# Unveiling Protein-DNA Interdependency: Harnessing Unified Multimodal Sequence Modeling, Understanding and Generation

Mingchen Li[1,2†], Yuchen Ren[2†], Peng Ye[2†], Jiabei Cheng[1,2,5†], Xinzhu Ma[2], Yuchen Cai[2], Wanli Ouyang[2], Bozitao Zhong[1,4], Banghao Wu[1,4], Nanqing Dong[2*], Liang Hong[1,2,3,4*], Pan Tan[1,2*]

[1]Shanghai National Centre for Applied Mathematics (SJTU Center), MOE-LSC, School of Physics and Astronomy & Institute of Natural Sciences, Shanghai Jiao Tong University, Shanghai 200240, China.
[2]Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.
[3]Zhangjiang Institute for Advanced Study, Shanghai Jiao Tong University, Shanghai 201203, China.
[4]School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China.
[5]Department of Automation, School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.

*Corresponding author(s). E-mail(s): dongnanqing@pjlab.org.cn; hongl3liang@sjtu.edu.cn; tpan1039@gmail.com;
Contributing authors: limc.19980301@gmail.com; renyuchen@pjlab.org.cn; yepeng@pjlab.org.cn; jiabei_cheng@sjtu.edu.cn; maxinzhu@pjlab.org.cn; caiyuchen@pjlab.org.cn; ouyangwanli@pjlab.org.cn; zbztzhz@gmail.com; wubh20@sjtu.edu.cn;
†These authors contributed equally to this work.

**Abstract**

The interactions between proteins and non-coding DNA remain underexplored in the context of large language models. In this study, we present a multimodal framework designed to investigate these critical interactions, utilizing a comprehensive pre-training dataset comprising over 129 million pairs of coding

and non-coding sequences. Our model effectively integrates protein sequences, coding sequences, and non-coding DNA elements through a BERT-like encoder and specialized decoder architecture. It demonstrates outstanding performance in zero-shot predictions and various supervised tasks, particularly in accurately forecasting metabolic pathway characteristics and optimizing protein expression through promoter and coding sequence manipulation. By elucidating the intricate relationships between proteins and non-coding DNA, this framework has the potential to advance protein engineering and metabolic pathway optimization in synthetic biology, paving the way for efficient biotechnological applications and sustainable production processes. This work underscores the importance of integrating multi-level biological data to drive innovation in synthetic biology.

# 1 Introduction

The field of synthetic biology represents a transformative frontier in science, dedicated to the engineering of biological systems and components in order to innovate novel biological functions and systems. Its primary objective is to construct intricate biological networks by employing fundamental biological modules, integrating protein engineering methodologies to design efficient protein components, and utilizing metabolic engineering to develop rational and effective gene regulation elements. This multifaceted approach facilitates unprecedented applications in areas such as biomanufacturing, energy production, disease treatment, and environmental remediation [1]. Through the construction and optimization of metabolic pathways, synthetic biology significantly enhances the yield of proteins and target small molecule products, which include bioactive compounds, therapeutic proteins, and industrial enzymes. Researchers have successfully engineered microorganisms for the production of biofuels and high-value pharmaceuticals, as well as for the reconstruction of metabolic pathways that detoxify environmental pollutants. The rising demand for high-value protein products, such as antimicrobial proteases that address resistance issues stemming from antibiotic misuse, highlights their critical role in external sterilization efforts [2]. Additionally, the development of sweet-tasting proteins as nontoxic food additives—characterized by high sweetness and essential amino acids—demonstrates market promise [3]. Nevertheless, the widespread application of these crucial proteins remains hindered by typically low biosynthetic yields, which contribute to elevated production costs.

Despite its substantial potential, synthetic biology grapples with significant technological challenges in practical applications. Notably, the complexities associated with protein design and gene sequence design within metabolic pathways continue to present obstacles, as existing tools and methodologies often struggle to accurately forecast the implications of specific enzyme mutations and gene regulation elements throughout entire metabolic networks. For instance, while techniques such as directed evolution can enhance the catalytic activity of specific key enzyme components in vitro, their integration, along with strong promoter elements, into cellular systems frequently results in negligible increases in final product yields, and can even lead to

reductions due to complex feedback inhibition effects among enzymes and regulatory DNA elements within a metabolic pathway [4]. Optimization of multiple metabolic components must simultaneously account for interactions among different elements along the entire pathway. Traditional approaches, which typically involve testing various combinations of metabolic components one by one, are both time-consuming and labor-intensive.

Recent advancements in deep learning—particularly the development of large language models for protein and DNA—have markedly progressed protein structural prediction [5], protein functional prediction [6], and the generation of protein and DNA sequences [7, 8]. These models excel in tasks such as predicting promoters and enhancers [9, 10], demonstrating remarkable zero-shot or few-shot predictive capabilities in protein mutation-fitness assessments [11, 12], thereby significantly reducing experimental costs [13]. However, there has yet to be a prominent application or exploration of large language models in the context of protein-non-coding DNA interactions for the modification of metabolic pathways within synthetic biology.

The present study introduces ProDMM (Protein-DNA Multi-Modality Language Model), a new deep learning framework that utilizes a multi-modality approach for the analysis of protein and DNA sequences. ProDMM has been pre-trained in a self-supervised manner on over 129 million paired sequences, allowing it to integrate the semantic features of protein sequences, coding DNA sequences (CDS), and their adjacent non-coding DNA sequences (NCDS), while capturing their interdependencies and interactions. The model facilitates the simultaneous input of DNA and protein sequence data within a unified representation and features a multi-modality sequence encoder primarily supported by Masked Language Modeling (MLM) pre-training alongside a cross-modality autoregressive generative decoder.

In benchmarking studies, ProDMM outperformed existing models that focus exclusively on protein sequences—such as ESM and ProtTrans [5, 14]—as well as those dedicated solely to DNA sequences, like CaLM, DNABERT2, GenSLM2.5B and Evo [8, 10, 15, 16]. This superior performance was observed across various tasks, including zero-shot protein mutation-fitness prediction, melting temperature and solubility prediction of native protein sequences, and protein abundance prediction. ProDMM further excelled in zero-shot mutation-fitness prediction tasks involving nucleotide sequences, such as the properties of non-coding RNA, zero-shot predictions targeting promoter-protein transcription, and CDS codon optimization, outperforming models such as Evo [8] and GenSLM 2.5B [16]. A notable feature of ProDMM is its promoter generation module, which is uniquely conditioned on downstream CDS. Considering the co-evolutionary interactions between NCDS and CDS, which together influence DNA transcription and protein translation efficiencies [17], our results show that ProDMM's architecture—integrating upstream and downstream non-coding sequences with coding data—generates promoter sequences with improved translation level compared to unconditional generation. Moreover, ProDMM's explicit modeling of interactions between protein sequences and their associated upstream and downstream DNA sequences enables zero-shot predictions of properties across entire metabolic pathways, without the need for labeled training data, significantly advancing current protein and nucleotide models. In summary, ProDMM holds great potential

in synthetic biology, protein engineering, promoter design, codon optimization, and metabolic pathway optimization, offering significant advancements in these fields.

# 2 Results

## 2.1 Architecture of ProDMM

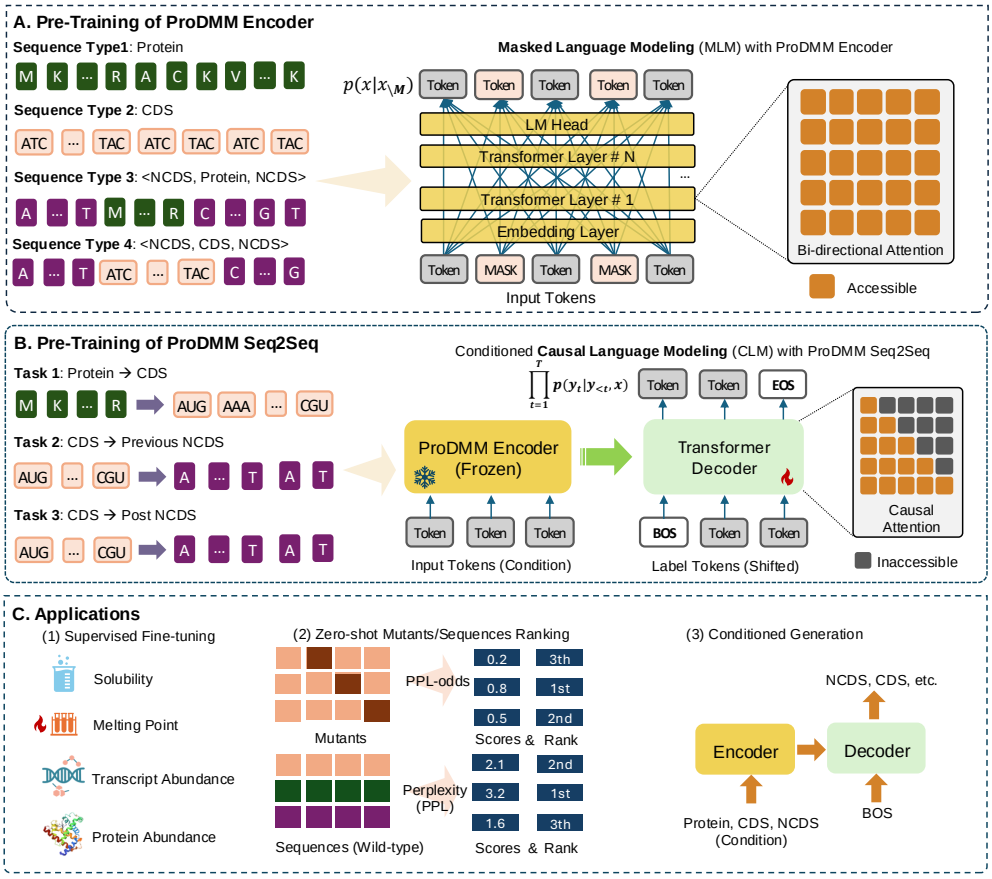


**Figure 1:** The architecture of ProDMM: (A) The ProDMM Encoder, a Transformer encoder, was pre-trained using masked language modeling on four types of sequences. (B) The ProDMM Seq2Seq equips the ProDMM Encoder with an additional decoder and was pre-trained with conditioned causal language modeling across three generation tasks. (C) Applications of ProDMM. It can be used as a pre-trained model for supervised fine-tuning across various tasks. It also functions as a model for scoring and ranking sequences or mutants (ProDMM Encoder) and for conditioned sequence generation (ProDMM Seq2Seq).

