Representation learning applications in biological sequence analysis

Hitoshi Iuchi[a,b,*], Taro Matsutani[a,b], Keisuke Yamada[b], Natsuki Iwano[b], Shunsuke Sumi[b,c], Shion Hosoda[a,b], Shitao Zhao[b], Tsukasa Fukunaga[b,d] and Michiaki Hamada[a,b,e,*]

[a] *Computational Bio Big-Data Open Innovation Laboratory, National Institute of Advanced Industrial Science and Technology, Okubo, Shinjuku, Tokyo 169–8555, Japan*

[b] *Department of Electrical Engineering and Bioscience, Faculty of Science and Engineering, Waseda University, Okubo, Shinjuku, Tokyo 169-8555, Japan*

[c] *Department of Life Science Frontiers, Center for iPS Cell Research and Application, Kyoto University, Kawahara-Cho, Shogoin, Sakyo, Kyoto 606-8507, Japan*

[d] *Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo, Hongo, Bunkyo, Tokyo 113-0032, Japan*

[e] *Graduate School of Medicine, Nippon Medical School, Sendagi, Bunkyo, Tokyo 113-8602, Japan*

ARTICLE INFO

Keywords:
natural language processing
representation learning
sequence analysis
word2vec
BERT

ABSTRACT

Remarkable advances in high-throughput sequencing have resulted in rapid data accumulation, and analyzing biological (DNA/RNA/protein) sequences to discover new insights in biology has become more critical and challenging. To tackle this issue, the application of natural language processing (NLP) to biological sequence analysis has received increased attention, because biological sequences are regarded as sentences and k-mers in these sequences as words. Embedding is an essential step in NLP, which converts words into vectors. This transformation is called representation learning and can be applied to biological sequences. Vectorized biological sequences can be used for function and structure estimation, or as inputs for other probabilistic models. Given the importance and growing trend in the application of representation learning in biology, here, we review the existing knowledge in representation learning for biological sequence analysis.

1. Introduction

Considerable advances in high-throughput sequencing have resulted in rapid data accumulation [1]. Although these modern technologies produce a large amount of data, they do not provide any interpretation or biological information. Thus, analyzing biological sequences, such as DNA/RNA/protein sequences, to make biological discoveries has become more critical and challenging. To tackle this issue, the application of natural language processing (NLP) to sequence analysis has attracted great attention in terms of biological sequences as sentences and k-mers in these sequences as words [2, 3].

NLP aims to allow computers to understand the contents of natural language, including the context, and accurately extract information and insights [4]. Natural language is composed of characters, such as the alphabet, and the meaning is constructed using grammar and semantics. In the same way, biological sequences can be regarded as sentences with different letters, and biophysical and biochemical rules define properties, such as the function and structure [5]. Biological sequences are consistent with natural language where characters define their meaning, and the meaning depends on the neighboring sequence. For example, whether the word "bank" in a sentence refers to a financial institution or raised portion of seabed depends on the context. Similarly, whether a part of an RNA sequence forms a secondary structure depends on its neighboring sequences. Thus, there are similarities between natural language and biological sequences, and it would be natural to apply NLP to a more in-depth understanding of the function and structure encoded in the biological sequence.

Representation learning is an essential step in NLP and indicates automatic systems to explore the representation of raw data, such as words or characters [6]. In general, the representation is provided as a real-valued vector, called *distributed representation*. Successful representation learning is expected to convert words into vectors while preserving their semantic similarity. For example, the names of foods, like "sushi" and "pizza," should be converted into similar vectors and the names of organisms, such as "frog," should be assigned entirely different vectors (Figure 1). In biological sequences, *N*-methyl-D-aspartate receptor and α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor, which are ionotropic glutamate receptors, are expected to be converted into similar vectors, whereas GFP, a fluorescent protein, is expected to be converted into a completely different vector. Thus, representation learning indicates the transformation from words to vectors while preserving the similarities and differences between words.

Biological sequences vectorized by representation learning can be directly used for biological tasks, such as function and structure prediction. If the vector similarity between proteins is high, it can be inferred that they have similar functions and structures. Note that vector similarity/distance can be calculated using linear algebra operations, such as dot product, Euclidian distance, and cosine similarity. In particular, the successful encoding of words via representation

[*] Corresponding author

hitosh.iuchi@gmail.com (H. Iuchi); mhamada@waseda.jp (M. Hamada)

ORCID(s): 0000-0002-2278-3443 (H. Iuchi); 0000-0001-6170-9927 (T. Matsutani); 0000-0003-1256-9299 (K. Yamada); 0000-0001-5603-8903 (N. Iwano); 0000-0003-1007-9472 (S. Sumi); 0000-0001-5093-6833 (S. Hosoda); 0000-0003-4442-6049 (T. Fukunaga); 0000-0001-9466-1034 (M. Hamada)

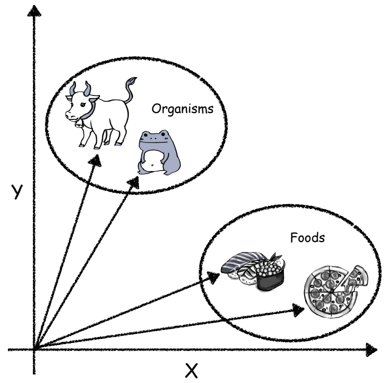

Figure 1: Ideal representation learning should convert the names of foods, such as "sushi" and "pizza," into similar vectors and assign different vectors to the names of organisms, such as "cow" and "frog."

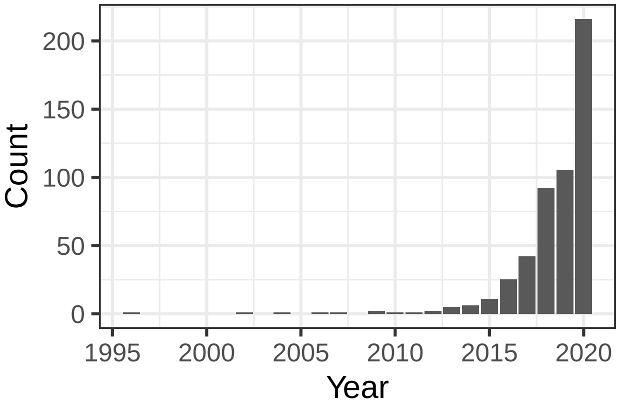

Figure 2: Change in the number of hits for the search term "representation learning" (with double quotation) in PubMed (https://pubmed.ncbi.nlm.nih.gov/).

learning has been recognized as an essential research area because the performance of NLP and deep learning depends on the quality of the representation [6]. Thus, a *good* representation of a biological sequence is critical for clustering, function, structure, and disorder prediction [2].

Given the significance and growing trend in the application of representation learning in biology (Figure 2), here, we describe a review of representation learning for sequence analysis. It should be noted that this review covers the application of representation learning to biological sequence analysis, and its use in biological literature and medical records is beyond the scope of this review. This review is organized as follows: Section 2 introduces the basic representation techniques for NLP. Section 3 provides a comprehensive survey of representation learning approaches for sequence analysis. Section 4 presents a summary and an outlook of representation learning applications in biological sequence analysis.

