SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2025, Vol. 68, Iss. 7, 170106:1–170106:13
https://doi.org/10.1007/s11432-024-4481-9

Special Topic: AI for Biology

OntoGene: knowledge-enhanced BERT for promoter identification

Yang LI[1], Mengli GAO[1], Jilong BIAN[1], Kaiqi ZHAO[2], Dan LI[1*] & Guohua WANG[1*]

[1]*College of Computer and Control Engineering, Northeast Forestry University, Harbin 150040, China*
[2]*School of Computer Science, University of Auckland, Auckland 1010, New Zealand*

Received 23 December 2024/Revised 11 February 2025/Accepted 16 March 2025/Published online 23 June 2025

Abstract Promoters are crucial parts of DNA sequences and play a significant role in understanding the process of gene regulation. Despite the existence of numerous methods for predicting promoters, they are typically built on single-sequence information and seldom incorporate external knowledge graphs (KGs). Knowledge graphs, however, can provide rich factual knowledge that enhances the accuracy of promoter identification. In this study, we propose a novel method called OntoGene, which integrates Gene Ontology (GO) into the bidirectional encoder representations from transformers model (BERT) for promoter identification. Specifically, OntoGene encodes both gene sequences and external knowledge into vectors using a BERT encoder. Our method jointly optimizes knowledge embedding and masked language modeling objectives, and utilizes a contrastive learning method with knowledge-aware negative sampling to optimize gene and knowledge graph embeddings. OntoGene is a BERT-based model designed to predict not only the presence of promoters but also their strength (strong or weak promoters) from gene sequences. Experimental results demonstrate OntoGene's high accuracy in predicting DNA promoters and their strengths, validating the effectiveness of this knowledge-enhanced approach. OntoGene and the dataset can be obtained from https://github.com/gaomengli7/OntoGene.

Keywords promoter identification, knowledge enhancement, knowledge graph, BERT

Citation Li Y, Gao M L, Bian J L, et al. OntoGene: knowledge-enhanced BERT for promoter identification. Sci China Inf Sci, 2025, 68(7): 170106, https://doi.org/10.1007/s11432-024-4481-9

1 Introduction

A gene's promoter, located upstream of the transcription start point in the DNA sequence [1], initiates the attachment of RNA polymerase to template DNA, thereby starting the transcription of specific genes. Promoters determine the direction, speed, and accuracy of DNA transcription, thereby controlling the expression of genes in an organism and when and where they are expressed [2]. Scientists have long been intrigued by understanding crucial details about protein-coding genes [3]. Therefore, the primary step in gene identification within DNA sequences involves the identification of promoters.

In promoting gene expression, the function of promoters is crucial. Therefore, accurately identifying promoters remains a challenging yet pivotal area in current biological and computational research [4]. In recent years, several new methods have emerged for promoter identification. For instance, Solovyev et al. [5] extracted promoter sequence features from a variety of prokaryotic and eukaryotic organisms using convolutional neural networks (CNNs), achieving good generalization in newly sequenced genomes. DeepPromoter [6] identifies short eukaryotic promoters from humans and mice. Xiao et al. [7] introduced a benchmark dataset and proposed predictive factors that can not only identify DNA promoters but also predict their promoter strength classes. Shujaat et al. [8] developed pcPromoter-CNN, a method that utilizes a one-hot encoding scheme with CNN for promoter prediction and classifies them into six subclasses. Ma et al. [9] presented a deep learning algorithm based on bidirectional long short-term memory (LSTM) and CNNs that efficiently identifies promoter points in eukaryotic genomes. Le et al. [10] proposed BERT-Promoter, which encodes DNA sequences with pre-trained BERT models and selects features through SHAP analysis, ultimately predicting DNA promoters and their activities with various machine learning classifiers. Zaytsev et al. [11] developed a novel promoter classification method combining genetic algorithms and MAHDS sequence comparison to identify potential promoter sequences in the

* Corresponding author (email: ld725725@126.com, ghwang@nefu.edu.cn)

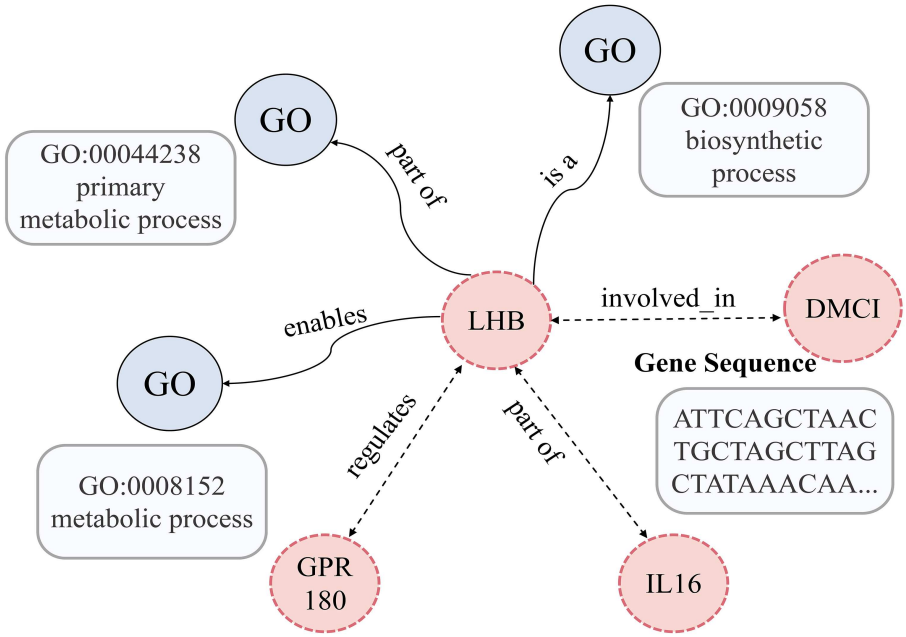

Figure 1 (Color online) Example of KG with entity description. The orange nodes represent gene sequences, while the blue nodes represent GO entities with biological descriptions.

human genome, significantly reducing false positive predictions compared to other methods. However, existing promoter prediction methods exhibit a common limitation: they rely exclusively on sequence information and overlook biological knowledge. Incorporating biological knowledge could improve promoter prediction accuracy and support various biological applications [12], such as more precise annotation of gene function regulated by promoters. Nevertheless, such knowledge is typically fragmented and limited, while DNA sequences are diverse and complex. For example, many biological knowledge bases in research provide a large amount of biological information, but this information is often incomplete and inconsistent. Moreover, the challenge of utilizing biological knowledge lies not only in the complexity of the knowledge itself but also in how to integrate these discrete and heterogeneous pieces of information into machine learning models. In genomic sequence analysis, biological knowledge often needs to be injected into the model through appropriate representation methods and model structures to help the model understand the complex biological context. However, because biological knowledge is typically hierarchical and multidimensional, selecting the appropriate knowledge sources and effectively integrating them is an important research direction. Therefore, the challenge lies not only in overcoming the lack of data but also in how to leverage existing biological knowledge to compensate for the limitations of the data itself.

