

Substrate Prediction for RiPP Biosynthetic Enzymes via Masked Language Modeling and Transfer Learning

Joseph D. Clark,[†] Xuenan Mi,[‡] Douglas A. Mitchell,[¶] and Diwakar Shukla^{*,‡,§,||}

[†]*School of Molecular and Cellular Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[‡]*Center for Biophysics and Quantitative Biology, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[¶]*Department of Chemistry, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

[§]*Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

^{||}*Department of Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

E-mail: diwakar@illinois.edu

Abstract

Ribosomally synthesized and post-translationally modified peptide (RiPP) biosynthetic enzymes often exhibit promiscuous substrate preferences that cannot be reduced to simple rules. Large language models are promising tools for predicting such peptide fitness landscapes. However, state-of-the-art protein language models are trained on relatively few peptide sequences. A previous study comprehensively profiled the

peptide substrate preferences of LazBF (a two-component serine dehydratase) and LazDEF (a three-component azole synthetase) from the lactazole biosynthetic pathway. We demonstrated that masked language modeling of LazBF substrate preferences produced language model embeddings that improved downstream classification models of both LazBF and LazDEF substrates. Similarly, masked language modelling of LazDEF substrate preferences produced embeddings that improved the performance of classification models of both LazBF and LazDEF substrates. Our results suggest that the models learned functional forms that are transferable between distinct enzymatic transformations that act within the same biosynthetic pathway. Our transfer learning method improved performance and data efficiency in data-scarce scenarios. We then fine-tuned models on each data set and showed that the fine-tuned models provided interpretable insight that we anticipate will facilitate the design of substrate libraries that are compatible with desired RiPP biosynthetic pathways.

Introduction

Ribosomally synthesized and post-translationally modified peptides (RiPPs) are a broad category of natural products with largely untapped clinical potential.^{1,2} A typical RiPP precursor peptide contains an N-terminal leader region followed by a core region (Figure 1).³ RiPP precursor peptides undergo post-translational modifications (PTMs) in the core region, which serve to restrict conformational flexibility, enhance proteolytic resistance, and chemically diversify the natural product.³ After modification of the core peptide, the leader region is cleaved, releasing the mature RiPP. The PTMs are installed by RiPP biosynthetic enzymes, some of which display high levels of specificity while others act on diverse peptides.⁴ A significant effort has been dedicated to characterizing the substrate preferences of RiPP biosynthetic enzymes and PTM enzymes in general, which, in many cases, cannot be explained by a simple set of rules.⁵⁻¹⁰ Consequently, machine learning and deep learning are increasingly used to develop predictive models of PTM specificity.^{5,11,12} For instance,

XGBoost was used to predict the protein substrates of phosphorylation and acetylation in multiple organisms,¹³ and a transformer-based protein language model was applied to predict glycation sites in humans.¹⁴ Finally, MusiteDeep is a web server for deep learning-based PTM site prediction and visualization for proteins.¹⁵

Characterizing RiPP biosynthetic enzyme specificity is challenging, mainly due to the complexity of substrate fitness landscapes and the scarcity of sequences labeled as substrates or non-substrates.^{18,19} Accordingly, pretrained protein language models can be used to embed peptides as information rich vector representations to combat data scarcity.²⁰ Protein language models are transformer-based neural networks that learn the biological properties of polypeptides by predicting the identities of hidden residues in a training paradigm called masked language modeling.^{21,22} Masked language modeling is a form of self-supervised learning, in which a model predicts features contained within the training data (e.g., masked residues) instead of experimentally determined property labels. The protein language model representations of polypeptide sequences, also called embeddings, can be extracted and used as feature vectors for training downstream machine learning models.^{23,24} This is a canonical example of transfer learning, in which knowledge learned during one task is utilized in a distinct but related task.^{25,26} Protein language model representations have seen widespread use in peptide prediction tasks such as antimicrobial activity and toxicity prediction.²⁷⁻³¹ However, protein language models have been trained mostly on protein sequences, which have are much larger and more structurally defined compared to peptides.^{32,33} Therefore, protein language models may not fully capture peptide-specific features. Sadeh *et al.* trained self-supervised language models on peptide data, but unfortunately their models are not publicly available.³⁴ To the best of our knowledge, no self-supervised, sequence-based peptide language models are publicly available. Peptide prediction models may benefit from transfer learning paradigms in which protein language models are further trained on peptide data that is closely relevant to the downstream task. In a few cases, there exist large, high quality data sets characterizing the substrate specificity of specific RiPP biosynthetic enzymes.^{5,35}

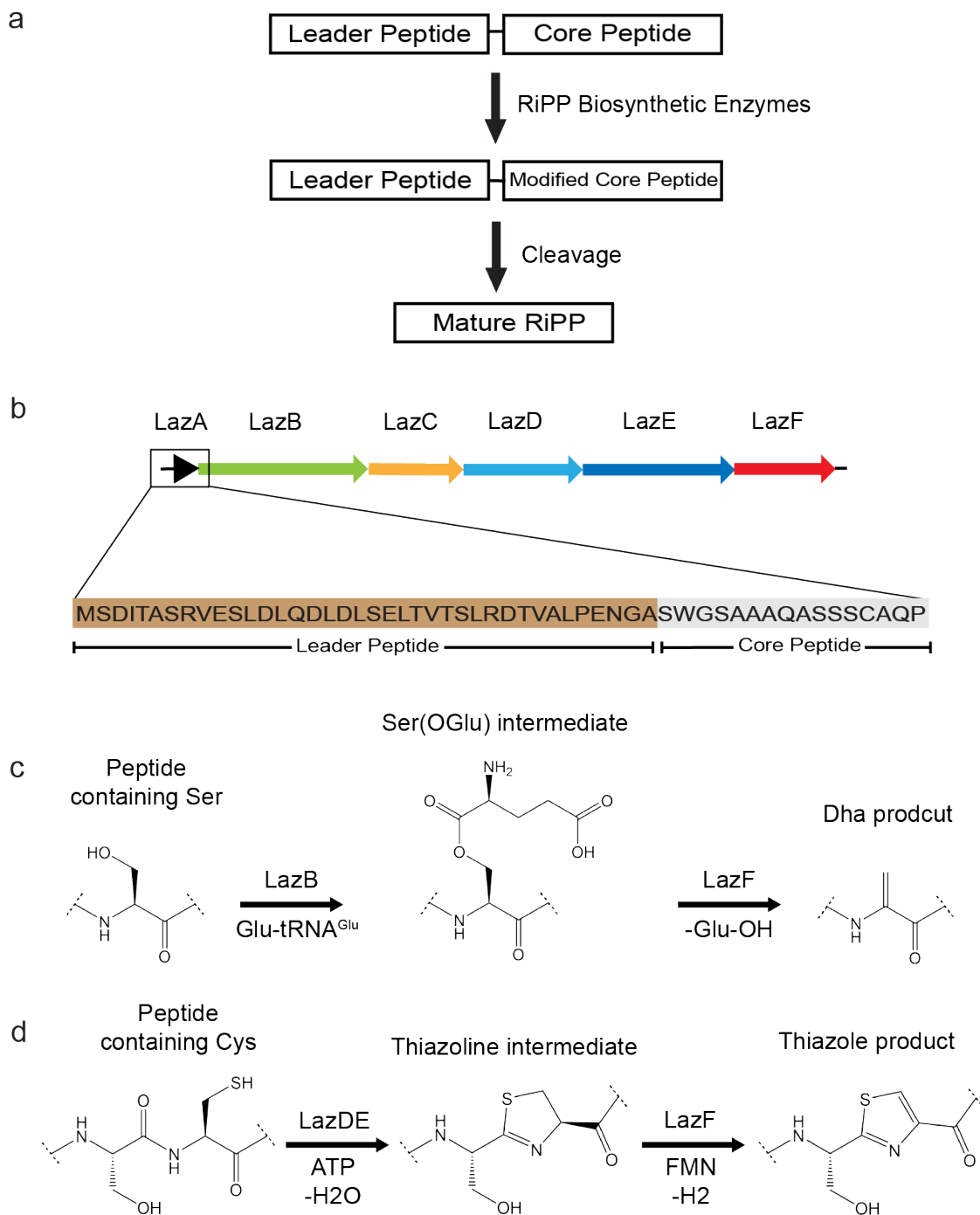


Figure 1: a) The generic biosynthesis pathway of RiPPs. RiPP precursor peptides contain a leader peptide and a core peptide. After post-translational modifications in the core peptide, the leader peptide is cleaved. b) The lactazole biosynthetic gene cluster contains six proteins. LazA is the precursor peptide. LazB (tRNA-dependent glutamylation enzyme) and the eliminase domain of LazF form a serine dehydratase while LazD (RRE-containing E1-like protein),¹⁶ LazE (YcaO cyclodehydratase),¹⁷ and the dehydrogenase domain of LazF comprise a thiazole synthetase. LazC is a pyridine synthase. c) Serine dehydration catalyzed by LazBF. d) Thiazole formation catalyzed by LazDEF.

In this work, we evaluated whether learning such data sets in a self-supervised fashion could more effectively capture functional forms that are transferable to prediction tasks of other enzymes from the same biosynthetic pathway.

Transfer learning between the substrate preferences of enzymes from the same biosynthetic pathway could potentially enhance data efficiency and model performance in situations with low data availability. To date, little work has been performed to investigate transfer learning between substrate prediction tasks of related enzymes. Lu *et al.* used a geometric machine learning approach to model the substrate preferences of protease enzymes.³⁶ This work found that models trained to predict the substrates of a single protease were able to generalize to other protease variants with multiple amino acid substitutions. In the case of RiPP biosynthetic enzymes, transfer learning could also help evaluate the degree of shared features between distinct enzymes. Such insights could aid peptide engineering tasks and facilitate a more holistic understanding of RiPP biosynthesis.

Thiopeptides are a specialized form of pyritide antibiotics deriving mostly from Bacillota and Actinomycetota.^{3,37,38} Lactazole A (LazA)³⁹ is a natural product from the pyritide family of RiPPs^{40,41} which is encoded by a biosynthetic gene cluster containing 5 synthetases (Figure 1). A diverse array of precursor peptides can be converted to lactazole-like products by these biosynthetic enzymes which catalyze post-translational modifications.⁴² LazBF is a split Ser dehydratase which installs a Dha residue in LazA precursor peptides.^{43,44} LazDEF is a split azole-forming enzyme complex which produces thiazoles in LazA precursor peptides.⁴⁵ A previous study comprehensively profiled the peptide fitness landscapes of LazBF and LazDEF (LazC was not included in their study) via the generation of two data sets each containing over 8 million LazA core sequences labeled as substrates or non-substrates.⁵ This study trained convolutional neural networks which showed excellent performance on substrate classification tasks. In the case of LazBF, dehydration sites and important residues were identified using integrated gradients,⁴⁶ an interpretable machine learning technique which determines the positive or negative contribution of each input feature to the model's

prediction. Despite the robust interpretability of their models, this study was unable to produce a general set of rules describing the substrate preferences of either LazBF or LazDEF. The comprehensive nature of the LazBF/DEF substrate data sets, and the fact that both data sets characterize related but distinct enzymes from the same biosynthetic pathway make them good candidates for exploring the plausibility of transfer learning between peptide substrate prediction tasks.

In this work, we used masked language modeling to further train protein language models on RiPP biosynthetic enzyme substrates and non-substrates. We then evaluated transfer learning between the substrate preferences of LazBF and LazDEF. Specifically, we observed that embeddings from a self-supervised language model trained on LazBF substrates and non-substrates outperformed baseline protein language model embeddings on either substrate classification task. We show a similar result in the opposite direction, where embeddings from a self-supervised model of LazDEF substrates and non-substrates outperformed baseline embeddings on either substrate classification task. Embeddings from LazBF/DEF-specific language models also outperformed embeddings from a baseline peptide language model trained on a subset of PeptideAtlas,⁴⁷ a diverse data set of mass-spectrometry identified peptides. We then trained our language models to directly classify peptides as substrates or non-substrates through a process called fine-tuning. Finally, we evaluated the transfer of interpretable machine learning techniques between the LazBF and LazDEF substrate prediction tasks. Specifically, we showed that a model fine-tuned to classify LazDEF substrates correctly identified the residue types and positions important for LazBF substrate fitness. Figure 2 presents a schematic representation of our overall workflow. Our results suggest that 1) some degree of features are shared between the fitness landscapes of LazBF and LazDEF, and 2) masked language modeling and transfer learning lead to improved predictive performance on RiPP biosynthetic enzyme prediction tasks, especially when large unlabeled data sets are available. With the increasing power of high-throughput methods, this work could enable improvement on other substrate prediction tasks by leveraging large data sets and

transfer learning.

Methods

Data Preprocessing

Vinogradov *et al.* used an mRNA display based profiling method and next-generation sequencing to generate two data sets of LazA core peptide sequences labeled as either substrates or non-substrates for LazBF and LazDEF respectively.⁵ For LazBF substrates/non-substrates, each core peptide contained a serine residue flanked by five N-terminal and five C-terminal residues (library 5S5). For LazDEF substrates/non-substrates, each core region contained cysteine flanked by six residues on each side (library 6C6). Duplicate sequences were removed from both libraries. Pairs of identical sequences found in the substrate and non-substrate bins were removed. For both libraries, a sample of 1.3 million sequences containing an equal number of substrates and non-substrates was selected. A subset of 50,000 peptides from each sample was excluded as “held-out” data for training and validation of downstream models after masked language modeling. The remaining 1.25 million LazA core peptide sequences in each sample were used as the training data for masked language modeling. Importantly, none of the held-out sequences were seen during masked language modeling. Figure 3 provides a schematic of the data preprocessing pipeline.

In a later study, Chang *et al.* used mRNA display based profiling to generate a data set of LazA core peptide sequences labeled as either substrates or non-substrates for the entire lactazole biosynthetic pathway (LazBCDEF).⁴⁸ This study comprehensively profiled the combined substrate preferences of all 5 synthetases as opposed to individual enzymes. This data set was preprocessed in a manner identical to the LazBF/DEF substrate data sets.