ProDMM is a Transformer-based [18] model consisting of two sub-models: ProDMM Encoder and ProDMM Seq2Seq. The ProDMM Encoder, as illustrated in Figure 1A, is a BERT-style [19] Transformer encoder that has been pre-trained using masked language modeling. Additionally, the ProDMM Seq2Seq, shown in Figure 1B, equipped the Transformer encoder with an additional Transformer decoder through cross-attention layers. Both the Transformer decoder and the cross-attention layer are pre-trained with auto-regressive generation tasks. Once the pre-training phase is completed, ProDMM can be utilized in various scenarios and tasks, as demonstrated in 1C. The model was pretrained on the GTDB (Genome Taxonomy DataBase), with the processing details outlined in the section of Methods, resulting in the approximately 129 million paired sequences. Figure S1A illustrates the length distribution of protein-coding sequences (CDS) within the dataset, revealing that sequences ranging from 200 to 400 nucleotides comprise the majority. Additionally, Figure S1B presents the species distribution at the phylum level within the pre-training dataset. This extensive dataset facilitates robust learning of the intricate relationships and dependencies among the various sequence types, thus enhancing the model's ability to predict functional attributes within biological systems.

### 2.1.1 ProDMM Encoder

The encoder of ProDMM is built upon a Transformer encoder architecture with bi-directional attention (Figure 1A). The encoder is pre-trained with a masked language modeling (MLM) objective to learn contextualized embeddings of tokens and sequence distribution. This task incorporates four types of sequence formats: Protein, CDS, NCDS-Protein-NCDS, and NCDS-CDS-NCDS, as illustrated in the left of Figure 1B. During the pre-training, 15% tokens from the input sequence are randomly chosen, of which 80% are replaced with a special <mask> token, 10% are swapped with different tokens, and the remaining 10% are left unchanged. This methodology can enhance the model's ability to capture dependencies between protein sequences and the regulatory sequences (NCDS), as well as to gather contextual information throughout the entire sequence. By leveraging these masked and altered inputs, the ProDMM encoder is better equipped to learn the intricate relationships within protein and DNA sequences. Details regarding this approach can be found in the Method Section A.2.

We observed that after pretraining, the ProDMM-Encoder can capture the relationships between codon tokens and amino acid tokens. As illustrated in Figure S2, we employed UMAP for dimensionality reduction to visualize the embedding vectors of different tokens generated by the ProDMM Encoder. The results indicate that codons corresponding to the same amino acid are clustered closely together, forming distinct groupings around their respective amino acids.

Moreover, Figure S2 reveals that amino acids with similar physicochemical properties exhibit relatively close embeddings. For instance, under neutral conditions, the positively charged amino acids arginine (R) and lysine (K) are located at a close distance to one another, while the negatively charged amino acids asparagine (N) and glutamine (Q) also share proximity in the embedding space. This observed clustering reflects the underlying biochemical similarities and functional relationships among the amino acids, as captured within the learned representations of the ProDMM Encoder.

Such patterns indicate the model's ability to encode meaningful relationships based on the physicochemical characteristics of the amino acids.

### 2.1.2 ProDMM Seq2Seq

The ProDMM-Seq2Seq is a complete Transformer model containing an encoder for encoding conditions and a decoder with cross attention layers for auto-regressive generation, as shown in Figure 1B. The encoder is inherited from the pre-trained ProDMM Encoder. ProDMM Seq2Seq is designed to perform the task of auto-regressive sequence generation. First, the ProDMM Encoder encodes the input sequence into hidden states, and then the Transformer Decoder utilizes Cross Attention to generate the next token based on these hidden states and the tokens that have already been generated. This decoder leverages the encoder's embeddings as input, effectively facilitating the modeling of complex protein-DNA interactions. The pre-training task of ProDMM-Seq2Seq encompasses three distinct conditional generation objectives: first, the generation of a coding sequence (CDS) based on an input protein sequence; second, the production of the pre-non-coding DNA sequence (pre-NCDS) that precedes the CDS; and third, the generation of the post-non-coding DNA sequence (post-NCDS) that follows the CDS. This architectural design enables the decoder to adeptly capture the dependencies between different sequence types, thereby ensuring that the generated sequences are not only contextually relevant but also biologically meaningful. By concentrating on these conditional generation tasks, ProDMM significantly augments its capacity to accurately predict and generate not only coding sequences but also the associated non-coding regions. The model's capability to integrate information from various sequence contexts allows for a more comprehensive understanding of the intricate relationships that govern gene expression. This innovative methodology positions ProDMM as a pivotal tool in genomic research, facilitating the exploration of gene regulatory networks and the design of more effective synthetic biology applications. By bridging the gap between protein, coding and non-coding sequences, ProDMM may contribute to the evolving landscape of molecular biology and genetic engineering. Its capability to elucidate interactions between proteins and non-coding elements has the potential to inform more precise and targeted genetic modifications, thus enhancing various biotechnological outcomes.

## 2.2 Benchmark Results of the Encoder

To evaluate the performance of the ProDMM encoder, we conducted a series of benchmarks across multiple downstream tasks, including zero-shot predictions of DNA and protein properties (Middle part of Figure 1C) as well as various supervised learning representation tasks (Left part of Figure 1C). Detailed methodologies can be found in the Methods Section A.4 and A.5.

### 2.2.1 Zero-Shot Tasks

Due to the distinctive architectural design of ProDMM, its encoder is proficient at capturing the intrinsic semantic and syntactic attributes of coding sequences (CDS), protein sequences, and their corresponding upstream and downstream non-coding

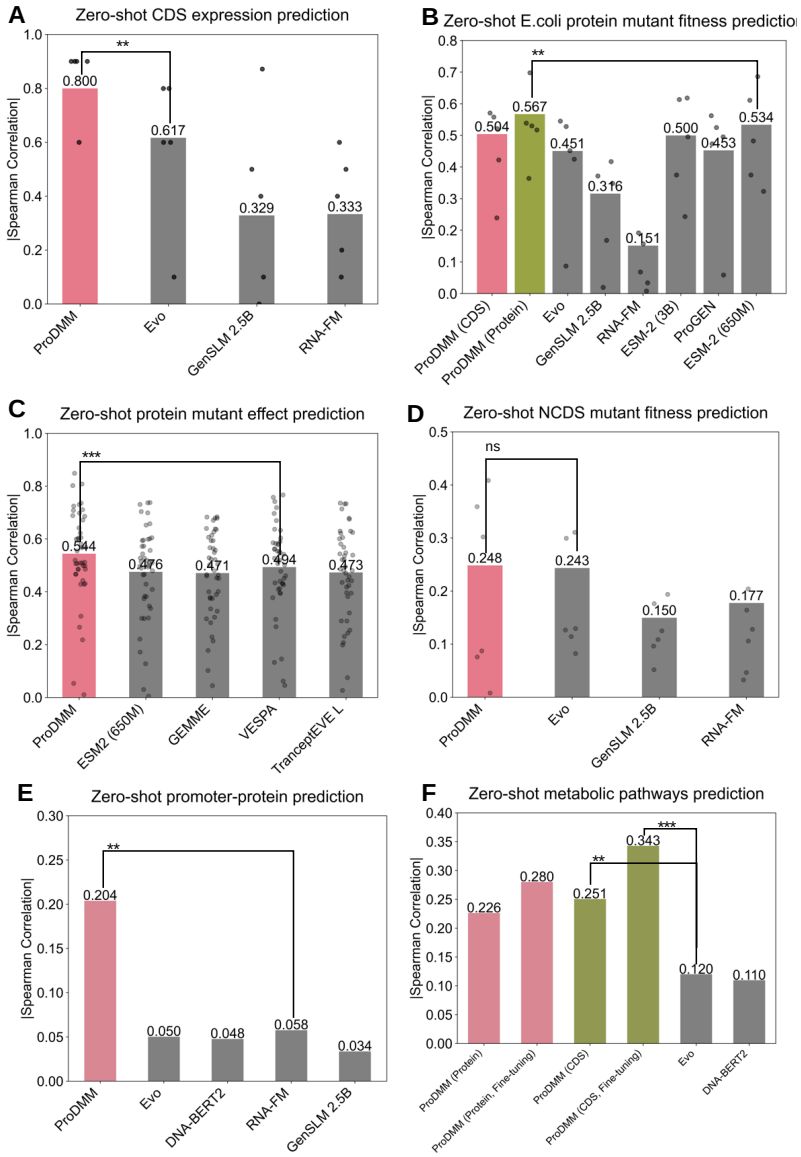


**Figure 2:** ProDMM performs zero-shot function prediction for CDS, proteins, NCDS, promoter-protein, and metabolic pathways. (A-D) Zero-shot prediction performance of ProDMM-Encoder. The bar height represents the Spearman rank correlation between the target sequence perplexity calculated by ProDMM-Encoder and the experimental fitness (expression level, activity, stability, etc.), while each dot corresponds to a specific dataset. (E) The Spearman correlation between the perplexity of promoter-protein complex sequence and expression level. (F) The Spearman correlation between the perplexity of metabolic (NCDS-Protein-NCDS-Protein-NCDS-Protein-NCDS-Protein) complex sequence and the experimental expression level. P-values (two-side t-test) are reported between the top two performing models; The asterics indicate the p-value, ranging from 0.01-0.05 (*), 0.001-0.01 (**), 0.0001-0.001 (***).

DNA sequences (NCDS). This capability extends to the effective assimilation of inter-molecular sequence interaction information, which is pivotal for understanding complex genetic networks. We hypothesized that the ProDMM-encoder possesses robust zero-shot predictive capabilities for individual attributes of CDS, protein, and non-coding sequences, as well as for the intricate interactions between promoter sequences (upstream NCDS) and protein sequences. To rigorously evaluate this hypothesis, we conducted a series of zero-shot benchmark assessments against various baseline models, thereby affirming ProDMM's predictive prowess.

#### *Zero-shot CDS expression prediction*

We present that the perplexity of CDS sequences calculated by the ProDMM-Encoder (for the method of calculating sequence perplexity, see Method Section A.4 and A.4.1) correlates with the expression levels of CDS sequences. To validate this, We constructed a dataset from literature [20], which contains expression levels from five different proteins. Each dataset records the expressive levels measured in experiments for different codons of the same protein. Figure 2A displays the Spearman rank correlation between the CDS perplexity and the experimental expression level, while each dot corresponds to a specific dataset. The results show a strong correlation between the perplexity and expression level (Average Spearman $r = 0.8$, two-sided t-distributed $p < 1 \times 10^{-5}$), indicating that lower perplexity corresponds to higher yield. We also compare ProDMM with alternative nucleotide baseline models, including Evo [8], RNA-FM [21], and GenSLM 2.5B [16]. ProDMM exhibits the best performance, demonstrating its superior ability to model gene expression accurately.

#### *Zero-shot E.coli protein mutant fitness prediction*

ProDMM-Encoder can predict the impact of CDS mutations on protein fitness. Specifically, the scoring of the mutant by ProDMM-Encoder is correlated with the fitness of the mutants (for the method of calculating the mutant score, refer to Method Section A.4 and A.4.3). We used a deep mutation scanning (DMS) dataset containing mutations from five proteins from E. coli, which was compiled by Evo [8]. Since the input of ProDMM-Encoder can be either protein or CDS, we separately tested the performance using protein as input and using CDS as input. As shown in Figure 2B, when ProDMM utilizes protein sequences as input, it demonstrates superior performance compared to other protein sequence models, such as ESM2 and progen [5, 7]. Conversely, when using CDS DNA sequences, ProDMM similarly excels against other nucleotide sequence models, including Evo [8], GenSLM 2.5B [16] and RNA-FM [21]. Moreover, ProDMM's performance with protein sequence input consistently surpasses that with CDS sequence input in this task.

