



Research article

Predicting DNA toehold-mediated strand displacement rate constants using a DNA-BERT transformer deep learning model

Ali Akay^{a,b}, Hemaprakash Nanja Reddy^a, Roma Galloway^a, Jerzy Kozyra^{a,**}, Alexander W. Jackson^{a,*}

^a *Nanoverly Limited, United Kingdom*

^b *Universita Degli Studi di Trento, Italy*

ARTICLE INFO

Keywords:

DNA nanotechnology
DNA-BERT
Deep learning
Toehold-mediated strand displacement
Convolutional neural network

ABSTRACT

Dynamic DNA nanotechnology is driving exciting developments in molecular computing, cargo delivery, sensing and detection. Combining this innovative area of research with the progress made in machine learning will aid in the design of sophisticated DNA machinery. Herein, we present a novel framework based on a transformer architecture and a deep learning model which can predict the rate constant of toehold-mediated strand displacement, the underlying process in dynamic DNA nanotechnology. Initially, a dataset of 4450 DNA sequences and corresponding rate constants were generated *in-silico* using KinDA. Subsequently, a 1D convolution neural network was trained using specific local features and DNA-BERT sequence embedding to produce predicted rate constants. As a result, the newly trained deep learning model predicted toehold-mediated strand displacement rate constants with a root mean square error of 0.76, during testing. These findings demonstrate that DNA-BERT can improve prediction accuracy, negating the need for extensive computational simulations or experimentation. Finally, the impact of various local features during model training is discussed, and a detailed comparison between the One-hot encoder and DNA-BERT sequences representation methods is presented.

1. Introduction

DNA nanotechnology utilises the predictability, specificity and programmability of Watson-Crick base pairing [1] to construct elaborate nanoscale architectures comprised entirely of synthetic DNA [2]. The field of DNA origami has demonstrated that DNA is a remarkable, smart material for the bottom-up fabrication of well-defined nanostructures [3]. At the turn of the century, the pioneering concept of toehold-mediated strand displacement (TMSD) was conceived [4]. This revolutionary technique employs a partially complementary DNA duplex to accelerate strand exchange. The single-stranded region of the duplex known as the ‘toehold’ recognises the Input strand promoting branch migration and strand exchange, resulting in the liberation of the Output strand and the enthalpically favourable formation of a fully complementary duplex (Fig. 1a).

This ground-breaking mechanism has since facilitated the field of *dynamic DNA nanotechnology* [5]. The integration of dynamic elements into DNA structures has enabled the development of sophisticated DNA nanorobots [6,7], which incorporate a plethora of motifs including switches [8], motors [9], catalytic cycles [10] and walkers [11]. The potential of strand displacement DNA systems

* Corresponding author.

** Corresponding author.

E-mail addresses: jurek@nanoverly.co.uk (J. Kozyra), alexander@nanoverly.co.uk (A.W. Jackson).

<https://doi.org/10.1016/j.heliyon.2024.e28443>

Received 10 October 2023; Received in revised form 15 March 2024; Accepted 19 March 2024

Available online 21 March 2024

2405-8440/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

[12] has far-reaching applications in molecular computing [13–15], synthetic biology [16], chemical synthesis [17], sensing [18], biomarker detection [19–21] and therapeutics [22,23].

One impediment in the application of dynamic DNA nanotechnology is the variable rates of TMSD, complex systems comprised of multiple strand displacement pathways can sometimes be negatively impacted when the rate of TMSD is low. Significant progress has been made in understanding the factors that dictate the rate of TMSD [24,25], which can vary from $10^1 - 10^6 \text{ M}^{-1} \text{ s}^{-1}$. The rate-limiting step of TMSD is the initial bimolecular stage, which involves toehold hybridization between the Probe and the Input strand and the initiation of branch migration. The rate constant of this bimolecular step is defined as k_1 (Fig. 1b). k_1 is affected by environmental factors such as temperature and ionic strength, as these parameters influence the strength of toehold binding. The rate of TMSD is also influenced by the domain-level design of the Probe. The longer the toehold length, the higher the toehold binding energy. This phenomenon plateaus at toehold lengths above 6–7 nucleotides and toehold bind energies below approximately -9 kcal/mol [24]. During our dataset generation we maintain a temperature of 25°C , a NaCl concentration of 1 M and a toehold length of 6 nucleotides. Under these constant conditions, the rate of TMSD is entirely sequence dependent. The Input strand sequence dictates the toehold binding energy, with high C/G content in the 5'-terminus affording strong toeholds. Additionally, undesirable secondary structures, which cause low nucleotide availability, negatively impact TMSD rate constants. Conversely, strong toeholds and good nucleotide availability furnish higher rates of TMSD.

The effective development of complex DNA machinery relies on accurate and predictable rates of TMSD. The rate of TMSD can be determined experimentally. However, this process can be expensive and highly laborious, especially if multiple Probes must be screened. *In-silico* simulations can provide great insights into the kinetics of dynamic DNA systems [26,27]. However, under certain circumstances, simulations can also be suboptimal. When presented with a long natural sequence, determining which section a Probe should target could require thousands of simulations, significantly increasing time and cost. For example, this study's dataset (4450 sequences and corresponding rate constants) was generated in 10 days on a modern machine. Screening a similar set of sequences via the *in-silico* simulation approach would take a comparable time. Hence, the ability to rapidly predict the rate of TMSD would be enormously beneficial.

Artificial intelligence could provide the solution to many issues in the design of dynamic DNA nanotechnologies. In particular, the ability to rapidly predict the rate of TMSD for thousands of potential Input strands would lay the foundation for the optimum design of nucleic acid circuits. Artificial Intelligence has already begun to impact several areas of DNA science. Generalization capacity and the ability of deep learning models to understand patterns within large datasets have enormous potential when applied to nucleic acid sequences. Deep learning models have been successfully employed to analyse next-generation sequencing data [28], in areas such as variant calling, metagenomics, transcriptomics, and epigenetics. However, applying deep learning models within the domain of dynamic DNA nanotechnology has seen little focus from the research community. One noteworthy exception is the study by Andrew Phillips and David Yu Zhang [29]. Specifically, the authors have deployed a bidirectional recurrent neural network deep learning model which predicts the rate constants of DNA hybridization and TMSD, marking the first instance of such integration in the literature. This model was trained using experimentally determined rate constants, a costly and lengthy process. Another exciting example comes from the Simmel group who use a deep learning model to predict the influence of random sequence pools on the kinetics of TMSD [30].

Computational scientists are attempting to find new ways to represent DNA sequences for downstream machine learning applications. Natural language processing (NLP) approaches can be applied to DNA molecules, resulting in unique ways of representing DNA sequences for generalization, classification, and regression. In 2018, the Google AI Language team presented a novel NLP transformer with impressive question-answering and language-understanding capabilities, namely the Bidirectional Encoder Representation from Transformers (BERT) model [31]. In 2021, an innovative variation of the BERT model was applied to DNA sequence understanding, namely the pre-trained model known as DNA-BERT [32]. This model achieved state-of-the-art performance for various sequence prediction tasks and has been successfully deployed in bioinformatics to identify DNA enhancers [33], promoters [34], and N6-methyladenine sites [35].

