Structure, Surface and Interface Informed Protein Language Model

Ioan Ieremie
Electronics and Computer Science
University of Southampton
ii1g17@soton.ac.uk

Niranjan Mahesan
Electronics and Computer Science
University of Southampton
mn@ecs.soton.ac.uk

Rob M. Ewing
Biological Sciences
University of Southampton
rob.ewing@soton.ac.uk

Abstract

Language models applied to protein sequence data have gained a lot of interest in recent years, mainly due to their ability to capture complex patterns at the protein sequence level. However, their understanding of why certain evolution-related conservation patterns appear is limited. This work explores the potential of protein language models to further incorporate intrinsic protein properties stemming from protein structures, surfaces, and interfaces. The results indicate that this multi-task pretraining allows the PLM to learn more meaningful representations by leveraging information obtained from different protein views. We evaluate and show improvements in performance on various downstream tasks, such as enzyme classification, remote homology detection, and protein engineering datasets. Trained models and code are available at github.com/Ieremie/general-protein-embeddings.

1 Introduction

With the constant growth of protein sequence data, sequence databases such as Uniref (Suzek et al., 2007) have grown to more than 200 million protein sequences. These massive datasets attracted the application of methods borrowed from Natural Language Processing to learn evolutionary information in an unsupervised way. However, protein language models are limited in their capacity to understand why specific parts of the protein sequence space exhibit higher levels of conservation compared to others. That being said, it is important to consider that conservation patterns observed in protein sequences can be explained by protein characteristics stemming from other views of the protein such as their structures, surfaces, and interactions. For instance, residues co-evolve at distant positions in the sequence to maintain the protein structure and function (Marks et al., 2012; Morcos et al., 2011), residues that are buried and more tightly packed evolve slower compared to residues that are loosely packed or residing at the molecular surface (Conant and Stadler, 2009; Franzosa and Xia, 2009; Bustamante et al., 2000; Bloom et al., 2006) and residues that are involved in protein-protein interfaces have a higher degree of conservation (Yang et al., 2012; Mintseris and Weng, 2005).

We explore the ability of a language model to incorporate further protein knowledge by constraining the generated embeddings to be good predictors of intrinsic protein properties. We hypothesize that a PLM that is trained simultaneously on multiple self-supervising tasks could learn conservation patterns that are not immediately observable at the sequence level. Therefore, embeddings learned by the PLM would be more general and fit for a wider variety of downstream tasks. To our knowledge, our work is the first to directly pretrain a PLM with structure, surface, and interface information.

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

2 Related work

Sequence-based methods. Initial work (Alley et al., 2019; Rao et al., 2019; Bepler and Berger, 2019) trained small LSTM-based language models on these datasets and showed how learned embeddings can compete on downstream tasks with classic protein profiles. The inevitable scaling to more weights ($\approx 650M$) and the introduction of attention-based architectures (Rives et al., 2021; Elnaggar et al., 2021; Rao et al., 2021) introduced new state-of-the-art models and various protein prediction tasks such as secondary structure prediction and contact map prediction. Current models (Lin et al., 2022; Chowdhury et al., 2022) have billions of weights and allow the prediction of protein structures using nothing but the protein sequence as input.

Surface-based methods. Gainza et al. (2020) has introduced a unique approach employing geodesic convolutions on molecular surfaces to capture interaction fingerprints. In contrast, other techniques, as demonstrated by Somnath et al. (2021), learn from protein surfaces by incorporating surface meshes into the protein structure graph.

Structure-base methods. Approaches to learning from protein structures usually employ 3D CNNSs (Townshend et al., 2019), graph neural networks (Somnath et al., 2021; Hermosilla et al., 2020; Jing et al., 2020) and continuous CNNs (Fan et al., 2022). Of particular interest are methods that perform self-supervised learning using information stemming from protein structures (Zhang et al., 2022; Wang et al., 2022).

3 Learning general protein embeddings

3.1 Protein Language Model

We employ a three-layer Bidirectional LSTM architecture similar to previous work (Bepler and Berger, 2021; Alley et al., 2019) with skip connections from each layer as a model for pretraining.