#### *Zero-shot protein mutant fitness prediction*

ProDMM-Encoder, pre-trained with masked language modeling, can serve as a protein mutant fitness predictor [22] (for the method of calculating the mutant score, refer to Method Section A.4 and A.4.4). Given that ProDMM's training set is derived from the GTDB [23] and only comprises prokaryotic genomes, Figure 2C shows the ProDMM's performance in the mutation-fitness prediction task against all 45 prokaryotic protein

datasets within the ProteinGym benchmark [11]. ProDMM consistently outperformed other baseline sequence models, including ESM2 [5], GEMME [24], VESPA [25] and TranceptionEVE [26], affirming its robustness and reliability in predicting evolutionary fitness under various genetic contexts.

#### *Zero-shot NCDS mutant fitness prediction*

In Figure 2D, we assess the zero-shot prediction capabilities of ProDMM concerning non-coding RNA (ncRNA), leveraging data obtained from Evo's research [8]. In this endeavor, ProDMM employs its NCDS modality to input DNA sequences (translated from ncRNA) and once again outperforms other nucleotide models, including GenSLM [16] and RNA-FM [21], thereby reinforcing the model's comprehensive predictive abilities.

#### *Zero-shot promoter-protein prediction*

Moreover, in Figure 2E, we conduct an in-depth examination of a zero-shot prediction task targeting promoter transcription intensity in E. coli, drawing upon data found in the literature [27]. In this study, the authors investigated the fluorescence intensity of EGFP under diverse promoter conditions, thereby generating insights into transcriptional efficiencies. We provided ProDMM with both the NCDS and the corresponding protein sequence. Interestingly, although the EGFP protein sequence remained constant throughout the experimental evaluations, ProDMM's predictions of transcriptional intensity significantly outperformed those of other nucleotide baseline models. This finding suggests that ProDMM has adeptly captured the complex interactions between promoter sequences and protein sequences, highlighting their critical role in accurately predicting the transcriptional efficiency of downstream proteins—an aspect that has often been insufficiently explored in previous studies.

#### *Zero-shot metabolic pathways prediction*

Building upon the above findings, we propose that ProDMM, having assimilated an extensive array of interaction features between proteins and their upstream and downstream non-coding sequences during its pre-training phase, is endowed with the capacity for zero-shot predictive capabilities regarding metabolite yields in metabolic pathways. To validate this proposition, we leveraged data from the existing literature [4], wherein authors constructed a sophisticated metabolic pathway in E. coli consisting of four distinct proteins and four strategically selected promoter sequences designed to facilitate the biosynthesis of naringenin from L-tyrosine. The authors employed a suite of experimental methodologies to quantitatively assess the yield of naringenin produced by these constructed pathways, investigating various promoter combinations while maintaining constant protein sequences.

To thoroughly evaluate the performance of ProDMM in this metabolic context, we employed the Perplexity-based Zero-shot Sequences Ranking method using the ProDMM-encoder, as delineated in the methods section. This approach permitted the direct scoring of different pathway sequences comprising eight components formed by the four NCDS coupled with their corresponding protein sequences. We subsequently assessed the scores in relation to the yield of the final product, naringenin. As depicted

in Figure 2F, the zero-shot predictive performance of the pre-trained ProDMM outperforms alternative nucleotide models (two-sided t-distributed $p < 0.01$), including Evo and DNABERT2. This remarkable distinction underscores the unique capabilities of ProDMM.

During its pre-training phase, ProDMM acquired knowledge regarding the interactions among individual protein sequences (or coding sequences, CDS) alongside their associated upstream and downstream non-coding sequences. However, this phase did not directly address long-range interactions among various protein components that are distributed throughout metabolic pathways. In contrast, other nucleic acid models such as Evo [8] and DNABERT2 [10] have leveraged information from the entire genome, thereby capturing longer-range interactions among CDS sequences to a greater extent. To further enhance the model's predictive capabilities further, we meticulously compiled and organized genomic and protein data from the antiSMASH database version 4 [28], encompassing 36,554 species. This dataset was structured into sequences formatted as NCDS-protein-NCDS-protein, containing four components per sequence. Following this, we fine-tuned ProDMM's encoder on this structured data using a masked language model (MLM) pretraining approach, enabling ProDMM to learn the interactions between adjacent proteins and their corresponding non-coding sequences. When we assessed the zero-shot predictive performance of the fine-tuned model on the aforementioned metabolic pathway dataset, we observed a significant enhancement in predictive accuracy. This finding underscores the effectiveness of explicitly training ProDMM to comprehend the interactions between neighboring proteins and their non-coding sequences, thereby amplifying its zero-shot predictive capabilities for more complex and extended metabolic pathways.

We anticipate that extending ProDMM's architecture to learn longer-range interactions between protein sequences will further improve its zero-shot predictive abilities concerning metabolic pathways. By providing invaluable insights into the complexities associated with metabolic pathway optimization, ProDMM establishes a robust framework for future research endeavors in this critical area. Ultimately, the work presented here not only advances our understanding of genetic interactions but also opens new avenues for innovative biotechnological applications and sustainable bioproduction processes. This contribution significantly enhances the broader fields of synthetic biology and metabolic engineering, promoting ongoing exploration and growth in these transformative disciplines.

### 2.2.2 Supervised Learning Tasks

We further evaluated the performance of ProDMM-encoder across a diverse array of supervised learning tasks, specifically focusing on its ability to predict melting temperatures for various native proteins (Figure 3A), forecast protein solubility (Figure 3B), estimate protein expression levels (Figure 3C), and assess the transcriptional abundance of proteins in Escherichia coli and Halobacterium volcanii (Figure 3D). To ensure consistency with prior studies, we utilized the Pearson correlation coefficient as the evaluation metric for these supervised learning tasks [15]. In each of these tasks, ProDMM consistently demonstrated superior performance when utilizing coding sequences (CDS) as input compared to utilizing protein sequences. Furthermore, for

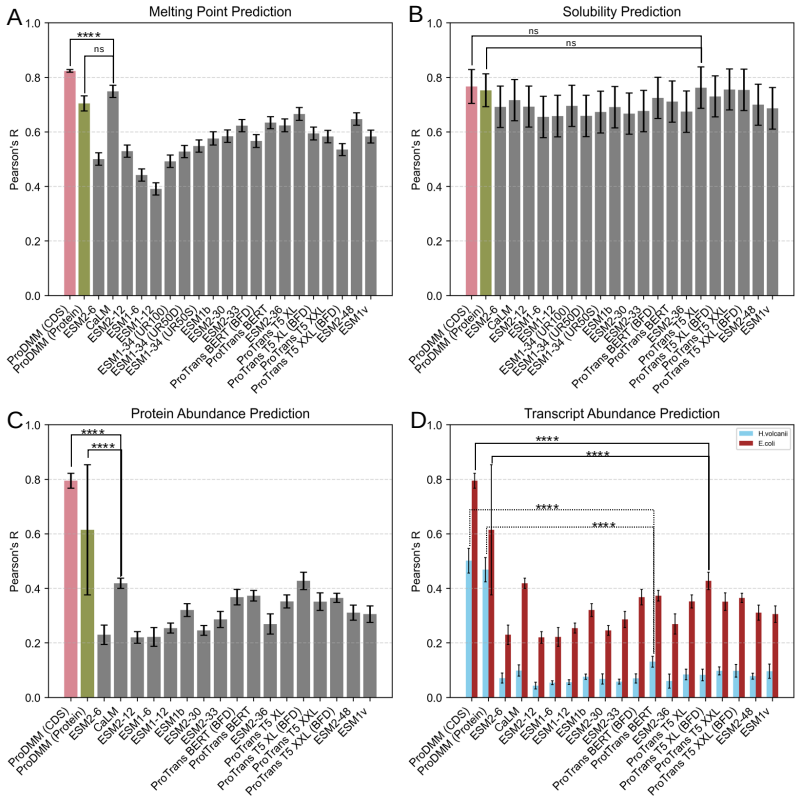


**Figure 3:** Evaluation of ProDMM-Encoder in protein property prediction tasks. This figure assesses the performance of the ProDMM model across four protein property prediction tasks. (A)-(D) show the results for melting point prediction, solubility prediction, protein abundance prediction, and transcript abundance prediction, respectively. P-values are reported between the top three performing models; "****" denotes $p < 0.0001$. Performance is measured using the Pearson correlation coefficient, reflecting the correlation between predicted and actual values. The data are presented as mean values, with error bars representing the standard deviation calculated from a fivefold cross-validation. The results from the ESM and ProtTrans series of models were obtained from the literature [15].

most of the tasks, ProDMM outperformed several established protein and nucleotide baseline models, including CaLM [15], ESM [5, 6, 22], and ProtTrans [14], underscoring its efficacy in handling these challenges. Comprehensive data can be found in Table S1 of the supplementary information.

Additionally, we visualized the embeddings generated by ProDMM for the top ten organisms with the highest volume of sequence data, utilizing both CDS sequences and protein sequences as input, as depicted in Figure S3. The visual representations derived from these figures provide compelling evidence that embeddings obtained from CDS sequences exhibit sharper and more distinct boundaries across various species

compared to those generated from protein sequences. This observation underscores the model's capability to delineate subtle differences in sequence information when CDS sequences are employed, suggesting that they encapsulate richer and more nuanced biological insights.

Collectively, these findings underscore the remarkable capabilities of the ProDMM model, particularly when leveraging CDS sequences. This enhanced performance across the aforementioned protein-related supervised learning tasks indicates that ProDMM is adept at capturing a comprehensive spectrum of biological information inherent in the sequences, thereby improving predictive accuracy and reliability. The ability to integrate such diverse data effectively may pave the way for further advancements in protein research and synthetic biology applications, providing a robust foundation for future explorations in the field. By harnessing the intricate details within CDS sequences, ProDMM not only elevates the quality of predictions but also contributes significantly to our understanding of protein behavior and function across different biological contexts.

## 2.3 Benchmark Results of the Generation

To evaluate the performance of the ProDMM Seq2Seq, we built up and conducted a series of benchmarks across multiple downstream tasks, including prokaryotic and eukaryotic organisms reverse translation of proteins, as well as various promoter-cds co-generation tasks (Figure 4). Detailed methodologies can be found in the Methods section.

### 2.3.1 Reverse Translation Generation

ProDMM's sequence-to-sequence (seq2seq) architecture intricately investigates the inherent relationships among proteins, CDS, and NCDS, thereby enhancing its cross-modal generative capabilities. In Figure 4, we validate the efficacy of this cross-modal generation by focusing on the task of generating codon sequences that correspond accurately to given protein sequences. This involves the reverse translation of proteins while implementing codon mask constraints. These constraints are crucial, as they ensure that the generated codons precisely match the specific amino acids, adhering to the standard codon table. This methodological approach highlights the model's proficiency in executing complex sequence transformations while maintaining strict biological fidelity.

