PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Deep self-supervised learning for biosynthetic gene cluster detection and product classification

Carolina Rios-Martinez[1,2], Nicholas Bhattacharya[1,3], Ava P. Amini[1], Lorin Crawford[1], Kevin K. Yang[1]*

1 Microsoft Research New England, Cambridge, Massachusetts, United States of America, **2** Department of Bioengineering, Stanford University, Stanford, California, United States of America, **3** Department of Mathematics, University of California, Berkeley, Berkeley, California, United States of America

* yang.kevin@microsoft.com

Citation: Rios-Martinez C, Bhattacharya N, Amini AP, Crawford L, Yang KK (2023) Deep self-supervised learning for biosynthetic gene cluster detection and product classification. PLoS Comput Biol 19(5): e1011162. https://doi.org/10.1371/journal.pcbi.1011162

Editor: Shihua Zhang, Academy of Mathematics and Systems Science, Chinese Academy of Science, CHINA

Received: July 24, 2022

Accepted: May 7, 2023

Published: May 23, 2023

Data Availability Statement: All relevant data is available on Zenodo at https://doi.org/10.5281/zenodo.6857704. Code is available at https://github.com/microsoft/protein-sequence-models and https://github.com/microsoft/bigcarp.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Natural products are chemical compounds that form the basis of many therapeutics used in the pharmaceutical industry. In microbes, natural products are synthesized by groups of colocalized genes called biosynthetic gene clusters (BGCs). With advances in high-throughput sequencing, there has been an increase of complete microbial isolate genomes and metagenomes, from which a vast number of BGCs are undiscovered. Here, we introduce a self-supervised learning approach designed to identify and characterize BGCs from such data. To do this, we represent BGCs as chains of functional protein domains and train a masked language model on these domains. We assess the ability of our approach to detect BGCs and characterize BGC properties in bacterial genomes. We also demonstrate that our model can learn meaningful representations of BGCs and their constituent domains, detect BGCs in microbial genomes, and predict BGC product classes. These results highlight self-supervised neural networks as a promising framework for improving BGC prediction and classification.

Author summary

Biosynthetic gene clusters (BGCs) encode for natural products of diverse chemical structures and function, but they are often difficult to discover and characterize. Many bioinformatic and deep learning approaches have leveraged the abundance of genomic data to recognize BGCs in bacterial genomes. However, the characterization of BGC properties remains the main bottleneck in identifying novel BGCs and their natural products. In this paper, we present a self-supervised masked language model that learns meaningful representations of BGCs with improved downstream detection and classification.

Introduction

Natural products are chemical compounds that form the basis of many pharmaceuticals and clinical therapeutics [1]. Their chemical structures are used in the development of

antimicrobial drugs, anticancer therapies, and other therapeutic areas [2]. To initiate the discovery of natural products, the pharmaceutical industry has traditionally relied on laboratory research, yet this approach cannot feasibly capture the entire chemical diversity of natural products. Thus, new methods are needed to advance natural product discovery [3].

Diverse natural products can be produced in living organisms via groups of genes called biosynthetic gene clusters (BGCs). Genome mining has become a powerful tool for exploring the complex and diverse chemical space of natural products [3]. Fast, inexpensive genome sequencing technology has contributed to the advancement of BGC identification and, by extension, natural product discovery. This approach has been particularly successful in microbes, where BGCs are often a group of physically colocalized genes whose sequence and function dictates the synthesis of natural products. This discovery of BGCs supports the assembly-line enzymology model, where biosynthetic systems are multimodular and each module contains a set of domains that collectively catalyze one round of elongation and chemical modification of the growing natural product peptide chain [4]. This type of natural product synthesis is particularly characteristic of multimodular polyketide synthases (PKS) and non-ribosomal peptide synthases (NRPS), which are two major biosynthetic systems that synthesize polyketides and non-ribosomal natural product peptides, respectively [5]. However, evidence suggests that much of the biosynthetic capacity of the microbial world remains unexplored [6]. Improved identification and characterization of BGCs directly from genomic data could accelerate the discovery of novel natural products with therapeutic relevance.

Identification of BGCs directly from genomic sequences is critical to navigating natural product space and nominating novel natural products. While complementary data modalities involving joint genome sequencing and mass-spectrometry data can be used to link products with gene clusters [7], the majority of known BGCs were characterized directly from DNA sequencing performed without any associated analysis of chemical structures in the sample. As such, computational methods which focus exclusively on identifying BGCs from genomes are essential components of BGC discovery pipelines.

antiSMASH (ANTIbiotics & Secondary Metabolite Analysis SHell) is an early tool for BGC discovery that uses a set of curated profile-Hidden Markov Models (pHMMs) to call biosynthetic gene families and a set of heuristics to tag a portion of a genome as a BGC [8, 9]. antiSMASH then annotates these called BGCs by using carefully curated rules based on expert knowledge. Similarly, ClusterFinder uses a Hidden Markov Model (HMM) to identify gene clusters of known and unknown classes [10]. Despite their effectiveness, HMM-based algorithms do not capture higher-order dependencies between genes, limiting their accuracy and generalizability [11]. Likewise, rule-based methods are limited by the need for human expertise and do not generalize well to new BGC classes.

A recent approach, DeepBGC, introduced a deep learning genome-mining strategy for biosynthetic gene cluster annotation that addresses these limitations [12]. Similar to antiSMASH, DeepBGC uses sets of curated pHMMs to call biosynthetic gene families; however, it uses a supervised neural network to predict BGC boundaries and annotate BGC function. Specifically, they employ a bidirectional long short-term memory (Bi-LSTM) recurrent neural network (RNN), which offers the advantage of capturing short- and long-term dependencies between adjacent and distant genes [13]. DeepBGC reported promising improvements in the identification of BGCs in microbial genomes. However, DeepBGC is trained on a small number of high-quality annotations, and the supervised approach requires mining examples of genes that are not part of BGCs. The quality of the predictions is therefore likely to depend on the quality of the negative examples, which must be similar to BGC sequences while ideally containing no false negatives.

