ProteomeLM: A proteome-scale language model allowing fast prediction of protein-protein interactions and gene essentiality across taxa

Cyril Malbranke[1,2,*], Gionata Paolo Zalaffi[1,2], Anne-Florence Bitbol[1,2,*]

1 Institute of Bioengineering, School of Life Sciences, EPFL, CH-1015 Lausanne, Switzerland
2 SIB Swiss Institute of Bioinformatics, CH-1015 Lausanne, Switzerland
* Corresponding authors: `cyril.malbranke@epfl.ch`, `anne-florence.bitbol@epfl.ch`

August 1, 2025

Abstract

Language models starting from biological sequence data are advancing many inference problems, both at the scale of single proteins, and at the scale of genomic neighborhoods. In this paper, we introduce ProteomeLM, a transformer-based language model that reasons on entire proteomes from species spanning the tree of life. Leveraging protein language model embeddings, ProteomeLM is trained to reconstruct masked protein embeddings using the whole proteomic context. It thus learns contextualized protein representations reflecting proteome-scale functional constraints. We show that ProteomeLM spontaneously captures protein-protein interactions (PPI) in its attention coefficients. We demonstrate that it screens whole interactomes orders of magnitude faster than amino-acid coevolution-based methods, and substantially outperforms them. We further develop ProteomeLM-PPI, a supervised PPI prediction network that combines ProteomeLM embeddings and attention coefficients, and achieves state-of-the-art performance across species and benchmarks. Finally, we introduce ProteomeLM-Ess, a supervised predictor of gene essentiality that generalizes across diverse taxa. Our results highlight the power of proteome-scale language models for addressing function and interactions at the organism level.

Introduction

Recently, deep learning approaches have brought important progress to inference from biological sequence data. Protein language models are deep learning models based on natural language processing methods. Trained on large ensembles of protein sequences, they learn sequence representations that encode structural and functional signals [1–11], and have advanced the prediction of protein structure [8, 9], subcellular localization [12], and mutational effects [13]. Similarly, genome language models [14–20] have given insight in non-coding DNA, gene expression and taxonomic classification [15–17], capturing operons and enzymatic function [19], and predicting mutation effects [18]. However, so far, these models span at most hundreds of kilobases or a few megabases [16, 17, 20]. As these models do not capture dependencies across entire genomes, especially in eukaryotes, they cannot predict emergent properties such as protein-protein interactions (PPI).

PPI are fundamental to most biological processes, including signal transduction, cellular metabolism, and immune responses. Knowing these interactions is critical for deciphering cellular processes and for developing therapeutic interventions. However, large-scale PPI determination – up to the full PPI networks of non-model species – remains a significant challenge [21]. Indeed, precise experimental methods are both labor-intensive and costly, particularly when scaled to entire proteomes, and high-throughput ones have limited accuracy [22]. While curated PPI databases now contain large datasets [23–25], they remain incomplete and biased toward well-studied species and interaction types. Computational PPI predictions

have been developed to overcome this gap. Structure-based ones, including docking [26–29] and multimeric folding algorithms like AlphaFold-Multimer [30], have achieved remarkable accuracy for specific interactions [31], but remain computationally intensive. Sequence-based methods, which rely on evolutionary signals, offer better scalability. Amino acids that are in contact at the interface of protein complexes need to maintain physico-chemical complementarity through evolution, yielding correlations in amino-acid usage at contacting sites, known as coevolution [32–35]. This signal can be detected by Potts models, also known as direct coupling analysis (DCA), which are trained on multiple sequence alignments of homologous proteins for each candidate PPI [36–42]. At a larger scale, during evolution, genes coding for proteins that interact tend to undergo similar evolutionary pressures and to be either present or absent together in each genome, thus having correlated co-occurence patterns across genomes. This coevolution between genes is exploited by phylogenetic profiling [43–51] and co-occurrence methods [52]. While such coevolution methods are often effective in bacteria and other well-represented clades, they struggle in eukaryotes or poorly sampled taxa, and require careful curation of orthologs, and for DCA, paired multiple sequence alignments [39, 40, 53, 54].

Protein language models trained with the masked language modeling (MLM) objective of predicting masked amino acids in sequences using the surrounding context [3, 8–11] learn coevolution between amino acids. This allows them, in particular those based on multiple sequence alignments [8] to capture protein structure, an ability exploited in AlphaFold2 via its EvoFormer module [55]. Given the success of protein language models at capturing coevolution between amino acids, it is tantalizing to develop such models at the proteome scale, i.e. taking as input the ensemble of proteins encoded by a genome. Indeed, we posit that such models should capture coevolution between proteins, thereby generalizing over phylogenetic profiling methods [43–51]. This should make them highly suited to provide predictions of complete protein-protein interaction networks, and of their evolution. Importantly, once trained, the computational cost of prediction on new species should be small. Moreover, such foundation models can also be used for other downstream applications where proteome context information is important, such as predicting gene essentiality [56].

In this paper, we introduce ProteomeLM, a novel transformer-based language model that uniquely reasons on entire proteomes from multiple species spanning the tree of life. ProteomeLM leverages embeddings from the protein language model ESM-Cambrian [11], and is thus informed of functional protein-level properties, which it integrates at the proteome scale. We show that ProteomeLM's attention coefficients learn PPI in an unsupervised way. Building on this emergent ability of ProteomeLM, we propose a new method to screen whole interactomes orders of magnitude faster than DCA pipelines, and we demonstrate that it substantially outperforms them. Next, we introduce ProteomeLM-PPI, a supervised PPI prediction network that combines ProteomeLM embeddings and attention coefficients, and we show that it achieves state-of-the-art supervised PPI prediction methods across species and benchmarks. Finally, as another example of downstream tasks allowed by ProteomeLM embeddings, we introduce ProteomeLM-Ess, a supervised predictor of gene essentiality that generalizes across diverse taxa. Our results show that representing proteins in their full proteome context allows ProteomeLM to capture emergent biological signals that remain inaccessible to models that reason on proteins or on local genomic contexts.

Results

A proteome-scale language model leveraging protein language model representations

We introduce ProteomeLM, a transformer-based Proteome Language Model (LM). We trained ProteomeLM on a large corpus of proteomes, spanning the tree of life, from bacteria and archaea to eukaryotes and viruses, to learn protein representations in the context of complete proteomes. ProteomeLM takes as input a proteome, i.e. the set of proteins encoded by a given genome, and aims to capture the functional and evolutionary signals that emerge between proteins at the proteome level.

Specifically, we start from each protein's amino-acid sequence, and represent it by an embedding generated by the protein language model ESM-Cambrian (ESM-C) [11], see Figure 1 and "Protein Repre-

sentations" in Methods. This allows our model to leverage the rich functional sequence-derived properties [57] learned for each protein by protein language models [3, 8, 9, 13] (see also [19]). During training, a subset of these protein embeddings is masked, and the model is tasked with reconstructing them using the remaining unmasked protein embeddings from the same proteome, see Figure 1. This masked language modeling prediction task allows ProteomeLM to learn the dependencies between proteins encoded by a given genome.

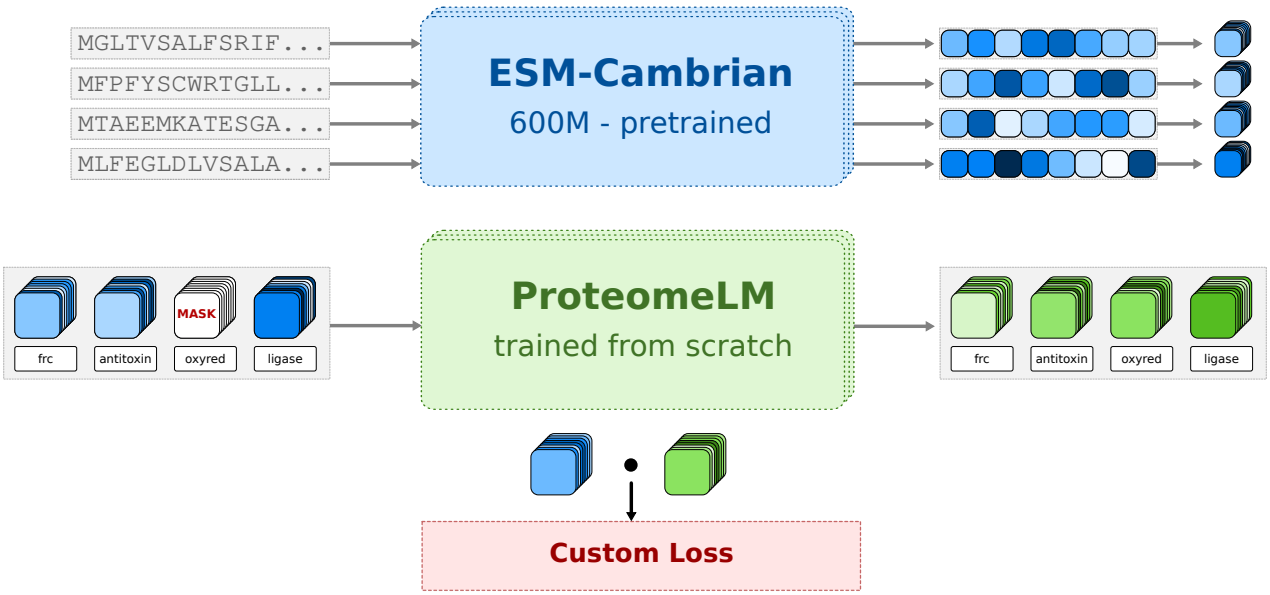

Figure 1: **ProteomeLM training.** Input amino-acid sequences, corresponding to proteins encoded by a genome, are embedded through the pretrained ESM-C model (see Methods), yielding a fixed-dimensional embedding for each protein. The embeddings serve as input to ProteomeLM, trained from scratch to predict the masked embeddings of proteins in the context of their proteome (see Methods). Proteins are annotated by their orthologous group, providing a functional encoding. ProteomeLM's training uses a custom polar loss (see Methods): for each masked protein, it essentially aims to minimize the difference between the ESM-C embedding and the ProteomeLM embedding, but in a protein family-specific manner.

A challenge when constructing a model that can reason across diverse organisms is the fact that their genome organization is very different. For instance, while bacterial genes are organized in operons, functionally related proteins are often not encoded in proximity in eukaryotic genomes, and gene order is less conserved in eukaryotes than in prokaryotes. Therefore, ProteomeLM does not employ positional encoding along the genome, which sets it apart from existing genome language models (and protein language models). Instead, we propose a *functional encoding*, based on orthology, which captures shared evolutionary and functional relationships across genes in different genomes (see "Functional Encoding" in Methods). Orthologous groups comprise genes in different species that descend from a common ancestral gene via speciation, and generally have retained the same function. Phylogenetic profiling methods have shown that the presence and absence data of orthologous groups, across genomes, contain information about functional relationships [43–45, 48, 58]. For instance, genes coding for interacting proteins tend to have correlated presence-absence vectors across genomes, allowing to predict interactions. Hence, we posit that functional encoding will help ProteomeLM to learn coevolution between proteins in a proteome, and thus, their functional relationships. In practice, orthologous groups were collected from OrthoDB [59]. Note that they were built using statistical homologs matching and no human annotations.

ProteomeLM was trained on nearly 32,000 annotated proteomes, spanning all domains of life (see "Dataset" in Methods). By learning to reconstruct masked proteins from their genomic context, ProteomeLM is encouraged to learn the relationships that define protein function and interactions. We trained versions

of ProteomeLM with multiple sizes, ranging from 6M to 328M parameters. We observed stable learning dynamics in all cases (see "Architecture and Training" in Methods).

ProteomeLM attention coefficients capture protein-protein interactions

ProteomeLM was trained on full proteomes, and informed by protein-level properties thanks to the use of embeddings from protein language models. Does it spontaneously learn protein-protein interactions (PPI) by being trained to predict a protein's embedding using the context of its proteome? To address this question, we examine the attention coefficients of the model [5, 8, 60]. Indeed, in a transformer model, attention coefficients encode the importance of each part of the context (here, of each protein) when performing masked language modeling (here, when aiming to predict another masked protein). Thus, we hypothesize that these coefficients should capture functional dependencies between proteins, such as interactions. For each pair of protein in five species, we compare these attention coefficients to known protein-protein interactions. The species considered are *Escherichia coli*, *Saccharomyces cerevisiae*, *Caenorhabditis elegans*, *Drosophila melanogaster*, *Mus musculus*, and *Homo sapiens*. Specifically, we use the D-SCRIPT dataset [21], which is derived from the STRING database [61], and focuses exclusively on experimentally validated physical interactions. Complete proteomes from the five species considered were processed through ProteomeLM, and we extracted attention coefficients for each positive and negative interaction pair across both datasets.

Figure 2A-C shows that many attention heads of ProteomeLM are predictive of interaction labels. Moreover, we observe that several attention heads possess significant predictive power across all species considered (3 extra species are shown in Figure S1). In particular, head 7 of layer 3 achieves an AUC of 0.92 in *E. coli*, while also performing strongly in other species. The larger ProteomeLM-M also exhibits strong unsupervised PPI recovery, see Figure S2. Thus, ProteomeLM can identify interacting proteins among thousands in a complete proteome (e.g., ∼4,000 proteins in *E. coli* and ∼20,000 in humans) in an unsupervised manner, without any fine-tuning. This is especially compelling given that ProteomeLM does not rely on gene order or local genomic context. The learning of PPI spontaneously emerges from the masked prediction training, which promotes the learning of dependencies between proteins in a proteome. Our result is reminiscent of the finding that protein language models' attention coefficients carry information about residue-residue interactions involved in the three-dimensional structure of proteins, while being trained only on sequence data with the objective of filling in masked amino acids [3, 8, 9].

Finally, in Figure 2D, we show that PPI are most accurately captured by attention heads in the central layers of the model. A similar trend is observed across other model sizes (XS, M, and L), see Figure S3. In protein language models, central layers are known to capture more complex interactions than early layers [5]. Thus, our finding supports the notion that higher-order interactions, rather than simple local features, are essential for understanding PPI, and that ProteomeLM can extract such complex biological signals.

