ProteomeLM: A proteome-scale language model enables accurate and rapid prediction of protein-protein interactions

Cyril Malbranke
Institute of Bioengineering, School of Life Sciences
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
cyril.malbranke@epfl.ch

Gionata Paolo Zalaffi
Institute of Bioengineering, School of Life Sciences
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
gionata.zalaffi@epfl.ch

Anne-Florence Bitbol
Institute of Bioengineering, School of Life Sciences
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland
anne-florence.bitbol@epfl.ch

Abstract

Language models trained on biological sequences are advancing inference tasks from the scale of single proteins to that of genomic neighborhoods. Here, we introduce ProteomeLM, a transformer-based language model that uniquely operates on entire proteomes from species spanning the tree of life. ProteomeLM is trained to reconstruct masked protein embeddings using the whole proteomic context, yielding contextualized protein representations that reflect proteome-scale functional constraints. ProteomeLM spontaneously captures protein-protein interactions (PPI) in its attention coefficients. Furthermore, it enables interactome-wide PPI screening that is substantially more accurate, and orders of magnitude faster, than amino-acid coevolution-based methods. We further develop ProteomeLM-PPI, a supervised model that combines ProteomeLM embeddings and attention coefficients to achieve state-of-the-art PPI prediction across benchmarks and species. Our results demonstrate the potential of proteome-scale language models for addressing function and interactions at the organism level. Data and code are made (anonymously) available at https://anonymous.4open.science/r/ProteomeLM-anonymized.

1 Introduction

Recently, deep learning approaches have brought important progress to inference from biological sequence data. Protein language models trained on large ensembles of protein sequences learn sequence representations that encode structural and functional signals [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], and have advanced the prediction of protein structure [9, 10], subcellular localization [13], and mutational effects [14, 15]. Similarly, genome language models [16, 17, 18, 19, 20, 21, 22, 23, 24, 25] have given insight in non-coding DNA, gene expression and taxonomic classification [17, 18, 19], capturing operons and enzymatic function [21], and predicting mutation effects [20]. However, so far,

Machine Learning for Structural Biology Workshop

these models span at most hundreds of kilobases or a few megabases [18, 22, 19, 23, 24, 25]. As they do not capture dependencies across entire genomes, especially in eukaryotes, these models 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. However, large-scale PPI determination remains a significant challenge [26]. Precise experimental methods are both labor-intensive and costly, particularly when scaled to entire proteomes, and high-throughput ones have limited accuracy [27]. While curated PPI databases have grown [28, 29, 30], they remain incomplete and biased toward well-studied species. Computational methods include structure-based approaches like AlphaFold-Multimer [31], which are computationally intensive, and sequence-based methods like direct coupling analysis (DCA) [32, 33, 34, 35, 36, 37, 38], which struggle in eukaryotes or poorly sampled taxa and require careful curation of orthologs [35, 36, 39, 40].

Given the success of protein language models at capturing coevolution between amino acids [3, 9, 11, 10, 12], it is tantalizing to develop such models at the proteome scale. We posit that such models should capture coevolution between proteins, thereby generalizing over phylogenetic profiling methods [41, 42, 43, 44, 45, 46, 47, 48, 49], making them highly suited to provide predictions of complete protein-protein interaction networks with small computational cost once trained.

In this paper, we introduce ProteomeLM, a transformer-based language model that uniquely reasons on entire proteomes from multiple species spanning the tree of life. ProteomeLM leverages embeddings from ESM-Cambrian [12] and learns to reconstruct masked protein embeddings using proteome context. We show that ProteomeLM's attention coefficients learn PPI in an unsupervised way, enable interactome screening orders of magnitude faster than DCA with substantially better performance, and support state-of-the-art supervised PPI prediction across species and benchmarks.

2 Methods

ProteomeLM is a transformer-based language model trained on 32,000 proteomes spanning all domains of life. Each protein is represented by an embedding from ESM-Cambrian (ESM-C) [12], allowing our model to leverage rich functional sequence-derived properties [10, 3, 14, 9]. During training, a subset of protein embeddings is masked, and the model reconstructs them using remaining unmasked embeddings from the same proteome (Figure 1). This masked language modeling task allows ProteomeLM to learn dependencies between proteins.

