Pretraining model for biological sequence data

Bosheng Song, Zimeng Li, Xuan Lin, Jianmin Wang, Tian Wang and Xiangzheng Fu

Corresponding author: Xiangzheng Fu, College of Information Science and Engineering, Hunan University, Changsha, Hunan, China.
Tel: 86-0731-88821907; E-mail: fxz326@hnu.edu.cn

Abstract
With the development of high-throughput sequencing technology, biological sequence data reflecting life information becomes increasingly accessible. Particularly on the background of the COVID-19 pandemic, biological sequence data play an important role in detecting diseases, analyzing the mechanism and discovering specific drugs. In recent years, pretraining models that have emerged in natural language processing have attracted widespread attention in many research fields not only to decrease training cost but also to improve performance on downstream tasks. Pretraining models are used for embedding biological sequence and extracting feature from large biological sequence corpus to comprehensively understand the biological sequence data. In this survey, we provide a broad review on pretraining models for biological sequence data. Moreover, we first introduce biological sequences and corresponding datasets, including brief description and accessible link. Subsequently, we systematically summarize popular pretraining models for biological sequences based on four categories: CNN, word2vec, LSTM and Transformer. Then, we present some applications with proposed pretraining models on downstream tasks to explain the role of pretraining models. Next, we provide a novel pretraining scheme for protein sequences and a multitask benchmark for protein pretraining models. Finally, we discuss the challenges and future directions in pretraining models for biological sequences.

Key words: biological sequence; pretraining model; deep learning

Introduction
Biological sequence data composed of protein, DNA, and RNA sequences are an important field in life science. Based on scientific research, biological sequence data imply life rules and offer an excellent window to explore biochemical roles [1]. Learning biological sequences by deep learning methods, researchers can not only infer the biological properties of unseen sequences but also predict interactions without understanding the underlying physical or biological mechanisms [2]. In particular, during the coronavirus disease 2019 (COVID-19) pandemic, many researchers explore related issues based on biological sequences [3–8]. However, considering that biological sequences are long and nonnumeric, finding a suitable way to convert biological sequences into processable representation is difficult. Moreover, the lack of labeled biological sequences affects their performance on corresponding tasks. Recently, unsupervised learning [9, 10] on biological sequences has attracted many researchers, among which pretraining model for biological sequence data is a field in great demand. The pretraining model is a saved model that has been trained in advance. In general, pretraining models are first trained on large datasets to be fitted and generalized. Subsequently, the trained parameters and weights of pretraining models will be saved. Finally, saved pretraining models are applied in other tasks directly or after fine-tuning on other datasets. During the development of the pretraining model for biological sequences, many different models have made contributions to this field. Convolutional neural network (CNN) [8, 11–13] models extract
Figure 1. Scheme of using pretraining models for biological sequences. It illustrates the process of using pretraining models for biological sequences and consists of input biological sequences, output representing vectors through pretraining models and downstream applications. Biological macromolecules are generated in Pymol.


Biological sequence can be regarded as a special life language, similar to human natural language. Thanks to the advancement of natural language processing (NLP) [19], pretraining models can effectively extract the characteristics of biological sequences and encode biological sequences after training in a large unlabeled corpus. As shown in Figure 1, pretraining language models and word embedding methods in NLP can embed biological sequence with processable low-dimensional representation, which improve the performance when applied in downstream tasks [20, 21]. Consequently, pretraining models using NLP methods have captured syntactic and semantic information in biological sequences.

As shown in many recent works, pretraining models have achieved many biological sequence tasks. Pretraining models for biological sequences have learnt context-sensitive representation from various unlabeled biological sequences, which implicitly reflect general knowledge in biological sequences. Using pretraining models, the knowledge learned from open field can be transferred to downstream tasks such as drug–target interaction (DTI) [22–24], enhancer–promoter interaction (EPI) [25] and protein classification [26–29], making most of the methods perform better with less cost. In addition, features extracted from several unlabeled sequences are beneficial to tasks without enough label data. Moreover, pretraining models can be used for transfer learning. Such models are initially pretrained on different datasets and then fine-tuned on the target datasets for specific tasks, which effectively optimize the network, improve performance and save time with good scalability.

In this survey, we review literature on pretraining models for biological sequence data to make readers understand the area approximately and enlighten other researchers in this area. This survey is important to novices and researchers who are looking for an alternative for improving this area, when it comes to using pretraining models to learn biological sequence data. The key contributions of this survey are as follows: (1) we summarize and briefly introduce some important biological sequence datasets appearing in these works. (2) We conduct a systematic review on pretraining models for biological sequences and organize the current methods by different basic methods. (3) We have introduced some application and methods for downstream tasks with proposed pretraining models. (4) We provide an important scheme and benchmark, which are proposed by previous studies, for protein pretraining models. (5) We discuss the challenges and future directions of pretraining models for biological sequence data, which may provide new ideas for researchers and promote development in the field.

The remaining of this review is organized as follows: we provide introduction to biological sequence data and related popular databases in Overview of biological sequence data. Pretraining models for biological sequences are summarized in Pretraining model. Some applications on downstream tasks with proposed pretraining models for biological sequences are illustrated in Application of pretraining model. Scheme and benchmark offers a novel pretraining model scheme and benchmark for protein sequences. Challenges and future directions discusses the challenges in the current methods and future directions in this field.

Overview of biological sequence data

Biological sequence data contain abundant biometric information, which are stored in the sequence structure [31]. Exploring hidden roles in healthy and diseases with biological
sequences is important. In addition, with the advancement of next-generation sequencing technology, a growing number of sequences can be obtained and invested in various deep learning research tasks. Consequently, in the past few years, many biological sequence datasets are proposed for different tasks in various published papers.

In this section, we introduce some details about biological sequence data and popular historical databases that are used frequently in research. We also summarize the biological sequence databases and datasets that are used by the surveyed papers in Table 1. The brief description and accessible URL are supposed to provide a convenient way for novices and researchers, which help them to obtain the necessary databases and datasets.

