Subject Section

ToxDL: Deep learning using primary structure and domain embeddings for assessing protein toxicity

Xiaoyong Pan[1,2,3,*, #], Jasper Zuallaert[2,4, #], Xi Wang[3], Hong-Bin Shen[1], Elda Posada Campos[3], Denys O. Marushchak[3], Wesley De Neve[2,4]

[1] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, 200240 Shanghai, China, [2] IDLab, ELIS, Ghent University, Ghent, Belgium, [3] BASF Belgium Coordination Center – Innovation Center Gent, Technologiepark-Zwijnaarde 101, Ghent, Belgium, [4] Center for Biotech Data Science, Ghent University Global Campus, Songdo, Incheon, South Korea.

# The two authors contributed equally to this work.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Genetically engineering food crops involves introducing proteins from other species into crop plant species or modifying already existing proteins with gene editing techniques. In addition, newly synthesized proteins can be used as therapeutic protein drugs against diseases. For both research and safety regulation purposes, being able to assess the potential toxicity of newly introduced/synthesized proteins is of high importance.

Results: In this study, we present ToxDL, a deep learning-based approach for *in silico* prediction of protein toxicity from sequence alone. ToxDL consists of (1) a module encompassing a convolutional neural network that has been designed to handle variable-length input sequences, (2) a domain2vec module for generating protein domain embeddings, and (3) an output module that classifies proteins as toxic or non-toxic, using the outputs of the two aforementioned modules. Independent test results obtained for animal proteins and cross-species transferability results obtained for bacteria proteins indicate that ToxDL outperforms traditional homology-based approaches and state-of-the-art machine learning techniques. Furthermore, through visualizations based on saliency maps, we are able to verify that the proposed network learns known toxic motifs. Moreover, the saliency maps allow for directed *in silico* modification of a sequence, thus making it possible to alter its predicted protein toxicity.

Availability: ToxDL is freely available at http://www.csbio.sjtu.edu.cn/bioinf/ToxDL/. The source code can be found at https://github.com/xypan1232/ToxDL.

Contact: 2008xypan@sjtu.edu.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Developing genetically engineered food crops involves modifying an already existing protein or introducing proteins from one species into a crop plant species, usually with the goal of improving agricultural traits such as yield, pest resistance, and herbicide tolerance (Hammond, et al., 2013). Gene editing technologies are widely used to modify crop genes. For example, CRISPR/Cas12a has been used to successfully modify a broad range of plant species (Bernabe-Orts, et al., 2019). In addition, newly synthesized proteins have emerged as therapeutic protein drugs against diseases (Vlieghe, et al., 2010). For both research and regulation purposes, being able to assess the potential allergenicity and toxicity of a newly introduced or gene-edited protein is of high importance, so to ensure food, drug, and environmental safety.

To date, the specific determinants for pathogenic effects are still unknown in many proteins. Indeed, experimental methods based on animal trials to verify protein toxicity are time consuming and costly. Moreover, animal trials are of limited value: due to inter-species differences and different disease models, the results obtained typically offer little guidance to human toxicity reactions (Mumtaz and Pohl, 2012). An *in silico* method to assess protein toxicity effects in animals or humans could narrow down sets of candidate proteins that need to be experimentally validated, thus preventing expenditure of resources on unviable candidate proteins.

Traditionally, potential toxicity is computationally determined by using sequence similarity tools, such as BLAST (Altschul, et al., 1997), inferring protein toxicity from homologous sequences (Negi, et al., 2017). However, this strategy has a number of limitations:

- The protein of interest is required to have homologous toxic proteins.
- Global sequence similarity is used, despite the fact that protein toxicity is mainly determined by local domain sequences (Negi, et al., 2017).
- An arbitrary sequence similarity or e-value cutoff is required.

Machine learning models can be used to predict protein toxicity with a high accuracy, also making it possible to identify biological knowledge related to toxicity. For example, ToxinPred (Gupta, et al., 2013) leverages a support vector machine (SVM) to classify toxic peptides using features derived from various peptide properties. ClanTox, on the other hand, analyzes short animal toxins by making use of boosted classifiers that take as input 545-D sequence-derived features (Naamati, et al., 2009).

Over the past few years, deep learning has achieved remarkable results in various domains, including computer vision and genomics (Eraslan, et al., 2019; Pan, et al., 2018). Convolutional neural networks (CNNs), in particular, have proven to be highly effective when dealing with spatial locality in data (Lecun, et al., 1998). Neural networks consist of multiple layers of abstraction, allowing for the automatic learning of both low- and high-level features from a vast amount of training data. This bypasses the need for manual feature engineering using human expertise. Deep learning has already been applied for predicting drug and environmental chemical toxicity, hereby outperforming other computational models (Klambauer, et al., 2017). In addition, deep learning has been demonstrated to be powerful in distinguishing venom proteins from non-venom proteins (Cole and Brewer, 2019).

Many proteins with experimentally verified toxic effects have already been collected, for instance by the Animal Toxin Annotation Project (Jungo, et al., 2012) in UniProt, where 6,164 animal toxin proteins from different species have been reviewed and curated to date, and by the concerted effort presented in (Negi, et al., 2017), where curated toxic proteins have been clustered to identify groups similar to protein families in the Pfam database (El-Gebali, et al., 2019), so to be able to detect a functional sequence signature for toxic proteins.

To represent proteins with more information than just sequence, functional domain information can be attached. Indeed, domains co-occurring in proteins tend to have more similar functionality than domains occurring in separate proteins (Menichelli, et al., 2018). Domain information for predicting protein function has been used before. GOLabeler (You, et al., 2018) for instance integrates 33,879 binary features to predict gene ontology terms, representing the presence of a large number of domains with other component models. However, GOLabeler assumes that the distance between any two domains is the same. This is unrealistic, as co-occurring domains should enjoy a higher similarity, since most protein domains tend to appear with a limited number of other domains on the same protein. Thus, a better representation of domains might be beneficial. In addition, the use of high-dimensional one-hot encoded domain features may lead to model overfitting, especially when the number of training examples is much smaller than the number of features.

In this study, we present ToxDL, a deep learning-based approach that is effective in distinguishing between toxic and non-toxic proteins using both sequence information and protein domain knowledge directly derived from that sequence. There are three main modules in ToxDL. The first module builds a CNN on top of the sequence, encoded as a one-hot matrix. The second module consists of domain2vec, which generates protein domain embeddings using a Skip-gram model (Mikolov, et al., 2013). After concatenating the outputs of those two modules, a third module, consisting of a fully connected layer and an output layer, generates a toxicity probability. Considering that protein sequences have variable lengths, but that neural networks typically work with a fixed input size, we evaluate several techniques to enable ToxDL to handle this discrepancy. Additionally, we infer toxicity motifs from the trained CNN models, leveraging attribution methods to highlight the local subsequence(s) that contribute(s) to the protein toxicity prediction.

2 Methods

In this section, we first describe the datasets constructed for benchmarking purposes. We then introduce domain2vec for learning protein domain embeddings. Next, we detail the network architecture of ToxDL. Finally, we describe how to infer toxic domains using the trained models.

2.1 Benchmark dataset construction

Toxic proteins were downloaded from the Animal Toxin Annotation Project (Jungo, et al., 2012) in UniProt, where 6,164 animal proteins have been annotated as toxic (May 2019) (that is, the aforementioned proteins have been annotated with the function *toxin activity*). All of these reviewed toxic proteins were used as positive samples. From the animal species studied in the aforementioned project, we randomly extracted 6,164 reviewed proteins and 903 venom proteins that have not been annotated as toxic in UniProt, using these 7,067 proteins as negative samples.

