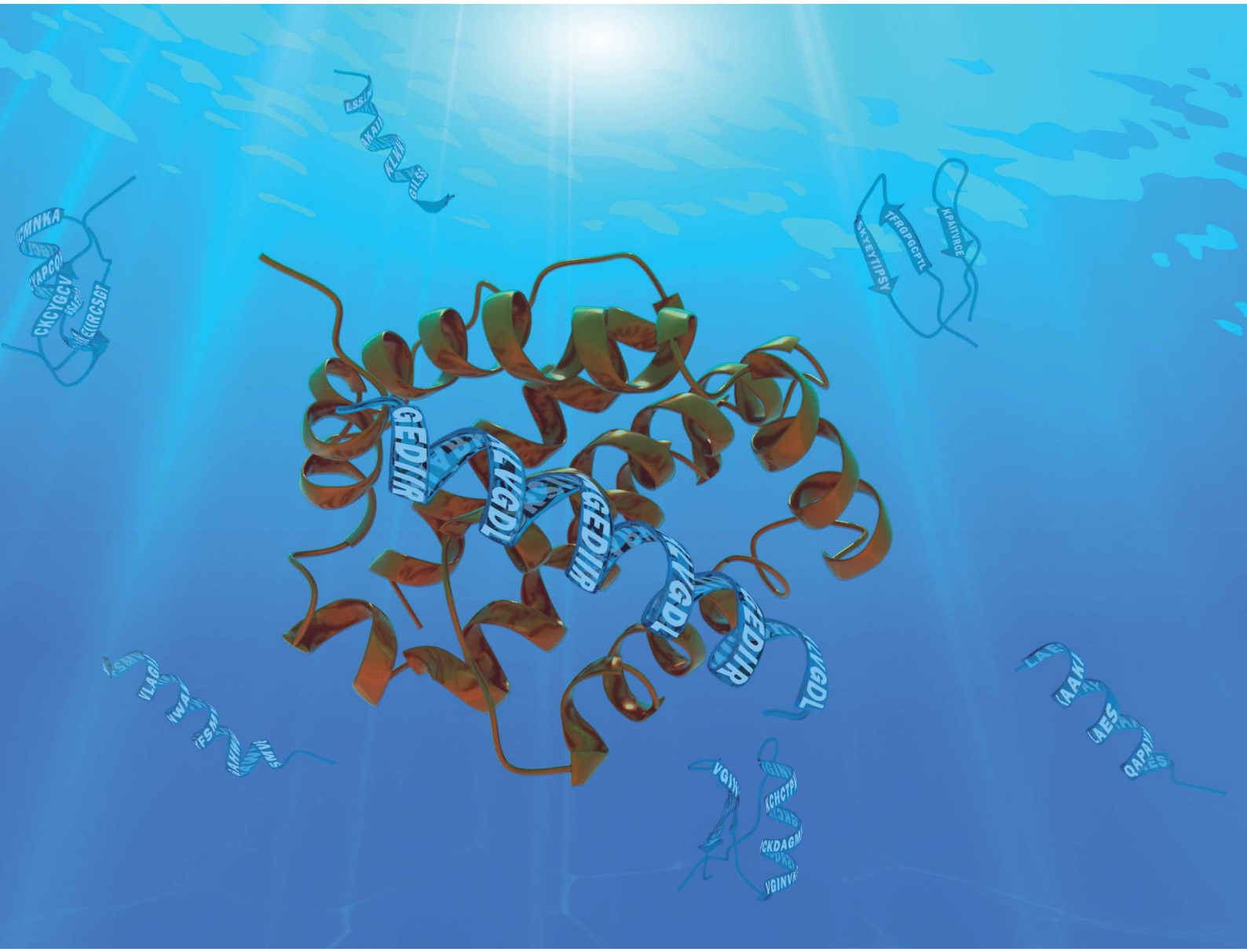
Volume 4
Number 2
February 2025
Pages 291-574

Digital Discovery

rsc.li/digitaldiscovery



ISSN 2635-098X

ROYAL SOCIETY OF CHEMISTRY

PAPER

Diwakar Shukla *et al.*

Substrate prediction for RiPP biosynthetic enzymes *via* masked language modeling and transfer learning

Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2025, **4**, 343

Substrate prediction for RiPP biosynthetic enzymes *via* masked language modeling and transfer learning†

Joseph D. Clark, [a] Xuenan Mi, [b] Douglas A. Mitchell [cd] and Diwakar Shukla [*befg]

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 the specificity of RiPP biosynthetic enzymes. 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 prediction of both LazBF and LazDEF substrates. Similarly, masked language modeling of LazDEF substrate preferences produced embeddings that improved prediction 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. We found that a single high-quality data set of substrates and non-substrates for a RiPP biosynthetic enzyme improved substrate prediction for distinct enzymes 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.

Received 20th June 2024
Accepted 28th November 2024

DOI: 10.1039/d4dd00170b

rsc.li/digitaldiscovery

1 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 (Fig. 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 their uninterpretable substrate preferences 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

[a]*School of Molecular and Cellular Biology, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA*
[b]*Center for Biophysics and Quantitative Biology, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA*
[c]*Department of Biochemistry, Vanderbilt University School of Medicine, Nashville, TN, 37232, USA*
[d]*Department of Chemistry, Vanderbilt University, Nashville, TN, 37232, USA*
[e]*Department of Chemical and Biomolecular Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA. E-mail: diwakar@illinois.edu*
[f]*Department of Bioengineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA*
[g]*Department of Chemistry, University of Illinois at Urbana-Chamapaign, Urbana, IL, 61801, USA*

† Electronic supplementary information (ESI) available: Fig. S1–S7 and Tables S1–S11. See DOI: https://doi.org/10.1039/d4dd00170b