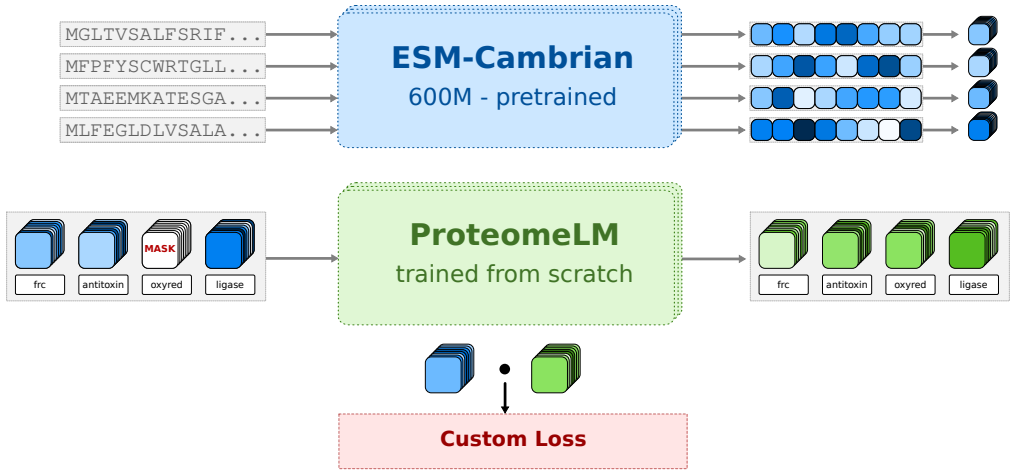

Figure 1: **ProteomeLM training.** Input amino-acid sequences are embedded through ESM-C, yielding fixed-dimensional embeddings. ProteomeLM predicts masked protein embeddings using proteome context. Proteins are annotated by orthologous groups, providing functional encoding. Training uses a custom polar loss that minimizes differences between ESM-C and ProteomeLM embeddings in a protein family-specific manner.

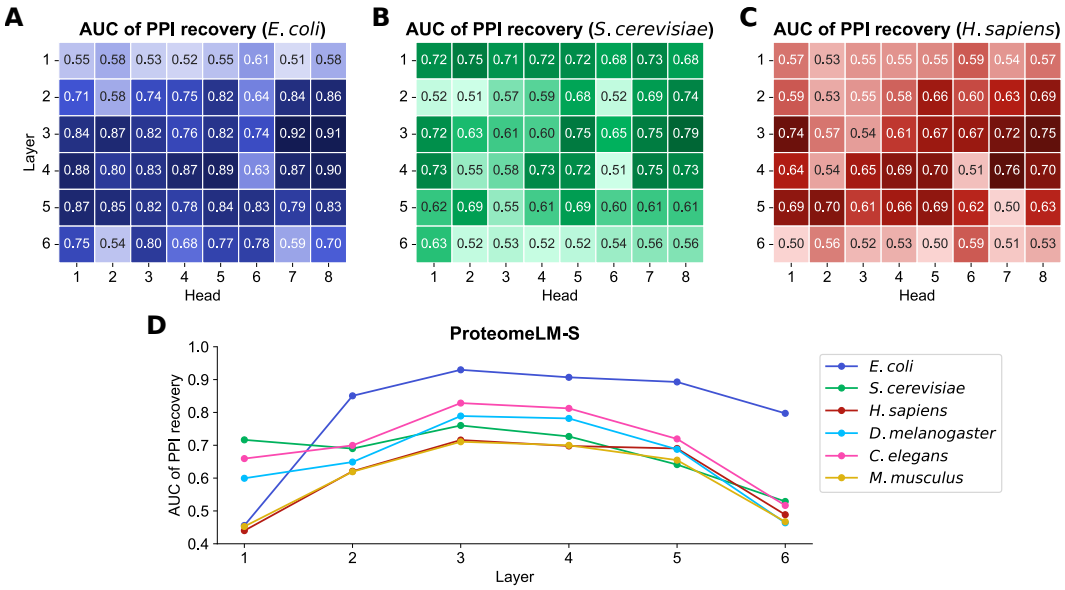

Figure 2: **Unsupervised detection of PPI using ProteomeLM attention coefficients.** We assess the ability of the attention coefficients of ProteomeLM-S (36M parameters) at predicting PPI from the D-SCRIPT dataset [26], using the Area Under the Receiver Operating Characteristic Curve (AUC) as a metric. **(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 six species considered.

