Binary Discriminator Facilitates GPT-based Protein Design

Zishuo Zeng[1*], Rufang Xu[1], Jin Guo[1], Xiaozhou Luo[1,2,3,4*]

1. Synceres Biosciences Co. Ltd., Shenzhen 518100, China
2. Shenzhen Key Laboratory for the Intelligent Microbial Manufacturing of Medicines, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, CN
3. CAS Key Laboratory of Quantitative Engineering Biology, Shenzhen Institute of Synthetic Biology, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, CN
4. Center for Synthetic Biochemistry, Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, CN

* Corresponding author: zengzishuo@synceres.com, xz.luo@siat.ac.cn

Abstract

Generative pre-trained transformers (GPT) models provide powerful tools for de novo protein design (DNPD). GPT-based DNPD involves three procedures: a) finetuning the model with proteins of interest; b) generating sequence candidates with the finetuned model; and c) prioritizing the sequence candidates. Existing prioritization strategies heavily rely on sequence identity, undermining the diversity. Here, we coupled a protein GPT model with a custom discriminator, which enables selecting candidates of low identity to natural sequences while highly likely with desired functions. We applied this framework to creating novel antimicrobial peptides (AMPs) and malate dehydrogenases (MDHs). Experimental verification pinpointed four broad-spectrum AMPs from 24 candidates, and comprehensive computational analyses on the prioritized MDHs candidates provide compelling evidence for the anticipated function. This framework is time- and data-efficient and may therefore considerably expedite the DNPD process.

Key words: GPT, transformers, protein *de novo* design, deep learning, transfer learning, antimicrobial peptide, malate dehydrogenase

Abbreviations used: generative pre-trained transformers (GPT), natural language processing (NLP), generative adversarial network (GAN), language model (LM), recurrent neural networks (RNN), antimicrobial peptide (AMP), malate dehydrogenase (MDH), enzyme commission (EC), minimum inhibitory concentration (MIC).

Introduction

de novo protein design (DNPD), i.e., generating artificial proteins with desired functions, has proven to be a powerful tool in overcoming bottlenecks across various fields, including drug discovery [1-3],

biosensor engineering [4-6], and synthetic biology [7-9]. DNPD is being accelerated by recent advancements in generative large language models (LLMs, including GPT) [10], for examples, ProtGPT2 [11] and ProGen [12]. Similar to the LLMs for natural language processing (NLP) [13], these protein generative models were trained on large amount of “text corpus” (proteomes) and can generate “sentences” (protein sequences) that captures the “language semantics” (physicochemical and thermodynamic properties similar to natural proteins). Besides, just like LLMs for NLP can be finetuned to adapt to a specific task (such as automatic dialogues [14], summarizing text [15], and poetry writing [16]), protein generative models can be finetuned to generate artificial proteins bearing a specific function of interest. Therefore, protein LLM holds great promise in DNPD.

However, evaluating the quality of the generated sequences remains a challenge, as protein sequences are not human-understandable like natural languages. In the work of ProGen [12], evaluating the generated sequences was mainly done by per-token log-likelihood scoring, which computes the per-residue conditional probabilities along the sequence, conceptually akin to perplexity, which measures the model's confidence in the generated sequence. However, limitations of perplexity in the NLP field have been noticed, such as sensitivity to text length and repeated content [17], inability to reflect semantic coherence [18], and disagreements with human judgements [19]. Another common approach for sequence evaluation is through its identity to known natural proteins of interest, which was used in a generative adversarial network (GAN)-based framework, ProteinGAN [20]. In this approach, the generated sequences are prioritized based on their similarity to the known homologs. Both per-token log-likelihood and sequence identity may overlook the promising (likely to possess the desired protein function) but novel (very different from known natural proteins) sequences – the dark matters in the potential proteome universe.

In addition to per-token log-likelihood and sequence identity, a variety of existing protein predictors may be used to prioritizing the generated protein candidates from different perspectives, such as protein structure [21] and enzyme commission (EC) number [22]. Johnson et al. also incorporated multiple sequence-based and structure-based tools (e.g., Rosetta [23], ESM-v1 [24], CARP-640M [25], ProteinMPNN [21], MIF-ST [26], and ESM-IF [27]) for predicting whether the generated enzymes possess

in vitro activity [28]. However, using these existing predictors suffer from multiple limitations. First, none of these predictors directly reports the likelihood of the desired functions. Second, it requires non-trivial efforts in benchmarking and programming environment setup. Third, these predictors may not be applicable to different types of proteins (e.g., EC number prediction only applies to enzymes).

To address the limitations of current prioritization strategies, here we coupled ProtGPT2 with a convolutional neural network (CNN)-based protein discriminator. Positive protein data are used to finetune the protein generator, ProtGPT2; while both positive and negative data are used to train the discriminator, which will be used to prioritize the generated sequence candidates, allowing a one-stop solution to DNPD for different types of proteins without the dependencies on external tools. A pre-trained LLM, ProtTrans, was used for protein feature extraction for the discriminator. Such LLM-based contextualized embedding technique has been proven to outperform traditional protein features such as one-hot encoding [29-31]. The efficacy of the framework was then demonstrated by two distinct types of sequences: AMPs as representatives of short sequences lacking homology, and malate dehydrogenases (MDHs) as representatives of longer sequences with inferable homology. In addition, the prompts length and size of finetuning set have been investigated to fully elucidate the potential of this framework. Altogether, our results position this framework as a powerful and efficient tool to advance DNPD.

Materials and Methods

Data preparation

We downloaded 3,011 antibacterial peptides from SATPdb database [32] (http://crdd.osdd.net/raghava/satpdb/, accessed on 8/24/2022) as positive set. We filtered this peptide set to only include those composed of canonical amino acids and that were less than or equal to 50 residues in length, resulting in 2,879 peptides, which we refer to as the AMP set. For the negative set, we downloaded 13,225 short (less than or equal to 50-residue-long) protein sequences/peptides from UniProt [33] (https://www.uniprot.org/, accessed on 8/16/2022). We removed sequences containing certain keywords (“antimic*”, “antibac*”, “*toxic*”, “defensive”) from the UniProt description. To create

a balanced negative set and to avoid bias in sequence length distribution, we randomly sampled from this UniProt pool strictly following the AMP set's length distribution. For example, if there were 179 AMPs that are 21-residue-long, then we sampled exactly 179 sequences that are 21-residue-long. In total, we collected 2,879 non-AMPs as negative set.

We gathered a positive set of 16,706 MDH sequences from the training set of Repecka et al.'s study [20]. For negative set, we searched "enzyme" on UniProt (https://www.uniprot.org/) and downloaded all reviewed sequences (n=568,002, accessed on 8/16/2022). To later evaluate how well our model discriminates MDH from functionally close enzymes, we used the validation set (n=213) from Repecka et al. as positive set; we also identified 5,766 EC1.1.1.X sequences (first three enzyme commission [EC] [34] levels are 1.1.1 and the last level is not 37) and 543 EC1.1.X.X sequences (first three EC levels are 1.1 and the third level is not 1) as negative set. We then narrowed these sequences to those a) whose EC numbers are not 1.1.1.37 (the EC number of MDH); b) whose sequence lengths are within the range of the positive set's (64-505); and c) that are not functionally close (EC1.1.1.X or EC1.1.X.X), resulting in 206,596 non-MDH sequences as negative set.

To compile a balanced negative set without bias in sequence length distribution, we enumerated the number of sequences by an interval of 50 and sampled that amount of non-MDH sequences for that interval. For example, if there were 39 MDH sequences that are 110-160 residues long, then we sampled 39 non-MDH sequences of that length range. As a result, we collected 16,706 non-MDH sequences as negative set.

Since the 16,706 MDH sequences may vary a lot (with a length range of [64-505]), to ensure consistency during the finetuning process, we clustered the MDH sequences into multiple clusters using cd-hit [35]. To reduce the number of resulting clusters, we used the lowest allowed sequence identity threshold (0.4) and word length (2) for clustering. A total of 82 clusters were generated and the majority of these clusters are singletons or very small. The four largest clusters constituted 97.3% (n=16,258) of all MDH sequences. We termed these four clusters as *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*, with the number inside the name indicating the number of sequences in that cluster (e.g., *cluster765* contain 765 sequences).

Generator finetuning and sample generation

For the AMP generation task, we finetuned ProtGPT2 [11] (https://huggingface.co/nferruz/ProtGPT2) with default parameters on the AMP set. We then generated 3,000 sequences using text generation pipelines from Python (v. 3.8.15) transformers package (v. 4.26.0.dev0) [36] (https://huggingface.co/docs/transformers/index) with maximum lengths being 20. Note that the generated sequences may exceed this maximum length. As negative control for the finetuning process, we also generated 3,000 sequences using the non-finetuned ProtGPT2 model.