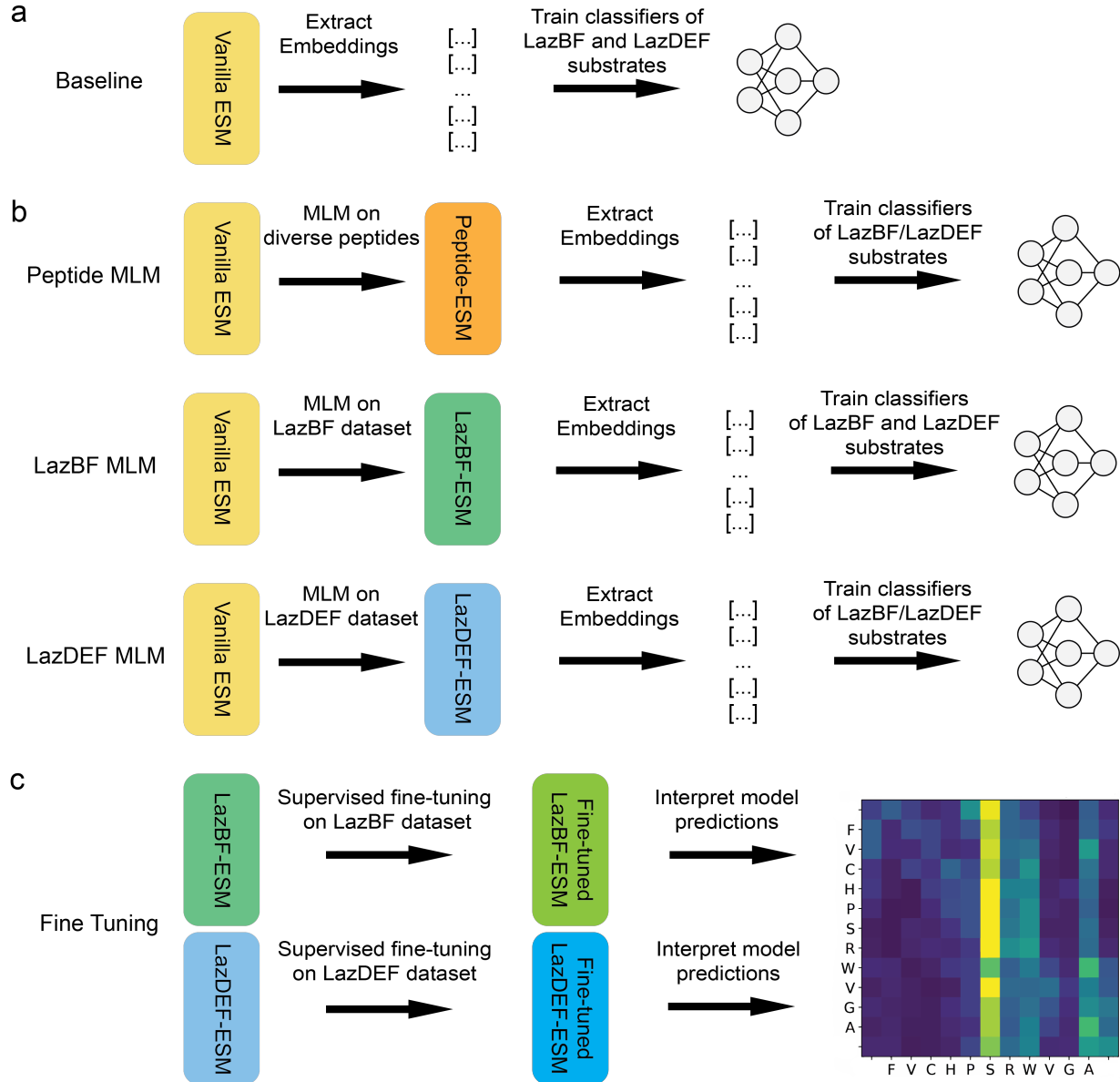


Figure 2: A schematic representation of the workflow for masked language modeling of LazBF and LazDEF substrate preferences. a) LazBF and LazDEF substrate/non-substrate embeddings were extracted from the protein language model ESM-2 (Vanilla-ESM). The baseline performance of downstream classification models was assessed. b) 3 copies of Vanilla-ESM were independently trained through masked language modeling of 3 peptide data sets. Embeddings were extracted and the performance of downstream classification models was compared to baseline. c) Models were further trained to directly classify LazBF/DEF substrates. The models' predictions were analyzed with interpretable machine learning techniques including attention analysis (see methods).

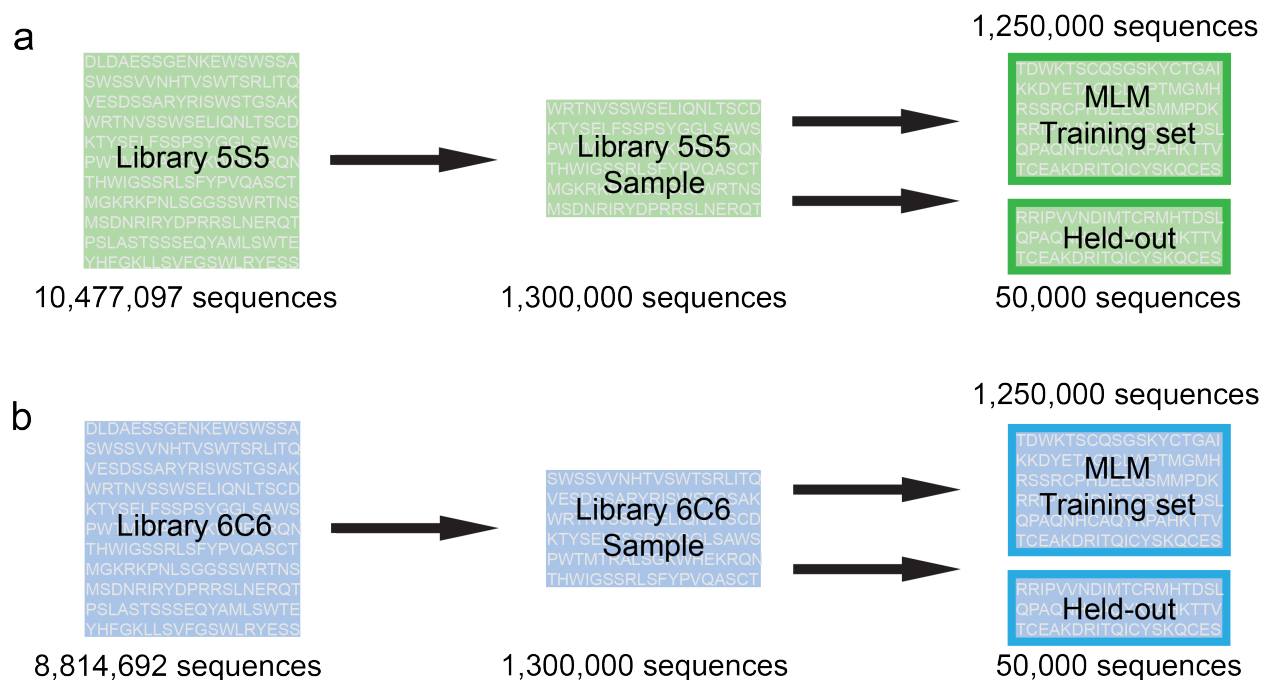


Figure 3: A schematic representation of our data preprocessing pipeline. a) LazA core sequences ($n = 1.3$ million) were selected from library 5S5. A ‘held-out’ data set of 50,000 peptides was set aside for downstream model training and evaluation. b) LazA core sequences ($n = 1.3$ million) were selected from library 6C6. A held-out data set of 50,000 peptides was set aside for downstream model training and evaluation.

Masked Language Modeling

Masked language modeling is a widely-used strategy for pretraining large language models.^{49,50} In the context of protein language models, masked language modeling takes a polypeptide sequence and replaces a random subset (15%) of the amino acids with a masking token ([MASK]). Partially masked polypeptides are fed into the model, which is optimized to predict the identity of masked residues given the context of the surrounding amino acids. This ‘self-supervised’ pretraining objective has enabled models to learn the biological features of proteins including secondary structure, long range residue-residue contacts, and mutational effects.²³ We hypothesized that, for a pretrained protein language model, further masked language modeling of the LazBF or LazDEF substrate preference data sets would update the model’s representations and enable better discrimination between substrates and non-substrates. Additionally, we sought to test how well the representations from a model

trained on LazBF substrates and non-substrates would be able to discriminate LazDEF substrates and vice versa.

ESM-2 is a family of transformer-based protein language models with state-of-the-art performance on various protein and peptide prediction tasks.^{23,51} ESM-2 is composed of a series of encoder layers, where each layer takes a numerically represented polypeptide as input and maps it to a continuous vector representation. Layers are stacked sequentially to produce increasingly rich representations. A 12-layer, 35 million parameter version of ESM-2 was used as a baseline model (Vanilla-ESM). 3 copies of Vanilla-ESM underwent additional training using masked language modeling. “LazBF-ESM” was trained on 1.25 million LazA core peptide sequences from the LazBF data set. “LazDEF-ESM” was trained on 1.25 million LazA core peptide sequences from the LazDEF data set. “Peptide-ESM” was trained on a random sample of 1.25 million sequences from Peptide Atlas.⁴⁷ Each model was trained for 1 epoch (i.e., one complete pass through the training data set) on their respective data sets with a learning rate of 3×10^{-6} and a batch size of 512.

Embedding Extraction and Downstream Model Training

Each layer of a protein language model produces vector representations of protein sequences that encode biological structure and function.^{52,53} Protein language model representations, are commonly used as the input to downstream machine learning models trained on various protein and peptide prediction tasks.^{54,55} The embeddings for all core peptides in the LazBF and LazDEF held-out data sets were extracted from Vanilla-ESM, Peptide-ESM, LazBF-ESM, and LazDEF-ESM. For each sequence, the last layer representation was obtained as a matrix of shape $L \times 480$, where L was the length of the sequence. The last layer representation was averaged across the length dimension to obtain a single 480-dimensional mean representation. The embeddings from the held-out LazBF and LazDEF data sets were used for training and validation of various machine learning models as described in the proceeding subsections. Each downstream model type was trained and validated independently on both

the LazBF and LazDEF held-out data sets. All downstream models were implemented in Scikit-learn.⁵⁶ `StandardScaler` was applied to all embeddings following standard protocols prior to training.

Unsupervised clustering

Unsupervised k -Means clustering was used to assess how well the distinction between substrates and non-substrates was represented in the high-dimensional ESM-2 embeddings. A k -Means clustering model with `n_clusters = 2` was fit to each set of embeddings. The accuracy between the ground truth labels and the k -Means predicted labels was then calculated, along with the precision, recall, area under the receiver operating characteristic (AUROC), and F1 score. For each set of embeddings, five separate k -Means models were trained with different `random_state` parameters. The model’s final performance was described by the average of the k -Means metrics.

Supervised classification models

Supervised learning models are trained by predicting properties of labeled data points (e.g., substrate or non-substrate). Logistic regression (LR), k -nearest neighbors classifier (KNN), random forest (RF), AdaBoost (AB), support vector classifier (SVC), and multi-layer perceptron (MLP) models were trained via supervised learning to predict LazBF and LazDEF substrates using the embeddings from each of the 4 protein language models as input. All embeddings were reduced to 50 dimensions with principal component analysis (PCA) before being used as the input for supervised classification models. Stratified 5-fold cross validation was performed for each model. For each fold, the accuracy, precision, recall, AUROC, and F1 score between the ground truth labels and the predicted labels was calculated. The final model performance was described by the average metrics for all 5 folds. To emulate real-world scenarios in which training data is limited, each model type was trained and validated under 3 conditions. In the “high-N” condition, 5-fold cross validation was performed such

that for each fold, 10,000 peptides were used for validation, and 40,000 peptides were used for model training. In the “medium-N” condition, 5-fold cross validation was performed such that for each fold, 10,000 peptides were used for validation, but only a random sample of 1,000 peptides were used for model training. In the “low-N” condition, 5-fold cross validation was performed such that for each fold, 10,000 peptides were used for validation, but only a random sample of 100 peptides were used for model training. Hyperparameters of each supervised model were optimized separately for each set of embeddings under each condition using grid search. The optimized hyperparameters for all downstream models are in Tables S1-S3.

Embedding space visualization

t-Distributed Stochastic Neighbor Embedding (t-SNE) was used to visualize the embeddings from each protein language model. A sample of 5,000 peptides from both held-out data sets were selected for visualization. The 480-dimensional embeddings were first reduced to 100 dimensions with PCA, and then further reduced to two dimensions with t-SNE.

Fine-Tuning, Integrated Gradients, and Attention Analysis

Fine-tuning refers to further training a language model to directly predict properties of labeled data points using supervised learning.⁵⁷ Fine-tuning boosts the model’s performance on a downstream task in part by transferring broader knowledge learned during masked language modelling. The embeddings from the language model are not extracted at any point during fine-tuning. Instead, all the model’s parameters are optimized to classify labeled training data. Both LazBF-ESM and LazDEF-ESM were fine-tuned using supervised learning on their respective data sets. For each model, the same sequences used for masked language modeling were used as the training set for fine-tuning. The same held-out data sets containing sequences unseen during masked language modeling and fine-tuning were used to evaluate the fine-tuned models. Vanilla-ESM was also fine-tuned to classify substrates of the

entire lactazole biosynthetic pathway. The accuracy on each held-out data set was calculated for each of the 3 fine-tuned models.

Integrated gradients are an interpretable machine learning technique used to quantify the positive or negative contribution of input features to a model’s prediction for a given data point.⁴⁶ In the context of predicting whether a peptide is the substrate of an enzyme, a positive value for a given residue implies that the residue is important for substrate fitness. A negative value for a given residue suggests that the residue is associated with being a non-substrate. The fine-tuned LazBF model and the fine-tuned LazDEF model were separately used to calculate the integrated gradients for each peptide in the held-out LazBF data set. For each model, and for each residue type, all contributions of that residue across all 50,000 sequences were summed and then divided by the frequency of that residue in the held-out LazBF data, producing two matrices of shape 1×20 representing the average contribution of each residue type according to the integrated gradients of each model. A similar procedure produced two 1×11 matrices, representing the average contribution of each position for each model. Finally, a similar procedure produced two 20×11 matrices, representing the average contribution of each residue type in each position for each model.

ESM-2 employs a multi-head self-attention mechanism, where each of the 12 layers produce 20 attention heads (240 attention heads in total).⁵⁸ Each attention head is a 2D matrix α of shape $L \times L$, where L is the length of the tokenized input sequence. The tokenized input sequence includes a “beginning of sequence” ([BOS]) and an “end of sequence” ([EOS]) character in addition to the amino acids. Individual attention weights $\alpha_{i,j}$ quantify how much the residue at position i affects the model’s representation of the residue at position j , with greater values suggesting greater influence. Attention weights have been shown to highlight biological features of proteins including residue-residue contacts and binding sites.⁵⁹ The pairwise nature of the self-attention mechanism resembles epistatic interactions in protein/peptide fitness landscapes.⁶⁰ Vinogradov *et al.* calculated pairwise epi-scores that attempted to quantify how the fitness of a residue at a given position is affected by residues

at other positions.⁵ Thus, we looked for similarities between self-attention matrices and the pairwise epi-scores calculated in previous work for one LazBF and one LazDEF substrate. All 240 attention matrices were obtained for both peptides.

Results and discussion

Vanilla-ESM Baseline

We first evaluated the performance of downstream LazBF and LazDEF substrate classification models trained on embeddings from a baseline protein language model (Vanilla-ESM). The performance of each model type was evaluated separately under a high-N, medium-N, and low-N condition defined by the number of sequences used for training. The results of each model type trained on embeddings from Vanilla-ESM – without any additional masked language modeling – are displayed in Table 1. Embeddings from Vanilla-ESM perform reasonably well on RiPP biosynthetic enzyme substrate classification tasks, particularly in the high-N condition for the LazBF substrate prediction task. The reasonable performance of Vanilla-ESM embeddings underscores the richness of protein language model representations, which can effectively generalize to novel tasks. Models trained on fewer training samples (i.e., medium-N and low-N) had lower performance. This reflects the importance of having sufficiently large and diverse training data in supervised learning paradigms.