DNA-BERT is a pre-trained model that learns bidirectional representations of DNA sequences based on the transformer architecture. Its embedding vector captures some general features of DNA sequences that are relevant for genome analysis. Our hypothesis is that the

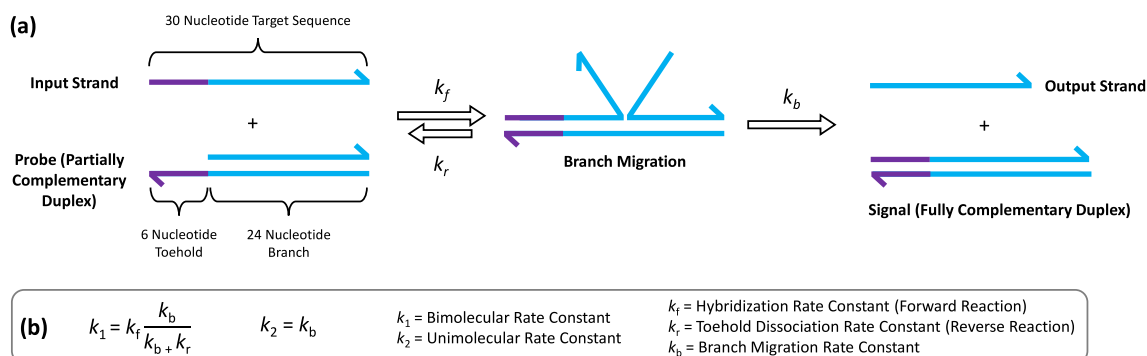


Fig. 1. (a) Overview of toehold-mediated strand displacement pathway and (b) rate constant definitions and equations.

DNA-BERT sequence embedding would also capture relevant information for the prediction of TMSD reactions, such as base composition, GC content and secondary structure. In this investigation, we do not directly use the pre-trained DNA-BERT embedding. We fine-tune the pre-trained DNA-BERT model using our TMSD data set, which captures the sequential and parallel dependencies of the input sequence and adapts its model weights and parameters to generate a new embedding vector for this specific purpose. The data presented demonstrates that the pre-tuned DNA-BERT model has some predictive power in absence of any additional local features, as it is fine-tuned using a TMSD data set, comprised of input sequence and corresponding rate constant.

1D CNNs are neural networks that apply one-dimensional convolution filters to the input data, usually represented as a sequence of word embeddings. The convolution filters can capture local features and n-gram patterns in the text, and the output of the convolution layer is usually passed to a pooling layer to reduce the dimensionality and extract the most salient features. The final layer of the 1D CNN is usually a fully connected layer that performs the classification task. 1D CNNs are fast and efficient to train and can handle variable-length inputs.

Herein, we outline a novel end-to-end solution for predicting TMSD k_1 rate constants (Fig. 2). Our framework (1) takes the gene of interest comprised of thousands of nucleotides, (2) extracts all sequential 30 nucleotide sequences, (3) determines each nucleotide availability within each sequence, (4) assigns nucleotide affinity as either strong (G and C) or weak (A and T), (5) employs a fine-tuned DNA-BERT to obtain the sequence embedding features before (6) predicting strand displacement rate constants for the corresponding TMSD reaction for each sequence. After training, validation and testing, this deep learning model can predict thousands of TMSD rate constants in minutes. This model was trained using simulation data from thousands of 30 nucleotide sequences, with a constant 5' toehold of 6 nucleotides and a NaCl concentration of 1 M at 25 °C. Therefore, the predicted rates hold under typical TMSD reaction conditions. In our methodology, we fine-tune the DNA-BERT transformer model to learn the representation of DNA sequences. A convolutional neural network (CNN) model was applied for training, which takes as inputs (1) the DNA-BERT sequence representation feature, (2) the local features (nucleotide availability and affinity), and (3) the target feature of strand displacement kinetic rate constants obtained from KinDA [27] simulations. The main advantage of our model is the rapid end-to-end prediction of TMSD k_1 values for thousands of Input sequences. As such, this work provides the first evidence of the capabilities of DNA-BERT embedding when combined with a CNN in the field of dynamic DNA nanotechnology.

2. Materials and Methods

The optimum workflow for developing our deep learning model (Model 9) is outlined in Fig. 3. The first step involves the extraction of 4450 sequences from various genes. These sequences are 30 nucleotides long, with 3404 used for model training, 896 for model validation, and 150 for model testing. The TMSD (k_1) rate is then obtained *in-silico* for each sequence, and these simulated kinetics rates comprise our target feature. To represent each sequence, DNA-BERT is employed to encode DNA sequence information, together with nucleotide affinity (either strong or weak) and availability (determined by NUPACK [36–38]). These three representations are considered local features and are subsequently used for training. Subsequently, a CNN is trained, using both the target and local features for each sequence, to predict the rate of TMSD.

A series of models are developed using either One-hot encoder or DNA-BERT sequence representation. Additionally, we compare various combinations of training features to assess the importance of each feature – 12 deep learning models are developed in total. The performance of each model and its influence on feature selection is discussed in the following section. Our deep learning models aim to achieve generalization and accurately predict k_1 when using unseen data. Therefore, we split our data into training, validation, and testing. We use training data to build the model, then split the validation and test set. A validation set aids in evaluating the fitted model without bias while fine-tuning the model parameters. The test set is used to understand how the model performs on unseen data and gain insight into its effectiveness.

2.1. Dataset generation

DNA sequences of 30 nucleotides in length were extracted from the BRAF, EGFR, BRCA2 and VEGF genes. As the primary real-

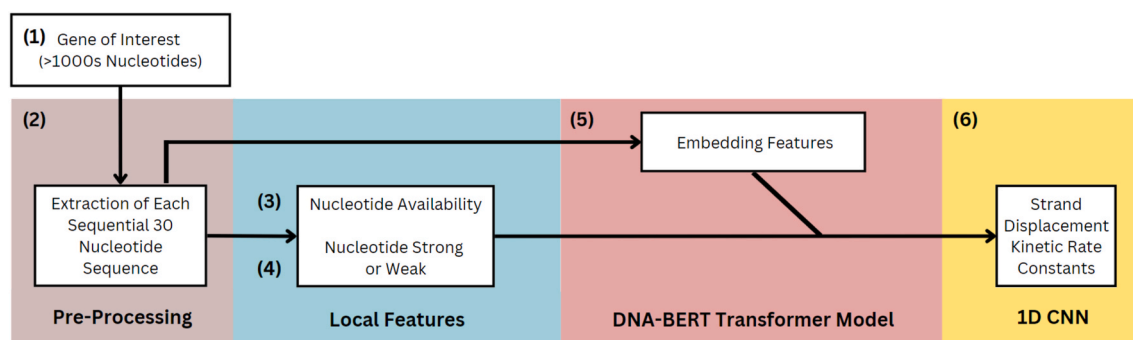


Fig. 2. Overview of the end-to-end DNA-BERT transformer deep learning model framework.