We adopt the dataset from Transducer [29], where the task is to generate high-expression CDS given a protein sequence. As shown in Figure 4A, on the high-expression data from E. coli, ProDMM outperforms baseline models based on unigram, trigram, and 5-gram frequencies, as well as their ensemble (1,3,5-gram) [30], and significantly surpasses the state-of-the-art Prot-CDS Transducer model. Additionally, Figure S4 demonstrates that ProDMM accurately produces the correct codon usage distribution for E. coli. In Figure 4B, we conducted ablation studies on the E. coli data, including the incorporation of codon mask constraints and a pretrain-finetune approach. Due to the scarcity of high-expression gene data in E. coli, we hypothesized that pretraining on a larger set of non-high-expression gene data from E.

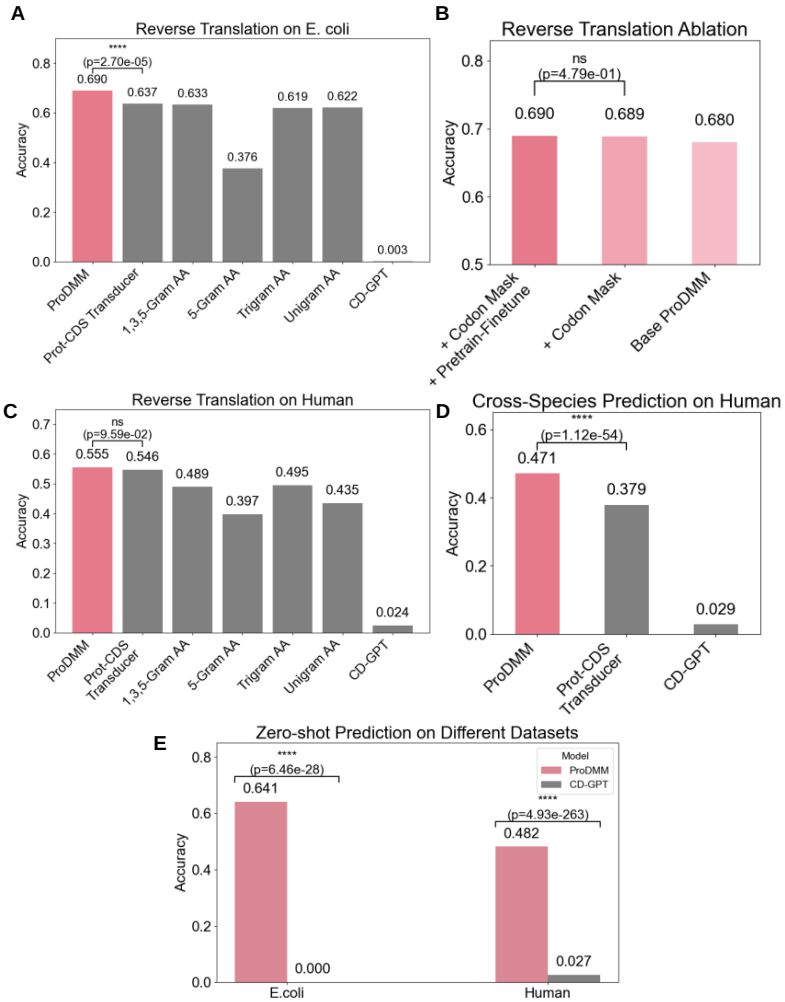


**Figure 4:** Evaluation of ProDMM on Reverse Translation Tasks. This figure assesses the performance of the ProDMM model across various reverse translation tasks. (A) displays the results for the reverse translation of high-expression CDS based on a protein sequence from E. coli. (B) shows the impact of ablation studies on codon mask constraints and the pretrain-finetune approach for reverse translation tasks. (C) illustrates the reverse translation of high-expression CDS based on a protein sequence from humans. (D) demonstrates cross-species prediction, where the model is fine-tuned on the E. coli dataset and tested on the human dataset. (E) evaluates the zero-shot prediction capabilities of reverse translation without any fine-tuning. Performance is measured using accuracy, reflecting the correlation between the generated codon sequences and the actual codon sequences.

coli, followed by finetuning on high-expression genes, would be beneficial—forming a pretrain-finetune pipeline. Experimental results confirmed that adding amino acid-codon mask constraints and employing the pretrain-finetune strategy both contribute positively to the generation performance.

Interestingly, as shown in Figure 4C, on high-expression human data, ProDMM's generated results also outperform those of the baseline models. Despite being pre-trained on a large amount of prokaryotic data, ProDMM can effectively transfer to eukaryotic data without experiencing model collapse, demonstrating that extensive pre-training has endowed the model with species adaptability. Furthermore, we conducted experiments without finetuning on human data to explore the capability of cross-species high-expression generation. As shown in Figure 4D, we directly generalized ProDMM and Prot-CDS Transducer—both finetuned on high-expression E. coli data—to the generation of high-expression human CDS. We found that ProDMM's cross-species generalization ability significantly surpasses that of Transducer. Similarly, as shown in Figure 4E, when we do not perform any finetuning on high-expression data and use only the pre-trained model for zero-shot prediction, ProDMM already outperforms the baseline biological model CD-GPT [31] on both E. coli and human data. This reflects the effectiveness of large-scale cross-modal pre-training in understanding and generating the intrinsic relationships between proteins and CDS.

### 2.3.2 Regulator-CDS Co-evolving Effect

Some studies [32, 33] have indicated that there exists a co-evolving interaction between gene regulatory elements and CDS, jointly influencing gene expression. Using the E. coli data from DeepExpression [33] , we trained a model called ProDMM-Expression. This model utilizes the ProDMM encoder to predict gene expression from DNA sequences, encompassing both coding and non-coding regions. In our approach, we first conditioned on the CDS to generate specific promoters, which were then concatenated with the CDS and fed into ProDMM-Expression. For control experiments, we separately used only the Ground Truth (GT) CDS or only the promoter as input. As depicted in Figure 5A, the tailored promoters, specifically designed to complement the CDS, when combined with the CDS, significantly enhance gene expression prediction. This method achieves superior results compared to using the CDS or the promoter alone, underscoring the synergistic effect of customized promoter-CDS combinations in predicting gene expression.

Given the significant impact of promoter specificity on protein expression [34], we have conducted an in-depth investigation into generating optimal promoters based on CDS information. To achieve this, we curated a dataset of promoter-CDS pairs of E. coli for model fine-tuning [35]. As illustrated in Figure 5B, ProDMM significantly outperforms existing methods, including the DNA-based approach Evo [8] and the baseline method CDS-Prom Transducer. Despite the availability of numerous methods for unconditional promoter generation, there is a conspicuous gap in research focusing on generating specific promoters conditioned on CDS. To emphasize the importance of conditioned promoter generation, we conducted a series of comparative experiments. These studies juxtaposed conditionally generated promoters combined

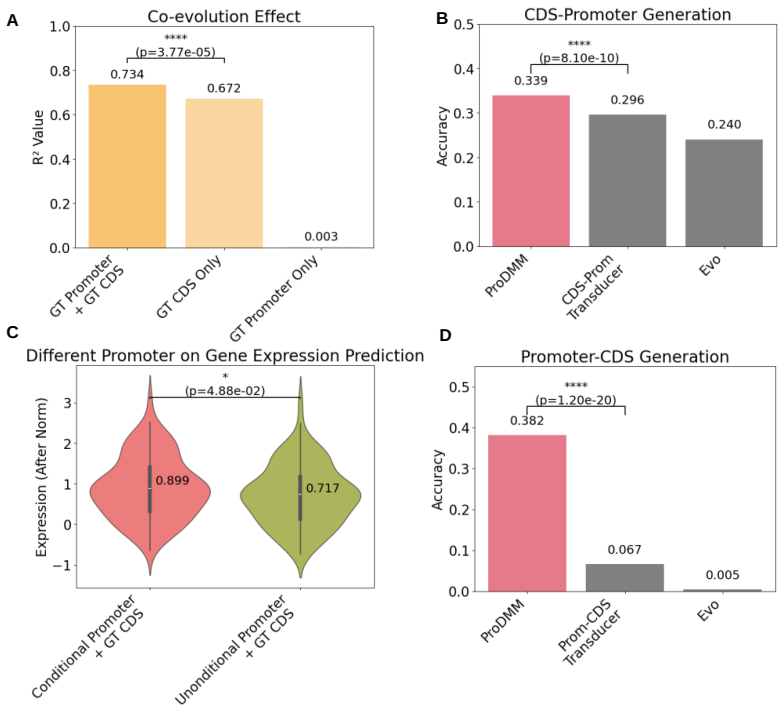


**Figure 5:** Evaluation of ProDMM on Co-evolution Effect Tasks. This figure assesses the performance of the ProDMM model across reverse translation tasks. (A)-(D) show the results for the co-evolution effect of the CDS and the promoter, CDS to promoter generation, comparison between conditional and unconditional promoter for gene expression, and promoter to CDS generation, respectively.

with CDS sequences against unconditionally generated ones, leveraging ProDMM-Expression as a reference for expression level predictions (Figure 5C). The findings indicate that promoters developed with CDS-specific conditioning yield substantially higher gene expression levels relative to those generated unconditionally. Additionally, Figure S5 illustrates that conditional generation successfully reproduces the distribution of key motifs in promoter sequences, such as the CCAAT box, the -75 region, and the TATAAT box in the -10 region, more effectively than unconditional generation. This not only accentuates the essential need to tailor promoter generation to CDS for improved gene expression but also showcases the pivotal role of ProDMM's advanced cross-modal generative capabilities in propelling this research frontier forward.

Building upon our previous co-evolution analysis, which established the specific and intricate relationship between CDS and promoters, we have embarked on an exploration of a novel task: the generation of CDS sequences guided by promoters. The objective is to produce CDS that more effectively harness the functional capabilities of their associated promoters. In this endeavor, we compared our model, ProDMM, with baseline models Evo and Prom-CDS Transducer. As illustrated in Figure 5D,

ProDMM's proficiency in generating promoter-specific CDS significantly surpasses that of the other two baseline approaches. Notably, unlike the reverse translation and CDS-guided promoter generation tasks, the Promoter-Specific CDS Generation task was not part of the model's pre-training regimen. This emphasizes the robustness of ProDMM's cross-modal generation capabilities, which facilitate the exploration of interactions across different modalities. Furthermore, this highlights ProDMM's ability to expand the frontiers of cross-modal generation, showcasing its potential to innovate in the domain of gene regulatory mechanisms.

# 3 Discussions

In this study, we present ProDMM, a cutting-edge multimodal protein-DNA model pre-trained on over 129 million pairs of non-coding DNA sequences (NCDS) and coding sequences (CDS). This innovative architecture supports three distinct input modalities in one entry—protein sequences, CDS, and NCDS—allowing for versatile applications across various biological contexts. At its core, ProDMM consists of a BERT-like encoder complemented by a decoder tailored for numerous generative tasks.