Masked Language Modeling Improves LazDEF Substrate Classification Performance

The accuracy of each supervised model type trained on LazDEF substrate/non-substrate embeddings from each of the four language models are presented in Figure 4. The precision, recall, AUROC, and F1 score of each model type is available in Figures S1-S4. LazDEF-ESM produced embeddings that significantly increased the performance of all downstream

Table 1: Classification Accuracy with Vanilla-ESM Embeddings

	Training Size	SVC	MLP	LR	RF	AB	KNN
LazBF	High-N	90.9 ± 0.13	90.2 ± 0.31	89.6 ± 0.27	88.4 ± 0.26	88.8 ± 0.42	87.4 ± 0.29
LazDEF	High-N	83.0 ± 0.27	81.5 ± 0.19	80.5 ± 0.21	79.7 ± 0.32	79.4 ± 0.48	78.0 ± 0.26
	Training Size	SVC	MLP	LR	RF	AB	KNN
LazBF	Medium-N	88.1 ± 0.57	88.0 ± 0.26	88.2 ± 0.09	84.9 ± 0.33	85.4 ± 0.45	83.8 ± 0.80
LazDEF	Medium-N	78.1 ± 0.41	76.8 ± 0.30	79.0 ± 0.47	76.4 ± 0.51	75.4 ± 0.35	73.9 ± 0.57
	Training Size	SVC	MLP	LR	RF	AB	KNN
LazBF	Low-N	82.2 ± 0.92	81.3 ± 0.93	80.9 ± 2.07	77.4 ± 1.75	76.1 ± 1.77	77.0 ± 0.80
LazDEF	Low-N	70.8 ± 0.97	72.3 ± 0.74	72.7 ± 0.94	69.1 ± 1.15	64.8 ± 1.35	67.6 ± 0.86

Accuracy of support vector classifier (SVC), multi-layer perceptron (MLP), logistic regression (LR), random forest (RF), AdaBoost (AB), and k -nearest neighbors classifier (KNN) models trained on embeddings from Vanilla-ESM on both substrate classification tasks. Values are Mean \pm SD. The best performing model in each row is highlighted.

LazDEF substrate classification models across all training sizes. We suspect that during masked language modeling, the model became attuned to specific features of the LazDEF data set, including the features that distinguish substrates from non-substrates. The model’s representations were updated in accordance with these features, allowing for improved discrimination of substrate and non-substrate sequences.

Strikingly, LazBF-ESM also produced embeddings that significantly increased the performance of LazDEF substrate classification models. Every LazDEF substrate classification model across all training sizes showed a sizable improvement in performance when trained on embeddings from LazBF-ESM, demonstrating that transfer learning improved the performance of the models. Embeddings from Peptide-ESM also improved LazDEF substrate classification models in nearly all cases, but to a lesser extent than LazBF-ESM or LazDEF-ESM embeddings. This indicated that large data sets characterizing the substrate preferences of specific RiPP biosynthetic enzymes provided the most utility in improving RiPP biosynthetic enzyme substrate classification models.

t-SNE was then used to reduce each set of embeddings to two dimensions for visualization. t-SNE plots of the Vanilla-ESM and Peptide-ESM embedding spaces of LazDEF substrates and non-substrates do not show any apparent distinction between substrates and non-substrates (Figures 5, S9). Notably, the LazBF-ESM embedding space shows a visi-

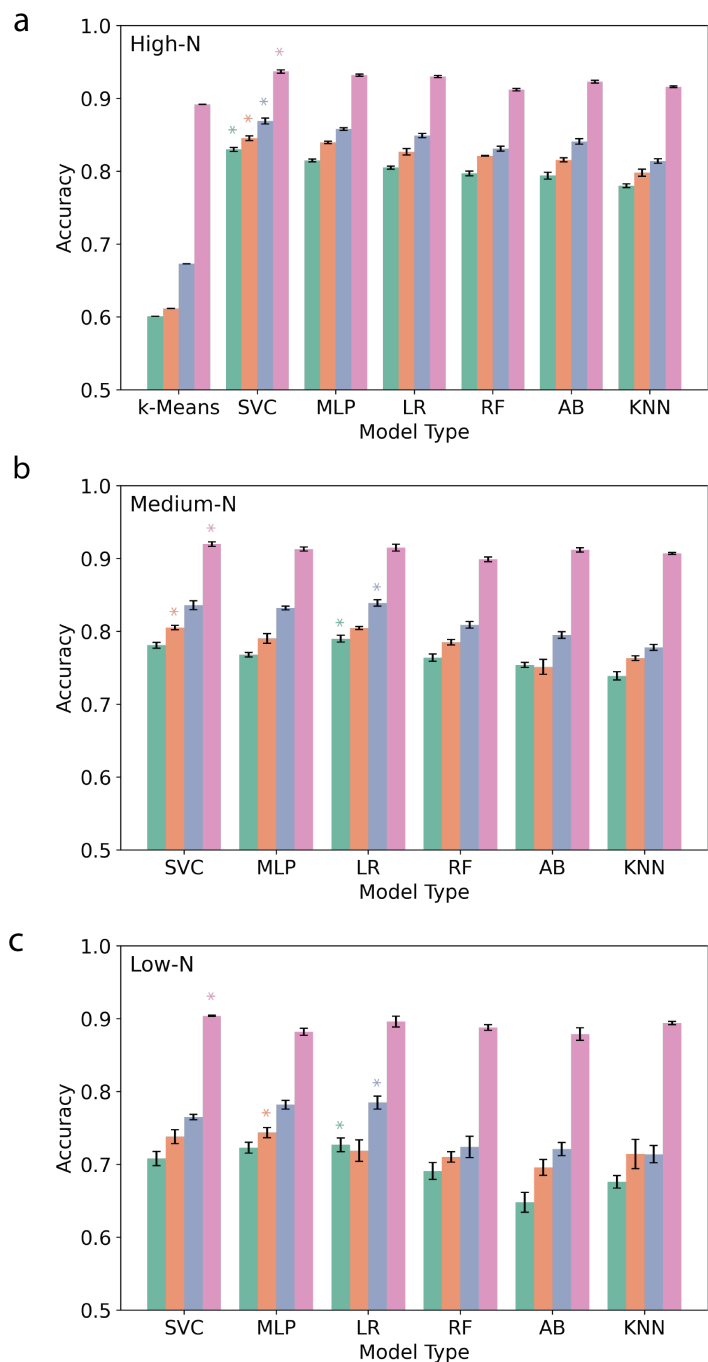


Figure 4: Accuracy of LazDEF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

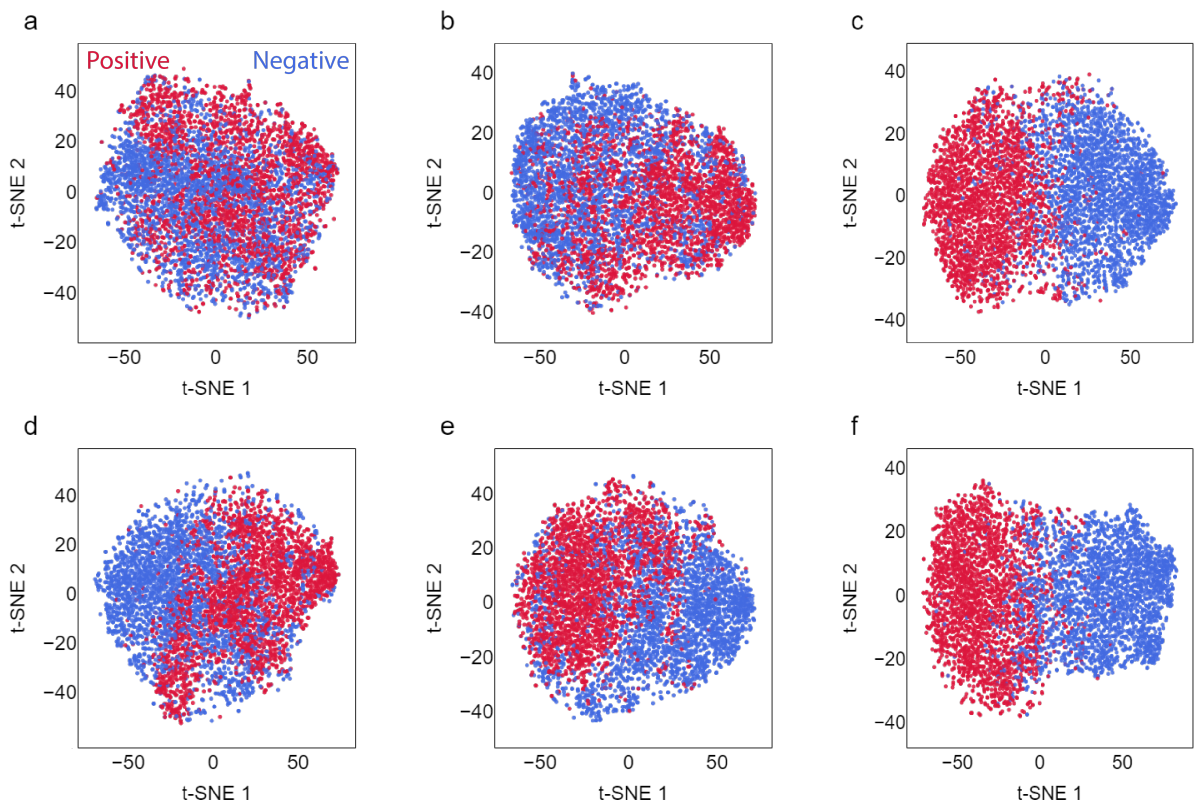


Figure 5: t-SNE visualization of the LazDEF embedding space for a) Vanilla-ESM, b) ESM trained on LazBF substrates/non-substrates, and c) ESM trained on LazDEF substrates/non-substrates. t-SNE visualization of the LazBF embedding space for d) Vanilla-ESM, e) ESM trained on LazDEF substrates/non-substrates, and f) ESM trained on LazBF substrates/non-substrates. Substrates are red and non-substrates samples are blue.

bly higher degree of clustering within substrates and non-substrates than the Vanilla-ESM embedding space. This agrees with the increase in downstream LazDEF substrate classification model performance observed after masked language modeling of the LazBF data set. Finally, the LazDEF-ESM embedding space shows the most obvious segregation (Figure 5). The increased ability to distinguish LazDEF substrates/non-substrates suggests that using embeddings from a language model trained on a large data set relevant to the task of interest can greatly increase the predictive power of downstream classifiers through transfer learning.

Masked Language Modeling of Either Data Set Improves LazBF Substrate Classification Performance

The accuracy of each model type trained on embeddings of LazBF substrates/non-substrates from each of the 4 protein language models are presented in Figure 6. The precision, recall, AUROC, and F1 score of each model type is available in Figures S5-S8. Similarly, LazBF-ESM produced embeddings that significantly improved the performance of both unsupervised k -Means clustering and supervised classification models of LazBF substrates across all training sizes. LazDEF-ESM also produced embeddings that improved the performance of most LazBF substrate classification models. In the high-N condition, all models showed performance increases, with unsupervised k -Means clustering showing the most improvement. Most supervised models trained using the medium-N and low-N conditions also showed improved performance. SVC and MLP showed the largest and most consistent increases across these two conditions. Expectedly, the low-N condition produced models with higher variance, which likely contributed to more unstable results. In most cases, LazDEF-ESM embeddings also outperformed Peptide-ESM embeddings.

A t-SNE plot of the LazBF substrate/non-substrate embeddings from Vanilla-ESM and Peptide-ESM show an already apparent distinction between substrates and non-substrates (Figures 5, S9). This suggests that the pretrained model is sensitive to differences inherent in LazBF substrates and non-substrates. The visual divergence of substrates and non-substrates is arguably more apparent in the embedding space of LazDEF-ESM (Figure 5e). Predictably, the embedding space of LazBF-ESM shows the most dramatic separation of substrates from non-substrates (Figure 5f). This is consistent with large increases in downstream LazBF substrate classification model performance after masked language modeling of the LazBF data set.

The observation that LazBF substrate classifiers showed improved performance when trained on embeddings from LazDEF-ESM suggests that information relevant to LazBF classification was learned during masked language modeling of the LazDEF substrates/non-

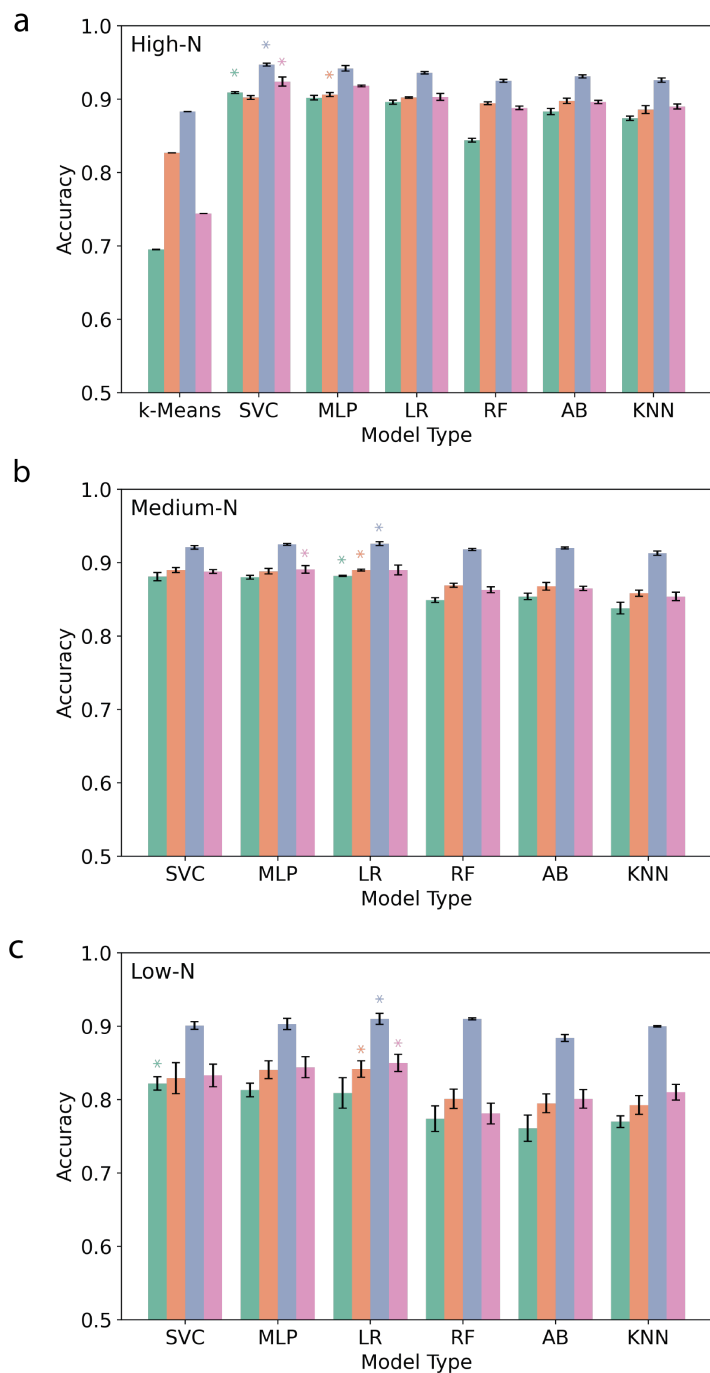


Figure 6: Accuracy of LazBF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

substrates. However, Vanilla-ESM embeddings already showed good performance on LazBF prediction tasks. We suspect that this left less room for improvement through masked language modeling of the LazDEF data set. However, any improvement is compelling given that 1) LazBF and LazDEF catalyze disparate transformations and 2) the substrate fitness landscapes of LazBF and LazDEF are reported to be divergent from one another, particularly in the degree to which pairwise positional epistasis affects fitness.⁵ Tanimoto similarity is a common metric used to quantify the chemical similarity between small molecules and peptides. The average Tanimoto similarity between peptides in the held-out LazBF and held-out LazDEF substrate data sets was calculated to be 0.354 ± 0.031 , suggesting that the data sets contained relatively dissimilar sequences. The results of this and the previous section show that knowledge learned during the unsupervised modeling of RiPP biosynthetic enzyme substrates/non-substrates can be transferred to other tasks, particularly those that involve related but distinct enzymes from the same biosynthetic pathway. Additionally, unsupervised modeling of RiPP biosynthetic enzyme substrates/non-substrates appear to produce better representations than unsupervised modeling of diverse peptides.