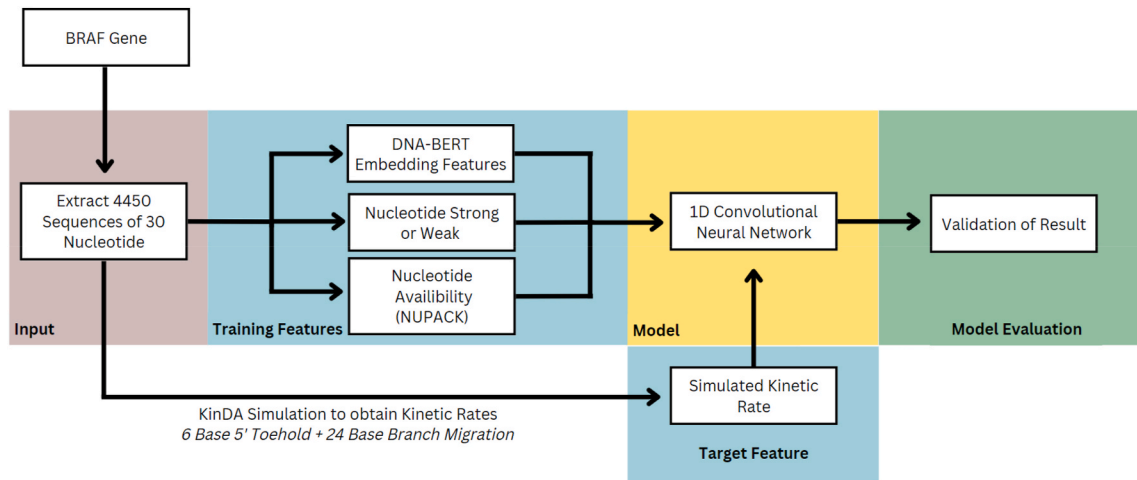


Fig. 3. The training process for the DNA-BERT transformer deep learning model. (1) Random sequence fragments (4450) with length of 30 nucleotides are extracted from 4 different genes, (2) 3 different training features (independent variables) which are later concatenated are generated for each sequence fragment, (3) simulated kinetic rates (dependent variable) are obtained using KinDA software, (4) 1D CNN deep learning model is trained and validated using 4300 sequence fragments and (5) 150 sequence fragments are used to test the model performance.

world application for this deep learning model is to determine the optimum section to probe within a large naturally occurring DNA sequence, training on natural sequences (as opposed to randomly generated artificial sequences) eliminates any potential problems that might arise from artificial sequences. Gene sequences were downloaded from the NCBI database in FASTA format. We selected 4450 target sequences across 4 genes, predominantly the BRAF gene. We generated the toehold-mediated strand displacement rate constants (k_1 and k_2) using publicly available DNA simulation software, namely KinDA [27]. For each sequence, a duplex probe was

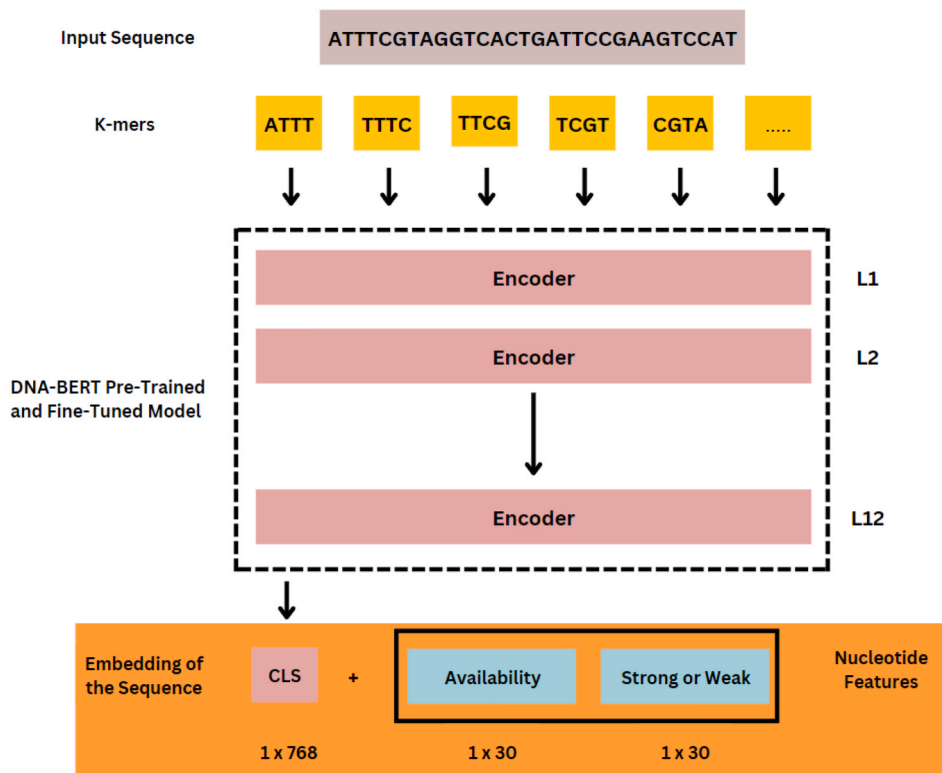


Fig. 4. Architecture of DNA-BERT model and fine-tuning (CLS = classification token). Input sequence is tokenised into k-mers of 4, 12-layered DNA-BERT model is used to generate the CLS output (fine-tuned) which is then concatenated with local features generating an Input with matrix of 1x828 for 1D CNN model.

constructed, comprised of a 6 nucleotide 3' toehold (used to target the 5' regions of the sequence) and a 24-nucleotide branch migration domain (Fig. 1a); rate constants were determined via simulation at 25 °C in 1 M NaCl. Nucleotide availability (equilibrium base-pairing probabilities) for each sequence was computed using locally installed NUPACK software [36]. Also, each base of the target sequence was encoded as either strong (C-G/G-C – 3 hydrogen bonds) or weak (A-T/T-A – 2 hydrogen bonds). Ultimately, we generated the complete dataset comprising target sequence, strong or weak for each base, availability, and kinetic rate constants k_1 and k_2 (the complete dataset is provided in the supplementary information, Table S1). From the data collected, we considered target sequence, nucleotide strong or weak and availability as independent variables and only k_1 as a dependent variable (as the biomolecular step is rate limiting, predicting k_1 provides the best insight into the kinetics of TMSD).

2.2. Deep transformers pre-trained DNA-BERT and fine-tuning

Transformers, primarily used in NLP, are deep neural network models wherein each step has direct access to all other steps. This attention mechanism computes similarity scores between each token and the entire sentence [39] (DNA sequence in this application). Unlike recurrent neural networks, in transformers, sentences are processed as an entire set rather than word by word. The objective is to encode information to a particular token's location inside a sentence using predetermined or learned weights. In the end, the input representation of a token is the sum of the token and positional embeddings. This study employs the DNA-BERT model [32] specifically trained for DNA sequence representation, to learn contextual relations between nucleotides. The original BERT model [31] has 12 layers (where each layer has a hidden size of 768 wt). It generates a feature vector 1×768 for each word/token and token embedding of the classification (CLS). For our application of DNA-BERT, we performed a fine-tuning of the pre-trained model using our dataset (Fig. 4). Different k-mers lengths of 3, 4, 5 and 6 were used on the model, with a k-mer size of 4 providing the best results. We use the DNA sequence length of 30 and k-mers of length 4 which generates 27 k-mers tokens, 1 classification token and 1 separator token (SEP). Resulting in 29 tokens for each sequence used as an input for the DNA-BERT model. After token generation for each sequence and to use the embedded features for regression task, the embedded features were fine-tuned using classification method to get the classification (CLS) output that is used as a feature, which is concatenated with other local features to predict the TMSD rate constant. For the classification method, our simulated k_1 rate constants were used. Labelling k_1 rate constants higher than 90000 as '1' and all others as '0'. Ultimately, we can obtain classified embedding features for each sequence. This fine-tuned DNA-BERT model is used for downstream sequence representation in our framework.