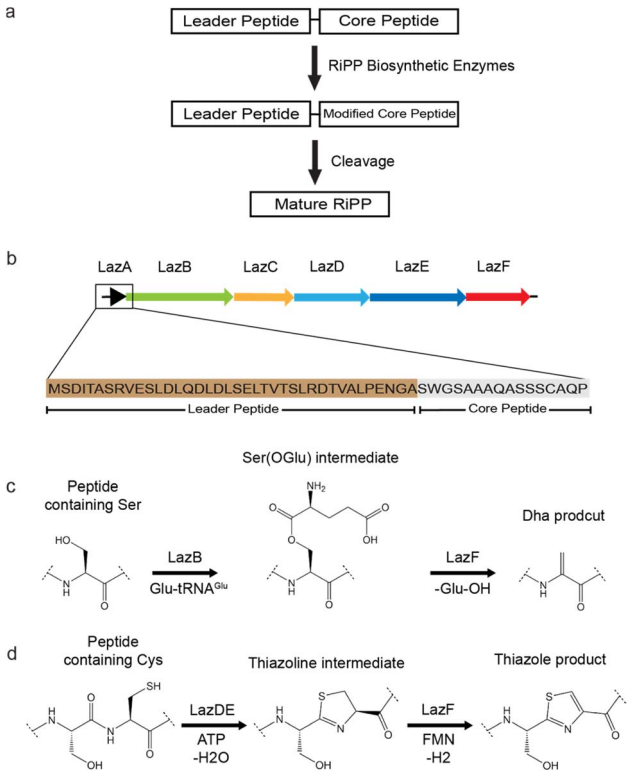


Fig. 1 (a) The generic biosynthesis pathway of RiPPs. RiPP precursor peptides contain a leader peptide and a core peptide. After post-translational modifications in the core peptide, the leader peptide is cleaved. (b) The lactazole biosynthetic gene cluster contains six proteins. LazA is the precursor 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.

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 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] 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 enhance data efficiency in situations with low data availability and accelerate the design of natural products with desired properties. 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 (Fig. 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 substrates 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 residue 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. Another language model trained on substrates and non-substrates for all 5 lactazole biosynthetic enzymes showed improved ability to predict the substrates of LazBF and LazDEF individually. Embeddings from our RiPP biosynthetic enzyme-specific language models also outperformed embeddings from a baseline peptide language model trained on a diverse collection peptides from UniRef50 (ref. 47) and PeptideAtlas,[48] a repository 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 transferability 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. We expect interpretable machine learning models to uncover patterns describing RiPP biosynthetic enzyme substrate specificity that are often difficult to infer manually, thereby promoting the design of novel substrates. Fig. 2 presents a schematic representation of our overall workflow. Our results suggest that (1) some degree of features are shared between the substrate preferences 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 advance natural product engineering by leveraging large data sets and transfer learning to predict RiPP biosynthetic enzyme substrates.

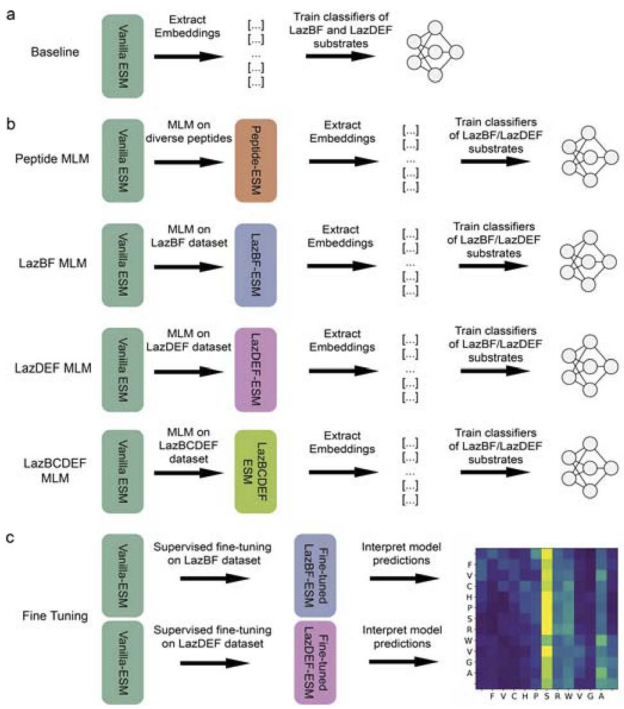


Fig. 2 A schematic representation of the workflow for masked language modeling (MLM) of LazBF and LazDEF substrate preferences. (a) LazBF and LazDEF substrate/non-substrate embeddings were extracted from the protein language model ESM-2 (Vanilla-ESM). The baseline performance of downstream classification models was assessed. (b) Peptide language models (Peptide-ESM, LazBF-ESM, LazDEF-ESM, LazBCDEF-ESM) were developed *via* masked language modeling of 4 peptide data sets. Embeddings were extracted and the performance of downstream substrate prediction models was compared to baseline. (c) Protein language 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).

2 Methods

2.1 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] As opposed to truly random sequences, the data sets contained high fitness substrates and low fitness non-substrates and were limited to the sequence space of peptides with either relatively high or low affinity for each enzyme.[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 random sample of 1.3 million sequences containing an equal number of substrates and non-substrates was selected. A random subset of 50 000 peptides

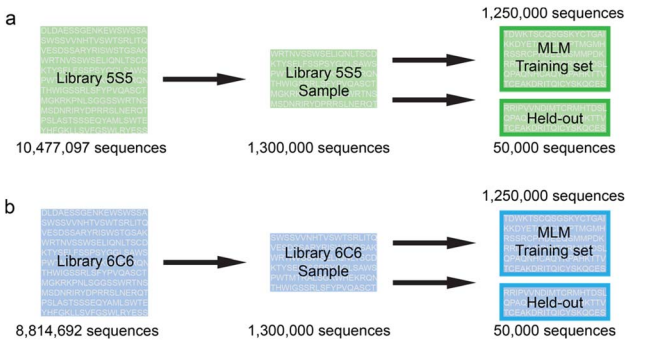


Fig. 3 A schematic representation of our data preprocessing pipeline. (a) LazA core sequences ($n = 1.3$ million) were selected from library 5S5 and used for masked language modeling (MLM) of LazBF substrate preferences. A 'held-out' data set of 50 000 peptides was set aside for downstream model training and evaluation. (b) LazA core sequences ($n = 1.3$ million) were selected from library 6C6 and used for masked language modeling (MLM) of LazDEF substrate preferences. A held-out data set of 50 000 peptides was set aside for downstream model training and evaluation.