**Biological sequence data**

In general, biological sequence is the long sequence represented by a string of different and fixed alphabets, in which different alphabets usually represent different macromolecules. For example, a DNA sequence is made up of a four-letter alphabet ‘A’, ‘C’, ‘T’ and ‘G’, which represent different kinds of deoxynucleotide in DNA [52].

DNA sequence, RNA sequence and protein sequence are collectively called biological sequence. Proteins are important fundamental macromolecules in the human body, which have vital functions in life activities. Proteins are always folded into a unique three-dimensional structure by several amino acid chains. Protein functions have been determined by protein structure and sequences for the greater part. Due to the specific and various 3D conformations, proteins with amino acid sequences have specific and wide array of functions, such as transmitting nerve pulses and binding specificity [53]. Consequently, amino acid sequence is often the key research object to explore protein properties and interactions, such as DTI [22], compound-protein interaction (CPI) [54], protein classification [26] and protein function prediction [55].

Similar to protein, DNA is also a bioactive macromolecule essential for the development and normal operation of organisms in biological cells. Given its vital genetic information, DNA not only controls the biological inheritance and activities but also serves as the basis of RNA and protein synthesis. Different from the folding structure of protein, the molecular structure of DNA is double helix formed by two polydeoxynucleotide chains [56]. In bioinformatics, DNA fragment sequences are usually used for scientific research, such as exploring the interactions between promoter and enhancer [25, 44]. As a genetic information carrier, RNA is another common biological macromolecule. Most of RNAs are single stranded, which are made up of a ribonucleotide chain [57]. RNA is closely related to proteins, which can control protein synthesis. Thus, RNA sequences can reflect life information from another point of view in bioinformatics [58].

**Databases for biological sequence**

Protein Data Bank (PDB) [32] is the important collection of biological macromolecules, which is a 2.5-dimensional structure database, which preserves crucial information on various biological macromolecules, such as atomic coordinates, sequences, references and level 1 and level 2 structural information. The Structural Classification of Proteins (SCOP) [30] and Structural Classification of Proteins—extended (SCOPe) [29] are commonly used protein structural databases that classify known proteins by family, superfamily, common fold and class, thereby providing rich information about proteins, such as structural, sequence data and evolutionary relationships. Pfam [35] is a well-known protein family database that divides proteins into different families by multiple sequence comparisons and hidden Markov models. Because of the protein family characteristics of Pfam, researchers always use Pfam as data sources when proteins with similar functions are needed by tasks.

Universal Protein (UniProt) [36] is the informative protein database that integrates the resources of three major databases: EBI, SIB and PIR. With the continuous increase of protein sequence data, UniProt has more than 120 million protein sequences and annotations. SWISS-PROT [33] is the complete annotated protein sequence database. The advantage of SWISS-PROT is the detailed annotation information and standardized nomenclature of protein sequences, which provide annotated protein sequences for various tasks. UniRef [37] has been a frequently used protein sequence database since its first release in 2004. On the basis of different sequence identity levels, UniRef is divided into three different protein sequence subsets: UniRef100, UniRef90 and UniRef50, which meet different task requirements.

DrugBank [45] is a bioinformatic–cheminformatic database that combines the information of drugs and targets. For drugs, DrugBank provides drug chemical structures, pharmacological effects, protein targets, drug–drug interactions, etc. As for protein targets, DrugBank stores related information, such as protein sequence, structure and approach. ChEMBL [48] is another outstanding database that provides reliable information of compound and target, which obtains bioactivity data and structures for small molecules from a variety of journals. Similar to ChEMBL, BindingDB [46] is also an open database that extracts data from scientific literature. BindingDB database primarily provides binding affinities between compound and target protein, focusing on drug-target proteins.

**Pretraining model**

With the development of NLP technology, pretraining models have gradually become a hot field in deep learning. Owing to the improvement of software and proposed new methods, pretraining models have substantially achieved state-of-the-art results in almost all NLP tasks [59]. Several methods have made contributions to biological sequence-related tasks as a pretraining model, to improve the performance and speed up the training process. In the early years, neural network models [11, 14, 15] have occupied the mainstream of pretraining models, which generate word vectors for representing sequences. In recent years, with the introduction of attention mechanism and development of Transformer-based methods [17, 18], pretraining language models succeeded in many fields. Table 2 summarizes four types of popular pretraining models and their brief description. In this section, we introduce four categories of pretraining models for biological sequence data that have been used in surveyed papers: CNN, word2vec, LSTM and Transformer.

**Convolutional neural network**

CNN [11], one of the classic neural network structures in deep learning, has outstanding performance in many fields. Inspired by local receptive field mechanism, CNN uses convolution operations to extract features with other network structure to crop features and transform output. The specific architecture of CNN determines the unique advantages in the Computer Vision (CV)
**Table 1. List of biological sequence databases and datasets**