For the MDH generation task, we finetuned the ProtGPT2 model using four separate datasets: *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*, resulting in four finetuned models. To generate new sequences, we supplied each finetuned model with a starter sequence of varying length, ranging from 0-100 residues, taken from the representative sequence of the corresponding finetuning set. The maximal length for generation is set to be 400. The results of this sample generation are summarized in Table 1 below. *All finetuning tasks were conducted on* NVIDIA A100 (80G) graphical processing unit (GPU).

Table 1. Summary of the MDH sequence generation.

Group name	Group meaning	Template	Starter seq. length	No. generated seq.
random	randomly generated according to the amino acid occurrence distribution of all MDH	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000
noft	generated by non-finetuned ProtGPT2	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000
cluster765	finetuned by *cluster765*	representative seq. of *cluster765*	0	1000
			10	1000

			50	1000
			100	1000
cluster2029	finetuned by *cluster2029*	representative seq. of *cluster2029*	0	1000
			10	1000
			50	1000
			100	1000
cluster5987	finetuned by *cluster5987*	representative seq. of *cluster5987*	0	1000
			10	1000
			50	1000
			100	1000
cluster7477	finetuned by *cluster7477*	representative seq. of *cluster7477*	0	1000
			10	1000
			50	1000
			100	1000

Discriminator training

For the AMP discriminator, we split the balanced AMP/non-AMP set (n=2,879+2,879) into training, validation, and testing set in 8:1:1 ratio. From the MDH/non-MDH balanced set (n=16,706+16,706), we randomly sampled 4,500 sequences for training (13.47%) and 500 sequences for validation (1.50%), and the remaining 28,412 sequences were used for testing (85.03%). We performed discriminator training, hyperparameter tuning, and performance evaluation on training set, validation set, and testing set, respectively.

To extract features from the protein sequences, we utilized the ProtT5-XL-UniRef50 model, which was pretrained and publicly available in the ProtTrans project [37]. We chose ProtT5-XL-UniRef50 among others in ProtTrans because it achieved the best performance most often in different predictive tasks [37]. We used the transformers package (v. 4.26.0.dev0) in Python (v. 3.8.15) to implement the embedding, with maximum lengths of 50 for the AMP task and 505 for the MDH task.

For the discriminator architecture, we used Keras package (v. 2.10.0) [38] to implement a convolutional neural network (CNN) followed by a fully connected neural network (FCNN) (Figure 1). Each convolution layer was coupled with a maximum-pooling layer, and the last maximum-pooling layer was flattened, which was fed to the FCNN. Recticfied linear unit [39] was used as activation function for all layers except for the output layer, where the activation function was sigmoid function [40] instead. We then used Adam [41] as the optimizer, binary cross-entropy (Eqn. 1) as the loss function, and accuracy (Eqn. 2) as the evaluation metric. We set maximum number of training epochs to be 100 with early stopping (patience=5). We adopted the following hyperparameter setup after tuning: three 3×3 convolution filter, each followed by 2×2 max-pooling layer; a flatten layer; three fully connected layers (with 1024, 512, 256 nodes, respectively) and one output layer. Tunable hyperparameters include number of filters in CNN, kernel size in CNN, maximum-pooling size, number of dense layers in FCNN, number of nodes in each FCNN layer, batch size, and learning rate of optimizer. Embedding and training were carried out on a NVIDIA A100 (80G) graphical processing unit.

$$\text{binary cross_enthropy} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)\right) \qquad \text{(Eqn. 1)}$$

where N is the number of samples, y_i and $\hat{y}_i$ are the *i*-th actual and predicted label, respectively.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \qquad \text{(Eqn. 2)}$$

where *TP*, *TN*, *FP*, and *FN* are true positive, true negative, false positive, and false negative, respectively.

Bioinformatics analysis

We performed Pfam domain [42] scanning on all generated MDH sequences using HMMER package [43] (v. 3.3.2). Pfam profiles were built using hmmerbuild command with Pfam seed alignment flatfile and and Pfam profile flatfile (downloaded from http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.seed.gz and http://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.hmm.gz , respectively. accessed on 11/10/2022). The E-value threshold for Pfam domain scanning was set as 0.001.and we considered

a sequence to have MDH's signature domain if at least one "lactate/malate dehydrogenase" or "Malate/L-lactate dehydrogenase" was identified.

We evaluated the sequence redundancy within *cluster765*, *cluster2029*, *cluster5987*, *cluster7477*, as well as the generated MDH sequences using cd-hit (parameters set as default). We defined "ratio of non-redundant sequences" as the number of non-redundant sequences determined by cd-hit divided by the number of input sequences.

For every generated MDH sequence, we performed pairwise alignment using BLAST standalone package [44] (v. 2.13.0+) against all known MDH sequences (n=16,706) and identified the "best hit" (highest bit score) sequence. We recorded the identity between query (generated) sequences and their corresponding best hit sequence. Based on these best hit sequences' identity and ungapped alignment length, we utilized HFSP curve (homology-derived functional similarity of proteins [45], HFSP=0) to visualize wehther a generated sequence share the same function as its best hit known MDH.

Sequence space visualization using t-SNE

We randomly selected 100 known MDHs from *cluster2029*, 100 MDH candidates generated by *cluster2029*-finetuned ProtGPT2 that are prioritized (non-redudant and discriminator prediction>0.95), 100 EC1.1.1.X and 100 EC1.1.X.X for visualization of sequence spatial distribution with t-distributed stochastic neighbor embedding (t-SNE) algorithm [46]. Each of the selected sequences were embedded by the ProtT5-XL-UniRef50 model [37] as an L×1024 matrix (where L is the length of the protein), which was then flattened into a L×1024-long vector as protein-level features. The t-SNE transformation from the embedded features to two-dimensional space was conducted using Python scikit-learn package (v1.1.2) [47].

Mapping conservation to enzyme functionally important sites

Finetuned-generated MDH candidates that are a) non-redundant and b) with discriminator prediction>0.95 were selected. We used a known MDH sequence (UniProt ID: P11708) with determined protein structure (protein data bank [PDB] [48] ID: 4MDH) as the reference sequence and

was BLASTed against the selected MDH candidates. From the selected MDH candidates with significant hit, 499 were randomly sampled and then subjected to multiple sequence alignment (MSA) along with P11708 using Clustal Omega [49] program on EMBL-EBI (https://www.ebi.ac.uk/Tools/msa/clustalo/). The MSA-derived conservation scores were computed using JalView program [50]. Annotation of functionally important sites (substrate binding, NAD binding, and sulfate binding) of 4MDH were obtained from PDB. Conservation scores were mapped to residue positions in 4MDH. Visualization of 4MDH was performed in PyMOL [51].

Statistical analysis and visualization

Statistical analysis and visualization were conducted in R (v. 4.2.2) with ggplot2 package (v. 3.4.0). Calculation for area under the receiver operating characteristic curve (ROC-AUC) was done with pROC package [52] (v. 1.18.0).

Experimental validation of AMP candidates

Solid-phase peptide synthesis was used to synthesize crude peptides (which contained 40-70% of the target peptide) by GenScript Biotech in Nanjing, China. Mass spectrometry was used to determine their precise molecular weights. The peptides used for the minimum inhibitory concentration (MIC) were further purified to a 90% degree of purity, as determined by high-performance liquid chromatography.

We used six strains of bacteria in our experiment: four Gram-positive strains (*Staphylococcus aureus* GDMCC 1.221, *Bacillus pumilus* GDMCC 1.225, *Bacillus subtilis* GDMCC 1.222, and *Kocuria rhizophila* GMDCC 1.226) and two Gram-negative strains (*Escherichia coli* GDMCC 1.335 and *E. coli* DH5α). All strains were obtained from Guangdong Microbial Culture Collection Center in Guangzhou, China, except for *E. coli* DH5α, which was maintained in our lab. Each bacterial strain was streaked on Luriae-Bertani (LB) agar medium and incubated overnight at the optimum growth temperature according to the provider's instructions. Individual colonies were picked and inoculated into LB culture medium, which was then shaken at 200 rpm at the optimum growth temperature overnight. The bacterial suspension was diluted 1:100 using fresh LB culture medium and then cultured for 4-5 hours to reach the logarithmic phase (OD600 of 0.4-0.6). The suspension was adjusted to OD600=0.1 by

LB solution and then diluted 1,000 times. A volume of 100 μL cell suspension was transferred into 96-well plates for the next test.

For the bacterial inhibition experiment, we used crude peptides with a target peptide content of approximately 50% for ease of calculation. Freeze-dried peptide powders were dissolved in dimethylsulfoxide (DMSO) to a concentration of 3 mM target peptide. For the antibacterial activity test, peptides (or an equal volume of DMSO solution as a negative control) were added to 96-well plates (each well containing 100 μL cell suspension). The peptide solution and fresh LB solution were adjusted to reach a concentration of 60 μM. Blank controls were set up using LB solution without peptide/DMSO. OD600 values were measured after shaking at 500 rpm at the optimum growth temperature for 12 hours using a high-speed oscillating incubator (ZQZY-88BHS, Shanghai Zhichu Instrument Co., Ltd., China). The blank control OD600 value was used for data normalization. Dunnett's test was used to compare experimental groups with the control group (two-sided) using OriginPro software (version 9.2, 64 bit). All experiments were performed with three independent replicates.