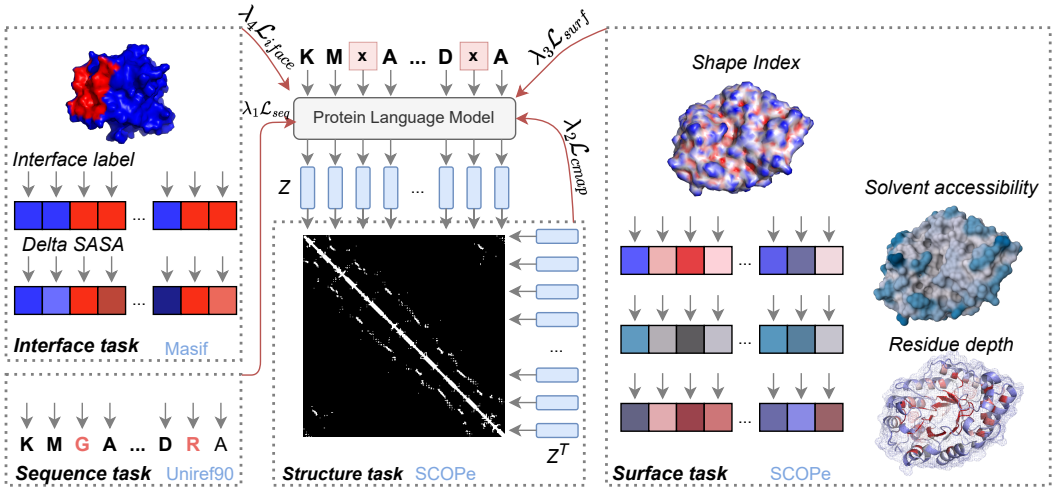


Figure 1: The training overview of the protein language model. The PLM is trained to recover masked residues as in the classical setting. This model is then further pretrained using information stemming from protein structures, surfaces, and protein interactions. The datasets used for pretraining are highlighted in blue.

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L]$ be the input protein sequence, where $\mathbf{x}_i$ is the one-hot encoded vector representation of the i-th amino acid in the sequence. First, the input sequence $\mathbf{X}$ is passed through a learned embedding layer, which maps each one-hot encoded vector $\mathbf{x}_i$ to a continuous embedding space $\mathbf{e}_i$. Next, the embedded sequence E is fed into the BiLSTM layers. To incorporate skip connections, the LSTM hidden states from each layer in the architecture are concatenated, resulting in the final hidden representation $\mathbf{z}_i$ at each position. The concatenation operation can be represented as follows: $\mathbf{z}_i = [\mathbf{e}_i; \mathbf{h}_i^{(1)}; \mathbf{h}_i^{(2)}; \mathbf{h}_i^{(3)}]$, where $(;)$ denotes the concatenation of vectors and $\mathbf{h}_i^{(l)}$ is the hidden representation at each layer.

The PLM is trained using masked token prediction on Uniref90 (Suzek et al., 2007) (July 2018). During training, we mask residues with a 10% probability. Instead of using a designated mask token, we replace the original residues with other amino acids drawn from a background distribution calculated using Uniref90. The model is trained to recover these residues using the cross-entropy loss: $\mathcal{L}_{masked} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{21} y_{ij}\log(p_{ij})$. In comparison to state-of-the-art protein language models like those described in (Lin et al., 2022), which have billions of parameters, our model is relatively lightweight with only 14M parameters. This allows us to train multiple models and analyse the importance of each component.

3.2 Structure head

For the structural task, we aim to predict the inter-residue contact map of a protein sequence. To achieve this, we utilize a learned weight matrix W and a bias term b. Given the amino acid embeddings $Z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L]$, where $\mathbf{z}_i$ represents the embedding vector of the i-th amino acid in the sequence, we compute the inter-residue contact map as $ZWZ^T + \mathbf{b}$. This operation results in a matrix of size $L \times L$, where L denotes the length of the protein sequence.

Each entry in the resulting matrix represents the probability that the residues are no further apart than 8 Å in the 3D protein structure. The weights are updated using the cross-entropy loss, which we name $\mathcal{L}_{cmap}$.

3.3 Surface head

To incorporate surface information into our model, we introduce an additional component that predicts the surface properties of the residues. This component consists of three linear layers, which are used to predict each residue's solvent-accessible surface area (SASA), depth, and shape index. To capture the combined effect of these surface predictions, we introduce an additive surface loss, denoted as $\mathcal{L}_{surf}$ which uses the MSE loss for the distance and shape prediction and MAE loss for the SASA prediction.