2.3. Convolutional neural network architecture

Convolutional neural networks (CNN) are a class of artificial neural networks. This deep learning method can successfully capture the spatial and temporal dependencies of the input, making CNN more powerful than traditional machine learning methods and providing the ability to solve complex tasks [40]. In this study, we used a 1D CNN model to learn from the features extracted from our fine-tuned DNA-BERT model, combined with the other training features, specifically nucleotide availability and strong/weak designation. Since we integrated the nucleotide availability feature of matrix 1×30 and the nucleotide affinity (strong or weak) feature of matrix 1×30 with the output of the DNA-BERT model of matrix 1×768 , the input to the CNN is of matrix 1×828 . Our CNN

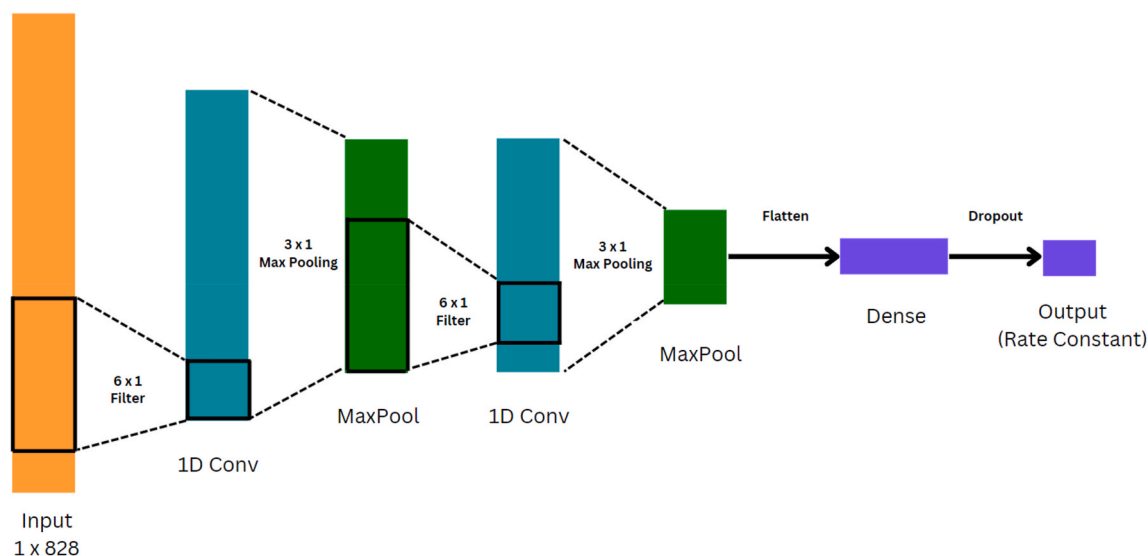


Fig. 5. The architecture of CNN deep learning model. Input with matrix size of 1×828 , two sets of alternative 1D Convolutional + ReLU + filters with kernel size 6×1 and max-pooling with kernel size 3×1 layer that convolutes and selects the relevant features respectively, two linear dense layers with a dropout layer used to prevent overfitting problem and final CNN output predicting the kinetic rate constants.

architecture consists of two sets of 1-dimensional convolutional and pooling layers and two linear dense layers (Fig. 5 and Table 1). Convolutional layers convolute the input and pass its output to the next layer – this process can be considered akin to a neuron’s response to the stimuli as it allows the model to explore the sequential correlations over the features in the input [41]. Pooling layer is used to decrease the dimensions of the feature map and choose the most appropriate features. Max-pooling [42] operation was used in this work. To find the optimal hyperparameters, various values for kernel size, learning rate, filter size and dropout were implemented and evaluated.

In the convolutional layers, a filter with a kernel size of 6×1 was applied over the input data, and the widely preferred rectified linear units (ReLU) activation function was used [43]. The output from the convolutional layer was passed to the pooling layer with a kernel size of 3×1 . The output from the last max-pooling layer consisting of extracted relevant features was flattened and passed to linear dense layers to obtain the final output of the model. The learning rate and dropout parameters were set to 0.001 and 0.1, respectively. A dropout layer [44] was applied between the two dense layers to prevent the overfitting problem.

2.4. Model evaluation

This work evaluated each model using quality parameters, specifically, root mean square error (RMSE) and R-squared value (R^2). To assess the accuracy and predictive power of each model, the root mean square error (RMSE) was utilized as a measure of the variance between the actual (k_{Obs}) and predicted (k_{Pred}) values. This allowed for an evaluation of the performance of each model and a better understanding of its capabilities.

3. Results and discussion

DNA sequence representation is the most critical part of deep learning implementation in DNA nanotechnology. It is required to reduce high-to low-dimensional data to discover DNA sequence patterns. In this study, we aimed to gain an insight into the potential of deep learning when applied to dynamic DNA nanotechnology. This research focused on two main objectives. Firstly, we wanted to assess the power of DNA-BERT sequence representation compared to the One-hot encoder method. The One-hot encoder method is commonly used for DNA sequence representation [45,46]. In models utilizing the One-hot encoder approach (Models 1–6), DNA sequences were encoded into a matrix of 30×4 for each sequence. Then we concatenate with availability (30×1) and strong or weak (30×1), affording a 30×6 matrix for each sequence. The input matrixes were fed into a 1D CNN to obtain TMSD k_1 rate constant predictions. Secondly, the influence of different nucleotide properties as prospective model features were investigated. The overarching goal is to develop a deep learning model that achieves a robust predictive performance with a minimal and optimal selection of features combined with the most potent sequence representation tool (Fig. 6a–c).

In total, we developed 12 CNN deep learning models, 6 deploying DNA-BERT sequence representation (Models 7–12) and 6 with the One-hot encoder approach (Models 1–6), with varying combinations of local features (Fig. 6). Scatter plots of the validation, model losses and test results display the One-hot encoder (Fig. 7) and DNA-BERT (Fig. 8) model capabilities.

To understand the contribution of distinctive features with respect to model performance. A series of models (Figs. 6–8) were constructed in which different features were removed from the model to understand which features contribute meaningfully to prediction performance measured by RMSE score and R^2 value.

Comparing Models 1 and 7 confirms that when used without additional features, the One-hot encoder sequence representation cannot predict TMSD k_1 (RMSE = 1.49 and R^2 = 0.09). Conversely, the DNA-BERT representation feature does provide some predicting capabilities (RMSE = 1.08 and R^2 = 0.52). Models 2 and 8 demonstrate the significant impact of nucleotide availability on the ability of the CNNs to predict K_1 . The advantage gained by adding availability is observed for both the One-hot encoder (RMSE = 0.90 and R^2 = 0.67) and DNA-BERT (RMSE = 0.86 and R^2 = 0.70) models.