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. Fig. 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 all 5 synthetases (LazBCDEF).[49] This study comprehensively profiled the combined substrate preferences of the entire biosynthetic pathway as opposed to individual enzymes. This data set was preprocessed in a manner identical to the LazBF/DEF substrate data sets.

2.2 Masked language modeling

Masked language modeling is a widely-used strategy for pretraining large language models.[50,51] 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.[52,53] 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 33-layer, 650 million parameter version of ESM-2 was used as a baseline model (Vanilla-ESM). 4 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. "LazBCDEF-ESM" was trained on 1.25 million LazA core peptide sequences from the LazBCDEF data set. The three models were each 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. "Peptide-ESM" was trained on 1 491 625 peptide sequences collected from UniRef50 (ref. 47) and Peptide Atlas.[48] Peptide-ESM underwent training for 2 epochs using a learning rate of 3×10^{-6} and a batch size of 256. Peptide-ESM was trained using GaLore, a memory efficient training strategy due to computational limitations at the time of training. All hyperparameters for additional training on peptide data are available in Table S4.†

2.3 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.[54,55] Protein language model representations, are commonly used as the input to downstream machine learning models trained on various protein and peptide prediction tasks.[56,57] 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 $L \times 1280$, where L was the length of the sequence. The last layer representation was averaged across the length dimension to obtain a single 1280-dimensional mean representation. The "beginning of sequence" ([BOS]) and "end of sequence" ([EOS]) token embeddings were included in the 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.[58] StandardScaler was applied to all embeddings following standard protocols prior to training.

2.3.1 Supervised classification models. Supervised learning models are trained by predicting properties of labeled data points (*e.g.*, substrate or non-substrate). Logistic regression (LR), 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 5 language models as input. Stratified 5-fold cross validation was performed for each model. For each fold, the accuracy between the ground truth labels and the predicted labels was calculated. The final model performance was described by the average metrics across 5 repeats of 5-fold cross validation. To emulate real-world scenarios in which training data is limited, each model type was trained and validated under 3 conditions. For each condition, a random subset of peptides was selected from the held-out data sets. In the "high-N" condition, 5-fold cross validation was performed with 1000 peptides. In the "medium-N" condition, 5-fold cross validation was performed with 500 peptides. In the "low-N" condition, 5-fold cross validation was performed with only 200 peptides. Hyperparameters of each supervised model were optimized separately for each set of embeddings under each condition using grid search. The grid for hyperparameter optimization is available in Table S1 and the optimized hyperparameters for all downstream models are in Tables S2 and S3.†

2.3.2 Embedding space visualization. t-Distributed Stochastic Neighbor Embedding (t-SNE) was used to visualize the embeddings from each protein language model. A sample of 5000 peptides from both held-out data sets were selected for

visualization. The 1280-dimensional embeddings were first reduced to 10 dimensions with PCA, and then further reduced to two dimensions with t-SNE.

2.4 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.[59] Fine-tuning boosts the model's performance on a downstream task in part by transferring broader knowledge learned during masked language modeling. The embeddings from the language model are not extracted at any point during fine-tuning. Instead, all the model's parameters are adjusted to directly classify labeled training data. A classification head was appended to the final layer of Vanilla-ESM for fine-tuning. A feed forward layer transformed the mean representation of Vanilla-ESM embeddings to another 1280-dimensional representation. Dropout regularization was applied before an additional feed forward layer projected the output to 2 dimensions and a Tanh activation function was used to compute logits. Fine-tuning was performed for one epoch with a learning rate of 2×10^{-4}. All hyperparameters for fine-tuning are available in Table S5.† 3 copies of Vanilla-ESM (each with 650 M parameters) were fine-tuned using supervised learning to predict the substrates of LazBF, LazDEF, and the entire lactazole biosynthetic pathway respectively. 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. The accuracy on each held-out data set was calculated for each of the 3 fine-tuned models. A lightweight version of Vanilla-ESM with only 12 layers and 35 million parameters was also fine-tuned on each of the 3 tasks in an identical manner.

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 vectors of length 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 33 layers produce 20 attention heads (660 attention heads in total).[52] 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.[62] The pairwise nature of the self-attention mechanism resembles epistatic interactions in protein/peptide fitness landscapes.[63] 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 attention matrices were obtained for both peptides.

3 Results and discussion

3.1 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. Vanilla-ESM embeddings also outperformed extended connectivity fingerprints (ECFPs), a common topological encoding for peptides/small molecules,[60] and embeddings from ProtBert,[61] an alternative protein language model with 420 million parameters (Fig. S1†). 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.*, low-N) had lower performance. This reflects the importance of having sufficiently large and diverse training data in supervised learning paradigms.