Rather than relying on expert-curated annotations and negative examples, self-supervised masked language models promise the ability to learn biologically-relevant patterns directly from a large set of BGC examples. Recently, self-supervised masked language models of biological sequences have been used to study proteins [14–21], DNA [22], RNA [23, 24], and glycans [25, 26]. In these models, a neural network is trained either to reconstruct the original sequence from a corrupted version of the sequence, or to predict the next element in the sequence given the preceding elements. After training on a large dataset, such as all protein sequences in UniProt [27], the model can be used for zero-shot predictions of fitness [28] or structure [29], and can additionally be fine-tuned on downstream supervised tasks [30, 31].

To accelerate identification and classification of BGCs, we developed a self-supervised neural network masked language model of BGCs from bacterial genomes (Fig 1). Our model represents BGCs as chains of functional protein domains, and uses ESM-1b [14], a protein masked language model, to obtain pretrained embeddings of functional protein domains with amino acid-level context. We then train a convolutional masked language model on these domains to develop meaningful learned representations of BGCs and their constituent domains. The architecture for our model is based off of convolutional autoencoding representations of proteins (CARP) [32], a masked language model of proteins, and we will therefore refer to it as **Bi**osynthetic **G**ene **CARP** (BiGCARP). We leverage these representations to detect BGCs from microbial genomes and then classify them based on their natural product class. We further investigate the potential advantages of our model by comparing our approach with DeepBGC, and demonstrate that BiGCARP achieves improvements in BGC prediction and natural product classification. BiGCARP highlights self-supervised neural networks as a promising framework for improving BGC characterization.

Results

Self-supervised training

We first developed a self-supervised training scheme to train BiGCARP to learn representations of BGCs. As BGCs have a hierarchical structure, they can be represented at four main levels. From the least-to-most granular, these are: genes, Pfam domains (families of evolutionary-related proteins), amino acids, and nucleotides. We note that more granular units of representation lead to longer sequences. BGCs typically contain several dozen genes, each of which contains one or more Pfam domains. Each Pfam domain contains tens to hundreds of amino acids, and each amino acid is encoded by three nucleotides. This introduces a trade-off between modeling short sequences where each unit is complex or modeling long sequences where each unit is simple. In order to balance input sequence length and information content of individual units, we chose to represent BGCs as sequences of Pfams. This is the same level chosen by DeepBGC [12]. As shown in Fig 1, during training, we append a BGC product class token to the start of each BGC Pfam sequence in order to learn BGC product classes from their Pfam domain sequences. We then corrupt the sequence according to the BERT [33] corruption scheme and train **Bi**osynthetic **G**ene **C**onvolutional Autoencoding **R**epresentations of **P**roteins (BiGCARP) to reconstruct the original class token and Pfam sequence. BiGCARP combines the ByteNet encoder dilated CNN architecture from [34] with linear input embedding and output decoding layers, as shown in Fig 2a.

Pfam embeddings map protein families from our vocabulary to vectors into a 1280-dimensional space, and thus serve as the inputs to BiGCARP. We train three versions of BiGCARP with different initial Pfam embeddings. The BiGCARP-ESM-1b-finetuned and BiGCARP-ESM-1b-frozen models are both initialized with Pfam embeddings obtained by averaging the per-residue output from ESM-1b for each domain. BiGCARP-ESM-1b-finetuned has its