2. Representation learning techniques

Currently, the acquisition of distributed representations of biological sequences is mainly achieved using neural networks developed in NLP. In representation learning for NLP, it is assumed that the words that appear in the same context should have similar meanings according to the *distribution hypothesis* [7]. Representation learning methods based on the distribution hypothesis attempt to vectorize words or phrases by training the neural networks with architectures specialized for capturing the relationships among words from a corpus, a set of documents. Various representation learning methods presented in this review are taken from neural-network-based language models specialized for biological sequences; thus, it is essential to understand the underlying architecture of the neural networks developed for NLP. In this section, we briefly summarize the development of basic representation learning techniques.

word2vec is the first successful method to obtain distributed representations using a neural network [8, 9]. There are two types of neural networks used in word2vec: skip-gram model that predicts the words around the input word and a continuous bag-of-words model that predicts the target word from the surrounding words. Until the advent of word2vec, researchers used neural networks to describe the syntactic structure [10, 11]. The skip-gram model proposed by Mikolov attracted attention for its ability to capture not only grammatical correctness but also semantic features, as described in the introduction. word2vec with the skip-gram model acquires a distributed representation for each word by training the three-layer neural network, as shown in Figure 3. Given a sentence with T words and the t-th word w_t, the model predicts the words present in the vicinity of w_t in that sentence. The parameters to be estimated in the skip-gram model include the weight matrix X to predict the d-dimensional hidden layer $h \in \mathbb{R}^d$ from the one-hot encoded input layer and weight matrix Y to predict the output from h. They are predicted using the formula described below:

$$\hat{X}, \hat{Y} = \arg\max_{X,Y} \frac{1}{T} \sum_{t=1}^{T} \left\{ \sum_{t'=t-c}^{t-1} \log p(w_{t'} \mid w_t, X, Y) + \sum_{t'=t+1}^{t+c} \log p(w_{t'} \mid w_t, X, Y) \right\} \quad (1)$$

, where c is a constant indicating the number of words farther away from w_t that should be included in the prediction. The model performs the same operation for all sentences to complete training. In this case, the weight matrix X is a $V \times d$ matrix, where V is the number of words in the vocabulary. If w_t is the v-th word in the vocabulary, we can obtain the distributed representation of the word w_t as the v-th vector of the predicted X (i.e., $\hat{X}_v \in \mathbb{R}^d$). The word2vec representation has additive compositionality and has become famous for allowing intuitive operations, such as $\hat{X}_{\text{Vietnam}} + \hat{X}_{\text{capital}} \approx \hat{X}_{\text{Hanoi}}$, as shown previously [9]. Hence, word2vec succeeded in obtaining highly interpretable distributed representations for the first time and di-

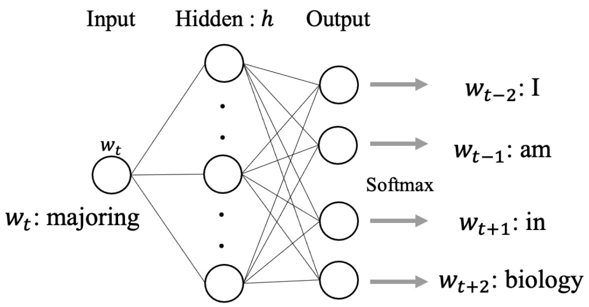

Figure 3: Skip-gram model used in word2vec. This neural network model has three fully connected layers: the input, hidden, and output layers. In this case, it attempts to learn the features from the sentence, "I am majoring in biology," and predict the words surrounding w_t, "majoring. "

rected subsequent development in representation learning.

The fact that word2vec captured semantic features was a remarkable breakthrough in representation learning, and various extended models based on word2vec were proposed. GloVe uses word co-occurrence matrices, which have been used in classical latent semantic analysis, such as singular value decomposition [12]. It shows higher semantic accuracy than word2vec. FastText is one of the embedding methods based on the skip-gram model [13]. It can train the model very quickly while maintaining the same accuracy as conventional methods. In addition, several methods were developed to obtain a distributed representation for each sentence (not word) based on the word2vec concept. doc2vec utilizes the paragraph vectors, which captures the context for each paragraph and provides the features for each sentence [14].

Although word2vec has enabled considerable progress in representation learning, it cannot express the semantic polysemy of words because it yields a single d-dimensional vector for a single lexicon, as mentioned above. For example, "right" that appears in "right to vote" and "turn right" differ in meaning, but they are embedded at the same point using word2vec. The approach to solving this problem is called word sense disambiguation in NLP [15], and it calls for architecture to consider the context and meaning of a sentence. In biological sequences, the context of a word in a sentence is equivalent to the role of a particular nucleic/amino acid in the whole sequence. Hence, the polysemy in biological sequences is critical, similar to that in natural languages. Here, we introduce two methods that can take such contexts into account: one that can achieve this by making the neural network recursive using a recurrent neural network (RNN) or long short-term memory (LSTM) [16] and another that uses the *attention* mechanism.

Embeddings from language models (ELMo) dissolves the polysemy problem using the model stacked with multiple bidirectional-LSTM (bi-LSTM) and yields the distributed representations by taking the linear weighted-sum of outputs of their hidden layers [17]. RNN and LSTM have been utilized mainly for sequential tasks, such as document generation and machine translation [18, 19]. In a language model with a simple forward LSTM, the occurrence probability of the t-th word in a sentence, w_t, depends on the set of words that appear before w_t (denoted as $\boldsymbol{w}_{1:t-1}$). The model trains the parameters to maximize the joint probability for all words, $\{w_1, \cdots, w_t, \cdots, w_T\}$. To calculate $p(w_t|\boldsymbol{w}_{1:t-1})$, LSTM uses the hidden layer of w_t (its output is denoted by $h_{\text{forward},t} \in \mathbb{R}^d$), which depends on w_{t-1} and $h_{\text{forward},t-1}$. As the hidden layer is computed recursively depending on the word order, LSTM-based models allow context-aware learning. Previously [17], bi-LSTM modules were used in their language model called the bidirectional language model (bi-LM). bi-LSTM considers not only the forward but also backward word dependency. In a backward LSTM, the hidden layer of w_t and its output $h_{\text{backward},t}$ depend on w_{t+1} and $h_{\text{backward},t+1}$ as opposed to the case in forward LSTM. By considering word dependency in the backward direction, bi-LM can incorporate relationships among words that cannot be captured by the forward LSTM alone. bi-LM contains a stack of L bi-LSTM modules (see Figure 4), and all the modules are trained to maximize the joint probability of generating the entire sentence as follows:

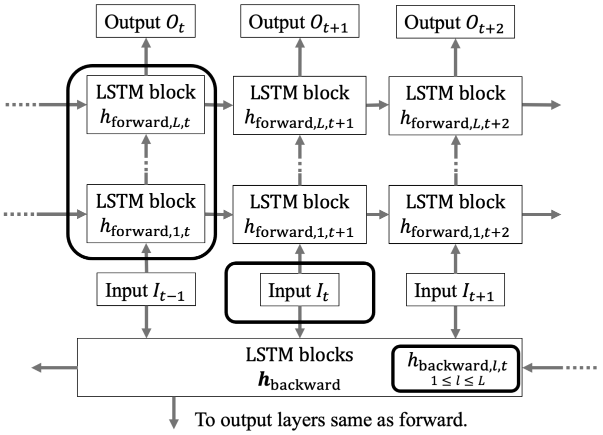
Output O_t
Output O_{t+1}
Output O_{t+2}
LSTM block $h_{\text{forward},L,t}$
LSTM block $h_{\text{forward},L,t+1}$
LSTM block $h_{\text{forward},L,t+2}$
LSTM block $h_{\text{forward},1,t}$
LSTM block $h_{\text{forward},1,t+1}$
LSTM block $h_{\text{forward},1,t+2}$
Input I_{t-1}
Input I_t
Input I_{t+1}
LSTM blocks $\boldsymbol{h}_{\text{backward}}$
$h_{\text{backward},l,t}$ $1 \le l \le L$
To output layers same as forward.

Figure 4: The graphical representation of a bidirectional language model. Input I_t shows the embedding of the t-th word in the sentence, w_t. The output, O_t, is transformed to a probability with a softmax function, and all the modules are trained to maximize the observed probability of w_t. $2L+1$ layers circled in squares with rounded corners are used to calculate ELMo$_t$, the distributed representation for w_t.

$$\frac{1}{T}\sum_{t=1}^{T}\left\{\log p(w_t \mid \boldsymbol{w}_{1:t-1}) + \log p(w_t \mid \boldsymbol{w}_{t+1:T})\right\} \quad (2)$$

Finally, the distributed representation, ELMo, is obtained by taking the weighted-sum of outputs from $2L+1$ layers, which are hidden layers for each LSTM module and an input embedding layer. ELMo succeeded in placing the same lexicon on different points in a high-dimensional space, depending on the context.

Another approach to solving the polysemy problem is to use the attention mechanism. In brief, attention quantifies the degree of correspondence between words [20, 21].