ProteomeLM provides fast and accurate PPI screening

ProteomeLM successfully captures PPI, and was trained on organisms spanning the tree of life. Can it thus contribute to shedding light on complete interactomes in various species? Current large-scale interactome prediction workflows generally rely on a two-stage pipeline [42, 62, 63]. First, relatively light sequence-based methods exploiting coevolution in multiple sequence alignments (MSAs), in particular Potts models, also known as Direct Coupling Analysis [36, 38] (DCA), are used to identify promising protein pairs that may interact. Second, heavier structure-based methods like AlphaFold-Multimer [30] or RoseTTAFold-PPI [63] are used to further analyze these candidate pairs, through computationally intensive structural modeling. While effective, the first step of this approach is limited by the computational cost of MSA generation, and by the sheer number of pairwise models required to scan large proteomes. Indeed, DCA is a family-specific model, so one model has to be trained per candidate pair. To this day, such large-scale PPI screens have thus only been performed on a few model organisms, namely *E. coli* [41] and *S. cerevisiae* [42], and very recently on *H. sapiens* [63], and on 19 human pathogens [62].

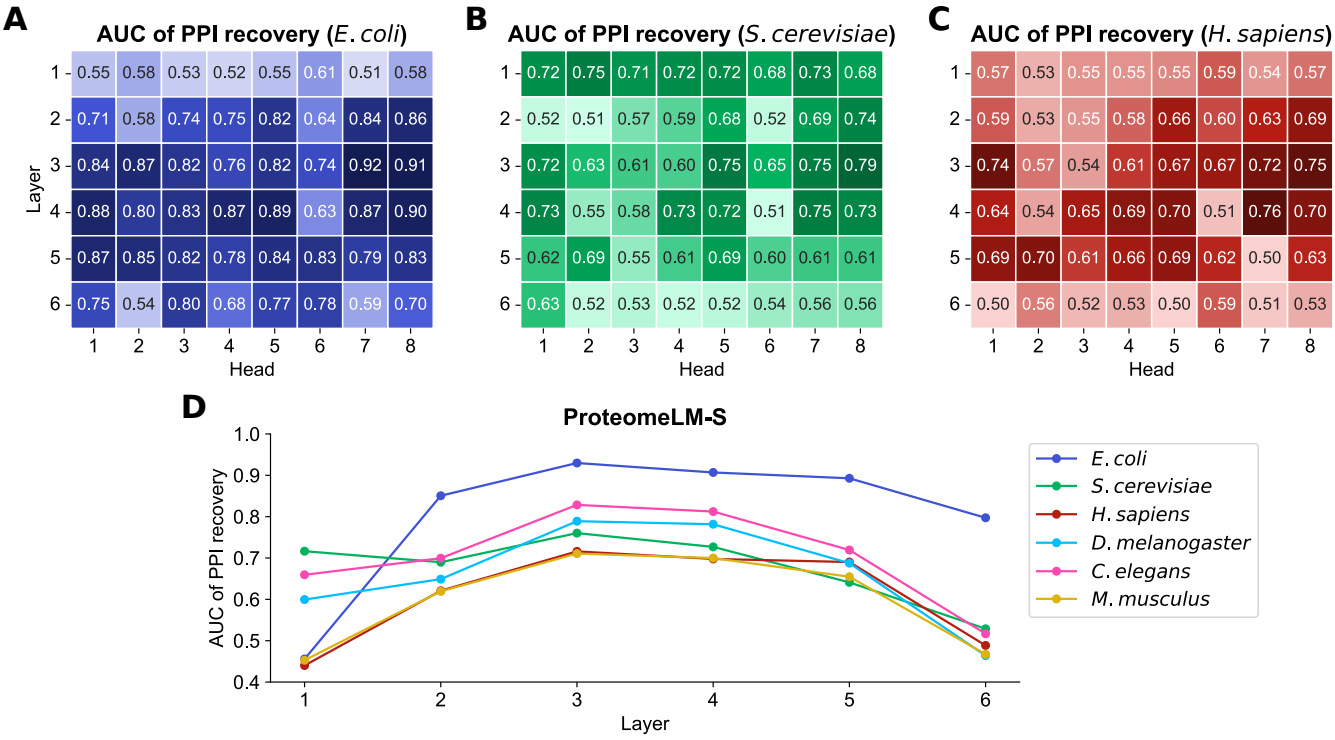

Figure 2: **Unsupervised detection of PPI using ProteomeLM attention coefficients.** We assess the ability of the attention coefficients of ProteomeLM-S (36M parameters) at predicting PPI from the D-SCRIPT dataset [21], using the Area Under the Receiver Operating Characteristic Curve (AUC) as a metric. **(A-C)** In three species, we report the AUC of each attention head in each layer of ProteomeLM, measuring its ability to distinguish interacting from non-interacting protein pairs (1: perfect classifier; 0.5: random). **(D)**: We show the AUC obtained for the sum of attention coefficients over all heads in each layer of ProteomeLM-S, in each of the five species considered.

To assess ProteomeLM's promise as a first filter for interactome prediction, we trained a lightly-supervised and lightweight classifier on attention coefficients from ProteomeLM and evaluated its performance on the full human interactome and across pathogen interactomes. We compare its computational requirements and its performance to those reported in a recent large-scale study that applied DCA respectively to over 190 million *Homo sapiens* protein pairs [63]. We also test our method on 19 human bacterial pathogens considered in [62], collectively comprising 102 million protein pairs. Specifically, our classifier is a logistic regression aiming to predict PPI from a combination of the 48 attention heads of ProteomeLM-S (see Figure 2A-C), and is trained over a small set of 100 interacting pairs and 1,000 random pairs (treated as non-interacting) held-out from evaluation later. Positive pairs are sampled among the pairs that have a confidence score above 0.99 (extremely high) in the STRING database [61], either among human pairs or among pathogenic pairs in the corresponding datasets.

In Ref. [63], the DCA pipeline required over 30 days on 50–100 GPUs to process the human proteome. In contrast, ProteomeLM inference, including ESM-C embeddings and attention coefficient computation, takes under 10 minutes per proteome (e.g., *H. sapiens*) on a single RTX A6000 GPU. Importantly, these features are calculated for all possible protein pairs, without the need to train a separate model for each candidate pair. Moreover, ProteomeLM training was completed in 3 days on a single H100 GPU, and generalizes to all downstream tasks without retraining. As shown in Figure 3A, ProteomeLM thus reduces overall compute time by up to six orders of magnitude for inference alone, and by three orders of magnitude when training is included (see Supplementary for full compute analysis).

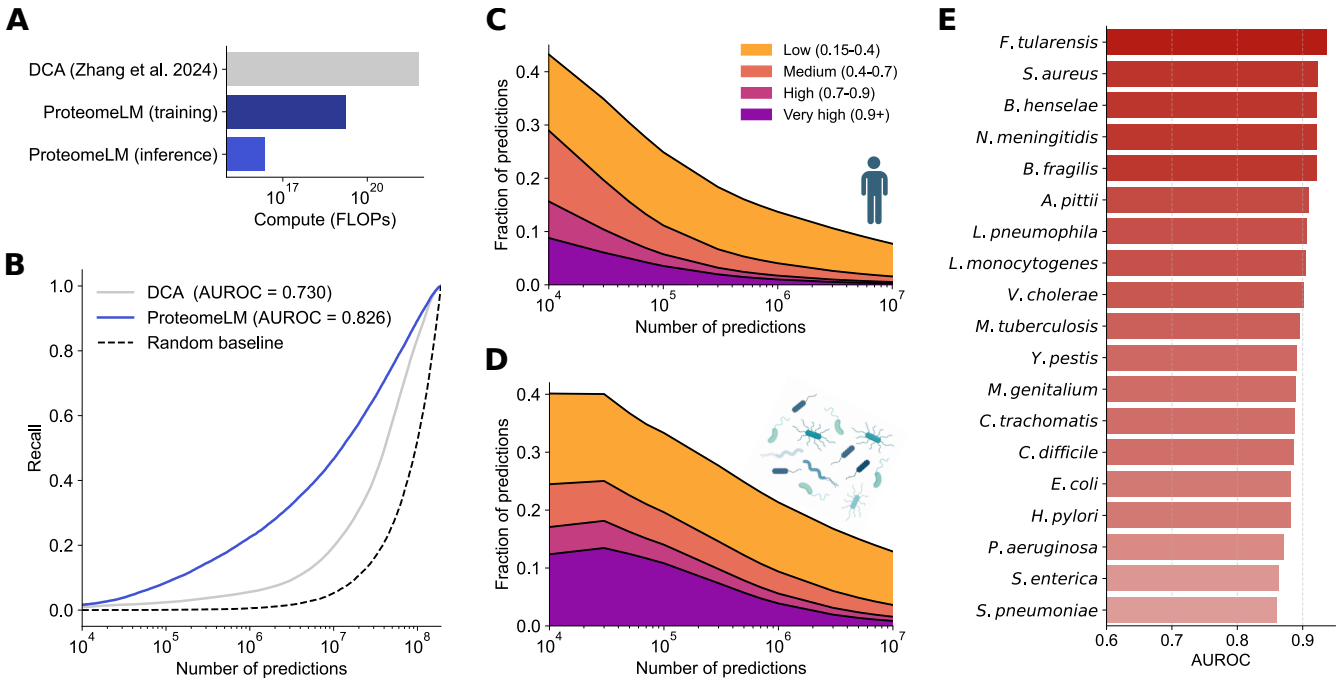

Figure 3: **Fast and high-precision screening of whole interactomes using ProteomeLM. (A)** Compute time required to analyze the full human proteome using ProteomeLM versus DCA. We compare the time to train 190 million DCA models [63] with the time to train ProteomeLM (which is done once across a large number of organisms) and with the time to apply ProteomeLM to the *H. sapiens* proteome (inference). ProteomeLM offers a speed-up of up to 6 orders of magnitude when focusing on inference, and of 3 orders of magnitude when including training (see Supplementary Information Section 1 for details). **(B)** *H. sapiens* interactome recovery by ProteomeLM and by DCA. Recall is shown versus the number of predictions made. Predictions are made by ranking all possible protein pairs by their score, either from Proteome or from DCA [63] **(C–D)** Fraction of top-scoring predictions that correspond to known interactions in the STRING database, for *H. sapiens* (C) and 19 human bacterial pathogens (D). **(E)** Interactome recovery performance across 19 pathogenic bacterial species, measured by AUC.

Moreover, Figure 3B shows that ProteomeLM significantly outperforms DCA in recovering experimentally validated interactions. In *H. sapiens*, ProteomeLM achieves an AUC of 0.83, compared to 0.73 for DCA. Among the top 10 million scored pairs, it recovers 50% of known PPI, versus only 20% for DCA. At higher precision thresholds (e.g., top 1 million), ProteomeLM also delivers improved recall. Thus, ProteomeLM has a very strong potential to reduce the burden on downstream structure-based modeling.

We further examined the overlap between ProteomeLM's top predictions and STRING [61] annotations. Figure 3C shows that in *H. sapiens*, over 40% of the top 10,000 predictions align with known or suspected interactions, and nearly 10% correspond to high-confidence interactions according to STRING. This suggests that ProteomeLM captures meaningful biological associations, even beyond the most confidently labeled pairs.

We extended our analysis to 19 human bacterial pathogens [62]. Figure 3D shows that more than 40% of the top 10,000 predictions are supported by STRING. The per-species breakdown is shown in Supplementary Figure S4, revealing consistent patterns of STRING support of ProteomeLM predictions across diverse pathogens. Figure 3E further reports AUC values for each of the 19 species, ranging from 0.87 to 0.92. These results confirm that ProteomeLM generalizes well and maintains consistently strong performance across highly diverse taxa.

Together, these findings demonstrate that ProteomeLM is a fast, scalable, and accurate framework for interactome screening. Its ability to recover known interactions and predict plausible novel ones,

combined with strongly reduced computational costs, demonstrates that it can be used as a replacement for coevolution-based filters in proteome-scale pipelines. ProteomeLM therefore opens the way to high-throughput interactome inference across species, including for organisms lacking curated interaction data.

ProteomeLM delivers state-of-the-art supervised PPI prediction

Since ProteomeLM spontaneously learns PPI through its attention coefficients (see Figure 2), and can provide fast PPI screening, it is tantalizing to further exploit this signal. Can features from ProteomeLM advance supervised sequence-based PPI prediction? To address this question, we introduce ProteomeLM-PPI, a new supervised PPI prediction network that employs both node-type features (i.e. embeddings from ESM-C and ProteomeLM for individual proteins) and edge-type features (i.e. ProteomeLM attention values for pairs of proteins), see Figure 4A.

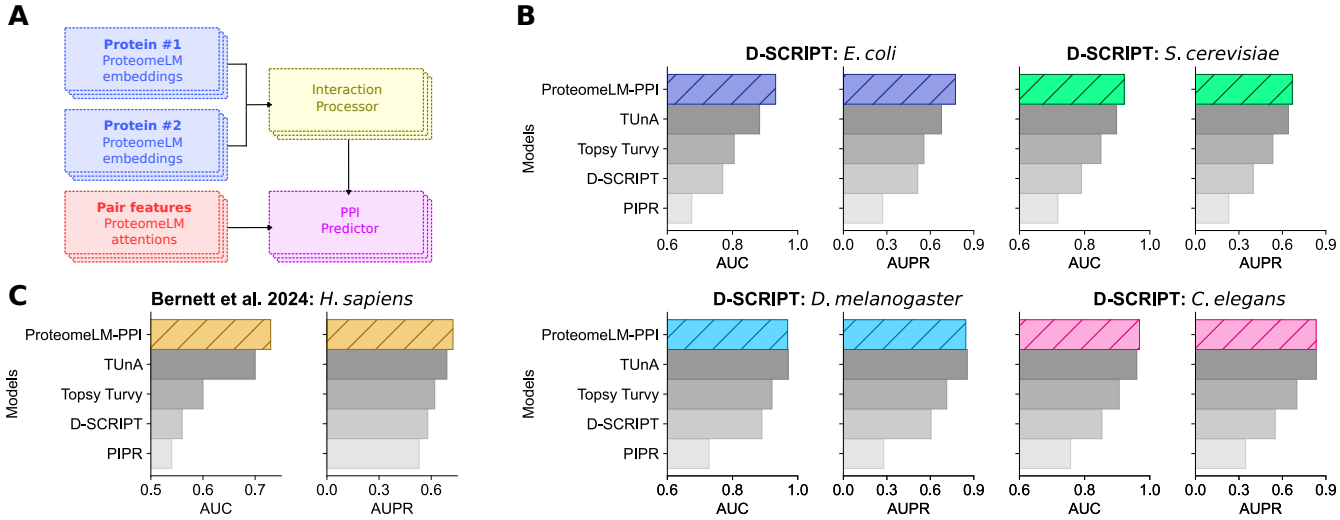

Figure 4: **Supervised prediction of PPI using ProteomeLM. (A)** Architecture of the supervised model trained to predict protein-protein interactions (PPI). The model comprises four components. For each protein in a candidate pair, its representations from ProteomeLM-S and ESM-C are independently processed through a feature encoder (blue), which is a feedforward neural network with two hidden layers. The resulting node features are combined in an interaction processor (yellow) that captures joint information. In parallel, edge features from ProteomeLM attention coefficients are processed through a separate module. The final PPI predictor is a classifier that integrates both node- and edge-level features to predict the interaction probability. **(B)** Cross-species generalization performance on the D-SCRIPT dataset [21], compared to state-of-the-art methods. All models are trained and validated on human PPI and tested on other species, and results are averaged over five replicates. **(C)** Performance on the dataset from [64], averaged over five technical replicates and compared to state-of-the-art methods.