A key innovation is our functional encoding based on orthology rather than genomic position. Since genome organization varies dramatically across species (bacterial operons vs. dispersed eukaryotic genes), ProteomeLM uses orthologous groups from OrthoDB [50] to capture shared evolutionary and functional relationships. Phylogenetic profiling methods have shown that orthologous group presence-absence patterns contain information about functional relationships [41, 42, 43, 51, 46], ProteomeLM improves over these methods. We trained models ranging from 6M to 328M parameters for 72h on single H100 GPUs. More information about the methods have been made available in supplementary section A.

3 Results

3.1 ProteomeLM Attention Captures Protein-Protein Interactions

We examined whether ProteomeLM spontaneously learns PPI through its attention coefficients by comparing them to known interactions from the D-SCRIPT dataset [26] across six species. Figure 2 shows that many attention heads achieve strong predictive power, with head 7 of layer 3 reaching AUC of 0.92 in *E. coli* while performing well across eukaryotic species. PPI are most accurately captured by intermediate layers, suggesting that higher-order interactions are essential for understanding PPI.

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.

3.2 Fast and Accurate Interactome Screening

Current interactome prediction workflows use DCA as a first filter, requiring training separate models for each candidate protein pair. We trained a lightweight classifier on ProteomeLM attention

coefficients and compared performance to recent large-scale DCA studies on human and pathogen proteomes [52, 53].

ProteomeLM inference takes under 10 minutes per proteome on a single GPU, compared to 30+ days on 50-100 GPUs for DCA on the human proteome, representing up to 6 orders of magnitude speedup (Figure 3**A**). ProteomeLM significantly outperforms DCA in recovering experimentally validated interactions, achieving AUROC of 0.83 vs 0.73 for DCA in humans, and recovering 50% vs 20% of known PPI among top 10 million scored pairs (Figure 3**B**). We further examine the overlap between ProteomeLM's top predictions and STRING [54] 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. We extended our analysis to 19 human bacterial pathogens [53]. Figure 3**D** shows that more than 40% of the top 10,000 predictions are supported by STRING with consistent results across species (Figure 3**E**).

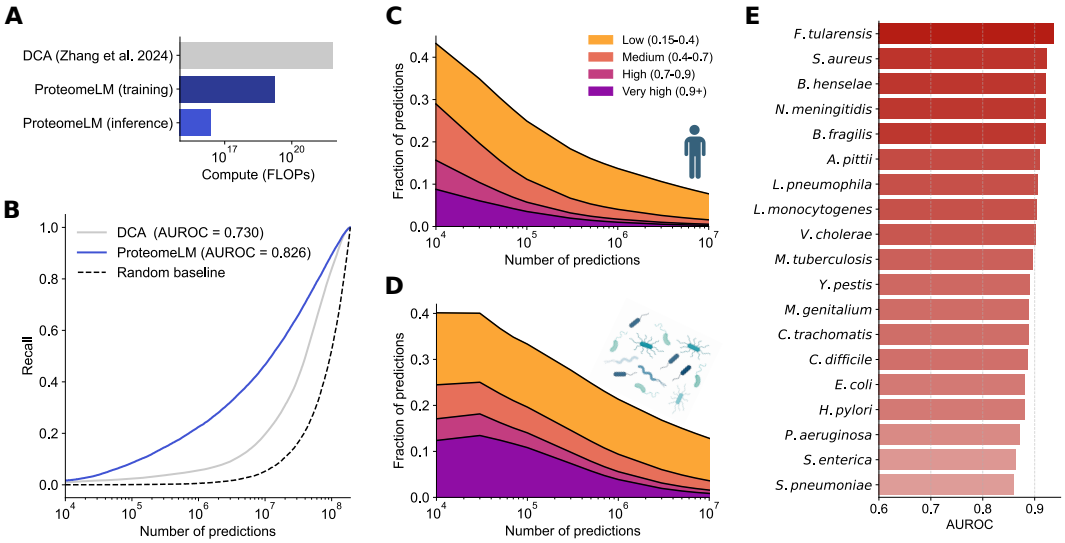

Figure 3: **Fast and high-precision screening using ProteomeLM. (A)** Compute comparison for human proteome analysis. **(B)** Human interactome recovery performance. **(C-D)** Fraction of top predictions corresponding to STRING database interactions. **(E)** Performance across 19 pathogenic bacterial species.