The peptides used for the MIC determination were more than 90% pure. We prepared a series of peptide concentrations ranging from 0.94 μM to 120 μM. The same culture steps were conducted as described above, except that the final incubation step lasted for 16-18 hours. MICs were determined as the minimum concentration of AMPs where bacteria showed no detectable growth. All experiments were performed with three independent replicates.

Results

Overview of the DNPD framework

We developed a convenient one-stop solution (Figure 1) for general-purpose protein design by integrating a state-of-the-art protein generative model, ProtGPT2 [11] with a custom-built protein discriminator. The ProtGPT2 model, a large-scale language model with 738 million parameters, serves as the foundation for generating new protein sequences. The protein discriminator, consisting

of a convolutional neural network followed by a fully connected neural network (Figure 1A), is used to assess and prioritize the generated sequences. Protein sequence features are extracted using ProtT5-XL-UniRef50, a model-based embedding framework in ProtTrans collection, which comprises of 3 billion parameters [37]. The generative model is finetuned with a positive set of protein sequences, while the discriminator is trained on a balanced dataset containing both positive and negative protein sequences. As proof of concepts, we applied this framework to the design of two distinct protein classes: antibacterial peptides (AMP) and malate dehydrogenases (MDH; EC1.1.1.37).

Discriminators accurately classifies AMP/non-AMP and MDH/non-MDH sequences

For the AMP/non-AMP discriminator, the AMP and non-AMP sequences were obtained from SATPdb database [32] and UniProt [33], respectively. Short peptides from UniProt that are shorter than 50 residues and without certain antimicrobial-relevant keywords were selected as non-AMPs. Similarly, for the MDH/non-MDH discriminator, we acquired MDH sequences from Repecka et al. [20] and non-MDH sequences randomly sampled from UniProt [33]. To ensure unbiased results, all training, validation, and test sets were carefully balanced by randomly sampling negative sets to mirror the sequence length distribution in positive sets. The discriminators achieved high accuracies and converged rapidly (<100 epochs), demonstrating the effectiveness of transfer learning-based embedding techniques. Specifically, the AMP/non-AMP discriminator achieved an accuracy of 0.81 on a balanced testing set of 576 samples (Figure 2A), while the MDH/non-MDH discriminator attained an accuracy of 0.99 on a larger balanced test set of 28,412 samples (Figure 2A). It is important to note that numerous AMP/non-AMP discriminators have been developed by other researchers [53]. However, our goal is not to compare our discriminator with these existing models. Instead, we aim to demonstrate that our framework is versatile and independent, capable of performing well in various protein discriminative tasks without relying on pre-existing discriminators.

To further evaluate the discriminators' capability to differentiate between functionally close enzymes, we assembled additional datasets comprising EC1.1.1.X enzymes (n=5,766, excluding EC1.1.1.37) and EC1.1.X.X enzymes (n=543, excluding EC1.1.1.X), as well as an additional reserved set of 213 MDHs for further examination. All of these sequences were unseen during training. The MDH/non-

MDH discriminator achieved a 0.99 ROC-AUC for both EC1.1.1.X vs. MDHs and EC1.1.X.X vs. MDHs (Figure 2B), highlighting its exceptional ability in differentiating between functionally similar enzymes. Overall, the results demonstrated the effectiveness of the discriminators in accurately distinguishing protein sequences of interest.

Finetuned-generated sequences are predicted to be likely AMP or MDH

Finetuning essentially involves adapting a pre-trained LM to a custom input dataset, enabling the updated LM's to generate content more closely aligned with the input's style [54]. We finetuned ProtGPT2 with 2,879 known AMPs and subsequently generated 3,000 AMP candidates using the finetuned model. During generation, the "maximal length" was set to 20 and no starter sequence was provided, however, it was to be noted that the actual generated sequences may exceed the set "maximal length". For comparison, we created an equal amount of a) random sequences, i.e., randomly composed sequences with identical amino acid and sequence length distribution as natural AMPs; and b) ProtGPT2-random sequences, i.e., sequences generated by the non-finetuned ProtGPT2. As expected, the finetuned-generated sequences are more likely to be actual AMPs than both random (prediction median 0.87 vs. 0.37) and ProtGPT2-random sequences (prediction median 0.87 vs. 0.01, Wilcoxon signed rank test p-value<2.2e-16; Figure 3A). The fact that even random sequences displayed a relatively high percentage (43.2%) of being predicted as AMPs by the discriminator is also noteworthy (Figure 3A), as it could indicate a need for more accurate negative sets or suggest that many short random peptides may exhibit antimicrobial effects.

To explore the impact of data size and sequence redundancy on finetuning results for MDHs, we clustered the 16,706 known MDHs and selected the four largest clusters (termed *cluster765*, *cluster2029*, *cluster5987*, and *cluster7477*) as finetuning sets, comprising 765, 2029, 5987, and 7477 sequences, respectively. As controls, we generated two sets of sequences: a) random sequences that followed the amino acid and length distribution of the known MDHs; and b) sequences generated by the non-finetuned model. Discriminator predictions revealed that sequences generated by the finetuned models were significantly more likely to be MDHs than random and non-finetuned model-generated sequences (prediction median 0.91 vs. 0.003 and 0.91 vs. 0.033, respectively; Wilcoxon

signed rank test p-value<2.2e-16 for both comparison; Figure 3B). Notably, we found that even a relatively small number of sequences (~700, in the case of cluster765) were sufficient to fine-tune ProtGPT2 for generating high-quality MDH sequences. Importantly, the discriminator's predicted scores do not correlate with ProtGPT2's perplexity during generation (Pearson correlation= -0.01). Considering the aforementioned limitations of perplexity, we further demonstrated the superiority of our discriminator in candidate prioritization.

Note that users can choose to provide a leading sequence to prompt ProtGPT2 during sequence generation, which we refer to as "starter sequence". To investigate whether such a starter sequence would guide the generation towards our desired functions, we incorporated starter sequences of variable lengths. We supplied the first 0 (i.e., none), 10, 50, or 100 residues of a representative sequence (the representative sequence in the corresponding cluster; see Materials and Methods) as a starter sequence during the generation process. As a result, providing a starter sequence did not necessarily improve the quality of the generated sequences, as candidates generated without starter sequences generally had higher discriminator predicted scores (Figure 3B). Notably, in *cluster5987 and cluster7477, employing a starter sequence adversely impacted prediction accuracy. This suggests that, for a diverse training set, the inclusion of a starter sequence might impose unnecessary constraints on sequence generation.*

We observed that most of the generated sequences were non-redundant identified by cd-hit [35], except for those generated by the *cluster2029*-finetuned model (ratio of non-redundant sequences =0.31 vs. 0.67-0.95 in other generated groups), likely due to the low non-redundancy within the cluster itself (ratio of non-redundant sequences =0.08 vs. 0.28-0.33 in other clusters; Figure 4A). The effect of supplying a starter sequence on the redundancy of generated sequences was not conclusive (Figure S1).

Prioritized generated AMP candidates are endowed with antibacterial activity

With the generated AMP candidates and their corresponding discriminator predictions available, we randomly selected 24 AMP candidates with predicted scores greater than 0.95; meanwhile, 10 random peptides were generated as control (Table S1). The amino acid and peptide length

distributions of the random peptides are the same as the natural AMPs'. These peptides were tested against six different bacterial strains, including four Gram-positive strains (*S. aureus*, *K. rhizophila*, *B. pumilus*, and *B. subtilis*) and two Gram-negative strains (*E. coli* GDMCC 1.335 and DH5α). Out of the 24 generated AMP candidates, six completely inhibited the growth of at least one bacterial strain (Figure S2). Notably, five of these candidates displayed broad-spectrum activity against all six bacterial strains. In comparison, only one of the randomly selected peptides exhibited activity against some bacterial strains, highlighting the presence of many unknown AMPs in nature, as reported in previous studies [55,56]. To accurately assess the potency of these candidates, five broad-spectrum AMPs were purified to 90% purity and their minimum inhibition concentrations (MICs) were determined by a growth-based assay. Four of these candidates displayed remarkably low MICs (with the lowest at 1.875 μM, Table 2), highlighting their potential for therapeutic and industrial applications.

Most finetuned-generated MDH candidates harbor MDH domains

A domain is a structural unit that carries a certain protein function. The function of MDHs is characterized by a "lactate/malate dehydrogenase" or a "Malate/L-lactate dehydrogenase" domain (here we denote as "MDH domain") catalogued in Pfam database [57]. Thus, bearing such signature domain serves as a good indicator of generation quality in addition to our discriminator. We found that not a single random or non-finetuned-generated sequence has the MDH domain (Figure S3), suggesting that this domain is extremely unlikely to occur by chance. In contrast, the vast majority (94.54%) of finetuned-generated sequences have MDH domains, increasing the confidence in their potential to possess the desired MDH function. Besides, the discriminator's predictions on sequences with MDH domains are significantly higher than those without MDH domains (prediction median 0.95 vs. 0.02, Wilcoxon signed rank test p-value<2.2e-16; Figure 4B), assuring that the discriminator can well capture the sequence-function relationship.