In contrast, Gene Ontology (GO) [13,14] encompasses various biological facts such as gene properties, functions, and interactions. It is organized into three main aspects: molecular function, cellular component, and biological process. The GO database represents facts with entity-relation-entity triplets, for instance, (GeneA, regulates, GeneB). Such triplets form a knowledge graph, where nodes in the graph represent entities and edges denote relations between them. By employing knowledge embedding techniques, GO's biological knowledge can be embedded into continuous entity and relationship embeddings [15,16], as illustrated in Figure 1. These embeddings contribute to the completion of the knowledge graph and enhance the accuracy and reliability of promoter prediction. However, unlike knowledge enhancement methods in natural language processing (NLP), gene sequences and gene ontologies are two distinct types of data. While gene sequences are composed of DNA sequences, gene ontologies consist of text-based knowledge graphs. Embedding gene ontologies into BERT models is challenging due to the disparity in information fusion between these two data types [17].

Inspired by Zhang et al. [18], we leverage entity descriptions to bridge the gap between knowledge embedding (KE) and BERT models [19], aligning the semantic space of genes with textual space to embed GO terms. To this end, this paper introduces a novel method OntoGene, which embeds Gene Ontology into the BERT model to facilitate the identification and classification of promoters.

Specifically, we propose a hybrid encoder to obtain the representations of both linguistic text and

gene sequences. This encoder maps Gene Ontology annotations and gene sequences into corresponding entity embeddings. Furthermore, we introduce contrastive learning with knowledge-aware negative sampling, aiming to jointly optimize the knowledge graph and gene sequence embeddings during fine-tuning. Consequently, our model incorporates richer biological characteristics of promoter sequences in its representation. The experimental results show that the OntoGene model achieves accuracies of 96.7% for identifying promoters and 98.5% for predicting their strength, surpassing the baseline models. To pretrain and evaluate OntoGene, we constructed a large-scale knowledge graph, GeneKG, which contains biological knowledge facts aligned with gene sequences. It includes Gene Ontology, gene sequences, and gene annotation data. The following are some benefits of OntoGene.

(1) OntoGene integrates biological knowledge and sequence information by employing knowledge embeddings to obtain representations of gene sequences. This approach effectively predicts promoters and their strengths.

(2) Using a masked language model, OntoGene has developed a thorough understanding of genes.

(3) We have developed a novel knowledge graph named GeneKG, which incorporates Gene Ontology, gene sequences, and gene annotation data.

2 Materials and methods

To improve the accuracy of promoter identifications, OntoGene incorporates Gene Ontology into a BERT model. To achieve the goal, OntoGene integrates external knowledge from Gene Ontology into language representations, optimizing knowledge embedding and masked language modeling (MLM) objectives simultaneously. Figure 2 depicts our approach's overall workflow. We will begin by introducing knowledge graphs, followed by a discussion about representation learning for textual and gene data. Next, we will delve into knowledge-aware negative sampling contrastive learning, and finally, we will describe the promoter prediction.

2.1 Gene and KG

To integrate Gene Ontology knowledge into the language model, we constructed a knowledge graph dataset, GeneKG, which aligns gene sequences and the descriptions of GO terms with gene entities. The construction process of the GeneKG knowledge graph dataset is as follows.

(1) Entity definition. GeneKG includes two core types of entities: gene entities and GO term entities. Gene entities represent specific protein-coding genes, while GO term entities represent gene functions, molecular activities, or biological processes in which the genes are involved. All entities are represented in a standardized manner to ensure their uniqueness and consistency.

(2) Relationship design. The design of relationships in the knowledge graph strictly follows the intrinsic connections between gene annotations and Gene Ontology. For example, the relationship between genes is defined as "belongs to", while the relationship between genes and Gene Ontology is defined as "regulates". Additionally, the relationship between genes and Gene Ontology can also be described as "interacts with".

(3) Triplet construction. Based on the defined entities and relationships, a knowledge graph containing 399766 triplets was constructed. Each triplet is represented as (head entity, relationship, tail entity), for example, (gene, regulates, gene) and (gene, part of, GO). All triplets are verified through strict logical rules and semantic checks to ensure their accuracy and consistency.

(4) Node description alignment. For each gene entity, a corresponding gene annotation text or gene sequence description is assigned, and this information is aligned with GO terms and gene entities. Gene Ontology describes the relationships between GO terms through GO-GO triplets, while gene annotations describe the relationships between genes and GO terms through Gene-GO triplets. The schematic of GeneKG is shown in Figure 3. In GeneKG, gene sequences and annotation content provide detailed semantic descriptions for each node, making it more suitable for downstream task learning needs.

Based on the above steps, this study constructs a new knowledge graph dataset, GeneKG, and describes all the nodes in the graph using gene annotation texts or gene sequence descriptions. For each GO term in Gene Ontology, it is aligned with its corresponding name and description, with a colon connecting them to form a complete description. For each gene in the gene annotations, it is aligned with the corresponding gene in National Center for Biotechnology Information (NCBI), and its corresponding sequence is extracted as a description. Given the close relationship between Gene Ontology and gene