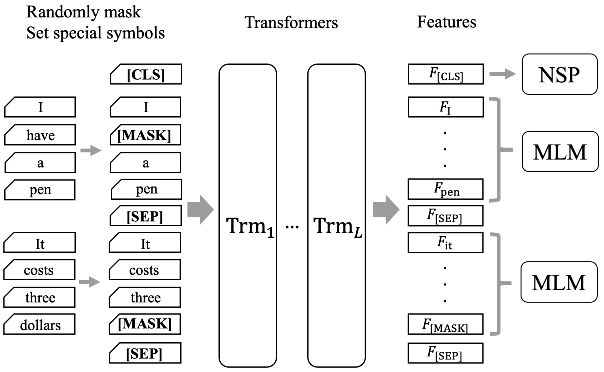

Figure 5: The graphical representation of Bidirectional encoder representations from transformers (BERT) architecture. Preparation of special tokens enables the model to extract features based on the self-attention of the whole sentence. BERT is trained with two tasks: masked language model (MLM) and next sentence prediction (NSP).

Neural networks with attention mechanisms have an attention weight that is obtained by calculating the association of hidden layers (e.g., using the inner product) for arbitrary combinations of words in sentences. If the two words used to compute the attention weight come from different sentences, this attention is called the source-target-attention; if they are from an identical sentence, it is called self-attention. Models that use attention weights in the forward propagation are highly expressive, and we can naturally introduce an attention mechanism to representation learning. Transformer, which implements attention mechanism and positional encoding [22] in Key-Value Memory neural network [23, 24] without conventional context-aware architectures, such as RNN or LSTM, has achieved the state-of-the-art accuracy in several tasks, including machine translation [25].

Bidirectional encoder representations from transformers (BERT) have multiple transformers with self-attention connected in series (see Figure 5) [26]. In the pre-training of BERT, the input is a set of tokens connecting two sentences. At this time, a part of the input words is masked, and the model predicts the masked words from the extracted features considering the context. In addition, the model performs a binary classification of whether the two input sentences are semantically consecutive. Similar to other methods, we can use the outputs of pre-trained transformer layers as the distributed representations of input sentences.

Neural networks with attention mechanisms, such as transformer and BERT, capture distal word associations better than conventional recursive models represented by RNN and LSTM [20, 27]. This is because, in recursive models, a hidden layer of a certain word depends on the hidden layers of the neighboring words only, and the contribution of distal words becomes small or converges to zero. In contrast, the use of attention is robust against such weight loss because the model always refers to the association of it with all words. This feature of BERT is attractive from the viewpoint of biology since distal interactions are important for structural prediction and other purposes. Another advantage of BERT is task-independent versatility. For instance, when we use ELMo, we should prepare a task-specific model to transfer the obtained distributed representations to other tasks. In contrast, with BERT, we can utilize the same architecture used in pre-training (as shown in Figure 5) without modification. Fine-tuning, which uses pre-trained hidden layers for initialization and optimizes the parameters for each task, has achieved state-of-the-art accuracy in many NLP tasks [26].

The main advantage of obtaining features through unsupervised learning is that it can retain versatility for transfer learning to various tasks. However, to build a specialized model for a specific task, representation learning in a *supervised* manner is also useful. StarSpace is a supervised learning method [28], which uses labeled documents as the training dataset, and embeds words and labels in the same space so that a label is close to words associated with it. Embedding with StarSpace allows for text classification, that is, prediction of labels used in the course of learning with higher accuracy than the other unsupervised methods, and provides highly interpretable vectors. As this example shows, supervised representation learning is also a practical option if the correct labels are known.

Since the development of word2vec in 2013, the field of representation learning in NLP has been growing at an astonishing pace. Considering the models based on transformer or BERT, several modern improved methods have continued to provide increased accuracy [29, 30]. Furthermore, similar to the big impact of the attention mechanism, the emergence of new concepts may also reconstruct the current paradigm of language modeling. These substantial developments in machine learning will be useful for bioinformatics and sequence analyses. As numerous examples are introduced in later sections, we believe that applying the latest representation learning techniques to biological sequences will lead to a discovery or elucidation of novel information in this domain.

3. Survey of representation learning applications in sequence analysis

We conducted an exhaustive survey, as shown in Table 1, for articles that met the following criteria: (i) Peer-reviewed and published in PubMed, except for BERT, which was recently published with a limited number of peer-reviewed articles; (ii) explicitly used a language model, such as word2vec or BERT; (iii) provided the source code or the model for repeatability or verification.

3.1. Applications for structure/function prediction

ProtVec is the first model to use the embedding method for biological sequences [31]. This method regarded *3-mers* of amino acids as words and used 546,790 protein sequences from the Swiss-Prot database as the training dataset. Subsequently, word2vec using the skip-gram model was applied to the dataset, and 100-dimensional protein vectors were calculated. Originally, ProtVec was evaluated based on protein family classification and disordered protein prediction accuracies and it achieved high performance in both. Currently,

Table 1
Comprehensive survey of representation learning application in biological sequences

Method name	Model	Training data	Task	Avail. and repr.	Ref.
ProtVec	word2vec	547K proteins	family classification, disorder prediction	+	[31]
HLA-vec	word2vec	HLA-I binding/non-binding peptides	HLA-I binding prediction	++	[32]
m-NGSG	word2vec	0.1K–3K proteins	protein classification	++	[33]
ene2vec	word2vec	89K positive and 495K negative mRNAs	N6-methyladenosine site prediction	++	[34]
–	word2vec	3K–101K of 300 bp genomic regulatory regions	regulatory region prediction	++	[35]
ProtVecX	word2vec	371–44K proteins	venom toxin prediction, enzyme prediction	+++	[36]
MHCSeqNet	word2vec	228K peptide-MHC pairs	MHC binding prediction	+++	[37]
–	word2vec	1M 16S rRNAs	sample class (e.g., body part) prediction	+++	[38]
fastDNA	word2vec	356–3K bacterial genomes	species identification	++	[39]
NucleoNN	word2vec	86/72 SNPs in the control/exposure samples	investigating allele-interactions	++	[40]
–	word2vec	3K–22K CPI pairs	CPI prediction	+++	[41]
FastTrans	word2vec	1K membrane transporter and 1K membrane non-transporter proteins	substrate prediction of transport proteins	++	[42]
INSP	word2vec	78 nuclear proteins	nuclear localization prediction	++	[43]
–	word2vec	9M proteins	function prediction	++	[44]
Its2vec	word2vec	126K ITSs	species identification	++	[45]
4mCNLP-Deep	word2vec	*C. elegans* genome (WBcel235/ce11)	N4-methylcytosine sites prediction	++	[46]
–	doc2vec	525K proteins	Localization, T50, absorption, enantioselectivity prediction	+++	[47]
EP2vec	doc2vec	650K enhancers and 93K promotors	enhancer-promoter interaction prediction	++	[48]
IDP-Seq2Seq	Seq2Seq	3K proteins	Disorder prediction	++	[49]
–	Glove	244K–504K chromatin accessible regions	chromatin accessibility prediction	++	[50]
CircSLNN	Glove	37 dataset of RBP-binding sites on circular RNAs	RBP-binding sites prediction of circRNAs	+	[51]
–	FastText	3K promoters and 3K non-promoters	promoter stregnth classification	++	[52]
iEnhancer -5Step	FastText	1K human enhancers and 1K human non-enhancers	enhancer prediction	++	[52]
TNFPred	FastText	18 tumor and 133 non-tumor necrosis factors	tumor necrosis factors classification	++	[53]
eDNN-EG	FastText	518 essential and 1K non-essential genes	essential gene prediction	+	[54]
ProbeRating	FastText	440K proteins and 274K nucleic acids	nucleic acid-binding proteins binding preference prediction	++	[55]
CSCS	bi-LSTM	4K–58K viral proteins	viral escape mutation prediction	+++	[56]
UniRep	mLSTM	24M proteins	structure and function prediction	+++	[57]
UDSMProt	AWD-LSTM language model	499K proteins	enzyme class prediction, gene ontology prediction, remote homology, fold detection	+++	[58]
USMPep	AWD-LSTM language model	23K–120K MHC binding peptides	MHC binding affinity prediction	++	[59]
BindSpace	StarSpace	505K TF-associated and 505K non-associated DNA	TF-binding prediction	++	[60]
MutSpace	StarSpace	cancer mutation sites	cancer type prediction	++	[61]
SeqVec	ELMo	33M proteins	3-state secondary structure prediction, disorder prediction, localization prediction, membrane prediction	++	[62]
DNA-transformer	transformer	*E. coli* genome (MG1655)	Transcription start sites, translation initiation sites, 4mC methylation sites prediction	++	[63]
NuSpeak	ULMfit	92K RNAs	designing RNA toehold switches	++	[64]
TAPE	BERT	31M proteins	3-state secondary structure prediction, contact prediction, remote homology detection, fluorescence prediction, stability prediction	+++	[65]
ESM-1b	BERT	27M–250M proteins	remote homology detection, 8-state secondary structure prediction, contact map prediction, quantitative prediction of mutational effects	++	[66]
ProtBert	BERT	216M–2B proteins	3-/8-state secondary structure prediction, subcellular localization prediction, membrane-boundness prediction	++	[67]
DNABERT	BERT	*H. sapiens* genome (GRCh38.p13)	promoter prediction, TF-binding site prediction, splicing site prediction, functional variant analysis	+++	[68]