SASA values are computed using the Shake-Rupley algorithm (Shrake and Rupley, 1973) and it represents the atomic surface in contact with a probe of radius *R* that rolls on the outside of the molecule by maintaining contact with the atom's van der Walls spheres. For each protein domain, we generate its surface using a similar approach to previous work (Gainza et al., 2020). We then compute the residue depth by averaging the distance of each residue's atoms to the closest vertex on the protein mesh. Let R be the residue, and V the set of all vertexes on the protein surface. The residue depth can be expressed as $R_d = \frac{1}{n}\sum_{j=1}^{n} \min_{v \in V} ||\mathbf{a}_j, \mathbf{v}||$ where $||.||$ represents the Euclidean distance between two points. Residue depth offers a descriptive view of the degree of residue burial. It provides complementary information to SASA, especially for residues with similar SASA values that have different degrees of burial (Chakravarty and Varadarajan, 1999).

To account for the shape of the protein surface at different locations, we compute the Shape index (Koenderink and Van Doorn, 1992) for each vertex in the protein mesh. This is of interest due to the importance of protein clefts in determining enzyme active sites (Laskowski et al., 1996). For each amino acid, we assign a shape index value based on the average of the 3 closest vertices to the residue (see Appendix A).

The pretraining dataset for both the surface and structure tasks is SCOPe v2.06 (Chandonia et al., 2019). We use the training split provided by Bepler and Berger (2021), with the training dataset having 22408 protein structures.

3.4 Interface head

For the interface task, we use the interface prediction dataset curated by Gainza et al. (2020) which contains 3003 protein structures composed of one or multiple chains. The protein complex surface generation step is different from Gainza et al. (2020) to remove interface overestimation (see Appendix B). We compute the interface label at the sequence level based on the solvent accessible surface area lost upon complex formation with the interacting partner. A residue is considered part of the interface if there is at least a 4% change and 5Å^2 difference in SASA upon complex formation(Porollo and Meller, 2007). We use three linear layers to predict if a residue is part of the interface, the delta

SASA, and the distance to the closest part of the surface labelled as interface. We include the last predictor to account for residues that are not part of the interface but could influence the interface structure.

3.5 Training details

Prediction heads for the second stage of pretraining are kept as simple as possible. We only use linear layers with a single hidden layer to arrive at the prediction of interest and train the model end-to-end. The intuition behind this design is to allow the modelling of structure, surface, and interface information to be handled by the protein language model rather than a complex architecture trained on top of the embeddings.

We followed a two-stage pretraining approach, where we first trained the language model and then further pre-trained with additional tasks. This is to ensure that the sequence information learned is not lost upon changing the training context. For the initial language model training (referred to as $\text{ProtEMB}_{\text{LM}}$), we employed the masked loss over a total of 240k training steps with an accumulative batch size of 1024 protein sequences. To accommodate memory constraints, protein sequences were randomly cropped to a maximum length of 500. This stage of pretraining took approximately 6 days, with an equivalent of ≈ 3.1 passes through Uniref90.

In the second stage of pretraining, we introduced additional tasks to the language model and conducted training with a multitask objective. For the contact map prediction task ($\text{ProtEMB}_{\text{LM+CMAP}}$), we extended training for an additional 240k steps. To guide the training process, we employed weight parameters: $\lambda_{masked} = 0.4$ and $\lambda_{cmap} = 0.6$. The batch size for the language task was reduced to 256, while a batch size of 64 was utilized for the contact map prediction. Similarly, for the language and surface task ($\text{ProtEMB}_{\text{+SURF}}$), we assigned weights $\lambda_{masked} = 0.4$ and $\lambda_{surf} = 0.6$. The surface weight was evenly distributed among the various surface losses. In the case of the interface task, we carried out pretraining of the language model for 150k steps, with weight values $\lambda_{masked} = 0.4$ and $\lambda_{iface} = 0.6$. The iface weight was distributed as 0.4 towards the iface label prediction and 0.2 towards distance and delta SASA prediction.

Finally, to combine the information from all tasks we further pretrain two language models, $\text{ProtEMB}_{\text{+CMAP+SURF}}$ and $\text{ProtEMB}_{\text{+CMAP+SURF+IFACE}}$ with $\lambda_{masked} = 0.2$ and the remaining weight being equally distributed among tasks.

4 Evaluation and results

We evaluate the ability of PLM to learn meaningful protein representations on a set of downstream tasks: enzyme classification (Hermosilla et al., 2020), remote homology detection (Rao et al., 2019) and three protein engineering datasets Dallago et al. (2021). For all the downstream tasks we fine-tune the language model along with an attention layer combined with an MLP that has a single hidden layer (Appendix E).