We trained and evaluated ProteomeLM-PPI on two PPI datasets. The first one is the multi-species D-SCRIPT dataset [21], already used in the previous sections. The second one is a human-specific dataset which was designed to address common biases in the training, validation, and test splits of previous PPI benchmarks, in particular leakage [64]. For both datasets, we used the same training, validation, and test splits as in the recent TUnA method [65]. Our model was trained on the training set, with early stopping based on validation performance to avoid overfitting.

In the D-SCRIPT dataset, the training set is composed of human PPI and the validation set comprises held-out human PPI [65]. Testing on PPI from other species thus allows to assess cross-species generalization power. Figure 4B shows that ProteomeLM-PPI outperforms state-of-the-art methods on *S. cerevisiae* and *E. coli*, and performs comparably to them on *D. melanogaster* and *C. elegans*. In particular, it leads to an

AUPR improvement of more than 0.1 (from 0.67 to 0.79) over the previous state of the art, TUnA [65] on *E. coli*. These results highlight ProteomeLM's strong ability to capture PPI signals, its robustness across diverse branches of the tree of life, and its ability to generalize from one species to other ones.

In Figure 4C, we report results on the dataset from Ref. [64]. We observe that ProteomeLM-PPI yields very strong AUC and AUPR scores, consistently reaching or outperforming state-of-the-art methods including TUnA, which employed ESM-2 embeddings. This result shows the robustness of ProteomeLM-PPI to biases of PPI prediction benchmarks [64].

To better understand the contribution of each feature used by ProteomeLM-PPI, we evaluate different combinations of these features in Supplementary Figure S5. We observe that both ProteomeLM embeddings and attention coefficients are individually informative, and that their combination consistently yields the best predictive performance. This suggests that embeddings and attention coefficients contain complementary information that is synergistic for PPI prediction.

ProteomeLM improves gene essentiality prediction

While we focused on PPI prediction so far, ProteomeLM is a foundation model that can be used for diverse tasks where proteome-level information is important. Here, we consider another important task, which consists in predicting which genes are essential, i.e. necessary for survival or reproduction of an organism. Both protein or gene sequence on the one hand, and genomic context and protein-protein interactions on the other hand, have been found to matter for predicting essentiality [56]. Can ProteomeLM, which exploits these diverse elements, advance gene essentiality prediction?

To address this question, we introduce ProteomeLM-Ess, a supervised predictor of gene essentiality that takes as input ProteomeLM embeddings. To train and test this classification model, we used the OGEE database [66], which collects gene essentiality data from 127 experimental studies, spanning 91 species, for a total of 213,608 labeled genes. To create the training, validation and test split, we clustered proteins based on sequence similarity (see Methods). ProteomeLM-Ess is a two-layer fully connected network, and can take as input ProteomeLM embeddings from any layer.

Figure 5A shows the performance of ProteomeLM-Ess versus the depth of the layer whose embeddings are used as its input, for all ProteomeLM sizes. This performance is also compared to that of a similar classifier that starts from ESM-C embeddings instead of ProteomeLM ones. We observe that classifiers based on ProteomeLM embeddings significantly outperform those based on ESM-C embeddings. This demonstrates that the contextualized whole proteome-aware information present in ProteomeLM embeddings allows to better capture gene essentiality than a protein language model. Besides, we observe that the embeddings from intermediate layers of ProteomeLM produce the best gene essentiality classifiers, which is consistent with our observations for unsupervised PPI prediction (see Figure 2D). Furthermore, the performance of ProteomeLM-Ess scales with ProteomeLM size. This suggests that larger models are able to encode more information about gene essentiality into their representations. Interestingly, larger sizes appear to be more useful for gene essentiality than for PPI prediction (see Figure S4). Overall, the best performing version of ProteomeLM-Ess is the one trained on the embeddings of layer 8 of ProteomeLM-L, yielding an AUC of 0.93.

Can ProteomeLM-Ess generalize to unseen proteomes? To address this question, we held out the entire proteomes of *S. cerevisiae* and of *E. coli* during training. Figure 5B (top) shows the performance of ProteomeLM-Ess on these two held-out proteomes. We obtain good results, especially in *E. coli*, where 71% (resp. 2%) of genes experimentally determined to be essential (resp. non-essential) were correctly predicted as essential (resp. non-essential) by ProteomeLM-Ess. To further explore generalization ability, we collected essentiality data for the synthetic cells JCVI-Syn1.0 [67] and JCVI-Syn3A [68, 69], which are not present in OGEE. Moreover, JCVI-Syn3A has a genome that is engineered to be close to minimal, so it is both far from genomes in the training set and a stringent test of gene essentiality prediction. Figure 5B (bottom) shows that we obtain good performance in these cases too, even for JCVI-Syn3A. Hence, ProteomeLM-Ess is able to generalize to unseen taxa.

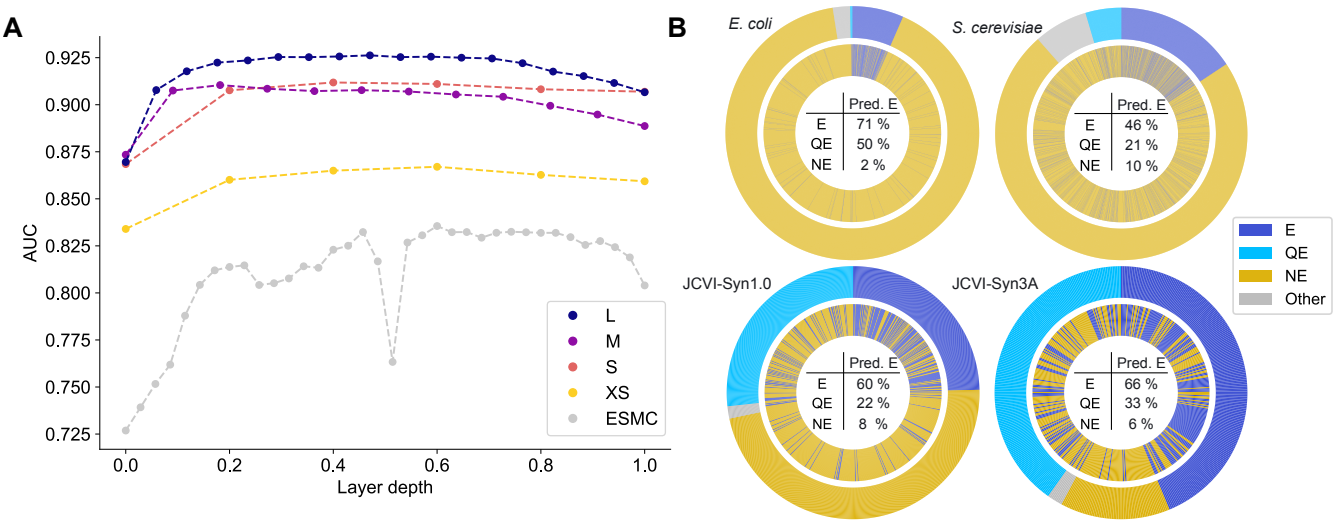

Figure 5: **Gene essentiality prediction with ProteomeLM-Ess. (A)** Area under the ROC curve (AUC) for binary classifiers that take as input embeddings from each layer of ProteomeLM (ProteomeLM-Ess) or ESM-C. To facilitate comparison between models with different sizes, the AUC is plotted versus the normalized layer depth, i.e. the ratio of the layer index to the number of layers. **(B)** Comparison between ProteomeLM-Ess predictions (inner rings) and experimental essentiality labels (outer rings) for *E. coli* K-12, *S. cerevisiae*, and the synthetic cells JCVI-Syn1.0 and JCVI-Syn3A. ProteomeLM-Ess uses embeddings from layer 8 of ProteomeLM-L. These genomes are not in the training set of ProteomeLM-Ess. For each of them, we predict as essential the top N genes in terms of ProteomeLM-Ess score, where N is the number of genes experimentally labeled essential. While ProteomeLM-Ess classifies genes as essential (E) or non-essential (NE), some genes in these species (many in minimal cells) are labeled as quasi-essential (QE). The percentage of genes labeled as E, QE or NE that are predicted as E by ProteomeLM-Ess is indicated in each case. AUC values: *E. coli*, 0.95; *S. cerevisiae*, 0.81; JCVI-Syn1.0, 0.88; JCVI-Syn3A, 0.83.

Discussion

We introduced ProteomeLM, a transformer-based language model that learns contextualized protein representations from complete proteomes spanning the whole tree of life. Trained to reconstruct masked protein representations from the other proteins of a proteome, ProteomeLM learns dependencies between proteins that reflect functional and evolutionary constraints. This makes it well-suited to predict emergent proteome-scale properties. We showed that ProteomeLM spontaneously captures PPI in its attention heads, without any supervision. We then demonstrated that ProteomeLM can be used as a highly scalable and effective tool for fast whole-interactome screening, yielding substantially higher accuracy than DCA, with a computational cost reduced by up to 6 orders of magnitude. This makes ProteomeLM a strong choice for large-scale PPI screening tasks, including in poorly annotated organisms. Next, we designed ProteomeLM-PPI, a supervised PPI prediction network that leverages both embeddings and attention coefficients from ProteomeLM, and achieves state-of-the-art performance across species and benchmarks. Finally, we trained ProteomeLM-Ess, a supervised classifier for gene essentiality prediction that starts from ProteomeLM embeddings. We showed that classifiers trained on ProteomeLM embeddings outperform those trained on ESM-C embeddings for this task, confirming that ProteomeLM's contextualized embeddings encode biologically relevant proteome-scale information.

In summary, ProteomeLM is a novel framework that learns whole proteome-aware protein representations across the tree of life. As a foundation language model trained using the MLM objective, ProteomeLM can be used for many downstream tasks. The possibility of using the pre-trained ProteomeLM, and even of using pre-computed ProteomeLM embeddings, means that using it in inference for downstream tasks can

be very computationally efficient. We demonstrated that ProteomeLM reaches very strong performance on tasks ranging from PPI screening to gene essentiality prediction, and combines speed, interpretability, and accuracy. We expect that proteome-aware language models will become increasingly important in the coming years, enabling new ways of modeling emergent biological properties at the scale of proteomes and cells.

Integrating local genomic context to protein language model embeddings has proved very useful for function inference [70] and genome annotation [71]. ProteomeLM demonstrates the power of integrating whole-proteome information across species. ProteomeLM reasons on proteomes at a coarse-grained level, representing each protein by a global representation from the protein language model ESM-C. These light representations allowed us to maintain reasonable context sizes and to work with full proteomes. In the future, the progress of long-context language models may enable ProteomeLM to directly operate at the amino acid level across whole proteomes, paving the way to localized cross-protein interactions and more granular modeling of functional dependencies. One could even envision using full genomic nucleotide sequences, in the spirit recent models that work on local genomic neighborhoods [15–17]. While this may become possible, we expect models working on complete raw genomic sequences to be much heavier and to require much more training data, as they need e.g. to learn about coding regions and codons. Models operating on raw genomic sequences can address regulatory functions or the role of non-coding DNA [15–17], which is beyond the scope of ProteomeLM. However, once they reach whole genomes, it remains to be seen whether they can efficiently and accurately predict whole-genome tasks such as PPI and gene essentiality, where ProteomeLM excels.

A major asset of ProteomeLM, allowed by our introduction of functional encodings instead of positional encodings, is that it can reason on proteomes across the tree of life. Nevertheless, in terms of absolute scores, ProteomeLM's performance remains stronger on prokaryotes than on eukaryotes. This may be due to the relative scarcity of high-quality eukaryotic proteomes in the training dataset and to the higher complexity of eukaryotic genome organization and gene regulation. While fine-tuning ProteomeLM on eukaryotic-only data, or training it only on that data, brings minor improvements on eukaryotes, it comes at the cost of reduced performance on bacteria. As eukaryotic genomic databases continue to grow, larger training sets may allow us to overcome this limitation. Along these lines, expanding the training set to include more eukaryotic proteomes, including less-well annotated ones, and metagenomic proteomes, should further improve ProteomeLM's performance across taxonomic groups.

Another interesting perspective for future work is to train ProteomeLM using embeddings from protein language models trained over additional modalities, beyond sequences, such as structure (SaProt [72], ProstT5 [73], ESM-3 [74], ProtTrek [75]). This may help ProteomeLM infer more complex functional relationships. This direction is particularly promising for PPI prediction, since protein structure is important in PPI, and since integrating structural information via graph neural networks has proved valuable for supervised PPI prediction [21].

ProteomeLM opens the way to many applications. Given its great computational efficiency at inference, ProteomeLM can be used to predict whole PPI networks in multiple species, and to study the evolution of these networks at a large scale. Similarly, ProteomeLM-Ess enables high-throughput *in silico* screens of gene essentiality, and comparisons across species. It would also be interesting to investigate joint essentiality [76–80] and environment-dependent essentiality [81] using ProteomeLM. Many other downstream tasks can potentially be addressed starting from ProteomeLM embeddings. This includes, for instance, context-aware fitness prediction, with the perspective of studying fitness landscapes on a large scale.