<table>
<thead>
<tr>
<th>Category</th>
<th>Dataset</th>
<th>Year</th>
<th>Entities</th>
<th>Description</th>
<th>URL (Source)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protein</td>
<td>PDB [32]</td>
<td>1971</td>
<td>2.5-dimensional structure of biological macromolecules</td>
<td>Protein structure database, containing 3D structures obtained through experiments</td>
<td><a href="http://www.rcsb.org">http://www.rcsb.org</a></td>
</tr>
<tr>
<td></td>
<td>SCOP [34]</td>
<td>1994</td>
<td>Protein sequences and structures</td>
<td>Database of protein structure classification according to the spatial characteristics of protein domains</td>
<td><a href="http://scop2.mrc-lmb.cam.ac.uk">http://scop2.mrc-lmb.cam.ac.uk</a></td>
</tr>
<tr>
<td></td>
<td>UniProt [36]</td>
<td>2002</td>
<td>Protein sequence</td>
<td>A database consisting of a large number of labeled and unlabeled primary protein sequences</td>
<td><a href="http://www.uniprot.org">http://www.uniprot.org</a></td>
</tr>
<tr>
<td></td>
<td>UniRef [37]</td>
<td>2004</td>
<td>Protein sequence</td>
<td>Unlabeled big data protein sequence</td>
<td><a href="http://www.uniprot.org">http://www.uniprot.org</a></td>
</tr>
<tr>
<td></td>
<td>DisProt [38]</td>
<td>2007</td>
<td>Protein sequence</td>
<td>The database of disordered proteins</td>
<td><a href="http://www.disprot.org">http://www.disprot.org</a></td>
</tr>
<tr>
<td></td>
<td>SCOPe [39]</td>
<td>2012</td>
<td>Protein structural relationships</td>
<td>59 514 protein database (PDB) entries, including more than 65% of the protein structures in the PDB</td>
<td><a href="http://scop.berkeley.edu">http://scop.berkeley.edu</a></td>
</tr>
<tr>
<td></td>
<td>BFD [40]</td>
<td>2018</td>
<td>Protein sequences</td>
<td>Largest set of protein sequences</td>
<td><a href="https://metaclust.mmseqs.org/">https://metaclust.mmseqs.org/</a></td>
</tr>
<tr>
<td>Nucleic acid</td>
<td>GENCODE [42]</td>
<td>2003</td>
<td>Genome annotation</td>
<td>Documented the functional annotation of the genome</td>
<td><a href="https://www.gencodegenes.org">https://www.gencodegenes.org</a></td>
</tr>
<tr>
<td></td>
<td>STITCH [47]</td>
<td>2007</td>
<td>Compound–protein interaction</td>
<td>Interactions between more than 30 000 small molecule compounds and 2.6 million proteins from 1133 species</td>
<td><a href="http://stitch.embl.de">http://stitch.embl.de</a></td>
</tr>
<tr>
<td></td>
<td>ChEMBL [48]</td>
<td>2009</td>
<td>Drug–target associations</td>
<td>Collected 12 482 targets, 1.879 million compounds and a total of 155 million pieces of biological activity information</td>
<td><a href="https://www.ebi.ac.uk/chembl/">https://www.ebi.ac.uk/chembl/</a></td>
</tr>
<tr>
<td></td>
<td>HIPPIE [49]</td>
<td>2012</td>
<td>Protein–protein interactions</td>
<td>Human PPI dataset with standardized scoring</td>
<td><a href="http://cbdm.uni-mainz.de/hippie/">http://cbdm.uni-mainz.de/hippie/</a></td>
</tr>
<tr>
<td></td>
<td>KIBA [50]</td>
<td>2014</td>
<td>Target–ligand associations</td>
<td>467 targets and 52 498 ligands collected from ChEMBL and STITCH</td>
<td><a href="https://pubs.acs.org/doi/abs/10.1021/ci400709d">https://pubs.acs.org/doi/abs/10.1021/ci400709d</a></td>
</tr>
<tr>
<td></td>
<td>GLASS [51]</td>
<td>2015</td>
<td>GPCR-ligand associations</td>
<td>A large number of experimentally verified GPCR-ligand associations</td>
<td><a href="http://zhanglab.ccmb.med.umich.edu/GLASS/">http://zhanglab.ccmb.med.umich.edu/GLASS/</a>.</td>
</tr>
</tbody>
</table>
Table 2. Summary of popular pretraining models for biological sequences

<table>
<thead>
<tr>
<th>Category</th>
<th>Algorithm</th>
<th>Author</th>
<th>Year</th>
<th>Description</th>
<th>Source code</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>CNN [11]</td>
<td>Lecun et al.</td>
<td>1998</td>
<td>A common deep learning network architecture inspired by biological natural visual cognitive mechanisms</td>
<td>code.google.com/p/word2vec</td>
</tr>
<tr>
<td></td>
<td>word2vec [14]</td>
<td>Mikolov et.al.</td>
<td>2013</td>
<td>A well-known unsupervised method to learn high-quality embedded vector representations of words</td>
<td>code.google.com/p/word2vec</td>
</tr>
<tr>
<td></td>
<td>BioVec [61]</td>
<td>Asgari et al.</td>
<td>2015</td>
<td>A new method designed for embedded representation of biological sequences</td>
<td><a href="http://dx.doi.org/10.7910/DVN/JMFHTN">http://dx.doi.org/10.7910/DVN/JMFHTN</a></td>
</tr>
<tr>
<td></td>
<td>dna2vec [62]</td>
<td>Ng et al.</td>
<td>2017</td>
<td>A method to gain DNA k-mer embedded representation</td>
<td><a href="https://pnpn.github.io/dna2vec/">https://pnpn.github.io/dna2vec/</a></td>
</tr>
<tr>
<td></td>
<td>AWD-LSTM [64]</td>
<td>Merity et al.</td>
<td>2018</td>
<td>A weight-decreasing LSTM that uses DropConnect as a form of cyclic regularization for hidden weights</td>
<td><a href="https://github.com/salesforce/awd-lstm-lm">https://github.com/salesforce/awd-lstm-lm</a></td>
</tr>
<tr>
<td></td>
<td>SeqVec [65]</td>
<td>Heinzinger et al.</td>
<td>2019</td>
<td>A new method to represent protein sequences as continuous vectors</td>
<td><a href="https://github.com/mheinzinger/SeqVec">https://github.com/mheinzinger/SeqVec</a></td>
</tr>
<tr>
<td>Transformer</td>
<td>Transformer [17]</td>
<td>Vaswani et al.</td>
<td>2017</td>
<td>Solve the sequence to sequence problem and replace LSTM with a full attention structure</td>
<td><a href="https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py">https://github.com/tensorflow/tensor2tensor/blob/master/tensor2tensor/models/transformer.py</a></td>
</tr>
<tr>
<td></td>
<td>ALBERT [70]</td>
<td>Lan et al.</td>
<td>2019</td>
<td>A lite Bert model with parameter sharing mechanism</td>
<td><a href="https://github.com/google-research/ALBERT">https://github.com/google-research/ALBERT</a>.</td>
</tr>
</tbody>
</table>
field. Over the years, CNN has also become a frequently used neural network model in other deep learning fields.