ProDMM has demonstrated extraordinary performance across a diverse array of downstream tasks, including zero-shot predictions for protein sequences, CDS sequences, NCDS sequences, and CDS expression levels. It excels in supervised learning tasks, accurately predicting the melting temperature $T_m$ and solubility of native protein sequences. Additionally, it performs exceptionally well in conditional generation tasks—generating CDS sequences based on protein sequences and promoter sequences based on CDS—far surpassing existing baseline models in both protein and nucleic acid prediction, such as ESM [5, 6, 22] and ProtTrans [14] for proteins, and Evo [8], DNABERT2 [10] for nucleic acids. Furthermore, we discovered that in supervised learning tasks related to protein sequences, utilizing the CDS as input provided superior predictive performance compared to using amino acid sequences. In promoter generation tasks, conditional generation based on CDS sequences yielded substantially higher protein expression levels compared to unconditional approaches. These findings underscore the significant implications for future initiatives in protein engineering and promoter design, highlighting the necessity for focused strategies in these domains. A key strength of ProDMM lies in its architectural design, which effectively captures the intricate interactions between protein sequences and their upstream and downstream NCDS. Notably, ProDMM can predict the characteristics of entire metabolic pathways—specifically, the yield of end products—without any reliance on labeled training data, thus showcasing its impressive zero-shot predictive capabilities. Moreover, supplementary training on data from adjacent protein sequences and their associated NCDS substantially enhances ProDMM's zero-shot predictive performance concerning metabolic pathways. In conclusion, ProDMM represents a transformative advance in the fields of synthetic biology, protein engineering, promoter design, codon optimization, and metabolic pathway optimization. Its versatility and predictive power open avenues for innovative strategies and improved outcomes in metabolic engineering and biotechnological applications, heralding a new era of advancements in these vital scientific disciplines.

## Funding

This work was supported by the grants from the National Natural Science Foundation of China (grant number 12104295), the Computational Biology Key Program of Shanghai Science and Technology Commission (23JS1400600), Shanghai Jiao Tong University Scientific and Technological Innovation Funds (21X010200843), and Science and Technology Innovation Key R&D Program of Chongqing (CSTB2022TIAD-STX0017), the Student Innovation Center at Shanghai Jiao Tong University, and Shanghai Artificial Intelligence Laboratory.

## Competing interests

The authors declare no competing interests.

## Ethics approval and consent to participate

Not applicable.

## Consent for publication

Not applicable.

## Code availability

The source code and pre-training checkpoint file will be made publicly available after the paper is published.

## Author contribution

P.T., L.H. and N.D. conceptualized and supervised this research project. P.T., M.L., Y.R. and P.Y. developed the architecture of ProDMM. P.T. collected and processed the pre-training dataset. M.L., Y.R., P.Y. and J.C. wrote the code and conducted the pre-trianing of the model. P.T. designed the benchmarks. M.L., Y.R., P.Y., J.C., N.D.,

X.M., Y.C., B.Z., W.O. and B.W. finished the benchmarks. M.L., Y.R., P.Y. and J.C. contributed equally to this work. All authors reviewed and accepted the manuscript.

# Appendix A Method

## A.1 Pre-training Dataset Processing

The pre-training dataset was sourced from the GTDB database, release number 220. To reduce the data volume, we only utilized sequences from representative genomes. We then employed the Prodigal [36] tool to predict the CDS sequences from each nucleotide FASTA file and obtained the corresponding translated protein sequences. During this process, we also recorded the non-coding sequences (NCDS) located between adjacent CDS sequences within the same FASTA file. This approach allowed us to retrieve each CDS sequence along with its preceding and post-NCDS sequences, as well as the translated protein sequences. During the processing, we discarded any non-full-length protein sequences and protein sequences exceeding 2000 amino acids in length. Additionally, protein sequences that were not translated using the standard codon table or contained non-standard amino acids were excluded. Following these filtering steps, we ultimately retained approximately 129 million paired sequences.

## A.2 ProDMM-Encoder

ProDMM-Encoder includes a universal tokenizer that converts protein sequences, coding sequences (CDS), and non-coding sequences (NCDS) into one-hot encodings. It features a Transformer encoder that is pre-trained using a masked language modeling (MLM) task to derive sequence latent representations. Additionally, it has a prediction head that generates token distributions from these latent representations, enabling ProDMM-Encoder to perform zero-shot sequence scoring tasks.

#### *Universal tokenizer*

The universal tokenizer splits a molecule sequence (Protein, CDS, or NCDS) into a small tokens list. It then maps each token to a unique integer index using a vocabulary table, followed by transforming these indices into one-hot vectors with all elements set to 0 except for a 1 at the position corresponding to the index of the token. In protein sequences and NCDS, each character is a token while every three characters form a token in the CDS. The vocabulary consists of 97 tokens: 64 codons, 20 natural amino acids, 4 nucleotide bases (A, T, C, G), and 9 special tokens. To distinguish between the abbreviations of amino acids and nucleotide bases in the program implementation, we use uppercase characters to represent amino acids and lowercase characters to represent nucleotide bases. For instance, uppercase '**A**' represents *alanine*, while lowercase '**a**' represents *adenine*; uppercase '**T**' stands for *threonine*, and lowercase '**t**' denotes *thymine*. The tokens, indices, and names in the vocabulary are listed in Table A1.

#### *Model Architecture*

The ProDMM is based on the Transformer architecture [18], but we have made several modifications that have been proven effective in practice. Specifically, we replaced

| Token | Index | Name | Token | Index | Name |
|---|---|---|---|---|---|
| <pad> | 0 | Padding Token | tag | 48 | Stop Codon TAG |
| <cls> | 1 | Classification Token | tta | 49 | Codon TTA |
| <sep> | 2 | Separator Token | ttt | 50 | Codon TTT |
| <unk> | 3 | Unknown Token | ttc | 51 | Codon TTC |
| <bos> | 4 | Beginning of Sequence | ttg | 52 | Codon TTG |
| <eos> | 5 | End of Sequence | tca | 53 | Codon TCA |
| <mask> | 6 | Mask Token | tct | 54 | Codon TCT |
| - | 7 | Gap | tcc | 55 | Codon TCC |
| * | 8 | Stop | tcg | 56 | Codon TCG |
| L | 9 | Leucine | tga | 57 | Stop Codon TGA |
| A | 10 | Alanine | tgt | 58 | Codon TGT |
| G | 11 | Glycine | tgc | 59 | Codon TGC |
| V | 12 | Valine | tgg | 60 | Codon TGG |
| S | 13 | Serine | caa | 61 | Codon CAA |
| E | 14 | Glutamic Acid | cat | 62 | Codon CAT |
| R | 15 | Arginine | cac | 63 | Codon CAC |
| T | 16 | Threonine | cag | 64 | Codon CAG |
| I | 17 | Isoleucine | cta | 65 | Codon CTA |
| D | 18 | Aspartic Acid | ctt | 66 | Codon CTT |
| P | 19 | Proline | ctc | 67 | Codon CTC |
| K | 20 | Lysine | ctg | 68 | Codon CTG |
| Q | 21 | Glutamine | cca | 69 | Codon CCA |
| N | 22 | Asparagine | cct | 70 | Codon CCT |
| F | 23 | Phenylalanine | ccc | 71 | Codon CCC |
| Y | 24 | Tyrosine | ccg | 72 | Codon CCG |
| M | 25 | Methionine | cga | 73 | Codon CGA |
| H | 26 | Histidine | cgt | 74 | Codon CGT |
| W | 27 | Tryptophan | cgc | 75 | Codon CGC |
| C | 28 | Cysteine | cgg | 76 | Codon CGG |
| aaa | 29 | Codon AAA | gaa | 77 | Codon GAA |
| aat | 30 | Codon AAT | gat | 78 | Codon GAT |
| aac | 31 | Codon AAC | gac | 79 | Codon GAC |
| aag | 32 | Codon AAG | gag | 80 | Codon GAG |
| ata | 33 | Codon ATA | gta | 81 | Codon GTA |
| att | 34 | Codon ATT | gtt | 82 | Codon GTT |
| atc | 35 | Codon ATC | gtc | 83 | Codon GTC |
| atg | 36 | Codon ATG | gtg | 84 | Codon GTG |
| aca | 37 | Codon ACA | gca | 85 | Codon GCA |
| act | 38 | Codon ACT | gct | 86 | Codon GCT |
| acc | 39 | Codon ACC | gcc | 87 | Codon GCC |
| acg | 40 | Codon ACG | gcg | 88 | Codon GCG |
| aga | 41 | Codon AGA | gga | 89 | Codon GGA |
| agt | 42 | Codon AGT | ggt | 90 | Codon GGT |
| agc | 43 | Codon AGC | ggc | 91 | Codon GGC |
| agg | 44 | Codon AGG | ggg | 92 | Codon GGG |
| taa | 45 | Stop Codon TAA | a | 93 | Adenine |
| tat | 46 | Codon TAT | c | 94 | Cytosine |
| tac | 47 | Codon TAC | g | 95 | Guanine |
| - | - | - | t | 96 | Thymine |

**Table A1**: Vocabulary table of tokens, indices, and names

absolute position encoding with rotary position encoding [37], replaced the ReLU activation function with the GeLU activation function [38], replaced Pre-LayerNorm with Post-LayerNorm [39], optimized the standard attention operation using flash attention [40], added a layer normalization layer after the embedding [19], and used embedding shrinking [41] technique to prevent loss spike.

Formally, given a molecular sequence, it is tokenized into a sequence of tokens $(t_1, t_2, \ldots, t_N)$, where $N$ denotes the sequence length. Each token $t_2$ is mapped to a unique integer index $x_i$ according to the vocabulary. These indices are then transformed into one-hot vectors $\mathbf{e}_i \in \mathbb{R}^V$, where $V$ is the size of the vocabulary. In each one-hot vector $\mathbf{e}_i$, all elements are zero except for a single one at the position corresponding to the index $x_i$.

The one-hot vectors are projected into a continuous embedding space using an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$, where $d$ is the dimensionality of the embedding space:

$$\mathbf{h}_i^{(0)} = \mathbf{E}^\top \mathbf{e}_i, i = 1, 2, \ldots, N \tag{A1}$$

These embeddings are fed into a stack of $L$ Transformer encoder layers. At each layer $l$ $1 \leq l \leq L$), the hidden representations are updated using multi-head self-attention mechanisms followed by position-wise feed-forward networks. The computation within each Transformer layer can be summarized as:

$$\mathbf{H}^{(l)} = \text{TransformerLayer}\big(\mathbf{H}^{(l-1)}\big) \tag{A2}$$

, where $\mathbf{H}^{(l-1)} = [\mathbf{h}_1^{(l-1)}, \mathbf{h}_2^{(l-1)}, \ldots, \mathbf{h}_N^{(l-1)}]$ represents the matrix of hidden states from the previous layer.

After passing through all $L$ layers, we obtain the final contextualized representations $\mathbf{H}^{(L)}$. For the masked language modeling task, these representations are projected back to the vocabulary space to compute the logits for each token position:

$$\mathbf{z}_i = \mathbf{W_2}(LayerNorm(\mathbf{W_1}\mathbf{h}_i^{(L)})) + \mathbf{b_2} \tag{A3}$$

, where $\mathbf{W_1} \in \mathbb{R}^{d \times d}$ is a projection matrix, $\mathbf{W_2} \in \mathbb{R}^{V \times d}$ is the output projection matrix and $\mathbf{b_2} \in \mathbb{R}^V$ is a bias vector.