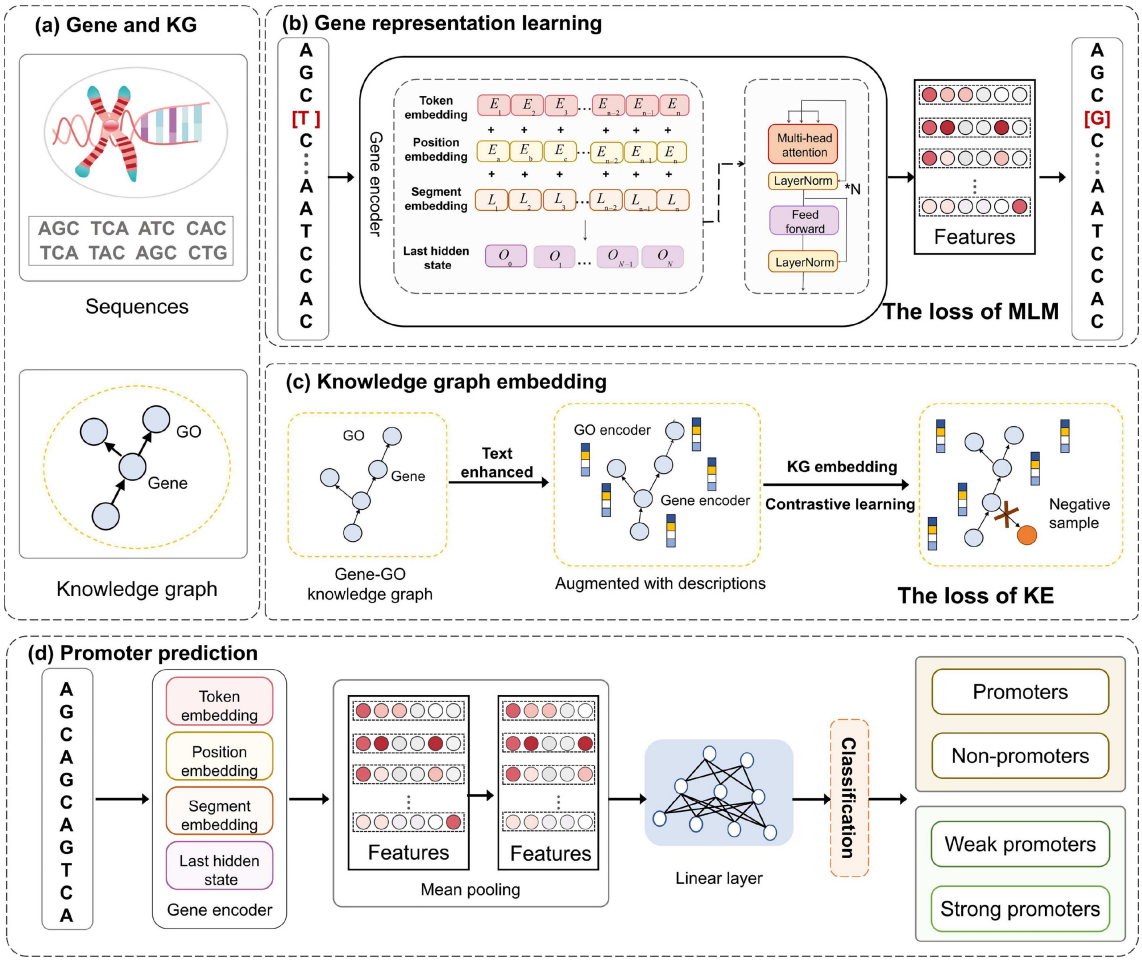

Figure 2 (Color online) Overall workflow of OntoGene. (a) Gene sequences and triplets are introduced to describe the relationships between genes and other entities. The gene sequences are represented by k-mer, and the triplets consist of head entities, relations, and tail entities, which represent the relationships between entities in the knowledge graph. (b) Gene representation learning is introduced to describe a BERT-based encoder that performs representation learning on gene sequences and triplets, learning gene feature vectors through contextual information. (c) Knowledge graph embedding is introduced to describe knowledge embedding techniques that transform gene and text information into continuous vectors for entities and relations, combining gene sequences with the knowledge graph to capture the deep connections between genes and Gene Ontology. (d) Promoter prediction is introduced to describe the use of these embedded representations for promoter prediction tasks, enabling precise classification and prediction of gene functions.

annotations, we combine these two structures into a unified knowledge graph. In GeneKG, the knowledge graph contains 20006 entities, 399766 triplets, and aligned node descriptions derived from GO annotations.

2.2 Gene and knowledge representation learning

BERT is a pre-trained language representation model proposed by Google in 2018 [20]. We first introduce the BERT encoder to encode gene sequences and biological descriptions of Gene Ontology terms. To achieve this, we design a hybrid encoder that combines two different types of encoders to simultaneously represent gene sequences and Gene Ontology knowledge. The hybrid encoder consists of two main components: the gene encoder and the GO encoder. Specifically, the gene encoder is responsible for processing gene sequence information, while the GO encoder handles biological descriptions from the GO terms. These two encoders are mapped to the same representation space using affine transformations, ensuring that they effectively integrate gene sequence information and GO knowledge, thus providing richer and more precise representations for downstream tasks.

For the gene encoder, we employed the DNABERT model pretrained by Ji et al. [21], which adopts the BERT architecture and is pretrained on the human genome GRCh38. Compared to the BERT model, DNABERT analyzes DNA sequences by considering the entire genomic context, rather than isolated segments. This allows DNABERT to capture long-range dependencies and contextual relationships across

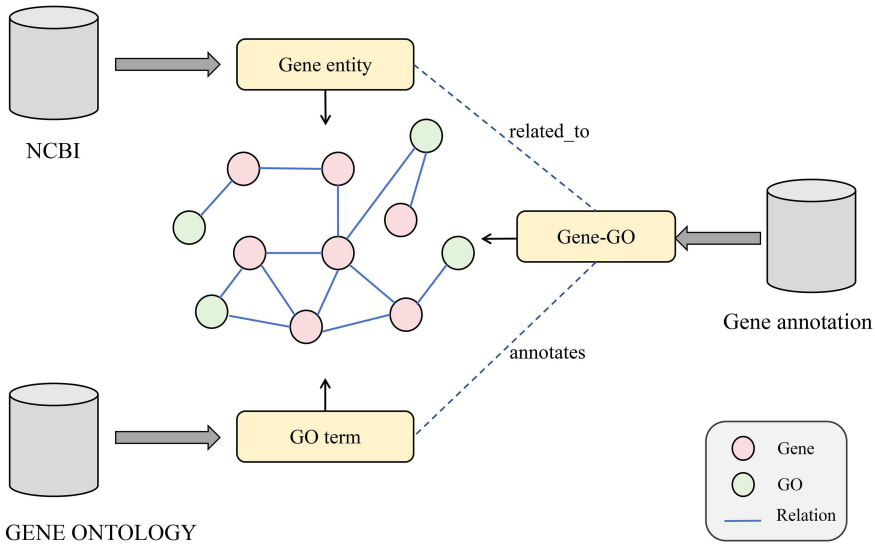

Figure 3 (Color online) Schematic of GeneKG.

the whole DNA sequence, which improves its ability to understand complex genomic structures. Additionally, DNABERT is designed to generalize well to unseen DNA sequences, meaning it can effectively apply the knowledge learned from one set of sequences to predict and analyze new, previously unseen genomic data. It ranks the importance of nucleotide molecules and analyzes the relationship between input sequence contexts to obtain better visualization information and precise motif extraction. For encoding gene sequences, we treated every k nucleotides as a "word", referred to as k-mer, and encoded DNA sequences using four different k values: $k = 3, 4, 5, 6$. In this study, We initially experimented with different values for k in the k-mer encoding approach, testing a range of values to determine which would yield the best performance for DNA sequence analysis. After evaluating the results, we found that $k = 3$ provided the most accurate and reliable results, and hence, we opted for $k = 3$ in our final model. The gene encoder takes a k-mer sequence as input and generates its contextual representation. Specifically, the gene encoder obtains the embedding of each input k-mer by adding three components: (1) its one-hot encoding, (2) its positional encoding, and (3) its segment embedding. The segment embedding is used to distinguish different input statements. With the input embeddings, the gene encoder employs stacked transformer encoders to capture the contextual semantics from the gene sequence.