In Table 1, the base model trained only on the language task obtains performances that are comparable to much larger models, such as ESM-1b (45x larger). The performance difference becomes more apparent in the Superfamily and Family classification where large language models can generalise better. However, when contact map prediction is introduced as an additional task during pretraining, the homology task shows a significant improvement. Similarly, the PLM further trained with surface information improves performance on the Enzyme task, surpassing HoloProt(Somnath et al., 2021) which uses surface information as input. When both contact map prediction and surface tasks are introduced during the pretraining stage, the PLM shows better improvements across downstream tasks. This suggests that the model retains and combines information learned from protein properties during pretraining. Conversely, constraining the learned embeddings to be good predictors of protein interface regions shows a degradation in performance. This could be attributed to the relatively small dataset which leads to overfitting (see Appendix D). While pretraining a PLM with further data coming from other protein properties improves performance, it does not match the results obtained by models working directly on 3D structures. The incorporation of more structural information, such as torsion angles and residue distograms (Zhang et al., 2022), could change this performance gap. The performance enhancement is not solely a result of extended training on the language task, as detailed in Appendix F.

Table 1: PLM evaluation on homology detection and enzyme reaction classification. Results marked with (*) are from (Hermosilla et al., 2020) and (†) from (Zhang et al., 2022). Standard deviations stem from 4 different runs for REACT and 6 for FOLD using slightly different learning rates. We report the percentage classification accuracy.

	Architecture	FOLD%			REACT %
		Fold	Super.	Fam.	
Sequence-based					
Rao et al. (2019)*	LSTM (43 M)	26.0	43.0	92.0	79.9
Rives et al. (2021)†	Transf. (650 M)	26.8	**60.1**	**97.8**	83.1
$\text{ProtEMB}_{\text{LM}}$	LSTM (14 M)	26.3 ± 0.96	43.3 ± 0.41	90.7 ± 0.44	81.8 ± 0.39
$\text{ProtEMB}_{\text{+CMAP}}$	-	32.7 ± 1.07	51.6 ± 0.47	94.8 ± 0.35	82.9 ± 1.19
$\text{ProtEMB}_{\text{+SURF}}$	-	29.3 ± 1.00	47.5 ± 0.85	92.9 ± 0.55	**83.3** ± 1.48
$\text{ProtEMB}_{\text{+IFACE}}$	-	25.2 ± 0.66	40.0 ± 1.03	84.1 ± 2.90	79.2 ± 1.46
$\text{ProtEMB}_{\text{+CMAP+SURF}}$	-	**33.4** ± 0.69	53.4 ± 0.90	95.7 ± 0.66	83.3 ± 1.23
$\text{ProtEMB}_{\text{+CMAP+SURF+IFACE}}$	-	33.0 ± 0.53	49.7 ± 0.74	93.1 ± 0.55	81.0 ± 1.15
Structure-based					
Hermosilla et al. (2020)	CNN (9.8 M)	45.0	69.7	98.9	87.2
Somnath et al. (2021)	GNN (0.6 M)	-	-	-	78.9
Zhang et al. (2022)	GNN	**54.1**	**80.5**	**99.9**	**87.5**

Table 2: Performance (Spearman correlation) on the FLIP datasets. Uncertainties are standard deviations over 3 seeds. For the Meltome dataset, we only train a single model due to the computational cost.