CNN is composed of three primary neural layers, namely, the convolutional layer, pooling layer and fully connected layer [71]. The simple architecture of CNN is shown in Figure 2(A). In the convolutional layer, multiple convolution kernels perform convolution operations on the input matrix and intermediate feature maps and then transmit the result represented by the feature matrix to the next layer for operation. The pooling layer is used to reduce the feature map dimensions and the number of network parameters, thereby speeding up the network training. The fully connected layer that is always located at the end can convert the two-dimensional feature map into one-dimensional feature vector, which reflects the results of tasks.

Recently, CNN has been applied in obtaining information from biological sequences for corresponding tasks. Through pretraining on other datasets and transferring to target datasets, the CNN-based model not only utilizes data characteristics from different datasets but also achieves remarkable results on target tasks with less training time.

**Word2vec**

Mikolov et al. [14] proposed word2vec in 2013, a well-known unsupervised method to learn high-quality embedded vectors to represent words. By designing two context word prediction tasks, the word2vec model learns the low-dimensional embedding representation of each word, which reflects the context and semantic information of words among sequences.

Word2vec comprises two important models: Skip-gram and Continuous Bag of Words (CBOW). Figure 2(B) shows the architectures of Skip-gram and CBOW. The training object of the Skip-gram model is predicting context words based on the target word, in which the input is the target word, and the output is the context words. Different from Skip-gram, the CBOW model can predict the target word based on context words, which changes the input as the surrounding context and output as the target word. The embedded representation of words is the low-dimensional vectors that are mapped by the hidden layer, which is the by-product of Skip-gram and CBOW. In particular, the Skip-gram model is described as follows:

\[ h = Wx_k \]  

\[ (y_1, y_2, \ldots, y_n) = Wh \]

where \( x_k \) is the input representing the target word; \((y_1, y_2, \ldots, y_n)\) is the output representing the context words; and \( h \) denotes the hidden representation with \( W \) and \( W' \) representing different weights. The CBOW model is introduced as follows:

\[ h = \text{AVG}(W (x_1 + x_2 + \cdots + x_n)) \]  

\[ y_k = W'h \]

where \( x_1, x_2, \ldots, x_n \) is the input representing the context words; \( y_k \) is the output representing the target words; and \( h, W \) and \( W' \) denote the hidden representation and weight. In addition, two training strategies in word2vec are proposed for reducing computational cost and speeding up training time, namely, Hierarchical Softmax and Negative Sampling.

For tasks based on biological sequences, pretraining word2vec-based models can capture syntax and semantic information among biological sequences. After pretraining on large unlabeled biological sequence datasets, word2vec-based models generate high-quality embedding vectors to represent biological sequences, which significantly improve performance after being used in downstream tasks.
Long short-term memory

LSTM [15] is an improved RNN model, which obtains not only information from single input but also contextual information from other input, having advantages in processing long sequences. The LSTM model can extract the semantic and grammatical information in the sequences with mapping sequences into low-dimensional vector space.

Figure 2(C) illustrates the internal structure of LSTM. Compared with RNN, LSTM establishes more delivery states (hidden state and cell state) to transport information, which addresses the gradient explosion and disappearance problem in training long sequences. Structurally, the input of the current unit and states, which are passed from previous units, jointly control the current output and states. The specific process can be described as follows: suppose $x_t$ is the input of LSTM unit $t$, and $h_{t-1}$ is the hidden state passed from previous units, the input information and three function gates can be obtained as follows:

$$z = \tanh(W^c \text{contact} (x_t, h_{t-1}))$$  \hspace{1cm} (5)
$$z' = \sigma(W^c \text{contact} (x_t, h_{t-1}))$$  \hspace{1cm} (6)
$$z'' = \sigma(W^d \text{contact} (x_t, h_{t-1}))$$  \hspace{1cm} (7)
$$z''' = \sigma(W^o \text{contact} (x_t, h_{t-1}))$$  \hspace{1cm} (8)

where $W^c$, $W^d$, $W^o$ represent different weights; $\sigma$ denotes sigmoid function and $z$ denotes input information. Three function gates are identified: input gate ($z''$) controls the information needed to be retained; forget gate ($z'$) controls the information that should be forgotten and output gate ($z'''$) controls the information that will be outputted. Next, suppose $c_{t-1}$ is the cell state passed from previous units, the current output and states are obtained as follows:

$$c_t = z'' \odot c_{t-1} + z' \odot z$$  \hspace{1cm} (9)
$$h_t = z' \odot \tanh(c_t)$$  \hspace{1cm} (10)
$$y_t = \sigma(W^h h_t)$$  \hspace{1cm} (11)

where $c_t$ and $h_t$ denote the cell state and hidden state of LSTM unit $t$, respectively, which will be passed to the next unit. $y_t$ represents the current output, which can be used for tasks. $\odot$ denotes the matrix Hadamard product.

A widely used variant of LSTM is bidirectional long short-term memory (Bi-LSTM), which obtains semantic information in two directions from long sequences. Bi-LSTM performs better in language modeling while extracting comprehensive information from sequences on the basis of LSTM.

The LSTM model gains success in processing long sequences. However, the training strategy and model structure limit the embedding representation of words generated by LSTM, which cannot represent polysemous words in different context. To overcome abovementioned shortcomings, Peters et al. [16] proposed an embedding model based on deep bidirectional language model (Bi-LM), named ELMo, to generate embedding vectors for representing corresponding words according to contextual information. The architecture of the language model in ELMo is presented in Figure 3(A). After embedding, the input sequences are encoded by the Bi-LSTM layers. The output of each LSTM layer is used as the context-dependent word vectors of each word. The final word vectors are generated by linearly combining word vectors of different layers, which represent specific meaning of words in specific context. Different from taking words as input in previous models, the input of ELMo is a sentence. Therefore, ELMo dynamically generates the word vectors on the basis of the context in sentences instead of generating fixed word vectors for words in different sequences.