3.3 State-of-the-Art Supervised PPI Prediction

ProteomeLM-PPI combines node features (ESM-C and ProteomeLM embeddings) with edge features (attention coefficients) for supervised PPI prediction. We evaluated on two datasets: the multi-species D-SCRIPT dataset and a human-specific dataset addressing benchmark biases [55].

Figure 4**B** shows ProteomeLM-PPI outperforms state-of-the-art methods on *E. coli* and *S. cerevisiae*, with AUPR improvement of more than 0.1 (from 0.67 to 0.79) over TUnA [56] on *E. coli*. Performance remains strong across diverse species, demonstrating robust generalization. Figure4**C** also shows that ProteomeLM is better than previous approach on the Bernett et al. [55] dataset built to prevent data leakage between training, validation and testing set.

3.4 Gene Essentiality Prediction

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 [57]. We therefore finetuned ProteomeLM to predict the essentiality. To train and test this classification model, we used the OGEE database [58], which collects gene essentiality data from 127 experimental studies. We tested the ability of each layer of the four models to predict gene essentiality.

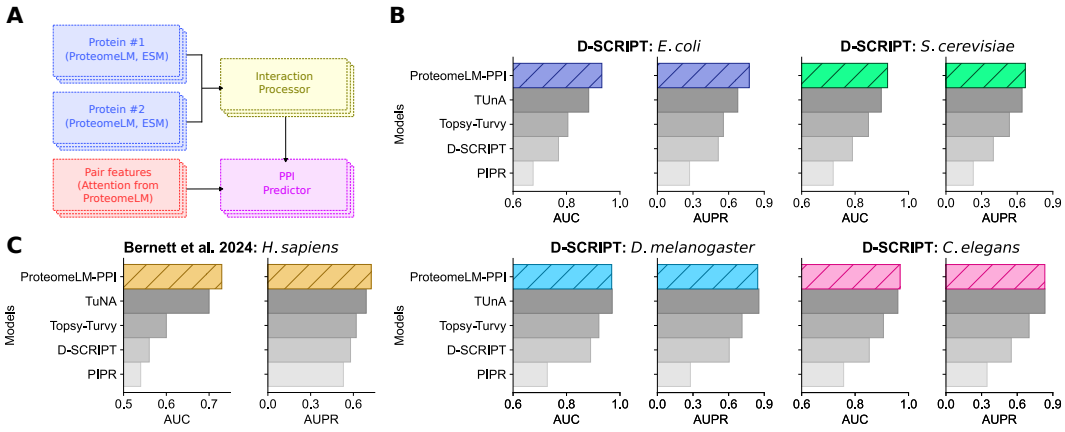

Figure 4: **Supervised PPI prediction.** **(A)** Architecture combining node and edge features. **(B)** Cross-species generalization on D-SCRIPT dataset. **(C)** Performance on bias-corrected dataset.

ProteomeLM-Ess significantly outperforms classifiers based on ESM-C embeddings alone for gene essentiality prediction (Figure 5**A**), demonstrating that contextualized proteome-aware information better captures essentiality than protein-level information. The best performing version achieves AUC of 0.93. ProteomeLM-Ess generalizes well to held-out proteomes including synthetic minimal cells JCVI-Syn1.0 and JCVI-Syn3A, correctly predicting 71% of essential genes in *E. coli*.