For the creation of a training set, about 80% of the obtained toxic and non-toxic proteins were selected in a random way, and the remaining proteins were used for the creation of a validation set and an independent test set. To reduce the impact of sequence similarity, we used cd-hit-2d (Fu, et al., 2012) to remove sequences from the test set with a sequence similarity of at least 40% with any sequence in the training set (40% is the minimum value for cd-hit-2d). Furthermore, we held out 10% of the original training set as a validation set, from which homologous sequences were removed in the same fashion. That way, we obtained an initial validation set of 309 non-toxic proteins and 25 toxic proteins, and an initial test set of 754 non-toxic proteins and 59 toxic proteins.

To further reduce the sequence homology, we used Pfam clans (El-Gebali, et al., 2019) to ensure the absence of proteins with domains from the same Pfam clans between the above sets. To do so, we first downloaded the Pfam clans from ftp://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-

A.clans.tsv.gz, subsequently recording all Pfam clans in the training set. We then removed proteins in the validation set and the test set with domains belonging to the clans in the training set. After doing so, we obtained a test set of 59 toxic proteins and 670 non-toxic proteins, and a validation set of 25 toxic proteins and 277 non-toxic proteins, as shown in Table 1. In this context, we did not remove similar proteins in the training set: such proteins were used as augmented samples, thus making it possible to obtain more training samples, a strategy that has shown to be highly effective for training deep learning models (Cui, et al., 2014).

In order to test the cross-species transferability of our approach towards protein toxicity prediction, we constructed a strictly independent test set only consisting of bacteria proteins. The bacteria test set can be directly downloaded from BTXpred (Saha and Raghava, 2007), containing 183 toxic proteins and 500 non-toxic proteins. Furthermore, we used cd-hit-2d to remove sequences from the bacteria test set that are similar to sequences in the entire animal set, applying a similarity cutoff of 40%.

A summary of the different datasets used for benchmarking can be found in Table 1. Following the removal of homologous sequences, a further imbalance between the number of positives and the number of negatives is introduced, which makes for a more realistic setting.

Table 1. Overview of the different datasets used for benchmarking.

Dataset	# of positives	# of negatives
Training set (animal proteins)	4,413	5,671
Validation set (animal proteins)	25	277
Test set (animal proteins)	59	670
Test set (bacteria proteins)	180	382

2.2 Protein domains

To integrate information about protein domains into ToxDL, we downloaded all UniProt protein domains generated by InterProScan (Jones, et al., 2014), providing us with 126,780,787 proteins and 36,713 domains. These were used for training a Skip-gram model to automatically learn protein domain embeddings.

We considered domains with *toxin* or *toxic* in their name as toxic. The resulting 269 domains, which can be downloaded from www.csbio.sjtu.edu.cn/bioinf/ToxDL/Data.htm, were used to check whether they can be located in the vicinity of each other in the embedding space generated by domain2vec.

Furthermore, we downloaded the HMM models from Pfam v32.0 (El-Gebali, et al., 2019) for use in our different baseline methods. In total, we obtained 34,353,433 Pfam domains, with 75 of these Pfam domains having *toxin* or *toxic* in their name.

2.3 domain2vec

Most often, one protein contains multiple associated domains that determine its function. Some domains may frequently co-occur to function together. For example, the PAZ and PIWI domains are often found together (Tahbaz, et al., 2004). These co-occurring domains should be similar in the embedding space (that is, they should be close to each other). However, when using a traditional one-hot vector to represent individual domain presence or absence, the distance between all domains is equal. To obtain a more in-depth representation of protein domains, we trained a Skip-gram model to learn domain embeddings, representing each domain by a vector of continuous values.

Given is a set of proteins, with each protein having a number of domains $d_1, d_2 ..., d_n$. Here, we treat each domain as a word, each protein as a sentence, and all UniProt proteins as the corpus. The Skip-gram model trains embeddings based on the co-occurrence of domains within a context window, by maximizing the following objective function:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-s \le j \le s,\, j \neq 0} \log p(d_{t+j} \mid d_j; \theta) \quad (1)$$

In the expression above, N denotes the number of proteins, s is the context window size, and θ are the weights of the model. We would like to refer the interested reader to (Mikolov, et al., 2013) for more details about the above objective function.

Once training is finished, the final embedding vector that corresponds to a given protein is calculated by averaging the embeddings of all domains found in this protein.

2.4 Network architecture

The CNN module of ToxDL takes a one-hot encoded protein sequence as input, and subsequently performs convolutional, dropout, and max pooling operations. Next, we make use of a specialized layer to deal with the variable length of the input sequences. As discussed below in more detail, we explored five different approaches to do so. After concatenating the output of the CNN module with the averaged domain embedding vector, we transfer the resulting vector to the output component, which consists of a fully connected layer, a dropout layer, and a softmax output unit.

The five approaches to deal with variable input lengths are as follows:

1. **Zero-padding only:** In this naïve approach, zero-padding is performed on the input layer to reach a fixed input length of 1,002 positions. No extra layer is added to reduce the output size of the CNN module, meaning that the outputs of the last max pooling layer are flattened and directly connected to the output component. As a result, the number of parameters in this architecture will be greater than that of the following approaches.
2. **Global max pooling:** A global max pooling operation is performed by pooling over the full width of the previous layer.
3. **Gated Recurrent Unit (GRU):** The outputs of the last max pooling layer are fed into a bidirectional GRU. The final hidden states for both directions are concatenated and used as input for the output unit.
4. **Dynamic max pooling:** A max pooling layer is added after the previous one, with a dynamic pool size and stride. The pool size and stride depend on the length of the input sequence, with longer sequences resulting in a higher pool size and stride. They are chosen in such a way that the output of the layer always results in the same number of output positions. Specifically, the stride is chosen to be half of the pool size, yielding overlapping windows. For each channel, the outputs after dynamic max pooling are calculated as follows:

   $$dyn_x(i) = \max\left(x_{i\frac{w}{2}}, \ldots, x_{\min\left(l,(i+2)\frac{w}{2}\right)}\right), \quad (2)$$

   with $w = \left\lfloor \frac{2l}{n+1} \right\rfloor$,

   and where $dyn_x(i)$ denotes the output at position i after dynamic max pooling, n the number of resulting outputs after dynamic max pooling, x_p the activation output on position p from the previous layer, and l the length of that activation. An illustration of this pooling strategy can be found in Fig. 1A.
5. **k-max pooling:** A dynamic k-max pooling layer, as described in (Kalchbrenner, et al., 2014), is added after the last max pooling layer, resulting in a fixed output size. Instead of keeping one value after max pooling, dynamic k-max pooling collects the k highest

activations in each channel in the same order of occurrence. An illustration of this pooling strategy can be found in Fig. 1B.