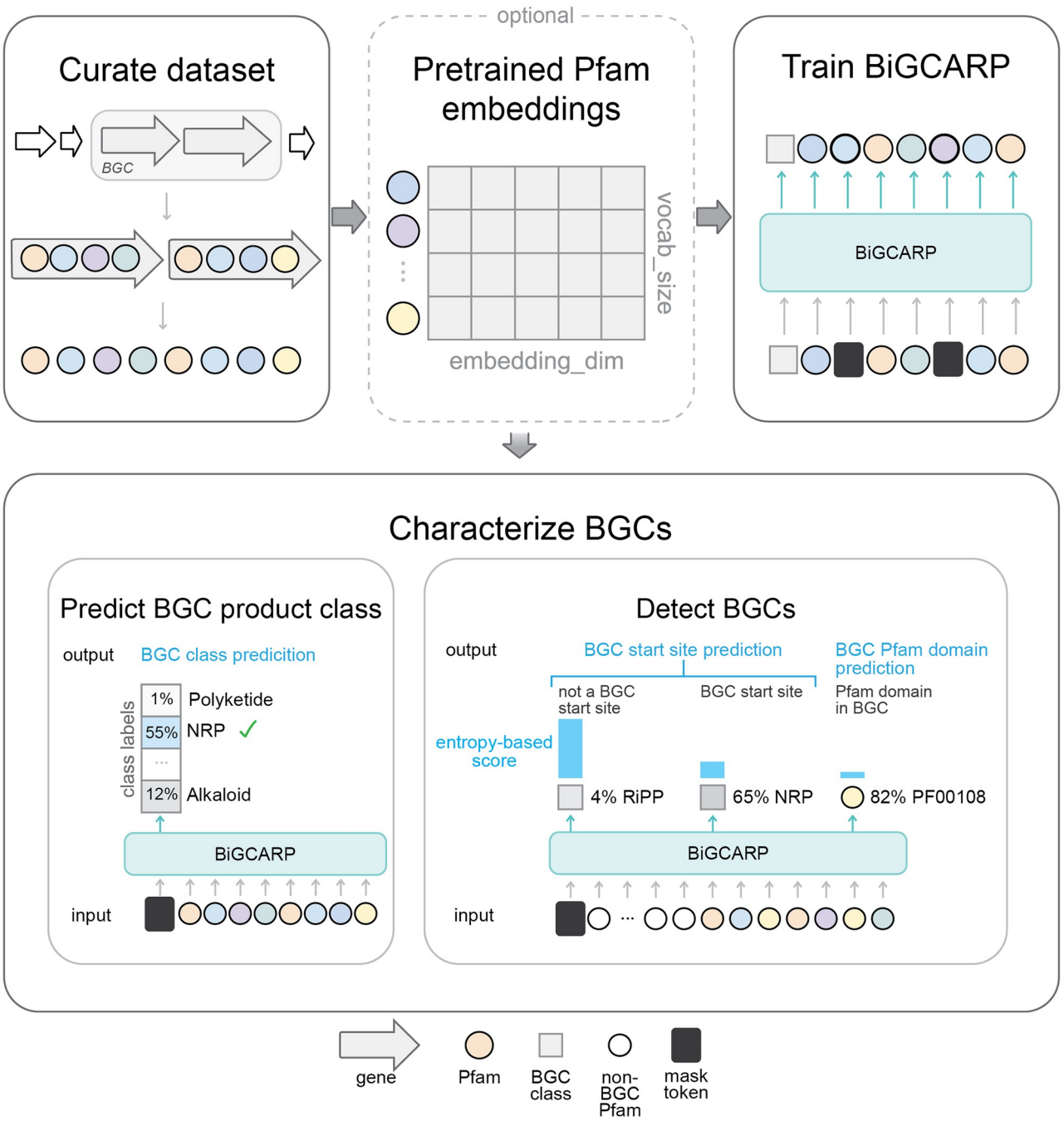

Fig 1. Self-supervised deep learning workflow for characterizing biosynthetic gene cluster (BGC) properties. Schematic of the workflow for characterizing BGCs with BiGCARP, a self-supervised deep neural network. We curate a dataset of annotated BGCs from antiSMASH for training BiGCARP. We then use ESM-1b [14], a protein masked language model, to obtain pretrained embeddings of protein family (Pfam) domains in our dataset and to explore whether pretrained Pfam domain embeddings show improvement on the quality of their representations. By representing BGCs as chains of Pfams, we train a self-supervised masked language model on these domains to characterize BGC properties in microbial genomes. We leverage these learned representations to detect BGCs from microbial genomes and to predict their natural product class.

https://doi.org/10.1371/journal.pcbi.1011162.g001

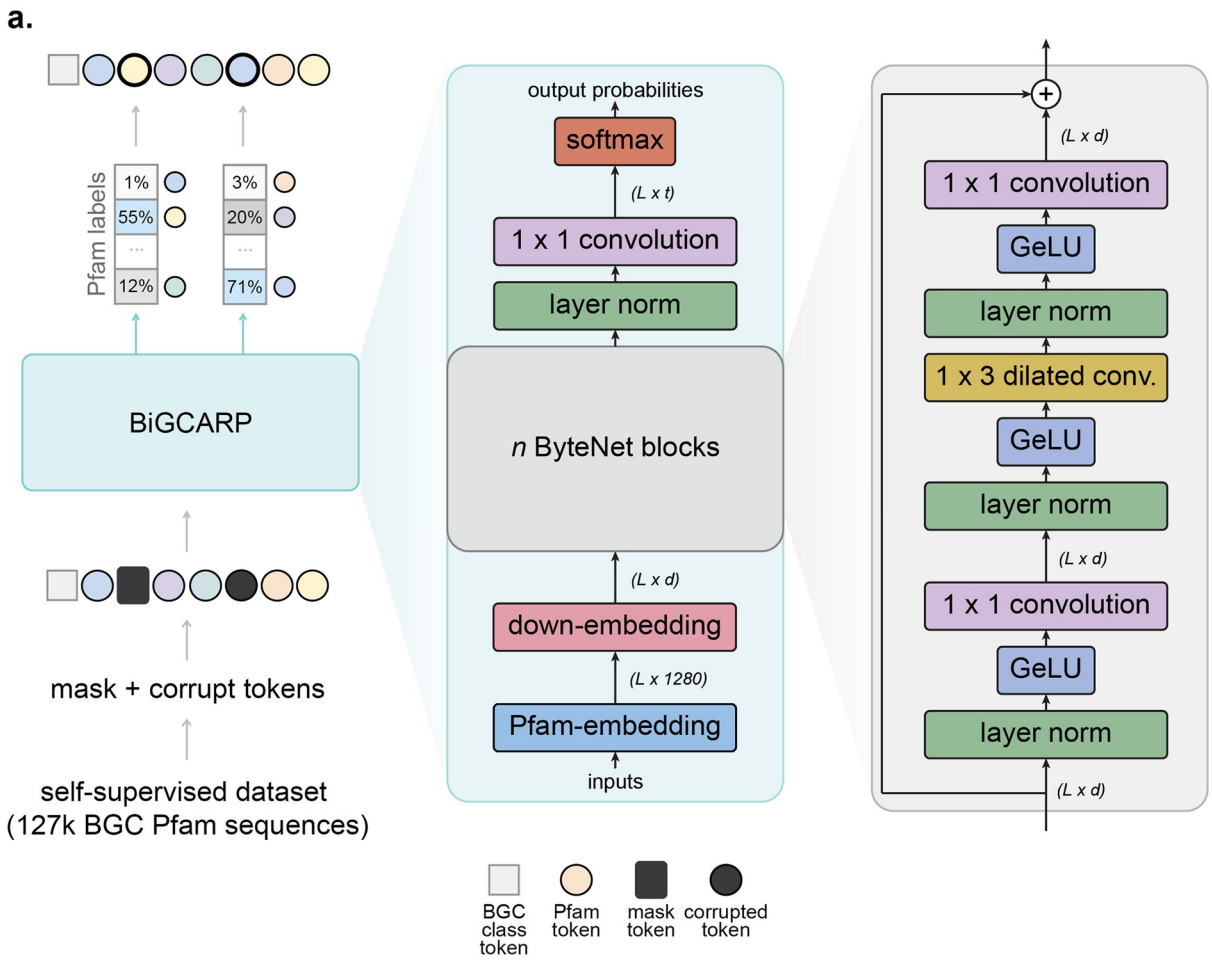

Fig 2. BiGCARP architecture with validation performance curves on the self-supervised dataset. (a) We use the masked language model objective described in [33] to train BiGCARP to reconstruct the BGC product class and Pfam sequence on our self-supervised dataset, which contains around 127,000 BGC Pfam sequences. BiGCARP is a dilated 1D-convolutional neural network masked language model based on CARP [32] and ByteNet [34]. (b) Validation loss (cross-entropy) and accuracy for BiGCARP with different initial Pfam embeddings.

https://doi.org/10.1371/journal.pcbi.1011162.g002

embeddings finetuned during self-supervised BGC training, while BiGCARP-ESM-1b-frozen has the initial embeddings frozen at the onset of self-supervised BGC training. Finally, BiGCARP-random is initialized with a random Pfam embedding, which is finetuned during self-supervised BGC training. All three versions of BiGCARP are trained on BGC sequences extracted from the antiSMASH dataset [8, 9]. We used approximately 127,000 BGC sequences and split the dataset 80/10/10 between training, validation, and testing, respectively. The training set is deduplicated against all datasets used in downstream evaluation. We refer the reader to Materials and Methods for details about the model training and architecture and the self-supervised training dataset.