Despite catalyzing different transformations, both LazBF and LazDEF bind LazA precursor peptides as substrates. Therefore, there is expected to be some degree of similarity between the substrate preferences of the two enzymes. However, we observed that more information about LazDEF substrate preferences was learned from masked language modeling of LazBF substrate preferences than vice versa. We hypothesize that this asymmetry results from the order of the post-translational modifications that occur during lactazole biosynthesis. In nature, LazDEF modifies LazA precursor peptides prior to LazBF.⁴⁵ Therefore, self-supervised modeling of LazBF substrate preferences learns the biophysical features of substrates that are likely to have been modified by LazDEF. However, the opposite is not necessarily true. This presents an intuitive explanation as to why transfer learning showed greater success at improving LazDEF substrate classification models.

Fine-Tuned Language Model Performance on RiPP Biosynthetic Enzyme Classification Tasks

LazBF-ESM and LazDEF-ESM were then trained to classify the substrates of their respective data sets through a training procedure called fine-tuning. Vanilla-ESM was also fine-tuned to classify substrates of the entire lactazole biosynthetic pathway. Each fine-tuned model showed excellent performance on its respective held-out data set (>0.95 accuracy in each case). We also evaluated how well each fine-tuned model performed on the other held-out data sets without any further training (Table 2). The fine-tuned LazBF-ESM model showed no ability to classify LazDEF substrates, and showed little ability to classify substrates for the entire pathway after supervised training. In contrast, the fine-tuned LazDEF model achieved 0.697 accuracy on the held-out LazBF substrate data set, likely due in part to the LazBF data set being more enriched (Figure 5d). This model also showed some ability to classify substrates of the entire lactazole biosynthetic pathway. Finally, the supervised model trained to classify substrates of the entire pathway showed some ability to classify LazBF and LazDEF substrates without any further training.

Integrated gradients can quantify how individual residues contribute to a model’s prediction. Inspired by the performance of LazDEF-ESM on the LazBF substrate classification task, we looked for similarities between the integrated gradients for LazBF substrates/non-substrates from both models (Figure 7). We observed that the average contribution of each residue type from fine-tuned LazDEF-ESM strongly correlated with the average contribution of each residue type from fine-tuned LazBF-ESM, with a spearman coefficient of 0.80 (Figure 7a). Similarly, the average contribution of each position from both fine-tuned models showed a 0.81 spearman coefficient (Figure 7b). The average contribution of each residue type in each position also showed a moderate correlation (0.59 spearman coefficient). These correlations exist despite fine-tuned LazDEF-ESM having never been trained on LazBF substrates. Therefore, to some extent, fine-tuned RiPP biosynthetic enzyme prediction models can produce valid and interpretable predictions about distinct, but related prediction tasks.

Table 2: Classification Accuracy Fine-Tuned Models

	Supervised LazBF	Supervised LazDEF	Supervised LazBCDEF
LazBF test set	99.3%	69.7%	64.8%
LazDEF test set	50.9%	99.2%	58.8%
LazBCDEF test set	52.3%	64.1%	95.9%

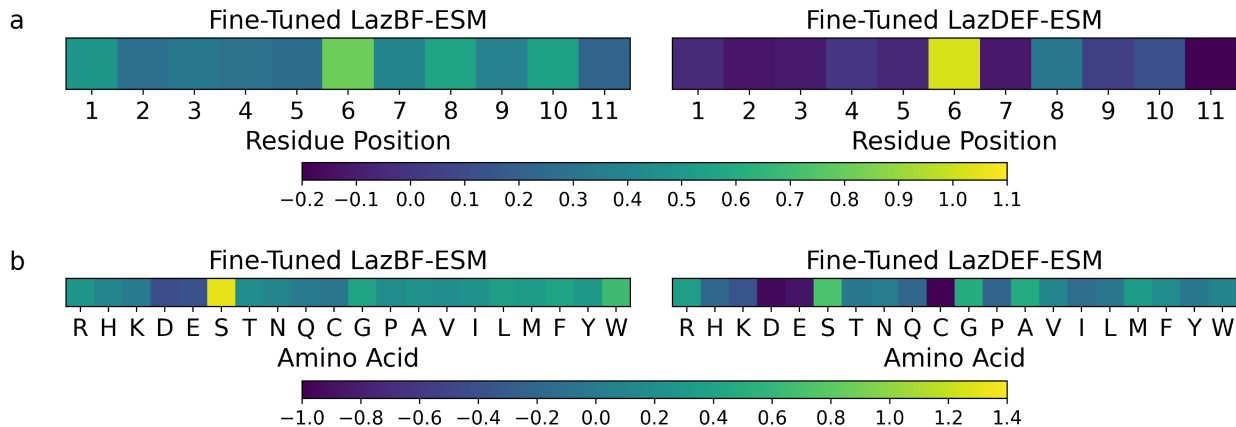


Figure 7: Fine-tuned LazBF-ESM and fine-tuned LazDEF-ESM produce correlated integrated gradients for LazBF substrates/non-substrates. a) The average contribution of each position to substrate fitness shows a 0.81 spearmanr between the two models. b) The average contribution of each amino acid to substrate fitness shows a 0.80 spearmanr between the two models.

Attention Analysis

Attention matrices describe the model’s perceived relevance or association between each pair of tokenized residues, including the [BOS] and [EOS] tokens added to the beginning and the end of the peptide respectively (see methods). Higher values between a pair of tokens indicates greater relevance between them. Analyzing attention matrices can provide insight into which residues the model regards as important. We observe a general trend in which the attention heads from earlier layers focus mainly on the [BOS] and [EOS] tokens, while heads from later layers dedicate significant attention to specific residues or motifs (Figures 8a, S10). Our observation that the model’s attention mechanism ‘zeros-in’ on important residues is consistent with the widespread claim that the per-layer representations of protein language models are hierarchical in nature, with earlier layers encoding low-level features and later layers encoding more global representations of structure and/or function.⁵⁹

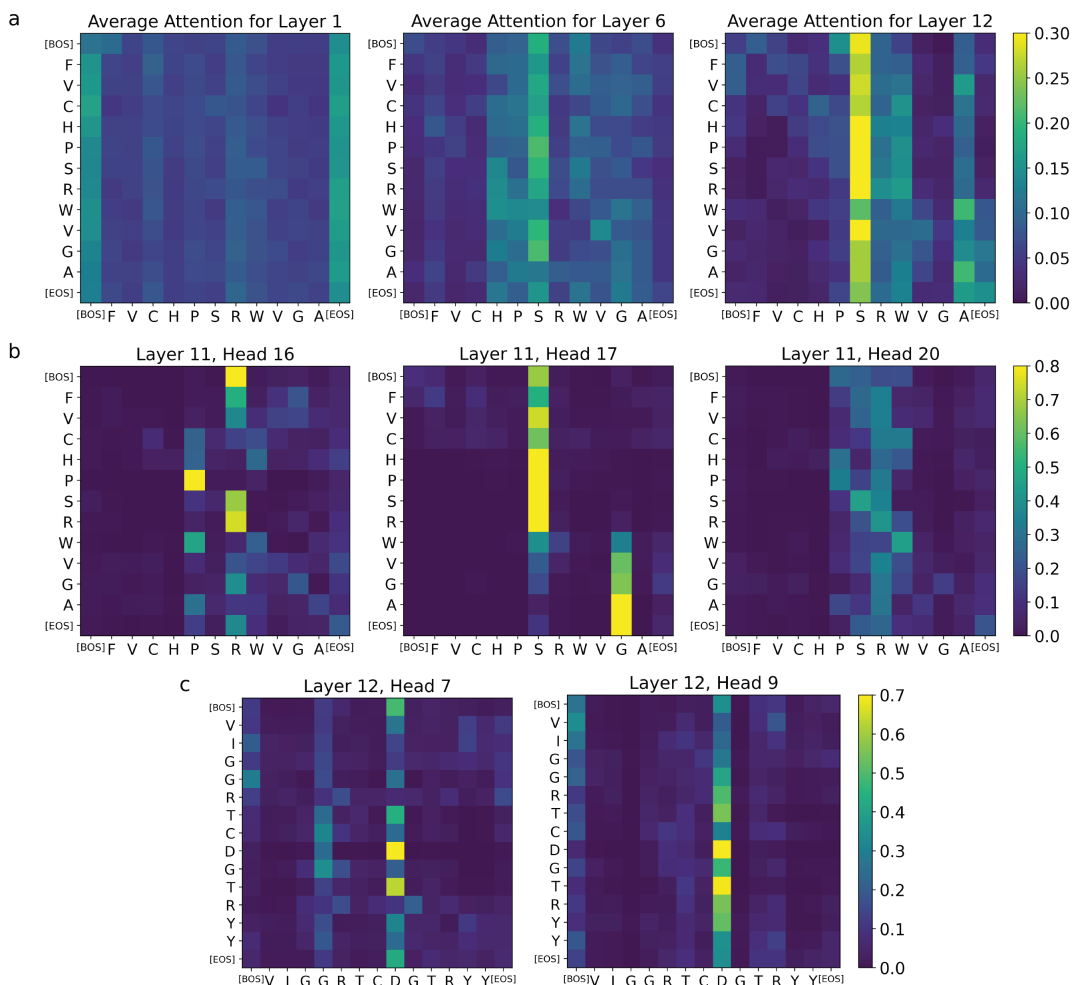


Figure 8: Attention maps from the fine-tuned LazBF-ESM. [BOS] and [EOS] tokens mark the “beginning of sequence” and “end of sequence” respectively. a) Middle and later layers focus on specific residues and motifs. b) Attention heads from the penultimate layer highlight a motif with high pairwise epi-scores in a LazBF substrate. c) Attention heads from the final layer highlight a residue important for substrate fitness in a LazDEF substrate.

Previous work utilized predictive machine learning models to calculate the pairwise epi-scores for LazBF substrates. Pairwise epi-score values provide an estimate of the strength with which amino acids in the core peptide region affect each other’s fitness.⁵ The self-attention mechanism found in transformer models resembles pairwise epi-scores by quantifying the degree to which one amino acid affects the representations of other amino acids in the peptide.^{59,60} For the LazBF substrate FVCHPSRWVGA, the computed pairwise epi-scores suggest that a His4-Pro5-Ser6-Arg7-Trp8 motif contributes to the fitness of the peptide.⁵

Figure 8b shows that multiple attention heads in the 11th layer of the fine-tuned LazBF-ESM dedicate attention between pairs of amino acids within this motif. This suggests that the supervised protein language model’s attention mechanism is somewhat able to highlight epistatic interactions and provide a rough idea of which residues are important for fitness.

Surprisingly, we observe that the fine-tuned version of LazBF-ESM also highlights some epistatic features of the LazDEF substrate VIGGRTCDGTRY (Figure 8c). Precalculated epi-scores for this peptide indicate that Asp8 has numerous positive and negative epistatic interactions with surrounding peptides including Thr6, Gly9, and Arg11. We find that multiple heads from the last layer of our fine-tuned LazBF-ESM dedicate significant attention between Asp8 and nearby residues, thus highlighting Asp8 as an important residue.

Conclusions

In this work, we enhanced the performance of protein language model embeddings for RiPP biosynthetic enzyme substrate prediction tasks by performing masked language modeling of substrate/non-substrate data. We applied transfer learning to improve the performance of peptide substrate prediction models for distinct enzymes from the same biosynthetic pathway. A limited number of studies have explored transfer learning in the domain of enzyme substrate prediction, and, to the best of our knowledge, this is the first work to investigate transfer learning between RiPP biosynthetic enzymes.

We focused on LazBF and LazDEF, a serine dehydratase and azole synthetase respectively, from the lactazole biosynthesis pathway. Masked language modeling was used to train two peptide language models on data sets comprised of LazA sequences labeled as substrates or non-substrates for LazBF and LazDEF respectively. An additional peptide language model was trained on a diverse set of non-LazA peptides. We found that all peptide language models produced embeddings that increased the performance of downstream classification models on both substrate prediction tasks. The LazBF/DEF models provided

the largest increases in performance. This suggested some information is shared between the two fitness landscapes, and that masked language modeling of one data set allowed the model to learn important features of the other data set. The performance enhancements were most significant for downstream LazDEF classification models, including the medium-N and low-N conditions. Our workflow enhances the ability to classify RiPP biosynthetic enzyme substrates in limited data regime. This is attractive in the context of peptide engineering, where it could expedite peptide design and discovery by reducing the need for comprehensive experimental profiling.

We also demonstrated that interpretable machine learning techniques are somewhat transferable between similar RiPP biosynthetic enzyme classification tasks. Specifically, we found that the integrated gradients for LazBF peptides from a supervised LazDEF model correlated with the integrated gradients from a supervised LazBF model. Due to the increasing abundance of sequence data and rapid advances in next-generation sequencing technology, we anticipate the development of large peptide data sets suitable for masked language modeling. Coupled with the growing size and sophistication of protein language models, we expect masked language modeling and transfer learning to aid enzyme substrate prediction tasks especially in cases where large data sets for related enzymes are available.

Acknowledgement

The authors thank Dr. Alexander A. Vinogradov, Dr. Yuki Goto, and Dr. Hiroaki Suga for sharing the data sets used in this study. JDC thanks Song Yin for his comments. DS acknowledges support from NIH grants R35GM142745 and R21AI167693.

Supporting Information Available

Source code for data analysis, downstream model training and validation, along with trained model weights are available at <https://github.com/ShuklaGroup/LazBFDEF>.

References

- (1) Ongpipattanakul, C.; Desormeaux, E. K.; DiCaprio, A.; van der Donk, W. A.; Mitchell, D. A.; Nair, S. K. Mechanism of Action of Ribosomally Synthesized and Post-Translationally Modified Peptides. *Chemical Reviews* **2022**, *122*, 14722–14814.
- (2) Fu, Y.; Jaarsma, A. H.; Kuipers, O. P. Antiviral activities and applications of ribosomally synthesized and post-translationally modified peptides (RiPPs). *Cellular and Molecular Life Sciences* **2021**, *78*, 3921–3940.
- (3) Montalbán-López, M. et al. New developments in RiPP discovery, enzymology and engineering. *Natural Product Reports* **2021**, *38*, 130–239.
- (4) Arnison, P. G. et al. Ribosomally synthesized and post-translationally modified peptide natural products: overview and recommendations for a universal nomenclature. *Nat. Prod. Rep.* **2013**, *30*, 108–160.
- (5) Vinogradov, A. A.; Chang, J. S.; Onaka, H.; Goto, Y.; Suga, H. Accurate Models of Substrate Preferences of Post-Translational Modification Enzymes from a Combination of mRNA Display and Deep Learning. *ACS Central Science* **2022**, *8*, 814–824.
- (6) Ivry, S. L.; Meyer, N. O.; Winter, M. B.; Bohn, M. F.; Knudsen, G. M.; O'Donoghue, A. J.; Craik, C. S. Global substrate specificity profiling of post-translational modifying enzymes. *Protein Science* **2017**, *27*, 584–594.
- (7) Tang, W.; Jiménez-Osés, G.; Houk, K. N.; van der Donk, W. A. Substrate control in stereoselective lanthionine biosynthesis. *Nature Chemistry* **2014**, *7*, 57–64.
- (8) Le, T.; Fouque, K. J. D.; Santos-Fernandez, M.; Navo, C. D.; Jiménez-Osés, G.; Sarkisian, R.; Fernandez-Lima, F. A.; van der Donk, W. A. Substrate Sequence Controls Regioselectivity of Lanthionine Formation by ProcM. *Journal of the American Chemical Society* **2021**, *143*, 18733–18743.