3.2 Masked language modeling of either data set improves LazDEF substrate classification performance

The accuracy of each supervised model type trained on LazDEF substrate/non-substrate embeddings from each of the five language models are presented in Fig. 4. LazDEF-ESM produced embeddings that significantly increased the performance of all downstream 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

Table 1 Classification accuracy with Vanilla-ESM embeddings[a]

	Training size	LR	RF	AB	SVC	MLP
LazBF	High-N	**91.2 ± 0.003**	87.3 ± 0.001	87.0 ± 0.004	89.3 ± 0.002	90.5 ± 0.003
LazDEF	High-N	**86.7 ± 0.003**	78.1 ± 0.004	77.7 ± 0.007	81.6 ± 0.003	81.2 ± 0.002
LazBF	Medium-N	**88.4 ± 0.003**	85.4 ± 0.002	86.2 ± 0.003	87.5 ± 0.003	87.6 ± 0.004
LazDEF	Medium-N	**82.3 ± 0.003**	75.1 ± 0.003	73.5 ± 0.008	78.4 ± 0.004	78.7 ± 0.007
LazBF	Low-N	**86.8 ± 0.003**	86.0 ± 0.006	84.2 ± 0.010	86.4 ± 0.006	85.5 ± 0.004
LazDEF	Low-N	73.9 ± 0.004	71.1 ± 0.008	71.7 ± 0.009	71.8 ± 0.010	**74.3 ± 0.009**

[a] Accuracy of logistic regression (LR), random forest (RF), AdaBoost (AB), support vector classifier (SVC), and multi-layer perceptron (MLP) models trained on embeddings from Vanilla-ESM on both substrate classification tasks. Values are mean ± SE across 5 repeats of 5-fold cross validation for the high-N condition (n = 1000), medium-N condition (n = 500), the low-N condition (n = 200). The best performing model in each row is highlighted.

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. Nearly 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. A similar trend was observed for LazBCDEF-ESM embeddings, but to a lesser extent. Embeddings from Peptide-ESM also improved LazDEF substrate classification models in nearly all cases, but to a smaller degree 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 (Fig. 5 and S5†). Notably, the LazBF-ESM embedding space shows a visibly 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 (Fig. 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.

3.3 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 5 language models are presented in Fig. 6. Similarly, LazBF-ESM produced embeddings that significantly improved the performance of 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 low-N condition, most models showed performance increases, with multi-layer perceptron showing the most improvement. Most supervised models trained using the medium-N and high-N conditions also showed improved performance. RF and SVC 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. Similarly, embeddings from LazBCDEF-ESM improved LazBF substrate prediction with performance increases being most pronounced in the low-N condition. Training LazDEF-ESM with higher learning rates (3×10^{-4}, 3×10^{-5}) and different batch sizes (64, 128) did not improve LazBF substrate prediction (Fig. S2 and S3†). Extending the training of LazDEF-ESM to 5 epochs with a learning rate of 3×10^{-6} did not improve LazBF substrate prediction (Fig. S4†).

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 (Fig. 5 and S5†). 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 (Fig. 5e). Predictably, the embedding space of LazBF-ESM shows the most dramatic separation of substrates from non-substrates (Fig. 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-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

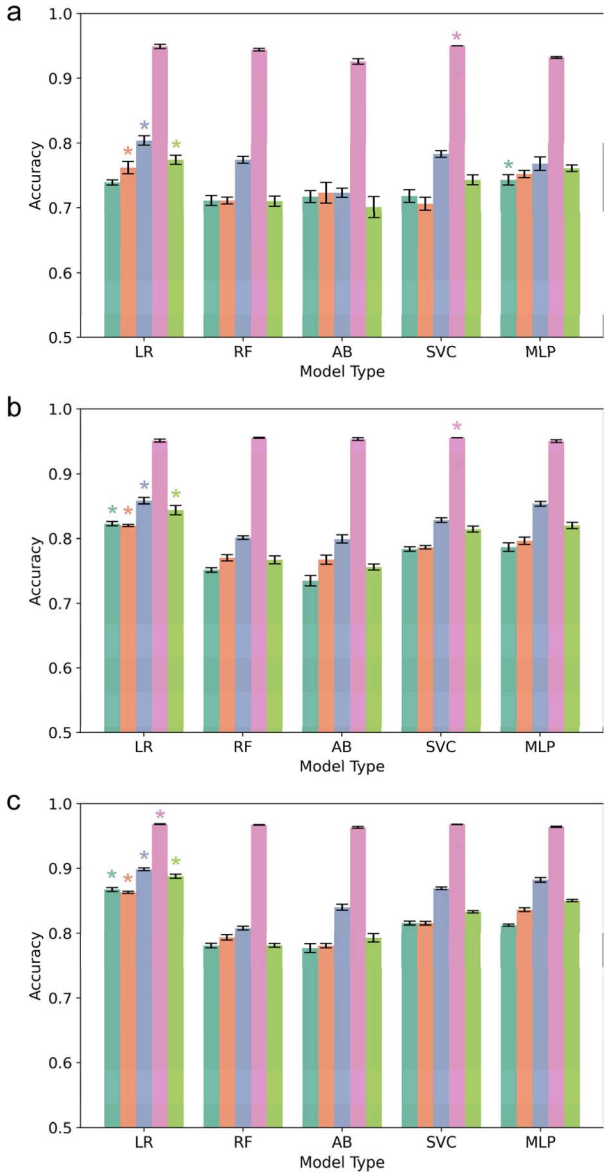


Fig. 4 Accuracy of logistic regression (LR), random forest (RF), AdaBoost (AB), support vector classifier (SVC), and multi-layer perceptron (MLP) models trained to predict LazDEF substrates. Models are trained on embeddings from a protein language model (green), a peptide language model trained on diverse peptides (orange), a peptide language model trained on LazBF substrates/non-substrates (blue), a peptide language model trained on LazDEF substrates/non-substrates (pink), and a peptide language model trained on substrates/non-substrates for the entire lactazole biosynthetic pathway (lime) in the (a) low-N condition ($n = 200$), (b) medium-N condition ($n = 500$), and (c) high-N condition ($n = 1000$). A star indicates the top performing model for each set of embeddings.