The LSTM-based model has evident advantages in dealing with long sequences. Thus, it is always used for embedding long biological sequences into low-dimensional vectors as pretraining models.

Transformer

Inspired by remarkable performance of attention mechanism [72] in many fields, the Transformer model is proposed on the basis of attention mechanism in the NLP field. Instead of using traditional CNN and RNN models, Vaswani et al. [17] creatively proposed Transformer, which is a full attention mechanism network. The architecture of the attention layer leads to the parallelism and long-term dependence of the Transformer model.

Figure 2(D) shows the initial structure of Transformer. The Transformer model has a typical encoder–decoder architecture, which is composed of multthead self-attention and feedforward neural network (FNN). In Transformer, sequences are first represented by low-dimensional vectors after word embedding and positional embedding. In encoder, embedding vectors are encoded through N-EncoderLayer, which has a multthead self-attention layer and FNN layer in each layer. In decoder, vectors are decoded through N-DecoderLayer that adds a masked multthead self-attention layer compared with EncoderLayer.

In particular, the multthead self-attention in Transformer can be described using the following equations. Suppose $X$ is the input, the three vectors for calculating attention are as follows:

$$Q = W^Q X$$  \hspace{1cm} (12)
$$K = W^K X$$  \hspace{1cm} (13)
$$V = W^V X$$  \hspace{1cm} (14)

where $Q, K$ and $V$ represent the query vector, key vector and value vector, respectively; $W^Q, W^K$ and $W^V$ are the different weights to calculate different vectors. In addition, suppose $\text{Att}_i$ is the $i$-head attention, output $A$ can be obtained as follows:

$$\text{Att}_i = \text{Attention}(Q_i, K_i, V_i) = \sigma\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right)V_i$$  \hspace{1cm} (15)
$$A = W^\text{contact}(\text{Att}_1, \text{Att}_2, \ldots, \text{Att}_i)$$  \hspace{1cm} (16)

where $d_k$ represents the dimension number in $Q$; $\sigma$ denotes the sigmoid function and $W$ denotes the weight.

In recent years, Devlin et al. [18] proposed a breakthrough multitask pretraining model on the basis of Transformer, Bidirectional Encoder Representations from Transformers (Bert), to learn high-quality vector representation of words. The structure of Bert is presented in Figure 3(B). The Bert model is made up of Bidirectional Transformer (Bi-Transformer) blocks; thus, representation generated by Bert is based on context information of all layers. Compared with previous embedding methods, Bi-Transformer in Bert can capture bidirectional information.
among the sequences more thoroughly. The pretrained task in Bert is Masked Language Model (MLM) and Next Sentence Prediction (NSP), which are used for obtaining embedding representation of words in self-supervised learning. Consequently, multitasks enable Bert to learn sequence information in different views. The Bert model pretrained on large corpus can provide other methods with outstanding embedding representation of words, which improve model performance and reduce training time. In addition, pretrained Bert can be applied in various tasks after fine-tuning according to special tasks, thereby obtaining excellent results.

The full attention mechanism structure enables the Transformer-based models to generate embedding vectors according to the importance of context information, which represent the members in biological sequences. After training on magnanimous unlabeled biological sequences, the pretrained Transformer-based model can provide embedding representation with rich features of biological sequences, which are beneficial to downstream tasks.

### Application of pretraining model

In general, most of the pretraining models for biological sequence data are used for sequence-embedded representation because of the difficulty in obtaining labeled data. Given the sequence \( w_1, w_2, \ldots, w_K \), where each token \( w_k \) represents a word, the embedding process can be described as follows:

\[
[V_1, V_2, \ldots, V_K] = f_{emb}(w_1, w_2, \ldots, w_K)
\]  

(17)

where \( f_{emb} \) represents the pretrained embedding models and \( V_K \) is an embedded low-dimensional vector for representing \( w_K \). Embedded representation can extract features of biological sequences and express such features in another form of vectors. In addition, the feature matrix of biological sequences that is created by embedded representation improves the generalization ability of the models, thereby speeding up the training process and achieving remarkable final output.

Table 3 lists some recent studies that used pretraining models for embedded representation to improve the performance of downstream tasks. In this section, we review some proposed pretraining models for biological sequence data and their application on downstream tasks. Concurrently, we summarize these methods on the basis of the type of pretraining models: CNN, word2Vec, LSTM and Transformer.

### Methods based on CNN

Previously, CNN was often used in CV to deal with image information. Recently, some studies [22, 73] have used CNN to pretrain biological sequence in transfer learning. Zhuang et al. [73] proposed a simple CNN model on the basis of DNA sequences to predict EPI. The simple model contains two input channels for enhancer and promoter sequences, followed by a convolution layer to encode sequences and a max-pool layer in each sequence. After matrix operations, a fully connected layer connects two results and finally outputs the EPI probability after dropout and sigmoid activation. The model is first pretrained with DNA sequences from other cell lines, where feature information was extracted from other sequences, and then trained with DNA sequences from target cell lines for CPI prediction. Playe et al. [22] designed a chemogenomic neural (CN) network, a deep neural network model that takes the embedded representation of the protein sequences and molecular graphs as input to predict DTI. The embedded representation of protein sequences is obtained by CNN encoder pretrained on DBEColi dataset from DrugBank database, which performs better than Bi-LSTM in test. As shown in the results, the CN model performs well in DTI prediction on large datasets.

### Methods based on Word2vec

In NLP field, some embedded methods [14, 60] are developed on the basis of word2vec, which can be applied to embedded biological sequences. Concurrently, many methods [61, 62] particularly designed for biological sequences are proposed in recent years. Some studies [25, 54, 58, 75, 81] have employed word2vec-based methods for embedded representation of biological sequences, which are pretrained over large corpus and performed well on downstream tasks.