- (9) Song, I.; Kim, Y.; Yu, J.; Go, S. Y.; Lee, H. G.; Song, W. J.; Kim, S. Molecular mechanism underlying substrate recognition of the peptide macrocyclase PsnB. *Nature Chemical Biology* **2021**, *17*, 1123–1131.
- (10) Mahajan, S. P.; Srinivasan, Y.; Labonte, J. W.; DeLisa, M. P.; Gray, J. J. Structural Basis for Peptide Substrate Specificities of Glycosyltransferase GalNAc-T2. *ACS Catalysis* **2021**, *11*, 2977–2991.
- (11) Meng, L.; Chan, W.-S.; Huang, L.; Liu, L.; Chen, X.; Zhang, W.; Wang, F.; Cheng, K.; Sun, H.; Wong, K.-C. Mini-review: Recent advances in post-translational modification site prediction based on deep learning. *Computational and Structural Biotechnology Journal* **2022**, *20*, 3522–3532.
- (12) Yan, Y.; Jiang, J.-Y.; Fu, M.; Wang, D.; Pelletier, A. R.; Sigdel, D.; Ng, D. C.; Wang, W.; Ping, P. MIND-S is a deep-learning prediction model for elucidating protein post-translational modifications in human diseases. *Cell Reports Methods* **2023**, *3*, 100430.
- (13) Smith, K.; Rhoads, N.; Chandrasekaran, S. Protocol for CAROM: A machine learning tool to predict post-translational regulation from metabolic signatures. *STAR Protocols* **2022**, *3*, 101799.
- (14) Liu, Y.; Liu, Y.; Wang, G.-A.; Cheng, Y.; Bi, S.; Zhu, X. BERT-Kgly: A Bidirectional Encoder Representations From Transformers (BERT)-Based Model for Predicting Lysine Glycation Site for Homo sapiens. *Frontiers in Bioinformatics* **2022**, *2*, 834153.
- (15) Wang, D.; Liu, D.; Yuchi, J.; He, F.; Jiang, Y.; Cai, S.; Li, J.; Xu, D. MusiteDeep: a deep-learning based webserver for protein post-translational modification site prediction and visualization. *Nucleic Acids Research* **2020**, *48*, W140–W146.
- (16) Burkhardt, B. J.; Hudson, G. A.; Dunbar, K. L.; Mitchell, D. A. A prevalent peptide-

- binding domain guides ribosomal natural product biosynthesis. *Nature Chemical Biology* **2015**, *11*, 564–570.
- (17) Burkhart, B. J.; Schwalen, C. J.; Mann, G.; Naismith, J. H.; Mitchell, D. A. YcaO-Dependent Posttranslational Amide Activation: Biosynthesis, Structure, and Function. *Chemical Reviews* **2017**, *117*, 5389–5456.
- (18) Zhao, Y.; Jensen, O. N. Modification-specific proteomics: Strategies for characterization of post-translational modifications using enrichment techniques. *PROTEOMICS* **2009**, *9*, 4632–4641.
- (19) Kaltashov, I. A.; Bobst, C. E.; Abzalimov, R. R.; Wang, G.; Baykal, B.; Wang, S. Advances and challenges in analytical characterization of biotechnology products: Mass spectrometry-based approaches to study properties and behavior of protein therapeutics. *Biotechnology Advances* **2012**, *30*, 210–222.
- (20) Brandes, N.; Ofer, D.; Peleg, Y.; Rappoport, N.; Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics* **2022**, *38*, 2102–2110.
- (21) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130.
- (22) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. S. Evaluating Protein Transfer Learning with TAPE. *arXiv* **2019**, preprint, DOI:10.48550/ARXIV.1906.08230.
- (23) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological structure and function emerge from scaling

- unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **2021**, *118*, 2016239118.
- (24) Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rihawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; Bhowmik, D.; Rost, B. ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv* **2020**, preprint, DOI:10.48550/ARXIV.2007.06225.
- (25) Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A Comprehensive Survey on Transfer Learning. *arXiv* **2019**, preprint, DOI:10.48550/ARXIV.1911.02685.
- (26) Shamsi, Z.; Chan, M.; Shukla, D. TLmutation: Predicting the Effects of Mutations Using Transfer Learning. *The Journal of Physical Chemistry B* **2020**, *124*, 3845–3854.
- (27) Wang, D.; Jin, J.; Li, Z.; Wang, Y.; Fan, M.; Liang, S.; Su, R.; Wei, L. StructuralDPPIV: A novel deep learning model based on atom-structure for predicting dipeptidyl peptidase-IV inhibitory peptides. *bioRxiv* **2023**, preprint, DOI:10.1101/2023.05.22.541389.
- (28) Wang, L.; Huang, C.; Wang, M.; Xue, Z.; Wang, Y. NeuroPred-PLM: an interpretable and robust model for neuropeptide prediction by protein language model. *Briefings in Bioinformatics* **2023**, *24*, bbad077.
- (29) Zhang, Y.; Lin, J.; Zhao, L.; Zeng, X.; Liu, X. A novel antibacterial peptide recognition algorithm based on BERT. *Briefings in Bioinformatics* **2021**, *22*, bbab200.
- (30) Ma, Z.; Zou, Y.; Huang, X.; Yan, W.; Xu, H.; Yang, J.; Zhang, Y.; Huang, J. pLMF-PPred: a novel approach for accurate prediction of functional peptides integrating embedding from pre-trained protein language model and imbalanced learning. *arXiv* **2023**, preprint, DOI:10.48550/ARXIV.2309.14404.

- (31) Du, Z.; Ding, X.; Hsu, W.; Munir, A.; Xu, Y.; Li, Y. pLM4ACE: A protein language model based predictor for antihypertensive peptide screening. *Food Chemistry* **2024**, *431*, 137162.
- (32) Fosgerau, K.; Hoffmann, T. Peptide therapeutics: current status and future directions. *Drug Discovery Today* **2015**, *20*, 122–128.
- (33) Muttenthaler, M.; King, G. F.; Adams, D. J.; Alewood, P. F. Trends in peptide drug discovery. *Nature Reviews Drug Discovery* **2021**, *20*, 309–325.
- (34) Sadeh, G.; Wang, Z.; Grewal, J.; Rangwala, H.; Price, L. Training self-supervised peptide sequence models on artificially chopped proteins. *arXiv* **2022**, preprint, DOI:10.48550/ARXIV.2211.06428.
- (35) Huang, H.; Arighi, C. N.; Ross, K. E.; Ren, J.; Li, G.; Chen, S.-C.; Wang, Q.; Cowart, J.; Vijay-Shanker, K.; Wu, C. H. iPTMnet: an integrated resource for protein post-translational modification network discovery. *Nucleic Acids Research* **2017**, *46*, D542–D550.
- (36) Lu, C.; Lubin, J. H.; Sarma, V. V.; Stentz, S. Z.; Wang, G.; Wang, S.; Khare, S. D. Prediction and design of protease enzyme specificity using a structure-aware graph convolutional network. *Proceedings of the National Academy of Sciences* **2023**, *120*, 2303590120.
- (37) Chan, D. C. K.; Burrows, L. L. Thiopeptides: antibiotics with unique chemical structures and diverse biological activities. *The Journal of Antibiotics* **2020**, *74*, 161–175.
- (38) Schwalen, C. J.; Hudson, G. A.; Kille, B.; Mitchell, D. A. Bioinformatic Expansion and Discovery of Thiopeptide Antibiotics. *Journal of the American Chemical Society* **2018**, *140*, 9494–9501.

- (39) Hayashi, S.; Ozaki, T.; Asamizu, S.; Ikeda, H.; Ōmura, S.; Oku, N.; Igarashi, Y.; Tomoda, H.; Onaka, H. Genome Mining Reveals a Minimum Gene Set for the Biosynthesis of 32-Membered Macrocyclic Thiopeptides Lactazoles. *Chemistry & Biology* **2014**, *21*, 679–688.
- (40) Vinogradov, A. A.; Suga, H. Introduction to Thiopeptides: Biological Activity, Biosynthesis, and Strategies for Functional Reprogramming. *Cell Chemical Biology* **2020**, *27*, 1032–1051.
- (41) Hudson, G. A.; Hooper, A. R.; DiCaprio, A. J.; Sarlah, D.; Mitchell, D. A. Structure Prediction and Synthesis of Pyridine-Based Macrocyclic Peptide Natural Products. *Organic Letters* **2020**, *23*, 253–256.
- (42) Vinogradov, A. A.; Nagai, E.; Chang, J. S.; Narumi, K.; Onaka, H.; Goto, Y.; Suga, H. Accurate Broadcasting of Substrate Fitness for Lactazole Biosynthetic Pathway from Reactivity-Profiling mRNA Display. *Journal of the American Chemical Society* **2020**, *142*, 20329–20334.
- (43) Vinogradov, A. A.; Nagano, M.; Goto, Y.; Suga, H. Site-Specific Nonenzymatic Peptide S/O-Glutamylation Reveals the Extent of Substrate Promiscuity in Glutamate Elimination Domains. *Journal of the American Chemical Society* **2021**, *143*, 13358–13369.
- (44) Vinogradov, A. A.; Shimomura, M.; Kano, N.; Goto, Y.; Onaka, H.; Suga, H. Promiscuous Enzymes Cooperate at the Substrate Level En Route to Lactazole A. *Journal of the American Chemical Society* **2020**, *142*, 13886–13897.
- (45) Vinogradov, A. A.; Shimomura, M.; Goto, Y.; Ozaki, T.; Asamizu, S.; Sugai, Y.; Suga, H.; Onaka, H. Minimal lactazole scaffold for in vitro thiopeptide bioengineering. *Nature Communications* **2020**, *11*, 2272.
- (46) Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. Pro-

- ceedings of the 34th International Conference on Machine Learning. 2017; pp 3319–3328.
- (47) Desiere, F. The PeptideAtlas project. *Nucleic Acids Research* **2006**, *34*, D655–D658.
- (48) Chang, J. S.; Vinogradov, A. A.; Zhang, Y.; Goto, Y.; Suga, H. Deep Learning-Driven Library Design for the De Novo Discovery of Bioactive Thiopeptides. *ACS Central Science* **2023**, *9*, 2150–2160.
- (49) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, preprint, DOI:10.48550/ARXIV.1810.04805.
- (50) Wang, H.; Li, J.; Wu, H.; Hovy, E.; Sun, Y. Pre-Trained Language Models and Their Applications. *Engineering* **2023**, *25*, 51–65.
- (51) 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. *bioRxiv* **2021**, preprint, DOI:10.1101/2021.07.09.450648.
- (52) Rao, R.; Meier, J.; Sercu, T.; Ovchinnikov, S.; Rives, A. Transformer protein language models are unsupervised structure learners. *bioRxiv* **2020**, preprint, DOI:10.1101/2020.12.15.422761.
- (53) Bepler, T.; Berger, B. Learning the protein language: Evolution, structure, and function. *Cell Systems* **2021**, *12*, 654–669.e3.
- (54) Marquet, C.; Heinzinger, M.; Olenyi, T.; Dallago, C.; Erckert, K.; Bernhofer, M.; Nechaev, D.; Rost, B. Embeddings from protein language models predict conservation and variant effects. *Human Genetics* **2021**, *141*, 1629–1647.
- (55) Weissenow, K.; Heinzinger, M.; Rost, B. Protein language-model embeddings for fast,

- accurate, and alignment-free protein structure prediction. *Structure* **2022**, *30*, 1169–1177.e4.
- (56) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
- (57) Howard, J.; Ruder, S. Universal Language Model Fine-tuning for Text Classification. *arXiv* **2018**, preprint, DOI:10.48550/ARXIV.1801.06146.
- (58) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv* **2022**, preprint, DOI:10.1101/2022.07.20.500902.
- (59) Vig, J.; Madani, A.; Varshney, L. R.; Xiong, C.; Socher, R.; Rajani, N. F. BERTology Meets Biology: Interpreting Attention in Protein Language Models. *arXiv* **2020**, preprint, DOI:10.48550/ARXIV.2006.15222.
- (60) Starr, T. N.; Thornton, J. W. Epistasis in protein evolution. *Protein Science* **2016**, *25*, 1204–1218.

Supporting information: Substrate Prediction for RiPP Biosynthetic Enzymes via Masked Language Modeling and Transfer Learning

Joseph D. Clark,[†] Xuenan Mi,[‡] Douglas A. Mitchell,[¶] and Diwakar Shukla^{*,‡,§,||}

[†]*School of Molecular and Cellular Biology, University of Illinois at
Urbana-Champaign, Urbana, IL 61801, USA*

[‡]*Center for Biophysics and Quantitative Biology, University of Illinois at
Urbana-Champaign, Urbana, IL 61801, USA*

[¶]*Department of Chemistry, University of Illinois at Urbana-Champaign, Urbana, IL 61801,
USA*

[§]*Department of Chemical and Biomolecular Engineering, University of Illinois at
Urbana-Champaign, Urbana, IL 61801, USA*

^{||}*Department of Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL
61801, USA*

