. Ensemble Machine Methods for Analysis of Transcription Factor and DNA Interactions

10. Bucher P. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. J Mol Biol 1990; 212:563–578

11. Stormo GD, Schneider TD, Gold L, et al. Use of the ‘Perceptron’ algorithm to distinguish translational initiation sites in E. coli. Nucleic Acids Res 1982; 10:2997–3011

12. Lawrence CE, Altschul SF, Boguski MS, et al. Detecting Subtle Sequence Signals: a Gibbs Sampling Strategy for Multiple Alignment. Science 1993; 262:208–214

13. Matys V, Kel-Margoulis OV, Fricke E, et al. TRANSFAC® and its module TRANSCompel®: transcriptional gene regulation in eukaryotes. Nucleic Acids Res 2006; 34:D108–D110

14. Sandelin A, Alkema W, Engström P, et al. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Res 2004; 32:D91-94

15. Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature Biotechnology, 33(8), 831–838

16. Trabelsi A, Chaabane M, Ben-Hur A. Comprehensive evaluation of deep learning architectures for prediction of DNA/RNA sequence binding specificities. Bioinformatics 2019; 35:i269–i277

17. Yang J, Ma A, Hoppe AD, et al. Prediction of regulatory motifs from human Chip-sequencing data using a deep learning framework. Nucleic Acids Res 2019; 47:7809–7824

18. Hertz GZ, Stormo GD. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics 1999; 15:563–577

19. Bailey TL, Boden M, Buske FA, et al. MEME Suite: tools for motif discovery and searching. Nucleic Acids Res 2009; 37:W202–W208

20. Roth FP, Hughes JD, Estep PW, et al. Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat Biotechnol 1998; 16:939–945

21. Ma W, Noble WS, Bailey TL. Motif-based analysis of large nucleotide data sets using MEME-ChIP. Nat Protoc 2014; 9:1428–1450

22. Kato M, Tsunoda T. MotifCombinator: a web-based tool to search for combinations of cis-regulatory motifs. BMC Bioinformatics 2007; 8:100

23. Kon M, Fan Y, Holloway D, et al. SVMotif: A Machine Learning Motif Algorithm.

24. Ching T, Himmelstein DS, Beaulieu-Jones BK, et al. Opportunities and obstacles for deep learning in biology and medicine. J R Soc Interface 2018; 15:20170387

25. Bailey TL, Gribskov M. Combining evidence using p-values: application to sequence homology searches. Bioinformatics 1998; 14:48–54

26. Jyoti, Ritu, Gupta, S. & Shankar, R. Comprehensive evaluation of plant transcription factors binding sites discovery tools. 2023.11.07.566153 Preprint at https://doi.org/10.1101/2023.11.07.566153 (2023)

27. Gupta S, Kesarwani V, Bhati U, et al. Deep Co-learning on transcription factors and their binding sites attains impeccable universality in plants. 2023; 2023.11.16.567355

28. Read RJ, Baker EN, Bond CS, et al. AlphaFold and the future of structural biology. IUCrJ 2023; 10:377–379

29. Zhou, T. et al. DNAshape: a method for the high-throughput prediction of DNA structural features on a genomic scale. Nucleic Acids Res 41, W56–W62 (2013)

30. Sharma, N. K. et al. RBPSpot: Learning on appropriate contextual information for RBP binding sites discovery. IScience 24, 103381 (2021)

31. Hombach, D., Schwarz, J. M., Robinson, P. N., Schuelke, M. & Seelow, D. A systematic, large-scale comparison of transcription factor binding site models. BMC Genomics 17, 388 (2016)

32. Wu J, Peng Y, Zhang S, et al. Masked Vision-Language Transformers for Scene Text Recognition. 2022;

33. Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics 2002; 311–318

34. Basak J, Soliman A, Deo N, et al. On Computing the Jaro Similarity Between Two Strings. Bioinformatics Research and Applications 2023; 31–44

35. Kielbasa SM, Gonze D, Herzel H. Measuring similarities between transcription factor binding sites. BMC Bioinformatics 2005; 6:237

36. Shiu S-H, Shih M-C, Li W-H. Transcription Factor Families Have Much Higher Expansion Rates in Plants than in Animals. Plant Physiol 2005; 139:18–26