The addition of nucleotide strong (C/G) or weak (A/T) furnishes further improvements (Models 3 and 9), although the increase is not as pronounced as with the previous addition of availability. Noticeably, DNA-BERT sequence representation (RMSE = 0.76 and R^2 = 0.76) outperforms the One-hot encoder (RMSE = 0.86 and R^2 = 0.70) approach.

Substituting nucleotide affinity (strong or weak) for nucleotide purine (A and G) or pyrimidine (T and C) hinders model performance (Models 4 and 10). This is expected as nucleotide purine or pyrimidine does not affect the rate of TMSD. These models still display reasonable performance as the critical feature of nucleotide availability remains incorporated.

The importance of nucleotide availability is further confirmed when nucleotide affinity (strong or weak) and type (purine or pyrimidine) are used in combination (Models 5 and 11). Here, the absence of availability is stark, as predictability falls back in line

Table 1
CNN total trainable parameters: 13,386,561.

Layer	Output Shape	Parameters
Conv-1D	(None, 64, 823)	448
MaxPooling1D	(None, 64, 821)	0
Conv1D	(None, 128, 816)	49,280
MaxPooling1D	(None, 128, 814)	0
Flatten	(None, 104192)	0
Dense Layer	(None, 128)	13,336,704
Dense Layer	(None, 1)	129

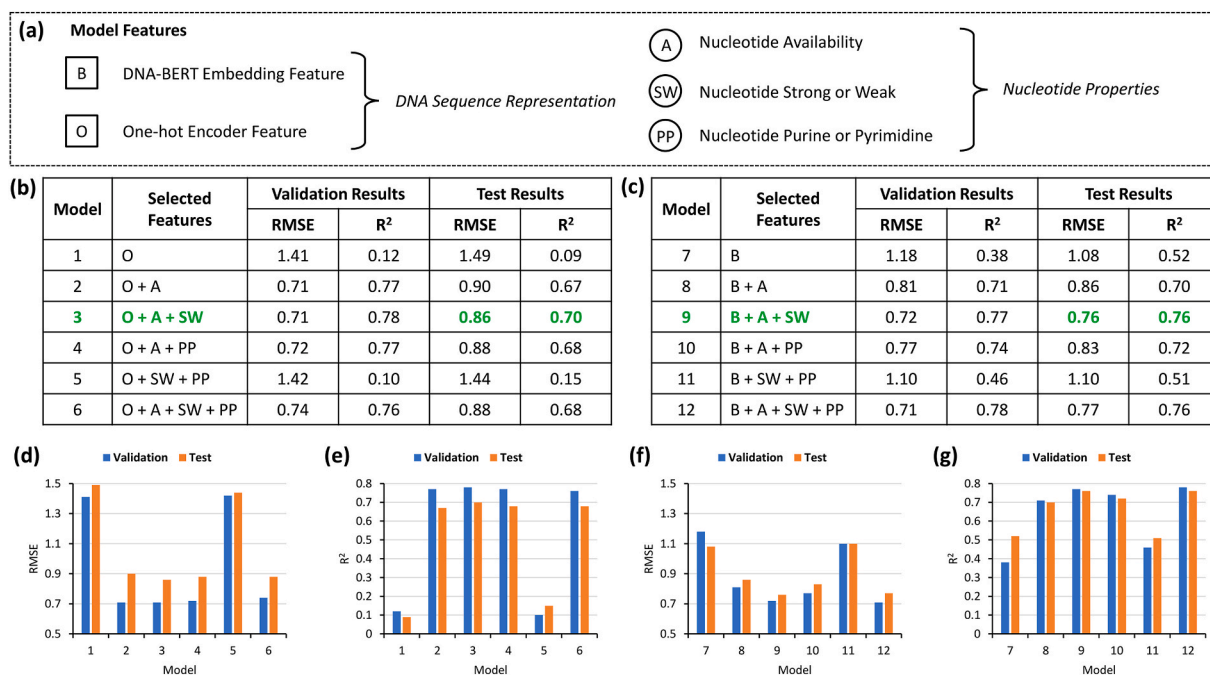


Fig. 6. (a) Selection of features including DNA sequence representation tools and nucleotide properties used throughout model development, (b) description of One-hot encoder models and corresponding validation and test results, (c) description of DNA-BERT models and corresponding validation and test results and (d–g) plots of validation and test results for each model (RMSE and R²) to better visualize disparities between validation and test performance.

with Models 1 and 6 when only sequence representation is deployed.

Finally, all available features are combined (Models 6 and 12), which affords no further improvement over the optimum selection of nucleotide availability and affinity (Models 3 and 9).

These results demonstrate that sequence representation is only one of the requirements in the development of a powerful deep learning model in the field of dynamic DNA nanotechnology. For the prediction of TMSD rate constants, incorporating local features, specifically nucleotide availability and affinity, are vital additions during model training. These findings concur with the established knowledge in dynamic DNA nanotechnology and TMSD. Poor nucleotide availability, caused by the formation of secondary structures, can inhibit toehold binding. Moreover, when the toehold length is fixed, the G/C content determines toehold binding energy and, subsequently, the rate of TMSD.

One of the most interesting observations is the similar performance between the One-hot encoder and DNA-BERT models during the validation stage. Models incorporating availability (Models 2, 3, 4 and 6 for One-hot encoder and Models 8, 9, 10 and 12 for DNA-BERT) show similar validation results. The One-hot encoder (Model 3) even slightly outperforms its DNA-BERT counterpart (Model 9). Nevertheless, these validation results are not replicated during the independent testing stage. DNA-BERT representation models display no significant difference between validation and test results (Fig. 6f and g). In contrast, the One-hot encoder models return significant variations between the validation and test results (Fig. 6d and e) with the ability to accurately predict k_1 diminishing during testing, confirming a generalization problem during model training for the One-hot encoder approach. Altogether, these results demonstrate that the rate of TMSD can be rapidly and accurately predicted using a powerful transformer model (DNA-BERT), a few local features (nucleotide availability and strong or weak) and a CNN.

To illustrate the full deep learning model framework (Model 9) end-to-end, we extracted a 1029 nucleotide sequence from the Tumor protein p53 (TP53) gene (*the complete sequence is provided in the supplementary information, Table S2*). Each sequence of 30 nucleotides in length is extracted, and the respective nucleotide availabilities for each sequence are determined using NUPACK [36], as previously described in the Materials and Methods section. The resulting list of 1000 sequences comprises the input for our deep learning model (*the complete list of sequences and nucleotide availabilities are provided in the supplementary information, Table S3*). Model 9 was then used to predict each TMSD k_1 value (*the complete results are provided in the supplementary information, Table S4*). The results are delivered in descending order with respect to predicted k_1 .