The predicted probability distribution over the vocabulary at each position is obtained by applying the softmax function to the logits:

$$\hat{\mathbf{y}}_i = \text{softmax}(\mathbf{z}_i) \tag{A4}$$

The model is trained by minimizing the cross-entropy loss between the predicted distributions and the true one-hot encoded targets $\mathbf{y}_i$:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i^\top \log \hat{\mathbf{y}}_i, \tag{A5}$$

where the logarithm is applied element-wise.

***Pre-training Method***

ProDMM-Encoder is pre-trained with masked language modeling (MLM) on the following sequences:

- Protein Sequence. <cls> token and <sep> token are added at the beginning and end of tokenized sequences.
- CDS Sequence. <cls> token and <sep> token are also added at the beginning and end of tokenized sequences.
- NCDS-CDS-NCDS Sequence. <cls> token and <sep> token are added at the beginning and end of tokenized sequences. Additionally, a <sep> token is inserted between NCDS and CDS, i.e., "{<cls>, NCDS tokens, <sep>, CDS tokens, <sep>, NCDS tokens, <sep>}".
- NCDS-Protein-NCDS Sequence. <cls> token and <sep> token are added at the beginning and end of tokenized sequences. Additionally, a <sep> token is inserted between NCDS and Protein, i.e., "{<cls>, NCDS tokens, <sep>, Protein tokens, <sep>, NCDS tokens, <sep>}"

Each <pre-NCDS, CDS, Protein, post-NCDS> quadruple in the dataset transitions to one of the sequences above with equal probability.

For each sequence, 15% of the tokens are randomly selected, of which 80% are replaced with the mask token, 10% are randomly replaced with a random token, and 10% are left unchanged [19]. The model is trained using cross-entropy loss in Equation A4. We opted to use the 15% masking ratio because this ratio has been extensively validated as the most effective in pre-training models for proteins [42], DNA [9, 10, 15], and molecules [43]. We utilized AdamW optimizer [44, 45] to train the model with a learning rate of $1.5 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 10^{-8}$, and a weight decay of 0.01. We warmed up the learning rate gradually during the initial 5,000 gradient update steps, starting from 0 and reaching $1.5 \times 10^{-4}$. Subsequently, we apply a cosine learning rate scheduler to decay the learning rate over the remaining steps. To monitor the training process, 5% of the training set was randomly selected for validation purposes. We trained the model with 24 NVIDIA H100 GPUs with 80GB VRAM memory over 40 days, which involved 260,000 gradient steps. We manually terminated the training as there was no improvement in validation loss for 10,000 training steps. We employed the zero-1 strategy within the DeepSpeed framework [46] to distribute optimizer parameters across GPUs for enhanced computational efficiency. We did not zero-2 and zero-3 due to the additional communication overhead they introduce when gradient accumulation is in use. The training curves for both the encoder training loss and validation loss are presented in Figure S6. The hyper-parameters configuration are listed at Table A2.

## A.3 ProDMM-Seq2Seq

ProDMM-Seq2Seq includes a prompt tokenizer based on the universal tokenizer in ProDMM-Encoder that expands cross-modality generation prompt tokens. It enables sequence-to-sequence (seq2seq) generation using a language modeling (LM) task to encode source sequence latent representations, then decode it to target sequence by using a casual decoder.

| Parameter | Value |
|---|---|
| Dropout | 0.1 |
| Hidden size | 1536 |
| Intermediate size | 6144 |
| LayerNorm epsilon | $1.0 \times 10^{-5}$ |
| Number of attention heads | 64 |
| Number of hidden layers | 48 |
| Training mini-batch size | 4096 |
| Evaluation mini-batch size | 4096 |
| Seed | 42 |
| Training steps | 260,000 |
| Gradient Accumulations | 8 |
| Learning rate | $1.5 \times 10^{-4}$ |
| Adam beta1 | 0.9 |
| Adam beta2 | 0.95 |
| Adam epsilon | 1e-8 |
| Gradient clip value | 1.0 |
| Gradient clip algorithm | norm |
| Precision | BF16 |
| Weight decay | 0.01 |
| Warmup steps | 5000 |
| Strategy | Zero-1 |

**Table A2**: Pre-training hyper-parameters of ProDMM-Encoder.

***Prompt tokenizer***

We expanded the universal vocabulary which originally contains 97 tokens, by adding three generation prompt tokens. These prompt tokens are designed to guide the model to perform cross-modal generation tasks more effectively. The vocabulary consists of 100 tokens: 9 special tokens, 64 codons, 20 natural amino acids, 4 nucleotide bases (A, T, C, G), and 3 generation prompt tokens.

| Token | Index | Name |
|---|---|---|
| <aas2cds> | 97 | Amino Acids Sequence to CDS Token |
| <cds2pre-ncds> | 98 | CDS to Pre-NCDS Token |
| <cds2post-ncds> | 99 | CDS to Post-NCDS Token |

**Table A3**: Seq2Seq prompt vocabulary table of tokens, indices, and names based on the universal vocabulary of the encoder.

***Pre-training Method***

ProDMM-Seq2Seq is pre-trained with language modeling (LM) on the following sequences:

- Protein Sequence - CDS. **Source Sequence:** Protein Sequence - <cls> token and <sep> token are added at the beginning and end of tokenized sequences. Additionally, a <aas2cds> token is inserted between Protein Sequence and <sep>, i.e.,

"{<cls>, Protein tokens, <aas2cds>, <sep>}". **Target Sequence:** CDS - <bos> token and <eos> token are added at the beginning and end of tokenized sequences, i.e., "{<bos>, CDS tokens, <eos>}".

- CDS - Pre-NCDS. **Source Sequence:** CDS - <cls> token and <sep> token are added at the beginning and end of tokenized sequences. Additionally, a <cds2pre-ncds> token is inserted between Protein Sequence and <sep>, i.e., "{<cls>, CDS tokens, <cds2pre-ncds>, <sep>}". **Target Sequence:** Pre-NCDS - <bos> token and <eos> token are added at the beginning and end of tokenized sequences, i.e., "{<bos>, Pre-NCDS tokens, <eos>}".
- CDS - Post-NCDS. **Source Sequence:** CDS - <cls> token and <sep> token are added at the beginning and end of tokenized sequences. Additionally, a <cds2post-ncds> token is inserted between Protein Sequence and <sep>, i.e., "{<cls>, CDS tokens, <cds2post-ncds>, <sep>}". **Target Sequence:** Post-NCDS - <bos> token and <eos> token are added at the beginning and end of tokenized sequences, i.e., "{<bos>, Post-NCDS tokens, <eos>}".

We utilized AdamW optimizer [44, 45] to train the model with a learning rate of $4 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-7}$, and a weight decay of 0.01. We warmed-up the learning rate gradually during the initial 5,000 gradient update steps, starting from 0 and reaching $1.5 \times 10^{-4}$. Subsequently, we apply a cosine learning rate scheduler to decay the learning rate over the remaining steps. To monitor the training process, 5% of the training set was randomly selected for validation purposes. We trained the model with 32 NVIDIA A100 GPUs with 80GB VRAM memory over 14 days, which involved 100,000 gradient steps. We manually terminate the training as there was no improvement in validation loss for 5,000 training steps. We employed DDP strategy for enhanced computational efficiency. The training curves for both the ProDMM-seq2seq training loss and validation loss are presented in Figure S7. The hyper-parameters configuration are listed in Table A4.

## A.4 Zero-shot Prediction Evaluations of ProDMM-Encoder

***Perplexity-based Zero-shot Sequences Ranking***

Perplexity is a function that maps a sequence to a real-coded value, reflecting the likelihood of the sequence occurring [47]. In the context of Masked Language Models (MLMs) such as BERT [19], the perplexity cannot be directly computed as in traditional autoregressive models [5], since MLMs are bidirectional and designed to predict masked tokens. Thus, we employ the *pseudo-perplexity*, can be calculated for MLMs by iteratively masking and scoring tokens in a given sequence. Specifically, given a sequence $\mathbf{S} = \{w_1, w_2, \ldots, w_n\}$, the pseudo-perplexity is computed by masking each token $w_i$ in turn and computing the log-probability assigned by the MLM to the true token $w_i$ given the context $\mathbf{S}_{\setminus i}$, where $\mathbf{S}_{\setminus i}$ denotes the sequence with the $i$-th token masked. Specifically, the contribution of each token to the pseudo-perplexity is given by:

$$\log P(w_i \mid \mathbf{S}_{\setminus i}) \tag{A6}$$

| Parameter | Value |
|---|---|
| Dropout | 0.1 |
| Hidden size | 1536 |
| Intermediate size | 6144 |
| LayerNorm epsilon | $1.0 \times 10^{-5}$ |
| Number of attention heads | 64 |
| Number of hidden layers | 12 |
| Training mini-batch size | 4096 |
| Evaluation mini-batch size | 4096 |
| Seed | 42 |
| Training steps | 100,000 |
| Gradient Accumulations | 8 |
| Learning rate | $4 \times 10^{-4}$ |
| Adam beta1 | 0.9 |
| Adam beta2 | 0.999 |
| Adam epsilon | 1e-7 |
| Gradient clip value | 1.0 |
| Gradient clip algorithm | norm |
| Precision | BF16 |
| Weight decay | 0.01 |
| Warmup steps | 5000 |

**Table A4**: Pre-training hyper-parameters of ProDMM-Seq2Seq.

where $P(w_i \mid \mathbf{S}_{\setminus i})$ is the probability of token $w_i$ being predicted correctly at the masked position. The overall pseudo-perplexity for the entire sequence $\mathbf{S}$ is then defined as:

$$\mathrm{PP}(\mathbf{S}) = \exp\left(-\frac{1}{n}\sum_{i=1}^{n}\log P(w_i \mid \mathbf{S}_{\setminus i})\right). \tag{A7}$$

The lower the pseudo-perplexity, the more likely the sequence is under the MLM's learned distribution. We can utilize the pseudo-perplexity to rank a set of sequences $\mathcal{S} = \{\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_k\}$ by computing the pseudo-perplexity for each sequence $\mathbf{S}_j$ and sorting them in ascending order or descending order, depending on the task context.

***Perplexity-based Zero-shot Mutants Ranking***

"Zero-shot" denotes that the model is able to generate prediction results without requiring additional task-specific training or labeled data. Previous studies have demonstrated that when trained on extensive and varied protein sequence databases, protein language models can predict experimental measurements of protein mutant fitness without further supervision [5, 11, 22]. Because ProDMM is trained on both protein and DNA sequences, it can predict the effects of mutants in protein sequences, as well as in CDS and NCDS.