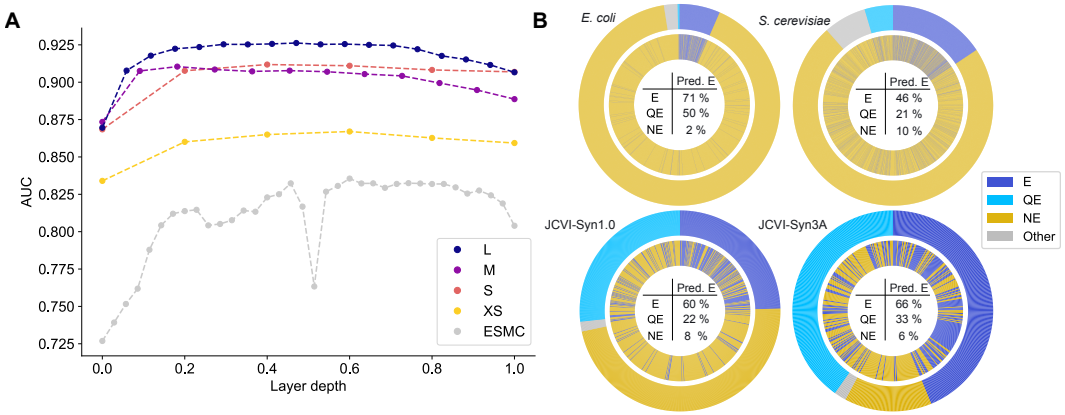

Figure 5: **Gene essentiality prediction.** **(A)** Performance vs layer depth for ProteomeLM-Ess and ESM-C baselines. **(B)** Comparison with experimental labels for four organisms.

4 Discussion

We introduced ProteomeLM, a transformer-based language model that learns contextualized protein representations from complete proteomes spanning the tree of life. ProteomeLM spontaneously captures PPI in its attention coefficients and enables highly scalable interactome screening with substantially higher accuracy than DCA and computational cost reduced by up to 6 orders of magnitude. Combined with supervised approaches, ProteomeLM achieves state-of-the-art performance on PPI prediction and gene essentiality tasks.

ProteomeLM represents proteins at a coarse-grained level using global ESM-C embeddings, enabling work with full proteomes while maintaining reasonable context sizes. Future work could leverage longer-context models to operate directly at amino-acid level or incorporate structural information from multimodal protein language models.

A key innovation is our functional encoding based on orthology rather than genomic position, allowing ProteomeLM to reason across diverse species with different genome organizations. While

performance remains stronger on prokaryotes than eukaryotes, likely due to training data composition, ProteomeLM demonstrates the power of proteome-scale language models for capturing emergent biological properties.

ProteomeLM opens applications including large-scale interactome prediction across species, evolutionary studies of protein networks, and context-aware fitness prediction. As a foundation model, ProteomeLM can be applied to diverse downstream tasks where proteome-level information matters, making it a valuable tool for systems biology and functional genomics.

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.

[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.

[4] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations*, 2021.

[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.

[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.

[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] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, 2022.

[9] 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.

[10] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Pranav Kantroo, Günter P. Wagner, and Benjamin B. Machta. Pseudo-perplexity in one fell swoop for protein fitness estimation. *arXiv*, page 2407.07265, 2024.

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

[17] 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.

[18] 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, Guillaume Richard, Marcin Skwark, Karim Beguir, Marie Lopez, and Thomas Pierrot. Nucleotide Transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

[19] Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton Rabideau, Yoshua Bengio, Stefano Ermon, Stephen A. Baccus, and Chris Ré. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. *Advances in neural information processing systems*, 36:43177–43201, 2023.

[20] 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.

[21] 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.

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

[23] Eric Nguyen, Michael Poli, Matthew G. Durrant, Brian Kang, Dhruva Katrekar, David B. Li, Liam J. Bartie, Armin W. Thomas, Samuel H. King, Garyk Brixi, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723):eado9336, 2024.

[24] Garyk Brixi, Matthew G. Durrant, Jerome Ku, Michael Poli, Greg Brockman, Daniel Chang, Gabriel A. Gonzalez, Samuel H. King, David B. Li, Aditi T. Merchant, Mohsen Naghipourfar, Eric Nguyen, Chiara Ricci-Tam, David W. Romero, Gwanggyu Sun, Ali Taghibakshi, Anton Vorontsov, Brandon Yang, Myra Deng, Liv Gorton, Nam Nguyen, Nicholas K. Wang, Etowah Adams, Stephen A. Baccus, Steven Dillmann, Stefano Ermon, Daniel Guo, Rajesh Ilango, Ken Janik, Amy X. Lu, Reshma Mehta, Mohammad R. K. Mofrad, Madelena Y. Ng, Jaspreet Pannu, Christopher Ré, Jonathan C. Schmok, John St John, Jeremy Sullivan, Kevin Zhu, Greg Zynda, Daniel Balsam, Patrick Collison, Anthony B. Costa, Tina Hernandez-Boussard, Eric Ho, Ming-Yu Liu, Thomas McGrath, Kimberly Powell, Dave P. Burke, Hani Goodarzi, Patrick D. Hsu, and Brian L. Hie. Genome modeling and design across all domains of life with Evo 2. *bioRxiv*, page 2025.02.18.638918, 2025.