For the GO encoder, we use the pre-trained PubMedBERT model [22], which is also based on the Transformer architecture and optimized for biomedical text. The GO encoder takes a sequence of N tokens $(X'_1, X'_2, \ldots, X'_n)$ as input and forms an embedding layer by summing the embeddings of tokens, positional embeddings, and segment embeddings. The embedded sequence is then passed through multiple Transformer layers. In each Transformer layer, a nonlinear transformation is applied to the representations of all tokens from the previous layer, generating context-aware representations for each token. Attention scores are computed by weighting the given token representations in the previous layer as queries. The final layer outputs context representations for all tokens, combining information from the entire text range. By projecting genes and text into the same space, we bridge the gap between them.

By mapping the outputs of the gene encoder and GO encoder to the same space, the hybrid encoder effectively bridges the gap between gene sequences and GO knowledge. The hybrid encoder generates unified representations by simultaneously processing gene sequences and GO terms, capturing both the structural information of gene sequences and the functional information from GO terms. Through parallel optimization under the MLM and KE objectives, we better integrate the deep relationships between gene sequences and GO knowledge into the model, thereby providing richer and more accurate information for subsequent tasks.

Our method does not incur additional overhead since, in contrast to earlier work on knowledge augmentation, adding more entity connectors what are entity connectors or knowledge integration what are knowledge integration layers.

2.3 Knowledge embedding

During the fine-tuning phase, we incorporate factual information from Gene Ontology into the model using KE techniques [23]. Knowledge embedding involves mapping entities and relations from a knowledge graph into a low-dimensional continuous vector space through distributed representation. This approach enables the integration and computation of various types of data. A knowledge graph is a graph structure where entities are represented as nodes and relationships between entities as edges [24]. As shown in Figure 2(c), we formally define the knowledge graph as a structure comprising an entity set E and a relation set R. A fact in the knowledge graph is represented by a triplet in the form of (h, r, t), representing the head entity $h \in E$, relation r, and tail entity t. The goal of the KE model is to embed every entity $h \in E$ and relationship $r \in R$ into a D-dimensional continuous space, where E and R stand for the collection of entities and relationships, respectively. To this end, each entity and relationship is given a D-dimensional vector. In addition, a score function for these embedding vectors' training is defined.

In our gene knowledge graph, there are two types of nodes, namely Gene node (e_{Gene}) and GO node (e_{GO}). A Gene node is associated with a gene sequence representation, while a GO node indicates a node in the Gene Ontology which can be described by annotated text. The triples in the knowledge graph can be categorized into two categories. If the head entity is a Gene node and the tail entity is a GO node, it can be defined as $T_{\text{Gene-GO}}$; if both the head entity and the tail entity are GO nodes, define them as $T_{\text{GO-GO}}$.

To address the issue of heterogeneous information fusion, we use the encoders introduced in Subsection 2.2 to encode gene sequences and GO annotations separately into vectors, and optimize them through the knowledge embedding objective. This process not only strengthens the relationship between gene sequences and GO concepts but also provides rich external knowledge for further learning.

2.4 Masked gene modeling

In this section, we utilize masked gene modeling to optimize gene representations, akin to the MLM objectives in BERT and RoBERTa [25]. The process is illustrated in Figure 2(b). At the fine-tuning stage of MLM, a subset of input positions are randomly masked, and the objective is to predict these masked tokens from a fixed dictionary. Specifically, MLM randomly masks or replaces some k-mers within the gene sequence, which are then treated as masked tokens for prediction by the gene sequence encoder. To ensure an adequate number of samples, approximately 15% of the consecutive k-mer regions in each sequence are randomly masked. The masking process involves the following details. (1) 80% of the time. The position is replaced with [MASK], indicating that the true information at that position is hidden, and the model needs to predict it using contextual information. (2) 10% of the time. The sequence remains unchanged, meaning no modification is applied to this portion, and the model must rely on context to infer its meaning. The remaining positions are replaced with random k-mer, which introduces randomness into the training process. Finally, the cross-entropy loss $\mathcal{L}_{\text{MLM}}$ is calculated at these selected positions, which measures the difference between the predicted probability and the true label for the masked positions in the given input sequence. The MLM loss function can be expressed as

$$\mathcal{L}_{\text{MLM}} = -\frac{1}{N}\sum_{i=1}^{N}\log(P(y_i|x)), \tag{1}$$

where N is the number of masked positions, y_i is the true label for the i-th masked position, $P(y_i|x)$ is the model's predicted probability for the i-th masked position given the input sequence x. We initialize the model using the pre-trained DNABERT model and incorporate MLM as one of our overall objectives. Through MLM, the model not only considers contextual information when understanding gene sequences but also takes into account information about the missing parts of the sequences, thereby enhancing its understanding of gene sequences.

2.5 Contrastive learning with knowledge-aware negative sampling

To further enhance the model's understanding of Gene Ontology knowledge, we employ contrastive learning to achieve low-dimensional representations of entities and relationships. In traditional contrastive learning, negative samples are usually generated through random or negative sampling methods. However, this approach may lead to insufficient learning of negative samples or suboptimal performance.

To address this issue, we introduce a knowledge-aware negative sampling strategy, incorporating rich GO knowledge to improve the process. Specifically, we leverage the structural information and entity relationships in the GO knowledge graph to guide the selection of negative samples, making them more "meaningful". This approach strengthens the model's understanding of GO knowledge and enhances the effectiveness of contrastive learning.

In contrastive learning, negative samples [26] are generated through a corruption distribution. The basic idea of the corruption distribution is to randomly "corrupt" an entity in a positive triplet to generate a negative triplet. This process forces the model to learn more discriminative features and key characteristics. Specifically, negative samples are obtained by randomly selecting and replacing the head or tail entity. We use a strategy where the head entity is fixed, and the tail entity is randomly sampled to generate negative samples. Through this method, the model not only learns the correct relationships between entities but also learns to identify negative samples that share similar features with the positive samples but do not actually conform to the true relationships.

To ensure that negative samples are more representative at the knowledge level, we incorporate GO knowledge into the selection of negative samples. The GO knowledge graph describes information such as molecular functions, cellular components, and biological processes of genes. By mapping this knowledge, we ensure that each negative sample has a certain degree of similarity to the positive sample, but the actual relationship is mismatched. This approach effectively reduces the generation of meaningless negative samples, ensuring that each negative sample provides valuable information for model learning, thus improving the quality of contrastive learning. As our KE target, we employ the loss suggested by Sun et al. [27], defined as

$$\mathcal{L}_{\mathrm{KE}} = -\log\sigma(\gamma - d(h,t)) - \sum_{i=1}^{n} \frac{1}{n} \log\sigma(d(h_i', t_i') - \gamma), \tag{2}$$

$$d_r(h,t) = \|h + r - t\|_p, \tag{3}$$

where (h_i', r, t_i') is the negative sample, both head and tail entities are randomly sampled from the corrupted tuple, n is the number of negative samples, γ denotes the margin, σ is the sigmoid function, and d is the scoring function. We employ the scoring function in (3), inspired by TransE [28]. This scoring function transforms the vector representations of the head entity, relation, and tail entity to make the representation of the relation tuple in vector space retain as much of the original structural information as possible. In this way, the entities and relations in the knowledge graph can be mapped into a low-dimensional continuous vector space. We employ a negative sampling technique, that fixes the head entities and randomly samples the tail entities, and assume that the paradigm p is 1.