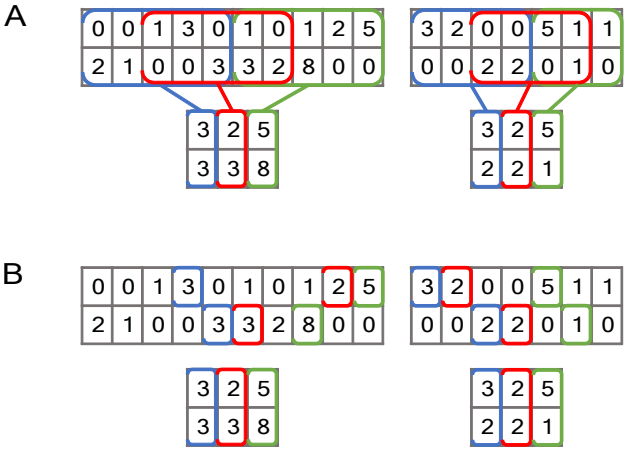

Fig. 1 (A) An illustration of dynamic max pooling, given two activation maps of length 11 and 7, respectively, with both activation maps using two channels. The chosen output size in this example is 3. (B) An illustration of k-max pooling, given two activation maps of length 11 and 7, respectively, with both activation maps using two channels, for $k = 3$.

The motivation behind the last three approaches is to be able to reduce the output size of the preceding layer to a fixed size, whilst retaining a notion of spatial information on where certain features were detected. When using the zero-padding only approach, we do not reduce the output size, hence including activations that are the result of zero-padded positions in the input. When using global max pooling, all spatial information is lost. For the GRU approach, the spatial information is handled implicitly in the recurrent unit, while dynamic max pooling and k-max pooling explicitly retain this information in their output.

2.5 ToxDL

ToxDL is a multi-modal deep learning-based approach for predicting protein toxicity. Given a protein, it concatenates the output from a CNN module with the average embeddings of all domains found in this protein, subsequently feeding the vector obtained into an output component that generates a toxicity probability. A flowchart characterizing ToxDL is depicted in Fig. 2.

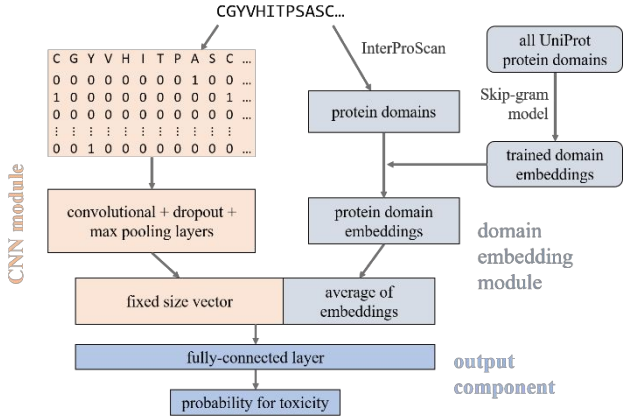

Fig. 2 Flowchart characterizing ToxDL and its three components. First, we have a CNN module that takes one-hot encoded sequences as input. In parallel, we have the module domain2vec for generating protein domain embeddings. Next, we have the output component, taking the concatenated output of the previous components as input, feeding it to a fully connected layer and subsequently generating a toxicity probability using a softmax output layer.

2.6 Identifying toxic domains using saliency maps

To gain insight into the reasoning process of our trained models, we generated saliency maps using the Integrated Gradients approach, as described in (Sundararajan, et al., 2017). This technique produces a proclaimed contribution score for each amino acid in an input sequence with respect to the output in the network. The magnitude of a contribution score indicates the importance of the amino acid, with positive values steering the network towards a positive prediction, and negative values steering the network towards a negative prediction.

As the aforementioned technique requires a reference input, we constructed an 'average' reference input sequence, based on the suggestion made in (Shrikumar, et al., 2017). There, the authors propose to construct a reference by taking the amino acid frequency on each position, for all negative samples in the training set. However, given the variable input size of protein sequences, this scheme requires adaptation. Specifically, we calculated the average amino acid distribution for the first and last five positions on each position, and for the remainder, we calculated the average of all other positions. This approach is illustrated in Fig. 3.

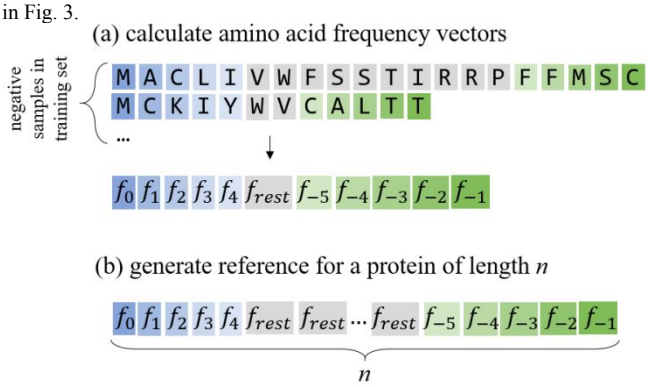

Fig. 3 The calculation of the reference input. (a) For each of the first five positions and for each of the last five positions, the amino acid frequency is calculated. Finally, the frequency is calculated for the remaining amino acids. (b) A reference input is then constructed as depicted, using the corresponding frequency vectors on the first five and the last five positions, and using f_{rest} on the remaining positions.

2.7 MEME and TOMTOM for detecting toxic protein motifs

MEME (Bailey, et al., 2015) applies expectation maximization to fit a mixture model to a set of sequences, finding one or more motifs. We used MEME to generate 120 sequence motifs with length 10 from toxic animal proteins and toxic bacteria proteins, respectively. To compare the similarities between motifs found in toxic animal proteins and motifs found in toxic bacteria proteins, we used TOMTOM (Gupta, et al., 2007), relying on an e-value of 0.05.

2.8 Baseline methods

In this study, we compare ToxDL with various baseline methods, including BLAST-based methods, InterProScan, hmmsearch, and machine learning methods. We also perform a comparative analysis of different ToxDL variants.

1. **BLAST:** A protein sequence that needs to be tested for toxicity is used as a query against the training sequences using BLAST. If it has any similar toxic sequence in the training set with e-value < 0.001, the given protein sequence is considered to be toxic. Otherwise, the protein sequence at hand is considered to be non-toxic.
2. **BLAST-score:** Given a protein sequence that needs to be tested for toxicity, BLAST-score first obtains a set E of similar proteins from the training set, with e-value < 0.001. It then calculates a score for the given protein sequence as follows:

$$\text{score (q)} = \frac{\sum_{s \in E} \text{bitscore(q, s)} * \text{I(s is a toxic protein)}}{\sum_{s \in E} \text{bitscore(q, s)}} \quad (3)$$

where bitscore is a normalized alignment score from the BLAST output, measuring sequence similarity independent from the query sequence length and the database size (the size of the training set), q is the query sequence, s is a similar sequence in the set E, and I is the indicator function.