Methods

Architecture and training of ProteomeLM

Dataset. We collected 31,947 proteomes from OrthoDB (version 12) [59]. Each of these proteomes is a list of protein sequences annotated by their respective orthologous groups. An orthologous group comprises

descendants of a common ancestral gene, separated by speciation, and usually retaining the same function. It is linked to functional annotations from Gene Ontology (GO) [82, 83], describing protein localization and biological processes. However, here, we do not use these functional annotations. Our functional encoding (described below) only relies on orthologous groups. Note that the definition of an orthologous group operates at a specific level of orthology. Here, we use all these levels (see below).

Protein representations. We used the ESM-Cambrian (ESM-C) model with 600 million parameters [11] to represent each of the 162 million proteins in our dataset. ESM-C is a protein language model, trained on a vast corpus of sequences, which is a successor of ESM-2 [9]. Note that, contrary to ESM-3, which is multimodal [74], it is a sequence-only model. We selected ESM-C because of its strong performance in capturing the structural and functional properties of protein sequences. Each protein sequence was encoded by ESM-C, and the per-amino acid ESM-C embeddings were averaged to obtain a global embedding of the protein with a fixed dimension of 1152.

To optimize computational efficiency, protein sequences were batched based on their length, reducing the overhead associated with the model's quadratic time complexity. The whole embedding computation process took 192 GPU-hours on one H100 GPU.

Functional encoding. In natural language processing [84], and also for protein sequences [8, 11, 85] and for genomic sequences [16, 70], BERT models usually rely on positional encoding, which provides the model with information on the order of tokens (words in a sentence, amino acids in a protein chain, nucleotides along the genome). However, such a positional encoding is not appropriate for our purpose, given the lack of conservation of genomic order across diverse species, and the lack of correlation between proximity along the genome and functional relationship or interaction in eukaryotic genomes. Instead, we designed a *functional encoding* based on OrthoDB orthologous groups.

We constructed a hierarchical representation of each OrthoDB group. For this, we defined a representation of each leaf orthologous group (i.e., each orthologous group containing proteins but not orthologous subgroups) by averaging the ESM-C embedding of each protein in that group. We then propagated these representations up the orthologous group hierarchy by recursively averaging the representations of each group's immediate subgroups, assigning equal weight to each child. We save the representations obtained at each taxonomic level, and use all of them (see next paragraph). For each protein, this gives rise to a representation of the hierarchy of its orthologs, summarizing protein family membership, which we employ as a functional encoding. It is given as input to ProteomeLM together with the ESM-C embedding of that protein.

Architecture. We trained a transformer encoder from scratch to learn complex relationships between the ESM-C embeddings of different proteins in a proteome. The core of the model is the DistillBERT architecture, available in Hugging Face's `transformers` library [86]. We used FlashAttention-2 [87] to accelerate training and inference.

Each proteome is represented as a list of protein embeddings with their associated functional encodings. For each protein, the functional encoding is sampled randomly among the representations of its orthologous groups at all taxonomic levels, allowing the functionality of each protein to be represented at any taxonomic level. At the input stage, both the protein encoding and its functional encoding pass through two separate embedding modules, each consisting of a single linear layer.

Training objective and loss design. ProteomeLM is trained using a masked language modeling (MLM) objective adapted to proteome-level inputs. During the training of ProteomeLM, we limit the size of the proteomes to 4096 by randomly subsampling proteins when proteomes are longer. While longer inputs could in principle be employed, since we use FlashAttention, which has linear memory complexity, we chose to limit the length of the input to 4096 to reduce computational time, which still has quadratic

memory complexity. We randomly mask 50% of the protein representations within a proteome, while their functional encodings are kept unmasked. Masked proteins are replaced by their functional encoding, and the model is trained to reconstruct the original protein embeddings based on contextual signals from the rest of the proteome.

The standard masked language modeling loss cannot be applied here, because we work directly with continuous input and not with tokens. A straightforward alternative loss function for this task would be the mean squared error (MSE) between actual and predicted embeddings. However, as will be shown below, this approach resulted in a degenerate solution, where the model simply reproduced the functional encoding. This behavior likely stems from the high similarity between functional encodings and protein embeddings within conserved families. To address this challenge, we introduced the following polar loss function:

$$\mathcal{L}(\hat{x}, x, \bar{x}) = \text{CosineEmbeddingLoss}\left[(\hat{x} - \bar{x}), (x - \bar{x})\right] + \left(\|\hat{x} - \bar{x}\|_2 - \|x - \bar{x}\|_2\right)^2 , \tag{1}$$

where x is the true protein embedding, while $\hat{x}$ is the embedding predicted by ProteomeLM, and $\bar{x}$ is the functional encoding. This loss jointly enforces directional alignment of the residuals, which are the differences between the predicted and true embeddings, and accurate prediction of the Euclidean norms of these residuals. This loss is minimized if and only if $\hat{x} = x$. Moreover, it avoids collapse. Indeed, the "lazy" solution where the model just predicts the functional encoding as the reconstructed embedding ($\hat{x} = \bar{x}$) would result in a high loss, due to the high cosine embedding loss between ground truth and reconstructed residuals. More details are given below, in the paragraph titled "Comparison of losses".

Training dynamics and scaling behavior. We trained four variants of ProteomeLM, differing by model sizes: XS (5.6M parameters), S (36M), M (112M), and L (328M). All models were trained for 210 epochs on a dataset comprising 31,000 proteomes with a total of 160 million proteins. Validation loss was measured on a 2% held-out set of proteomes randomly sampled from the training set.

Training remained stable across all model sizes, showing smooth convergence, see Figure 6A. Figure 6A and B show that performance, assessed by loss value, improved steadily from XS to M, suggesting that the model benefits from increased capacity. However, the L model failed to outperform M, and in some cases showed degraded performance. In particular, in Figure 6B, the trend follows a scaling law from XS to M, before performance degrades for the L model. We attribute this to overfitting, given that the number of trainable parameters exceeds the number of unique training proteins in the training set for ProteomeLM-L. To rule out architecture-specific factors, we tested variants of ProteomeLM-L with different numbers of layers, heads, and embedding dimensions. These variants exhibited similar behavior in early training, reinforcing the interpretation that training data volume is the limiting factor.

In Figure S6, we show the training dynamics of each attention head in ProteomeLM-S by tracking their AUC for unsupervised PPI recovery during training. Certain heads become increasingly predictive of PPI as training progresses. Some of them display taxon-specific specialization, while others exhibit consistent predictive power across species. These results highlight that ProteomeLM's attention heads learn distinct, biologically meaningful signals during training.

We also used the D-Script dataset [21] to assess the performance of the models on PPI recovery (training and validation on disjoint sets of *H. sapiens* PPI, test on other species). The left panel of Figure 6C shows that, for unsupervised PPI recovery, AUPR increases with model size up to M, and decreases for size L, thus confirming the trend observed for the loss. Note also that performance on human data slightly decreased from S to M, indicating that larger models do not always generalize better across all species. The right panel of Figure 6C shows that AUPR increases over model size on eukaryotes, while it degrades after model S on *E. coli*, again showing the better generalization capabilities of the smaller models.

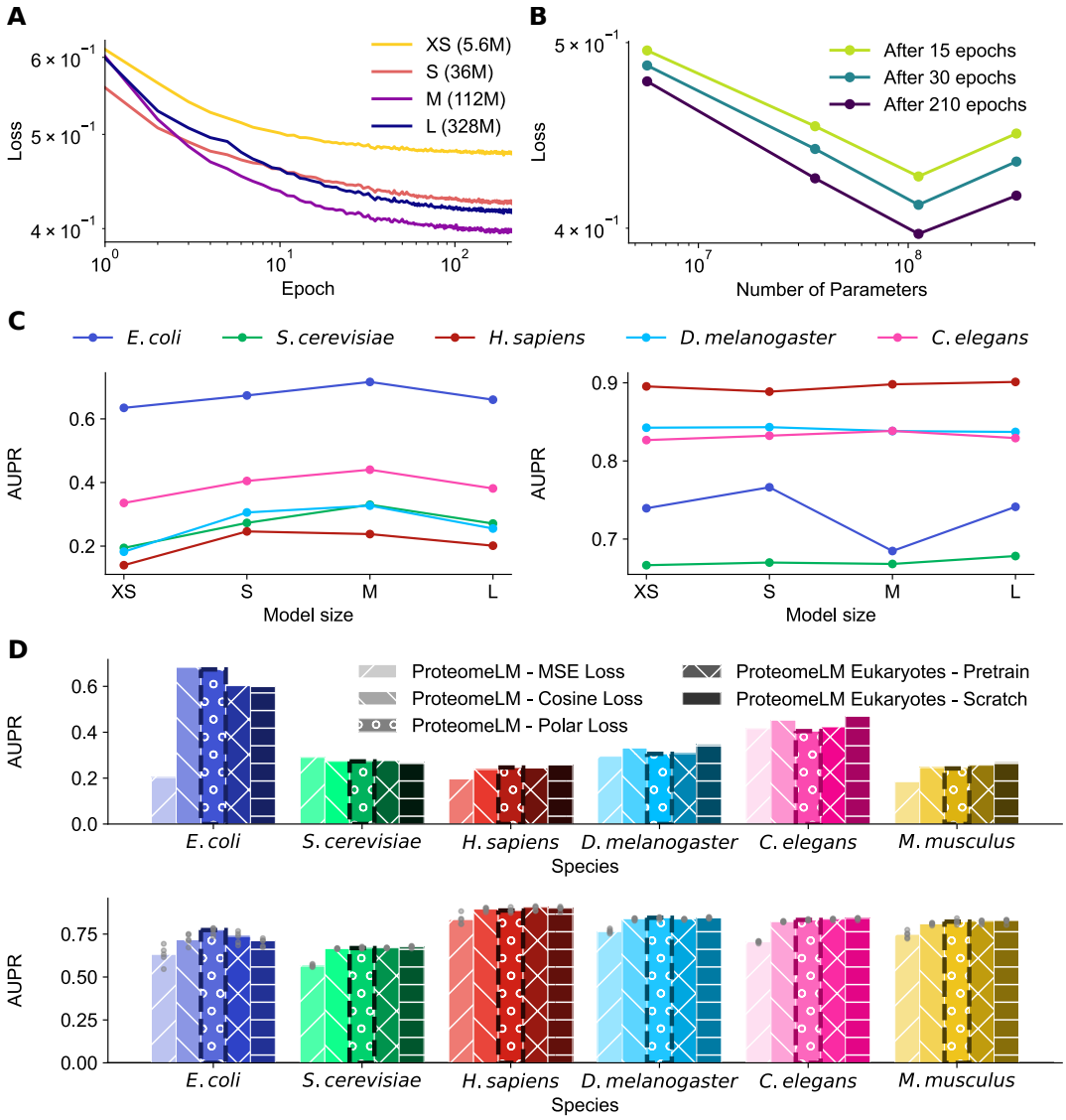

Figure 6: **Training dynamics, scaling behavior, and training strategies for ProteomeLM. (A)** Evaluation loss during training for four ProteomeLM model sizes: XS (5.6M parameters), S (36M), M (112M), and L (328M). **(B)** The final evaluation loss is shown for three different epochs. **(C)** The AUPR for unsupervised PPI prediction using summed attention coefficients over all heads and layers (left) and supervised PPI prediction through the ProteomeLM-PPI architecture (right) is shown across five species on the D-SCRIPT dataset. **(D)** Comparison of three different loss functions (MSE, cosine and polar, where polar is the one retained throughout), and evaluation of two training strategies focused on eukaryotic data (fine-tuning and training from scratch). AUPR are shown for both unsupervised (top) and supervised (bottom) PPI prediction tasks. As in C (left panel), summed attention coefficients over all heads and layers are used for unsupervised prediction. C-D: All models used in these comparisons were trained for 210 epochs under identical hardware and random seed conditions to ensure fair evaluation.

Comparison of losses. We evaluated our polar loss function against two natural alternatives, namely the mean squared error (MSE) loss $\text{MSELoss}(\hat{x}, x)$ and the cosine embedding loss $\text{CosineEmbeddingLoss}[(\hat{x} - \bar{x}), (x - \bar{x})]$. (Note that we do not consider $\text{CosineEmbeddingLoss}[\hat{x}, x]$ because the vectors x and $\hat{x}$ tend to have a similar orientation anyway within each protein family.) For our comparison of loss functions, we trained a version of ProteomeLM for 72h with each of these two alternative losses. Performance comparisons were conducted in both unsupervised and supervised protein-protein interaction (PPI) prediction tasks. In the unsupervised setting, we used the sum of attention weights over all heads and layers as a simple estimate of interaction probability for each protein pair. In the supervised setting, we trained a downstream classifier (see Figure 4A) using frozen representations obtained from models trained with each loss function.

As shown in Figure 6D, our polar loss consistently outperformed both the MSE loss and the cosine loss across multiple evaluation metrics. In particular, the unsupervised AUC scores were generally performing higher when the model was trained with the polar loss or the cosine embedding loss than with the MSE loss. Likewise, in the supervised PPI prediction task, the models trained with the polar loss yielded higher precision-recall performance (AUPR). The MSE loss showed the weakest performance in both settings. As mentioned above, it is because the model then converges towards a degenerate solution, where the model simply reproduced the functional encoding ($\hat{x} = \bar{x}$).

Note that, in addition to predictive accuracy, we observed that the polar loss produced embeddings with properties more closely aligned to those of ESM-C, thus facilitating interoperability between models, e.g. improving compatibility in transfer learning applications. These results validate the polar loss as an appropriate objective for reconstructing protein embeddings within a proteome context.

Improving performance on eukaryotic data. We observe that ProteomeLM's accuracy is comparatively lower on eukaryotic datasets than on prokaryotic ones (see Figures 2 and 4). We explored two approaches to improve performance on eukaryotes: fine-tuning the pretrained ProteomeLM on eukaryotic data, and training a new model from scratch on eukaryotic data only.

Figure 6D shows that both approaches provide moderate improvements on eukaryotic benchmarks, both for unsupervised and supervised PPI prediction. However, these improvements come with a decrease in performance on prokaryotes such as *E. coli*, indicating a trade-off between specialization and generalization.