Avail. and repr. indicate availability and reproductivity, respectively. (+++) The source code to generate the model, pre-trained model, and detailed documentation, including data links and installation instructions, are available. (++) Either the source code to generate the model or the pre-trained model is available, and detailed documentation, including data links and installation instructions, are available. (+) Either the source code to generate the model or the pre-trained model is available, but the documentation is limited. *Model* indicates a general model (described in section 2) utilized in the method. K, kilo; M, mega; B, billion; HLA, human leukocyte antigen; MHC, major histocompatibility complex; CPI, compound–protein interaction; RBP, RNA binding protein; TF, transcription factor

ProtVec has also been utilized for predicting kinase activity [69] and gene function [70]. As ProtVec is a straightforward model, various extensions have been proposed. One of the extensions is seq2vec, which embeds not the k-mers of amino acids but the whole protein sequences [71]. Seq2vec utilizes doc2vec [14], an NLP method that embeds documents instead of words, which showed a higher performance than ProtVec in terms of protein family classification performance. Another extension is dna2vec [72], which embeds variable-length k-mers rather than fixed-length DNA k-mers using word2vec. ProtVecX is a similar method that uses word2vec to embed variable-length amino acid k-mers [36].

SeqVec is the first model that uses ELMo to achieve amino acid representation based on the whole protein sequence [62]. ELMo was applied to the UniRef50 dataset, which contains 33M proteins with 9.6G residues, regarding single amino acids as words. The extracted sequence profile was then fed to the per-residue prediction and per-protein prediction. With and without the evolutionary information, the model accurately predicted the secondary structure, disorder, localization, and membrane binding. The performance did not exceed that of the state-of-the-art methods [73, 74]. However, it was better than ProtVec [31] which is a context-independent model. In some tasks, such as protein function prediction, it outperformed one-hot encoding of k-mer-based embeddings and showed the competitive results obtained using ELMo [75].

UDSMProt is another language model representation extractor using a variant of LSTM [58]. The structure used was called AWD-LSTM [76], which is a three-layered bi-LSTM that introduces different types of dropout methods to achieve accurate word-level language modeling. UDSMProt was initially applied to the Swiss-Prot database and then fine-tuned for specific tasks, such as enzyme commission classification, gene ontology prediction, and remote homology detection. UDSMProt showed that upon pre-training with external data, the model performed as well as the existing methods that were tailored to the task using a position-specific scoring matrix (PSSM) and outperformed them in two out of three tasks. In addition, it demonstrated that utilizing pre-training information can compensate for the lack of data, compared to the case where PSSM information is provided. These results and extensions, such as USMPep, which revealed the ability to successfully predict MHC class I binding [59], imply that language models can efficiently contextualize and achieve word-based representation.

ESM-1b is a BERT-based model trained on a massive biological corpus, particularly amino acid sequences [66]. The study presented a series of BERT models with varying parameter sizes. After pre-training on up to 250 million protein sequences, where each amino acid residue in a sequence was treated as a word, models accurately predicted the structural characteristics of proteins, including remote homology, secondary structure, and residue–residue contact. Representations emitted from the pre-trained 34-layer model were merged with multiple sequence alignments, which were the original input of existing secondary structure or contact prediction methods, and their prediction accuracy was improved. This result indicated that embedded representations from the pre-trained BERT incorporated more information than the multiple sequence alignments. Furthermore, the 34-layer model was fine-tuned to predict the quantitative effect of mutations and outperformed the state-of-the-art methods. As an attractive topic, other protein BERT models, such as TAPE transformer and ProtBert, have also been developed [65, 67]. Meticulous inspection of the TAPE transformer revealed that attention maps extracted from the pre-trained model reflect the context of input amino acid sequences [77]. For instance, one attention module, which specializes in deciphering residue–residue interactions, exhibited a significant correlation with experimental labels even though no structural information was provided. This phenomenon was later investigated by reconstructing protein contact maps from the attention maps of pre-trained ESM-1b [78]. The collection of studies illustrates that BERT-based models are highly interpretable and widely applicable to protein-related bioinformatics problems.

DNABERT, in contrast, is the only model, currently, to pre-train BERT-based models using a whole human reference genome [68]. During preprocessing, the genome, whose gaps and unannotated regions were excluded, was split into 5 to 510 consequent nucleotide sequences without overlapping and subsequently converted to 3- to 6-mer representations. Simply put, each subsequence of length 3 to 6 was regarded as a word. BERT models were pre-trained using k-mers with a masked language modeling objective and applied to downstream tasks. Upon task-specific fine-tuning, DNABERT demonstrated state-of-the-art or comparative performance in predicting promoter regions, binding sites of transcription factors (TFs), and splice sites. Attention analysis revealed that fine-tuned models captured the characteristics of each set of target sequences. For example, DNABERT fine-tuned using splicing datasets exhibited high attention weights in intronic regions in addition to target splice sites, indicating the ability of the model to learn the contextual significance of splicing enhancers or silencers in predicting splice sites. The study further applied DNABERT to predict promoters in the mouse genome and reported higher performance than those of existing deep learning methods. Overall, two-step training of the BERT architecture demonstrated its broad application to translate various genomic features in a cross-organism manner.

3.2. Applications for molecular interactions

Tsubaki *et al.* proposed a model by combining a graph neural network for compounds and a convolutional neural network (CNN) for proteins to predict compound–protein interactions (CPIs) [41]. Representations of compounds and proteins were obtained in an end-to-end manner. The word embeddings in the protein were learned from the training dataset using word2vec (3-mer of amino acids as words). To obtain protein vector representation, the average value of a set of hidden vectors was used with d-dimensional embedding after a hierarchical convolutional filter. Extensive evaluations were conducted on three CPI datasets (human, *C. elegans* [79] and DUD-E dataset [80]). The results showed that using the raw amino acid sequence as the input, the proposed approach significantly outperformed existing methods utilizing traditional chemical and biological features. They also established that the model could highlight 3D structural interaction sites between the compounds and proteins through an attention mechanism similar to that of words in sentences.

ProbeRating is a neural network-based recommender system utilizing word embeddings in NLP to infer binding profiles for unexplored nucleic acid-binding proteins (NBPs) [55]. ProbeRating achieves this goal using a two-stage framework. In the first stage, representation learning is performed using a package called FastBioseq, implementing FastText. Thus, the input feature vectors are extracted from the NBP sequences and nucleic acid probes. Authors chose 3-mers amino acids for proteins and 5-mers for nucleic acids as words. Three datasets (Uniprot400k [81], RRM3k [82], and Homeo8k [83]) were used to pre-train the FastBioseq protein embedding models, whereas RNA embedding models were trained directly from the RRM162 dataset [82]. In contrast, 8-mer frequency features were used for the DNA sequences in the Homeo215 dataset [84]. In the second stage, predicting the NBP binding preference was redefined as a recommender system formulation, where NBPs are like users and RNAs or DNAs are like products to be recommended. When no preference was available for a given user, the authors adapted and extended a strategy that converted the *binding intensity prediction* problem into a *similarity prediction* problem, solved it, and then converted it back. Extensive evaluation experiments were conducted on two tasks: RBP–RNA interaction and TF–DNA interaction. The results showed that ProbeRating outperformed three baseline methods (Nearest-Neighbor, Co-Evo [85] and AffinityRegression [84]). Further analysis suggested that this advantage was beneficial using both the neural network approach and input features extracted via word embeddings.