Homology inference suggests MDH function in most finetuned-generated MDH candidates

The MDH candidates generated through fine-tuning exhibit high sequence identity (median=83.65%) with their best hit known MDH (Figure 4C). Meanwhile, we believe that some high identity candidates may not actually be MDH (Figure 4D, bottom right corner) according to our discriminator. In contrast,

some candidates with moderate best hit identities (60-75%) exhibit high predicted scores, enabling researchers to select MDH candidates with a relatively high degree of novelty. Next, to further validate the utility of the discriminator in light of homology, we selected a set of generated candidates highly likely to be MDH, i.e., prediction>0.95, defined as *prioritized candidates*. We applied HFSP (homology-derived functional similarity of proteins) [45] to the *prioritized candidates*. HFSP assesses whether a pair of proteins share the same function based on ungapped alignment length and alignment identity. Although resolution of HFSP may not be able to reach the fourth level of EC number [45], it is still a useful baseline for homology reference. Notably, HFSP predicted that all prioritized generated candidates have the same function as their corresponding best hit sequence (Figure 4E), i.e., a known MDH. These results reinforced our confidence in the quality and functionality of the MDH sequences generated through fine-tuning, as supported by homology analysis.

Prioritized MDH candidates are close to known MDHs in sequence space

To better understand the spatial relationships between sequence-derived features of the known MDHs and the discriminator-prioritized MDH candidates, we first embedded the protein sequences into L×1024-dimensional vectors (where L is the length of a protein) using ProtTrans ProtT5-XL-UniRef50, we than used t-SNE algorithm [46] to transform the vectors into a two-dimensional space. Here, we randomly selected 100 known MDHs and 100 *prioritized candidates* to be visualized. To avoid bias caused by redundant sequences, we further limited the *prioritized candidates* to be non-redundant with each other determined by cd-hit [35] during the random selection. Functionally similar enzymes (EC1.1.1.X and EC1.1.X.X) were also added into the comparison as reference. The resulting visualization (Figure 5) showed that known MDHs and the *prioritized candidates* clustered closely together, while functionally similar enzymes barely locate in the space between the known MDHs and the prioritized MDH candidates. This finding suggests that, in terms of sequence context, the *prioritized candidates* are systematically in line with the known MDHs and are distinct from functionally similar enzymes.

Prioritized MDH candidates are conserved at important functional sites

We randomly selected 499 *prioritized candidates* to perform multiple sequence alignment (MSA) with a known MDH with structural information (UniProt ID: P11708; PDB ID: 4MDH). From the MSA, we derived position-specific conservation scores and mapped them to 4MDH's structural annotations. We found that the conservation scores at substrate binding sites, NAD binding sites, and sulfate binding sites were significantly higher than those at other sites (11 vs. 7, t-test p-value<0.00025 for all three comparisons; Figure 6). These results suggest that our framework has successfully learned the sequence patterns required for MDH function and that the resulting MDH candidates are likely to perform this function.

Discussion

AMPs offer hopes in addressing the global antibiotic-resistance crisis [58] due to their advantages over small molecule antibiotics, e.g., modulation of immune response, broad-spectrum activity, multiple mechanisms of action, and diverse source organisms from all domains of life [59]. Numerous attempts have been made to predict and *de novo* design novel AMPs [60-62]. Recently, there are four deep generative language model-based methods for AMP design with experimental validation [56,63-65]. The first three methods involved an RNN-based generator and a predictor, both trained on known AMPs. In addition to the predictor, a variety of physicochemical properties were also employed to facilitate candidate filtering. The fourth method [65] is more complex—an autoencoder generator was trained on all known short peptide sequences to better reflect peptide properties and enable high diversity. Meanwhile, multiple attribute classifiers (such as antimicrobial activity, toxicity, sequence similarity, and physicochemical properties) were trained on the latent space of the autoencoder, and the generation process was guided by these attribute classifiers altogether for desired peptide properties, and the attribute classifiers and a series of external predictors were used for candidate screening. All of these language model-based methods, as well as GAN-based methods, e.g., [66], have successfully identified experimentally valid novel AMPs, some of which are broad-spectrum. However, performance comparison across these methods is difficult due to variations in data size, time consumption for training, threshold of "inhibitory effect", definition of "broad-spectrum", bacterial

strains tested, peptide purity, and sequence novelty. From a usage convenience standpoint, our framework has two major advantages over other language model-based methods. First, it does not need to train a generator from scratch, as the existing generator (ProtGPT2) assumably performs better due to the tremendous training protein dataset. Second, the candidate prioritization is straightforward as it relies on only one discriminator, instead of multiple classifiers or additional calculation of physicochemical properties.

Enzyme engineering has become an essential aspect of metabolic engineering and synthetic biology [67,68] . When incorporating a heterogenous enzyme into an engineering chassis, it is often necessary to adjust the enzyme properties to better fit the working condition. Generating artificial protein sequences offers more options for selecting the ideal enzyme version. For example, a variety of tools can be applied to screen for more suitable solubility [69,70], catalytic capacity [71,72], and protein-protein interaction [73,74]. Many attempts have been made for enzyme DNPD [75,76]. One of the most successful non-LLM frameworks is ProteinGAN [20], a GAN-based generator. As a showcase, ProteinGAN was trained on 16,706 MDHs and produced 55 candidates for testing, of which 13 candidates experimentally displayed MDH catalytic activity.

Compared to ProteinGAN, our framework has several advantages. First, our discriminator was trained on much less protein sequences than ProteinGAN (n=2,029 vs. n=16,706) without compromising the accuracy (0.99 accuracy on a much larger testing set: n=28,412). Second, it can distinguish MDHs from functionally similar enzymes at a high resolution, surpassing the limitations of traditional homology-based methods [45]. The candidate screening process is therefore remarkably simplified and improved, eliminating the need for additional predictors and replacing sequence identity-based criteria with a straightforward sequence-function scoring scheme. With that said, it allows us to select both conservative (high sequence identity) and novel (low sequence identity) candidates while maintaining high functional fidelity (high discriminator scoring). Last but not least, our framework is time-efficient. The time consumption for the MDH task (in the case of the *cluster2029* for example) is dissected as follows: ~10 min generator finetuning (n=2,029), ~1 h embedding the training/validation sets (n=4,500 + 500), 2 min discriminator training/validation (n=4,500 + 500), ~4 h

generation (1,000 sequences), ~20 min embedding the generated sequences (n=1,000) , and ~1 min prediction. The MDH task can be completed in 6 hours (without parallel computing), which is a small fraction of the time it took for ProteinGAN (9 days). Designing novel MDHs was also attempted in Madani et al.'s work on ProGen [12]. In their work, the aforementioned per-token log-likelihood scoring, as well as ProteinGAN's discriminator were used for the prioritization on the generated MDH candidates, achieving 0.94 and 0.87 ROC-AUC, respectively, on 56 samples obtained from the ProteinGAN paper [20]. In comparison, our discriminator achieved higher ROC-AUC (0.99) on a much larger test set (n=28,412), demonstrating the accuracy of our discriminator.

A major limitation of our framework is that finetuning ProtGPT2 requires a large GPU memory: among the NVIDIA GPU hardware we have tried (A100, A10, V100, T4, P4, P100), only NVIDIA A100 (80G) was able to handle the workload. A minor disadvantage is that, unlike GAN requires only positive set, a negative set is also needed to train our discriminator. However, obtaining a negative set from protein sequence databases such as UniProt [33] is straightforward.

In summary, we developed a framework as a one-stop solution for general DNPD tasks. This framework consists of ProtGPT2 and a CNN-based discriminator with protein features extracted by a pre-trained embedding scheme. Our proof-of-concept experiments have shown that the discriminator is efficient in sequence generation and accurate in protein classification. Therefore, this framework may have the potential to significantly accelerate AI-based DNPD.

Data and code availability

Codes for the framework and result analysis, as well as the data to replicate this work, were deposited in GitHub repository (https://github.com/zishuozeng/GPT_protein_design).

Acknowledgements

We would like to thank Yu Han (SIAT) for the constructive discussion and all researchers who made the data and tools relevant to this study available.

Author contributions

Z.Z. and X.L. conceived the ideas; Z.Z. performed the modeling, data analysis, and result visualization; R.X. performed the AMP experimental validation and wrote the methods for it; Z.Z. wrote all other parts of the manuscript; J.G. performed preliminary finetuning experiments; X.L. revised the manuscript.