37. Jin J, Tian F, Yang D-C, et al. PlantTFDB 4.0: toward a central hub for transcription factors and regulatory interactions in plants. Nucleic Acids Research 2017; 45:D1040–D1045

38. Rodríguez-Lumbreras LA, Jiménez-García B, Giménez-Santamarina S, et al. pyDockDNA: A new web server for energy-based protein-DNA docking and scoring. Front Mol Biosci 2022; 9:988996

39. Pettersen EF, Goddard TD, Huang CC, et al. UCSF Chimera—A visualization system for exploratory research and analysis. Journal of Computational Chemistry 2004; 25:1605–1612

40. Seeliger D, de Groot BL. Ligand docking and binding site analysis with PyMOL and Autodock/Vina. J Comput Aided Mol Des 2010; 24:417–422

41. Si J, Zhao R, Wu R. An Overview of the Prediction of Protein DNA-Binding Sites. Int J Mol Sci 2015; 16:5194–5215

42. Konda AK, Farmer R, Soren KR, et al. Structural modelling and molecular dynamics of a multi-stress responsive WRKY TF-DNA complex towards elucidating its role in stress signalling mechanisms in chickpea. Journal of Biomolecular Structure and Dynamics 2018; 36:2279–2291

43. Mukherjee K, Pandey DM, Vidyarthi AS. In Silico Characterization and Analysis of RTBP1 and NgTRF1 Protein Through MD Simulation and Molecular Docking: A Comparative Study. Interdiscip Sci Comput Life Sci 2015; 7:275–286

44. Pandey B, Sharma P, Tyagi C, et al. Structural modeling and molecular simulation analysis of HvAP2/EREBP from barley. Journal of Biomolecular Structure and Dynamics 2016; 34:1159–1175

45. Zhu Q, Zou J, Zhu M, et al. In silico analysis on structure and DNA binding mode of AtNAC1, a NAC transcription factor from Arabidopsis thaliana. J Mol Model 2014; 20:2117

46. Shen W, Pan J, Wang G, et al. Deep learning-based prediction of TFBSs in plants. Trends in Plant Science 2021; 26:1301–1302

47. Shrikumar A, Tian K, Avsec Ž, et al. Technical Note on Transcription Factor Motif Discovery from Importance Scores (TF-MoDISco) version 0.5.6.5. 2020;

48. Ghelli R, Brunetti P, Napoli N, et al. A Newly Identified Flower-Specific Splice Variant of AUXIN RESPONSE FACTOR8 Regulates Stamen Elongation and Endothecium Lignification in Arabidopsis. The Plant Cell 2018; 30:620–637

Acknowledgements

The work was carried out under the aegis of The Himalayan Centre for High-throughput Computational Biology (HiCHiCoB), a BIC supported by DBT, Govt. of India. SG and VK are thankful to DBT, India for financial support as project associateship. UB is thankful for DBT JRF fellowship. Jyoti is thankful for CSIR-UGC JRF fellowship. This MS has CSIR-IHBT MSID ##.

Author’s contributions

SG carried out the major parts of this study and developed the web-server of PTF-Vāc. Jyoti carried out the benchmarking study. UB and VK carried out the analysis in the study. RS conceptualized, designed, analyzed, and supervised the entire study. SG, Jyoti, UB, VK, and RS wrote the MS.

Declaration of competing interest

The authors declare that they have no competing interests.

Funding

The work was carried out under the aegis of The Himalayan Centre for High-throughput Computational Biology (HiCHiCoB), a BIC supported by DBT, Govt. of India [BT/PR40122/BTIS/137/30/2021] and its National Network Project, S2S [Sanction order number: 102/IFD/SAN/397/2022-23 dated 26.08.2022].

Software and Data availability

All the secondary data used in the present study were publicly available and their due references and sources have been provided in **Supplementary Table S1-2**. The software has also been made available as a webserver at https://scbb.ihbt.res.in/PTF-Vac.

Tables

Table 1: The details of docking results for five TFs for their *In-silico* motif validation through Molecular docking. PTF-Vāc gave almost similar binding stabilizing energy as those observed for the experimentally reported binding sites. All of the reported binding sites formed highly stable complex with the corresponding transcription factor.