Benefit by great performance of word2vec in embedded representation, some methods [54, 58] select word2vec as a pretrained model for biological sequences. Chen et al. [54] proposed a novel method, namely, TransformerCPI for CPI prediction, which is fed with protein sequences and compound sequences. The model converts the protein sequences into a real-value 100-dimensional vector by word2vec pretrained on UniProt database, to extract features and represent protein sequences. Chaabane et al. [58] used the Skip-gram model (word2vec) to generate an embedded matrix of RNA sequences for circular RNA classification. During embedding, word2vec is first pretrained with RNA sequences and fine-tuned during subsequent training.

Although word2vec performed well in representing words, it still cannot avoid the order and semantics of words in...
<table>
<thead>
<tr>
<th>Author</th>
<th>Year</th>
<th>Dataset</th>
<th>Method</th>
<th>Pretraining model</th>
<th>Application</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Based on CNN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Playe et al. [22]</td>
<td>2020</td>
<td>DrugBank, DBHuman, DBEcoli</td>
<td>CN</td>
<td></td>
<td>DTI</td>
<td><a href="https://github.com/bplaye/NNk_DT">https://github.com/bplaye/NNk_DT</a></td>
</tr>
<tr>
<td><strong>Based on word2vec</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chen et al. [54]</td>
<td>2020</td>
<td>UniProt, DrugBank, BindingDB, GPCR, Kinase</td>
<td>TransformerCPI</td>
<td>word2vec</td>
<td>CPI</td>
<td><a href="https://github.com/lifanchen-simm/transformerCPI">https://github.com/lifanchen-simm/transformerCPI</a></td>
</tr>
<tr>
<td>Yang et al. [74]</td>
<td>2018</td>
<td>UniProt</td>
<td>doc2vec</td>
<td></td>
<td>Four tasks</td>
<td><a href="https://github.com/fhalab/embeddings_reproduction/">https://github.com/fhalab/embeddings_reproduction/</a></td>
</tr>
<tr>
<td>Hong et al. [25]</td>
<td>2020</td>
<td>TargetFinder (EPI)</td>
<td>EPIVAN</td>
<td>dna2vec</td>
<td>EPI</td>
<td><a href="https://github.com/hzy95/EPIVAN">https://github.com/hzy95/EPIVAN</a></td>
</tr>
<tr>
<td><strong>Based on LSTM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Based on Transformer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rives et al. [77]</td>
<td>2019</td>
<td>UniProt, UniParc, CBS135926_filtered</td>
<td>Bert</td>
<td>Bert</td>
<td>Variant activity prediction</td>
<td><a href="https://github.com/agemagician/ProtTrans">https://github.com/agemagician/ProtTrans</a></td>
</tr>
<tr>
<td>Vig et al. [78]</td>
<td>2020</td>
<td>ProteinNet, TAPI</td>
<td>Bert</td>
<td>Bert</td>
<td>Interpretability</td>
<td><a href="https://github.com/salesforce/provis">https://github.com/salesforce/provis</a></td>
</tr>
<tr>
<td>Nambiar et al. [79]</td>
<td>2020</td>
<td>SWISS-PROT, HIPPIE</td>
<td>ProBERTa</td>
<td>RoBERTa</td>
<td>PFC PPI</td>
<td><a href="https://github.com/nambiar/ProBERTa">https://github.com/nambiar/ProBERTa</a></td>
</tr>
<tr>
<td>Elnaggar et al. [80]</td>
<td>2020</td>
<td>UniRef100, BFD</td>
<td>ProtTrans</td>
<td></td>
<td>HPC in protein LMs</td>
<td><a href="https://github.com/agemagician/ProtTrans">https://github.com/agemagician/ProtTrans</a></td>
</tr>
</tbody>
</table>
sentences. Mikolov et al. [60] proposed doc2vec, an improved unsupervised embedding method for sequences with variable length to address the challenges of word2vec. Yang et al. [74] used pretrained doc2vec to embed protein sequences into 64-dimensional space, which is first trained on unlabeled protein sequences from UniProt. The experimental results show that the embedded representation of protein sequences performs well for predicting protein property.

Word2vec makes great contributions to the embedded representation of biological sequences as a pretrained model. Consequently, some word2vec-based methods [61, 62] designed particularly for embedding biological sequences have been proposed in recent years. Asgari et al. [61] proposed Bio2vec, which is an embedded method particularly for biological sequences. Bio2vec can generate continuously distributed representation of biological sequences using a pretrained Skip-gram model, which is divided into Protvec (for protein) and Genvec (for gene) according to different training objects. Deznabi et al. [75] applied Protvec that trained protein sequences from SWISS-PROT to embed phosphosite into 1300-dimensional vector, which provides embedded representations of phosphosite for kinase–phosphosite association prediction. Ng et al. [62] obtained DNA k-mer-embedded representation by word2vec trained with human DNA sequences, which is called dna2vec. Hong et al. [25] proposed a novel method called EPIVAN to predict EPI with only genomic sequences. For representing enhancer and promoter, EPIVAN used the DNA vectors generated by pretrained dna2vec to encode DNA sequences.

Methods based on LSTM

Word2vec is an effective model in generating vectors representing words, but it only provides limited help for embedding long sequences. Another effective neural network model for processing sequence information is LSTM. Given its advantages, many LSTM-based embedding methods [15, 16, 63–66] are proposed and used for representing biological sequences as pretrained models in some methods [26, 53, 55, 76].

Bepler et al. [53] proposed SSA frame, which predicts protein structural similarity from amino acid sequences. Their proposed method maps protein sequences to embed vectors by Bi-LSTM models pretrained on protein sequences in Pfam. Sutskover et al. [63] proposed seq2seq, an encoder–decoder LSTM model that used attention mechanism to output sequences with uncertain length. In seq2seq, sequences are first embedded into vectors with fixed length by an LSTM model and then converted into ideal sequences by another LSTM model. Karimi et al. [76] designed a semisupervised deep learning model, which predicts compound–protein affinity with unlabeled and labeled data, namely, DeepAffinity. For leveraging rich information from compound and protein, they used a seq2seq model pretrained on unlabeled sequences to embed labeled sequences.