Funding

The Project Supported by National Key R&D Program of China (2019YFA0904100), National Natural Science Foundation of China (32001079), Shenzhen Science and Technology Program (ZDSYS20210623091810032), Shenzhen Institute of Synthetic Biology Scientific Research Program (JCHZ20200004).

Competing interests

X.L. has a financial interest in Demetrix and Synceres.

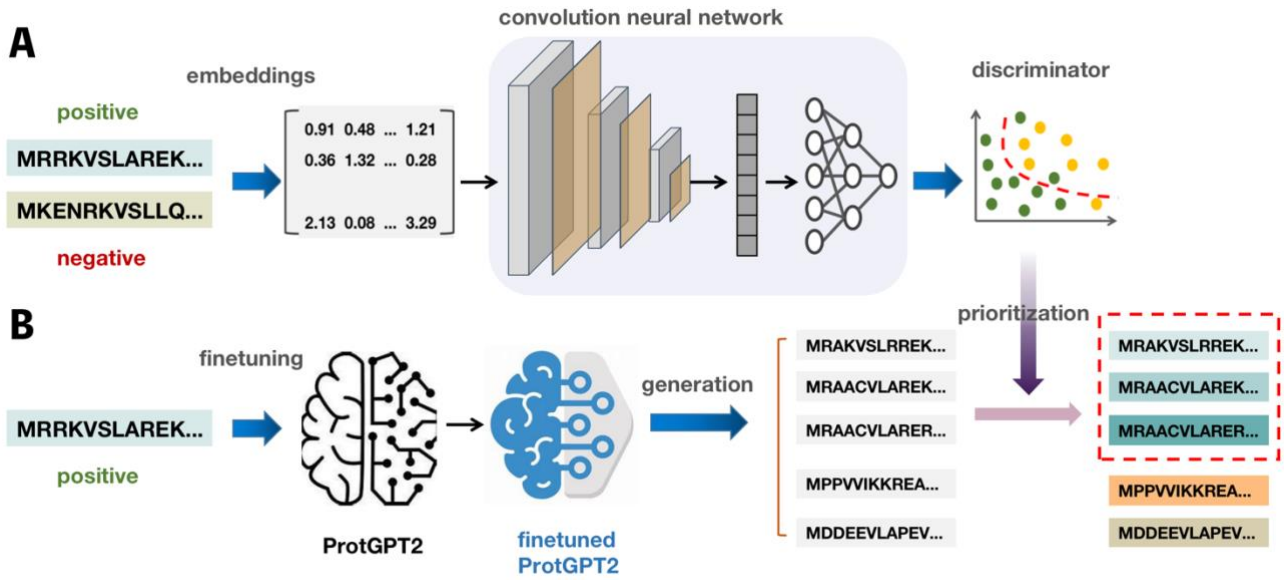

Figure 1. **Discriminator (A) and generator (B) architecture.** In panel A, positive and negative sequences are embedded using ProtTrans (each residue is converted to a 1,024-value long vector). The embedded inputs are fed to a neural network with three convolution layers and three max-pooling layers, followed by a fully connected neural network. Meanwhile, in panel B, the positive sequences are used to finetune ProtGPT2, which is then used to generate positive candidate sequences. In the end, the discriminator is used to screen the most likely positive sequence candidates.

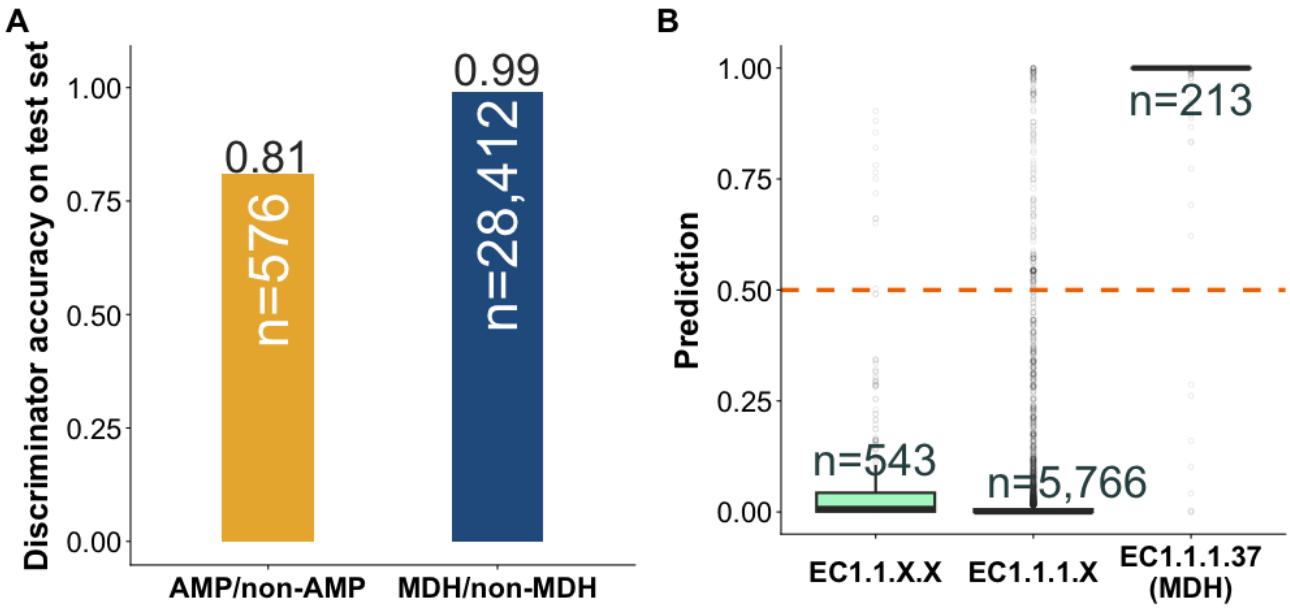

Figure 2. Accuracies (A) and resolutions (B) of the discriminators. In panel A, numbers denoted inside the bars are size of testing set (both balanced). Panel B shows MDH/non-MDH discriminator's predictions on MDHs and functionally close enzymes: "EC1.1.1.X" are enzymes with the first three EC digits being 1.1.1. (excluding EC1.1.1.37); likewise, "EC1.1.X.X" are enzymes with the first two EC digits being 1.1. (excluding EC1.1.1.X). Numbers above the boxes are data size, and all data are not included in training set. AUCs for EC1.1.X.X vs. MDH and EC1.1.1.X VS. MDH are both 0.99.

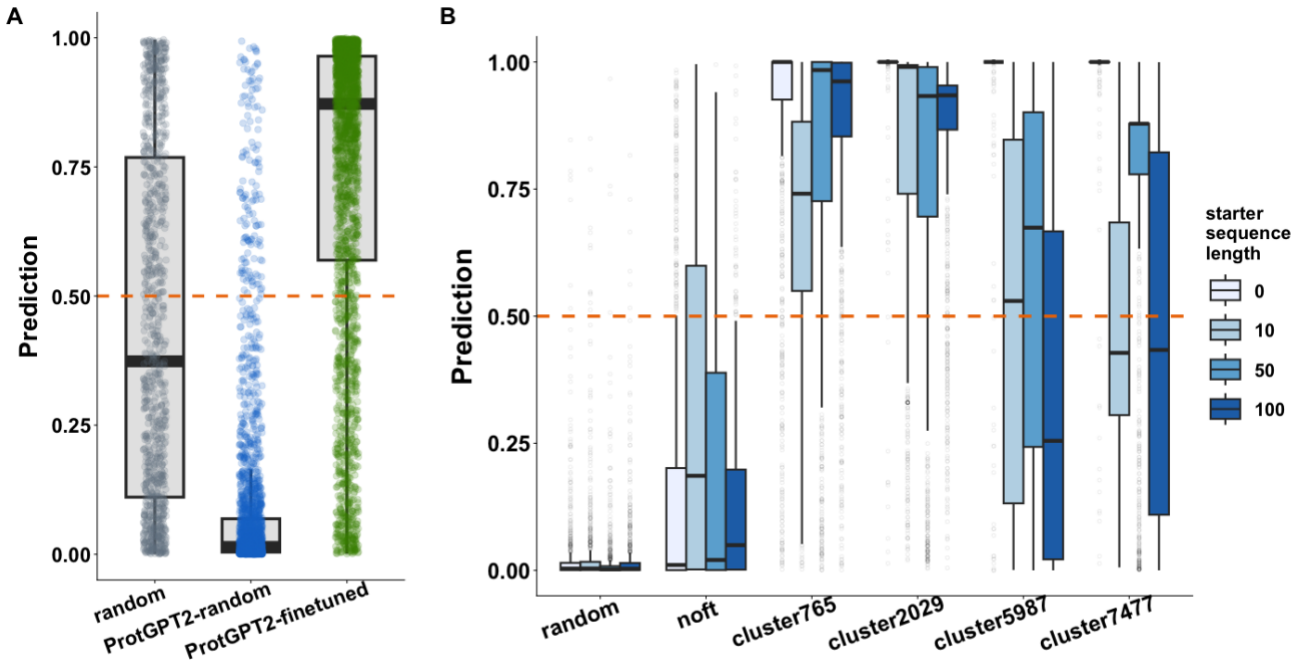

Figure 3. Effects of generator finetuning on AMP (A) and MDH (B) likelihood of the generated sequences. Group "random" in panel A and B means random sequences following amino acid distributions of known AMPs and MDHs, respectively. "ProtGPT2-random" in panel A and "noft" in panel B means candidates generated by non-finetuned ProtGPT2, respectively. Other groups in panel B (*cluster765*, *cluster2029*, *cluster5987* and cluster7377) indicates sequences generated by ProtGPT2 finetuned by the respective sequence cluster (the numbers indicate corresponding cluster size).