3. **InterProScan:** Of the 269 toxic domains, the domains that are also in the training set are kept, leading to a set of 72 retained toxic domains. Given a protein that needs to be tested for toxicity, if it has a domain belonging to the aforementioned set of 72 retained toxic domains, then this protein is considered to be toxic. Otherwise, this protein is considered to be non-toxic.
4. **hmmsearch:** As described in HMMER (Potter, et al., 2018), we run hmmsearch against the test sequences to search for Pfam HMM models with an e-value < 1e-5. If a given protein contains one of the 75 toxic Pfam domains, then it is considered to be toxic. Otherwise, it is considered to be non-toxic.
5. **ToxinPred:** ToxinPred is a traditional machine learning method that uses amino acid and dipeptide composition as input features. In this study, we use the features of ToxinPred to train both an SVM and a random forest (RF), from here onwards labeled ToxinPred-SVM and ToxinPred-RF, respectively. Appropriate hyperparameters are selected using a grid search on the training set with 3-fold cross-validation. For constructing ToxinPred-SVM, the grid search covered both a linear kernel and an RBF kernel, with [1, 2, 4, 6, 10] as possible values for C and with [0.5, 1.0, 2.0, 6.0] as possible values for gamma. For constructing ToxinPred-RF, the grid search covered the values [20, 50, 100] as the number of trees.
6. **ClanTox:** ClanTox is a meta-classifier consisting of ten boosted decision stumps for distinguishing short toxin and toxin-alike proteins from non-toxic proteins using 545-D sequence-derived features. The final score is defined as the mean of the scores from the ten classifiers. We uploaded our test sequences to the ClanTox webserver, with this webserver generating detailed mean scores and labels for the uploaded test sequences.
7. **TOXIFY**: TOXIFY is a recurrent neural network-based method for classifying venom proteins using a numerical representation derived from protein sequences. In this study, we use the TOXIFY model that can be found at https://github.com/tijeco/toxify to predict probability scores for our test sequences. However, this trained TOXIFY model can only predict scores for proteins shorter than 500 amino acids. Thus, whenever necessary, we split a protein sequence into different parts (e.g., 1-500, 400-900, 800-1,300, and so on), using an overlap of 100 amino acids. We then leveraged TOXIFY to predict scores for each part, using the maximum score obtained as the toxicity score for the given protein sequence. Of the 729 proteins available in the animal test set, 241 proteins have a length greater than 500. Of the 562 proteins available in the bacteria test set, 235 proteins have a length greater than 500.

Variants of ToxDL

ToxDL consists of a CNN module and a domain2vec module, with the two aforementioned modules coming before the output module. We designed a number of variants of ToxDL, using either a single module or different modules.

8. **ToxDL-ODE:** This variant only uses the 256-D protein domain embeddings as its input, which is then directly connected to the output component.
9. **ToxDL-CNN:** This variant only uses the CNN module.
10. **ToxDL-One:** Instead of using learned embeddings for representing protein domains, we use a one-hot encoding for the 269 toxic protein domains. Specifically, each protein is represented using a 269-D binary vector, with a one indicating the presence of a particular domain. This vector is directly fed to the output component.
11. **ToxDL-OD:** For this variant, the one-hot encoded vectors for the 269 toxic protein domains, as described for the ToxDL-One variant, are concatenated with the output of the CNN module. This combination is then fed to the output component.

2.9 Evaluation metrics

In this study, our experimental analysis is performed using four metrics: the F1 score, the Matthews correlation coefficient (MCC), the area under the receiver operating characteristic curve (auROC), and the area under the precision-recall curve (auPRC). For the baseline methods BLAST, InterProScan, and hmmsearch, we only calculate the F1 score and the MCC, given that these methods only output a binary value.

Table 2. Details of the network architecture used by ToxDL.

	Layer	Details	Output size
NN module	one-hot encoded input sequence	zero padded (for practical implementation reasons)	(1002, 20)
	conv layer (+ReLU)	200 filters of size 9	(1002, 200)
	dropout layer	$p = 0.5$	(1002, 200)
	max pooling layer	pool size 3, stride 3	(334, 200)
	conv layer (+ReLU)	200 filters of size 7	(334, 200)
	dropout layer	$p = 0.5$	(334, 200)
	max pooling layer	pool size 3, stride 3	(111, 200)
	conv layer (+ReLU)	200 filters of size 7	(111, 200)
	dropout layer	$p = 0.5$	(111, 200)
	max pooling layer	pool size 3, stride 3	(37, 200)
	Five approaches, choose one		
	zero-padding	no extra layer present	(37, 200)
	global max pooling	pool size 37	(200)
	GRU	hidden state size 256	(512)
	dynamic max pooling	10 outputs per channel	(10, 200)
	k-max pooling	$k = 10$	(10,200)
domain embedding module	average domain embedding as input	embedding size 256	(256)
output component	concatenated layer	output size depends on the approach chosen	(x + 256)
	fully connected layer	64 neurons	(64)
	sigmoid output unit	1 neuron	(1)

2.10 Experimental settings

Protein domain embeddings are learned by training a Skip-gram model for 10 epochs with a context window of 5 and an embedding size of 256. A context window of 5 was chosen since the number of proteins with five domains is the highest.

For the CNN module, we truncated all input sequences to a maximum length of 1002, for practical implementation reasons. The hyperparameters for the convolutional, max pooling, and dropout layers of the network are listed in Table 2; these hyperparameters were optimized using an independent dataset constructed for the gene ontology (GO) term prediction task of the CAFA2 challenge. These hyperparameters can thus also be applied to other protein function prediction tasks using variable-length protein sequences. We used the Adam optimizer with a learning rate of 0.001 to optimize the categorical cross-entropy cost function. Training lasted for 10 epochs, finally retaining the model with the lowest validation loss for performing an evaluation on our test sets.

For each method, the average effectiveness over 10 experiments was calculated. The different architectures tested used the five different approaches to deal with variable input lengths, as well as a combination of dynamic max pooling and *k*-max pooling, where the results of both were simply concatenated. The latter approach was used when determining the ToxDL results presented below.

3 Results

In this study, we first evaluate the effectiveness of ToxDL on the independent animal protein test set. Next, we report the cross-species effectiveness of ToxDL in order to evaluate its generalizability to other species. We also compare the effectiveness of different ToxDL network architectures. Lastly, given the trained CNN model, we investigate to what extent we are able to infer toxic domains and motifs that align well with experimentally verified toxic domains and motifs available in public databases.

3.1 The effectiveness of ToxDL

In this experiment, we perform a comparative investigation of the effectiveness of ToxDL on the animal test set. The results obtained can be found in Table 3. Despite the removal of homologous sequences from the test set, BLAST obtains the highest F1 score among the different baseline approaches (that is, the BLAST-based approaches, InterProScan, and hmmsearch). When compared to TOXIFY, BLAST achieves an F1 score of 0.800 and an MCC of 0.801, yielding a relative improvement over TOXIFY of 11.9% and 16.1%, respectively. When doing a comparison in terms of auPRC, BLAST-score yields an auPRC of 0.818, which is higher than the auPRC obtained by TOXIFY, ClanTox, ToxinPred-SVM, and ToxinPred-RF. However, ToxinPred-RF does have a higher auROC of 0.948, where BLAST-score only reaches an auROC of 0.868.

When comparing our final ToxDL architecture to the best results obtained by ToxinPred-RF and BLAST-score, we can see that ToxDL improves upon the auROC of ToxinPred-RF from 0.948 to 0.989 (a relative improvement of 4.3%), and upon the auPRC of BLAST-score from 0.818 to 0.913 (a relative improvement of 11.6%). At a cutoff of 0.5, ToxDL reaches an F1 score of 0.809, outperforming all other approaches. However, again at a cutoff of 0.5, the MCC of 0.793 obtained by ToxDL is slightly lower than the MCC of 0.801 obtained by BLAST (with the latter only generating binary output). On the other hand, when scanning for the optimal cutoff threshold, we find that the highest F1 score and the highest MCC obtained by ToxDL is 0.870 and 0.864, respectively. We can thus conclude that ToxDL is the most effective approach across all metrics. In addition, we compare ToxDL with TOXIFY on the subset consisting of 488 proteins with a size shorter than 500. As shown in Supplementary Table S1, we find ToxDL to be superior to TOXIFY.

Table 3. The effectiveness of ToxDL and our baseline methods on the animal protein test set. For the deep learning-based methods, we report the average after having performed each experiment ten times, along with the standard deviation. ClanTox uses its default mean score of -0.02 as the threshold for toxic protein classification.