Merry et al. [64] regularized and optimized an LSTM model by using DropConnect on hidden-to-hidden weights and presented ASGD Weight-Dropped LSTM (AWD-LSTM), which performs better in embedding sequences than LSTM. Strothoff et al. [26] proposed UDSMProt pretrained on unlabeled protein sequences, which classified proteins from sequences. UDSMProt used AWD-LSTM as a pretrained language model to understand the good embedding of protein sequences, which also performs well when transferring to other three tasks. Alley et al. [66] build unified representation (UniRep) from massive unlabeled amino acid sequences by a multi-LSTM model. As shown in their results, the statistical representation of protein sequences contains rich semantical information, which can be broadly applied to other methods as pretrained embedded representation.

Due to the outstanding performance of EMLO [16] in sequences processing, the pretraining ELMO model also performed well in embedding biological sequences. Heinzinger et al. [65] proposed Seq2Vec, a novel embedding model based on ELMO pretrained on UniRef50, to represent protein sequences by continuous vectors. This method can also be used as pretrained model for embedding biological sequences in other methods. Amelia et al. [55] used embedded representation of protein sequences with additional protein contact map to predict protein function, in which high-quality embedded vectors are generated by LSTM-based SeqVec model pretrained on PDB database.

Methods based on transformer

Although the LSTM-based methods have achieved good results in embedding biological sequence as pretrained models, they are still limited in training long sequences. In recent years, many models [17, 18, 67–70] based on Transformer have been proved to perform well in embedded representation of biological sequence as pretrained models, particularly after Bert proposed.

Rives et al. [77] trained Bert on 86 billion amino acids from 250 million sequences. In their experiment, raw protein sequences are mapped into representation space reflecting biological structure at many levels. The representations offer various information and feature of proteins, which can be extracted and used by other methods according to downstream tasks. Vig et al. [78] focused on interpretability of embedded representation learned by Transformer architectures (Bert). Their results show that attention mechanism can capture the folding structure of proteins and target binding sites and focus on biophysical properties.

Recent advances in Transformer-based methods made it valid to embed biological sequence as high-quality vector representation. Dai et al. [67] improved the Transformer model in accepting variable length context in language modeling, proposing Transformer-XL. With novel segment-level recurrence mechanism and positional encoding scheme, Transformer-XL performed well in capturing long-term dependency and processing context fragmentation. Yang et al. [68] overcame the limitations of MLM in Bert by replacing the autoregressive model with autoencoding model simultaneously and designing a novel generalized autoregressive pretraining method, namely, XLNet. Given the two-stream self-attention mechanism and integrating advantages of Transformer-XL, XLNet outperformed Bert in various NLP tasks. Liu et al. [69] proposed an improved Bert model-RoBERTa, which adjusted training details of Bert and achieved dynamic masking mechanism. Lan et al. [70] proposed ALBERT, a lite Bert with cross-layer parameter sharing and factorized embedding parameterization, thereby speeding up the training phase. For embedding protein sequences, Nambiar et al. [79] designed ProBERTa, a neural network architecture based on RoBERTa. After pretraining on SWISS-PROT database and fine-tuning, their method performs well in protein family classification and protein interaction prediction. Elnaggar et al. [80] combined Transformer-based models with high-performance computing to map protein sequences as embedding vectors, namely, ProtTrans. In their experiment, researchers trained four models (Transformer-XL, XLNet, BERT and Albert) on 93 billion amino acids from 2.1 billion protein sequences. As shown in the results, Transformer-based models pretrained on a large amount of labeled data extracted the
biophysical information of proteins and achieved good results in various downstream tasks.

Scheme and benchmark

In this section, we introduce a novel pretraining scheme for protein sequences and a multitask benchmark for protein embedding methods. Hopefully, pretraining scheme and protein embedding benchmark can provide novices with a way, in which researchers can quickly design methods for protein sequences and evaluate the performance of protein embedding models.

Min et al. [82] proposed a novel pretraining scheme for protein sequences, namely, PLUS, in which embedding models are pretrained with protein-specific pretraining task to obtain information in unlabeled protein sequences. It reflects the difference between protein sequences and natural language sequences. PLUS contains two pretraining tasks (MLM and Same Family Prediction), which obtain sequence information and protein-specific information. In particular, protein sequences are first masked 15% at random and then transformed to embedding vectors by representation models with two pretraining tasks. The transferability of PLUS enables various embedding models for biological sequences to be pretrained and fine-tune on downstream tasks, such as Bi-LSTM and Transformer. With the help of PLUS, researchers can focus on designing their pretraining embedding methods regardless of auxiliary tasks and training procedures.

Rao et al. [83] collected and designed a multitask standard benchmark, Tasks Assessing Protein Embeddings (TAPE), to make up the gaps in standardized evaluation indicators and datasets for protein semisupervised learning. TAPE reflects multiple functions of the protein sequences and evaluates the pretraining models for protein sequence from multiple aspects. TAPE consists of five semisupervised learning tasks relevant to proteins (secondary structure prediction, contact prediction, remote homology detection, fluorescence landscape prediction and stability landscape prediction), which cover three areas of protein biology: structure prediction, evolutionary understanding and protein engineering. For datasets, TAPE provides an unlabeled protein sequence dataset constructed from Pfam database and many supervised preprocessed datasets for downstream tasks. In addition, the experimental results indicate that self-supervised pretraining models for biological sequences can significantly improve the performance on downstream tasks. These tasks and protein datasets in TAPE can be used for evaluating the performance of protein pretraining methods in multispects, which offers a fair and open benchmark for measuring the effectiveness of pretraining models.