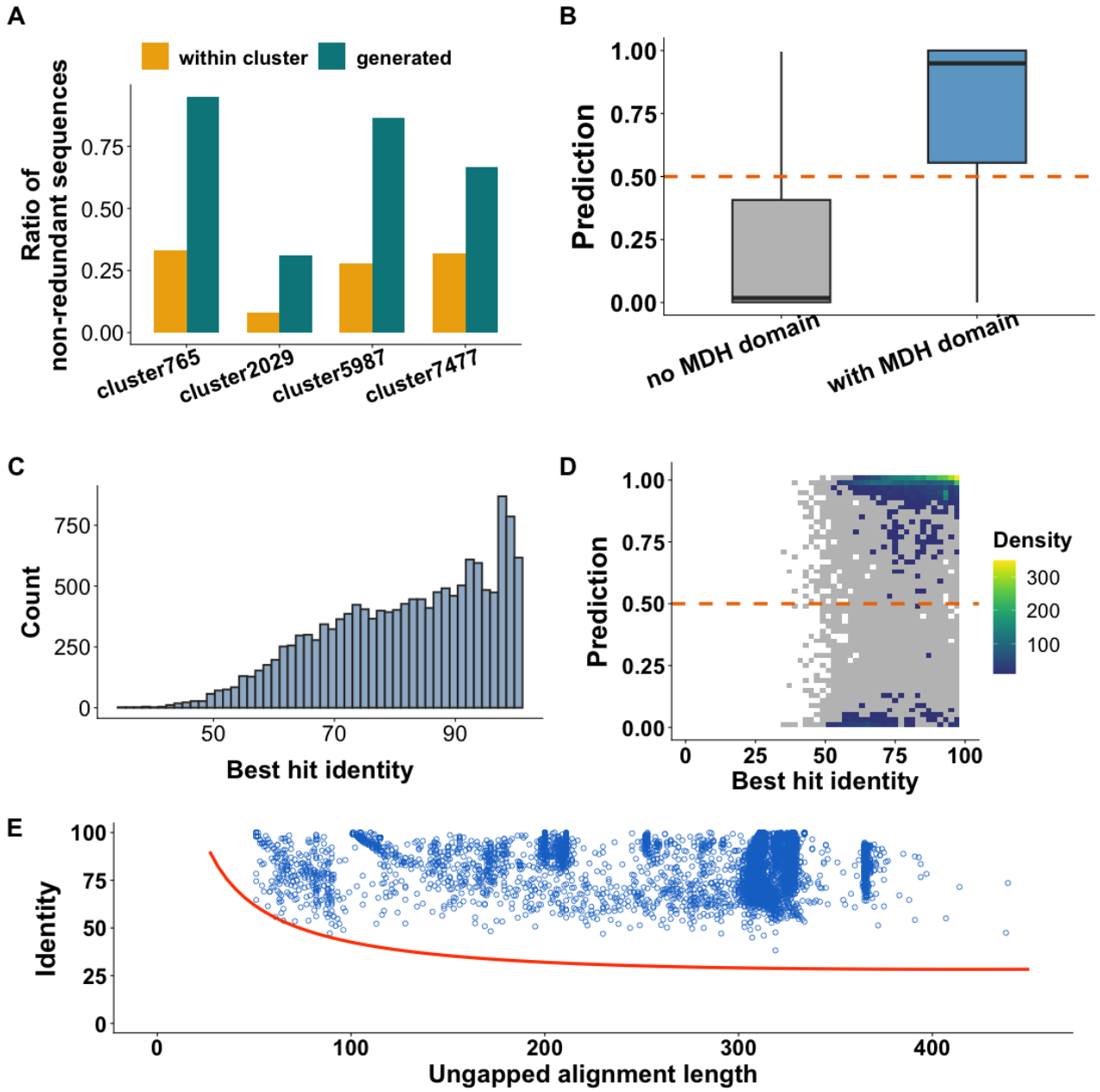

Figure 4. Redundancy, MDH domain, best hit identity, and homology inference of the generated MDH candidates. Panel A shows the ratios of non-redundancy within each cluster (*cluster765*, *cluster2029*, *cluster5987* and cluster7377) and among the generated sequences finetuned by the corresponding cluster. Panel B shows discriminator predictions on generated MDH candidates with and without MDH domains. Panel C shows the histogram of best hit identity of the generated MDH candidates. Panel D shows discriminator's prediction vs. best hit identity (presented as data point density, grey grids contain <10 data points) for the finetuned-generated MDH candidates. Panel E shows the best hit's identities and ungapped alignment lengths of generated sequences (blue dots) with high predicted scores (>0.95). Red curve in panel E indicates HFSP curve: sequence pairs above and below the curve are predicted by HFSP as having the same and different enzyme functions, respectively.

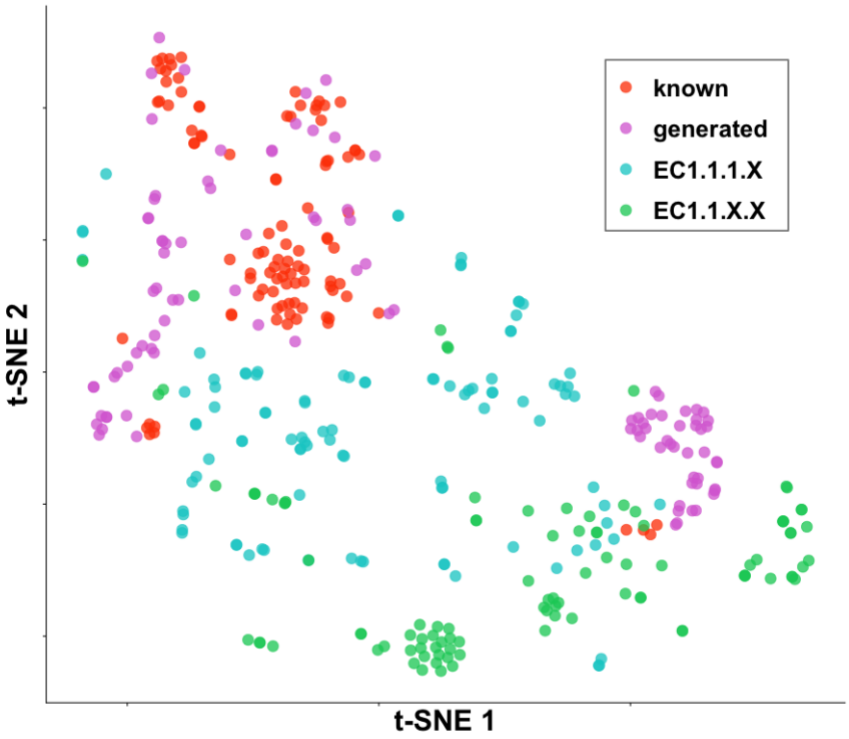

Figure 5. Two-dimension t-SNE plot on sequences embedded as vectors. The known MDH sequences (n=100) were sampled from *cluster2029*. The generated sequences (n=100) were sampled from MDH candidates generated by ProtGPT finetuned by *cluster2029* that are prioritzied (non-redundant, discriminator prediction>0.95). Enzymes with EC number EC1.1.1.X (excluding EC1.1.1.37, i.e., MDH, n=100) and EC1.1.X.X (excluding EC1.1.1.X, n=100) were added to the plot to illustrate the relative spatial proximity between MDHs and functionally close enzymes. The t-SNE plot was generated based on protein numerical vector representation embedded by the ProtT5-XL-UniRef50 model.