Model	AAV			
	1-vs-many	2-vs-many	7-vs-many	low-vs-high
CARP-640M	**0.73** ± 0.05	**0.81** ± 0.03	**0.77**± 0.03	0.19 ± 0.008
PromptPROTEIN-640M	0.55	-	-	-
$\text{ProtEMB}_{\text{LM}}$	0.41 ± 0.08	0.48 ± 0	0.55 ± 0.06	0.20 ± 0.04
$\text{ProtEMB}_{\text{+CMAP}}$	0.44 ± 0.05	0.45 ± 0	0.62 ± 0.03	0.23 ± 0.01
$\text{ProtEMB}_{\text{+SURF}}$	0.48 ± 0.01	0.54 ± 0.09	0.58 ± 0.02	0.20 ± 0.02
$\text{ProtEMB}_{\text{+IFACE}}$	0.47 ± 0.02	0.49 ± 0.05	0.61 ± 0.03	0.19 ± 0.07
$\text{ProtEMB}_{\text{+CMAP+SURF}}$	0.41 ± 0.06	0.48 ± 0.05	0.58 ± 0.06	0.19 ± 0.02
$\text{ProtEMB}_{\text{+CMAP+SURF+IFACE}}$	0.48 ± 0.02	0.46 ± 0.05	0.62 ± 0.02	**0.23** ± 0.02
	GB1			Meltome
	2-vs-many	3-vs-many	low-vs-high	mixed-split
CARP-640M	**0.73** ± 0.03	**0.87** ± 0	0.43 ± 0.04	0.53
PromptPROTEIN-640M	0.55	0.78	-	**0.69**
$\text{ProtEMB}_{\text{LM}}$	0.64 ± 0.03	0.82 ± 0	0.39 ± 0.09	0.28
$\text{ProtEMB}_{\text{+CMAP}}$	0.67 ± 0.02	0.81 ± 0	0.45 ± 0.02	0.27
$\text{ProtEMB}_{\text{+SURF}}$	0.63 ± 0.03	0.82 ± 0	**0.50** ± 0.05	0.28
$\text{ProtEMB}_{\text{+IFACE}}$	0.61 ± 0.07	0.82 ± 0	0.41 ± 0.03	0.25
$\text{ProtEMB}_{\text{+CMAP+SURF}}$	0.65 ± 0.01	0.82 ± 0	0.40 ± 0.09	0.29
$\text{ProtEMB}_{\text{+CMAP+SURF+IFACE}}$	0.65 ± 0.03	0.82 ± 0	0.44 ± 0.05	0.29

The second pretraining stage also yields performance improvements on the protein engineering datasets (Table 2). PLMs pretrained with structural and surface information exhibit similar performances with larger models (Yang et al., 2022). Additionally, these pretrained models show better results across various splits when compared to the base PLM. They perform particularly well on splits containing sequences with low fitness in the training set and high fitness sequences in the test set (low-vs-high).

By integrating information from various protein properties, the protein language model enhances its understanding of the connections between protein sequences, structures, and interactions. Through empirical evidence, we demonstrate improvements in diverse downstream tasks ranging from understanding enzyme activity to mutational landscapes. The PLM is not confined to capturing only

sequence conservation patterns; it can also deduce the factors that induce conservation. Moreover, these models facilitate the generation of more general embeddings, effectively distinguishing between protein structural classes and families (refer to Appendix C).

5 Conclusion

PLMs trained on large datasets of protein sequences learn evolution patterns that are visible at the sequence level. However, having a deep understanding of why certain conservation patterns appear requires looking at protein structures and their interactions. Here, we probe the idea of further pretraining a language model to improve performance on a set of downstream tasks. While models that exclusively require protein sequences as input might be preferable in some cases, the release of the AlphaFold database (Varadi et al., 2022) and the ability to generate protein structures on the fly allows performing unsupervised learning on protein structures. Rather than chasing models with billions of parameters, a more carefully designed pretraining regime could generate more meaningful representations and reduce compute resources.

References

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

Bepler, T. and Berger, B. (2019). Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*.

Bepler, T. and Berger, B. (2021). Learning the protein language: Evolution, structure, and function. *Cell systems*, 12(6):654–669.

Bloom, J. D., Drummond, D. A., Arnold, F. H., and Wilke, C. O. (2006). Structural determinants of the rate of protein evolution in yeast. *Molecular biology and evolution*, 23(9):1751–1761.

Bustamante, C. D., Townsend, J. P., and Hartl, D. L. (2000). Solvent accessibility and purifying selection within proteins of escherichia coli and salmonella enterica. *Molecular biology and evolution*, 17(2):301–308.

Chakravarty, S. and Varadarajan, R. (1999). Residue depth: a novel parameter for the analysis of protein structure and stability. *Structure*, 7(7):723–732.

Chandonia, J.-M., Fox, N. K., and Brenner, S. E. (2019). Scope: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic acids research*, 47(D1):D475–D481.

Chowdhury, R., Bouatta, N., Biswas, S., Floristean, C., Kharkar, A., Roy, K., Rochereau, C., Ahdritz, G., Zhang, J., Church, G. M., et al. (2022). Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40(11):1617–1623.

Conant, G. C. and Stadler, P. F. (2009). Solvent exposure imparts similar selective pressures across a range of yeast proteins. *Molecular biology and evolution*, 26(5):1155–1161.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. (2021). Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11.

Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J., and Deane, C. M. (2014). Sabdab: the structural antibody database. *Nucleic acids research*, 42(D1):D1140–D1146.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. (2021). Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127.

Fan, H., Wang, Z., Yang, Y., and Kankanhalli, M. (2022). Continuous-discrete convolution for geometry-sequence modeling in proteins. In *The Eleventh International Conference on Learning Representations*.

Finn, R. D., Clements, J., and Eddy, S. R. (2011). Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37.

Franzosa, E. A. and Xia, Y. (2009). Structural determinants of protein evolution are context-sensitive at the residue level. *Molecular biology and evolution*, 26(10):2387–2395.

Gainza, P., Sverrisson, F., Monti, F., Rodola, E., Boscaini, D., Bronstein, M., and Correia, B. (2020). Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192.

Hermosilla, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez, P. P., Kozlíková, B., Krone, M., Ritschel, T., and Ropinski, T. (2020). Intrinsic-extrinsic convolution and pooling for learning on 3d protein structures. *arXiv preprint arXiv:2007.06252*.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J., and Dror, R. (2020). Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*.

Koenderink, J. J. and Van Doorn, A. J. (1992). Surface shape and curvature scales. *Image and vision computing*, 10(8):557–564.

Laskowski, R. A., Luscombe, N. M., Swindells, M. B., and Thornton, J. M. (1996). Protein clefts in molecular recognition and function. *Protein science: a publication of the Protein Society*, 5(12):2438.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. (2022). Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*.

Marks, D. S., Hopf, T. A., and Sander, C. (2012). Protein structure prediction from sequence variation. *Nature biotechnology*, 30(11):1072–1080.

Mintseris, J. and Weng, Z. (2005). Structure, function, and evolution of transient and obligate protein–protein interactions. *Proceedings of the National Academy of Sciences*, 102(31):10930–10935.

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L., Tosatto, S. C., Paladin, L., Raj, S., Richardson, L. J., et al. (2021). Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419.

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. (2011). Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301.

Pierce, B. G., Wiehe, K., Hwang, H., Kim, B.-H., Vreven, T., and Weng, Z. (2014). Zdock server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, 30(12):1771–1773.

Porollo, A. and Meller, J. (2007). Prediction-based fingerprints of protein–protein interactions. *Proteins: Structure, Function, and Bioinformatics*, 66(3):630–645.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. (2019). Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. (2021). Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118.

Shrake, A. and Rupley, J. A. (1973). Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of molecular biology*, 79(2):351–371.

Somnath, V. R., Bunne, C., and Krause, A. (2021). Multi-scale representation learning on proteins. *Advances in Neural Information Processing Systems*, 34:25244–25255.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007). Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288.

Townshend, R., Bedi, R., Suriana, P., and Dror, R. (2019). End-to-end learning on 3d protein structure for interface prediction. *Advances in Neural Information Processing Systems*, 32.

Varadi, M., Anyango, S., Deshpande, M., Nair, S., Natassia, C., Yordanova, G., Yuan, D., Stroe, O., Wood, G., Laydon, A., et al. (2022). Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444.

Wang, R., Fang, X., Lu, Y., Yang, C.-Y., and Wang, S. (2005). The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119.

Wang, Z., Zhang, Q., Shuang-Wei, H., Yu, H., Jin, X., Gong, Z., and Chen, H. (2022). Multi-level protein structure pre-training via prompt learning. In *The Eleventh International Conference on Learning Representations*.

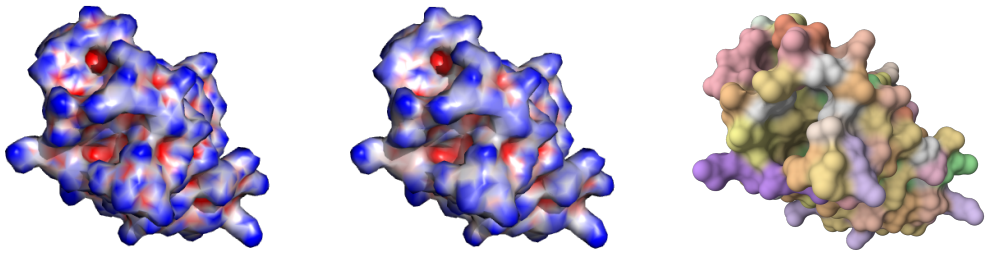
Yang, J.-R., Liao, B.-Y., Zhuang, S.-M., and Zhang, J. (2012). Protein misinteraction avoidance causes highly expressed proteins to evolve slowly. *Proceedings of the National Academy of Sciences*, 109(14):E831–E840.