		Experimentally Reported		**PTF-Vāc**	
S.No.	**TF**	**binding site**	**Binding Energy (ΔG in Kcal/mol)**	**binding site**	**Binding Energy (ΔG in Kcal/mol)**
1	ABI5	TGACACGTGT	-67.43	GACACGTG	-76.6
2	GATA11	TAGATCTG	-111.1	CAGATCTG	-111.33
3	MYB55	ACCTACCG	-69.94	ACCAACCT	-68.79
4	WRKY45	CGTTGACC	-97.86	CGTTGACC	-97.96
5	WRKY75	AAGTCAAC	-97	AAGTCAAC	-97.39

Figure Legend

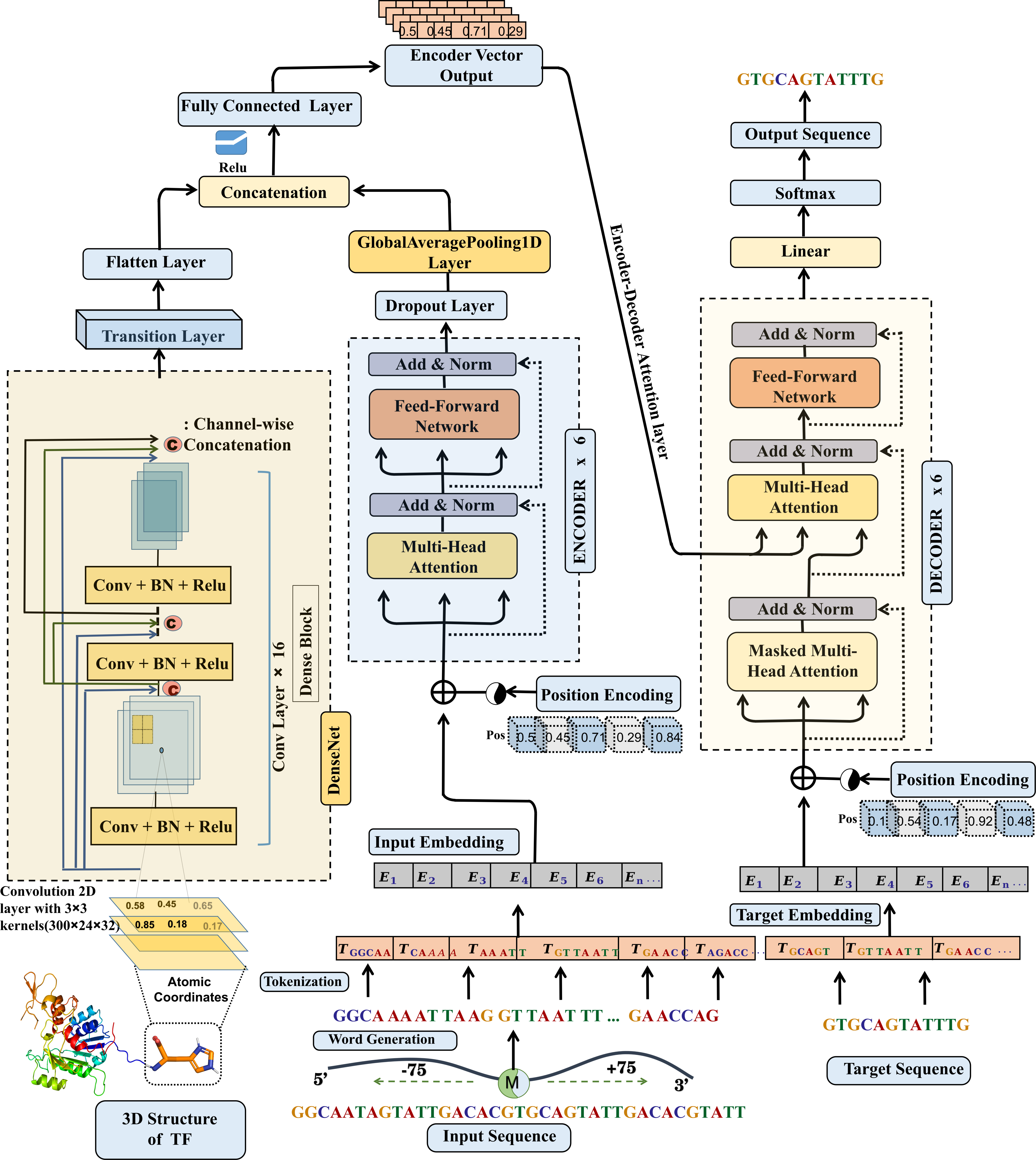
Figure 1: The PTF-Vāc Deep Co-learning system. Bi-modal deep-learning by Transformers and DenseNet and their decoders were implemented to detect TFBS in TF binding regions derived from PTFSpot. The encoder processes binding region sequences represented in penta, hexa, and heptamer word representations, while the decoder receives the binding site information for the corresponding binding region. Six layers encoder:decoder pairs were incorporated. Simultaneously, a DenseNet consisting of 121 layers accepts input in the form of the associated TF's structure. The learned representations from the encoder and DenseNet are merged and passed to the encoder-decoder multi-head attention layer, which also receives input from decoder layers.