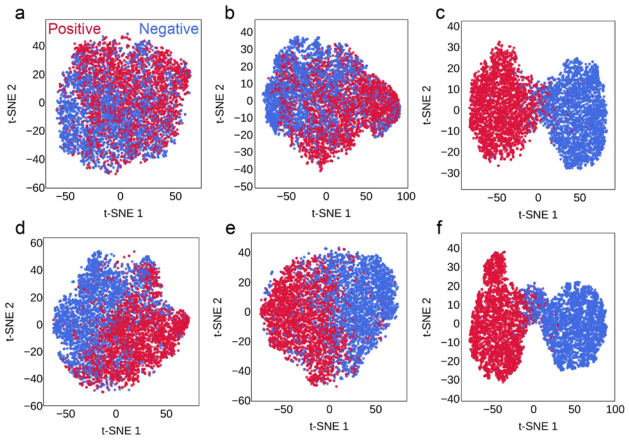


Fig. 5 t-SNE visualization of the LazDEF embedding space for (a) a protein language model, (b) a peptide language model trained on LazBF substrates/non-substrates, and (c) a peptide language model trained on LazDEF substrates/non-substrates. t-SNE visualization of the LazBF embedding space for (d) a protein language model, (e) a peptide language model trained on LazDEF substrates/non-substrates, and (f) a peptide language model trained on LazBF substrates/non-substrates. Substrates are red and non-substrates samples are blue.

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, training on the substrates/non-substrates of the entire biosynthetic pathway enabled enhanced substrate prediction for individual enzymes from the pathway. Finally, 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 suggest that this asymmetry stems from the fact that Vanilla-ESM performs better at LazBF substrate classification and additional improvement is harder to achieve. We also speculate that this observation could result 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 alternative explanation as to why transfer learning showed greater success at improving LazDEF substrate classification models.

3.4 Fine-tuned language model performance on RiPP biosynthetic enzyme classification tasks

3 copies of Vanilla-ESM were then trained to classify the substrates LazBF, LazDEF, and the entire lactazole biosynthetic

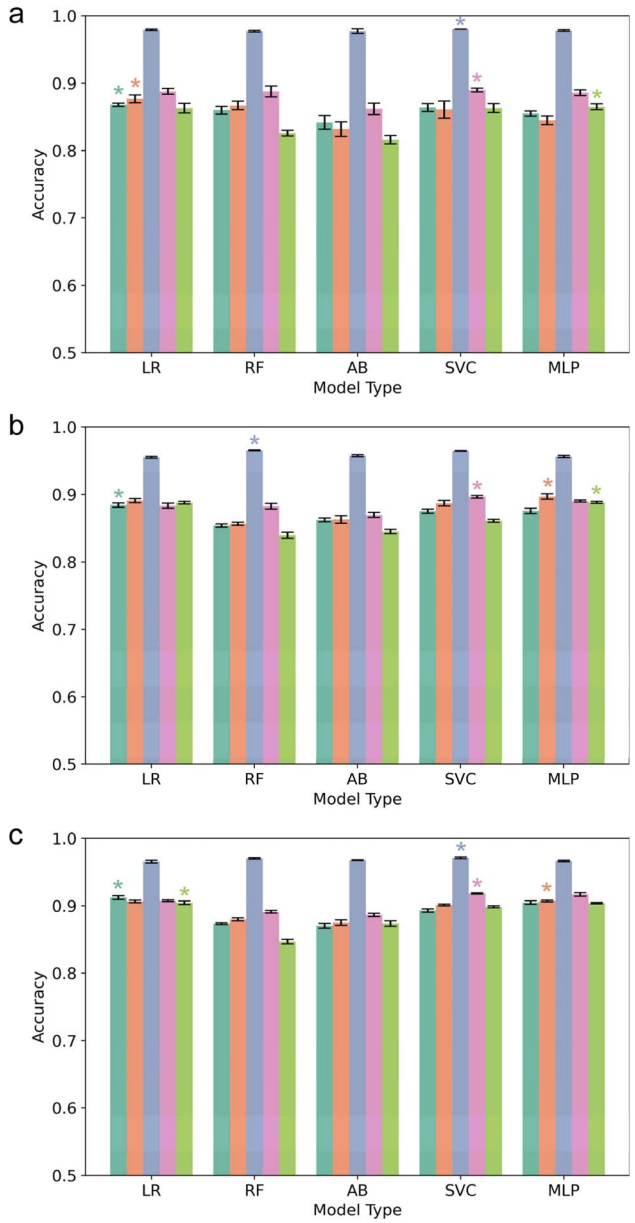


Fig. 6 Accuracy of logistic regression (LR), random forest (RF), AdaBoost (AB), support vector classifier (SVC), and multi-layer perceptron (MLP) models trained to predict LazBF substrates. Models are trained on embeddings from a protein language model (green), a peptide language model trained on diverse peptides (orange), a peptide language model trained on LazBF substrates/non-substrates (blue), a peptide language model trained on LazDEF substrates/non-substrates (pink), and a peptide language model trained on substrates/non-substrates for the entire lactazole biosynthetic pathway (lime) in the (a) low-N condition ($n = 200$), (b) medium-N condition ($n = 500$), and (c) high-N condition ($n = 1000$). A star indicates the top performing model for each set of embeddings.

pathway through a training procedure called fine-tuning. 3 copies of a smaller version of Vanilla-ESM with over 94% fewer parameters were also trained on each task. All 6 fine-tuned model showed excellent performance on their respective held-out data set (>0.95 accuracy in each case). When fine-tuned, the larger copies of Vanilla-ESM did not outperform the smaller copies despite having significantly more parameters (Table S6†). This implies that when large high-quality data sets are available, smaller models with fewer parameters can achieve satisfactory performance. Fine-tuned LazBF substrate prediction models trained with classification head dropout probabilities greater than 0.1 did not show improved performance (Table S7†). Additionally, fine-tuned models trained for additional epochs did not show significant increases in performance (Tables S9–S11†). The smaller fine-tuned models with dropout probability set to 0.1 were used for evaluation. We tested 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 (Fig. 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. Notably, downstream LazBF/DEF substrate prediction models trained on as few as 200 examples outperformed the zero-shot performance of the fine-tuned models when evaluated on the full held-out test sets (Table S8†).