Given our goal to design a model that works across diverse organisms, we retained the baseline ProteomeLM models for our main analyses. However, the specialized eukaryotic alternatives can be valuable for specific applications.

Supervised protein-protein interaction prediction: ProteomeLM-PPI

Architecture and input. The supervised ProteomeLM-PPI model relies on a modular neural network that processes both individual protein embeddings (node features) and attention coefficients (edge features) through distinct but integrated modules, see Figure 4A.

Specifically, the node feature module reduces each protein embedding from 640 to 256 dimensions through two layers ($640 \to 512 \to 256$), with layer normalization and dropout to enhance stability and weight regularization. To model the interaction between two proteins, the network combines their transformed representations by concatenating each of the two representations, their element-wise multiplication, and their absolute difference, thus resulting in a 1024-dimensional vector (4×256), which is then compressed to 64 dimensions through the interaction processor ($1024 \to 128 \to 64$). In parallel, the edge feature module reduces the 48-dimensional input (from the 48 attention heads of ProteomeLM-S) to 32 dimensions ($48 \to 64 \to 32$). The 64-dimensional processed interaction features from the interaction processor are then concatenated with the 32-dimensional pairwise feature vector to form a 96-dimensional input to the final PPI prediction classifier module. This input then passes through two layers (dimensions: $96 \to 128 \to 64$), before the PPI predictor outputs a final interaction score. This modular design enables the model to flexibly integrate different sets of learned features while maintaining strong inductive biases for capturing protein-protein relationships.

Training. To train the model, we used a train-validation-test split. During training, the model is optimized on the training set, while its performance is monitored on the validation set to apply early stopping, preventing overfitting. During training, the proteins pairs of the training set are fed into the model in mini-batches as triplets comprising ProteomeLM embeddings of both proteins involved, and ProteomeLM attentions weights between the two of them. The training minimizes binary cross-entropy with logits using the Adam optimizer [88]. At each epoch, predictions on the validation set are evaluated using the area under the precision-recall curve (AUPR), and the best model state is saved based on this metric. After training, the best model is evaluated on the held-out test set, and we report AUC and AUPR.

Our training approach ensures a fair assessment of the model's generalization ability by performing model selection on a dedicated validation set, rather than the test set, thereby avoiding overfitting to the test data. For both the dataset from [64] and the D-SCRIPT dataset [21], clean and non-overlapping splits into training, validation, and test sets were already available, and we employed them.

Supervised gene essentiality prediction: ProteomeLM-Ess

Architecture and input. ProteomeLM-Ess is a two-layer fully connected classifier that takes as input embeddings from any ProteomeLM model, has a hidden layer of size 2048, and outputs two logits, which are normalized with a softmax function to obtain an essentiality score. In the hidden layer, ProteomeLM-Ess has a ReLU activation and dropout with probability 0.5. Protein embeddings are normalized using the genome-wide mean and standard deviation before being given as input to ProteomeLM-Ess.

Data and training. ProteomeLM-Ess is trained in a supervised way, using ProteomeLM embeddings together with essentiality labels from the OGEE database [66]. We recovered the protein sequences associated to the essentiality labels from other databases, by matching the gene names (gene IDs) provided by OGEE. Specifically, we collected protein sequences from UniProt [89], NCBI [90], *Saccharomyces* Genome Database (SGD) [91] and Fitness Browser [92], and obtained data for 87 taxonomic IDs. Relying on curated complete proteomes whenever possible allowed to minimize ambiguities coming from the presence of isoforms or duplicate protein sequences (i.e. identical sequences with different protein IDs). The remaining duplicate entries were merged, while keeping both IDs.

Out of the 87 total genomes, we used 83 to train ProteomeLM-Ess, holding out the genomes of *S. cerevisiae* and of 4 strains of *E. coli*. We also collected essentiality data for the synthetic cells JCVI-Syn1.0 [67] and JCVI-Syn3A [68, 69], to evaluate the model after training. For the 83 genomes used for training, we split the proteins into training, validation and test sets by clustering proteins across all genomes according to sequence similarity. Specifically, we clustered sequences using MMSeqs2 [93] with a 40% similarity threshold. We designed our split so that if two labeled proteins belong to the same cluster, then they are either both in the training set, in the validation set, or in the test set. The data split is performed at the protein level and not at the genome level, to avoid the model relying on sequence similarity between, say, two orthologs in similar genomes, as a shortcut to predict essentiality. All protein sequences are given as input to ProteomeLM to build contextualized embeddings. The training procedure and objective used for ProteomeLM-Ess are the same as the ones used for ProteomeLM-PPI (see above).

Data and code availability

We provide code for training and using ProteomeLM at https://github.com/Bitbol-Lab/ProteomeLM. Code weights, and all the material needed to reproduce our results, are available there.

Acknowledgments

This research was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 851173, to A.-F. B.). We thank Jing Zhang

and Ian Humphreys for useful discussions and additional material for PPI screening, Liedewij Laan for useful discussions about gene essentiality and James Sáenz for useful discussions about minimal cells.

References

[1] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, 2019.

[2] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

[3] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. USA*, 118(15):e2016239118, 2021. ISSN 0027-8424. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/content/118/15/e2016239118.

[4] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pages 2020–12, 2020.

[5] Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Rajani. BERTology meets biology: Interpreting attention in protein language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YWtLZvLmud7.

[6] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language modeling for protein generation. *bioRxiv*, page 2020.03.07.982272, 2020. URL http://arxiv.org/abs/2004.03497.

[7] A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos, C. Xiong, Z. Z. Sun, R. Socher, J. S. Fraser, and N. Naik. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.*, 41(8):1099–1106, 2023.

[8] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA Transformer. *Proceedings of the 38th International Conference on Machine Learning*, 139:8844–8856, 2021.

[9] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[10] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

[11] ESM Team et al. ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning. *Evolutionary Scale Website, https://www.evolutionaryscale.ai/blog/esm-cambrian*, 2024.

[12] Vineet Thumuluri, José Juan Almagro Armenteros, Alexander Rosenberg Johansen, Henrik Nielsen, and Ole Winther. Deeploc 2.0: multi-label subcellular localization prediction using protein language models. *Nucleic acids research*, 50(W1):W228–W234, 2022.

[13] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *NeurIPS*, 2021. URL https://openreview.net/forum?id=uXc42E9ZPFs.

[14] D. Miller, A. Stern, and D. Burstein. Deciphering microbial gene function using natural language processing. *Nat. Commun.*, 13(1):5731, 2022.

[15] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome. *Bioinformatics*, 37 (15):2112–2120, 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab083. URL https://doi.org/10.1093/bioinformatics/btab083.

[16] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P de Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

[17] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, et al. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

[18] Gonzalo Benegas, Carlos Albors, Alan J. Aw, Chengzhong Ye, and Yun S. Song. GPN-MSA: an alignment-based DNA language model for genome-wide variant effect prediction. *bioRxiv*, page 2023.10.10.561776, 2024. doi: 10.1101/2023.10.10.561776. URL https://www.biorxiv.org/content/early/2024/04/06/2023.10.10.561776.

[19] Yunha Hwang, Andre L. Cornman, Elizabeth H. Kellogg, Sergey Ovchinnikov, and Peter R. Girguis. Genomic language model predicts protein co-regulation and function. *Nature Communications*, 15 (1):2880, 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-46947-9. URL https://www.nature.com/articles/s41467-024-46947-9.

[20] Bin Shao and Jiawei Yan. A long-context language model for deciphering and generating bacteriophage genomes. *Nature Communications*, 15(1):9392, 2024.

[21] Samuel Sledzieski, Rohit Singh, Lenore Cowen, and Bonnie Berger. D-script translates genome to phenome with sequence-based, structure-aware, genome-scale predictions of protein-protein interactions. *Cell Systems*, 12(10):969–982, 2021.

[22] S. V. Rajagopala, P. Sikorski, A. Kumar, R. Mosca, J. Vlasblom, R. Arnold, J. Franca-Koh, S. B. Pakala, S. Phanse, A. Ceol, R. Hauser, G. Siszler, S. Wuchty, A. Emili, M. Babu, P. Aloy, R. Pieper, and P. Uetz. The binary protein-protein interaction landscape of Escherichia coli. *Nat. Biotechnol.*, 32 (3):285–290, 2014.

[23] Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, et al. The biogrid database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Science*, 30(1):187–200, 2021.

[24] Peter D Karp, Wai Kit Ong, Suzanne Paley, Richard Billington, Ron Caspi, Carol Fulcher, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Peter E Midford, et al. The ecocyc database. *EcoSal Plus*, 8(1):10–1128, 2018.

[25] Noemi Del Toro, Anjali Shrivastava, Eliot Ragueneau, Birgit Meldal, Colin Combe, Elisabet Barrera, Livia Perfetto, Karyn How, Prashansa Ratan, Gautam Shirodkar, et al. The intact database: efficient access to fine-grained molecular interaction data. *Nucleic acids research*, 50(D1):D648–D653, 2022.

[26] Rodrigo V. Honorato, Mikael E. Trellet, Brian Jiménez-García, Jörg J. Schaarschmidt, Marco Giulini, Victor Reys, Panagiotis I. Koukos, João P. G. L. M. Rodrigues, Ezgi Karaca, Gydo C. P. van Zundert, Jorge Roel-Touris, Charlotte W. van Noort, Zuzana Jandová, Adrien S. J. Melquiond, and Alexandre M. J. J. Bonvin. The HADDOCK2.4 web server for integrative modeling of biomolecular complexes. *Nature Protocols*, 19(11):3219–3241, 2024.

[27] Mohamed Amine Ketata, Cedrik Laue, Ruslan Mammadov, Hannes Stark, Menghua Wu, Gabriele Corso, Céline Marquet, Regina Barzilay, and Tommi S Jaakkola. Diffdock-pp: Rigid protein-protein docking with diffusion models. In *ICLR 2023-Machine Learning for Drug Discovery workshop*, 2023.

[28] Azam Shirali, Vitalii Stebliankin, Ukesh Karki, Jimeng Shi, Prem Chapagain, and Giri Narasimhan. A comprehensive survey of scoring functions for protein docking models. *BMC bioinformatics*, 26(1): 25, 2025.

[29] Ameya Harmalkar, Sergey Lyskov, and Jeffrey J Gray. Reliable protein–protein docking with alphafold, rosetta, and replica exchange. *Elife*, 13:RP94029, 2025.

[30] Richard Evans et al. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, page 2021.10.04.463034, 2021.

[31] Patrick Bryant, Gabriele Pozzati, and Arne Elofsson. Improved prediction of protein-protein interactions using alphafold2. *Nature communications*, 13(1):1265, 2022.

[32] U. Göbel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins*, 18(4):309–317, 1994.

[33] G. Casari, C. Sander, and A. Valencia. A method to predict functional residues in proteins. *Nat. Struct. Biol.*, 2(2):171–178, 1995.

[34] Michael Socolich, Steve W. Lockless, William P. Russ, Heather Lee, Kevin H. Gardner, and Rama Ranganathan. Evolutionary information for specifying a protein fold. *Nature*, 437(7058):512–518, 2005. ISSN 14764687. doi: 10.1038/nature03991.

[35] S. D. Dunn, L. M. Wahl, and G. B. Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, 2008.

[36] M. Weigt, R. A. White, H. Szurmant, J. A. Hoch, and T. Hwa. Identification of direct residue contacts in protein-protein interaction by message passing. *Proc. Natl. Acad. Sci. USA*, 106(1):67–72, 2009.

[37] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander. Protein 3D structure computed from evolutionary sequence variation. *PLOS ONE*, 6(12):e28766, 2011.

[38] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl. Acad. Sci. USA*, 108(49):E1293–1301, 2011.

[39] Anne-Florence Bitbol, Robert S Dwyer, Lucy J Colwell, and Ned S Wingreen. Inferring interaction partners from protein sequences. *Proc. Natl. Acad. Sci. USA*, 113(43):12180–12185, 2016.

[40] T. Gueudre, C. Baldassi, M. Zamparo, M. Weigt, and A. Pagnani. Simultaneous identification of specifically interacting paralogs and interprotein contacts by direct coupling analysis. *Proc. Natl. Acad. Sci. USA*, 113(43):12186–12191, 2016.

[41] Qian Cong, Ivan Anishchenko, Sergey Ovchinnikov, and David Baker. Protein interaction networks revealed by proteome coevolution. *Science*, 365(6449):185–189, 2019.

[42] Ian R Humphreys, Jimin Pei, Minkyung Baek, Aditya Krishnakumar, Ivan Anishchenko, Sergey Ovchinnikov, Jing Zhang, Travis J Ness, Sudeep Banjade, Saket R Bagde, et al. Computed structures of core eukaryotic protein complexes. *Science*, 374(6573):eabm4805, 2021.

[43] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA*, 96(8):4285–4288, 1999.

[44] G. Croce, T. Gueudré, M. V. Ruiz Cuevas, V. Keidel, M. Figliuzzi, H. Szurmant, and M. Weigt. A multi-scale coevolutionary approach to predict interactions between protein domains. *PLOS Comput. Biol.*, 15(10):e1006891, 2019.

[45] Idit Bloch, Dana Sherill-Rofe, Doron Stupp, Irene Unterman, Hodaya Beer, Elad Sharon, and Yuval Tabach. Optimization of co-evolution analysis through phylogenetic profiling reveals pathway-specific signals. *Bioinformatics*, 36(14):4116–4125, 2020. doi: 10.1093/bioinformatics/btaa281. URL https://doi.org/10.1093/bioinformatics/btaa281.

[46] D. Moi, L. Kilchoer, P. S. Aguilar, and C. Dessimoz. Scalable phylogenetic profiling using MinHash uncovers likely eukaryotic sexual reproduction genes. *PLOS Comput. Biol.*, 16(7):e1007553, 2020.

[47] A. M. Altenhoff, C. M. Train, K. J. Gilbert, I. Mediratta, T. Mendes de Farias, D. Moi, Y. Nevers, H. S. Radoykova, V. Rossier, A. Warwick Vesztrocy, N. M. Glover, and C. Dessimoz. OMA orthology in 2021: website overhaul, conserved isoforms, ancestral gene order and more. *Nucleic Acids Res.*, 49 (D1):D373–D379, 2021.