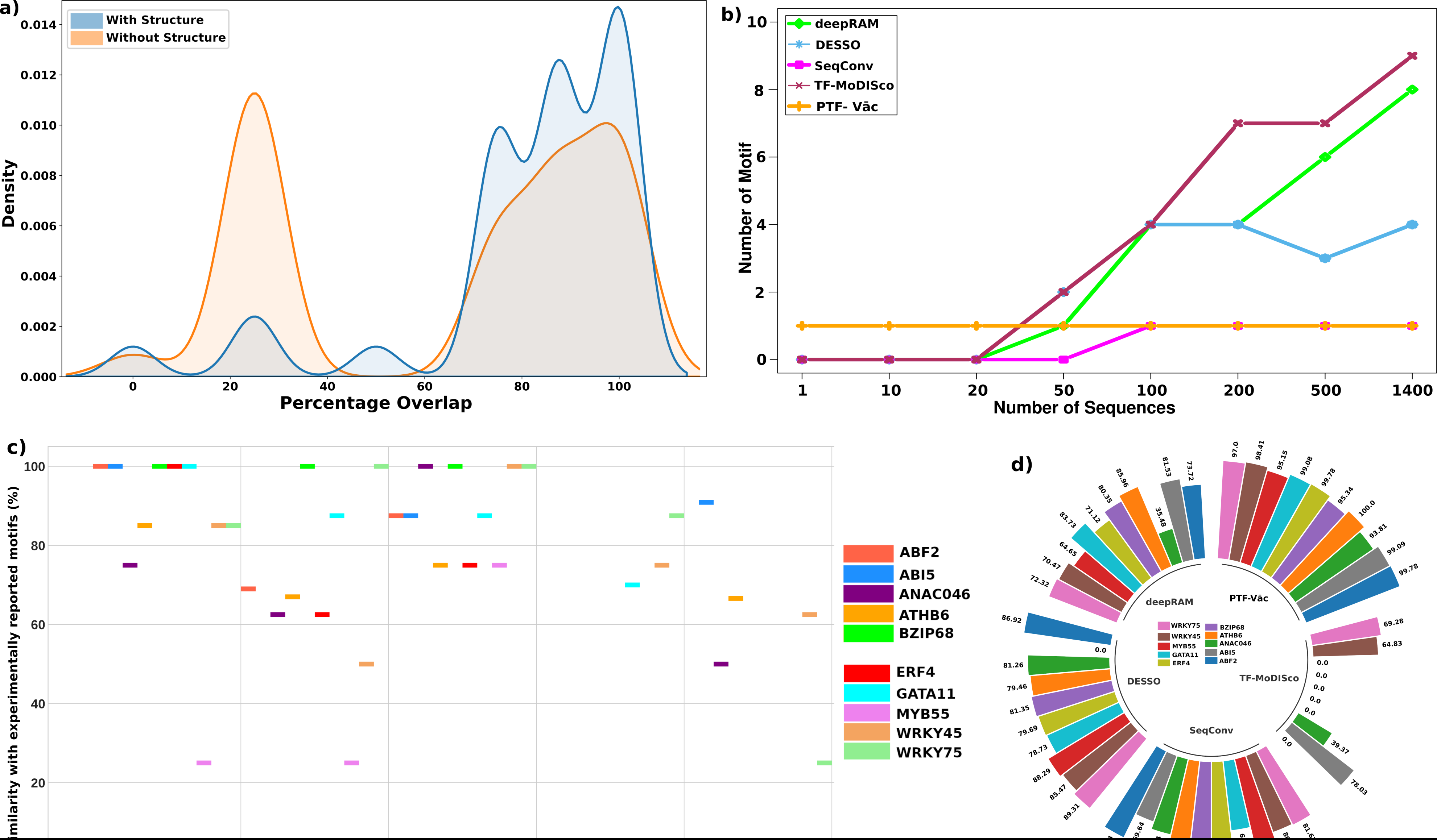
Subsequently, the resulting output undergoes conditional probability estimation, playing a pivotal role in the decoding process. Following layer normalization, the model proceeds to calculate the conditional probability distribution over the vocabulary for the next token. Finally, the resultant tokens were converted back into words, representing the binding site for the given sequence.