Formally, we denote a $k$-order mutant of a sequence $\mathbf{S}$ by a set of triplets:

$$\mathbf{M} = \{(p_i, \mathbf{wt}_i, \mathbf{mt}_i) \mid i = 1, 2, \ldots, k\} \tag{A8}$$

, where $1 \leq p_i \leq n$ represents the mutant position, $n$ is the length of the sequence, $\mathbf{mt}_i$ is the mutant token, and $\mathbf{wt}_i$ is the original token. Specifically, when $k$ is 1, the mutant is a single-point mutation. The mutant effect prediction is computed by its log odds ratio:

$$\mathrm{ME}(\mathbf{M}) = \sum_{i=1}^{k} (\log P(\mathbf{S}[p_i] = \mathbf{mt}_i \mid \mathbf{S}) - \log P(\mathbf{S}[p_i] = \mathbf{wt}_i \mid \mathbf{S})) \tag{A9}$$

Given a set of mutants of a sequence $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$, we can use this Equation A9 to rank the mutants based on their predicted effects. Specifically, the mutants with higher $\mathrm{ME}(\mathbf{M}_i)$ scores are considered to have a stronger predicted effect, either beneficial or deleterious, depending on the task context.

### A.4.1 Coding DNA Sequence Expression Prediction

Codon usage bias complexly affects protein expression levels when recombinant proteins are heterologously expressed [48]. Given that ProDMM has been trained on coding DNA sequences, we hypothesize that the expression level of a CDS is negatively correlated with its pseudo-perplexity. This assumption follows that a lower pseudo-perplexity reflects a higher likelihood of the sequence occurring in the learned distribution of the model, indicating that the sequence is more likely to be biologically reasonable and efficient in expression.

#### *Dataset*

To validate this hypothesis, we first gather a dataset in which [20] experimentally evaluated the expression levels of 6 distinct proteins, each expressed using different coding sequences in the *E.coli* and a real-coded value represents their expression level.

#### *Scoring Method*

Given a set of CDS sequences that translated into the same protein, we compute the pseudo-perplexity for each CDS according to Equation A7 and rank them in ascending order of their pseudo-perplexity.

#### *Evaluation Metrics*

We evaluate the ProDMM-Encoder and other models by computing the Spearman ranking correlation coefficient between the model-derived scores and truth expression level. Meanwhile, we utilize a two-tailed t-test to assess whether the performance difference between the ProDMM-Encoder and the previous state-of-the-art model is significant.

### A.4.2 Zero-shot Non-Coding DNA Sequence Mutant Effect Prediction

Non-coding DNA Sequences are stored in the genome and play crucial roles in protein synthesis and gene regulation. Mutant to NCDS may enhance or decrease its function

fitness (Fitness is a metric specific to a study that quantifies how effectively an NCDS performs a particular function). Since ProDMM-Encoder have been trained with Non-coding DNA Sequences, we would like to test its performance on zero-shot NCDS mutant effect prediction.

***Dataset***

We utilize the NCDS mutant effect prediction benchmark from Evo [8]. The benchmark contains 7 NCDS mutant datasets from literature, containing 4 ribozyme mutant datasets [49–52], 2 tRNA mutant datasets [53, 54], and an rRNA mutagenesis dataset [55].

***Scoring Method***

We hypothesize that the mutant effect level of an NCDS is positively correlated with its log odds ratio. We score the mutants according to Equation A9.

***Evaluation Metrics***

We utilize the absolute value of the Spearman ranking correlation coefficient between the model-derived scores and truth mutant fitness. A two-tailed t-test assesses whether the performance difference between the ProDMM-Encoder and the previous state-of-the-art model is significant.

### A.4.3 Zero-shot Protein Coding DNA Sequence Mutant Effect Prediction

Mutations to a protein coding sequence can affect its various function fitness. Pre-training on prokaryote proteins and coding sequences enables ProDMM-Encoder to perform zero-shot predictions on protein coding sequence mutants [5, 8, 22].

***Dataset***

We gather the protein coding sequence mutant effect prediction benchmark from Evo [8] and ProteinGym [11]. The ProteinGym benchmark contains millions of assay-label protein mutants from various taxons. However, it does not supply the protein coding DNA sequence. Evo [8] collects coding DNA sequence and their codon mutations for six proteins from the ProteinGym benchmark [56–61] that tested in the *E. coli* expression system.

***Scoring Method***

The mutant scoring is also based on the hypothesis that the mutant effect level of a protein-coding sequence is positively correlated with its log odds ratio. The mutant score can be scored with two methods with Equation A9. The first method scores the mutants based on the CDS, while the second is based on the protein sequence. To distinguish the two methods, the first method is denoted as "ProDMM-Encoder (CDS)" and the second is denoted as "ProDMM-Encoder (Protein)"

***Evaluation Metrics***

The scoring results are evaluated with the absolute value of the Spearman ranking correlation coefficient between the model-derived scores and truth mutant fitness.

### A.4.4 Zero-shot Protein Sequence Mutant Effect Prediction

Protein language models have demonstrated zero-shot prediction ability on protein mutant effect prediction [5, 22].

***Dataset***

As the pre-training dataset of ProDMM-Encoder only consists of prokaryotic organisms, we only evaluate it on the proteins from prokaryotes. We assess the ProDMM-Encoder using a subset of the ProteinGym benchmark [11] that contains 50 deep mutational scanning (DMS) datasets of distinct proteins from prokaryotes. The subset is denoted as "ProteinGym (Pro.)".

***Scoring Method***

We assume that the mutant effect positively correlates with its log odds ratio. The mutants are scored with Equation A9, where the mutant token is the mutant amino acid, the original token is the wild-type amino acid, and the sequence $\mathbf{S}$ is its wild-type sequences.

***Evaluation Metrics***

We consider each DMS dataset as an individual prediction dataset, where we score each mutant and compute the Spearman rank correlation between the predicted scores and the experimental fitnesses.

### A.4.5 Zero-shot Promoter Ranking

Promoters, as non-coding DNA sequences, play a crucial role in gene expression and protein synthetic biology. They are key elements that regulate the initiation of transcription, thereby influencing the levels of gene expression. The interaction between non-coding DNA sequences and downstream coding sequences is a product of co-evolutionary processes that influence both DNA transcription and protein translation efficiencies [17]. This interplay underscores the importance of evaluating the strength of a promoter in conjunction with the downstream protein it regulates. ProDMM is pre-trained on the NCDS-CDS-NCDS and NCDS-protein-NCDS sequence; we can compute the pseudo-perplexity of the promoter as a proxy of expression level.

***Dataset***

We obtained the dataset from [62], which includes various combinations of different promoters and enhanced green fluorescent protein (EGFP, NCBI ID: AAB02572.1) along with their corresponding expression levels.

***Scoring Method***

We assume the conditioned pseudo-perplexity of the promoter is positively correlated with the expression level. The conditioned pseudo-perplexity is:

$$\mathrm{PP}(S) = \exp\left(-\frac{1}{L_1}\sum_{i=1}^{L_1}\log P(w_i \mid \mathbf{S}_{\setminus i})\right). \tag{A10}$$

, where $\mathbf{S}$ is the concatenation of the NCDS sequence and CDS (protein) sequence. $\mathbf{S} = \{q_1, q_2, \ldots, q_{L_1}, c_1, c_2, \ldots, c_{L_2}\}$, in which $q_i$ is promoter (NCDS) token and $c_i$ is CDS (protein) toke as well as $L_1$ and $L_2$ are the length of promoter and CDS (protein) respectively.

***Evaluation Metrics***

We utilize Equation A10 to score each combination sequence of promoter and CDS and we use the Spearman rank correlation between the score predicted by the model and the ground truth expression level.

## A.5 Supervised Fine-tuning ProDMM-Encoder on Down-stream Tasks

### A.5.1 Fine-tuning configurations

During the fine-tuning process, CDS or protein sequences are input into the frozen, pre-trained ProDMM-Encoder, and the mean value of each output serves as the proxy for each sequence. These proxies are then fed into task-specific downstream heads. These heads are all composed of two multiple-layer perceptions (MLPs), with two dropout layers and one GELU activation function, but their parameters vary based on the specific supervised signals used for fine-tuning. All the results are calculated from fivefold cross-validation.

### A.5.2 Tasks

The ProDMM-Encoder is assessed across four downstream tasks with inputs comprising either protein sequences or CDS sequences. These tasks include predicting the melting point, solubility, protein abundance, and transcript abundance of various native protein sequences. For the transcript abundance predictions, the model's performance is evaluated on two species: Escherichia coli (E. coli) and Haloferax volcanii (H. volcanii).

### A.5.3 Dataset

All datasets used in the fine-tuning process are downloaded from the https://github.com/oxpig/CaLM. Five-fold validation is performed using the "KFold" function from the scikit-learn package.

### A.5.4 Evaluation Metric

Pearson's correlation coefficient is used as the evaluation metric, which is consistent with CaLM [15]. Pearson's R is calculated as follows:

$$Pearson's\ \ R = \frac{\sum(x-\mu_x)(\hat{x}-\mu_{\hat{x}})}{\sqrt{\sum(x-\mu_x)^2\sum(\hat{x}-\mu_{\hat{x}})^2}}, \tag{A11}$$

where x and $\hat{x}$ represent truth values and predicted values, with $\mu_x$ and $\mu_{\hat{x}}$ being their respective means. The calculation is performed using the "pearsonr" function from the SciPy library.

### A.5.5 Baselines

The fine-tuned ProDMM-Encoder is benchmarked against 17 models, categorized into two types. The first type are the models that utilize only protein sequences: ESM2-6 [5], ESM2-12, ESM1-6 [63], ESM1-12, ESM1b [6], ESM2-30, ESM2-33, ProTrans BERT (BFD) [14], ProtTrans BERT, ESM2-36, ProTrans T5 XL, ProTrans T5 XL (BFD), ProTrans T5 XXL, ProTrans T5 XXL (BFD), ESM2-48, and ESM1v [22]. The second type is CaLM [15], which received only coding sequences. To ensure a fair comparison, the ProDMM-Encoder is tested on two different input types: protein sequences and CDS, labeled ProDMM_pro and ProDMM_cds, respectively.

## A.6 ProDMM-Seq2seq

### A.6.1 Protein Reverse Translation

***Task***

During the fine-tuning phase, we froze the encoder of ProDMM and trained only the decoder. The input sequence to the encoder consisted of the protein sequence appended with the <aas2cds> prompt token, and the decoder outputed the corresponding CDS.

***Dataset***

We obtained the dataset from [29], which includes various E. coli and human high-expression gene data including protein sequence and CDS. Due to the limited availability of high-expression E. coli data, we also collected non-high-expression gene data to assist in pre-training and fine-tuning our analysis (Figure S4). After applying the Amino Acid–Codon Mask, we observed that even without utilizing the pre-training and fine-tuning pipeline, the performance remained exceptional.

***Evaluation Metrics***

To evaluate the performance of the trained models, we compared both accuracy and perplexity (PPL). Accuracy was measured by generating a reverse-translated DNA sequence from a given protein and then assessing how many codons were predicted correctly. A lower perplexity in our transducer model indicates reduced uncertainty over

the distribution of codons. For neural models, perplexity was calculated by exponentiating the average cross-entropy loss, while for the frequency-based baseline models, it was computed by exponentiating the average negative log-likelihood.