[48] E. Dembech, M. Malatesta, C. De Rito, G. Mori, D. Cavazzini, A. Secchi, F. Morandin, and R. Percudani. Identification of hidden associations among eukaryotic genes through statistical analysis of coevolutionary transitions. *Proc. Natl. Acad. Sci. USA*, 120(16):e2218329120, 2023.

[49] D. Stupp, E. Sharon, I. Bloch, M. Zitnik, O. Zuk, and Y. Tabach. Co-evolution based machine-learning for predicting functional interactions between human genes. *Nat. Commun.*, 12(1):6454, 2021.

[50] David Moi and Christophe Dessimoz. Reconstructing protein interactions across time using phylogeny-aware graph neural networks. *bioRxiv*, page 2022.07.21.501014, 2022. URL https://www.biorxiv.org/content/early/2022/07/22/2022.07.21.501014.

[51] N. Konno and W. Iwasaki. Machine learning enables prediction of metabolic system evolution in bacteria. *Sci. Adv.*, 9(2):eadc9130, 2023.

[52] Anna G Green, Hadeer Elhabashy, Kelly P Brock, Rohan Maddamsetti, Oliver Kohlbacher, and Debora S Marks. Large-scale discovery of protein interactions at residue resolution using co-evolution calculated from genomic sequences. *Nature communications*, 12(1):1396, 2021.

[53] A.-F. Bitbol. Inferring interaction partners from protein sequences using mutual information. *PLOS Comput. Biol.*, 14(11):e1006401, 2018.

[54] Umberto Lupo, Damiano Sgarbossa, and Anne-Florence Bitbol. Pairing interacting protein sequences using masked language modeling. *Proc. Natl. Acad. Sci. USA*, 121(27):e2311887121, 2024. doi: 10.1073/pnas.2311887121. URL https://www.pnas.org/doi/10.1073/pnas.2311887121.

[55] John Jumper et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596:583–589, 2021.

[56] O. Aromolaran, D. Aromolaran, I. Isewon, and J. Oyelade. Machine learning approach to gene essentiality prediction: a review. *Brief. Bioinform.*, 22(5):bbab128, 2021.

[57] Tristan Bepler and Bonnie Berger. Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6):654–669, 2021.

[58] David Moi and Christophe Dessimoz. Phylogenetic profiling in eukaryotes comes of age. *Proc. Natl. Acad. Sci. USA*, 120(19):e2305013120, 2023. doi: 10.1073/pnas.2305013120. URL https://www.pnas.org/doi/full/10.1073/pnas.2305013120.

[59] Dmitry Kuznetsov, Fredrik Tegenfeldt, Mosè Manni, Mathieu Seppey, Matthew Berkeley, Evgenia V Kriventseva, and Evgeny M Zdobnov. OrthoDB v11: annotation of orthologs in the widest sampling of organismal diversity. *Nucleic Acids Res.*, 51(D1):D445–D451, 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac998. URL https://doi.org/10.1093/nar/gkac998.

[60] Umberto Lupo, Damiano Sgarbossa, and Anne-Florence Bitbol. Protein language models trained on multiple sequence alignments learn phylogenetic relationships. *Nat. Commun.*, 13:6298, 2022.

[61] Damian Szklarczyk, Rebecca Kirsch, Mikaela Koutrouli, Katerina Nastou, Farrokh Mehryary, Radja Hachilif, Annika L Gable, Tao Fang, Nadezhda T Doncheva, Sampo Pyysalo, Peer Bork, Lars J Jensen, and Christian von Mering. The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest. *Nucleic Acids Res.*, 51(D1): D638–D646, 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1000. URL https://doi.org/10.1093/nar/gkac1000.

[62] Ian R Humphreys, Jing Zhang, Minkyung Baek, Yaxi Wang, Aditya Krishnakumar, Jimin Pei, Ivan Anishchenko, Catherine A Tower, Blake A Jackson, Thulasi Warrier, et al. Protein interactions in human pathogens revealed through deep learning. *Nature microbiology*, 9(10):2642–2652, 2024.

[63] Jing Zhang, Ian R Humphreys, Jimin Pei, Jinuk Kim, Chulwon Choi, Rongqing Yuan, Jesse Durham, Siqi Liu, Hee-Jung Choi, Minkyung Baek, et al. Computing the human interactome. *bioRxiv*, pages 2024–10, 2024.

[64] Judith Bernett, David B Blumenthal, and Markus List. Cracking the black box of deep sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, 25(2), 2024.

[65] Young Su Ko, Jonathan Parkinson, Cong Liu, and Wei Wang. Tuna: an uncertainty-aware transformer model for sequence-based protein–protein interaction prediction. *Briefings in Bioinformatics*, 25(5), 2024.

[66] Sanathoi Gurumayum, Puzi Jiang, Xiaowen Hao, Tulio L Campos, Neil D Young, Pasi K Korhonen, Robin B Gasser, Peer Bork, Xing-Ming Zhao, Li-jie He, and Wei-Hua Chen. OGEE v3: Online GEne Essentiality database with increased coverage of organisms and human cell lines. *Nucleic Acids Research*, 49(D1):D998–D1003, 2021.

[67] Clyde A. Hutchison, Ray-Yuan Chuang, Vladimir N. Noskov, Nacyra Assad-Garcia, Thomas J. Deerinck, Mark H. Ellisman, John Gill, Krishna Kannan, Bogumil J. Karas, Li Ma, James F. Pelletier, Zhi-Qing Qi, R. Alexander Richter, Elizabeth A. Strychalski, Lijie Sun, Yo Suzuki, Billyana Tsvetanova, Kim S. Wise, Hamilton O. Smith, John I. Glass, Chuck Merryman, Daniel G. Gibson, and J. Craig Venter. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.

[68] Marian Breuer, Tyler M Earnest, Chuck Merryman, Kim S Wise, Lijie Sun, Michaela R Lynott, Clyde A Hutchison, Hamilton O Smith, John D Lapek, David J Gonzalez, Valérie de Crécy-Lagard, Drago Haas, Andrew D Hanson, Piyush Labhsetwar, John I Glass, and Zaida Luthey-Schulten. Essential metabolism for a minimal cell. *eLife*, 8:e36842, 2019.

[69] Tiago Pedreira, Christoph Elfmann, Neil Singh, and Jörg Stülke. SynWiki: Functional annotation of the first artificial organism Mycoplasma mycoides JCVI-syn3A. *Protein Science*, 31(1):54–62, 2022.

[70] Yunha Hwang, Andre L Cornman, Elizabeth H Kellogg, Sergey Ovchinnikov, and Peter R Girguis. Genomic language model predicts protein co-regulation and function. *Nature communications*, 15(1): 2880, 2024.

[71] Nishant Jha, Joshua Kravitz, Jacob West-Roberts, Cong Lu, Antonio Pedro Camargo, Simon Roux, Andre Cornman, and Yunha Hwang. Gaia: An ai-enabled genomic context–aware platform for protein sequence annotation. *Science Advances*, 11(25):eadv5109, 2025.

[72] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. *bioRxiv*, page 2023.10.01.560349, 2023. URL https://www.biorxiv.org/content/early/2023/10/02/2023.10.01.560349.1.

[73] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Martin Steinegger, and Burkhard Rost. ProstT5: Bilingual language model for protein sequence and structure. *bioRxiv*, page 2023.07.23.550085, 2023. URL https://www.biorxiv.org/content/early/2023/07/25/2023.07.23.550085.

[74] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. URL https://www.science.org/doi/10.1126/science.ads0018.

[75] Jin Su, Xibin Zhou, Xuting Zhang, and Fajie Yuan. Protrek: Navigating the protein universe through tri-modal contrastive learning. *bioRxiv*, pages 2024–05, 2024.

[76] Michael Costanzo, Anastasia Baryshnikova, Jeremy Bellay, Yungil Kim, Eric D. Spear, Carolyn S. Sevier, Huiming Ding, Judice L.Y. Koh, Kiana Toufighi, Sara Mostafavi, Jeany Prinz, Robert P. St. Onge, Benjamin VanderSluis, Taras Makhnevych, Franco J. Vizeacoumar, Solmaz Alizadeh, Sondra Bahr, Renee L. Brost, Yiqun Chen, Murat Cokol, Raamesh Deshpande, Zhijian Li, Zhen-Yuan Lin, Wendy Liang, Michaela Marback, Jadine Paw, Bryan-Joseph San Luis, Ermira Shuteriqi, Amy Hin Yan Tong, Nydia van Dyk, Iain M. Wallace, Joseph A. Whitney, Matthew T. Weirauch, Guoqing Zhong, Hongwei Zhu, Walid A. Houry, Michael Brudno, Sasan Ragibizadeh, Balázs Papp, Csaba Pál, Frederick P. Roth, Guri Giaever, Corey Nislow, Olga G. Troyanskaya, Howard Bussey, Gary D. Bader, Anne-Claude Gingras, Quaid D. Morris, Philip M. Kim, Chris A. Kaiser, Chad L. Myers, Brenda J. Andrews, and Charles Boone. The Genetic Landscape of a Cell. *Science*, 327(5964):425–431, 2010. doi: 10.1126/science.1180823. URL https://www.science.org/doi/10.1126/science.1180823.

[77] Michael Costanzo, Benjamin VanderSluis, Elizabeth N. Koch, Anastasia Baryshnikova, Carles Pons, Guihong Tan, Wen Wang, Matej Usaj, Julia Hanchard, Susan D. Lee, Vicent Pelechano, Erin B. Styles, Maximilian Billmann, Jolanda van Leeuwen, Nydia van Dyk, Zhen-Yuan Lin, Elena Kuzmin, Justin Nelson, Jeff S. Piotrowski, Tharan Srikumar, Sondra Bahr, Yiqun Chen, Raamesh Deshpande, Christoph F. Kurat, Sheena C. Li, Zhijian Li, Mojca Mattiazzi Usaj, Hiroki Okada, Natasha Pascoe, Bryan-Joseph San Luis, Sara Sharifpoor, Emira Shuteriqi, Scott W. Simpkins, Jamie Snider, Harsha Garadi Suresh, Yizhao Tan, Hongwei Zhu, Noel Malod-Dognin, Vuk Janjic, Natasa Przulj, Olga G. Troyanskaya, Igor Stagljar, Tian Xia, Yoshikazu Ohya, Anne-Claude Gingras, Brian Raught, Michael Boutros, Lars M. Steinmetz, Claire L. Moore, Adam P. Rosebrock, Amy A. Caudy, Chad L. Myers, Brenda Andrews, and Charles Boone. A global genetic interaction network maps a wiring

diagram of cellular function. *Science*, 353(6306):aaf1420, 2016. doi: 10.1126/science.aaf1420. URL https://www.science.org/doi/10.1126/science.aaf1420.

[78] Michael Costanzo, Elena Kuzmin, Jolanda van Leeuwen, Barbara Mair, Jason Moffat, Charles Boone, and Brenda Andrews. Global Genetic Networks and the Genotype-to-Phenotype Relationship. *Cell*, 177(1):85–100, 2019. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2019.01.033. URL https://www.cell.com/cell/abstract/S0092-8674(19)30096-0.

[79] Elena Kuzmin, Benjamin VanderSluis, Wen Wang, Guihong Tan, Raamesh Deshpande, Yiqun Chen, Matej Usaj, Attila Balint, Mojca Mattiazzi Usaj, Jolanda van Leeuwen, Elizabeth N. Koch, Carles Pons, Andrius J. Dagilis, Michael Pryszlak, Jason Zi Yang Wang, Julia Hanchard, Margot Riggi, Kaicong Xu, Hamed Heydari, Bryan-Joseph San Luis, Ermira Shuteriqi, Hongwei Zhu, Nydia Van Dyk, Sara Sharifpoor, Michael Costanzo, Robbie Loewith, Amy Caudy, Daniel Bolnick, Grant W. Brown, Brenda J. Andrews, Charles Boone, and Chad L. Myers. Systematic Analysis of Complex Genetic Interactions. *Science (New York, N.Y.)*, 360(6386):eaao1729, 2018. ISSN 0036-8075. doi: 10.1126/science.aao1729. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6215713/.

[80] Enzo Kingma, Floor Dolsma, Leila Margarita Iñigo De La Cruz, and Liedewij Laan. Saturated transposon analysis in yeast as a one-step method to quantify the fitness effects of gene disruptions on a genome-wide scale. *bioRxiv*, page 2023.09.08.556793, 2023.

[81] Michael Costanzo, Jing Hou, Vincent Messier, Justin Nelson, Mahfuzur Rahman, Benjamin VanderSluis, Wen Wang, Carles Pons, Catherine Ross, Matej Ušaj, Bryan-Joseph San Luis, Emira Shuteriqi, Elizabeth N. Koch, Patrick Aloy, Chad L. Myers, Charles Boone, and Brenda Andrews. Environmental robustness of the global yeast genetic interaction network. *Science*, 372(6542):eabf8424, 2021. doi: 10.1126/science.abf8424. URL https://www.science.org/doi/10.1126/science.abf8424.

[82] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[83] Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. The gene ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031, 2023.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

[85] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3095381. URL https://ieeexplore.ieee.org/document/9477085. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

[86] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[87] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[88] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[89] The UniProt Consortium. UniProt: The Universal Protein Knowledgebase in 2025. *Nucleic Acids Research*, 53(D1):D609–D617, 2025.

[90] Eric W. Sayers, Jeffrey Beck, Evan E. Bolton, J. Rodney Brister, Jessica Chan, Ryan Connor, Michael Feldgarden, Anna M. Fine, Kathryn Funk, Jinna Hoffman, Sivakumar Kannan, Christopher Kelly, William Klimke, Sunghwan Kim, Stacy Lathrop, Aron Marchler-Bauer, Terence D. Murphy, Chris O'Sullivan, Erin Schmieder, Yuriy Skripchenko, Adam Stine, Francoise Thibaud-Nissen, Jiyao Wang, Jian Ye, Erin Zellers, Valerie A. Schneider, and Kim D. Pruitt. Database resources of the National Center for Biotechnology Information in 2025. *Nucleic Acids Research*, 53(D1):D20–D29, 2025.

[91] Stacia R. Engel, Suzi Aleksander, Robert S. Nash, Edith D. Wong, Shuai Weng, Stuart R. Miyasato, Gavin Sherlock, and J. Michael Cherry. Saccharomyces Genome Database: Advances in genome annotation, expanded biochemical pathways, and other key enhancements. *Genetics*, 229(3):iyae185, 2025.