A selection of 5 sequences (Fig. 9) was analysed to illustrate how the predicted k_1 values align with the established understanding of TMSD. The fastest rates of TMSD are observed when there is a high C/G content in the 5' toehold recognition region and the absence of any secondary structures (Sequence No. 840). The introduction of A/T content and reduced nucleotide availability due to the presence of secondary structures reduces the predicted k_1 values. When the secondary structure is present in the branch (Sequence No. 118), the effect is not as detrimental as when the secondary structure blocks the 5' toehold region (Sequence No. 74). Again, this observation is aligned with the current understanding of TMSD. The corresponding KinDA simulated rate constants are also shown, which

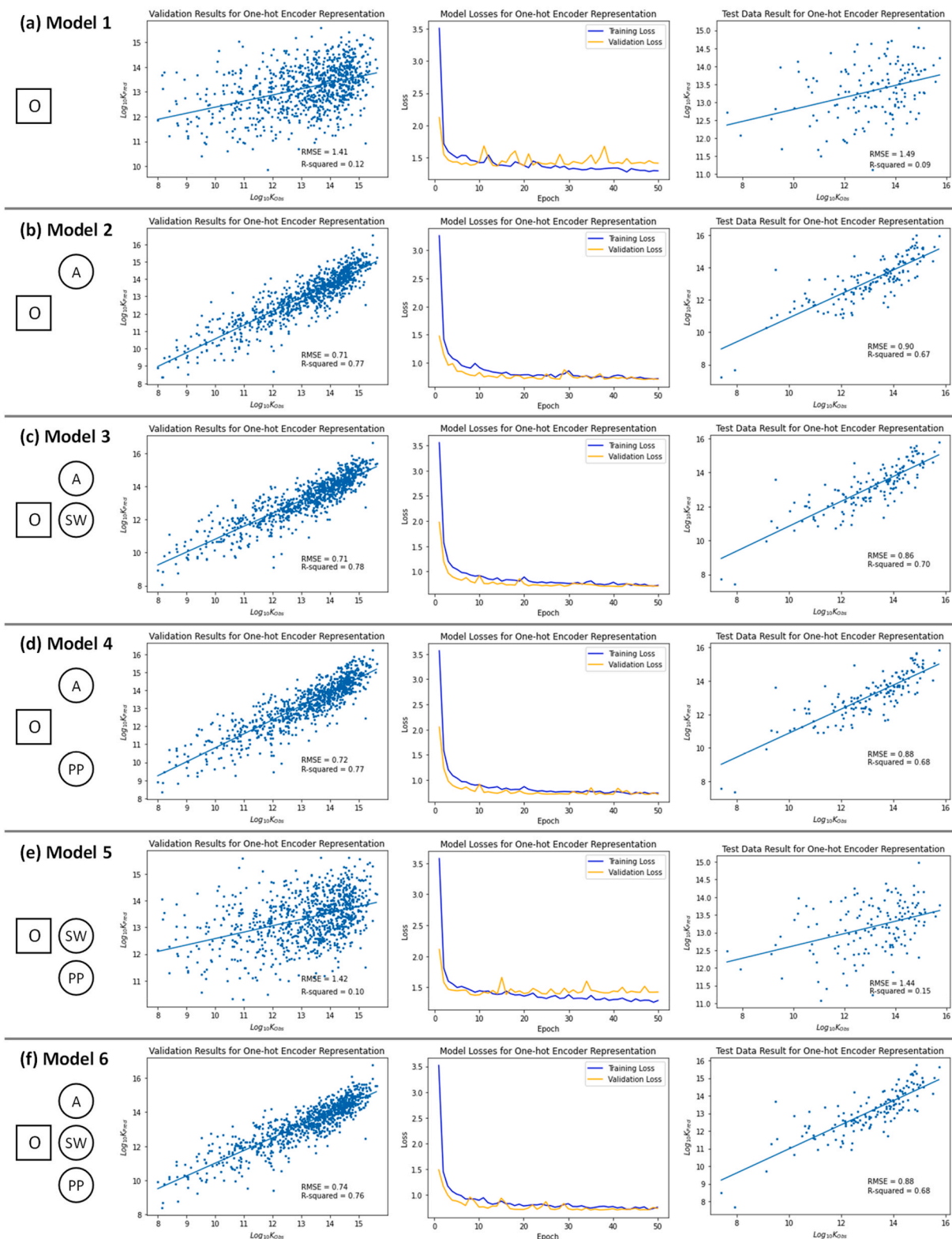


Fig. 7. (a–f). Validation, model losses and test results for One-hot encoder sequence representation (Models 1–6, a – f respectively). Scatter plots compare predicted k_1 values (K_{Pred}) from the corresponding deep learning model with simulated (K_{Obs}) values obtained *in-silico* via KinDA.