Fig 2b plots the learning curves of the validation performance on the self-supervised dataset for all three versions of BiGCARP. We discover BiGCARP-ESM-1b-frozen is outperformed by BiGCARP-ESM-1b-finetuned and BiGCARP-random, which both show similar performance and attain an accuracy of around 75%.

Learned embeddings encode relevant representations of Pfam domains

We used uniform manifold approximation and projection (UMAP) to visualize the input Pfam embeddings after self-supervised training on the antiSMASH training set (Fig 3). Each protein family is represented as a single point, and protein families of similar sequence and function should have similar representations and thus be mapped to nearby points. In order to determine if our embeddings capture these properties of related Pfam domains, we plot every Pfam domain that falls under the ten most common Pfam superfamilies (clans) in our self-supervised dataset: NADP Rossman (CL0063), P-loop NTPase (CL0023), Zn Beta Ribbon (CL0167), E-set (CL0159), HTH (CL0123), TPR (CL0020), PDDEXK (CL0236), MBB (CL0193), Beta propeller (CL0186), and OB (CL0021) [35].

We find that initializing Pfam domain embeddings using ESM-1b improves the quality of the learned representations, as these embeddings take into account protein family amino-acid sequence and protein structural information. Fig 3 indicates BiGCARP-ESM-1b and BiGCARP-ESM-1b frozen embeddings form clear clusters of structurally related Pfam domains, and we find that the Pfam domains close in representation space have similar protein structure rather than amino acid residue sequence, as shown in S1 Fig. Randomly initialized Pfam embeddings shows minimal interpretable information after self-supervised BGC training;

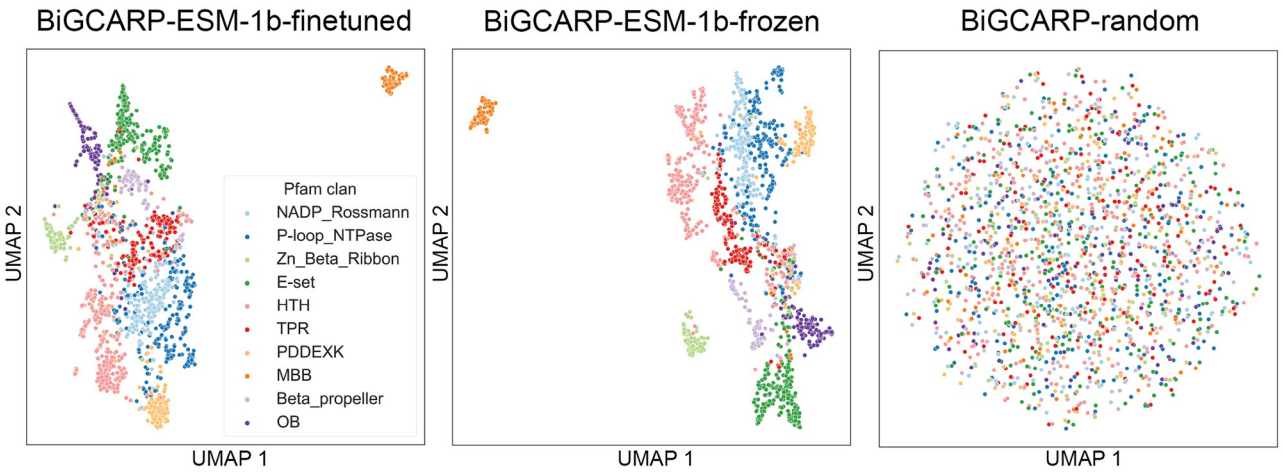

Fig 3. Relevant representations of Pfam domains are encoded in learned ESM-1b embeddings. Uniform manifold approximation and projection (UMAP) visualization of learned representations of Pfam domains from BiGCARP with different initial Pfam embeddings.

https://doi.org/10.1371/journal.pcbi.1011162.g003

Table 1. Pretraining results, including the exponentiated cross entropy (ECE) metric on the pretraining test set and area under the receiver operating characteristic curve (AUROC) for BGC start locations and domains on the 9-genomes validation set.

		BiGCARP		
		ESM-1b-finetuned	ESM-1b-frozen	random
pretrain test set (ECE)	Pfam domain	**4.64**	4.99	4.67
	product class	1.50	**1.46**	1.50
9-genomes (AUROC)	start	0.720	0.701	**0.723**
	domain	**0.876**	0.611	0.856

https://doi.org/10.1371/journal.pcbi.1011162.t001

however, the variance explained in the randomly initialized Pfam embeddings is likely more uniformly distributed over the 1280 dimensions, and thus Fig 3 may not be illustrating how much information is actually being captured over the entire randomly initialized Pfam embedding.

BiGCARP captures meaningful patterns in BGCs

We next evaluated BiGCARP's pretraining performance after self-supervised training (Table 1). We use the exponentiated cross entropy (ECE) metric for evaluating BiGCARP. This metric provides a measure for a model's ability to narrow its prediction of a token from the set of options. An ideal model would have an ECE of 1, whereas a model choosing at random would have an ECE of the vocabulary size, which in our case is 19,550 for Pfam domains and 55 for BGC product classes. On our antiSMASH dataset test set, BiGCARP-ESM-1b-finetuned achieves the lowest ECE on the Pfam domains, while BiGCARP-ESM-1b-frozen achieves the lowest ECE on the product classes despite performing worse on domain ECE.

In addition, using the 9-genomes validation set from DeepBGC [12], we evaluate whether BiGCARP can identify the start locations of BGCs and whether each domain is in a BGC without further supervised training. We append a mask token to the beginning of every window of 64 domains in the dataset and pass them through BiGCARP. Intuitively, if the window is the start of a BGC, the model's BGC class prediction should have low entropy, and its reconstructions of the domains should be both low-entropy and have low cross-entropy with the original input domain. This scheme is shown in Fig 1. We refer the reader to Materials and Methods for details about scoring start positions and BGC Pfam domains. As shown in Table 1, all three versions of BiGCARP can detect BGC start locations and whether domains are part of a BGC, with BiGCARP-ESM-1b-frozen performing worse on both tasks than the other two versions.