Method	F1 score (threshold = 0.5)	MCC (threshold = 0.5)	auROC	auPRC
BLAST	0.800	**0.801**	-	-
BLAST-score	0.789	0.775	0.868	0.818
InterProScan	0.347	0.402	-	-
hmmsearch	0.185	0.307	-	-
ClanTox	0.620	0.604	0.903	0.612
TOXIFY	0.715	0.690	0.930	0.743
ToxinPred-RF	0.667	0.638	0.948	0.716
ToxinPred-SVM	0.677	0.648	0.938	0.712
ToxDL-One	0.356 (±0.000)	0.435 (±0.000)	0.609 (±0.001)	0.572 (±0.002)
ToxDL-OD	0.769 (±0.029)	0.749 (±0.032)	0.977 (±0.004)	0.852 (±0.024)
ToxDL-ODE	0.599 (±0.007)	0.599 (±0.009)	0.954 (±0.002)	0.648 (±0.014)
ToxDL-CNN	0.761 (±0.030)	0.743 (±0.033)	0.978 (±0.004)	0.846 (±0.020)
ToxDL	**0.809 (±0.022)**	0.793 (±0.024)	**0.989 (±0.002)**	**0.913 (±0.014)**

When comparing the different ToxDL variants, we can observe that the use of protein domain embeddings is indeed beneficial. ToxDL-ODE only uses protein domain embeddings, and it outperforms ToxDL-One, which only uses a one-hot representation for the protein domains, improving the auROC from 0.609 to 0.954 (a relative increase of 56.7%) and the auPRC from 0.572 to 0.648 (a relative increase of 13.3%). This is also confirmed when evaluating the aforementioned models with the CNN module added. ToxDL (which uses protein domain embeddings) outperforms ToxDL-OD (which uses a one-hot representation for the protein domains), by improving the auROC from 0.977 to 0.989 and the auPRC from 0.852 to 0.913. Finally, we can also observe that ToxDL-CNN already achieves a high effectiveness, with an auROC of 0.978 and an auPRC of 0.846. However, adding the protein domain embeddings to this CNN-only architecture makes it possible to increase the auROC to 0.989 and the auPRC to 0.913 (relative improvements of 1.1% and 7.9%, respectively). We can thus conclude that including protein domain embeddings indeed

boosts the prediction effectiveness. Note that the output generated by all approaches can be found in the supplementary data.

To better understand the benefits of using the CNN and protein domain embedding modules in parallel, we investigate the sequence length of toxic proteins that are correctly predicted by ToxDL-CNN and ToxDL-ODE, respectively. The average length of toxic proteins correctly predicted by ToxDL-CNN is 102.4 (±5.2) amino acids, which is 15.8% shorter than the average length of 118.6 (±4.4) amino acids for ToxDL-ODE. This suggests that the CNN module is able to better handle shorter protein sequences than the protein domain embeddings module, as InterProScan often struggles to find relevant domains in shorter protein sequences.

Table 4. The cross-species effectiveness of ToxDL and our baseline methods, using the animal protein training and validation sets, as obtained for the bacteria protein test set. For the deep learning-based methods, we report the average after having performed each experiment ten times, along with the standard deviation. ClanTox uses its default mean score of -0.02 as the threshold for toxic protein classification.

Method	F1 score (threshold = 0.5)	MCC (threshold = 0.5)	auROC	auPRC
BLAST	0.000	-0.128	-	-
BLAST-score	0.000	-0.224	0.575	0.160
InterProScan	0.011	0.062	-	-
hmmsearch	0.000	0.000	-	-
ClanTox	0.022	0.054	0.654	0.274
TOXIFY	0.128	0.108	0.721	0.506
ToxinPred-RF	0.090	0.061	0.545	0.351
ToxinPred-SVM	0.123	0.051	0.612	0.410
ToxDL-One	0.356 (±0.000)	0.398 (±0.000)	0.608 (±0.000)	**0.734 (±0.000)**
ToxDL-OD	0.126 (±0.025)	0.073 (±0.043)	0.625 (±0.026)	0.417 (±0.012)
ToxDL-ODE	**0.604 (±0.012)**	0.408 (±0.019)	0.782 (±0.010)	0.418 (±0.024)
ToxDL-CNN	0.110 (±0.023)	0.048 (±0.043)	0.622 (±0.038)	0.415 (±0.025)
ToxDL	0.097 (±0.022)	0.022 (±0.028)	**0.786 (±0.021)**	0.525 (±0.030)

3.2 Cross-species effectiveness of ToxDL

To evaluate the transferability of ToxDL, the models that we trained on animal proteins were leveraged to predict toxicity for bacteria proteins. The results obtained are shown in Table 4. BLAST-score no longer outperforms the other baseline methods, due to the heterogeneity between the training set and the test set. In particular, BLAST-score has an F1 score of 0.000 for a cutoff of 0.5, an auROC of 0.575, and an auPRC of 0.160. The highest F1 score among the different baseline methods used is achieved by TOXIFY (0.128). TOXIFY also yields the highest auROC and auPRC among the different baseline methods used: 0.721 and 0.506, respectively.

When comparing the different architectures of ToxDL, we can observe that the CNN-only model (ToxDL-CNN) struggles to perform well, as it reaches an auROC of 0.622 and an auPRC of 0.415. Here, the predetermined protein domain knowledge proves crucial, as the one-hot protein domain representation (ToxDL-One) already improves the auPRC to 0.734, making for a relative improvement of 76.9%, while the auROC is only slightly lower (0.608). Furthermore, the use of protein domain embeddings no longer outperforms the use of a one-hot protein domain representation on all metrics, as the auPRC drops from 0.734 to 0.418, although the auROC increases from 0.608 to 0.782. Additionally, adding the CNN module to the one-hot protein domain representation no longer increases the effectiveness, although this is still the case when adding it to the protein domain embeddings.

Given the above-mentioned inconsistency between the auROC and auPRC values obtained by ToxDL-One, compared to the values obtained by the other approaches, we perform a detailed check of the output generated by ToxDL-One, finding that ToxDL-One outputs the same probability score for all non-toxic proteins and most toxic proteins in the test set. One potential reason for doing so is that the animal and bacteria toxic proteins almost have no domains in common with the 269 toxic domains identified by InterProScan (see Fig. 4B). Indeed, most of the one-hot encodings of the animal and bacteria toxic proteins do not have values of one in their respective vectors, implying that most proteins (both positives and negatives) are represented by a vector that only consists of zeros. As a result, ToxDL-One cannot learn patterns that are truly related to protein toxicity, which is supported by the low auROC of 0.608. Furthermore, this is also supported by the standard deviations, which are equal to zero for both the auROC and the auPRC. In addition, when calculating the entropy of the predicted output values, we find that the entropy of ToxDL and TOXIFY is 0.998 and 0.994, respectively. However, the entropy of ToxDL-One is 0.067, which is close to 0, indicating that the predicted values of ToxDL-One are almost the same for all test proteins.

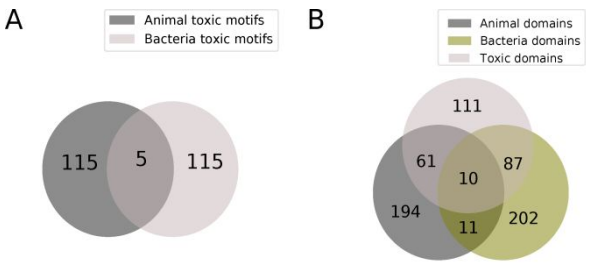

Fig. 4 Common (A) motifs and (B) toxic protein domains between the animal and bacteria protein test sets.