[25] Žiga Avsec, Natasha Latysheva, Jun Cheng, Guido Novati, Kyle R. Taylor, Tom Ward, Clare Bycroft, Lauren Nicolaisen, Eirini Arvaniti, Joshua Pan, Raina Thomas, Vincent Dutordoir, Matteo Perino, Soham De, Alexander Karollus, Adam Gayoso, Toby Sargeant, Anne Mottram,

Lai Hong Wong, Pavol Drotár, Adam Kosiorek, Andrew Senior, Richard Tanburn, Taylor Applebaum, Souradeep Basu, Demis Hassabis, and Pushmeet Kohli. AlphaGenome: advancing regulatory variant effect prediction with a unified DNA sequence model. *bioRxiv*, page 2025.06.25.661532, 2025.

[26] 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.

[27] 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.

[28] Rose Oughtred, Jennifer Rust, Christie Chang, Bobby-Joe Breitkreutz, Chris Stark, Andrew Willems, Lorrie Boucher, Genie Leung, Nadine Kolas, Frederick Zhang, Sonam Dolma, Jasmin Coulombe-Huntington, Andrew Chatr-Aryamontri, Kara Dolinski, and Mike Tyers. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Science*, 30(1):187–200, 2021.

[29] Peter D Karp, Wai Kit Ong, Suzanne Paley, Richard Billington, Ron Caspi, Carol Fulcher, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Peter E Midford, Pallavi Subhraveti, Socorro Gama-Castro, Luis Muñiz-Rascado, César Bonavides-Martinez, Alberto Santos-Zavaleta, Amanda Mackie, Julio Collado-Vides, Ingrid M. Keseler, and Ian Paulsen. The EcoCyc database. *EcoSal Plus*, 8(1):10–1128, 2018.

[30] 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.

[31] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, page 2021.10.04.463034, 2021.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] Ian R. Humphreys, Jimin Pei, Minkyung Baek, Aditya Krishnakumar, Ivan Anishchenko, Sergey Ovchinnikov, Jing Zhang, Travis J. Ness, Sudeep Banjade, Saket R. Bagde, Viktoriya G. Stancheva, Xiao-Han Li, Kaixian Liu, Zhi Zheng, Daniel J. Barrero, Upasana Roy, Jochen

Kuper, Israel S. Fernández, Barnabas Szakal, Dana Branzei, Josep Rizo, Caroline Kisker, Eric C. Greene, Sue Biggins, Scott Keeney, Elizabeth A. Miller, J. Christopher Fromme, Tamara L. Hendrickson, Qian Cong, and David Baker. Computed structures of core eukaryotic protein complexes. *Science*, 374:1340, 2021.

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

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] David Moi and Christophe Dessimoz. Reconstructing protein interactions across time using phylogeny-aware graph neural networks. *bioRxiv*, page 2022.07.21.501014, 2022.

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

[50] 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.

[51] David Moi and Christophe Dessimoz. Phylogenetic profiling in eukaryotes comes of age. *Proc. Natl. Acad. Sci. USA*, 120(19):e2305013120, 2023.

[52] Jing Zhang, Ian R Humphreys, Jimin Pei, Jinuk Kim, Chulwon Choi, Rongqing Yuan, Jesse Durham, Siqi Liu, Hee-Jung Choi, Minkyung Baek, David Baker, and Qian Cong. Computing the human interactome. *bioRxiv*, page 2024.10.01.615885, 2024.

[53] Ian R. Humphreys, Jing Zhang, Minkyung Baek, Yaxi Wang, Aditya Krishnakumar, Jimin Pei, Ivan Anishchenko, Catherine A. Tower, Blake A. Jackson, Thulasi Warrier, Deborah T. Hung, S. Brook Peterson, Joseph D. Mougous, Qian Cong, and David Baker. Protein interactions in human pathogens revealed through deep learning. *Nature microbiology*, 9(10):2642–2652, 2024.