We then finetuned BiGCARP on the training dataset reported in DeepBGC v0.1.0 [12], which uses all BGC domain sequences from MiBIG (version 1.4) as positive BGC samples and 10128 negative examples for 100 epochs and choose the epoch with the highest area under the receiver operating characteristic curve (AUROC) on the 9-genomes validation set for further testing (Methods). Table 2 shows domain-level classification performance using AUROC on

Table 2. Domain AUROC and average precision after supervised training on the DeepBGC training set.

pretraining	6 genomes		9 genomes
	AUROC	AvgPrec	AUROC
BiGCARP ESM-1b finetuned	0.941	0.447	0.950
BiGCARP ESM-1b frozen	0.940	0.405	0.949
BiGCARP random	0.936	0.435	0.943
none	0.937	0.429	0.950
DeepBGC	0.921	0.398	0.934

https://doi.org/10.1371/journal.pcbi.1011162.t002

the 9-genomes validation set and AUROC and average precision (AvgPrec) on the 6-genomes test set from DeepBGC. Note that the DeepBGC results on 9-genomes are for cross-validation directly on 9-genomes. All three versions of BiGCARP outperform DeepBGC on the 6-genomes test set and 9-genomes validation set. However, self-supervised training did not improve performance on the 6-genomes test set for BiGCARP.

BiGCARP predicts BGC product classes

In addition to detecting BGCs in microbial genomes, predicting their product classes would provide further aid in discovering new natural products. BiGCARP learns to predict a BGC's product class from its Pfam sequence by reconstructing masked class tokens during self-supervised training (Fig 1). During self-supervised training, we use the antiSMASH product classes. In order to compare BiGCARP's performance to DeepBGC, we map antiSMASH product classes to those in the Minimum Information about a Biosynthetic Gene cluster (MIBiG) dataset used in DeepBGC [12, 36]. DeepBGC trains a random forest classifier on its embeddings to predict BGC product classes. In contrast, we simply append a mask token to the beginning of each BGC sequence and evaluate the model's predictions for the identity of the mask, removing the need to train an additional model.

All three versions of BiGCARP out-perform DeepBGC on average AUROC across the product classes, and ensembling their predictions via their arithmetic mean further improves accuracy, as shown in Table 3 and S1 Table. BiGCARP-ensemble outperforms DeepBGC on AUROC for four out of seven product classes. This is likely because the antiSMASH training set is approximately 100-times larger than MIBiG. Performance is generally similar for product classes that are well-represented in both datasets, with the largest gains coming in the "other" and alkaloid classes, which are under-represented in MIBiG. This underscores the importance and utility of training on a large and diverse BGC dataset. However, BiGCARP does not do as well as DeepBGC on precision and recall. This is likely because directly training on MIBiG labels enables better calibrated predictions. We note that DeepBGC is further advantaged here by reporting 5-fold cross-validation results on MIBiG, while BiGCARP is not trained on any sequences from MIBiG.

BiGCARP identifies BCGs from unannotated microbial genomes

We compared the ability of BiGCARP and antiSMASH 6.1.1 [37] to identify BGCs in 773 randomly-chosen bacterial genomes released after the antiSMASH 3.0 database. antiSMASH 6.1.1 identified 4287 clusters (renamed 'regions' in antiSMASH 5.0 and later). Of these, we were able to match the Pfam domain sequences for 3174 clusters to those produced by our domain annotation pipeline (see Methods). We treat these 3174 clusters identified by antiSMASH as ground

Table 3. Product classification results on MIBiG.

		antiSMASH			BiGCARP ensemble			DeepBGC		
	n MIBiG	AUROC	precision	recall	AUROC	precision	recall	AUROC	precision	recall
polyketide	644	0.870	0.901	0.806	0.898	0.838	0.835	0.903	0.882	0.890
NRP	433	0.915	0.939	0.852	0.898	0.791	0.849	0.907	0.910	0.857
RiPP	199	0.897	0.958	0.799	0.963	0.616	0.875	0.907	0.931	0.824
saccharide	179	0.607	0.769	0.223	0.773	1.000	0.006	0.811	0.904	0.606
other	154	0.671	0.594	0.370	0.763	0.318	0.343	0.583	0.840	0.157
terpene	120	0.744	0.908	0.492	0.869	0.815	0.367	0.824	0.870	0.663
alkaloid	39	0.785	0.434	0.590	0.820	0.222	0.051	0.607	0.533	0.154

https://doi.org/10.1371/journal.pcbi.1011162.t003

Table 4. Area under the receiver operating characteristic curve (AUROC) for BGC start locations and domains on 773 bacterial genomes released after antiSMASH 3.0 database.

	BiGCARP		
	ESM-1b-finetuned	ESM-1b-frozen	random
Pfam domain	0.641	0.648	0.644
start	0.778	0.792	0.773

https://doi.org/10.1371/journal.pcbi.1011162.t004

truth labels against which we evaluate matched BiGCARP predictions. Table 4 shows unsupervised BiGCARP performance in predicting the locations of clusters in these genomes, using the same methods as on the 9 genomes test set. We also identify 199 possible start locations with better scores than any of the clusters found by antiSMASH 6.1.1, which may be a fruitful starting point for further investigation. All datasets and annotations used can be found on Zenodo.

Discussion

Biosynthetic gene clusters (BGCs) are a promising source of natural products, but are difficult to discover, express, and characterize. Recent work in self-supervised deep learning has shown promise for modeling DNA, RNA, proteins, and glycans. We develop **Bi**osynthetic **G**ene **C**onvolutional Autoencoding **R**epresentations of **P**roteins (BiGCARP), a masked language model that learns representations of BGCs based on their Pfam domains, detects BGCs, and predicts their product classes. To our knowledge, this is the first work to use Pfam domains as tokens in a masked language model. We demonstrate that our model learns biologically-reasonable representations of Pfam domains. Representing BGCs as Pfam domains was a compromise between limiting the sequence length while having fine-grained sequence information. Models on the level of amino acid residues or the nucleotide sequence may be able to resolve more details at the cost of more computation. BiGCARP is a strong BGC detector even without seeing negative examples, and achieves state-of-the-art accuracy in product class prediction.