Considering that an acceptable classifier should have an auROC > 0.7 (Mandrekar, 2010), ToxDL, ToxDL-ODE, and TOXIFY are the only three approaches that meet this criterion. In addition, ToxDL has a higher auPRC (0.527) than both ToxDL-ODE (0.418) and TOXIFY (0.506). As mentioned above, 0.5 is not necessarily the optimal cut-off threshold for calculating the F1score. Over all possible threshold values, ToxDL yields a maximum F1 score of 0.688 and a maximum MCC of 0.529, which are

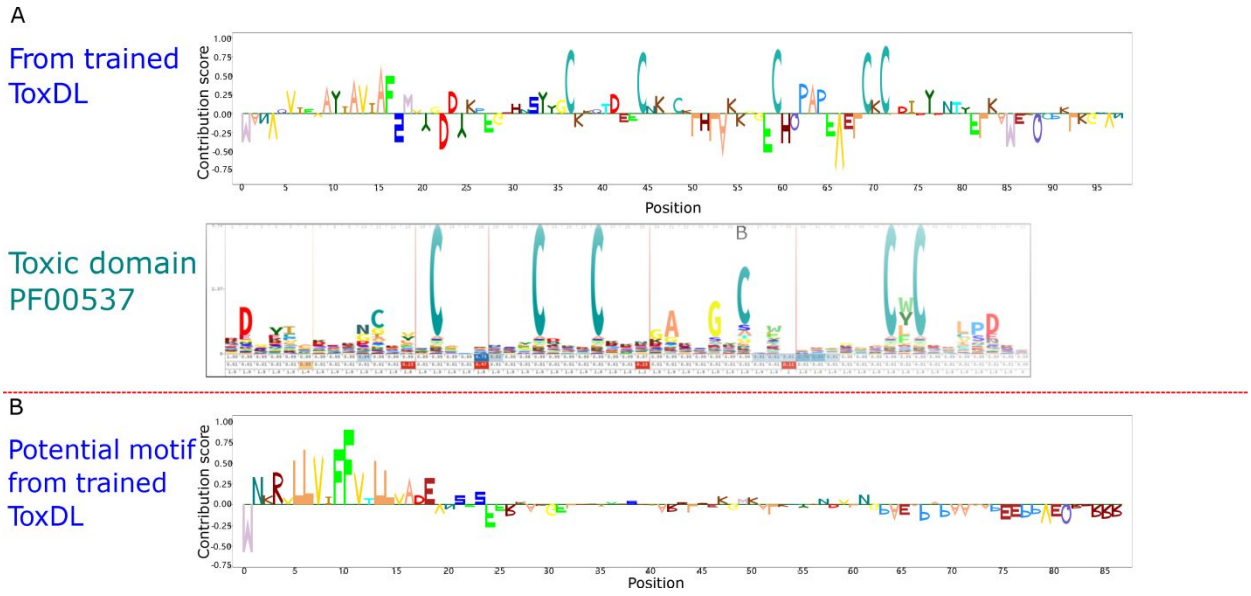

Fig. 6 A number of protein domains detected by ToxDL. The motif shown in (A) is observed to have a high correspondence with the verified protein domain PF00537 in the Pfam database, whereas the motif shown in (B) is newly found by ToxDL

higher than the respective values obtained by the different methods used. Our results indicate that ToxDL is the best choice for cross-species toxicity prediction when considering both auROC and auPRC. Note that the output generated by all methods can be found in the supplementary data.

The cross-species effectiveness of ToxDL is inferior to the effectiveness it obtained for the test set consisting of animal proteins. One possible reason is that the toxic protein domains between animal species and bacteria differ greatly. As shown in Fig. 4, of the 269 toxic protein domains found by making use of InterProScan, 71 were found in the animal protein test set, and 97 were found in the bacteria protein test set. However, as shown by Supplementary Table S2, only ten similar toxic protein domains can be found between the animal and bacteria toxic proteins.

Additionally, we also investigated the detected motifs in the animal and bacteria toxic proteins. We used MEME to generate 120 motifs for the animal and bacteria toxic proteins, respectively. These are then compared using TOMTOM to detect similar motifs between animal and bacteria toxic proteins. As shown in Fig. 4A, only five such motifs (see Supplementary Figure S1) could be detected. Our results suggest that there are indeed differences between toxic protein domains in animal species and bacteria, thus justifying that the cross-species effectiveness of any strategy for protein toxicity prediction is inferior to its effectiveness on the animal protein test set.

3.3 Comparison of different network architectures with and without domain embeddings

In this experiment, we evaluate the different strategies to deal with a variable input size, both with and without domain embeddings. The results obtained are shown in Fig. 5A. All strategies except GRU improve on the naïve zero-padding approach, which reaches an auPRC of 0.893. Moreover, incorporating the domain embeddings improves all network

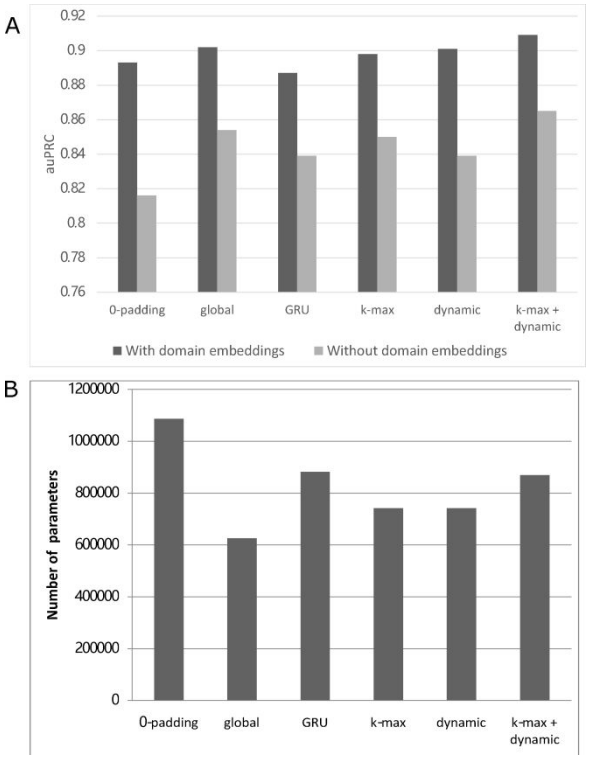

Fig. 5 Different network architectures for dealing with a variable input size. (A) Effectiveness of the different network architectures with and without adding protein domain embeddings. (B) The number of parameters of the different network architectures (with protein domain embeddings added). Note that 0-padding refers to zero-padding only, global refers to global max pooling, k-max refers to k-max pooling, and dynamic refers to dynamic max pooling.

architectures with a large margin, demonstrating the effectiveness of domain embeddings. The most effective strategy is the combination of k-

max pooling and dynamic max pooling, for which an auPRC of 0.913 is measured. However, the difference with other strategies is minimal.

For the sake of completeness, we also compare the number of parameters used by the individual network architectures. As shown in Fig. 5B, we can see that zero-padding comes with a considerably larger number of trainable parameters when compared to the other architectures, for which the number of parameters is similar.