3.3. Applications in synthetic biology

Valeri *et al.* proposed a model that predicts synthetic riboregulators called toehold switches [86]. The model comprised a language model for toehold switch classification and a CNN-based model for toehold switch performance regression. In the language model, a sequence of toehold switches was embedded using ULMfit regarding a nucleotide as a word. They trained the model using toehold switches experimentally characterized by Angenent-Mari *et al.* [87]. The results showed that the model exhibited good and robust performance even for sparse training data and that the features obtained by the model revealed unknown properties of the toehold switches. They also showed that the trained model is easily fine-tuned by transfer learning using small external data [88, 89], and the fine-tuned model exhibited superior performance compared to an existing model. Finally, they showed that the fine-tuned model could help in the efficient design of toehold switches for various applications, such as SARS-CoV2 detection.

UniRep is a representation that comprehensively summarizes the semantics of arbitrary proteins and can be useful for various types of prediction tasks [90]. A protein sequence is embedded into UniRep using multiplicative LSTM (mLSTM), trained with 24M UniRef50 sequences [91], where an amino acid is regarded as a word. UniRep recapitulates biophysical properties, phylogenetics, and secondary structures of proteins. The authors also showed that UniRep outperformed other representations for predicting the structural and functional properties of *de novo* proteins, single point mutants, and natural proteins. These results suggest that UniRep is useful for the rational design of proteins. As a proof-of-concept, UniRep re-trained using deep mutational scanning data of GFP [92] was shown to effectively extrapolate GFP brightness outside the training domain. Therefore, UniRep was suggested to drastically reduce the cost for the rational design of GFP. Collectively, UniRep embodies various known protein characteristics and may be a versatile representation for protein bioinformatics.

3.4. Applications for other tasks

StarSpace is a *supervised* embedding method, which is different from the unsupervised embedding methods that we have introduced in section 2 [28]. Although StarSpace was originally developed for general NLP tasks, such as text classification, there are currently two bioinformatics applications. The first application is **BindSpace**, which predicts the binding sites of TFs [60]. BindSpace uses HT-SELEX experiments as the training dataset and applies StarSpace to the dataset by considering 8-mers and TFs as words and labels, respectively. In performance evaluation using the ENCODE ChIP-seq dataset, BindSpace achieved high classification performance even between paralogous TFs, which have highly similar binding motifs. The second application is **MutSpace**, which estimates the cancer types of patients from somatic mutation patterns [61]. This method regarded mutation patterns and cancer types as words and labels, respectively. MutSpace shows state-of-the-art performance in a breast cancer subclass classification problem. The high performance of these two applications means that StarSpace is likely to perform well in other bioinformatics problems.

A constrained semantic change search (**CSCS**) is a method for discovering word changes that significantly alter the semantics from an original sentence based on embedding techniques [93]. The key feature of this method is that it does not detect word changes that would abolish the grammar of the sentence but those that preserve the grammatical structure. For example, in an NLP task, CSCS can change “winegrowers revel in *good* season” to “winegrowers revel in *flu* season.” We briefly introduce the CSCS method. We define x and $\hat{x}$ as the original and mutated sentences, respectively. The embedded representations of x and $\hat{x}$ are defined as z and $\hat{z}$, respectively. Here, the semantic change is modeled as the distance between these embedded representations, that is, $||z-\hat{z}||$. Additionally, the preservation of the grammatical structure is evaluated by $p(\hat{x}|x)$, which is also modeled using embedding techniques. Finally, $\hat{x}$ maximizing $||z-\hat{z}|| + \beta p(\hat{x}|x)$, where β is a scaling factor. One biological application of CSCS is the modeling of viral evolution [56]. This application regarded viral proteins, preservation of the infectivity, and escape from antibody recognition as sentences, preservation of grammar, and semantic change, respectively, and detected escape mutations from immune systems as a result of the CSCS analysis. The analyses of HIV-1 and influenza viruses showed that mutations detected by the CSCS were in good agreement with the experimental mutation results.

Woloszynek *et al.* applied word2vec to a metagenomic dataset by regarding *4–15-mers* in sequencing reads as words [38]. They trained word2vec with a skip-gram model using 2,262,986 full-length 16S rRNA amplicon sequences from GreenGenes [94], a microbial 16S rRNA sequence database obtained using metagenomic analysis. They verified the robustness of the model in a taxonomic identification task using an independent dataset of 16,699 full-length 16S rRNA sequences from the KEGG REST server [95] as a validation dataset. The embedding features exhibited superior performance to the k-mer frequency features. In addition, the embedding as also performed using the American Gut project dataset [96], which has 11,341 partial 16S rRNA sequences from three body sites (gut, skin, and oral cavity), and showed comparable performance to conventional methods, such as sequence alignment in the body site classification task. These results suggest the availability of embedding with pre-trained models instead of sequence alignment for metagenomic sequence profiling.

4. Summary and outlook

In this study, we introduced basic algorithms and reviewed the recent literature concerning representation learning applications in sequence analysis. Heinzinger, *et al.* pointed out three difficulties in biological sequence modeling with NLP [62]as follows: (i) Proteins range from approximately 30 to 33,000 residues, which is much longer than the average English sentence, which consists of 15 to 30 words [97]; (ii) proteins use only 20 amino acids in most cases; if we consider one amino acid as a word, the word repertoire is 1/100,000 of English language, and if we consider 3-mer as a word, the word repertoire is 1/10 to 1/100 of English language; (iii) UniProt is more than ten times the size of Wikipedia, and extracting information from a huge biological database may require a commensurate model. Embedding biological sequences using NLP overcomes these difficulties and outperforms existing methods in many tasks, such as function, structure, localization, and disorder prediction (Table 1). In addition to these general biological tasks, representation learning has also been used to solve specific problems, such as RNA aptamer optimization [98], viral mutation prediction [56], and venom toxin prediction [36]. In these studies, representation learning of biological sequences could capture biophysical and biochemical properties of biological systems, and representation learning may reveal the grammar of life.

The development of novel representation learning methods has been actively studied in machine learning research. For example, hyperbolic embedding methods have been pursued in recent years [99, 100]. These methods embed the data not in Euclidean space, which is utilized in all the studies introduced in this paper, but in the *hyperbolic* space. The hyperbolic space has constant negative curvature; thus it shows characteristic geometric features not seen in Euclidean space, such as the sum of the interior angles of a triangle being less than 180°. Changes in the embedding space can considerably alter the efficiency of representation learning, and theoretical and experimental analyses have shown that hyperbolic embedding methods are suitable for data with hierarchical latent structure. Therefore, hyperbolic embedding methods have recently been used for biological analysis, such as phylogenetic analyses [101] and single-cell RNA-seq analyses [102]. Furthermore, research on embedding into more complex spaces, such as mixed-curvature spaces, has also attracted attention [103]. The application of these embedding techniques in non-Euclidean space for biological sequence analyses should be an essential research direction in the future.

New approaches are released every day in this field, and the scientific community is trying to compare their accuracy and validate their uses [65, 104, 105]. Therefore, it is important to make the models available in an easy-to-use form with documentation. In addition, considering the rapid growth of biological databases, the source code for creating models should be made available for future updates. Only a limited number of studies have released both the source code and the pre-trained model with the relevant documentation. Participants in this community need to publish their papers in a form that can be reproduced and verified.

In this study, we comprehensively surveyed and reviewed the application of representation learning to biological sequence analysis. Although NLP-based biological sequence analysis is still in its early stages and requires further development, in the light of novel challenges in biology, such as single-cell analysis, genome design, and personalized medicine, representation learning may help the progress of bioinformatics studies.