Yang, K. K., Fusi, N., and Lu, A. X. (2022). Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv*, pages 2022–05.

Zhang, Z., Xu, M., Jamasb, A., Chenthamarakshan, V., Lozano, A., Das, P., and Tang, J. (2022). Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*.

A Shape index smoothing

The shape index is a measure that describes the surface shape of the local curvature. Highly concave regions have values of -1 while highly convex regions have values of $+1$. This measurement allows the evaluation of the local shape independently of the local size (Koenderink and Van Doorn, 1992). We choose to smooth the shape index, as the surface exposed by each amino acid contains more than one vertex. In Figure 2, it can be observed that the smoothing effect better describes the overall shape of each amino acid compared to focusing on the closest vertex.



(a) Shape Index (b) Shape index (avg. 3) (c) Residue type

Figure 2: Protein domain with the SCOPe id *D3GKZA1* represented as a molecular surface. In Figure (a), the shape index of each vertex is drawn as a heat value going from red to blue in the range $[-1, 1]$. (b) shows the same index, but averaged over with the closest 2 neighbours. In Figure (C) the protein surface is coloured based on the amino acid type. The averaging of the shape index maintains the curvature information while better capturing the overall curvature displayed by each amino acid residue.

B Surface generation for protein complexes

A number of 5902 protein interactions are taken from the dataset gathered by (Gainza et al., 2020). This dataset contains interactions from multiple databases (PDBBind (Wang et al., 2005), SAbDab (Dunbar et al., 2014), ZDock(Pierce et al., 2014)) where only the interacting protein chains with high shape complementary are kept. We made a slight change in the way we define the interaction interface. In the source code provided by Gainza et al. (2020), the interface is considered the region on the molecular surface of the protein chains which is not visible upon the analysis of the complex molecular surface. However, they consider the protein complex as all the protein chains appearing in the structure file taken from the Protein Data Bank. The problem arising is that in the case we are interested in the interactions between two protein chains, a third chain could 'cover up' a significant portion of the molecular surface which would incorrectly be labeled as an interface. It can also be the case where contacting chains found in asymmetric crystal units (which are not part of the biological unit) add further noise. Interacting regions with other protein chains from the biological unit might indeed be valid interfaces, however, those are not the interactions of interest. In the provided Supplementary Information it is suggested that interactions with high-shape complementary are chosen to remove protein chain interactions formed in the crystal unit, therefore it is highly unlikely that the surface was deliberately generated this way. We compare the interface size for all protein chains (see Figure 3) and we find that the surfaces provided by Gainza et al. (2020) tend to have a larger interface surface area, with many outliers (all surface is considered an interface). These interactions where the interface is larger than 75% of the protein chain's surface or smaller than 30 vertexes were removed during their training and testing regime. We found that from the data set of protein interactions, 3415 contained extra chains in the PDB file. Out of the 5902 protein interactions, 3003 are selected for protein interface prediction.

C Analysing protein embeddings

We generate embeddings for the SCOPe pretraining dataset using the base PLM model and the model is further pretrained using information stemming from sequence, structure, and surface. In 4, the 2D projection of these embeddings labelled by the structural class is shown.

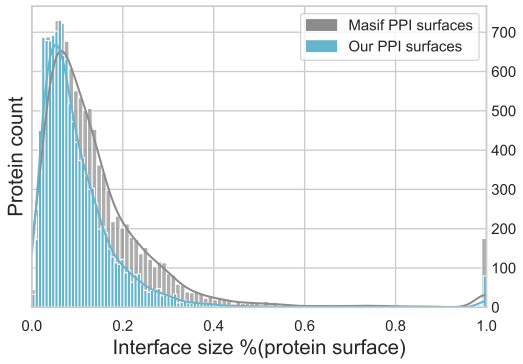


Figure 3: The comparison between protein interface sizes as generated by Gainza et al. (2020) and our implementation. Protein interfaces are overestimated due to the inclusion of additional protein chains within the protein complex which are not the protein interaction of interest. Note the high number of protein interactions where the interface is the entire molecular surface.