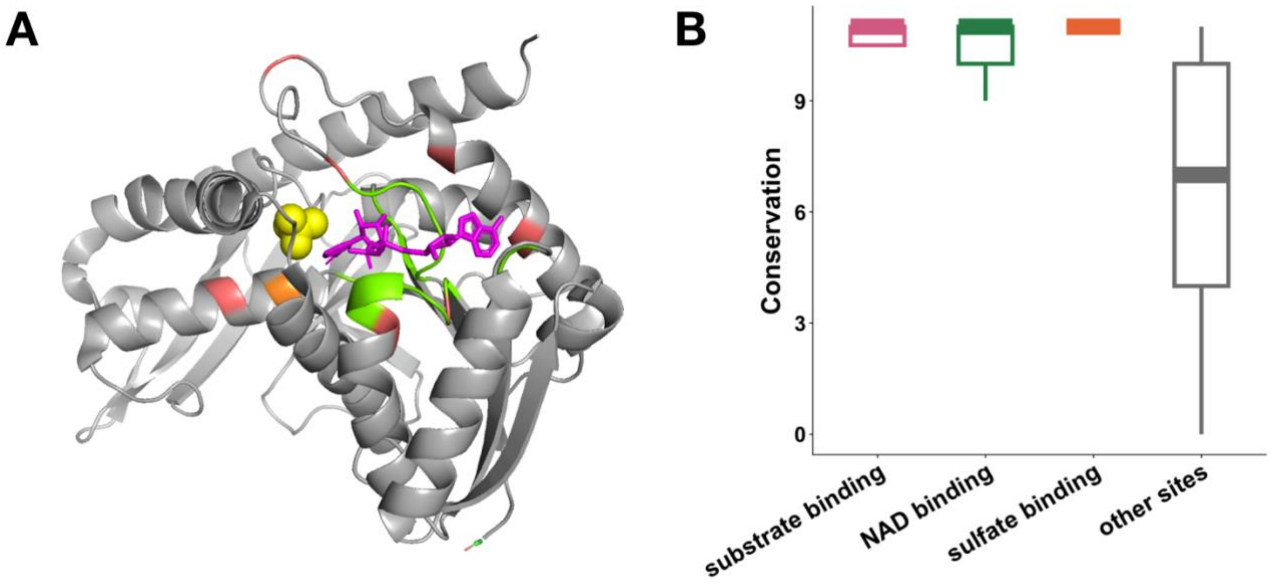

Figure 6. Conservation along generated MDH candidates with respect to functionally important sites. Position-specific conservation scores were derived from multiple sequence alignment (MSA) among 500 sequences (a known MDH sequence [UniProt ID: P11708] and 499 randomly sampled finetuned-generated MDH candidates with discriminator prediction>0.95). The determined protein structure (PDB ID: 4MDH) of P11708 was added to the MSA to map the protein positions to functionally important sites. Panel A shows the structure of 4MDH with ligands sulfate (yellow spheres) and NAD (purple licorie) highlighted. Resides highlighted in pink, green, and orange indicate substrate binding, NAD binding, and sulfate binding sites, respectively. Panel B summarizes the conservation scores at different sites in boxplot.

Table 2. MIC of the identified broad-spectrum AMPs.

Peptides ID	sequence	length	prediction	mass	MIC (μM)					
					S. aureus **GMDCC1.221**	***B. pumilus*** **GMDCC1.225**	***B. subtilis*** **GDMCC1.222**	***K. rhizophila*** **GDMCC1.226**	***E. Coli*** **GDMCC1.355**	***E. Coli*** **DH5α**
AMP3	GLFKIFKKSVKHACKKLK	18	0.97	2103.7	15	3.75	3.75	15	7.5	3.75
AMP6	KYLRRKGGRKWVKNAFRNLM	20	0.97	2522.1	60	15	60	120	120	120
AMP13	YRGIRGKGWWKAAKAALKAVSAAAKH	26	0.99	2796.3	1.875	1.875	1.875	3.75	15	7.5
AMP23	KIKEKLRQKLGNIMEKHLHKLAHRILLKK	29	0.98	3545.4	3.75	15	7.5	60	30	15

References

1. Linsky, T.W., Vergara, R., Codina, N., Nelson, J.W., Walker, M.J., Su, W., Barnes, C.O., Hsiang, T.-Y., Esser-Nobis, K., and Yu, K. (2020). De novo design of potent and resilient hACE2 decoys to neutralize SARS-CoV-2. Science *370*, 1208-1214.

2. Sesterhenn, F., Yang, C., Bonet, J., Cramer, J.T., Wen, X., Wang, Y., Chiang, C.-I., Abriata, L.A., Kucharska, I., and Castoro, G. (2020). De novo protein design enables the precise induction of RSV-neutralizing antibodies. Science *368*, eaay5051.

3. Pan, X., and Kortemme, T. (2021). Recent advances in de novo protein design: Principles, methods, and applications. Journal of Biological Chemistry *296*.

4. Quijano-Rubio, A., Yeh, H.-W., Park, J., Lee, H., Langan, R.A., Boyken, S.E., Lajoie, M.J., Cao, L., Chow, C.M., and Miranda, M.C. (2021). De novo design of modular and tunable protein biosensors. Nature *591*, 482-487.

5. Liu, K., Zhang, Y., Liu, K., Zhao, Y., Gao, B., Tao, X., Zhao, M., Wang, F.-Q., and Wei, D. (2022). De novo design of a transcription factor for a progesterone biosensor. Biosensors and Bioelectronics *203*, 113897.

6. Jackson, C., Anderson, A., and Alexandrov, K. (2022). The present and the future of protein biosensor engineering. Current Opinion in Structural Biology *75*, 102424.

7. Madhavan, A., Arun, K., Binod, P., Sirohi, R., Tarafdar, A., Reshmy, R., Awasthi, M.K., and Sindhu, R. (2021). Design of novel enzyme biocatalysts for industrial bioprocess: Harnessing the power of protein engineering, high throughput screening and synthetic biology. Bioresource Technology *325*, 124617.

8. Ennist, N.M., Zhao, Z., Stayrook, S.E., Discher, B.M., Dutton, P.L., and Moser, C.C. (2022). De novo protein design of photochemical reaction centers. Nature Communications *13*, 4937.

9. Currin, A., Swainston, N., Day, P.J., and Kell, D.B. (2015). Synthetic biology for the directed evolution of protein biocatalysts: navigating sequence space intelligently. Chemical Society Reviews *44*, 1172-1239.

10. Ferruz, N., and Höcker, B. (2022). Controllable protein design with language models. Nature Machine Intelligence *4*, 521-532.

11. Ferruz, N., Heinzinger, M., Akdel, M., Goncearenco, A., Naef, L., and Dallago, C. (2022). From sequence to function through structure: deep learning for protein design. Computational and Structural Biotechnology Journal.

12. Madani, A., Krause, B., Greene, E.R., Subramanian, S., Mohr, B.P., Holton, J.M., Olmos Jr, J.L., Xiong, C., Sun, Z.Z., and Socher, R. (2023). Large language models

generate functional protein sequences across diverse families. Nature Biotechnology, 1-8.

13. Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I., and Roth, D. (2021). Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys.

14. Budzianowski, P., and Vulić, I. (2019). Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems. arXiv preprint arXiv:1907.05774.

15. Alexandr, N., Irina, O., Tatyana, K., Inessa, K., and Arina, P. (2021). Fine-tuning gpt-3 for russian text summarization. (Springer), pp. 748-757.

16. Liao, Y., Wang, Y., Liu, Q., and Jiang, X. (2019). Gpt-based generation for classical chinese poetry. arXiv preprint arXiv:1907.00151.

17. Wang, Y., Deng, J., Sun, A., and Meng, X. (2022). Perplexity from PLM Is Unreliable for Evaluating Text Quality. arXiv preprint arXiv:2210.05892.

18. Newman, D., Noh, Y., Talley, E., Karimi, S., and Baldwin, T. (2010). Evaluating topic models for digital libraries. pp. 215-224.

19. Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J., and Blei, D. (2009). Reading tea leaves: How humans interpret topic models. Advances in neural information processing systems *22*.

20. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., and Abuajwa, W. (2021). Expanding functional protein sequence spaces using generative adversarial networks. Nature Machine Intelligence *3*, 324-333.

21. Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R.J., Milles, L.F., Wicky, B.I., Courbet, A., de Haas, R.J., and Bethel, N. (2022). Robust deep learning–based protein sequence design using ProteinMPNN. Science *378*, 49-56.

22. Yu, T., Cui, H., Li, J.C., Luo, Y., Jiang, G., and Zhao, H. (2023). Enzyme function prediction using contrastive learning. Science *379*, 1358-1363.

23. Nivón, L.G., Moretti, R., and Baker, D. (2013). A Pareto-optimal refinement method for protein design scaffolds. PloS one *8*, e59004.

24. Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. (2021). Language models enable zero-shot prediction of the effects of mutations on protein function. Advances in Neural Information Processing Systems *34*, 29287-29303.

25. Yang, K.K., Fusi, N., and Lu, A.X. (2022). Convolutions are competitive with transformers for protein sequence pretraining. bioRxiv, 2022.2005. 2019.492714.

26. Yang, K.K., Zanichelli, N., and Yeh, H. (2022). Masked inverse folding with sequence transfer for protein representation learning. bioRxiv, 2022.2005. 2025.493516.

27. Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. (2022). Learning inverse folding from millions of predicted structures. (PMLR), pp. 8946-8970.

28. Johnson, S.R., Fu, X., Viknander, S., Goldin, C., Monaco, S., Zelezniak, A., and Yang, K.K. (2023). Computational Scoring and Experimental Evaluation of Enzymes Generated by Neural Networks. bioRxiv, 2023.2003. 2004.531015.