***Baseline***

On E. coli and human benchmarks, we compared the fine-tuned ProDMM with frequency-based methods, including unigram, trigram, and 5-gram models, as well as their ensemble (1,3,5-gram) [30], CD-GPT [31], and the state-of-the-art method Transducer [29]. For the zero-shot setting without fine-tuning, ProDMM was compared with CD-GPT, which was also pre-trained with reverse translation capabilities.

### A.6.2 Co-evolving Effect

***Task***

During the sequence-to-expression fine-tuning phase, we utilized the Encoder of ProDMM augmented with a linear layer as the head. During training, we froze the Encoder and trained only the head. The input sequence is represented in the form 'N'.

***Dataset***

We collected coding sequences (CDS) of *E. coli* and their adjacent non-coding regions, along with the corresponding expression levels, from DeepExpression and Ensemble databases. The input sequence comprises the CDS and 300 bp upstream and downstream non-coding regions, which include promoters, 5' UTRs, 3' UTRs, and terminators.

***Evaluation Metric***

The evaluation metric is the coefficient of determination ($R^2$), calculated using the following formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{A12}$$

In this formula, $y_i$ represents the true expression value, $\hat{y}_i$ is the predicted expression value, $\bar{y}$ denotes the mean of the true expression values, and $n$ is the number of samples.

### A.6.3 CDS-guided Promoter Generation

***Task***

During the fine-tuning phase, we froze the encoder of ProDMM and trained only the decoder. The input to the encoder is the CDS sequence appended with the <cds2prencds> prompt token, and the decoder outputs the corresponding promoter sequence.

***Dataset***

We collected CDS and their corresponding promoter data of E. coli from RegulonDB [35] to construct a benchmark for CDS-to-promoter generation.

***Baseline***

As we are the first to study generating specific promoters from CDS, the available baselines are limited. Therefore, we designed the CDS-Pro Transducer based on the Transducer model [29] as a baseline, and also extended Evo [8] for CDS-NCDS cross-modal generation.

### A.6.4 Promoter-specific CDS Generation

***Task***

During the fine-tuning phase, we froze the encoder of ProDMM and trained only the decoder. The input to the encoder is the CDS sequence, and the decoder outputs the corresponding promoter sequence.

***Dataset***

We collected CDS and their corresponding promoter data of E. coli from RegulonDB [35] to construct a benchmark for promoter-to-CDS generation.

***Baseline***

As we are the first to study generating specific CDS from the promoter sequence, the available baselines are limited. Therefore, we designed the Pro-CDS Transducer based on the Transducer model [29] as a baseline, and also extended Evo [8] for NCDS-CDS cross-modal generation.

## References


[1] Benner, S. A. & Sismour, A. M. Synthetic biology. *Nature reviews genetics* **6**, 533–543 (2005).

[2] Culp, E. & Wright, G. D. Bacterial proteases, untapped antimicrobial drug targets. *The Journal of antibiotics* **70**, 366–377 (2017).

[3] Novik, T. S. *et al.* Sweet-tasting natural proteins brazzein and monellin: safe sugar substitutes for the food industry. *Foods* **12**, 4065 (2023).

[4] Deng, H. *et al.* Pathway evolution through a bottlenecking-debottlenecking strategy and machine learning-aided flux balancing. *Advanced Science* **11**, 2306935 (2024).

[5] Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).

[6] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021).

[7] Madani, A. *et al.* Large language models generate functional protein sequences across diverse families. *Nature Biotechnology* **41**, 1099–1106 (2023).

[8] Nguyen, E. *et al.* Sequence modeling and design from molecular to genome scale with evo. *bioRxiv* (2024).

[9] Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics* **37**, 2112–2120 (2021).

[10] *DNABERT-2: Efficient Foundation Model and Benchmark For Multi-Species Genomes*. URL https://openreview.net/forum?id=oMLQB4EZE1.

[11] Notin, P. *et al.* Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems* **36** (2023).

[12] Li, M. *et al.* Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv* (2024). URL https://www.biorxiv.org/content/early/2024/05/17/2024.04.15.589672.

[13] Zhou, Z. *et al.* Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications* **15**, 5566 (2024).

[14] Elnaggar, A. *et al.* Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* **44**, 7112–7127 (2021).

[15] Outeiral, C. & Deane, C. M. Codon language embeddings provide strong signals for use in protein engineering. *Nature Machine Intelligence* **6**, 170–179 (2024).

[16] Zvyagin, M. *et al.* Genslms: Genome-scale language models reveal sars-cov-2 evolutionary dynamics. *The International Journal of High Performance Computing Applications* **37**, 683–705 (2023).

[17] Zrimec, J. *et al.* Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. *Nature communications* **11**, 6141 (2020).

[18] *Attention is all you need*, NIPS'17 (Curran Associates Inc., Red Hook, NY, USA, 2017).

[19] Burstein, J., Doran, C. & Solorio, T. (eds). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (Association for Computational Linguistics, Minneapolis, Minnesota, 2019).

[20] Fu, H. *et al.* Codon optimization with deep learning to enhance protein expression. *Scientific reports* **10**, 17617 (2020).

[21] Chen, J. *et al.* Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300* (2022).

[22] Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems* **34**, 29287–29303 (2021).

[23] Parks, D. H. *et al.* Gtdb: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic acids research* **50**, D785–D794 (2022).

[24] Laine, E., Karami, Y. & Carbone, A. Gemme: a simple and fast global epistatic model predicting mutational effects. *Molecular biology and evolution* **36**, 2604–2619 (2019).

[25] Marquet, C., Schlensok, J., Abakarova, M., Rost, B. & Laine, E. Vespag: Expert-guided protein language models enable accurate and blazingly fast fitness prediction. *bioRxiv* 2024–04 (2024).

[26] Notin, P. *et al.* Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval 16990–17017 (2022).

[27] Zhou, S. *et al.* Obtaining a panel of cascade promoter-5'-utr complexes in escherichia coli. *ACS Synthetic Biology* **6**, 1065–1075 (2017).

[28] Blin, K., Shaw, S., Medema, M. H. & Weber, T. The antismash database version 4: additional genomes and bgcs, new sequence-based searches and more. *Nucleic acids research* **52**, D586–D589 (2024).

[29] Yang, D. K., Goldman, S. L., Weinstein, E. & Marks, D. Generative models for codon prediction and optimization. *Machine Learning in Computational Biology* (2019).

[30] Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C. & Mercer, R. L. Class-based n-gram models of natural language. *Computational linguistics* **18**, 467–480 (1992).

[31] Zhu, X. *et al.* Cd-gpt: A biological foundation model bridging the gap between molecular sequences through central dogma. *bioRxiv* 2024–06 (2024).

[32] Espinar, L., Tamarit, M. À. S., Domingo, J. & Carey, L. B. Promoter architecture determines cotranslational regulation of mrna. *Genome research* **28**, 509–518 (2018).

[33] Zrimec, J. *et al.* Deep learning suggests that gene expression is encoded in all parts of a co-evolving interacting gene regulatory structure. *Nature communications* **11**, 6141 (2020).

[34] Cheng, J., Maier, K. C., Avsec, Ž., Rus, P. & Gagneur, J. Cis-regulatory elements explain most of the mrna stability variation across genes in yeast. *Rna* **23**, 1648–1659 (2017).

[35] Salgado, H. *et al.* Regulondb v12. 0: a comprehensive resource of transcriptional regulation in e. coli k-12. *Nucleic Acids Research* **52**, D255–D264 (2024).

[36] Hyatt, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics* **11**, 1–11 (2010).

[37] Su, J. *et al.* Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).

[38] Hendrycks Dan, G. K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[39] PMLR. *On layer normalization in the transformer architecture.*

[40] Dao, T., Fu, D., Ermon, S., Rudra, A. & Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems* **35**, 16344–16359 (2022).

[41] *GLM-130B: An Open Bilingual Pre-trained Model.*

[42] Cheng, X. *et al.* Training compute-optimal protein language models. *bioRxiv* 2024–06 (2024).

[43] *SMILES-BERT: Large Scale Unsupervised Pre-Training for Molecular Property Prediction*, BCB '19 (Association for Computing Machinery, New York, NY, USA, 2019). URL https://doi.org/10.1145/3307339.3342186.

[44] Kingma, D. P. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[45] Loshchilov, I., Hutter, F. *et al.* Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101* **5** (2017).

[46] *Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters.*

[47] Manning, C. & Schutze, H. *Foundations of statistical natural language processing* (MIT press, 1999).

[48] Angov, E. Codon usage: nature's roadmap to expression and folding of proteins. *Biotechnology journal* **6**, 650–659 (2011).

[49] Kobori, S., Nomura, Y., Miu, A. & Yokobayashi, Y. High-throughput assay and engineering of self-cleaving ribozymes by sequencing. *Nucleic acids research* **43**, e85–e85 (2015).

[50] Andreasson, J. O., Savinov, A., Block, S. M. & Greenleaf, W. J. Comprehensive sequence-to-function mapping of cofactor-dependent rna catalysis in the glms ribozyme. *Nature communications* **11**, 1663 (2020).

[51] Hayden, E. J., Ferrada, E. & Wagner, A. Cryptic genetic variation promotes rapid evolutionary adaptation in an rna enzyme. *Nature* **474**, 92–95 (2011).

[52] Pitt, J. N. & Ferré-D'Amaré, A. R. Rapid construction of empirical rna fitness landscapes. *Science* **330**, 376–379 (2010).

[53] Domingo, J., Diss, G. & Lehner, B. Pairwise and higher-order genetic interactions during the evolution of a trna. *Nature* **558**, 117–121 (2018).

[54] Guy, M. P. *et al.* Identification of the determinants of trna function and susceptibility to rapid trna decay by high-throughput in vivo analysis. *Genes & Development* **28**, 1721–1732 (2014).

[55] Zhang, Z. D., Nayar, M., Ammons, D., Rampersad, J. & Fox, G. E. Rapid in vivo exploration of a 5s rrna neutral network. *Journal of microbiological methods* **76**, 181–187 (2009).

[56] Firnberg, E., Labonte, J. W., Gray, J. J. & Ostermeier, M. A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular biology and evolution* **31**, 1581–1592 (2014).

[57] Jacquier, H. *et al.* Capturing the mutational landscape of the beta-lactamase tem-1. *Proceedings of the National Academy of Sciences* **110**, 13067–13072 (2013).

[58] Adkar, B. V. *et al.* Protein model discrimination using mutational sensitivity derived from deep sequencing. *Structure* **20**, 371–381 (2012).

[59] Tsuboyama, K. *et al.* Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 434–444 (2023).

[60] Giacomelli, A. O. *et al.* Mutational processes shape the landscape of tp53 mutations in human cancer. *Nature genetics* **50**, 1381–1387 (2018).

[61] Weeks, R. & Ostermeier, M. Fitness and functional landscapes of the e. coli rnase iii gene rnc. *Molecular Biology and Evolution* **40**, msad047 (2023).

[62] Zhou, S. *et al.* Obtaining a panel of cascade promoter-5'-utr complexes in escherichia coli. *ACS Synthetic Biology* **6**, 1065–1075 (2017).

[63] Hsu, C. *et al.* Learning inverse folding from millions of predicted structures. *ICML* (2022). URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.