E-mail: diwakar@illinois.edu

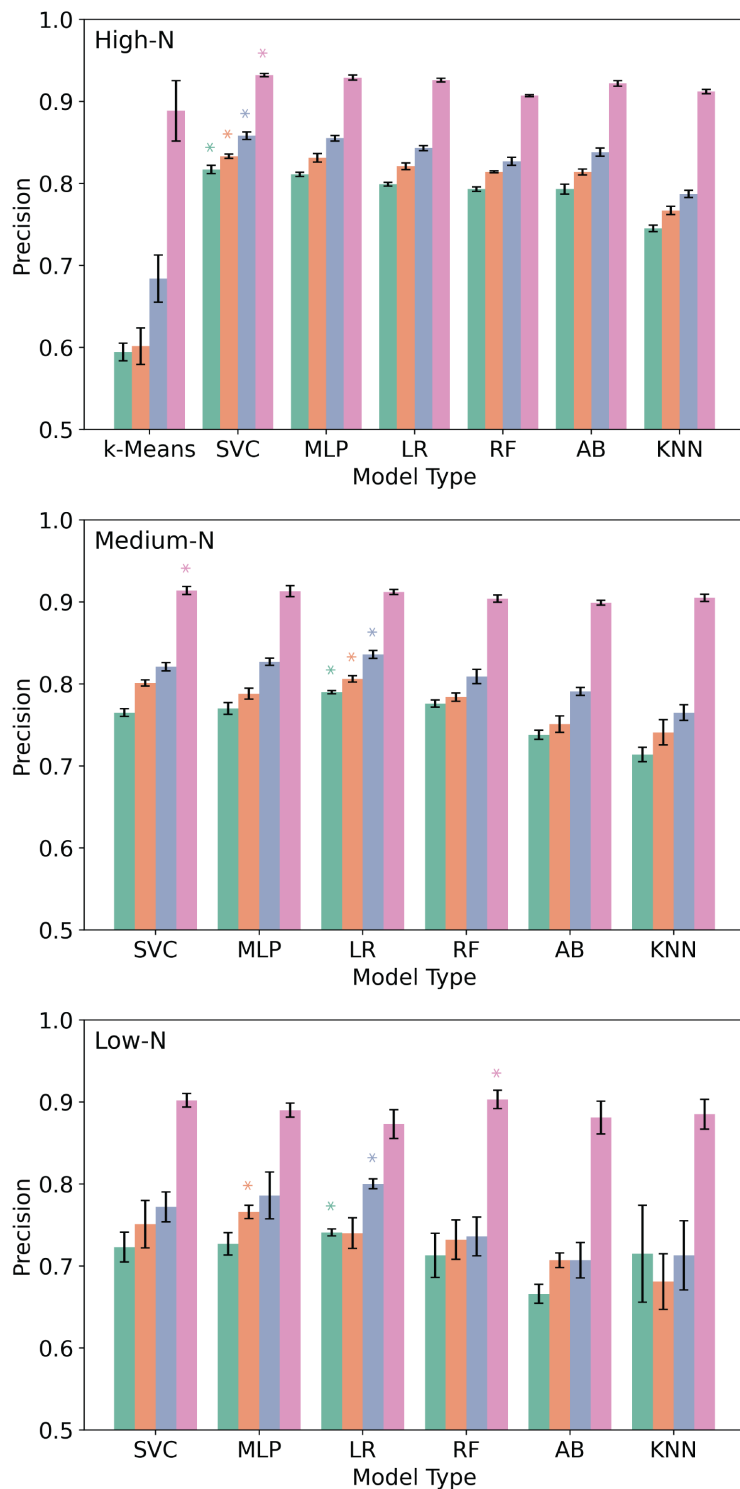


Figure S1: Precision of LazDEF substrate classification models trained on embeddings from Vanilla-ESM (green), ESMA trained on a subset of PeptideAtlas (orange), ESMA trained on LazBF substrates/non-substrates (blue), and ESMA trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

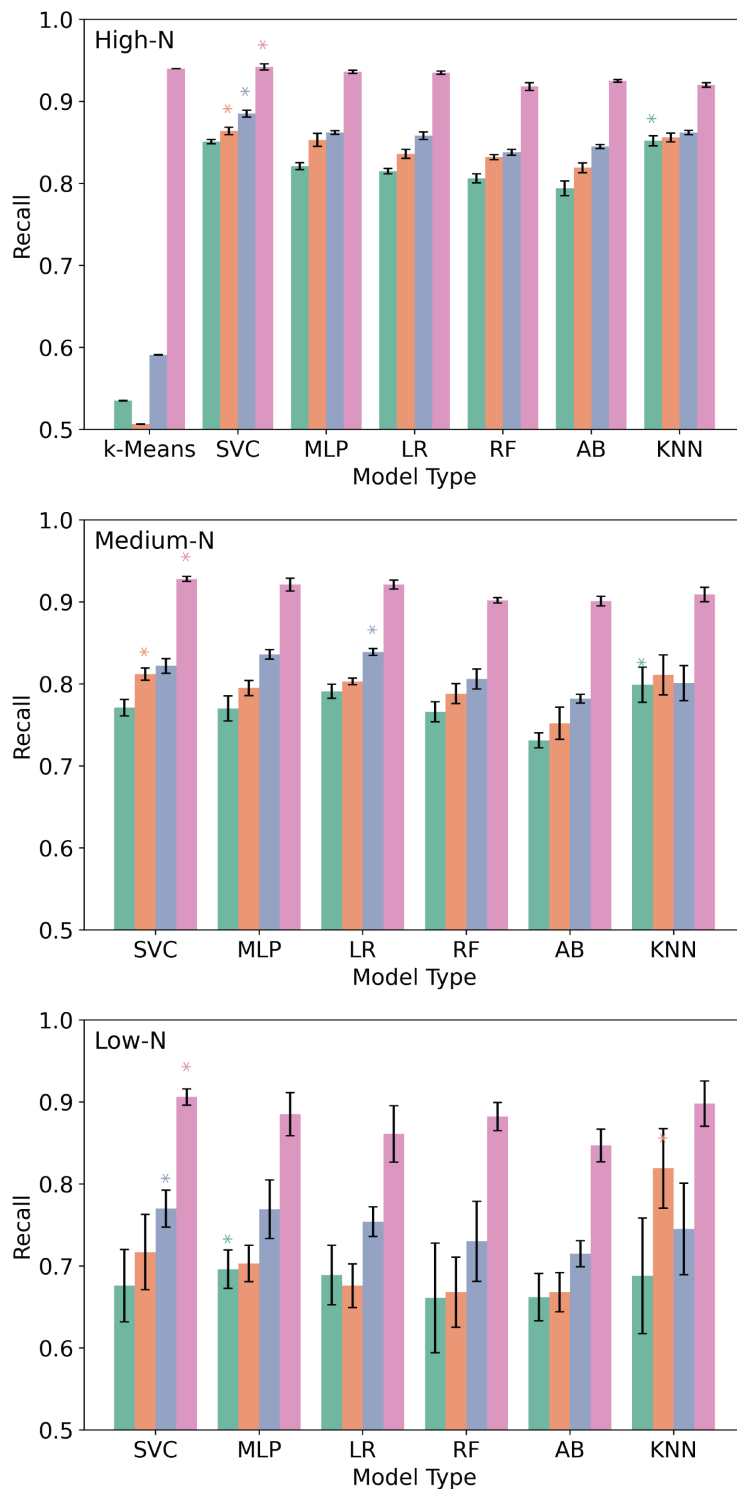


Figure S2: Recall of LazDEF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

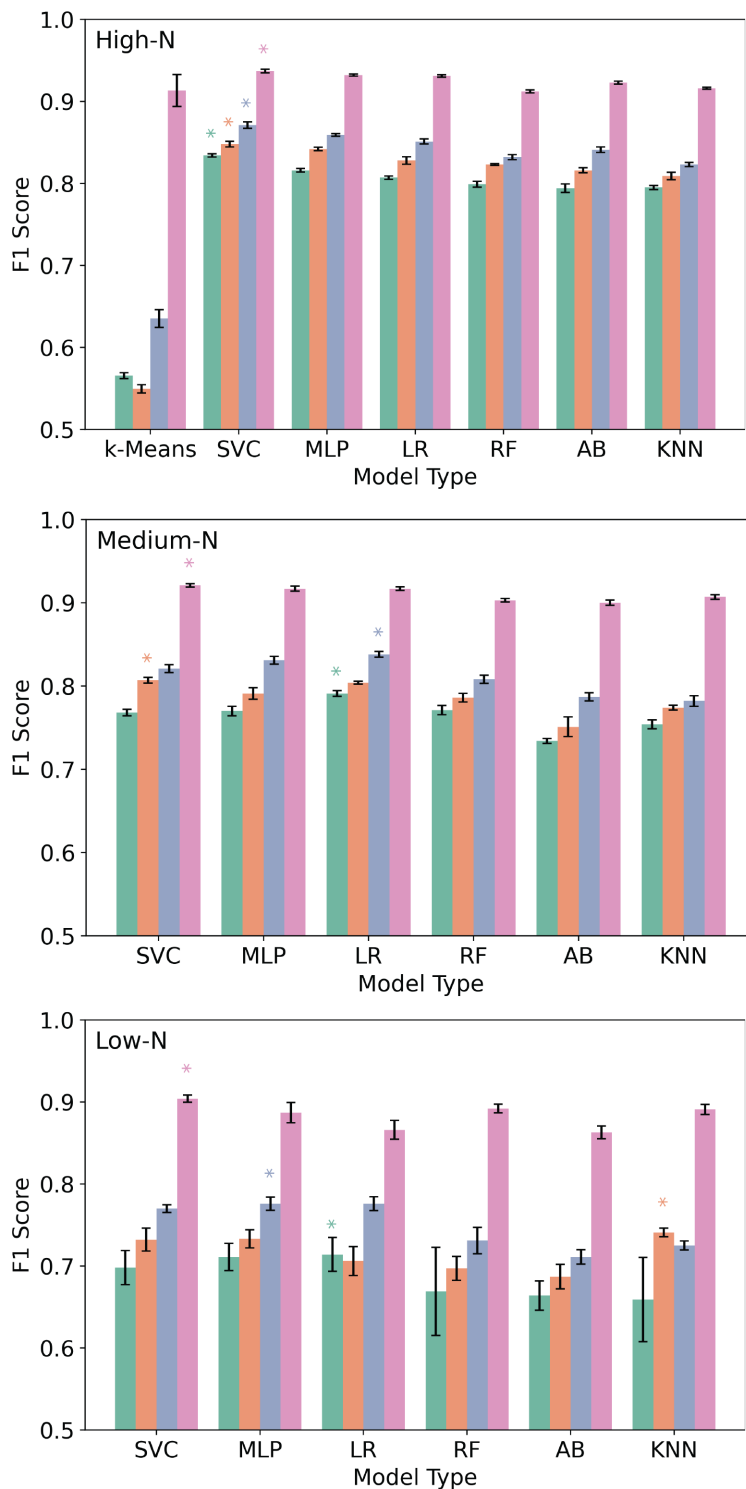


Figure S3: F1 score of LazDEF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

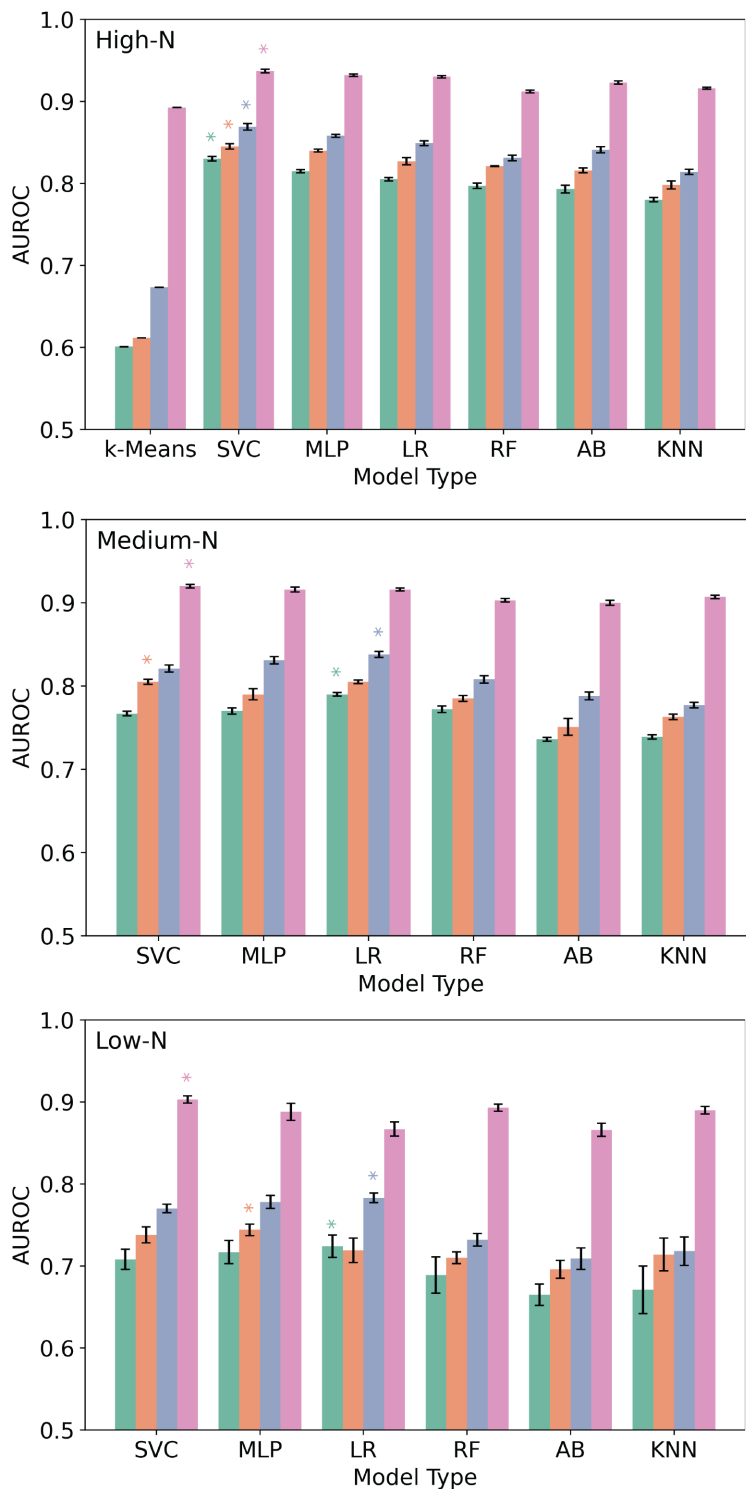


Figure S4: AUROC of LazDEF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

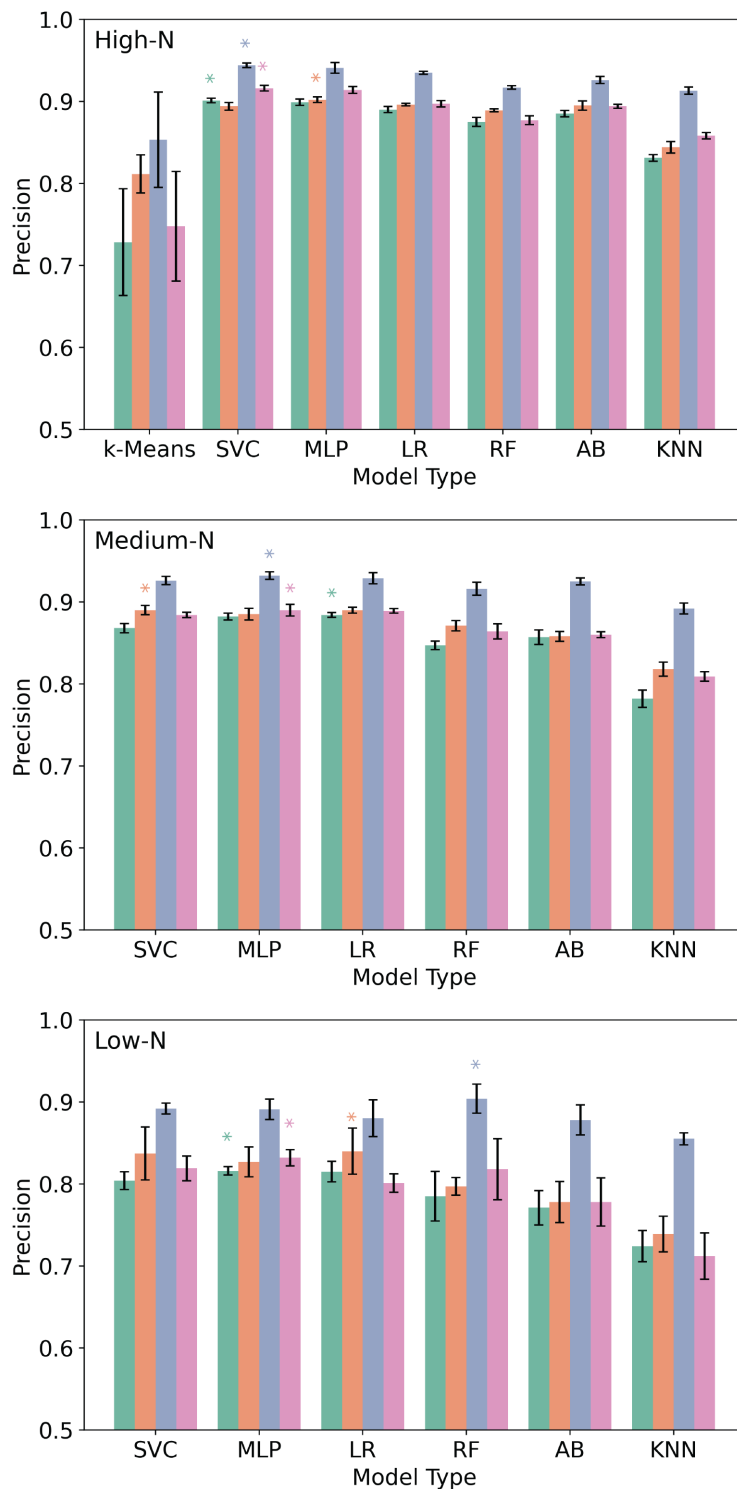


Figure S5: Precision of LazBF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