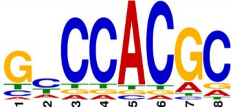
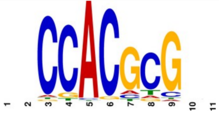
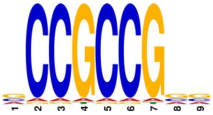
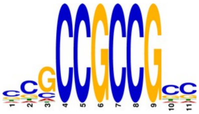
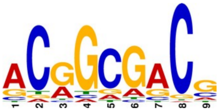
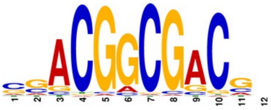
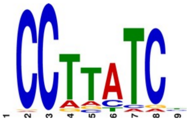
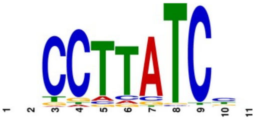
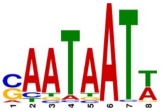
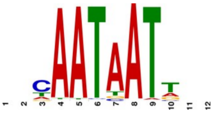
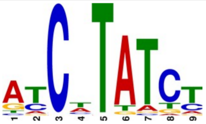
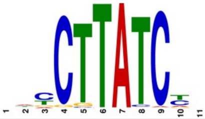
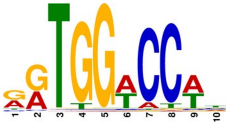
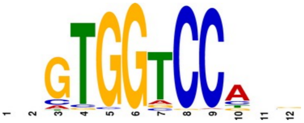
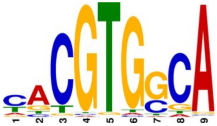
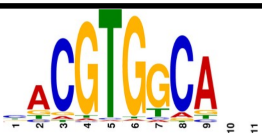
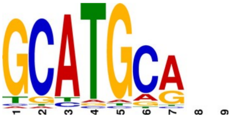
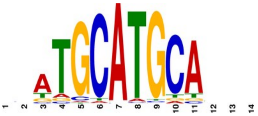
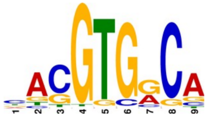
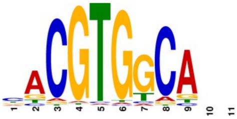
Figure 2: Performance benchmarking of the PTF-Vāc Deep Co-learning system. **a)** The performance of the PTF-Vāc over the test set with and without associated TF 3D structure. Incorporation of co-learning with TF structure played a major role in accurately finding the binding sites. **b)** Performance of different software on WRKY45 TF dataset of varying lengths ranging from one sequence to the whole dataset. Except PTF-Vāc, of of the compared tools' performance varied a lot with variation in the input volume. PTF-Vāc performance was totally unaffected by this, demonstrating its robustness and credibility. **c)** Comparative benchmarking showcasing the discovered binding sites' similarity with the experimentally reported motifs for the selected 10 TF for representation purpose. **d) C**omparative benchmarking showcasing the discovered motif data coverage for 10 TF (for representation purpose). For all studied TFs, the details have been made available in the Supplementary Table S1-2.