[54] 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.

[55] 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.

[56] 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.

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

[58] 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.

[59] 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, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

[64] 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.

[65] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 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.

[66] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv*, page 2307.08691, 2023.

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

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

[69] 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.

[70] 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.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

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

A Methods

A.1 Architecture and training of ProteomeLM

Dataset. We collected 31,947 proteomes from OrthoDB (version 12) [50]. 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) [59, 60], 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 [12] 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 [10]. Note that, contrary to ESM-3, which is multimodal [61], 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 [62], and also for protein sequences [9, 63, 12] and for genomic sequences [18, 64], 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 [65]. We used FlashAttention-2 [66] 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 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:

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

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 **??**, 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 [26] 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.

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.

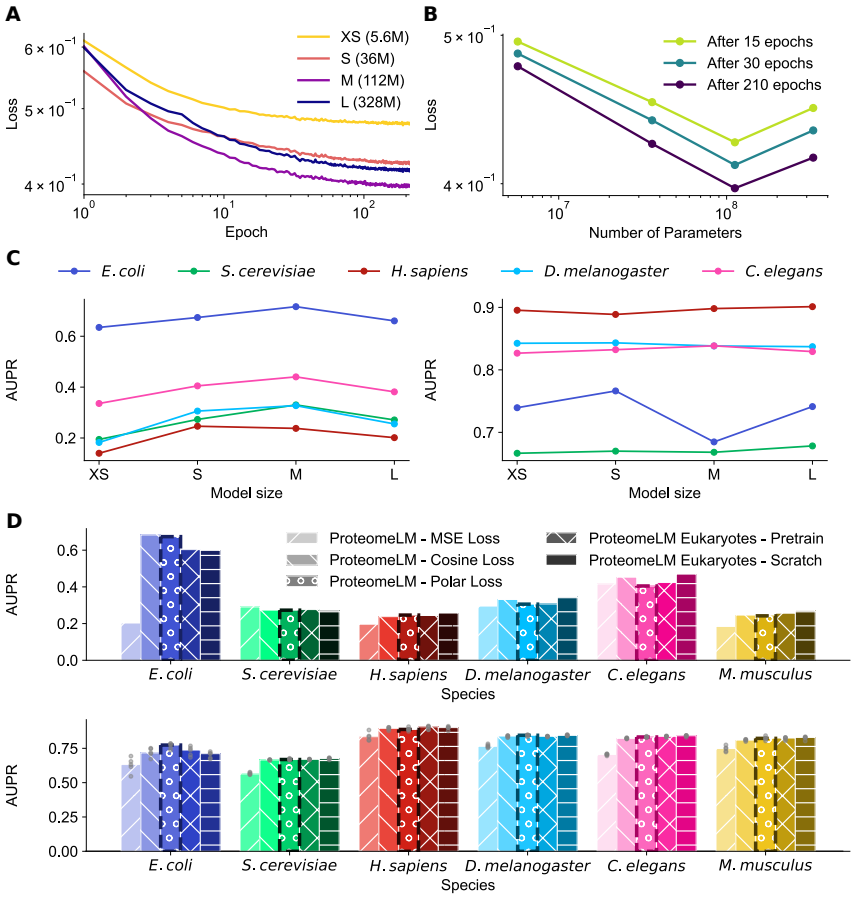

Figure 6: **Training dynamics, scaling behavior, and training strategies for ProteomeLM. (A)** Evaluation loss during training for four ProteomeLM model sizes: XS (5.6M parameters), S (36M), M (112M), and L (328M). **(B)** The final evaluation loss is shown for three different epochs. **(C)** The AUPR 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.

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.

A.2 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 [67]. 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 [55] and the D-SCRIPT dataset [26], clean and non-overlapping splits into training, validation, and test sets were already available, and we employed them.

A.3 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 [58]. 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 [68], NCBI [69], *Saccharomyces* Genome Database (SGD) [70] and Fitness Browser [71], 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 [72] and JCVI-Syn3A [73, 74], 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 [75] 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).