We define the set of triplets in the knowledge as T, containing both $T_{\text{Gene-GO}}$ and $T_{\text{GO-GO}}$, and the entity set as E. All entities fall into one of three categories: molecular functions (MFO), cellular components (CCO), or biological processes (BPO) since GO provides three facets of knowledge in the biological domain.

Finally, defining the negative T' and positive triples as (h, r, t), the sampling process can be described as

$$T'_{\text{GO-GO}(h,r,t)} = \{(h', r, t) \mid h' \in E', h \in E'\} \cup \{(h, r, t') \mid t' \in E', t \in E'\}, \tag{4}$$

$$T'_{\text{Gene-GO}(h,r,t)} = \{(h, r, t') \mid t' \in E'\}, \tag{5}$$

where only the tail entity $E' \in \{E_{\mathrm{MFO}}, E_{\mathrm{CCO}}, E_{\mathrm{BPO}}\}$ and $T_{\text{Gene-GO}}$ is replaced.

The contrastive learning method with knowledge-aware negative sampling effectively promotes the model to learn more discriminative and meaningful features, avoiding the inefficiency or overfitting issues brought by traditional random negative sampling methods. By incorporating the GO knowledge graph into the negative sample generation process, the model is better able to capture the deep connections between gene sequences and GO terms. Ultimately, the optimization of negative samples not only improves the effectiveness of contrastive learning but also further supplements the introduction of external knowledge during the MLM fine-tuning process, thereby enhancing the representation ability of gene sequences.

2.6 Training objectives

We address the challenge of integrating heterogeneous information from GO annotation texts and gene sequences by encoding them together. To achieve this, we design an overall loss function that jointly

optimizes the objectives of MLM and KE. The total loss function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{KE}}, \tag{6}$$

where $\mathcal{L}_{\text{MLM}}$ represents the loss of MLM and $\mathcal{L}_{\text{KE}}$ represents the loss of KE.

By jointly optimizing these two objectives, the model not only improves its understanding of gene sequences but also effectively integrates external knowledge from the Gene Ontology knowledge graph into the learning of gene sequences. This joint optimization enhances the relationship between gene sequences and GO concepts, improving both the accuracy and generalization ability of the model in gene function annotation tasks.

2.7 Promoter prediction

The task of the promoter prediction module is to predict the presence of a promoter based on the gene sequence and also predict its strength. In this study, the workflow of the promoter prediction module includes the following steps.

Firstly, the gene sequence is encoded through the gene encoder. The gene encoder is based on the pre-trained DNABERT model and utilizes the masked gene modeling method to pre-train the gene sequence and extract latent features from it. MLM works by randomly masking some k-mers in the input sequence and predicting them using context information, allowing the model to capture both global and local features of the sequence. While encoding the gene sequence, we also introduce a knowledge embedding objective, where we use the knowledge embedding technique to map entities and relations in the Gene Ontology into a continuous low-dimensional vector space.

The encoded gene sequence representations are then pooled using the mean pooling operation, resulting in a comprehensive feature representation for each gene sequence. Next, a linear layer is used for classification to predict whether the gene sequence contains a promoter region. Specifically, the model predicts the presence of the promoter region by processing the encoded vector representation.

During the training process, the cross-entropy loss function is used to measure the discrepancy between the predicted category and the true labels, and the model is optimized via backpropagation to improve its classification ability. By jointly training the masked language modeling and knowledge graph embedding representations, the model can establish connections between global and local contexts, thereby improving the accuracy of promoter prediction in lower-dimensional space.

3 Experiment

We have conducted experiments to validate the efficacy of our method. During the fine-tuning phase, we curated a novel knowledge graph dataset incorporating gene sequences and gene ontologies. Subsequently, we conducted experiments on promoter prediction and intensity classification, leveraging the TAPE benchmark component to showcase the validity of our approach.

3.1 Dataset collection

Datasets utilized in this study were collected from various sources and are accessible to the public.

Fine-tuning dataset. The gene sequences utilized for fine-tuning were sourced from the National Library of Medicine (https://www.ncbi.nlm.nih.gov), which integrates publicly available databases including Genbank and PubMed, providing a classification number for each species. From the NCBI database, we extracted 20006 human protein-coding genes. Gene Ontology data were obtained from the GENE ONTOLOGY database (https://www.geneontology.org), the largest repository of gene function information worldwide. We download the structural data and annotation data of the Gene Ontology from this database. Leveraging these datasets, we constructed a new dataset named GeneKG, aligning descriptions with gene sequences to GO terms and gene entities, respectively.

Promoter dataset. A portion of the promoter datasets used for training and testing is sourced from the Eukaryotic Promoter Database, including a curated, non-redundant collection of eukaryotic Pol II promoters annotated with experimentally determined transcription start sites from 4806 species (available at https://epd.expasy.org/epd). Human TATA and non-TATA promoter data, including sequences from -249 to $+50$ bp and -34 to $+35$ bp, were downloaded from EPDnew [29]. Another portion of the dataset utilized benchmark data from Li et al. [3], where promoter sequences were sourced from RegulonDB,

Table 1 Parameter settings for each period.

Task	Warmup_ratio	Learning_rate	Optimizer	Frozen_bert
Fine-tuning	0.1	2E−5	SGD	–
Promoter	0.1	1E−5	SGD	False
Strength	0.08	1E−5	SGD	False

Table 2 Experimental results of OntoGene.

Predictor	Accuracy	F1	Recall
Promoter	0.967	0.983	0.967
Strength	0.985	0.992	0.985

consisting of 3382 promoter and 3382 non-promoter samples. Among the 3382 promoter samples, 1591 were strong promoters and 1792 were weak promoters used for identifying promoter strength. The entire dataset was split into 80% for training, 10% for testing, and 10% for validation.

Benchmark dataset. We acquired five datasets to benchmark on our model in order to determine whether it is a promoter or not: Le et al. [3], He et al. [30], Shujaat et al. [8], Hernández et al. [31], and Nagda et al. [32]. We experiment on these benchmark datasets to validate the efficacy and generalizability of our method.

3.2 Performance evaluation metrics

To evaluate the prediction quality more intuitively, we employed three distinct rating metrics that have been previously used in other state-of-the-art techniques. To more intuitively evaluate prediction quality, we employed three distinct evaluation metrics that have been adopted by other state-of-the-art techniques. They are F1 Score (F1), Accuracy, Recall. The corresponding mathematical expressions for these metrics are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}, \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$\text{F1} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{9}$$

In the above equation, TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively.

3.3 Experimental setup

When fine-tuning the model and identifying promoters, we conducted our experiments using Python 3.8 and PyTorch 1.9.0. We initialized our models with pre-trained DNABERT and PubMedBERT, and utilized a small-batch SGD optimizer with a weight decay of 0.1 on the dataset. Detailed parameters are reported in Table 1.