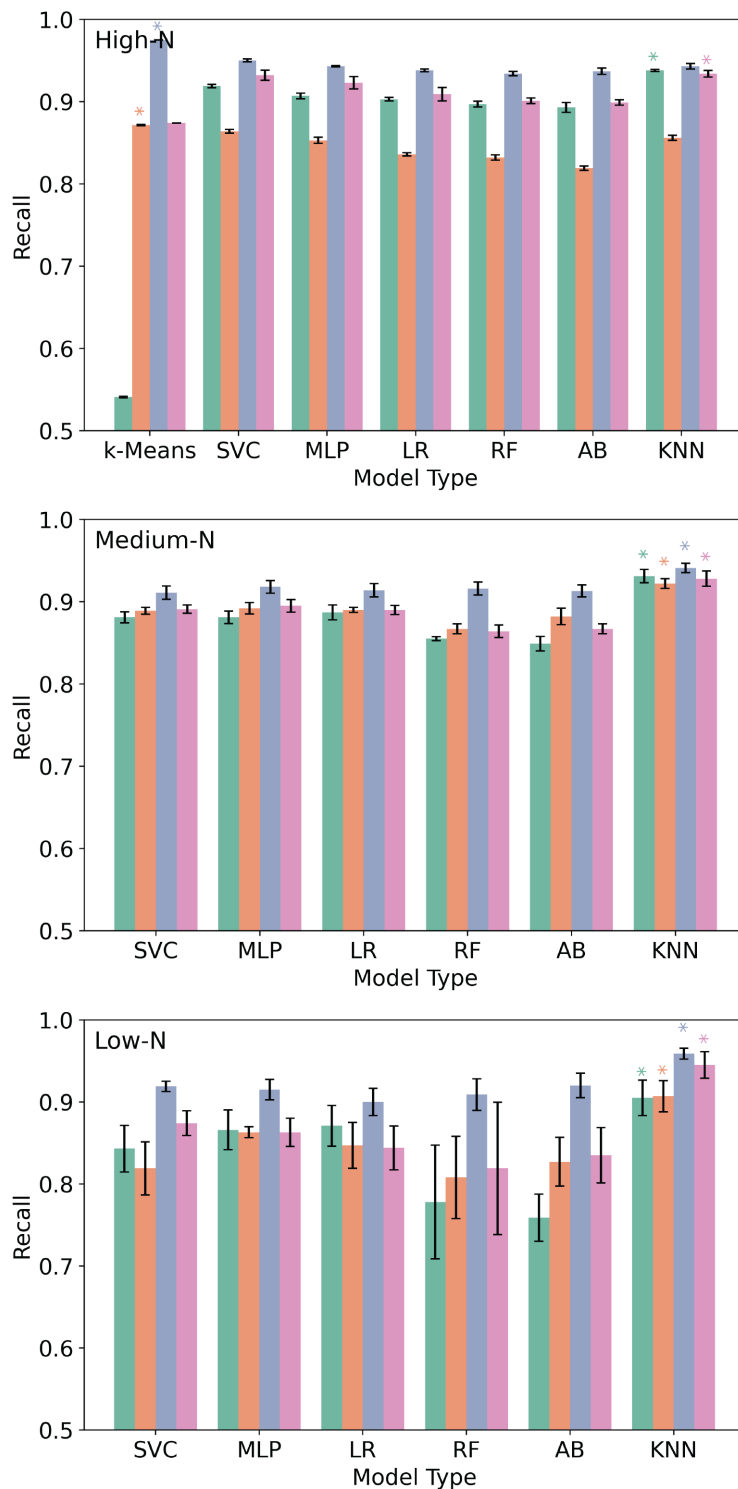


Figure S6: Recall of LazBF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

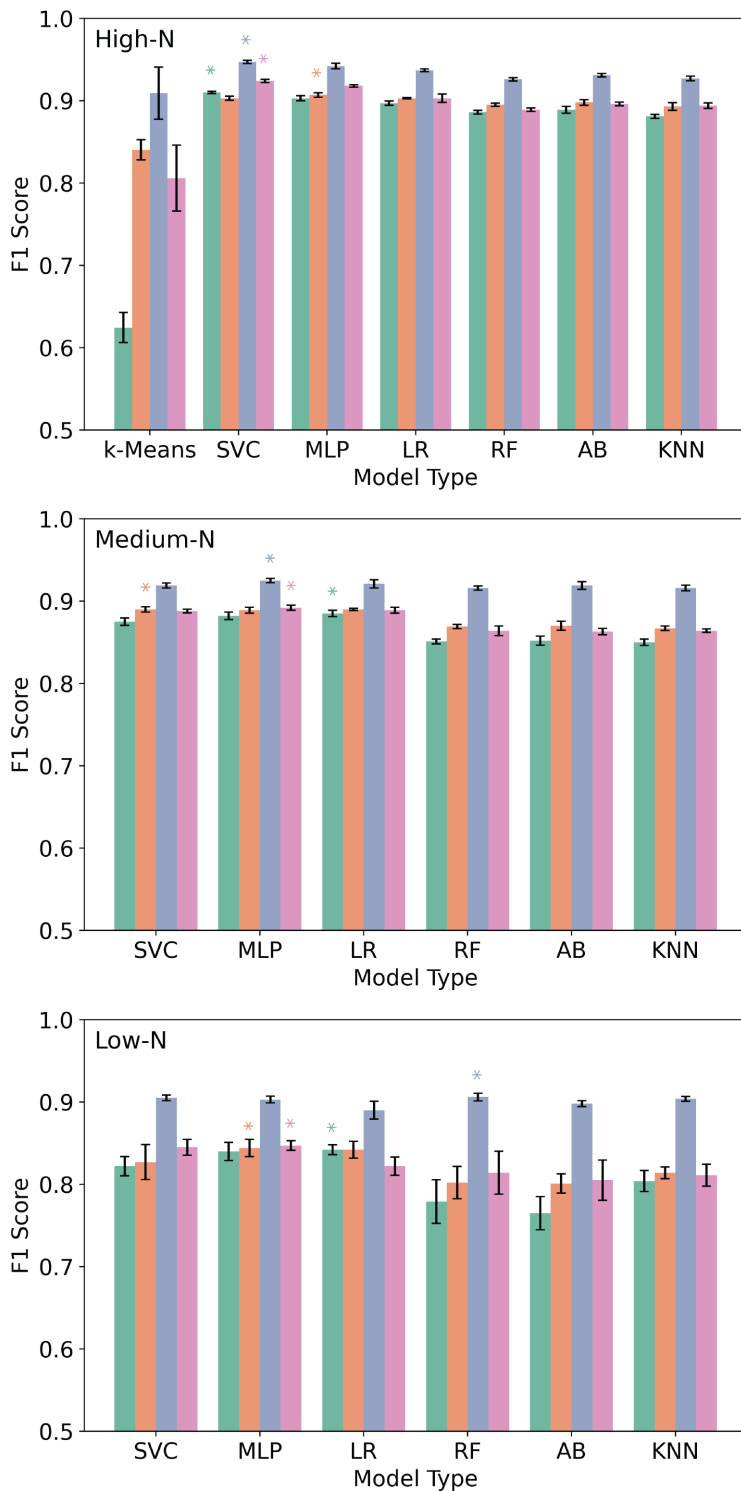


Figure S7: F1 score of LazBF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

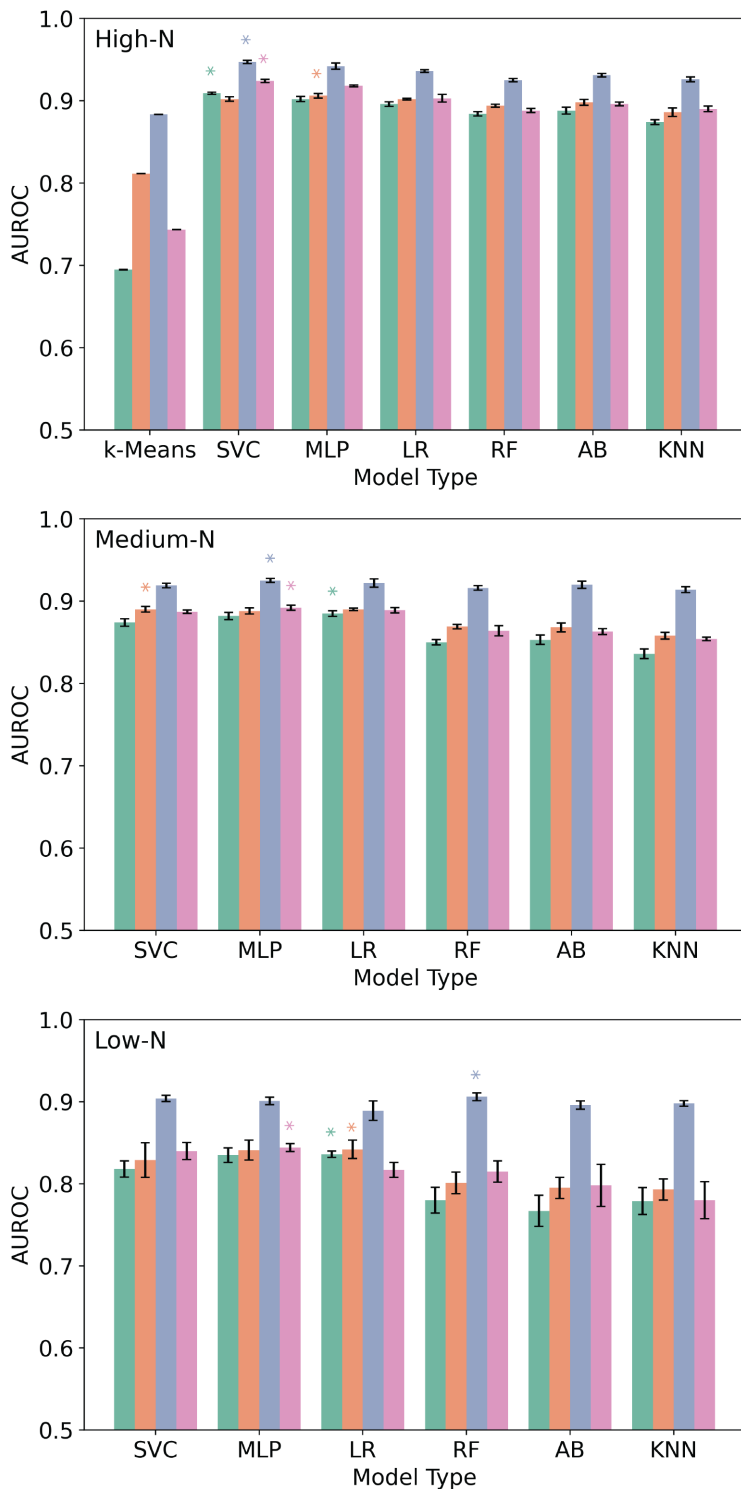


Figure S8: AUROC of LazBF substrate classification models trained on embeddings from Vanilla-ESM (green), ESM trained on a subset of PeptideAtlas (orange), ESM trained on LazBF substrates/non-substrates (blue), and ESM trained on LazDEF substrates/non-substrates (pink) in the a) high-N condition, b) medium-N condition, and c) low-N condition. A star indicates the top performing model for each set of embeddings.

Table S1: The optimal hyperparameters for each downstream model type trained on each set of embeddings for the high-N condition.

Embedding	Downstream Model	Hyperparameters
LazBF Vanilla-ESM	SVC	C=5, kernel='rbf'
LazBF Vanilla-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazBF Vanilla-ESM	LR	C= 0.1, penalty='l2'
LazBF Vanilla-ESM	RF	n_estimators=500, criterion='log_loss'
LazBF Vanilla-ESM	AB	learning_rate=1, n_estimators=500
LazBF Vanilla-ESM	KNN	n_neighbors=25, weights='distance'
LazDEF Vanilla-ESM	SVC	C=5, kernel='rbf'
LazDEF Vanilla-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazDEF Vanilla-ESM	LR	C=0.1, penalty='l2'
LazDEF Vanilla-ESM	RF	n_estimators=500, criterion='gini'
LazDEF Vanilla-ESM	AB	learning_rate=1, n_estimators=500
LazDEF Vanilla-ESM	KNN	n_neighbors=50, weights='uniform'
LazBF LazBF-ESM	SVC	C=10, kernel='rbf'
LazBF LazBF-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazBF LazBF-ESM	LR	C=0.1, penalty=None
LazBF LazBF-ESM	RF	n_estimators=100, criterion='gini'
LazBF LazBF-ESM	AB	learning_rate=0.1, n_estimators=500
LazBF LazBF-ESM	KNN	n_neighbors=25, weights='distance'
LazDEF LazBF-ESM	SVC	C=5, kernel='rbf'
LazDEF LazBF-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazDEF LazBF-ESM	LR	C=0.1, penalty='l2'
LazDEF LazBF-ESM	RF	n_estimators=500, criterion='log_loss'
LazDEF LazBF-ESM	AB	learning_rate=1, n_estimators=500
LazDEF LazBF-ESM	KNN	n_neighbors=50, weights='distance'
LazBF LazDEF-ESM	SVC	C=5, kernel='rbf'
LazBF LazDEF-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazBF LazDEF-ESM	LR	C=0.1, penalty=None
LazBF LazDEF-ESM	RF	n_estimators=500, criterion='log_loss'
LazBF LazDEF-ESM	AB	learning_rate=1, n_estimators=500
LazBF LazDEF-ESM	KNN	n_neighbors=50, weights='uniform'
LazDEF LazDEF-ESM	SVC	C=1, kernel='rbf'
LazDEF LazDEF-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazDEF LazDEF-ESM	LR	C=10, penalty=None
LazDEF LazDEF-ESM	RF	n_estimators=500, criterion='gini'
LazDEF LazDEF-ESM	AB	learning_rate=1, n_estimators=500
LazDEF LazDEF-ESM	KNN	n_neighbors=50, weights='uniform'

Table S2: The optimal hyperparameters for each downstream model type trained on each set of embeddings for the medium-N condition.