Acknowledgements

The illustrations in Figure 1 were kindly provided by Kae Namie. This work was supported by the Ministry of Education, Culture, Sports, Science, and Technology (KAKENHI) [grant numbers: JP17K20032, JP16H05879, JP16H06279 JP19H01152 and JP20H00624 to MH, JP19K20395 to TF, JP19J20117 to SH and JP20J20016 to TM] and JST CREST [grant numbers: JPMJCR1881 and JPMJCR21F1 to MH].

References

[1] F. Cunningham, P. Achuthan, W. Akanni, J. Allen, M. R. Amode, I. M. Armean, R. Bennett, J. Bhai, K. Billis, S. Boddu, C. Cummins, C. Davidson, K. J. Dodiya, A. Gall, C. G. Girón, L. Gil, T. Grego, L. Haggerty, E. Haskell, T. Hourlier, O. G. Izuogu, S. H. Janacek, T. Juettemann, M. Kay, M. R. Laird, I. Lavidas, Z. Liu, J. E. Loveland, J. C. Marugán, T. Maurel, A. C. McMahon, B. Moore, J. Morales, J. M. Mudge, M. Nuhn, D. Ogeh, A. Parker, A. Parton, M. Patricio, A. I. Abdul Salam, B. M. Schmitt, H. Schuilenburg, D. Sheppard, H. Sparrow, E. Stapleton, M. Szuba, K. Taylor, G. Threadgold, A. Thormann, A. Vullo, B. Walts, A. Winterbottom, A. Zadissa, M. Chakiachvili, A. Frankish, S. E. Hunt, M. Kostadima, N. Langridge, F. J. Martin, M. Muffato, E. Perry, M. Ruffier, D. M. Staines, S. J. Trevanion, B. L. Aken, A. D. Yates, D. R. Zerbino,

P. Flicek, Ensembl 2019., Nucleic Acids Res. 47 (D1) (2019) D745–D751. doi:10.1093/nar/gky1113.

[2] C. Angermueller, T. Pärnamaa, L. Parts, O. Stegle, Deep learning for computational biology., Mol. Syst. Biol. 12 (7) (2016) 878. doi:10.15252/msb.20156651.

[3] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani, A. Telenti, A primer on deep learning in genomics., Nat. Genet. 51 (1) (2019) 12–18. doi:10.1038/s41588-018-0295-5.

[4] G. G. Chowdhury, Natural language processing, Annu. Rev. Inf. Sci. Technol. 37 (1) (2005) 51–89. doi:10.1002/aris.1440370103.

[5] L. Yu, D. K. Tanwar, E. D. S. Penha, Y. I. Wolf, E. V. Koonin, M. K. Basu, Grammar of protein domain architectures., Proc. Natl. Acad. Sci. U. S. A. 116 (9) (2019) 3636–3645. doi:10.1073/pnas.1814684116.

[6] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives., IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–828. arXiv:1206.5538, doi:10.1109/TPAMI.2013.50.

[7] Z. S. Harris, Distributional Structure, WORD 10 (2-3) (1954) 146–162. doi:10.1080/00437956.1954.11659520.

[8] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc. (2013). arXiv:1301.3781.

[9] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, Adv. Neural Inf. Process. Syst. (2013). arXiv:1310.4546.

[10] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of Machine Learning Research 3 (2003) 1137–1155.

[11] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 160–167. doi:10.1145/1390156.1390177.

[12] J. Pennington, R. Socher, C. Manning, Glove: Global Vectors for Word Representation, in: Proc. 2014 Conf. Empir. Methods Nat. Lang. Process., Association for Computational Linguistics, Stroudsburg, PA, USA, 2014, pp. 1532–1543. doi:10.3115/v1/D14-1162.
URL http://aclweb.org/anthology/D14-1162

[13] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146. arXiv:1607.04606, doi:10.1162/tacl_a_00051.

[14] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: 31st Int. Conf. Mach. Learn. ICML 2014, 2014. arXiv:1405.4053.

[15] W. Weaver, Translation, Machine translation of languages 14 (15-23) (1955) 10.

[16] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.

[17] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep Contextualized Word Representations, in: Proc. 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Vol. 1 (Long Pap., Association for Computational Linguistics, Stroudsburg, PA, USA, 2018, pp. 2227–2237. arXiv:1802.05365, doi:10.18653/v1/N18-1202.
URL http://aclweb.org/anthology/N18-1202

[18] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, in: Eleventh annual conference of the international speech communication association, 2010.
URL https://www.isca-speech.org/archive/interspeech_2010/i10_1045.html

[19] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to Sequence Learning with Neural Networks, Adv. Neural Inf. Process. Syst. (2014). arXiv:1409.3215.

[20] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. (2014). arXiv:1409.0473.

[21] Y. Kim, C. Denton, L. Hoang, A. M. Rush, Structured Attention Networks, arXiv (2017). arXiv:1702.00887.

[22] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional Sequence to Sequence Learning, 34th Int. Conf. Mach. Learn. ICML 2017 (2017). arXiv:1705.03122.

[23] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, End-to-end memory networks, in: Adv. Neural Inf. Process. Syst., 2015. arXiv:1503.08895.

[24] A. H. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, J. Weston, Key-value memory networks for directly reading documents, in: EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc., 2016. arXiv:1606.03126, doi:10.18653/v1/d16-1147.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, Adv. Neural Inf. Process. Syst. (2017). arXiv:1706.03762.

[26] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf. (2018). arXiv:1810.04805.

[27] M.-T. Luong, H. Pham, C. D. Manning, Effective Approaches to Attention-based Neural Machine Translation, arXiv:1508.04025 [cs] (Sep. 2015). arXiv:1508.04025.

[28] L. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, J. Weston, StarSpace: Embed All The Things! (2017). arXiv:1709.03856.

[29] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, arXiv preprint arXiv:1906.08237 (2019). arXiv:1906.08237.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, arXiv preprint arXiv:1910.10683 (2019). arXiv:1910.10683.

[31] E. Asgari, M. R. K. Mofrad, Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics., PLoS One 10 (11) (2015) e0141287. doi:10.1371/journal.pone.0141287.

[32] Y. S. Vang, X. Xie, HLA class I binding prediction via convolutional neural networks., Bioinformatics 33 (17) (2017) 2658–2665. arXiv:1701.00593, doi:10.1093/bioinformatics/btx264.

[33] S. M. Islam, B. J. Heil, C. M. Kearney, E. J. Baker, Protein classification using modified n-grams and skip-grams, Bioinformatics (2018). doi:10.1093/bioinformatics/btx823.

[34] Q. Zou, P. Xing, L. Wei, B. Liu, Gene2vec: gene subsequence embedding for prediction of mammalian N6-methyladenosine sites from mRNA., RNA 25 (2) (2019) 205–218. doi:10.1261/rna.069112.118.

[35] M. K. Mejía-Guerra, E. S. Buckler, A k-mer grammar analysis to uncover maize regulatory architecture., BMC Plant Biol. 19 (1) (2019) 103. doi:10.1186/s12870-019-1693-2.

[36] E. Asgari, A. C. McHardy, M. R. K. Mofrad, Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX)., Sci. Rep. 9 (1) (2019) 3577. doi:10.1038/s41598-019-38746-w.

[37] P. Phloyphisut, N. Pornputtapong, S. Sriswasdi, E. Chuangsuwanich, MHCSeqNet: a deep neural network model for universal MHC binding prediction., BMC Bioinformatics 20 (1) (2019) 270. doi:10.1186/s12859-019-2892-4.

[38] S. Woloszynek, Z. Zhao, J. Chen, G. L. Rosen, 16S rRNA sequence embeddings: Meaningful numeric feature representations of nucleotide sequences that are convenient for downstream analyses., PLoS Comput. Biol. 15 (2) (2019) e1006721. doi:10.1371/journal.pcbi.1006721.

[39] R. Menegaux, J.-P. Vert, Continuous Embeddings of DNA Sequencing Reads and Application to Metagenomics., J. Comput. Biol. 26 (6) (2019) 509–518. doi:10.1089/cmb.2018.0174.