In presenting the first use of Pfam domains as tokens for a self-supervised language model, this work opens opportunities for future method development and model refinement. For example, our results indicated minimal benefit to using ESM-1b pre-trained Pfam embeddings. Future work could evaluate whether complementary methods for protein sequence pretraining, such as the convolutional-based CARP model [32], confer greater benefit relative to language models like ESM-1b. Additionally, sensitivity analyses could assess how the predictive performance of BiGCARP and other competing methods change as a function of different BGC representations, such as amino acid residues or nucleotide sequences. The BGC masked language model introduced here additionally demonstrates promise for the expansion of BGC science and engineering. In natural language processing and protein engineering, masked language models are often fine-tuned on downstream tasks of interest.

For BGCs, these downstream tasks could include predicting their expression conditions or the chemical structures of their products. Future work to assess the performance benefit of fine-tuning BiGCARP will help determine the potential utility of BiGCARP for a variety of downstream tasks. Without fine-tuning, our models are useful for detecting previously unknown BGCs in microbial genomes and predicting BGC product classes.

In summary, we present BiGCARP, a self-supervised masked language model for the detection and characterization of biosynthetic gene clusters. The BiGCARP model described here could be deployed for downstream tasks of interest, including chemical product structure characterization and BGC mining. This study highlights the potential of self-supervised deep learning as a framework for BGC discovery and characterization.

Materials and methods

In this section, we elaborate on details of our self-supervised deep learning framework for detecting BGCs from bacterial genomes and classifying them into their natural product classes. The workflow is summarized in Fig 1, which consists of curating data, pretraining Pfam domain embeddings, training BiGCARP, and using BiGCARP to characterize BGCs.

Data

Pretraining dataset curation. To curate our pretraining dataset, we ran antiSMASH (ANTIbiotics & Secondary Metabolite Analysis SHell) 2.0, a microbial genome mining tool for BGC identification and analysis [8], on a database of 6,200 full bacterial genomes and 18,576 bacterial draft genomes [9]. This led to 142,821 total BGCs spanning 55 classes identified for model development and evaluation. Our choice of representing BGCs as Pfam domains led to a vocabulary size of 19,500 unique Pfam domains collected from Pfam database versions 31 and 32 [35]. We also remove sequences from the self-supervised training and validation sets that contain substrings from or are substrings of sequences from the MIBiG, 9-genomes, and 6-genomes datasets from DeepBGC described below. This results in 127,294 BGCs in our pretraining dataset prior to data splitting. All datasets used can be found on Zenodo. Future work may include using newer antiSMASH databases for pretraining for improvement of BiGCARP.

Pretraining data split for training and evaluation. Our training, validation, and test sets were produced from an 80/10/10 split of the total set. Note that random splitting of data is widely avoided in biological sequence modeling, since it leads to evaluation of overly simple generalization. For example, in protein modeling, one instead uses sequence-identity based splits as a proxy for evolutionary signal [38]. Proper splitting of BGCs is more complex, as evolution of BGCs is poorly understood. To reduce redundancy between the data splits, we ensured that no example in one set was a strict substring of an example in another set.

DeepBGC datasets for evaluating BGC detection and product classification. We evaluated the performance of our models and compared it to the DeepBGC model by testing its ability to detect BGCs within bacterial genomes and to predict their corresponding product classes. To do this, we utilized DeepBGC's training set with 617 positive and 10128 negative BGC samples to finetune our models [12]. We also used their 6-genomes and 9-genomes datasets to perform supervised domain classification tasks. For BGC product classification, we used DeepBGC's MIBiG dataset, which contains 1406 BGCs. Our mapping from antiSMASH product types to common MIBiG compound classes can be found on Zenodo.

Unannotated bacterial genomes for evaluating BiGCARP performance. To demonstrate the applicability of BiGCARP in advancing natural product discovery, we curated 773 unannotated bacterial genomes for BGC identification. We obtained all bacterial genomes with an assembly level of 'complete', 'chromosome', or 'scaffold' in GenBank and FASTA format released after 4 September 2020 available to download from the NCBI Datasets Genome Data Package using the ncbi-genome-download tool, yielding 108,007 assemblies that are unannotated in antiSMASH database version 3.0. We randomly chose 773 to analyze.

We then used Prodigal [39] version 2.6.3 with default parameters to predict open reading frames in all 773 bacterial genomes. Protein family domains were identified using HMMER [40] version 3.3.2 hmmsearch and Pfam database version 32 [35]. Hmmsearch tabular output was filtered using cath-resolve-hits [41] to obtain a final set of non-overlapping domain assignments. The resulting list of Pfam domains was sorted by the gene and the domain start location.

Embeddings of Pfam domains with ESM-1b

We represent each Pfam as a vector. To do this, we take the first sequence in the alignment for a Pfam, then use ESM-1b [14], a protein masked language model, to embed all amino acids of this sequence. We averaged the embeddings over the full sequence, yielding a representation vector of size 1280. By obtaining pretrained embeddings of Pfam domains with ESM-1b, our model takes into account sequence details. To explore whether pre-trained Pfam domain embeddings show improvement on the quality of Pfam domain representations, we use three different initial Pfam embeddings for BiGCARP: ESM-1b embeddings finetuned, ESM-1b embeddings frozen, and randomly initialized embeddings updated throughout training. ESM-1b-finetuned and ESM-1b-frozen have the same initialization at the start of self-supervised training. All other model weights were randomly initialized.

BiGCARP architecture and training

We train BiGCARP using the masked language model objective described in [33]. We prepend a token representing the antiSMASH BGC class to each BGC sequence. Each sequence is then corrupted by changing some tokens to a special mask token or another Pfam domain token, and the model is tasked with reconstructing the original sequence. Specifically, 15% of tokens from each sequence are randomly selected for supervision during each training step. For those 15% of tokens, 80% are replaced by the mask token, 10% are replaced by a randomly-chosen Pfam domain token, and 10% remain unchanged. The model is trained to minimize the batch average cross entropy loss between its predictions for the selected tokens and the true tokens at those locations.