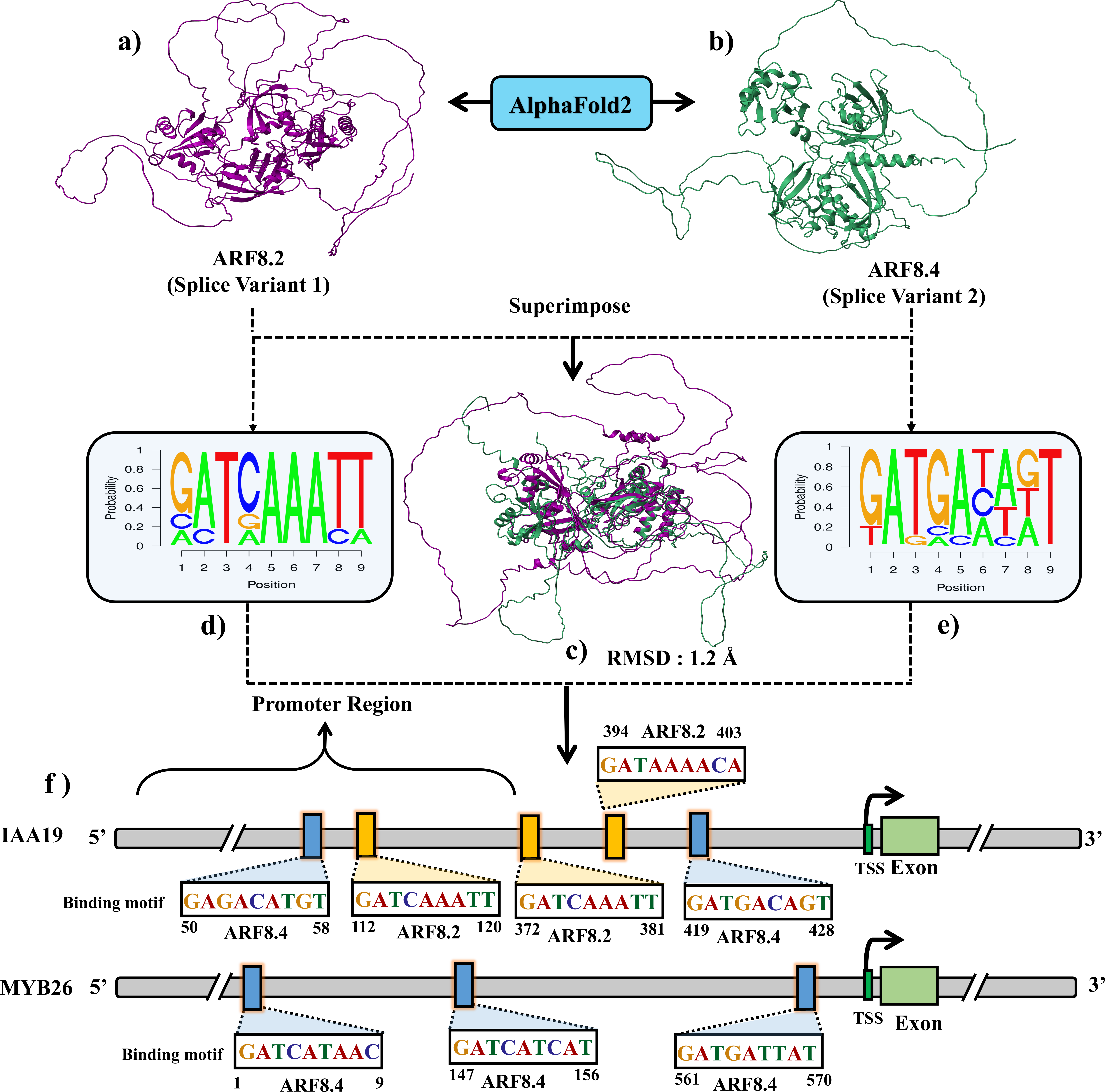
Figure 3: Cross-species performance of PTF-Vāc establishes it as the most reliable tool to identify TFBS across various plant species. Experimentally validated binding data and motifs from Zea *mays* and Glycine *max* were considered for test PTF-Vāc's validity for its ability for cross-species performance. For all these 10 compared TFs and associated binding data, the binding sites and motifs reported by PTF-Vāc were found highly accurate and matching the experimentally reported sites with high significance when tested by TOMTOM.

Figure 4: Structural Superimposition and binding sites of ARF8 Splice Variants. a-b) Presents the 3D structures of ARF8.1 and ARF8.2 splice variants, respectively. **c)** The structural superimposition underscores significant differences between these variants with 1.2 Å RMSD. **d-e**) Motif logos for ARF8.2 and ARF8.4 illustrate distinct binding patterns. **f)** binding sequences reveal the interaction of each splice variant with transcription factor binding sites on the promoter regions of IAA19 and MYB26 genes. Color-coded sequences emphasize variant-specific binding patterns. This figure provides a clear insight into the structural disparities and differential binding sites interactions of splice variants of a TF even within the same species, which most of the existing software fail to detect.