3.4 Results and comparison with existing promoter prediction methods

Our model, OntoGene, achieved the results presented in Table 2 for identifying promoters, non-promoters, and their respective intensities. From Table 2, we observe that OntoGene achieved an Accuracy and Recall of 96.7%, and an F1 Score of 98.3% for identifying promoters and non-promoters. Regarding the identification of promoter intensity, OntoGene achieved an accuracy and recall of 98.5%, and an F1 Score of 99.2%.

These results indicate that our model, OntoGene, successfully integrates Gene Ontology knowledge and textual information. The results show that OntoGene is effective in identifying the promoters. By jointly encoding and optimizing gene sequences and GO annotation texts, our model can better understand gene regulatory mechanisms, enabling accurate identification and classification of promoters.

Baselines. In our experiments, we conducted a comparative analysis of OntoGene against four baseline models to determine its effectiveness in promoter identification. The first is a lightweight promoter prediction and classification model called PromoterLCNN [31], which is based on convolutional neural

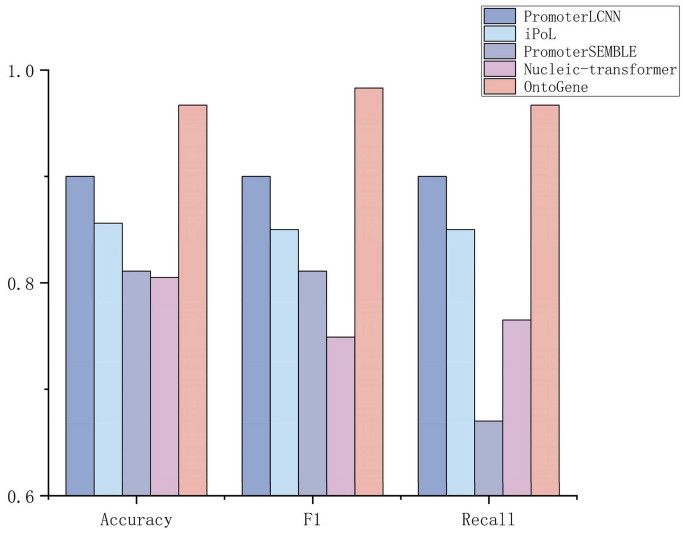

Figure 4 (Color online) Experimental results for the four baseline methods and OntoGene.

networks. The second is PromoterSEMBLE [32], which proposes a new integrated learning technique using deep recurrent neural networks with convolutional feature extraction and hard negative pattern mining to detect several types of promoter types. The third model is Nucleic-transformer [30], which applies self-attention and convolution to process nucleic acid sequences for outstanding performance in viral genome identification and E. coli promoter classification. Finally, iProL [33] adopts a CNN + BiLSTM combination for promoter prediction. For each approach, we employ the same experimental setup and dataset. The experimental results for these four baseline methods in identifying promoters and OntoGene are illustrated in Figure 4.

Results. Figure 4 illustrates that our model outperforms most baseline methods in predicting promoters. Specifically, our model significantly exceeds other methods in Accuracy, F1, and Recall, with scores of 0.967, 0.983, and 0.967, respectively. PromoterLCNN follows closely behind, achieving scores of 0.9 in all three metrics, nearly 0.1 lower than our model. In contrast, the Nucleic-transformer performs the worst, with scores of 0.805, 0.749, and 0.765 for Accuracy, F1, and Recall, respectively, making our model nearly 0.2 higher. This is because none of these four base models incorporate knowledge of external Gene Ontology. These findings demonstrate that OntoGene benefits from such rich biological knowledge, exhibiting strong predictive and generalization capabilities upon integrating external Gene Ontology knowledge.

3.5 Evaluation on other promoter datasets

Baselines. We further evaluate our method on five benchmark datasets used in other research papers. Specifically, DeepPromoter used the benchmark dataset from Oubounyt et al. [6], which included 3382 promoter and 3382 non-promoter samples. PromoterSEMBLE used 29598 human promoter sequences from the EPD database. pcPromoter-CNN used 5720 E. coli promoter sequences. Nucleic-transformer used 5720 human TATA and non-TATA promoters (300). PromoterLCNN used 5720 E. coli promoter sequences. Figure 5 provides a summary of the experimental results.

Results. From Figure 5, we can observe that our model achieves good accuracy across all five benchmark datasets. The experiments compare five datasets, and the algorithm comparison for each dataset is as follows. On the PromoterLCNN dataset, the Accuracy, F1 Score, and Recall of PromoterLCNN are 0.937, 0.942, and 0.931, while our model achieves 0.987, 0.993, and 0.987, outperforming it by nearly 5%. On the pcPromoter-CNN dataset, pcPromoter-CNN scores 0.898, 0.903, and 0.901 in these three evaluation metrics, while our model surpasses it by nearly 9%. On the PromoterSEMBLE dataset, PromoterSEMBLE's scores are 0.917, 0.992, and 0.917, and our model outperforms it by nearly 5%. On the DeepPromoter dataset, DeepPromoter's scores are 0.917, 0.917, and 0.835, while our model surpasses it by nearly 8%. In contrast, on the Nucleic-transformer dataset, the Nucleic-transformer's performance is slightly inferior, with scores of 0.883, 0.766, and 0.893, and our model outperforms it by nearly 10%.

To provide a more comprehensive view of the performance of each algorithm on the datasets, we have

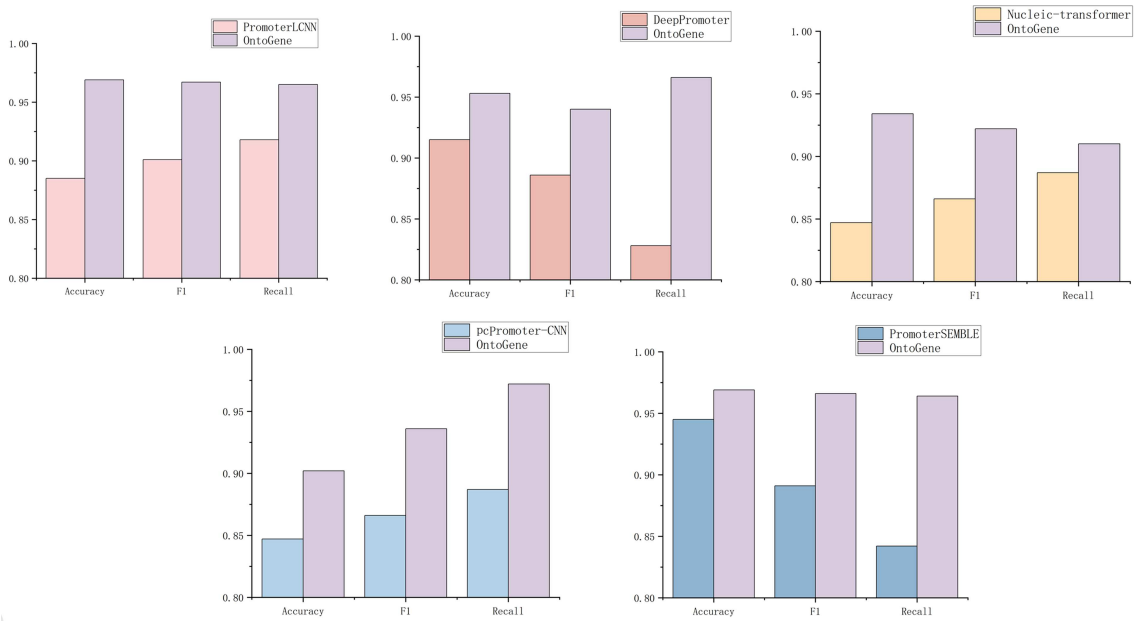

Figure 5 (Color online) Results of the five datasets' experiments on OntoGene.