BiGCARP is a dilated 1D-convolutional neural network masked language model based on ByteNet [34] and CARP [32]. The input is a sequence of Pfam domains represented by 1280-dimensional vectors. Model hyperparameters include the following: kernel width of 3, maximum dilation of 128, 32 layers, and a hidden dimension of 256 for a total of 34 million parameters. Training parameters include the following: batch size of 64, Adam optimizer with a learning rate of 10^{-4}, and mixed precision training using PyTorch [42] and NVIDIA Apex. Each version of BiGCARP was trained on one 32GB NVIDIA V100 GPU for 300 epochs. The epoch with the lowest validation loss was selected for downstream experiments. Model weights and datasets are available on Zenodo; training code is available on our BiGCARP repository and code to run pretrained BiGCARP models is available on our protein sequence models repository on Github, with a command line manual for how to use the models. We do not report replicates for results as that would require training each model from scratch multiple times.

Evaluation on 9-genomes and 6-genomes

We use the intuition that the model should make more confident predictions when given BGC sequences than non-BGC sequences to predict BGC start locations and whether each domain is part of a BGC. For each bacterial genome, we prepend a mask token to each possible subsequence of 64 domains and pass the resulting sequences to BiGCARP. With the exception of domains at the beginning and end of the genome, each domain is thus scored 64 times. For each window, we calculate the entropy of the predictions for the prepended mask token (start entropy), the entropy for each of the 64 domains in the window (domain entropy), and the negative log-likelihood of each domain in the window (negative log-likelihood). We predict whether a domain is the start of a BGC using the start entropy of the window for which it is the first domain; positions with a lower start entropy are more likely to be BGC start locations.

We predict whether each domain is part of a BGC using the average of the start entropies for every window in which it appears and its domain entropy and negative log-likelihood within each window in which it appears (a total of 64 × 3 values). Domains with lower scores are more likely to be within a BGC.

Supervised training on DeepBGC training set

We follow the supervised training procedure described in DeepBGC. Using the positive BGC domain sequences from MiBIG (version 1.4) and 10128 negative BGC domain sequences from DeepBGC, at each epoch, we shuffle the sequences into a "genome" and then predict whether each domain is part of a BGC. We fine-tune the self-supervised versions of BiGCARP as well as a randomly-initialized version using the Adam optimizer and a learning rate of 10^{-4} with early stopping using supervised results on 9-genomes.

Supporting information

S1 Fig. BiGCARP-ESM-1b-frozen Pfam embeddings demonstrate Pfam domain representation spaces are not explained by their underlying amino acid residue sequence. (a) Heatmap of Euclidean distances between domain embeddings in the following Pfam clans: NADP_Rossmann, P-loop NTPase, and MBB. (b) Average Euclidean distance between Pfam domain embeddings in the aforementioned Pfam clans (c) Pairwise sequence alignment percent identity matrix between Pfam clans.
(TIF)

S1 Table. Product classification results for individual BiGCARPs on MIBiG.
(XLSX)

Acknowledgments

This research was conducted using computational resources and services at Microsoft. We thank David Prihoda for assistance with the DeepBGC validation and test datasets and Jackson Cahn for inspiring discussions on BGCs.

Author Contributions

Conceptualization: Carolina Rios-Martinez, Nicholas Bhattacharya, Kevin K. Yang.

Data curation: Carolina Rios-Martinez, Nicholas Bhattacharya, Kevin K. Yang.

Formal analysis: Carolina Rios-Martinez, Nicholas Bhattacharya, Kevin K. Yang.

Methodology: Carolina Rios-Martinez, Nicholas Bhattacharya, Ava P. Amini, Lorin Crawford, Kevin K. Yang.

Project administration: Ava P. Amini, Lorin Crawford, Kevin K. Yang.

Software: Carolina Rios-Martinez, Kevin K. Yang.

Supervision: Ava P. Amini, Lorin Crawford, Kevin K. Yang.

Visualization: Carolina Rios-Martinez, Kevin K. Yang.

Writing – original draft: Carolina Rios-Martinez, Nicholas Bhattacharya, Kevin K. Yang.

Writing – review & editing: Ava P. Amini, Lorin Crawford.

References

1. Newman DJ, Cragg GM. Natural products as sources of new drugs from 1981 to 2014. Journal of natural products. 2016; 79(3):629–661. https://doi.org/10.1021/acs.jnatprod.5b01055 PMID: 26852623
2. Walsh CT, Tang Y. Natural product biosynthesis. Royal Society of Chemistry; 2017.
3. Ziemert N, Alanjary M, Weber T. The evolution of genome mining in microbes–a review. Natural product reports. 2016; 33(8):988–1005. https://doi.org/10.1039/C6NP00025H PMID: 27272205
4. Nivina A, Herrera Paredes S, Fraser HB, Khosla C. GRINS: Genetic elements that recode assembly-line polyketide synthases and accelerate their diversification. Proceedings of the National Academy of Sciences. 2021; 118(26):e2100751118. https://doi.org/10.1073/pnas.2100751118 PMID: 34162709
5. Chen R, Wong HL, Burns BP. New approaches to detect biosynthetic gene clusters in the environment. Medicines. 2019; 6(1):32. https://doi.org/10.3390/medicines6010032 PMID: 30823559
6. Pye CR, Bertin MJ, Lokey RS, Gerwick WH, Linington RG. Retrospective analysis of natural products provides insights for future discovery trends. Proceedings of the National Academy of Sciences. 2017; 114(22):5601–5606. https://doi.org/10.1073/pnas.1614680114 PMID: 28461474
7. Mohimani H, Liu WT, Kersten RD, Moore BS, Dorrestein PC, Pevzner PA. NRPquest: coupling mass spectrometry and genome mining for nonribosomal peptide discovery. Journal of natural products. 2014; 77(8):1902–1909. https://doi.org/10.1021/np500370c PMID: 25116163
8. Blin K, Medema MH, Kazempour D, Fischbach MA, Breitling R, Takano E, et al. antiSMASH 2.0—a versatile platform for genome mining of secondary metabolite producers. Nucleic acids research. 2013; 41 (W1):W204–W212. https://doi.org/10.1093/nar/gkt449 PMID: 23737449
9. Blin K, Pascal Andreu V, de los Santos ELC, Del Carratore F, Lee SY, Medema MH, et al. The antiSMASH database version 2: a comprehensive resource on secondary metabolite biosynthetic gene clusters. Nucleic acids research. 2019; 47(D1):D625–D630. https://doi.org/10.1093/nar/gky1060 PMID: 30395294
10. Cimermancic P, Medema MH, Claesen J, Kurita K, Brown LCW, Mavrommatis K, et al. Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. Cell. 2014; 158 (2):412–421. https://doi.org/10.1016/j.cell.2014.06.034 PMID: 25036635
11. Choo KH, Tong JC, Zhang L. Recent applications of hidden Markov models in computational biology. Genomics, proteomics & bioinformatics. 2004; 2(2):84–96. https://doi.org/10.1016/S1672-0229(04)02014-5 PMID: 15629048
12. Hannigan GD, Prihoda D, Palicka A, Soukup J, Klempir O, Rampula L, et al. A deep learning genome-mining strategy for biosynthetic gene cluster prediction. Nucleic acids research. 2019; 47(18):e110–e110. https://doi.org/10.1093/nar/gkz654 PMID: 31400112
13. Hochreiter S, Heusel M, Obermayer K. Fast model-based protein homology detection without alignment. Bioinformatics. 2007; 23(14):1728–1736. https://doi.org/10.1093/bioinformatics/btm247 PMID: 17488755
14. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proceedings of the National Academy of Sciences. 2021; 118(15). https://doi.org/10.1073/pnas.2016239118 PMID: 33876751
15. Madani A, McCann B, Naik N, Keskar NS, Anand N, Eguchi RR, et al. ProGen: Language Modeling for Protein Generation. arXiv. 2020;.
16. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning; 2021.
17. Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, et al. Deep neural language modeling enables functional protein generation across families. bioRxiv. 2021;.
18. Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. ProteinBERT: A universal deep-learning model of protein sequence and function. bioRxiv. 2021;.
19. Ferruz N, Schmidt S, Höcker B. A deep unsupervised language model for protein design. bioRxiv. 2022;.
20. Hesslow D, ed Zanichelli N, Notin P, Poli I, Marks DS. RITA: a Study on Scaling Up Generative Protein Sequence Models; 2022.
21. Nijkamp E, Ruffolo J, Weinstein EN, Naik N, Madani A. ProGen2: Exploring the Boundaries of Protein Language Models. arXiv preprint arXiv:220613517. 2022;.
22. Ji Y, Zhou Z, Liu H, Davuluri RV. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. Bioinformatics. 2021; 37(15):2112–2120. https://doi.org/10.1093/bioinformatics/btab083 PMID: 33538820
23. Akiyama M, Sakakibara Y. Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning. NAR genomics and bioinformatics. 2022; 4(1):lqac012. https://doi.org/10.1093/nargab/lqac012 PMID: 35211670