[92] Morgan N. Price, Kelly M. Wetmore, R. Jordan Waters, Mark Callaghan, Jayashree Ray, Hualan Liu, Jennifer V. Kuehl, Ryan A. Melnyk, Jacob S. Lamson, Yumi Suh, Hans K. Carlson, Zuelma Esquivel, Harini Sadeeshkumar, Romy Chakraborty, Grant M. Zane, Benjamin E. Rubin, Judy D. Wall, Axel Visel, James Bristow, Matthew J. Blow, Adam P. Arkin, and Adam M. Deutschbauer. Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature*, 557(7706):503–509, 2018.

[93] Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):2542, 2018.

Supplementary Information

Contents

1 Estimate of compute requirements **24**

2 Supplementary figures **25**

1 Estimate of compute requirements

Here, we estimate the floating-point operations (FLOPs) required for each stage of our study and for Ref. [63]. For this, we employ the peak FP32 throughput of relevant GPUs and the reported wall-clock runtimes. All values represent theoretical upper bounds.

DCA [63]. According to the authors of Ref. [63] (Jing Zhang, private communication), the Direct Coupling Analysis (DCA) step was performed over 1–2 months using between 50 and 100 GPUs, including NVIDIA RTX 6000, RTX 8000, A100, and A40. For our estimate, we assume:

- 75 concurrent GPUs over 45 days (i.e., 81,000 GPU-hours);
- Even distribution across the four GPU models;
- Peak FP32 throughput: 16.3 TFLOPS (RTX 6000 and RTX 8000), 19.5 TFLOPS (A100), and 37.4 TFLOPS (A40).

The resulting compute requirement is:

$$\text{Total FLOPs} \approx 6.5 \times 10^{21} \text{ FLOP (6.5 ZFLOP)}.$$

ProteomeLM training. Our model was trained on a single NVIDIA H100 SXM5 GPU during 72 hours. With a peak FP32 throughput of 67 TFLOPS, the total estimated compute is:

$$\text{Total FLOPs} = 67 \times 10^{12} \times 72 \times 3600 \approx 1.74 \times 10^{19} \text{ FLOP (17.4 EFLOP)}.$$

This value assumes uninterrupted training with full GPU utilization.

ProteomeLM inference. A 10-minute inference run was performed on an NVIDIA RTX A6000 GPU. With a peak FP32 throughput of 38.7 TFLOPS, the total compute estimated is:

$$\text{Total FLOPs} = 38.7 \times 10^{12} \times 600 \approx 2.32 \times 10^{16} \text{ FLOP (23 PFLOP)}.$$

2 Supplementary figures

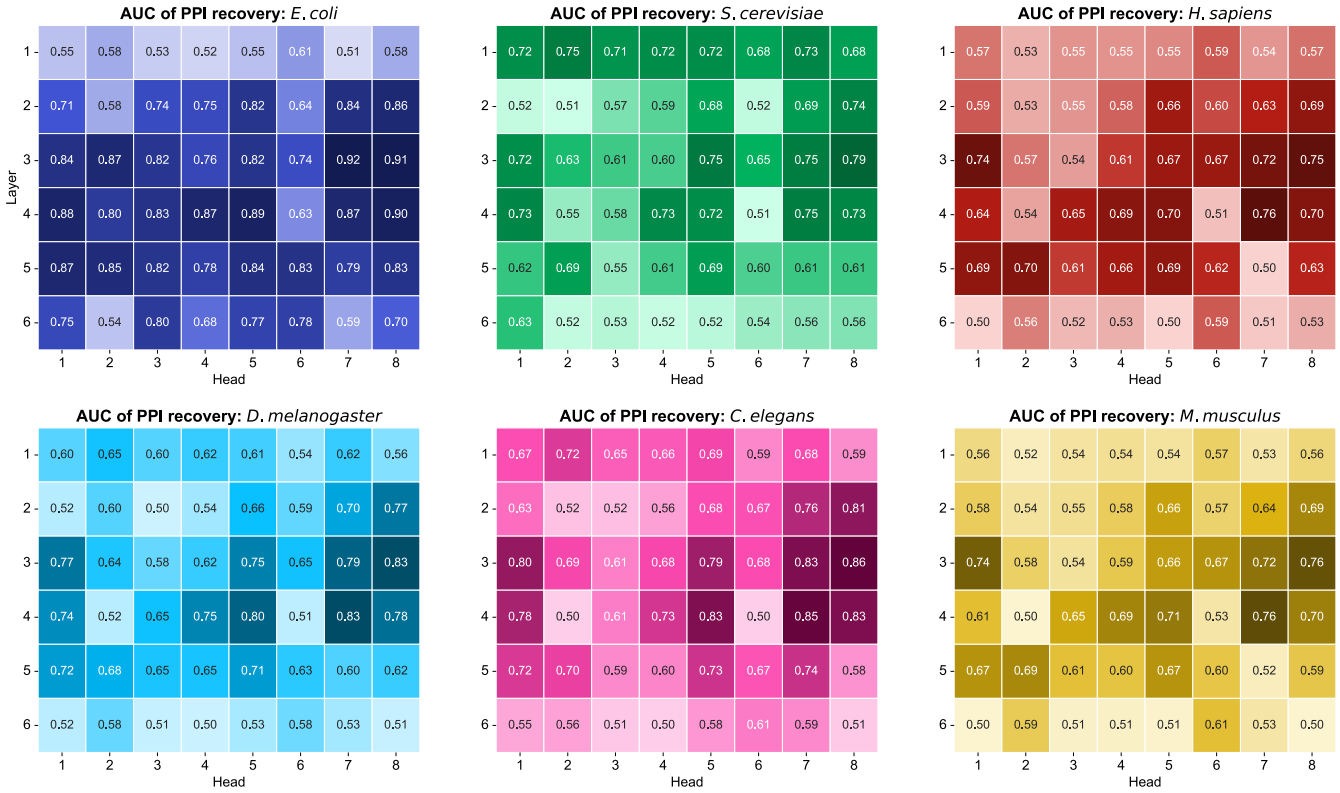

Figure S1: **Attention head-wise PPI recovery across species in ProteomeLM-S.** The area under the ROC curve (AUC) for protein-protein interaction (PPI) prediction using each individual attention head of ProteomeLM-S (6 layers, 8 heads) is reported across six species: *E. coli*, *S. cerevisiae*, *H. sapiens*, *D. melanogaster*, *C. elegans*, and *M. musculus*. Each heatmap shows the AUC for one species, with values computed separately for each attention head and layer. Attention heads from intermediate layers (particularly layer 3) consistently exhibit high predictive power, especially in *E. coli*. The first three panels (top) are also shown in Figure 2, and are reproduced here to facilitate inter-species comparison.

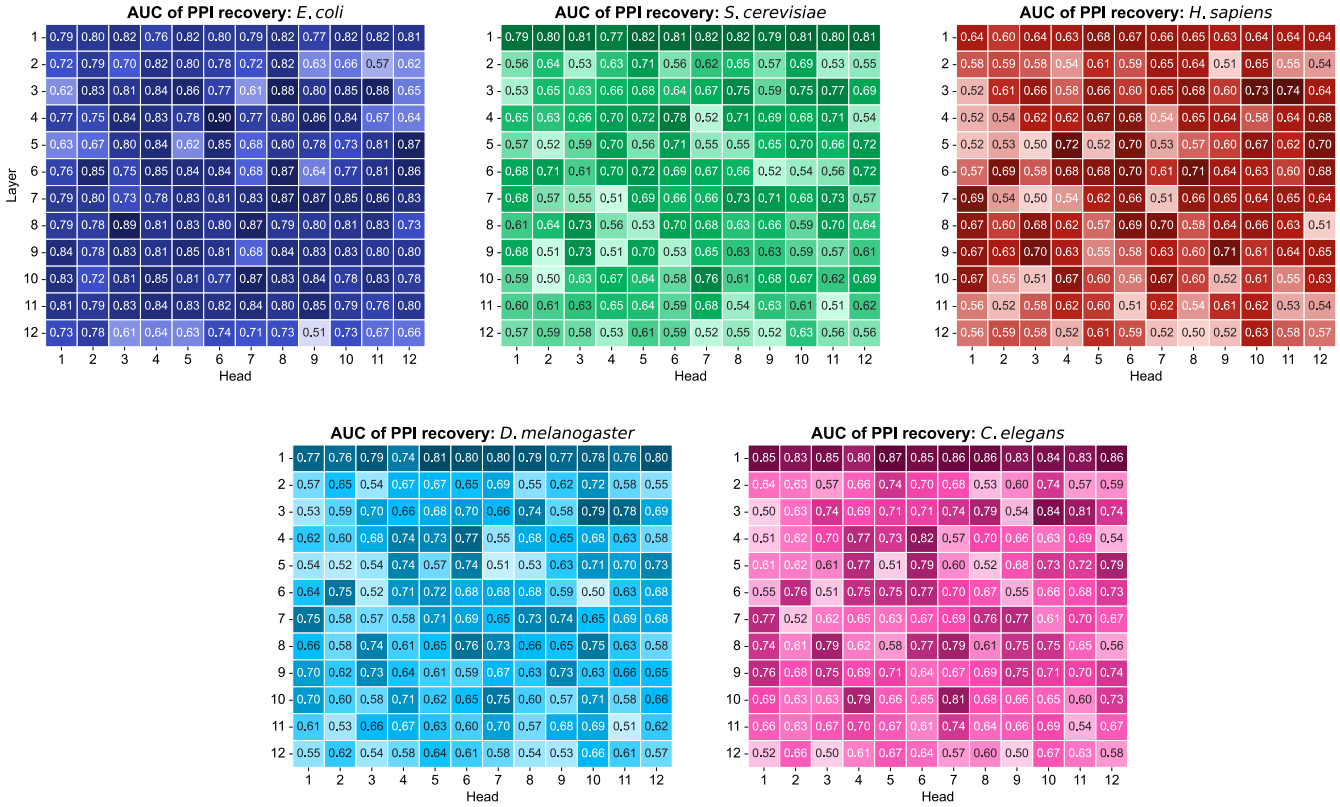

Figure S2: **Attention head-wise AUC of PPI recovery across species in ProteomeLM-M.** The AUC of protein-protein interaction (PPI) prediction is reported for each of the 12 attention head from all 12 layers of ProteomeLM-M (112M parameters), for five species: *E. coli*, *S. cerevisiae*, *H. sapiens*, *D. melanogaster*, and *C. elegans*. As in ProteomeLM-S (Figure S1), central layers tend to concentrate the most predictive heads. However, we note that the first layer is also a good predictor of PPI, in particular in eukaryotes. Note that the proteome of *M. musculus* is too long to allow the use of ProteomeLM-M.

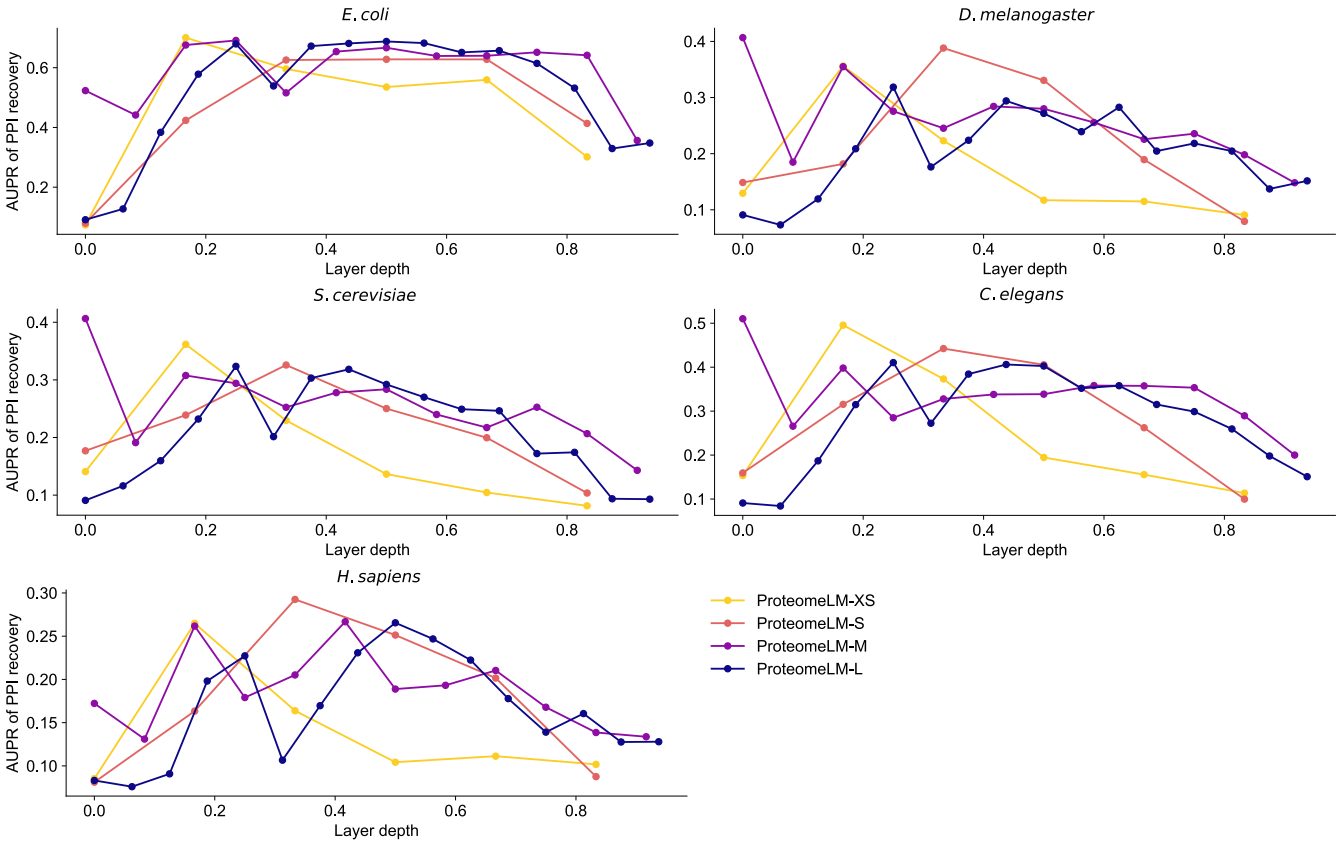

Figure S3: **Unsupervised PPI prediction by ProteomeLM across model sizes and layers.** The area under the precision-recall curve (AUPR) for unsupervised PPI prediction using summed attention scores is shown for each layer, across four model sizes (XS, S, M, L) and five species: *E. coli*, *S. cerevisiae*, *H. sapiens*, *D. melanogaster*, and *C. elegans*. To facilitate comparison between models with different sizes, the AUPR is plotted versus the normalized layer depth. Performance peaks at intermediate layers across species and model sizes.