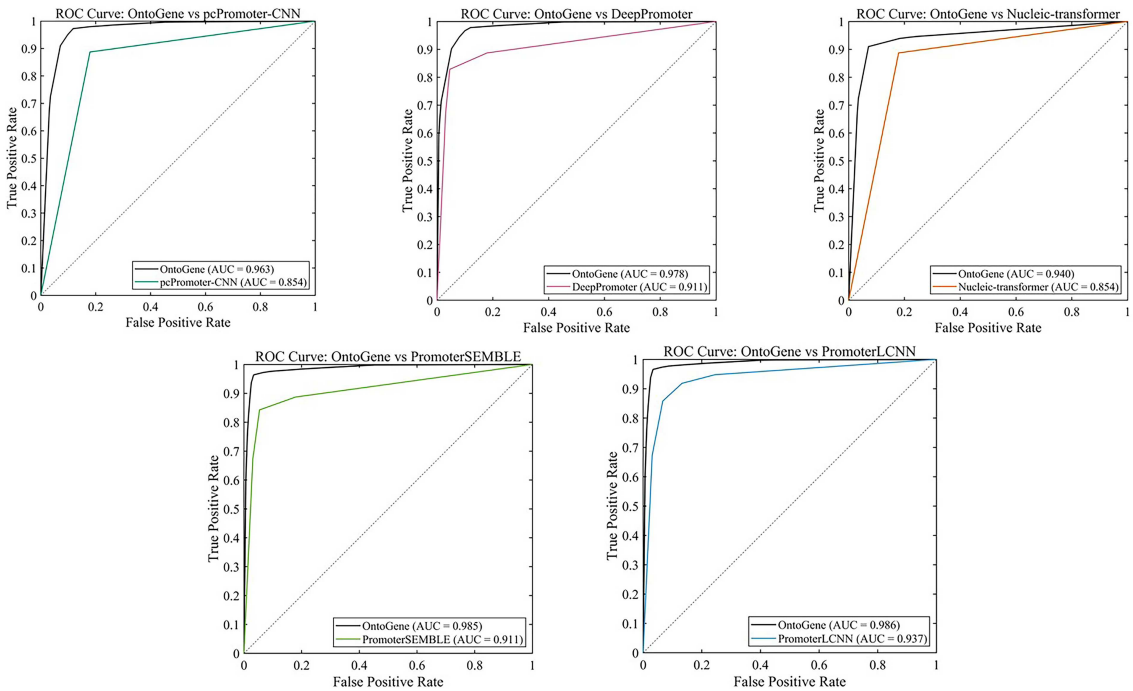

Figure 6 (Color online) ROC curves.

plotted the corresponding ROC curves, as shown in Figure 6. It is evident that OntoGene's ROC curve is closer to the upper-left corner, indicating that the model can maintain a lower false positive rate while achieving a high true positive rate. These results clearly demonstrate the generalization ability of our method and further validate the robustness and classification performance of the model through the ROC curve.

3.6 Ablation experiments

To further validate the effectiveness of integrating Gene Ontology and contrastive learning, we conducted an ablation study comparing OntoGene with models that do not incorporate external knowledge and

Table 3 Ablation experiments. The best results are in bold.

Knowledge	Contrastive learning	Accuracy	F1	Recall
✓	✓	**0.967**	**0.983**	**0.967**
✓	×	0.880	0.873	0.846
×	✓	0.893	0.841	0.867

lack the contrastive learning module. We employed the same experimental setup and datasets, and the results are presented in Table 3.

As shown in Table 3, OntoGene significantly outperforms the model without the integration of the knowledge graph in promoter prediction, with improvements of 0.165, 0.181, and 0.165 in Accuracy, Recall, and F1 Score, respectively. This demonstrates that OntoGene benefits from the rich biological knowledge graph, leading to superior performance.

On the other hand, in the absence of contrastive learning, the model's performance on all evaluation metrics is inferior to that of the OntoGene model. This indicates that without contrastive learning, the representation between gene sequences and GO terms fails to fully capture the deep connections between them, which in turn affects the model's reasoning and predictive capability. This underscores the crucial role of contrastive learning in enhancing the representation of the relationship between gene sequences and GO terms, significantly improving the model's performance.

In summary, the ablation experiments validate the importance of Gene Ontology knowledge and contrastive learning in the promoter prediction task. By employing contrastive learning, gene sequences and GO knowledge are optimized in the same representation space, thus enhancing the overall performance of the model. The integration of Gene Ontology knowledge and the contrastive learning mechanism results in significant advantages for the OntoGene model in metrics such as Accuracy, Recall, and F1 Score. This demonstrates that the combination of external biological knowledge and the joint optimization strategy of contrastive learning is an effective approach to improving promoter prediction performance.

4 Conclusion

The classification of promoter and non-promoter DNA sequences is an essential task in medicine and bioinformatics, as it holds significance in understanding gene regulation, disease diagnosis and treatment, and bioinformatics research. In this regard, identifying the strength of promoter DNA sequences is particularly crucial, as the strength of promoters can reflect gene expression levels, thereby influencing various biological processes within organisms. External knowledge graphs can provide rich biological structural information, such as biological knowledge in Gene Ontology. By integrating this informative biological knowledge, we can enhance our understanding of promoters, thus improving the accuracy of promoter identification.

In this paper, we propose a simple yet effective method to embed the biological structural facts of Gene Ontology into the BERT model to facilitate the identification and classification of promoters. Our method encodes both gene sequences and knowledge graphs into vectors using a hybrid Transformer-based encoder and introduces a contrastive learning method with knowledge-aware negative sampling. We also jointly optimize the embeddings of genes and knowledge graphs by combining knowledge embedding and masked language modeling objectives. This approach enables us to accurately identify and classify promoters directly from sequence data.

Experimental results demonstrate that effective knowledge injection can significantly improve promoter prediction accuracy. Our method not only predicts the presence of DNA promoters, but also predicts their strength. In most evaluation metrics, our method outperforms the baseline, and it performs well in different benchmark datasets, further confirming the effectiveness and versatility of our approach.