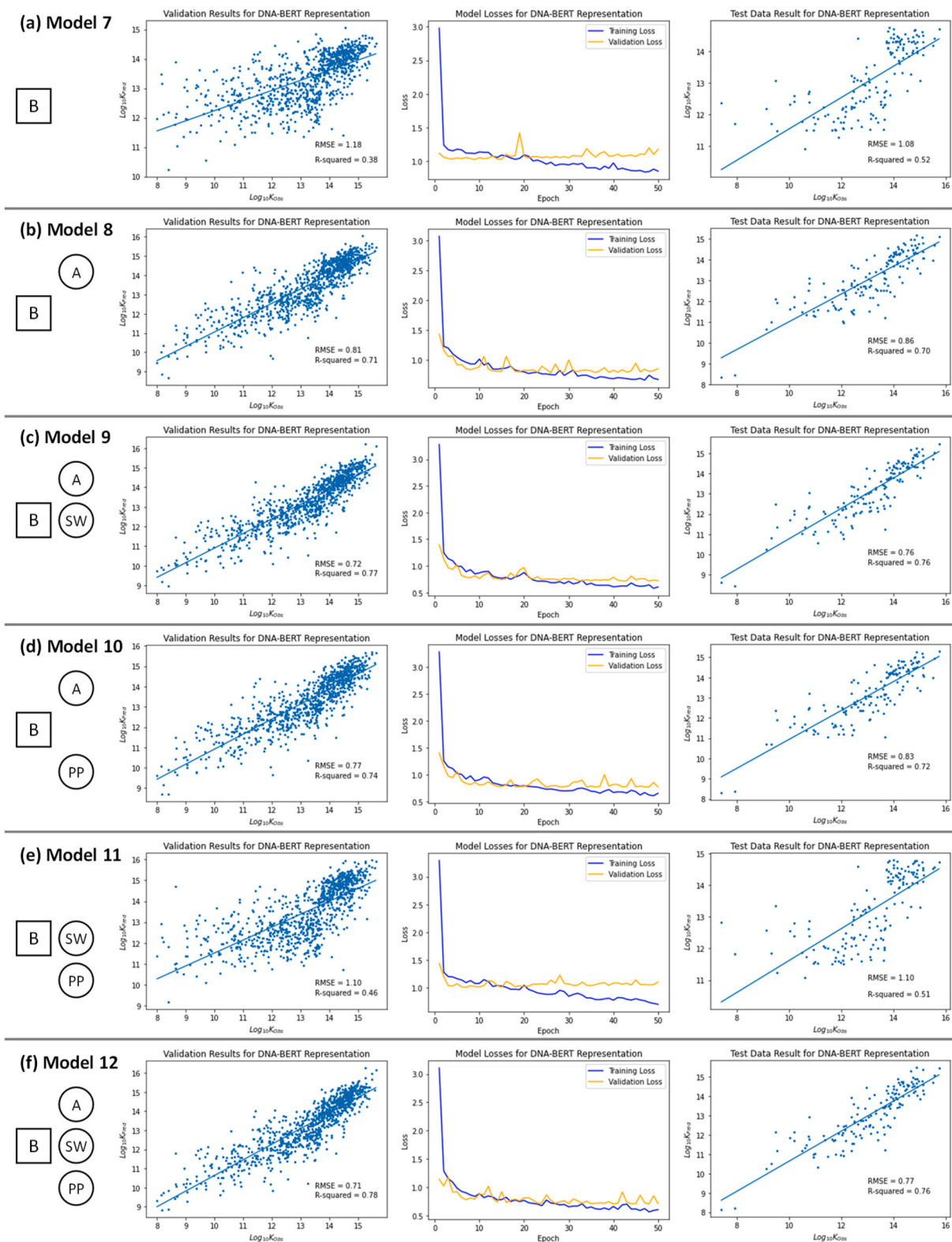

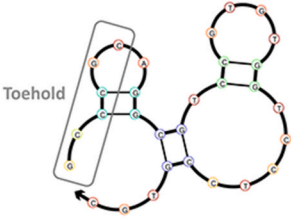

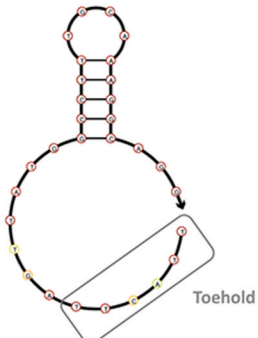
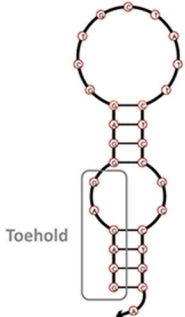
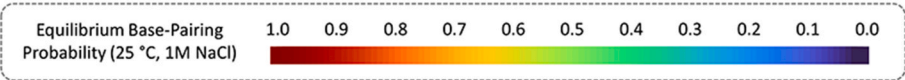


Fig. 8. (a–f). Validation results, model losses and test results for DNA-BERT sequence representation (Models 7–12, a – f respectively). Scatter plots compare predicted k_1 values (K_{Pred}) obtained by the corresponding deep learning model with simulated (K_{Obs}) values obtained *in-silico* via KinDA.

Sequence No.	Minimum Free Energy Structure	Deep Learning Model Predicted k_1 & Analysis	KinDA Predicted k_1
840		$6.11 \times 10^6 \text{ M}^{-1} \text{ S}^{-1}$ • Toehold – High C/G • Secondary Structures – None	$5.58 \times 10^6 \text{ M}^{-1} \text{ S}^{-1}$
489		$1.44 \times 10^6 \text{ M}^{-1} \text{ S}^{-1}$ • Toehold – High C/G • Secondary Structures – Moderate (Toehold and Branch)	$1.09 \times 10^6 \text{ M}^{-1} \text{ S}^{-1}$
847		$8.07 \times 10^5 \text{ M}^{-1} \text{ S}^{-1}$ • Toehold – Low C/G • Secondary Structures – None	$6.56 \times 10^5 \text{ M}^{-1} \text{ S}^{-1}$
118		$4.02 \times 10^4 \text{ M}^{-1} \text{ S}^{-1}$ • Toehold – Low C/G • Secondary Structures – High (Branch)	$1.07 \times 10^4 \text{ M}^{-1} \text{ S}^{-1}$
74		$1.19 \times 10^4 \text{ M}^{-1} \text{ S}^{-1}$ • Toehold – High C/G • Secondary Structures – High (Toehold)	$1.67 \times 10^4 \text{ M}^{-1} \text{ S}^{-1}$



Equilibrium Base-Pairing Probability (25 °C, 1M NaCl)

Fig. 9. Selection of 5 sequences (obtained from TP53 gene), corresponding minimum free energy structure (obtained via NUPACK [36–38]), deep learning predicted k_1 values using Model 9 and KinDA predicted k_1 values.

demonstrates excellent correlation with the predictions of Model 9. These results further confirm the ability of the deep learning model to accurately predict TMSD rate constants and the importance of nucleotide availability as a local feature.

4. Conclusion

This study presents a novel deep learning model capable of rapidly predicting thousands of toehold-mediated strand displacement rate constants (k_1). This is achieved by combining DNA-BERT sequence embedding representation, additional sequence local features and a convolutional neural network. These findings demonstrate the power and effectiveness of machine learning approaches in understanding and optimizing DNA nanotechnology systems. By leveraging the predictive power of machine learning, researchers can

potentially design and engineer novel DNA-based systems with greater efficiency and accuracy. One drawback to this study is the entirely *in-silico* generated data set. The advantage of this approach is the ability to generate data sets of sufficient size to train a deep learning model. To further assess the validity of these models, future work will incorporate experimentally determined kinetic rate constants within the training set and predicted k_1 values will be compared to experimentally determined values. Investigating the accuracy of deep learning models with varying toehold and branch migration lengths will also be examined to further demonstrate the potential of machine learning in the field of dynamic DNA nanotechnology. This initial study centres on rate constant predictions for a single toehold-mediated strand displacement reaction, with an emphasis on target sequence selection from a pool of potential sequences. However, deep learning models could be used to evaluate more complex systems comprised of multiple toehold-mediated strand displacement reaction pathways. This would drive forward progress in the application of dynamic DNA nanotechnology within molecular computing, data storage, signal amplification and other sophisticated DNA nanorobotic systems.

Data availability statement

Data will be made available on request.

CRedit authorship contribution statement

Ali Akay: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Hemaprakash Nanja Reddy:** Validation, Software, Resources, Methodology, Data curation. **Roma Galloway:** Supervision, Funding acquisition. **Jerzy Kozyra:** Writing – review & editing, Supervision, Funding acquisition. **Alexander W. Jackson:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ali Akay reports financial support was provided by the Erasmus+ Programme. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The project was funded in part through the Erasmus+ programme.

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.heliyon.2024.e28443>.