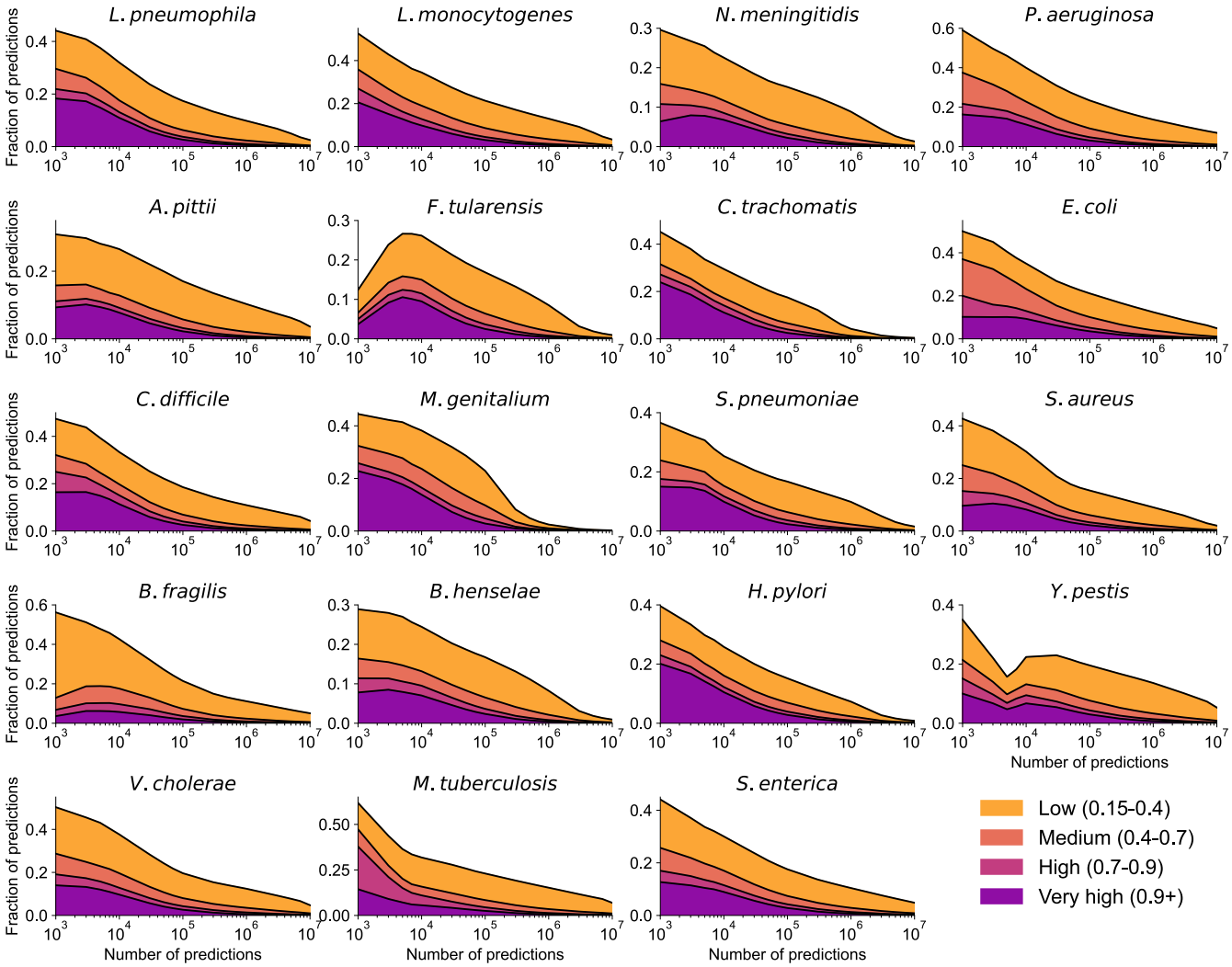

Figure S4: **Accuracy of fast PPI screening by ProteomeLM across 19 pathogens.** The fraction of top-scoring predicted PPI that correspond to known interactions in the STRING database is shown across 19 human bacterial pathogens. Predictions are sorted by the ProteomeLM classifier score, and their cumulative fraction in each of four bins of STRING confidence score is plotted as a function of the number of top-ranked predictions considered. Across most pathogens, a substantial fraction of high-scoring predictions correspond to known or high-confidence interactions. The 19 pathogens considered are the same as in Figure 3, and aggregated results are shown in Figure 3D.

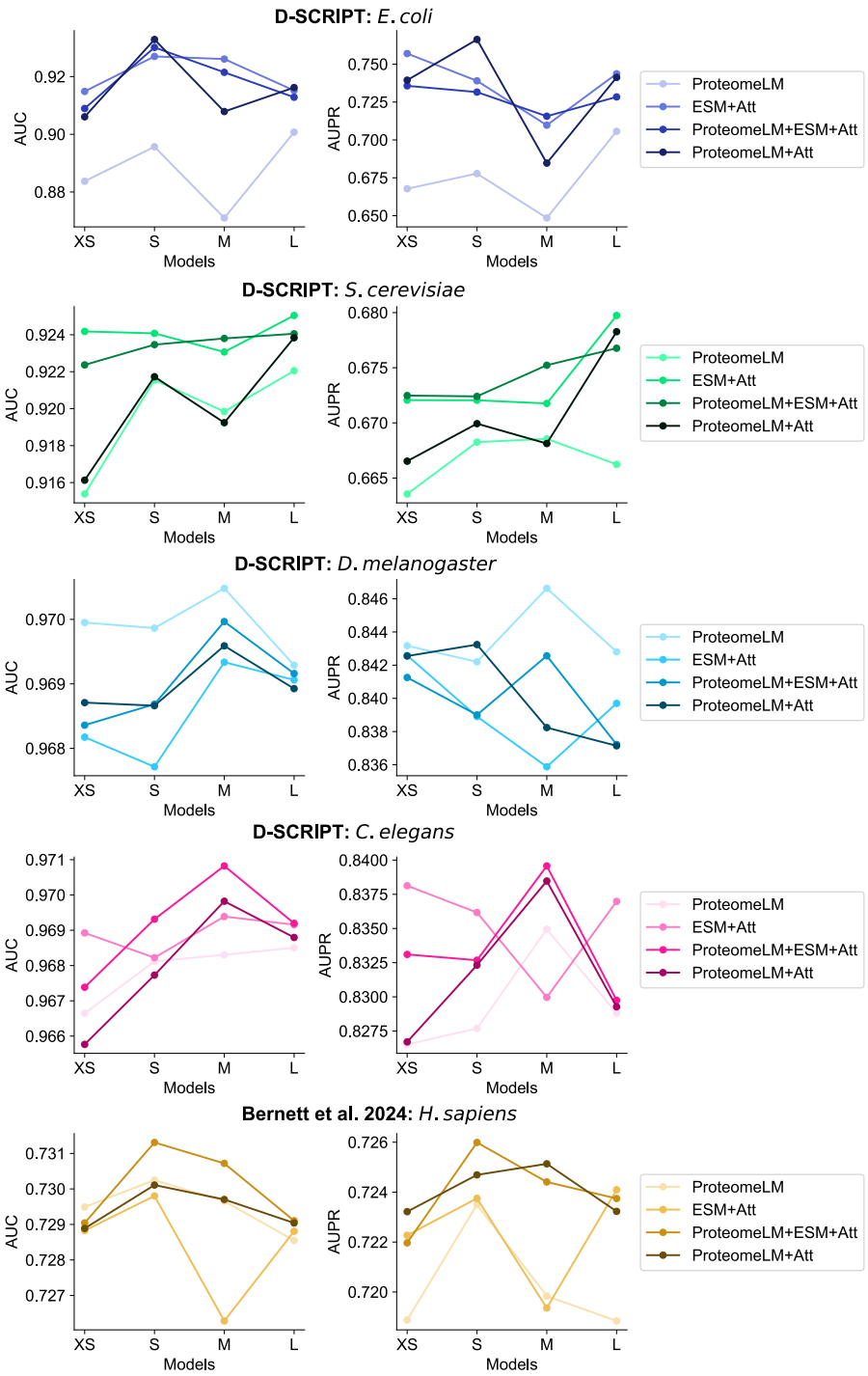

Figure S5: **Impact of embeddings and attention values on supervised PPI prediction.** Evaluation of supervised PPI prediction across four ProteomeLM model sizes (XS, S, M, L), on five benchmarks: the D-SCRIPT datasets for *E. coli*, *S. cerevisiae*, *D. melanogaster*, *C. elegans*, and the dataset from Ref. [64] for *H. sapiens*. For each of these benchmarks, both the AUC (left) and the AUPR (right) are reported using four different input feature configurations for PPI prediction: ProteomeLM embeddings only ("ProteomeLM"), ESM-C embeddings and ProteomeLM attention ("ESM+Att"), ESM-C embeddings, ProteomeLM embeddings and ProteomeLM attention ("ProteomeLM+ESM+Att"), and finally, ProteomeLM embeddings and ProteomeLM attention ("ProteomeLM+Att"). The latter is the configuration we retained throughout, and called ProteomeLM-PPI. Indeed, we observe here that combining ProteomeLM embeddings and ProteomeLM attention yields the best or near-best performance across species and metrics.

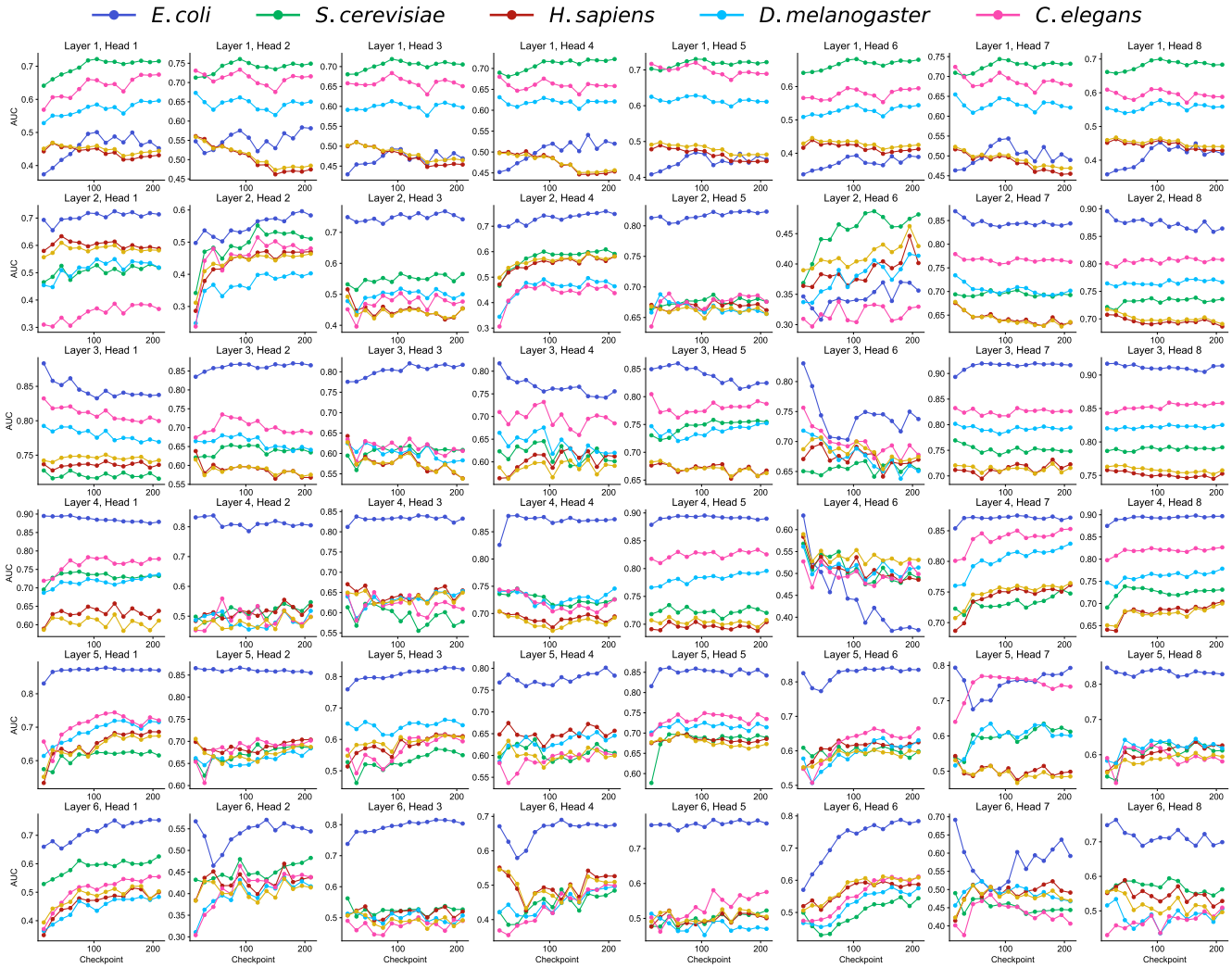

Figure S6: **Unsupervised recovery of PPI by ProtomeLM-S attention heads during training.** The area under the ROC curve (AUC) for unsupervised PPI recovery is shown versus training checkpoints for each attention head of ProteomeLM-S. Curves are shown separately for five species: *E. coli*, *S. cerevisiae*, *H. sapiens*, *D. melanogaster*, and *C. elegans*. Different heads exhibit different dynamics and specialization. Some become increasingly precise on *E. coli* (e.g. layer 6, head 3), while others (e.g. layer 1, head 1) are more specialized on eukaryotes. Many heads exhibit increasingly good PPI recovery along training.