0.5 0.45 0.71 0.29
Encoder Vector Output
Fully Connected Layer
Relu
Concatenation
Flatten Layer
Transition Layer
GlobalAveragePooling1D Layer
Dropout Layer
Add & Norm
Feed-Forward Network
Add & Norm
Multi-Head Attention
ENCODER x 6
Encoder-Decoder Attention layer
: Channel-wise
C Concatenation
Conv + BN + Relu
C
Conv + BN + Relu
C
Conv + BN + Relu
Conv Layer × 16
Dense Block
DenseNet
Position Encoding
Pos 0.5 0.45 0.71 0.29 0.84
Input Embedding
E_1 E_2 E_3 E_4 E_5 E_6 E_n . . .
Convolution 2D layer with 3×3 kernels(300×24×32)
0.58 0.45 0.65
0.85 0.18 0.17
Atomic Coordinates
3D Structure of TF
Tokenization
T_{GGCAA} T_{CAAAA} T_{AAATT} $T_{GTTAATT}$ T_{GAACC} T_{AGACC} . . .
GGCA AA AT TA AG GT TA AT TT ... GA ACCAG
Word Generation
5' -75 M +75 3'
GGCAATAGTATTGACACGTGCAGTATTGACACGTATT
Input Sequence
GTGCAGTATTTG
Output Sequence
Softmax
Linear
Add & Norm
Feed-Forward Network
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
DECODER x 6
Position Encoding
Pos 0.1 0.54 0.17 0.92 0.48
E_1 E_2 E_3 E_4 E_5 E_6 E_n . . .
Target Embedding
T_{GCAGT} $T_{GTTAATT}$ T_{GAACC} . . .
GTGCAGTATTTG
Target Sequence


a)
With Structure
Without Structure
Density
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0
20
40
60
80
100
Percentage Overlap
b)
deepRAM
DESSO
SeqConv
TF-MoDISco
PTF- Vāc
Number of Motif
0
2
4
6
8
10
1
10
20
50
100
200
500
1400
Number of Sequences
c)
imilarity with experimentally reported motifs (%)
20
40
60
80
100
ABF2
ABI5
ANAC046
ATHB6
BZIP68
ERF4
GATA11
MYB55
WRKY45
WRKY75
d)
deepRAM
PTF-Vāc
DESSO
TF-MoDISco
SeqConv
WRKY75
WRKY45
MYB55
GATA11
ERF4
BZIP68
ATHB6
ANAC046
ABI5
ABF2
72.32
70.47
64.65
83.73
71.12
80.35
85.96
35.48
81.53
73.72
97.0
98.41
95.15
99.08
99.78
95.34
100.0
93.81
99.09
99.78
69.28
64.83
0.0
0.0
0.0
0.0
0.0
39.37
0.0
78.03
86.92
0.0
81.26
79.46
81.35
79.69
78.73
88.29
85.47
89.31

Species	TF	PTF-Vāc	Experimentally reported	TOMTOM p-value
Zea mays	**bHLH47**			0.000826
	EREB97			0.00273
	EREB172			8.59E-06
	GLK1			4.54E-07
	HB34			4.05E-05
	MYBR17			4.63E-05
	TCP23			3.29E-06
Glycine max	**GLYMA-06G314400**			1.68E-07
	GLYMA-08G357600			3.98E-06
	GLYMA-13G317000			2.2E-07


a)
b)
AlphaFold2
ARF8.2
(Splice Variant 1)
ARF8.4
(Splice Variant 2)
Superimpose
Probability
1
0.8
0.6
0.4
0.2
0
1 2 3 4 5 6 7 8 9
Position
GATCAAATT
d)
e)
GATGATAGT
c)
RMSD : 1.2 Å
Promoter Region
f)
394 ARF8.2 403
GATAAAACA
IAA19
5’
3’
TSS
Exon
Binding motif
GAGACATGT
50 ARF8.4 58
GATCAAATT
112 ARF8.2 120
GATCAAATT
372 ARF8.2 381
GATGACAGT
419 ARF8.4 428
MYB26
5’
3’
TSS
Exon
Binding motif
GATCATAAC
1 ARF8.4 9
GATCATCAT
147 ARF8.4 156
GATGATTAT
561 ARF8.4 570