Compared to ToxDL with the combination of k-max pooling and dynamic max pooling, the network architecture with global max pooling yields a similar effectiveness for the animal test set at a lower computational complexity (a fewer number of parameters). However, the architecture with global max pooling yields an auROC of 0.718 and an auPRC of 0.462 on the bacteria test set, which is much lower than the auROC of 0.786 and the auPRC of 0.525 of ToxDL, respectively. In addition, on the bacteria test set, the single k-max pooling strategy yields an auROC of 0.781 and an auPRC of 0.521, whereas the single dynamic max pooling strategy achieves an auROC of 0.743 and an auPRC of 0.475. Both strategies are superior to the use of global max pooling, demonstrating better transferability. Thus, the combination of k-max pooling and dynamic max pooling is used in ToxDL.

3.4 Identifying protein domains related to toxicity

By using Integrated Gradients to generate saliency maps, we can identify important motifs associated with protein toxicity. In the saliency map visualizations of Fig. 6, larger letters denote amino acids that come with higher contribution to the toxicity prediction. As shown in Fig. 6A, we can observe a high correspondence between the motif detected by ToxDL and the toxic Pfam domain PF00537 (Scorpion toxin), as we observe a regular occurrence of C amino acids that substantially contribute to the prediction in a positive way. In addition to verified toxic domains, ToxDL is also able to detect novel candidate motifs that can possibly be associated with protein toxicity. Such a motif can be found in Fig. 6B. However, no experimental evidence is currently available that shows that this motif is effectively related to protein toxicity.

3.5 Visualization of the learned protein domain embeddings and the impact of order

In this experiment, we use the protein domain embeddings learned by domain2vec to investigate whether the toxic protein domains are more similar than other protein domains. To that end, we map the 256-D embeddings of the 36,713 UniProt protein domains to a 2-D space using t-SNE (van der Maaten and Hinton, 2008). As shown in Fig. 7 (red circle), about 30 of the 269 toxic domains are very close in distance, indicating that certain toxic domain embeddings are similar. We can also observe that some domains are much more similar than others in the embedding space; this is a characteristic that cannot be captured by one-hot encodings. In nature language processing, sentences come with word order. However, in proteins, we only know which domains are located in proteins, and not their order. Thus, we also learn protein domain embeddings from a corpus consisting of shuffled domains for each protein. We calculate the pairwise distances between protein domains using the embeddings learned from (1) a corpus consisting of protein domains of which the order has been preserved and (2) a corpus consisting of protein domains of which the order has been shuffled. For each of the 36,713 protein domains, we randomly select another protein domain to create a protein domain pair. As shown in Supplementary Figure S2A, the pairwise distance between protein domains in both embedding spaces is similar: the R-value of the best-fitting line is 0.953. Furthermore, as shown in Supplementary Figure S2B and S2C, no substantial difference in effectiveness can be observed for the two test sets used, indicating that the order of the domains in proteins does not have a significant impact on the learned embeddings. This can possibly be attributed to the fact that each protein has a limited number of domains within a context window of size five (see Supplementary Figure S3).

3.6 Using the ToxDL web service to modify protein toxicity

We provide an online version of ToxDL for predicting protein toxicity, with this web service only requiring the input sequence of a protein to predict its toxicity. For this web service, we trained a new prediction model on a combination of the original training and validation sets, doing subsequent validation using the original test set. That way, we have more data to train on, making it possible for the web service to produce predictions that are more accurate. The ToxDL source code and the underlying datasets are available at https://github.com/xypan1232/ToxDL. The ToxDL web service can also be used to estimate toxicity after applying one or more mutations to a protein sequence. As an example, we consider the annotated toxic protein TMC_CINAN without the toxic domain information generated by InterProScan (see Fig. 8). ToxDL predicts TMC_CINAN to be a toxic protein with a probability of 0.722, though InterProScan does not detect any toxic domains in this protein. As suggested by the detected motif shown in Fig. 8A, cysteine amino acids (highlighted in red) seem crucial for toxicity.

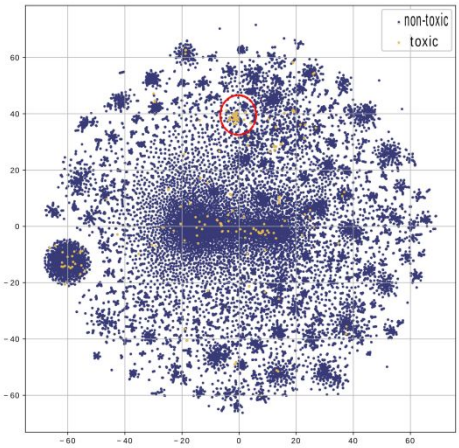

Fig. 7 Visualization of the learned protein domain embeddings in a 2-D space using t-SNE.

Given the above, we mutate the amino acid C to R (requiring only a single nucleotide mutation) in the two occurrences of C that provide the highest positive contribution to the prediction. As shown in Fig. 8B, the probability for the protein to be toxic drops to 0.405. This demonstrates

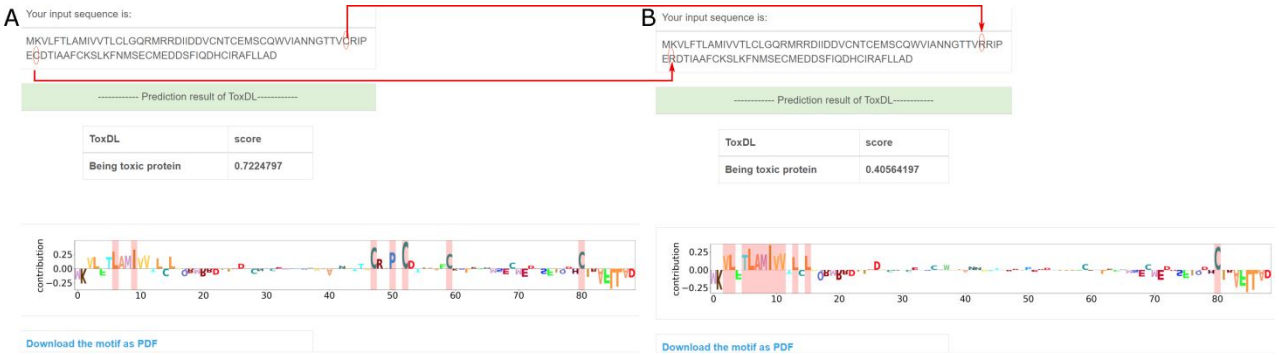

Fig. 8 Changing the toxicity of the toxic protein TMC_CINAN using our ToxDL web service. (A) ToxDL predicts TMC_CINAN to be a toxic protein with a probability of 0.722. (B) ToxDL predicts that the modified protein comes with a toxicity probability of 0.405. The modified amino acids can be found in the red circles. The key residues with a contribution score that is at least 50% of the maximum score are highlighted in red.

that we can use ToxDL to modify the proclaimed toxicity of a protein in a directed way, by changing the important subsequences in the *in silico* detected motif. As another example, the protein TXCL1_CALPA with two toxic protein domains (highlighted in yellow) was modified to be non-toxic (see Supplementary Figure S4) by again changing C to R, with the toxicity probability decreasing from 0.990 to 0.369.