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 (Fig. 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 0.78 (Fig. 7a). We ignored the contribution of serine since its importance to substrate fitness was known. The average contribution of each position from both fine-tuned models showed a 0.73 spearman coefficient (Fig. 7b). We ignored the contribution of position 6 since it was fixed. The average contribution of each residue type in each position also showed a 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 Zero-shot classification accuracy of fine-tuned models

	Supervised LazBF (%)	Supervised LazDEF (%)	Supervised LazBCDEF (%)
LazBF test set	99.4	69.7	64.4
LazDEF test set	50.9	99.2	58.9
LazBCDEF test set	52.3	64.2	95.8

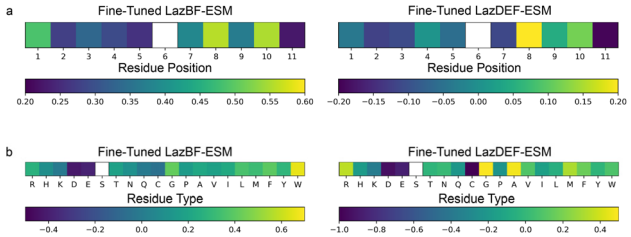


Fig. 7 A LazBF substrate prediction model and a LazDEF substrate prediction model produce correlated integrated gradients for LazBF substrates/non-substrates. (a) The average contribution of each position to substrate fitness shows a 0.73 spearman coefficient between the two models. Position 6 is ignored due to containing a fixed serine residue. (b) The average contribution of each amino acid to substrate fitness shows a 0.78 spearman coefficient between the two models. Serine is ignored because its importance for substrate fitness is established.

3.5 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 for substrate fitness. 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 (Fig. 8a and S6†). 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.[62]

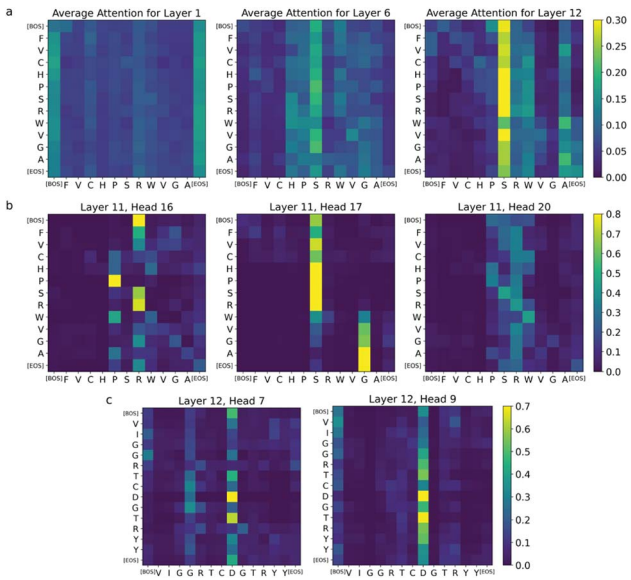


Fig. 8 Attention maps from a language model trained to predict LazBF substrates. [BOS] and [EOS] tokens mark the "beginning of sequence" and "end of sequence" respectively. (a) Middle and later layers focus on specific residues and motifs. (b) Attention heads from the penultimate layer highlight a motif 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.[62,63] 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] Fig. 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. Fig. S7a† provides additional examples of this motif represented in the attention matrices of a larger language model fine-tuned to predict LazBF substrates.

Surprisingly, we observe that the fine-tuned version of LazBF-ESM also highlights some epistatic features of the LazDEF substrate VIGGRTCDGTRYY (Fig. 8c). Precalculated epi-scores for this peptide indicate that Asp8 has numerous positive and negative epistatic interactions with surrounding residues 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. Fig. S7b† provides additional examples of this residue represented in the attention matrices of a larger language model fine-tuned to predict LazBF substrates.

4 Conclusion

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 substrates/non-substrates for the entire lactazole

biosynthetic pathway, and a final 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 data sets, 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 enhanced the ability to classify RiPP biosynthetic enzyme substrates in limited data regimes. Crucially, our results indicated that a single high quality data set containing substrates and non-substrates for a RiPP biosynthetic enzyme can be leveraged to improve substrate prediction for other enzymes from the same biosynthetic pathway, including when little data is available. This holds potential to strengthen the understanding of RiPP biosynthesis by increasing accuracy in the absence of data. 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.

Data availability

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

Author contributions

D. S. accquired funding for the project. J. D. C., X. M., and D. S. designed the research. J. D. C. performed all model training and data analysis. X. M. and D. A. M. participated in discussion of results. D. S. supervised the research. All authors reviewed the manuscript.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

The authors thank Dr Alexander A. Vinogradov, Dr Yuki Goto, and Dr Hiroaki Suga for sharing the data sets used in this study. J. D. C. thanks Song Yin and Diego E. Kleiman from the Shukla group for their comments. D. S. acknowledges support from NIH grants R35GM142745 and R21AI167693.