24. Chen J, Hu Z, Sun S, Tan Q, Wang Y, Yu Q, et al. Interpretable RNA foundation model from unannotated data for highly accurate RNA structure and function predictions. arXiv preprint arXiv:220400300. 2022;.

25. Bojar D, Camacho DM, Collins JJ. Using natural language processing to learn the grammar of glycans. bioRxiv. 2020;.

26. Burkholz R, Quackenbush J, Bojar D. Using graph convolutional neural networks to learn a representation for glycans. Cell Reports. 2021; 35(11):109251. https://doi.org/10.1016/j.celrep.2021.109251 PMID: 34133929

27. Consortium U. UniProt: the universal protein knowledgebase. Nucleic Acids Research. 2017; 45(D1): d158–d169. https://doi.org/10.1093/nar/gkw1099

28. Meier J, Rao R, Verkuil R, Liu J, Sercu T, Rives A. Language models enable zero-shot prediction of the effects of mutations on protein function. In: Ranzato M, Beygelzimer A, Nguyen K, Liang PS, Vaughan JW, Dauphin Y, editors. Advances in Neural Information Processing Systems 34; 2021.

29. Rao R, Meier J, Sercu T, Ovchinnikov S, Rives A. Transformer protein language models are unsupervised structure learners. Biorxiv. 2020;.

30. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen P, Canny J, et al. Evaluating protein transfer learning with TAPE. In: Advances in Neural Information Processing Systems; 2019. p. 9686–9698.

31. Dallago C, Mou J, Johnston KE, Wittmann B, Bhattacharya N, Goldman S, et al. FLIP: Benchmark tasks in fitness landscape inference for proteins. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2); 2021.

32. Yang KK, Lu AX, Fusi NK. Convolutions are competitive with transformers for protein sequence pretraining. bioRxiv. 2022;.

33. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018;.

34. Kalchbrenner N, Espeholt L, Simonyan K, Oord Avd, Graves A, Kavukcuoglu K. Neural machine translation in linear time. arXiv preprint arXiv:161010099. 2016;.

35. Finn RD, Bateman A, Clements J, Coggill P, Eberhardt RY, Eddy SR, et al. Pfam: the protein families database. Nucleic acids research. 2014; 42(D1):D222–D230. https://doi.org/10.1093/nar/gkt1223 PMID: 24288371

36. Kautsar SA, Blin K, Shaw S, Navarro-Muñoz JC, Terlouw BR, Van Der Hooft JJ, et al. MIBiG 2.0: a repository for biosynthetic gene clusters of known function. Nucleic acids research. 2020; 48(D1): D454–D458. https://doi.org/10.1093/nar/gkz882 PMID: 31612915

37. Blin K, Shaw S, Kloosterman AM, Charlop-Powers Z, Van Wezel GP, Medema MH, et al. antiSMASH 6.0: improving cluster detection and comparison capabilities. Nucleic acids research. 2021; 49(W1): W29–W35. https://doi.org/10.1093/nar/gkab335 PMID: 33978755

38. Petti S, Eddy SR. Constructing benchmark test sets for biological sequence analysis using independent set algorithms. bioRxiv. 2021;.

39. Hyatt D, et al. Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics. 2010; 11:119–119. https://doi.org/10.1186/1471-2105-11-119 PMID: 20211023

40. Eddy SR. Profile hidden Markov models. Bioinformatics (Oxford, England). 1998; 14(9):755–763. PMID: 9918945

41. Lewis TE, Sillitoe I, Lees JG. cath-resolve-hits: a new tool that resolves domain matches suspiciously quickly. Bioinformatics. 2019; 35(10):1766–1767. https://doi.org/10.1093/bioinformatics/bty863 PMID: 30295745

42. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox E, Garnett R, editors. Advances in Neural Information Processing Systems 32. Curran Associates, Inc.; 2019. p. 8024–8035. Available from: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.