However, despite the promising results achieved by our method in promoter prediction, there are still some limitations. First, the current model relies on a fixed Gene Ontology knowledge graph, and for specific biological species or new gene function annotations, it may not be updated in a timely manner, resulting in limitations in the timeliness and comprehensiveness of the knowledge. Second, although we have introduced a knowledge-aware negative sampling strategy, the selection of negative samples can still be further optimized. Specifically, how to effectively reduce the impact of noise and irrelevant negative samples remains an important area for future improvement.

Future research can focus on the following aspects: on one hand, updating knowledge graphs tailored

to specific species and implementing multi-level knowledge injection strategies could enhance the model's generalization ability across different species; on the other hand, combining more refined negative sample generation mechanisms could further improve the model's robustness and accuracy. Additionally, the model's adaptability when faced with larger-scale and more diverse biological data will also be a promising direction for exploration. Through these improvements, we expect to further enhance the model's performance and generalization ability, providing more powerful tools for bioinformatics research.

Acknowledgements This work was partially supported by National Natural Science Foundation of China (Grant Nos. 62450112, 62225109, 62372098) and Key Project of the Natural Science Foundation of Heilongjiang Province (Grant No. ZD2024F001). We thank the anonymous reviewers for their constructive comments.

References

1 Pedersen A G, Baldi P, Chauvin Y, et al. The biology of eukaryotic promoter prediction—a review. Comput Chem, 1999, 23: 191–207
2 Le N Q K, Do D T, Nguyen T T D, et al. A sequence-based prediction of Kruppel-like factors proteins using XGBoost and optimized features. Gene, 2021, 787: 145643
3 Le N Q K, Yapp E K Y, Nagasundaram N, et al. Classifying promoters by interpreting the hidden information of DNA sequences via deep learning and combination of continuous FastText N-grams. Front Bioeng Biotechnol, 2019, 7: 305
4 Mishra A, Dhanda S, Siwach P, et al. A novel method SEProm for prokaryotic promoter prediction based on DNA structure and energetics. Bioinformatics, 2020, 36: 2375–2384
5 Solovyev V V, Umarov R K. Prediction of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. 2016. ArXiv:1609.04133
6 Oubounyt M, Louadi Z, Tayara H, et al. DeePromoter: robust promoter predictor using deep learning. Front Genet, 2019, 10: 286
7 Xiao X, Xu Z C, Qiu W R, et al. iPSW(2L)-PseKNC: a two-layer predictor for identifying promoters and their strength by hybrid features via pseudo K-tuple nucleotide composition. Genomics, 2019, 111: 1785–1793
8 Shujaat M, Wahab A, Tayara H, et al. pcPromoter-CNN: a CNN-based prediction and classification of promoters. Genes, 2020, 11: 1529
9 Ma Z W, Zhao J P, Tian J, et al. DeeProPre: a promoter predictor based on deep learning. Comput Biol Chem, 2022, 101: 107770
10 Le N Q K, Ho Q T, Nguyen V N, et al. BERT-Promoter: an improved sequence-based predictor of DNA promoter using BERT pre-trained model and SHAP feature selection. Comput Biol Chem, 2022, 99: 107732
11 Zaytsev K, Fedorov A, Korotkov E. Classification of promoter sequences from human genome. Int J Mol Sci, 2023, 24: 12561
12 Zhang N, Bi Z, Liang X, et al. OntoProtein: protein pretraining with gene ontology embedding. 2022. ArXiv:2201.11147
13 Blake J A, Rutherford K M, Chan J, et al. Gene Ontology Consortium: going forward. Nucleic Acids Res, 2014, 43: D1049–D1056
14 Ashburner M, Ball C A, Blake J A, et al. Gene Ontology: tool for the unification of biology. Nat Genet, 2000, 25: 25–29
15 Yang Z, Dai Z, Yang Y, et al. XLNet: generalized autoregressive pretraining for language understanding. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 32: 5754–5764
16 Le T, Le N, Le B. Knowledge graph embedding by relational rotation and complex convolution for link prediction. Expert Syst Appl, 2022, 214: 119122
17 Yan Q, Fan J, Li M, et al. A survey on knowledge graph embedding. In: Proceedings of the 7th IEEE International Conference on Data Science in Cyberspace (DSC), 2022. 576–583
18 Zhang Z, Cao L, Chen X, et al. Representation learning of knowledge graphs with entity attributes. IEEE Access, 2020, 8: 7435–7441
19 Logeswaran L, Chang M W, Lee K, et al. Zero-shot entity linking by reading entity descriptions. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019. 3449–3460
20 Devlin J, Chang M, Lee K, et al. BERT: pre-training of deep bidirectional transformers for language understanding. Proc NAACL, 2019, 1: 4171–4186
21 Ji Y, Zhou Z, Liu H, et al. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. bioRxiv, 2020. doi: 10.1101/2020.07.11.198044
22 Gu Y, Tinn R, Cheng H, et al. Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthcare, 2020, 3: 1–23
23 Wang Q, Mao Z, Wang B, et al. Knowledge graph embedding: a survey of approaches and applications. IEEE Trans Knowl Data Eng, 2017, 29: 2724–2743
24 Wang S, Xu F, Li Y, et al. KG4SL: knowledge graph neural network for synthetic lethality prediction in human cancers. Bioinformatics, 2021, 37: i418–i425
25 Wang X, Gao T, Zhu Z, et al. KEPLER: a unified model for knowledge embedding and pre-trained language representation. Trans Assoc Comput Linguist, 2021, 9: 176–194
26 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems, 2013. 26: 3111–3119
27 Sun Z, Deng Z, Nie J, et al. RotatE: knowledge graph embedding by relational rotation in complex space. 2019. ArXiv:1902.10197
28 Bordes A, Usunier N, Garcia-Durán A, et al. Translating embeddings for modeling multi-relational data. In: Proceedings of Advances in Neural Information Processing Systems, 2013. 26: 2787–2795
29 Meylan P, Dreos R, Ambrosini G, et al. EPD in 2020: enhanced data visualization and extension to ncRNA promoters. Nucleic Acids Res, 2020, 48: D65–D69
30 He S, Gao B, Sabnis R, et al. Nucleic transformer: classifying DNA sequences with self-attention and convolutions. ACS Synth Biol, 2023, 12: 3205–3214
31 Hernández D, Jara N, Araya M, et al. PromoterLCNN: a light CNN-based promoter prediction and classification model. Genes, 2022, 13: 1126
32 Nagda B M, Nguyen V M, White R T. promSEMBLE: hard pattern mining and ensemble learning for detecting DNA promoter sequences. IEEE ACM Trans Comput Biol Bioinf, 2023, 21: 208–214
33 Peng B, Sun G, Fan Y. iProL: identifying DNA promoters from sequence information based on longformer pre-trained model. BMC Bioinf, 2024, 25: 224