Embedding	Downstream Model	Hyperparameters
LazBF Vanilla-ESM	SVC	C=10, kernel='linear'
LazBF Vanilla-ESM	MLP	hidden_layer_sizes=500, activation='relu'
LazBF Vanilla-ESM	LR	C=10, penalty=None
LazBF Vanilla-ESM	RF	n_estimators=200, criterion='gini'
LazBF Vanilla-ESM	AB	learning_rate=1, n_estimators=200
LazBF Vanilla-ESM	KNN	n_neighbors=50, weights='distance'
LazDEF Vanilla-ESM	SVC	C=1, kernel='rbf'
LazDEF Vanilla-ESM	MLP	hidden_layer_sizes=100, activation='relu'
LazDEF Vanilla-ESM	LR	C=0.1, penalty=None
LazDEF Vanilla-ESM	RF	n_estimators=200, criterion='entropy'
LazDEF Vanilla-ESM	AB	learning_rate=0.1, n_estimators=500
LazDEF Vanilla-ESM	KNN	n_neighbors=50, weights='uniform'
LazBF LazBF-ESM	SVC	C=0.1, kernel='linear'
LazBF LazBF-ESM	MLP	hidden_layer_sizes=50, activation='tanh'
LazBF LazBF-ESM	LR	C=0.1, penalty='l2'
LazBF LazBF-ESM	RF	n_estimators=200, criterion='entropy'
LazBF LazBF-ESM	AB	learning_rate=0.1, n_estimators=200
LazBF LazBF-ESM	KNN	n_neighbors=25, weights='uniform'
LazDEF LazBF-ESM	SVC	C=1, kernel='linear'
LazDEF LazBF-ESM	MLP	hidden_layer_sizes=100, activation='tanh'
LazDEF LazBF-ESM	LR	C=0.1, penalty='l2'
LazDEF LazBF-ESM	RF	n_estimators=500, criterion='gini'
LazDEF LazBF-ESM	AB	learning_rate=1, n_estimators=200
LazDEF LazBF-ESM	KNN	n_neighbors=25, weights='uniform'
LazBF LazDEF-ESM	SVC	C=0.1, kernel='linear'
LazBF LazDEF-ESM	MLP	hidden_layer_sizes=500, activation='relu'
LazBF LazDEF-ESM	LR	C=0.1, penalty='l2'
LazBF LazDEF-ESM	RF	n_estimators=500, criterion='log_loss'
LazBF LazDEF-ESM	AB	learning_rate=1, n_estimators=200
LazBF LazDEF-ESM	KNN	n_neighbors=50, weights='uniform'
LazDEF LazDEF-ESM	SVC	C=1, kernel='rbf'
LazDEF LazDEF-ESM	MLP	hidden_layer_sizes=100, activation='relu'
LazDEF LazDEF-ESM	LR	C=5, penalty=None
LazDEF LazDEF-ESM	RF	n_estimators=50, criterion='entropy'
LazDEF LazDEF-ESM	AB	learning_rate=0.1, n_estimators=500
LazDEF LazDEF-ESM	KNN	n_neighbors=25, weights='uniform'

Table S3: The optimal hyperparameters for each downstream model type trained on each set of embeddings for the low-N condition.

Embedding	Downstream Model	Hyperparameters
LazBF Vanilla-ESM	SVC	C=5, kernel='rbf'
LazBF Vanilla-ESM	MLP	hidden_layer_sizes=100, activation='relu'
LazBF Vanilla-ESM	LR	C=0.1, penalty=None
LazBF Vanilla-ESM	RF	n_estimators=100, criterion='entropy'
LazBF Vanilla-ESM	AB	learning_rate=1, n_estimators=500
LazBF Vanilla-ESM	KNN	n_neighbors=5, weights='uniform'
LazDEF Vanilla-ESM	SVC	C=0.1, kernel='linear'
LazDEF Vanilla-ESM	MLP	hidden_layer_sizes=1000, activation='relu'
LazDEF Vanilla-ESM	LR	C=0.1, penalty='l2'
LazDEF Vanilla-ESM	RF	n_estimators=100, criterion='entropy'
LazDEF Vanilla-ESM	AB	learning_rate=1, n_estimators=200
LazDEF Vanilla-ESM	KNN	n_neighbors=10, weights='uniform'
LazBF LazBF-ESM	SVC	C=0.1, kernel='rbf'
LazBF LazBF-ESM	MLP	hidden_layer_sizes=500, activation='relu'
LazBF LazBF-ESM	LR	C=0.1, penalty='l2'
LazBF LazBF-ESM	RF	n_estimators=200, criterion='entropy'
LazBF LazBF-ESM	AB	learning_rate=5, n_estimators=200
LazBF LazBF-ESM	KNN	n_neighbors=5, weights='uniform'
LazDEF LazBF-ESM	SVC	C=5, kernel='rbf'
LazDEF LazBF-ESM	MLP	hidden_layer_sizes=750, activation='relu'
LazDEF LazBF-ESM	LR	C=0.1, penalty='l2'
LazDEF LazBF-ESM	RF	n_estimators=200, criterion='gini'
LazDEF LazBF-ESM	AB	learning_rate=1, n_estimators=500
LazDEF LazBF-ESM	KNN	n_neighbors=50, weights='distance'
LazBF LazDEF-ESM	SVC	C=0.1, kernel='linear'
LazBF LazDEF-ESM	MLP	hidden_layer_sizes=750, activation='relu'
LazBF LazDEF-ESM	LR	C=0.1, penalty='l2'
LazBF LazDEF-ESM	RF	n_estimators=50, criterion='log_loss'
LazBF LazDEF-ESM	AB	learning_rate=1, n_estimators=500
LazBF LazDEF-ESM	KNN	n_neighbors=10, weights='uniform'
LazDEF LazDEF-ESM	SVC	C=1, kernel='rbf'
LazDEF LazDEF-ESM	MLP	hidden_layer_sizes=500, activation='tanh'
LazDEF LazDEF-ESM	LR	C=0.1, penalty='l2'
LazDEF LazDEF-ESM	RF	n_estimators=200, criterion='log_loss'
LazDEF LazDEF-ESM	AB	learning_rate=0.1, n_estimators=50
LazDEF LazDEF-ESM	KNN	n_neighbors=50, weights='uniform'

Table S4: The optimal hyperparameters for each downstream model type trained on Peptide-ESM embeddings for the low-N, med-N, and high-N conditions.

Embedding	Downstream Model	Hyperparameters
LazBF Peptide-ESM low-N	SVC	C=0.1, kernel='linear'
LazBF Peptide-ESM low-N	MLP	hidden_layer_sizes=750, activation='relu'
LazBF Peptide-ESM low-N	LR	C=1, penalty=None
LazBF Peptide-ESM low-N	RF	n_estimators=50, criterion='entropy'
LazBF Peptide-ESM low-N	AB	learning_rate=1, n_estimators=50
LazBF Peptide-ESM low-N	KNN	n_neighbors=10, weights='uniform'
LazDEF Peptide-ESM low-N	SVC	C=1, kernel='linear'
LazDEF Peptide-ESM low-N	MLP	hidden_layer_sizes=500, activation='relu'
LazDEF Peptide-ESM low-N	LR	C=1, penalty='None'
LazDEF Peptide-ESM low-N	RF	n_estimators=200, criterion='gini'
LazDEF Peptide-ESM low-N	AB	learning_rate=1, n_estimators=200
LazDEF Peptide-ESM low-N	KNN	n_neighbors=25, weights='uniform'
LazBF Peptide-ESM med-N	SVC	C=1, kernel='linear'
LazBF Peptide-ESM med-N	MLP	hidden_layer_sizes=500, activation='tanh'
LazBF Peptide-ESM med-N	LR	C=1, penalty=None
LazBF Peptide-ESM med-N	RF	n_estimators=500, criterion='entropy'
LazBF Peptide-ESM med-N	AB	learning_rate=0.1, n_estimators=200
LazBF Peptide-ESM med-N	KNN	n_neighbors=10, weights='uniform'
LazDEF Peptide-ESM med-N	SVC	C=0.1, kernel='linear'
LazDEF Peptide-ESM med-N	MLP	hidden_layer_sizes=1000, activation='relu'
LazDEF Peptide-ESM med-N	LR	C=0.1, penalty='None'
LazDEF Peptide-ESM med-N	RF	n_estimators=200, criterion='entropy'
LazDEF Peptide-ESM med-N	AB	learning_rate=1, n_estimators=500
LazDEF Peptide-ESM med-N	KNN	n_neighbors=25, weights='distance'
LazBF Peptide-ESM high-N	SVC	C=0.1, kernel='linear'
LazBF Peptide-ESM high-N	MLP	hidden_layer_sizes=750, activation='tanh'
LazBF Peptide-ESM high-N	LR	C=5, penalty=None
LazBF Peptide-ESM high-N	RF	n_estimators=500, criterion='entropy'
LazBF Peptide-ESM high-N	AB	learning_rate=1, n_estimators=500
LazBF Peptide-ESM high-N	KNN	n_neighbors=50, weights='uniform'
LazDEF Peptide-ESM high-N	SVC	C=5, kernel='linear'
LazDEF Peptide-ESM high-N	MLP	hidden_layer_sizes=50, activation='relu'
LazDEF Peptide-ESM high-N	LR	C=1, penalty='None'
LazDEF Peptide-ESM high-N	RF	n_estimators=500, criterion='log loss'
LazDEF Peptide-ESM high-N	AB	learning_rate=1, n_estimators=500
LazDEF Peptide-ESM high-N	KNN	n_neighbors=50, weights='uniform'

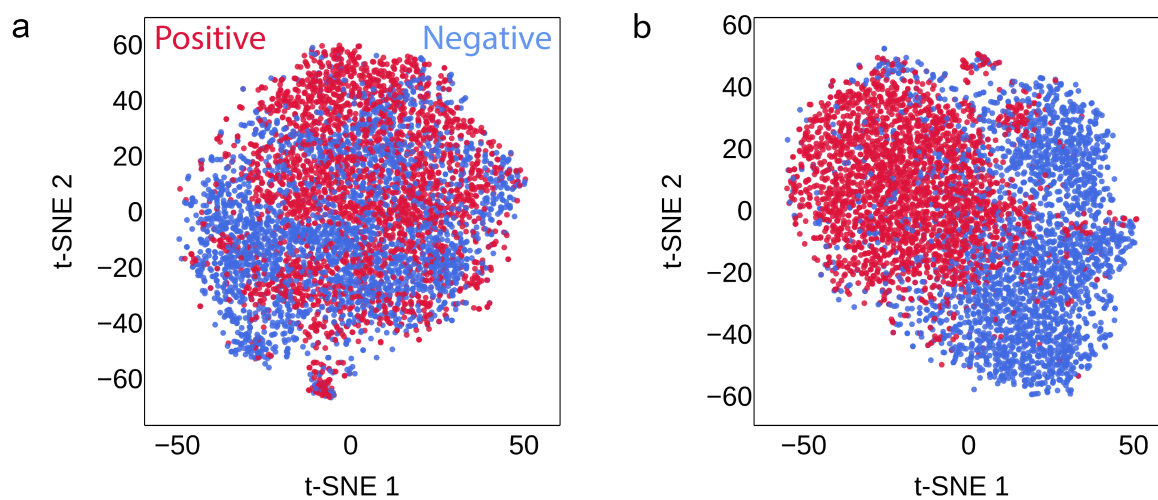


Figure S9: t-SNE visualization of the embedding space from ESM trained on a subset of PeptideAtlas for a) LazDEF substrates/non-substrates, and b) LazBF substrates/non-substrates. Substrates are red and non-substrates samples are blue.

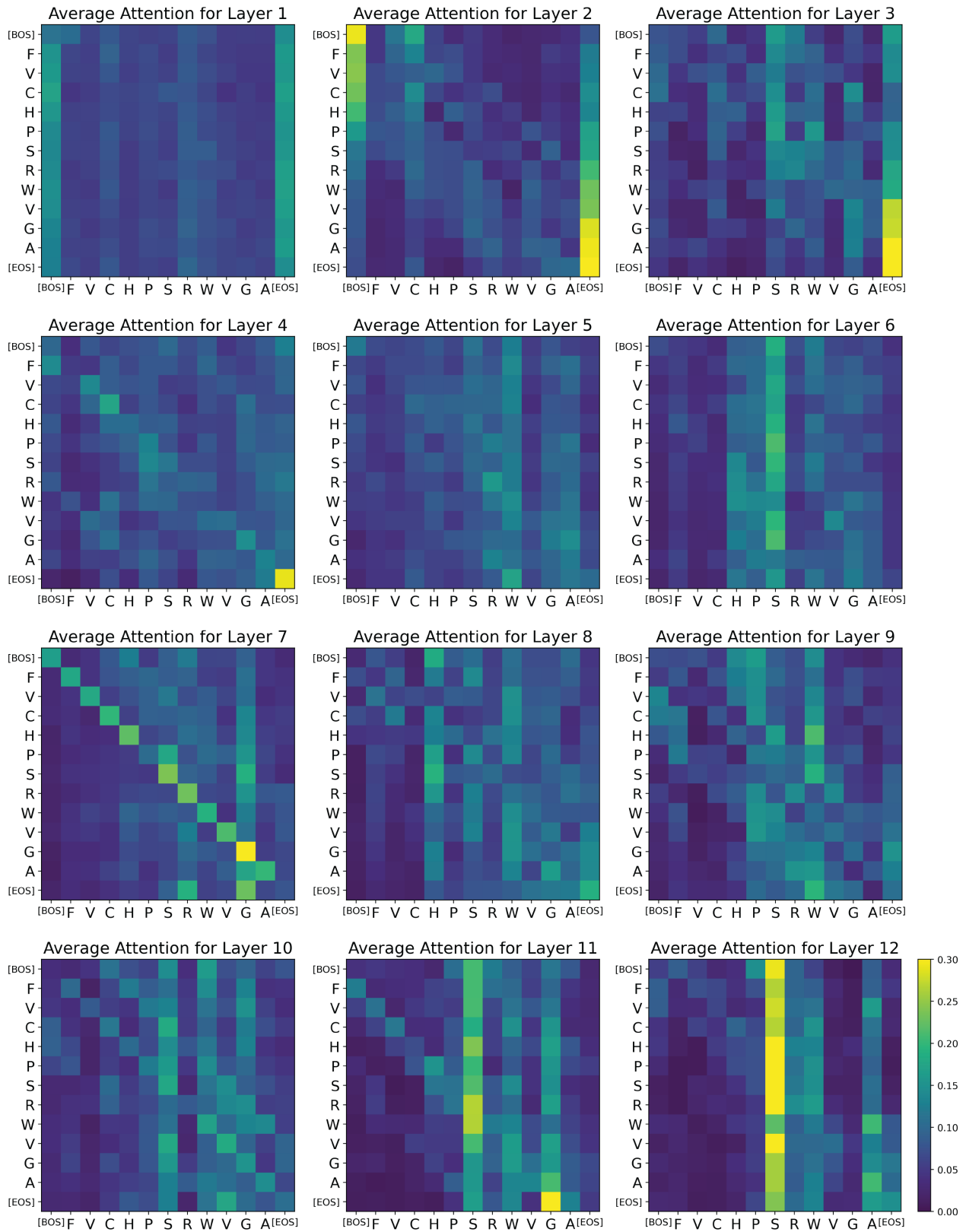


Figure S10: The average attention for all 12 layers of the fine-tuned LazBF-ESM for the LazBF substrate FVCHPSRWVGA.