References

- [1] J.D. Watson, F.H.C. Crick, Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid, *Nature* 171 (1953) 737–738.
- [2] N.C. Seeman, H.F. Sleiman, DNA nanotechnology, *Nat. Rev. Mater.* 3 (2017) 17068.
- [3] S. Dey, et al., DNA origami, *Nature Reviews Methods Primers* 1 (2021) 13.
- [4] B. Yurke, A.J. Turberfield, A.P. Mills, F.C. Simmel, J.L. Neumann, A DNA-fuelled molecular machine made of DNA, *Nature* 406 (2000) 605–608.
- [5] D.Y. Zhang, G. Seelig, Dynamic DNA nanotechnology using strand-displacement reactions, *Nat. Chem.* 3 (2011) 103–113.
- [6] J. Bath, A.J. Turberfield, DNA nanomachines, *Nat. Nanotechnol.* 2 (2007) 275–284.
- [7] S. Nummelin, et al., Robotic DNA nanostructures, *ACS Synth. Biol.* 9 (2020) 1923–1940.
- [8] A.A. Green, P.A. Silver, J.J. Collins, P. Yin, Toehold switches: de-novo-designed regulators of gene expression, *Cell* 159 (2014) 925–939.
- [9] P. Yin, H.M.T. Choi, C.R. Calvert, N.A. Pierce, Programming biomolecular self-assembly pathways, *Nature* 451 (2008) 318–322.
- [10] D.Y. Zhang, A.J. Turberfield, B. Yurke, E. Winfree, Engineering entropy-driven reactions and networks catalyzed by DNA, *Science* (1979) 318 (2007) 1121–1125.
- [11] A.J. Thubagere, et al., A cargo-sorting DNA robot, *Science* (1979) 357 (2017) eaan6558.
- [12] F.C. Simmel, B. Yurke, H.R. Singh, Principles and applications of nucleic acid strand displacement reactions, *Chem Rev* 119 (2019) 6326–6369.
- [13] L. Qian, E. Winfree, J. Bruck, Neural network computation with DNA strand displacement cascades, *Nature* 475 (2011) 368–372.
- [14] T. Song, S. Garg, R. Mokhtar, H. Bui, J. Reif, Analog computation by DNA strand displacement circuits, *ACS Synth. Biol.* 5 (2016) 898–912.
- [15] M.R. Lakin, D. Stefanovic, Supervised learning in adaptive DNA strand displacement networks, *ACS Synth. Biol.* 5 (2016) 885–897.
- [16] A.A. Green, et al., Complex cellular logic computation using ribocomputing devices, *Nature* 548 (2017) 117–121.
- [17] W. Meng, et al., An autonomous molecular assembler for programmable chemical synthesis, *Nat. Chem.* 8 (2016) 542–548.
- [18] B. Shlyahovsky, et al., Spotlighting of cocaine by an autonomous aptamer-based machine, *J. Am. Chem. Soc.* 129 (2007) 3814–3815.
- [19] C. Jung, A.D. Ellington, Diagnostic applications of nucleic acid circuits, *Acc. Chem. Res.* 47 (2014) 1825–1835.
- [20] C. Zhang, et al., Cancer diagnosis with DNA molecular computation, *Nat. Nanotechnol.* 15 (2020) 709–715.
- [21] Z. Dong, X. Xue, H. Liang, J. Guan, L. Chang, DNA nanomachines for identifying cancer biomarkers in body fluids and cells, *Anal. Chem.* 93 (2021) 1855–1865.
- [22] L. Shen, P. Wang, Y. Ke, DNA nanotechnology-based biosensors and therapeutics, *Adv Healthc Mater* 10 (2021) 2002205.

- [23] J. Chen, S. Fu, C. Zhang, H. Liu, X. Su, DNA logic circuits for cancer theranostics, *Small* 18 (2022) 2108008.
- [24] D.Y. Zhang, E. Winfree, Control of DNA strand displacement kinetics using toehold exchange, *J. Am. Chem. Soc.* 131 (2009) 17303–17314.
- [25] N. Srinivas, et al., On the biophysics and kinetics of toehold-mediated DNA strand displacement, *Nucleic Acids Res.* 41 (2013) 10641–10658.
- [26] T.E. Ouldridge, A.A. Louis, J.P.K. Doye, DNA nanotweezers studied with a coarse-grained model of DNA, *Phys. Rev. Lett.* 104 (2010) 178101.
- [27] J. Berleant, et al., Automated sequence-level analysis of kinetics and thermodynamics for domain-level DNA strand-displacement systems, *J R Soc Interface* 15 (2018) 20180107.
- [28] B. Schmidt, A. Hildebrandt, Deep learning in next-generation sequencing, *Drug Discov. Today* 26 (2021) 173–180.
- [29] J.X. Zhang, et al., A deep learning model for predicting next-generation sequencing depth from DNA sequence, *Nat. Commun.* 12 (2021) 4387.
- [30] T. Mayer, L. Oesinghaus, F.C. Simmel, Toehold-mediated strand displacement in random sequence pools, *J. Am. Chem. Soc.* 145 (2023) 634–644.
- [31] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: *NAACL*, 2019.
- [32] Y. Ji, Z. Zhou, H. Liu, R.v. Davuluri, DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome, *Bioinformatics* 37 (2021) 2112–2120.
- [33] N.Q.K. Le, Q.-T. Ho, T.-T.-D. Nguyen, Y.-Y. Ou, A transformer architecture based on BERT and 2D convolutional neural network to identify DNA enhancers from sequence information, *Brief Bioinform* 22 (2021) bbab005.
- [34] N.Q.K. Le, Q.-T. Ho, V.-N. Nguyen, J.-S. Chang, BERT-Promoter: an improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection, *Comput. Biol. Chem.* 99 (2022) 107732.
- [35] N.Q.K. Le, Q.-T. Ho, Deep transformers and convolutional neural network in identifying DNA N6-methyladenine sites in cross-species genomes, *Methods* 204 (2022) 199–206.
- [36] R.M. Dirks, N.A. Pierce, An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots, *J. Comput. Chem.* 25 (2004) 1295–1304.
- [37] J.N. Zadeh, et al., NUPACK: analysis and design of nucleic acid systems, *J. Comput. Chem.* 32 (2011) 170–173.
- [38] M.E. Fornace, et al., NUPACK: analysis and design of nucleic acid structures, devices, and systems, in: *Cambridge: Cambridge Open Engage*. This Content Is a Preprint and Has Not Been Peer-Reviewed, 2022. [ChemRxiv](https://doi.org/10.26434/chemrxiv-2022-10-10).
- [39] A. Vaswani, et al., Attention is all you need, in: I. Guyon, et al. (Eds.), *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [40] S. Indolia, A.K. Goswami, S.P. Mishra, P. Asopa, Conceptual understanding of convolutional neural network- A deep learning approach, *Procedia Comput. Sci.* 132 (2018) 679–688.
- [41] G. Sateesh Babu, P. Zhao, X.-L. Li, Deep convolutional neural network based regression approach for estimation of remaining useful life, in: S.B. Navathe, et al. (Eds.), *Database Systems for Advanced Applications*, Springer International Publishing, Cham, 2016, pp. 214–228.
- [42] J. Nagi, et al., Max-pooling convolutional neural networks for vision-based hand gesture recognition, in: 2011 *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 342–347, <https://doi.org/10.1109/ICSIPA.2011.6144164>.
- [43] H. Ide, T. Kurita, Improvement of learning for CNN with ReLU activation by sparse regularization, in: 2017 *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2684–2691, <https://doi.org/10.1109/IJCNN.2017.7966185>.
- [44] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.* 15 (2014) 1929–1958.
- [45] D. Quang, X. Xie, DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences, *Nucleic Acids Res.* 44 (2016) e107.
- [46] Z. Lv, H. Ding, L. Wang, Q. Zou, A convolutional neural network using dinucleotide one-hot encoder for identifying DNA N6-methyladenine sites in the rice genome, *Neurocomputing* 422 (2021) 214–221.