Notes and references

1 C. Ongpipattanakul, E. K. Desormeaux, A. DiCaprio, W. A. van der Donk, D. A. Mitchell and S. K. Nair, *Chem. Rev.*, 2022, **122**, 14722–14814.
2 Y. Fu, A. H. Jaarsma and O. P. Kuipers, *Cell. Mol. Life Sci.*, 2021, **78**, 3921–3940.
3 M. Montalbán-López, T. A. Scott, S. Ramesh, I. R. Rahman, A. J. van Heel, J. H. Viel, V. Bandarian, E. Dittmann, O. Genilloud, Y. Goto, M. J. G. Burgos, C. Hill, S. Kim, J. Koehnke, J. A. Latham, A. J. Link, B. Martínez, S. K. Nair, Y. Nicolet, S. Rebuffat, H.-G. Sahl, D. Sareen, E. W. Schmidt, L. Schmitt, K. Severinov, R. D. Süssmuth, A. W. Truman, H. Wang, J.-K. Weng, G. P. van Wezel, Q. Zhang, J. Zhong, J. Piel, D. A. Mitchell, O. P. Kuipers and W. A. van der Donk, *Nat. Prod. Rep.*, 2021, **38**, 130–239.
4 P. G. Arnison, M. J. Bibb, G. Bierbaum, A. A. Bowers, T. S. Bugni, G. Bulaj, J. A. Camarero, D. J. Campopiano, G. L. Challis, J. Clardy, P. D. Cotter, D. J. Craik, M. Dawson, E. Dittmann, S. Donadio, P. C. Dorrestein, K.-D. Entian, M. A. Fischbach, J. S. Garavelli, U. Göransson, C. W. Gruber, D. H. Haft, T. K. Hemscheidt, C. Hertweck, C. Hill, A. R. Horswill, M. Jaspars, W. L. Kelly, J. P. Klinman, O. P. Kuipers, A. J. Link, W. Liu, M. A. Marahiel, D. A. Mitchell, G. N. Moll, B. S. Moore, R. Müller, S. K. Nair, I. F. Nes, G. E. Norris, B. M. Olivera, H. Onaka, M. L. Patchett, J. Piel, M. J. T. Reaney, S. Rebuffat, R. P. Ross, H.-G. Sahl, E. W. Schmidt, M. E. Selsted, K. Severinov, B. Shen, K. Sivonen, L. Smith, T. Stein, R. D. Süssmuth, J. R. Tagg, G.-L. Tang, A. W. Truman, J. C. Vederas, C. T. Walsh, J. D. Walton, S. C. Wenzel, J. M. Willey and W. A. van der Donk, *Nat. Prod. Rep.*, 2013, **30**, 108–160.
5 A. A. Vinogradov, J. S. Chang, H. Onaka, Y. Goto and H. Suga, *ACS Cent. Sci.*, 2022, **8**, 814–824.
6 S. L. Ivry, N. O. Meyer, M. B. Winter, M. F. Bohn, G. M. Knudsen, A. J. O'Donoghue and C. S. Craik, *Protein Sci.*, 2017, **27**, 584–594.
7 W. Tang, G. Jiménez-Osés, K. N. Houk and W. A. van der Donk, *Nat. Chem.*, 2014, **7**, 57–64.
8 T. Le, K. J. D. Fouque, M. Santos-Fernandez, C. D. Navo, G. Jiménez-Osés, R. Sarksian, F. A. Fernandez-Lima and W. A. van der Donk, *J. Am. Chem. Soc.*, 2021, **143**, 18733–18743.
9 I. Song, Y. Kim, J. Yu, S. Y. Go, H. G. Lee, W. J. Song and S. Kim, *Nat. Chem. Biol.*, 2021, **17**, 1123–1131.
10 S. P. Mahajan, Y. Srinivasan, J. W. Labonte, M. P. DeLisa and J. J. Gray, *ACS Catal.*, 2021, **11**, 2977–2991.