Challenges and future directions

Pretraining models that are not related to specific tasks are obtained by self-supervised learning on large-scale data. With the emergence of pretraining models and their successful applications in fields such as NLP [84], CV depicts the power of pretraining technology. Pretraining models are applicable to almost all tasks that rely on large amounts of data, particularly unlabeled data.

Regarding biological sequence data, pretraining models can generate embedded representation that reflects the semantic information of biological sequence after training on large corpus, which speeds up training process, improves performance on downstream tasks and supports new tasks with fine-tune. However, despite the success of pretraining models for biological sequence in recent years, such models still face challenges and need further development in this field. Herein, we summarize challenges and potential future directions in pretraining models for biological sequence.

Data

Pretraining models for biological sequence require large amount of data to learn sufficient features in sequences, but reliable biological sequence data are not enough. Although high-throughput sequencing technology has brought various new sequence data [85, 86], it still cannot meet the developing pretraining models, particularly DNA and RNA sequences. In addition, the expensive cost of obtaining labeled data and lack of negative samples hinder the transfer of pretraining models for biological sequence in many tasks. On the other hand, the multimodal pretraining model is a good solution. Compared with previous methods, multimodal pretraining models [87] fuse abstract feature from different types of data such as sequence, image and graph, which learn good feature representation from multimodal data while making up for the lack of sequence data. Therefore, multimodal pretraining models can make full use of more data and perform better on downstream tasks. We hypothesize that more multimodal pretraining methods are proposed for biological sequence in the future. Meta-learning is a novel learning strategy, which helps the model learn information quickly with a small number of samples. The main idea in meta-learning is making models to learn based on previous experience and knowledge. Combining meta-learning with pretraining models for biological sequences would be a potential future direction.

Pretraining tasks

Based on specific pretraining tasks, pretraining models can learn abundant feature representation on large datasets. A variety of NLP pretraining models [18, 69, 88] used LM or MLM as a pretraining task. Some NLP pretraining models [68, 70, 89–91] are also transferred to biological sequences. However, tasks in NLP reflect partial characteristics of biological sequences [82]. Simultaneously, a single task has limited effects on pretraining models. For the former, more tasks reflecting specific information in biological sequences are proposed. In addition, a new future direction is contrastive learning [92], learning the semantic information from similarity and difference of sequence pairs. Recently, many contrastive learning pretraining models [93–96] are proposed, which also have a good prospect in biological sequence data. For the latter, one improvement that can be made is using multitask pretraining models [97] instead of single pretraining models. The models in multitask learning are trained through a set of related tasks, which improve the generalization ability of the models. By taking the relation and difference between different tasks into consideration, multitask pretraining models perform better than single-task models. Multitask pretraining models become increasingly popular, when single-task models have been unable to meet pretraining requirements gradually.

Pretraining models

Existing pretraining models for biological sequences are derived from NLP domain. After the Transformer architecture came out, Transformer-based pretraining models in NLP reached a new
height and faced some new challenges. The most prominent problem is too many parameters in Transformer models, which requires expensive computing resources and long time to fit the pretraining models. An interesting future direction is to propose new architectures to overcome the disadvantages of Transformer models. In recent years, some valid training strategies are designed for compressing pretraining models, such as model trimming [98], parameter sharing [70], etc. At present, Knowledge Distillation (KD) [99] is a novel research direction in reducing pretraining models. Two models are identified in KD: student (small) model and teacher (large) model, in which the student model is obtained through transferring knowledge from a trained teacher model. KD has an ability to transfer a large model to a small model that retains the performance close to the large model. In addition to model compression, new pretraining model architecture and interpretability of pretraining models are popular future directions. Knowledge graphs also have been applied in the prediction of drug repurposing [100], disease genes [101, 102], circular RNAs [103] miRNAs [104, 105]. It would be interesting to study using knowledge graphs for the pretraineding of biological sequences.

**Conclusion**

In this paper, we provided a review that aimed to introduce recent development and studies on pretraining models for biological sequence data. In general, we included in this review the background of pretraining models for biological sequence, a brief introduction to biological sequence data and corresponding datasets, popular pretraining models in previous works, application of pretraining models for biological sequences, a novel scheme and benchmark on pretraining models for protein sequences, and challenges and future directions.

In particular, we first illustrated the deep learning background of pretraining models for biological sequences, containing the role of biological sequence data and introduction of pretraining models. Then, we made a brief introduction of biological sequences and several notable biological sequence databases. We also collected and presented some datasets with brief description and available link. Next, we proposed a classification scheme for pretraining models and reviewed the literature on the basis of the categories of pretraining models. Moreover, we separately introduced the corresponding structure, features and mechanisms of pretraining models. We further detailed some methods for downstream tasks with proposed pretraining models to explain the application of pretraining models, such as DTI, EPI, PPI [106, 107], protein function prediction [108–111] and RNA classification [112]. In addition, we provided a novel pretraining scheme and benchmark for protein sequences, which helped researchers to design and verify their methods. Finally, we discussed existing challenges and popular future research directions of pretraining models for biological sequence to guide future works. We hope that this survey can provide readers with a general understanding toward this field, some resources for conducting research and feasible ideas for future research in pretraining models for biological sequence data.

**Key Points**

- Summarize popular pretraining models for biological sequences based on four categories: CNN, word2vec, LSTM and transformer.

- Present some applications of pretraining models for biological sequences on downstream tasks to explain the role of pretraining models.

- Discuss the challenges and future research directions in pretraining models for biological sequences.

**Funding**

The work was supported in part by National Natural Science Foundation of China (61872309, 61972138, 62002111), in part by the Fundamental Research Funds for the Central Universities (531118010355), in part by China Postdoctoral Science Foundation (2019 M662770), in part by Hunan Provincial Natural Science Foundation of China (2020J4215), in part by Key Research and Development Program of Changsha (kq2004016) and in part by Changsha Municipal Natural Science Foundation (kq2014058).

**Conflict of interest**

The authors confirm that this article content has no conflict of interest.

**References**


Pretraining model for biological sequence data