4 Discussion

Known protein domains are highly informative for function prediction. In ToxDL, we represent each domain by making use of an embedding (that is, a vector of numerical values), with this embedding encapsulating domain context information such as co-occurrence. Indeed, compared to a one-hot encoded vector that merely indicates the presence of specific domains, domain embeddings can leverage contextual similarity. Our experimental results show that integrating protein domain embeddings into ToxDL significantly improves the effectiveness of toxicity prediction. Due to the fact that the average length of a protein in the Animal Toxin Annotation Project database is substantially shorter than the average length of any protein (see Supplementary Figure S5; in the training set, the average length of toxic and non-toxic proteins is 124.18 and 478.98, respectively), we evaluated ToxDL on an animal test subset consisting of 419 proteins with a length between 100 and 500. As shown in Supplementary Table S3, ToxDL yields an F1 score of 0.710 (±0.032), an MCC of 0.696 (±0.035), an auROC of 0.981(±0.005), and an auPRC of 0.836 (±0.030). These values are lower than the values obtained for the full animal test set, but still superior to the values obtained by the other baseline methods that have been applied to this test subset. Furthermore, we added the normalized protein length (i.e., the protein length is in [0, 1]) as an additional feature to ToxDL, denoting this new method as ToxDL-length. ToxDL-length yields an F1-score of 0.812 (±0.023), an MCC of 0.797 (±0.025), an auROC of 0.987 (±0.003), and an auPRC of 0.897 (±0.026). As shown in Supplementary Table S4, ToxDL-length yields a similar effectiveness as ToxDL. These results demonstrate that adding the protein length to ToxDL does not improve the prediction effectiveness. Moreover, as shown in Supplementary Figure S6, the protein length distributions of toxic and non-toxic proteins in the bacteria test set are similar but different from those of the training set, with ToxDL being superior to the baseline methods used. In addition, ToxDL is able to identify known motifs related to protein toxicity, as shown in Fig. 6. These results show that ToxDL learns patterns truly related to protein toxicity instead of being biased towards protein length. In future work, a more stringent benchmark dataset consisting of toxic and non-toxic proteins with a similar length distribution could be constructed.

Our experimental results also indicate that cross-species effectiveness obtained for bacteria proteins is much lower than the effectiveness obtained for animal proteins, since toxic domains are dissimilar between bacteria and animals. However, given the small number of annotated toxic proteins available for bacteria, it is currently still infeasible to train a standalone deep learning model for predicting bacteria protein toxicity. As more and more annotated toxic proteins are becoming available for bacteria, we expect that, in time, it will become possible to train an equivalent ToxDL model for bacteria.

5 Conclusions

In this paper, we proposed ToxDL, a deep learning-based approach for protein toxicity prediction, with this *in silico* approach being able to deal with variable-length sequences in input. In addition, we developed domain2vec, a tool for generating protein domain embeddings, and where this tool is part of ToxDL. We trained ToxDL on toxic proteins from animal species like snakes and spiders, given the relatively large amount of data made available through the Animal Toxin Annotation Project, as well as the potential for numerous applications in pharmacology and drug research. We demonstrated that ToxDL is outperforming other state-of-the-art approaches, even when generating cross-species predictions. Furthermore, ToxDL is able to highlight toxic motifs, as supported by evidence available in the public domain. Finally, the ToxDL webserver allows for directed *in silico* modification of a protein sequence, thus making it possible to alter its predicted toxicity.

Funding

This work has been supported by the National Natural Science Foundation of China (No. 61903248, 61725302, 61671288) and the Science and Technology Commission of Shanghai Municipality (No. 17JC1403500), BASF, Ghent University, Ghent University Global Campus, Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research-Flanders (FWO-Flanders), and the European Union.

Conflict of Interest: none declared.

References

Altschul, S.F., *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;25(17):3389-3402.
Bailey, T.L., *et al.* The MEME Suite. *Nucleic Acids Res* 2015;43(W1):W39-49.
Bernabe-Orts, J.M., *et al.* Assessment of Cas12a-mediated gene editing efficiency in plants. *Plant Biotechnol J* 2019;17(10):1971-1984.
Cole, T.J. and Brewer, M.S. TOXIFY: a deep learning approach to classify animal venom proteins. *PeerJ* 2019;7:e7200.
Cui, X.D., Goel, V. and Kingsbury, B. Data Augmentation for Deep Neural Network Acoustic Modeling. *Int Conf Acoust Spee* 2014.
El-Gebali, S., *et al.* The Pfam protein families database in 2019. *Nucleic Acids Res* 2019;47(D1):D427-D432.
Eraslan, G., *et al.* Deep learning: new computational modelling techniques for genomics. *Nat Rev Genet* 2019;20(7):389-403.
Fu, L., *et al.* CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;28(23):3150-3152.
Gupta, S., *et al.* In silico approach for predicting toxicity of peptides and proteins. *PLoS One* 2013;8(9):e73957.
Gupta, S., *et al.* Quantifying similarity between motifs. *Genome Biol* 2007;8(2):R24.
Hammond, B., *et al.* Toxicological evaluation of proteins introduced into food crops. *Crit Rev Toxicol* 2013;43:25-42.
Jones, P., *et al.* InterProScan 5: genome-scale protein function classification. *Bioinformatics* 2014;30(9):1236-1240.
Jungo, F., *et al.* The UniProtKB/Swiss-Prot Tox-Prot program: A central hub of integrated venom protein data. *Toxicon* 2012;60(4):551-557.
Kalchbrenner, N., Grefenstette, E. and Blunsom, P. A Convolutional Neural Network for Modelling Sentences. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* 2014:655–665.
Klambauer, G., *et al.* DeepTox: Toxicity prediction using deep learning. *Toxicol Lett* 2017;280:S69-S69.
Lecun, Y., *et al.* Gradient-based learning applied to document recognition. *P Ieee* 1998;86(11):2278-2324.
Mandrekar, J.N. Receiver Operating Characteristic Curve in Diagnostic Test Assessment. *Journal of Thoracic Oncology* 2010;5(9):1315-1316.
Menichelli, C., Gascuel, O. and Brehelin, L. Improving pairwise comparison of protein sequences with domain co-occurrence. *Plos Comput Biol* 2018;14(1):e1005889.
Mikolov, M., *et al.* Distributed representations of words and phrases and their compositiobality. *Proceedings of the 26th International Conference on Neural Information Processing Systems* 2013;2:3111-3119.
Mumtaz, M.M. and Pohl, H.R. Interspecies uncertainty in molecular responses and toxicity of mixtures. *Exp Suppl* 2012;101:361-379.
Naamati, G., Askenazi, M. and Linial, M. ClanTox: a classifier of short animal toxins. *Nucleic Acids Res* 2009;37(Web Server issue):W363-368.
Negi, S.S., *et al.* Functional classification of protein toxins as a basis for bioinformatic screening. *Sci Rep-Uk* 2017;7.
Pan, X., *et al.* Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC Genomics* 2018;19(1):511.
Potter, S.C., *et al.* HMMER web server: 2018 update. *Nucleic Acids Res* 2018;46(W1):W200-W204.
Saha, S. and Raghava, G.P. BTXpred: prediction of bacterial toxins. *In Silico Biol* 2007;7(4-5):405-412.
Shrikumar, A., Greenside, P. and Kundaje, A. Learning important features throughpropagating activation differences. *Proceedings of the 34th International Conferenceon Machine Learning* 2017;70:3145-3153.
Sundararajan, M., Taly, A. and Yan, Q. Axiomatic Attribution for Deep Networks. *arXIV* 2017;arXiv:1703.01365.
Tahbaz, N., *et al.* Characterization of the interactions between mammalian PAZ PIWI domain proteins and Dicer. *EMBO Rep* 2004;5(2):189-194.
van der Maaten, L.J.P. and Hinton, G.E. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 2008;9:2579-2605.
Vlieghe, P., *et al.* Synthetic therapeutic peptides: science and market. *Drug Discov Today* 2010;15(1):40-56.
You, R., *et al.* GOLabeler: improving sequence-based large-scale protein function prediction by learning to rank. *Bioinformatics* 2018;34(14):2465-2473.