[40] H. Shim, Feature Learning of Virus Genome Evolution With the Nucleotide Skip-Gram Neural Network, Evol. Bioinforma. 15 (2019)

117693431882107. doi:10.1177/1176934318821072.

[41] M. Tsubaki, K. Tomii, J. Sese, Compound-protein interaction prediction with end-to-end learning of neural networks for graphs and sequences, Bioinformatics (2019). doi:10.1093/bioinformatics/bty535.

[42] T.-T.-D. Nguyen, N.-Q.-K. Le, Q.-T. Ho, D.-V. Phan, Y.-Y. Ou, Using word embedding technique to efficiently represent protein sequences for identifying substrate specificities of transporters., Anal. Biochem. 577 (2019) 73–81. doi:10.1016/j.ab.2019.04.011.

[43] Y. Guo, Y. Yang, Y. Huang, H. B. Shen, Discovering nuclear targeting signal sequence through protein language learning and multivariate analysis, Anal. Biochem. (2020). doi:10.1016/j.ab.2019.113565.

[44] D. W. A. Buchan, D. T. Jones, Learning a functional grammar of protein domains using natural language word embedding techniques., Proteins 88 (4) (2020) 616–624. doi:10.1002/prot.25842.

[45] C. Wang, Y. Zhang, S. Han, Its2vec: Fungal Species Identification Using Sequence Embedding and Random Forest Classification., Biomed Res. Int. 2020 (2020) 2468789. doi:10.1155/2020/2468789.

[46] A. Wahab, H. Tayara, Z. Xuan, K. T. Chong, DNA sequences performs as natural language processing by exploiting deep learning algorithm for the identification of N4-methylcytosine., Sci. Rep. 11 (1) (2021) 212. doi:10.1038/s41598-020-80430-x.

[47] K. K. Yang, Z. Wu, C. N. Bedbrook, F. H. Arnold, Learned protein embeddings for machine learning, Bioinformatics (2018). doi:10.1093/bioinformatics/bty178.

[48] W. Zeng, M. Wu, R. Jiang, Prediction of enhancer-promoter interactions via natural language processing., BMC Genomics 19 (Suppl 2) (2018) 84. doi:10.1186/s12864-018-4459-6.

[49] Y.-J. Tang, Y.-H. Pang, B. Liu, IDP-Seq2Seq: Identification of Intrinsically Disordered Regions based on Sequence to Sequence Learning., Bioinformatics (2020). doi:10.1093/bioinformatics/btaa667.

[50] X. Min, W. Zeng, N. Chen, T. Chen, R. Jiang, Chromatin accessibility prediction via convolutional long short-term memory networks with k-mer embedding., Bioinformatics 33 (14) (2017) i92–i101. doi:10.1093/bioinformatics/btx234.

[51] Y. Ju, L. Yuan, Y. Yang, H. Zhao, CircSLNN: Identifying RBP-Binding Sites on circRNAs via Sequence Labeling Neural Networks., Front. Genet. 10 (2019) 1184. doi:10.3389/fgene.2019.01184.

[52] N. Q. K. Le, E. K. Y. Yapp, N. Nagasundaram, H.-Y. Yeh, Classifying Promoters by Interpreting the Hidden Information of DNA Sequences via Deep Learning and Combination of Continuous FastText N-Grams, Front. Bioeng. Biotechnol. 7 (2019). doi:10.3389/fbioe.2019.00305.

[53] T.-T.-D. Nguyen, N.-Q.-K. Le, Q.-T. Ho, D.-V. Phan, Y.-Y. Ou, TNF-Pred: identifying tumor necrosis factors using hybrid features based on word embeddings., BMC Med. Genomics 13 (Suppl 10) (2020) 155. doi:10.1186/s12920-020-00779-w.

[54] N. Q. K. Le, D. T. Do, T. N. K. Hung, L. H. T. Lam, T.-T. Huynh, N. T. K. Nguyen, A Computational Framework Based on Ensemble Deep Neural Networks for Essential Genes Identification., Int. J. Mol. Sci. 21 (23) (2020). doi:10.3390/ijms21239070.

[55] S. Yang, X. Liu, R. T. Ng, ProbeRating: a recommender system to infer binding profiles for nucleic acid-binding proteins., Bioinformatics 36 (18) (2020) 4797–4804. doi:10.1093/bioinformatics/btaa580.

[56] B. Hie, E. D. Zhong, B. Berger, B. Bryson, Learning the language of viral evolution and escape., Science 371 (6526) (2021) 284–288. doi:10.1126/science.abd7331.

[57] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning., Nat. Methods 16 (12) (2019) 1315–1322. doi:10.1038/s41592-019-0598-1.

[58] N. Strodthoff, P. Wagner, M. Wenzel, W. Samek, UDSMProt: universal deep sequence models for protein classification., Bioinformatics 36 (8) (2020) 2401–2409. doi:10.1093/bioinformatics/btaa003.

[59] J. Vielhaben, M. Wenzel, W. Samek, N. Strodthoff, USMPep: universal sequence models for major histocompatibility complex binding affinity prediction., BMC Bioinformatics 21 (1) (2020) 279. doi:10.1186/s12859-020-03631-1.

[60] H. Yuan, M. Kshirsagar, L. Zamparo, Y. Lu, C. S. Leslie, BindSpace decodes transcription factor binding signals by large-scale sequence embedding, Nat. Methods (2019). doi:10.1038/s41592-019-0511-y.

[61] Y. Zhang, Y. Xiao, M. Yang, J. Ma, Cancer mutational signatures representation by large-scale context embedding., Bioinformatics 36 (Supplement_1) (2020) i309–i316. doi:10.1093/bioinformatics/btaa433.

[62] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, B. Rost, Modeling aspects of the language of life through transfer-learning protein sequences., BMC Bioinformatics 20 (1) (2019) 723. doi:10.1186/s12859-019-3220-8.

[63] J. Clauwaert, W. Waegeman, Novel transformer networks for improved sequence labeling in genomics, IEEE/ACM Trans. Comput. Biol. Bioinforma. (2020) 1–1doi:10.1109/TCBB.2020.3035021.

[64] J. A. Valeri, K. M. Collins, P. Ramesh, M. A. Alcantar, B. A. Lepe, T. K. Lu, D. M. Camacho, Sequence-to-function deep learning frameworks for engineered riboregulators., Nat. Commun. 11 (1) (2020) 5058. doi:10.1038/s41467-020-18676-2.

[65] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, Y. Song, Evaluating protein transfer learning with tape, Advances in Neural Information Processing Systems 32 (2019).

[66] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, D. Guo, M. Ott, C. L. Zitnick, J. Ma, R. Fergus, Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, bioRxiv (2020). doi:10.1101/622803.

[67] A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik, B. Rost, ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing, bioRxiv (2020). doi:10.1101/2020.07.12.199554.

[68] Y. Ji, Z. Zhou, H. Liu, R. V. Davuluri, DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome, Bioinformatics (02 2021). doi:10.1093/bioinformatics/btab083.

[69] I. Deznabi, B. Arabaci, M. Koyutürk, O. Tastan, DeepKinZero: zero-shot learning for predicting kinase-phosphosite associations involving understudied kinases, Bioinformatics 36 (12) (2020) 3652–3661. doi:10.1093/bioinformatics/btaa013.

[70] Y. Cai, J. Wang, L. Deng, SDN2GO: An integrated deep learning model for protein function prediction, Front Bioeng Biotechnol 8 (2020) 391. doi:10.3389/fbioe.2020.00391.

[71] D. Kimothi, A. Soni, P. Biyani, J. M. Hogan, Distributed representations for biological sequence analysis, arXiv (2016) 1608.05949.

[72] P. Ng, dna2vec: Consistent vector representations of variable-length k-mers, arXiv (2017) 1701.06279.

[73] M. S. Klausen, M. C. Jespersen, H. Nielsen, K. K. Jensen, V. I. Jurtz, C. K. Sønderby, M. O. A. Sommer, O. Winther, M. Nielsen, B. Petersen, et al., Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning, Proteins: Structure, Function, and Bioinformatics 87 (6) (2019) 520–527.

[74] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, O. Winther, Deeploc: prediction of protein subcellular localization using deep learning, Bioinformatics 33 (21) (2017) 3387–3395.