29. Villegas-Morcillo, A., Makrodimitris, S., van Ham, R.C., Gomez, A.M., Sanchez, V., and Reinders, M.J. (2021). Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. Bioinformatics *37*, 162-170.

30. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F., and Rost, B. (2019). Modeling aspects of the language of life through transfer-learning protein sequences. BMC bioinformatics *20*, 1-17.

31. Jha, K., Saha, S., and Singh, H. (2022). Prediction of protein–protein interaction using graph neural networks. Scientific Reports *12*, 8360.

32. Singh, S., Chaudhary, K., Dhanda, S.K., Bhalla, S., Usmani, S.S., Gautam, A., Tuknait, A., Agrawal, P., Mathur, D., and Raghava, G.P. (2016). SATPdb: a database of structurally annotated therapeutic peptides. Nucleic acids research *44*, D1119-D1126.

33. Consortium, T.U. (2021). UniProt: the universal protein knowledgebase in 2021. Nucleic acids research *49*, D480-D489.

34. McDonald, A.G., and Tipton, K.F. (2021). Enzyme nomenclature and classification: The state of the art. The FEBS Journal.

35. Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics *22*, 1658-1659.

36. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., and Funtowicz, M. (2019). Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

37. Elnaggar, A., Heinzinger, M., Dallago, C., Rihawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., and Steinegger, M. (2020). ProtTrans: towards cracking the language of Life's code through self-supervised deep learning and high performance computing. arXiv preprint arXiv:2007.06225.

38. Chollet, F. (2015). Keras: Deep learning library for theano and tensorflow. URL: https://keras. io/k *7*, T1.

39. Agarap, A.F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

40. Rasamoelina, A.D., Adjailia, F., and Sinčák, P. (2020). A review of activation function for artificial neural network. (IEEE), pp. 281-286.

41. Kingma, D.P., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

42. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., and Sonnhammer, E.L. (2004). The Pfam protein families database. Nucleic acids research *32*, D138-D141.

43. Eddy, S.R. (2011). Accelerated profile HMM searches. PLoS computational biology *7*, e1002195.

44. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T.L. (2009). BLAST+: architecture and applications. BMC bioinformatics *10*, 1-9.

45. Mahlich, Y., Steinegger, M., Rost, B., and Bromberg, Y. (2018). HFSP: high speed homology-driven function annotation of proteins. Bioinformatics *34*, i304-i312.

46. Hinton, G.E., and Roweis, S. (2002). Stochastic neighbor embedding. Advances in neural information processing systems *15*.

47. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., and Dubourg, V. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research *12*, 2825-2830.

48. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., and Bourne, P.E. (2000). The protein data bank. Nucleic acids research *28*, 235-242.

49. Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., and Söding, J. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. Molecular systems biology *7*, 539.

50. Waterhouse, A.M., Procter, J.B., Martin, D.M., Clamp, M., and Barton, G.J. (2009). Jalview Version 2—a multiple sequence alignment editor and analysis workbench. Bioinformatics *25*, 1189-1191.

51. DeLano, W.L. (2002). Pymol: An open-source molecular graphics tool. CCP4 Newsl. Protein Crystallogr *40*, 82-92.

52. Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. BMC bioinformatics *12*, 1-8.

53. Xu, J., Li, F., Leier, A., Xiang, D., Shen, H.-H., Marquez Lago, T.T., Li, J., Yu, D.-J., and Song, J. (2021). Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides. Briefings in Bioinformatics *22*, bbab083.

54. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

55. Ma, Y., Guo, Z., Xia, B., Zhang, Y., Liu, X., Yu, Y., Tang, N., Tong, X., Wang, M., and Ye, X. (2022). Identification of antimicrobial peptides from the human gut microbiome using deep learning. Nature Biotechnology *40*, 921-931.

56. Nagarajan, D., Nagarajan, T., Roy, N., Kulkarni, O., Ravichandran, S., Mishra, M., Chakravortty, D., and Chandra, N. (2018). Computational antimicrobial peptide design and evaluation against multidrug-resistant clinical isolates of bacteria. Journal of Biological Chemistry *293*, 3492-3509.

57. Finn, R.D., Coggill, P., Eberhardt, R.Y., Eddy, S.R., Mistry, J., Mitchell, A.L., Potter, S.C., Punta, M., Qureshi, M., and Sangrador-Vegas, A. (2015). The Pfam protein families database: towards a more sustainable future. Nucleic acids research *44*, D279-D285.

58. Roope, L.S., Smith, R.D., Pouwels, K.B., Buchanan, J., Abel, L., Eibich, P., Butler, C.C., Tan, P.S., Walker, A.S., and Robotham, J.V. (2019). The challenge of antimicrobial resistance: what economics can contribute. Science *364*, eaau4679.

59. Bahar, A.A., and Ren, D. (2013). Antimicrobial peptides. Pharmaceuticals *6*, 1543-1575.

60. Chen, X., Li, C., Bernards, M.T., Shi, Y., Shao, Q., and He, Y. (2021). Sequence-based peptide identification, generation, and property prediction with deep learning: A review. Molecular Systems Design & Engineering *6*, 406-428.

61. Yan, J., Cai, J., Zhang, B., Wang, Y., Wong, D.F., and Siu, S.W. (2022). Recent Progress in the Discovery and Design of Antimicrobial Peptides Using Traditional Machine Learning and Deep Learning. Antibiotics *11*, 1451.

62. Ramazi, S., Mohammadi, N., Allahverdi, A., Khalili, E., and Abdolmaleki, P. (2022). A review on antimicrobial peptides databases and the computational tools. Database *2022*.

63. Capecchi, A., Cai, X., Personne, H., Köhler, T., Van Delden, C., and Reymond, J.-L. (2021). Machine learning designs non-hemolytic antimicrobial peptides. Chemical science *12*, 9221-9232.

64. Dean, S.N., Alvarez, J.A.E., Zabetakis, D., Walper, S.A., and Malanoski, A.P. (2021). PepVAE: variational autoencoder framework for antimicrobial peptide generation and activity prediction. Frontiers in Microbiology *12*, 725727.

65. Das, P., Sercu, T., Wadhawan, K., Padhi, I., Gehrmann, S., Cipcigan, F., Chenthamarakshan, V., Strobelt, H., Dos Santos, C., and Chen, P.-Y. (2021). Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. Nature Biomedical Engineering *5*, 613-623.

66. Tucs, A., Tran, D.P., Yumoto, A., Ito, Y., Uzawa, T., and Tsuda, K. (2020). Generating ampicillin-level antimicrobial peptides with activity-aware generative adversarial networks. ACS omega *5*, 22847-22851.

67. Jang, W.D., Kim, G.B., Kim, Y., and Lee, S.Y. (2022). Applications of artificial intelligence to enzyme and pathway design for metabolic engineering. Current Opinion in Biotechnology *73*, 101-107.

68. Pleiss, J. (2011). Protein design in metabolic engineering and synthetic biology. Current opinion in biotechnology *22*, 611-617.

69. Khurana, S., Rawi, R., Kunji, K., Chuang, G.-Y., Bensmail, H., and Mall, R. (2018). DeepSol: a deep learning framework for sequence-based protein solubility prediction. Bioinformatics *34*, 2605-2613.

70. Han, X., Wang, X., and Zhou, K. (2019). Develop machine learning-based regression predictive models for engineering protein solubility. Bioinformatics *35*, 4640-4646.

71. Li, F., Yuan, L., Lu, H., Li, G., Chen, Y., Engqvist, M.K., Kerkhoven, E.J., and Nielsen, J. (2022). Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. Nature Catalysis *5*, 662-672.

72. Yu, H., and Luo, X. (2022). Pretrained language models and weight redistribution achieve precise kcat prediction. bioRxiv, 2022.2011. 2023.517595.

73. Hu, X., Feng, C., Ling, T., and Chen, M. (2022). Deep learning frameworks for protein-protein interaction prediction. Computational and Structural Biotechnology Journal.

74. Casadio, R., Martelli, P.L., and Savojardo, C. (2022). Machine learning solutions for predicting protein–protein interactions. Wiley Interdisciplinary Reviews: Computational Molecular Science *12*, e1618.

75. Jiang, L., Althoff, E.A., Clemente, F.R., Doyle, L., Rothlisberger, D., Zanghellini, A., Gallaher, J.L., Betker, J.L., Tanaka, F., and Barbas III, C.F. (2008). De novo computational design of retro-aldol enzymes. science *319*, 1387-1391.

76. Siegel, J.B., Zanghellini, A., Lovick, H.M., Kiss, G., Lambert, A.R., St. Clair, J.L., Gallaher, J.L., Hilvert, D., Gelb, M.H., and Stoddard, B.L. (2010). Computational design of an enzyme catalyst for a stereoselective bimolecular Diels-Alder reaction. Science *329*, 309-313.