11 L. Meng, W.-S. Chan, L. Huang, L. Liu, X. Chen, W. Zhang, F. Wang, K. Cheng, H. Sun and K.-C. Wong, *Comput. Struct. Biotechnol. J.*, 2022, **20**, 3522–3532.
12 Y. Yan, J.-Y. Jiang, M. Fu, D. Wang, A. R. Pelletier, D. Sigdel, D. C. Ng, W. Wang and P. Ping, *Cells Rep. Methods*, 2023, **3**, 100430.
13 K. Smith, N. Rhoads and S. Chandrasekaran, *STAR Protoc.*, 2022, **3**, 101799.
14 Y. Liu, Y. Liu, G.-A. Wang, Y. Cheng, S. Bi and X. Zhu, *Front. Bioinform.*, 2022, **2**, 834153.
15 D. Wang, D. Liu, J. Yuchi, F. He, Y. Jiang, S. Cai, J. Li and D. Xu, *Nucleic Acids Res.*, 2020, **48**, W140–W146.
16 B. J. Burkhart, G. A. Hudson, K. L. Dunbar and D. A. Mitchell, *Nat. Chem. Biol.*, 2015, **11**, 564–570.
17 B. J. Burkhart, C. J. Schwalen, G. Mann, J. H. Naismith and D. A. Mitchell, *Chem. Rev.*, 2017, **117**, 5389–5456.
18 Y. Zhao and O. N. Jensen, *Proteomics*, 2009, **9**, 4632–4641.
19 I. A. Kaltashov, C. E. Bobst, R. R. Abzalimov, G. Wang, B. Baykal and S. Wang, *Biotechnol. Adv.*, 2012, **30**, 210–222.
20 N. Brandes, D. Ofer, Y. Peleg, N. Rappoport and M. Linial, *Bioinformatics*, 2022, **38**, 2102–2110.
21 Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido and A. Rives, *Science*, 2023, **379**, 1123–1130.
22 R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel and Y. S. Song, *arXiv*, 2019, preprint, arXiv:1906.08230, DOI: **10.48550/arXiv.1906.08230**, **https://arxiv.org/abs/1906.08230**.
23 F. A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma and R. Fergus, *Proc. Natl. Acad. Sci. U. S. A.*, 2021, **118**, 2016239118.
24 A. Elnaggar, M. Heinzinger, C. Dallago, G. Rihawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik and B. Rost, *arXiv*, 2020, preprint, arXiv:2007.06225, DOI: **10.48550/arXiv.2007.06225**, **https://arxiv.org/abs/2007.06225**.
25 F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong and Q. He, *arXiv*, 2019, preprint, arXiv:1911.02685, DOI: **10.48550/arXiv.1911.02685**, **https://arxiv.org/abs/1911.02685**.
26 Z. Shamsi, M. Chan and D. Shukla, *J. Phys. Chem. B*, 2020, **124**, 3845–3854.
27 D. Wang, J. Jin, Z. Li, Y. Wang, M. Fan, S. Liang, R. Su and L. Wei, *bioRxiv*, 2023, preprint, DOI: **10.1101/2023.05.22.541389**, **https://www.biorxiv.org/content/10.1101/2023.05.22.541389v2**.
28 L. Wang, C. Huang, M. Wang, Z. Xue and Y. Wang, *Briefings Bioinf.*, 2023, **24**, bbad077.
29 Y. Zhang, J. Lin, L. Zhao, X. Zeng and X. Liu, *Briefings Bioinf.*, 2021, **22**, bbab200.
30 Z. Ma, Y. Zou, X. Huang, W. Yan, H. Xu, J. Yang, Y. Zhang and J. Huang, *arXiv*, 2023, preprint, arXiv:2309.14404, DOI: **10.48550/arXiv.2309.14404**, **https://arxiv.org/abs/2309.14404**.
31 Z. Du, X. Ding, W. Hsu, A. Munir, Y. Xu and Y. Li, *Food Chem.*, 2024, **431**, 137162.
32 K. Fosgerau and T. Hoffmann, *Drug Discovery Today*, 2015, **20**, 122–128.
33 M. Muttenthaler, G. F. King, D. J. Adams and P. F. Alewood, *Nat. Rev. Drug Discovery*, 2021, **20**, 309–325.
34 G. Sadeh, Z. Wang, J. Grewal, H. Rangwala and L. Price, *arXiv*, 2022, preprint, arXiv:2211.06428, DOI: **10.48550/arXiv.2211.06428**, **https://arxiv.org/abs/2211.06428**.
35 H. Huang, C. N. Arighi, K. E. Ross, J. Ren, G. Li, S.-C. Chen, Q. Wang, J. Cowart, K. Vijay-Shanker and C. H. Wu, *Nucleic Acids Res.*, 2017, **46**, D542–D550.
36 C. Lu, J. H. Lubin, V. V. Sarma, S. Z. Stentz, G. Wang, S. Wang and S. D. Khare, *Proc. Natl. Acad. Sci. U. S. A.*, 2023, **120**, 2303590120.
37 D. C. K. Chan and L. L. Burrows, *J. Antibiot.*, 2020, **74**, 161–175.
38 C. J. Schwalen, G. A. Hudson, B. Kille and D. A. Mitchell, *J. Am. Chem. Soc.*, 2018, **140**, 9494–9501.
39 S. Hayashi, T. Ozaki, S. Asamizu, H. Ikeda, S. Ōmura, N. Oku, Y. Igarashi, H. Tomoda and H. Onaka, *Chem. Biol.*, 2014, **21**, 679–688.
40 A. A. Vinogradov and H. Suga, *Cell Chem. Biol.*, 2020, **27**, 1032–1051.
41 G. A. Hudson, A. R. Hooper, A. J. DiCaprio, D. Sarlah and D. A. Mitchell, *Org. Lett.*, 2020, **23**, 253–256.
42 A. A. Vinogradov, E. Nagai, J. S. Chang, K. Narumi, H. Onaka, Y. Goto and H. Suga, *J. Am. Chem. Soc.*, 2020, **142**, 20329–20334.
43 A. A. Vinogradov, M. Nagano, Y. Goto and H. Suga, *J. Am. Chem. Soc.*, 2021, **143**, 13358–13369.
44 A. A. Vinogradov, M. Shimomura, N. Kano, Y. Goto, H. Onaka and H. Suga, *J. Am. Chem. Soc.*, 2020, **142**, 13886–13897.
45 A. A. Vinogradov, M. Shimomura, Y. Goto, T. Ozaki, S. Asamizu, Y. Sugai, H. Suga and H. Onaka, *Nat. Commun.*, 2020, **11**, 2272.
46 M. Sundararajan, A. Taly and Q. Yan, *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3319–3328.
47 B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey and C. H. Wu, *Bioinformatics*, 2014, **31**, 926–932.
48 F. Desiere, *Nucleic Acids Res.*, 2006, **34**, D655–D658.
49 J. S. Chang, A. A. Vinogradov, Y. Zhang, Y. Goto and H. Suga, *ACS Cent. Sci.*, 2023, **9**, 2150–2160.
50 J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *arXiv*, 2018, preprint, arXiv:1810.04805, DOI: **10.48550/arXiv.1810.04805**, **https://arxiv.org/abs/1810.04805**.
51 H. Wang, J. Li, H. Wu, E. Hovy and Y. Sun, *Engineering*, 2023, **25**, 51–65.
52 Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido and A. Rives, *bioRxiv*, 2022, preprint, DOI: **10.1101/2022.07.20.500902**.
53 J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu and A. Rives, *bioRxiv*, 2021, preprint, DOI: **10.1101/2021.07.09.450648**.
54 R. Rao, J. Meier, T. Sercu, S. Ovchinnikov and A. Rives, *bioRxiv*, 2020, preprint, DOI: **10.1101/2020.12.15.422761**.
55 T. Bepler and B. Berger, *Cell Systems*, 2021, **12**, 654.e3–669.e3.

56 C. Marquet, M. Heinzinger, T. Olenyi, C. Dallago, K. Erckert, M. Bernhofer, D. Nechaev and B. Rost, *Hum. Genet.*, 2021, **141**, 1629–1647.
57 K. Weissenow, M. Heinzinger and B. Rost, *Structure*, 2022, **30**, 1169.e4–1177.e4.
58 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
59 J. Howard and S. Ruder, *arXiv*, 2018, preprint, arXiv:1801.06146, DOI: **10.48550/arXiv.1801.06146**, **https://arxiv.org/abs/1801.06146**.
60 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
61 A. Elnaggar, M. Heinzinger, C. Dallago, G. Rehawi, Y. Wang, L. Jones, T. Gibbs, T. Feher, C. Angerer, M. Steinegger, D. Bhowmik and B. Rost, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, **44**, 7112–7127.
62 J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, *arXiv*, 2020, preprint, arXiv:2006.15222, DOI: **10.48550/arXiv.2006.15222**, **https://arxiv.org/abs/2006.15222**.
63 T. N. Starr and J. W. Thornton, *Protein Sci.*, 2016, **25**, 1204–1218.