[75] A. Villegas-Morcillo, S. Makrodimitris, R. C. H. J. van Ham, A. M. Gomez, V. Sanchez, M. J. T. Reinders, Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function., Bioinformatics (2020). doi:10.1093/bioinformatics/btaa701.

[76] S. Merity, N. S. Keskar, R. Socher, Regularizing and optimizing lstm language models, arXiv (2017).

[77] J. Vig, A. Madani, L. R. Varshney, C. Xiong, richard socher, N. Rajani, BERTology meets biology: Interpreting attention in protein language models, International Conference on Learning Representations (2021).

[78] R. Rao, J. Meier, T. Sercu, S. Ovchinnikov, A. Rives, Transformer protein language models are unsupervised structure learners, International Conference on Learning Representations (2021).

[79] H. Liu, J. Sun, J. Guan, J. Zheng, S. Zhou, Improving compound–protein interaction prediction by building up highly credible negative samples, Bioinformatics 31 (12) (2015) i221–i229. doi:10.1093/bioinformatics/btv256.

[80] M. M. Mysinger, M. Carchia, J. J. Irwin, B. K. Shoichet, Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking, Journal of medicinal chemistry 55 (14) (2012) 6582–6594. doi:10.1021/jm300687e.

[81] Uniprot: The universal protein knowledgebase in 2021, Nucleic Acids Research 49 (D1) (2021) D480–D489. doi:10.1093/nar/gkaa1100.

[82] D. Ray, H. Kazan, K. B. Cook, M. T. Weirauch, H. S. Najafabadi, X. Li, S. Gueroussov, M. Albu, H. Zheng, A. Yang, et al., A compendium of rna-binding motifs for decoding gene regulation, Nature 499 (7457) (2013) 172–177. doi:10.1038/nature12311.

[83] M. T. Weirauch, A. Yang, M. Albu, A. G. Cote, A. Montenegro-Montero, P. Drewe, H. S. Najafabadi, S. A. Lambert, I. Mann, K. Cook, et al., Determination and inference of eukaryotic transcription factor sequence specificity, Cell 158 (6) (2014) 1431–1443. doi:10.1016/j.cell.2014.08.009.

[84] R. Pelossof, I. Singh, J. L. Yang, M. T. Weirauch, T. R. Hughes, C. S. Leslie, Affinity regression predicts the recognition code of nucleic acid–binding proteins, Nature biotechnology 33 (12) (2015) 1242–1249. doi:10.1038/nbt.3343.

[85] S. Yang, J. Wang, R. T. Ng, Inferring rna sequence preferences for poorly studied rna-binding proteins based on co-evolution, BMC bioinformatics 19 (1) (2018) 1–12. doi:10.1186/s12859-018-2091-8.

[86] J. A. Valeri, K. M. Collins, P. Ramesh, M. A. Alcantar, B. A. Lepe, T. K. Lu, D. M. Camacho, Sequence-to-function deep learning frameworks for engineered riboregulators, Nature communications 11 (1) (2020) 1–14. doi:10.1038/s41467-020-18676-2.

[87] N. M. Angenent-Mari, A. S. Garruss, L. R. Soenksen, G. Church, J. J. Collins, A deep learning approach to programmable rna switches, Nature communications 11 (1) (2020) 1–12. doi:10.1038/s41467-020-18677-1.

[88] A. A. Green, P. A. Silver, J. J. Collins, P. Yin, Toehold switches: de-novo-designed regulators of gene expression, Cell 159 (4) (2014) 925–939. doi:10.1016/j.cell.2014.10.002.

[89] K. Pardee, A. A. Green, M. K. Takahashi, D. Braff, G. Lambert, J. W. Lee, T. Ferrante, D. Ma, N. Donghia, M. Fan, et al., Rapid, low-cost detection of zika virus using programmable biomolecular components, Cell 165 (5) (2016) 1255–1266. doi:10.1016/j.cell.2016.04.059.

[90] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning, Nature methods 16 (12) (2019) 1315–1322. doi:10.1038/s41592-019-0598-1.

[91] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, U. Consortium, Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches, Bioinformatics 31 (6) (2015) 926–932. doi:10.1093/bioinformatics/btu739.

[92] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez, et al., Local fitness landscape of the green fluorescent protein, Nature 533 (7603) (2016) 397–401. doi:10.1038/nature17995.

[93] B. Hie, E. Zhong, B. Bryson, B. Berger, Learning mutational semantics, Advances in Neural Information Processing Systems 33 (2020).

[94] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, G. L. Andersen, Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB., Appl. Environ. Microbiol. 72 (7) (2006) 5069–72. doi:10.1128/AEM.03006-05.

[95] D. Tenenbaum, Keggrest: Client-side rest access to kegg, R package version 1 (1) (2016).

[96] D. McDonald, E. Hyde, J. W. Debelius, J. T. Morton, A. Gonzalez, G. Ackermann, A. A. Aksenov, B. Behsaz, C. Brennan, Y. Chen, L. DeRight Goldasich, P. C. Dorrestein, R. R. Dunn, A. K. Fahimipour, J. Gaffney, J. A. Gilbert, G. Gogul, J. L. Green, P. Hugenholtz, G. Humphrey, C. Huttenhower, M. A. Jackson, S. Janssen, D. V. Jeste, L. Jiang, S. T. Kelley, D. Knights, T. Kosciolek, J. Ladau, J. Leach, C. Marotz, D. Meleshko, A. V. Melnik, J. L. Metcalf, H. Mohimani, E. Montassier, J. Navas-Molina, T. T. Nguyen, S. Peddada, P. Pevzner, K. S. Pollard, G. Rahnavard, A. Robbins-Pianka, N. Sangwan, J. Shorenstein, L. Smarr, S. J. Song, T. Spector, A. D. Swafford, V. G. Thackray, L. R. Thompson, A. Tripathi, Y. Vázquez-Baeza, A. Vrbanac, P. Wischmeyer, E. Wolfe, Q. Zhu, American Gut Consortium, R. Knight, American Gut: an Open Platform for Citizen Science Microbiome Research., mSystems 3 (3) (may 2018). doi:10.1128/mSystems.00031-18.

[97] E. Schils, Characteristics of Sentence Length in Running Text, Lit. Linguist. Comput. 8 (1) (1993) 20–26. doi:10.1093/llc/8.1.20.

[98] N. Iwano, T. Adachi, K. Aoki, Y. Nakamura, M. Hamada, RaptGen: A variational autoencoder with profile hidden Markov model for generative aptamer discovery, bioRxiv (2021) 2021.02.17.431338doi:10.1101/2021.02.17.431338.

[99] M. Nickel, D. Kiela, Poincaré embeddings for learning hierarchical representations, Advances in Neural Information Processing Systems (2017).

[100] O.-E. Ganea, G. Bécigneul, T. Hofmann, Hyperbolic neural networks, Advances in Neural Information Processing Systems (2018).

[101] H. Matsumoto, T. Mimori, T. Fukunaga, Novel metric for hyperbolic phylogenetic tree embeddings, bioRxiv (2020). doi:10.1101/2020.10.09.334243.

[102] A. Klimovskaia, D. Lopez-Paz, L. Bottou, M. Nickel, Poincaré maps for analyzing complex hierarchies in single-cell data., Nat. Commun. 11 (1) (2020) 2966. doi:10.1038/s41467-020-16822-4.

[103] A. Gu, F. Sala, B. Gunel, C. Ré, Learning mixed-curvature representations in product spaces, in: International Conference on Learning Representations, 2018.
URL https://openreview.net/forum?id=HJxeWnCcF7

[104] D. Duong, A. Uppunda, L. Gai, C. Ju, J. Zhang, M. Chen, E. Eskin, J. J. Li, K.-W. Chang, Evaluating Representations for Gene Ontology Terms, bioRxiv (2020). doi:10.1101/765644.

[105] S. Unsal, H. Ataş, M. Albayrak, K. Turhan, A. C. Acar, T. Doğan, Evaluation of Methods for Protein Representation Learning: A Quantitative Analysis (2020). doi:10.1101/2020.10.28.359828.