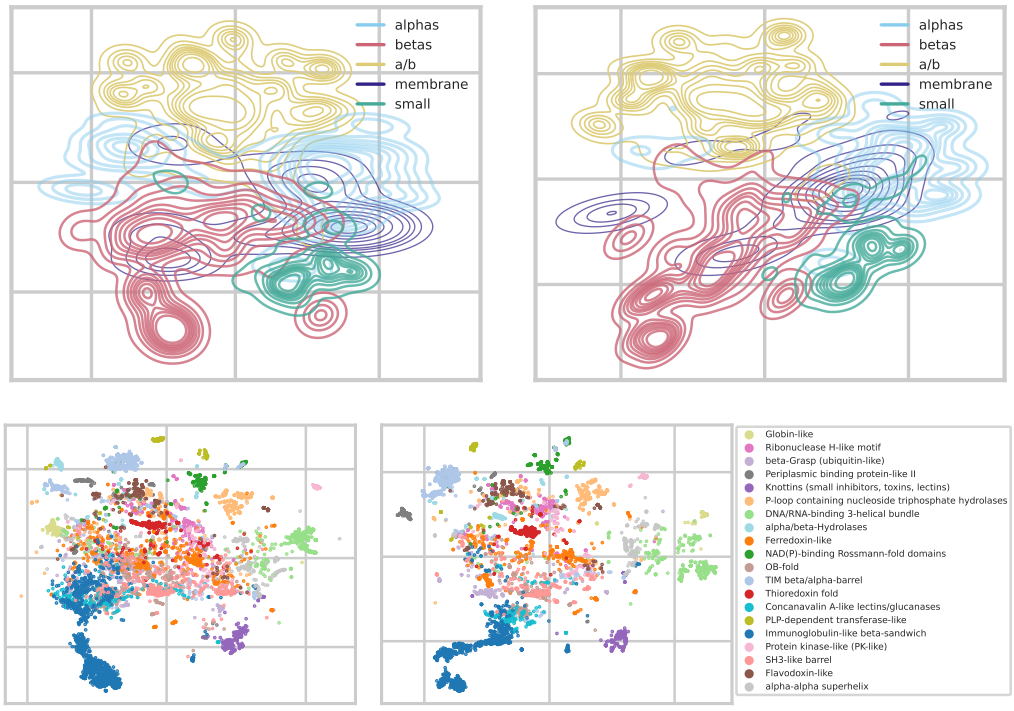


Figure 4: Protein embeddings plots in two dimensions using t-SNE for the SCOPe training dataset. On the left, the embeddings come from the protein language model trained purely on the sequence task. On the right, the embeddings are generated using the model trained on sequence, structure, and surface. The top row shows the separation of SCOPe structural classes, and the bottom row highlights the separation of the top 20 most frequent protein families.

D Interface prediction performance

To understand the loss in performance on the downstream tasks when the PLM is further trained with interface information, we benchmark the model on a dataset of 53 transient interactions (Gainza et al., 2020). In Figure 5, it can be observed that the language model trained with interface information (LM+IFACE) does not achieve high performance. Using the solvent-accessible surface area predictor of a language model trained with surface information (LM+SURF) and no interface data, a similar performance is achieved. This suggests that the information captured from the interaction dataset during the pretraining phase is low and it leads the PLM to overfit. This could explain the loss in performance displayed by the PLM trained with interface data on the downstream tasks.

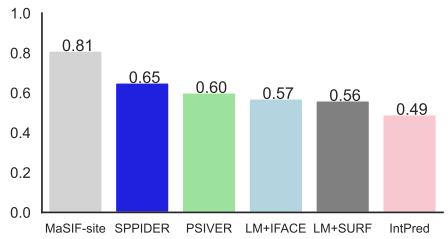


Figure 5: Median AUC values for interface prediction on a benchmark of 53 protein interfaces coming from transient protein interactions. We use the distance to interface predictor to compute interface scores as it offers the best performance compared to the ΔSASA and IFACE predictors.

E Fine tuning on downstream tasks

E.1 Enzyme classification (REACT)

During the fine-tuning phase of the enzyme task, we employ a batch size of 32 protein sequences and utilize an MLP with a single hidden layer size of 1024. We apply a dropout value of 0.7. To ensure the PLM weights are not wrongly updated, we freeze the PLM weights for the initial 15 epochs. We select the best-performing model based on validation accuracy, with a patience threshold of 30 epochs. We fine-tune the PLM 4 different times, each time using distinct learning rate pairs (Head LR|Encoder LR): $1e{-}4|5e{-}5$, $1e{-}4|3ee{-}5$, $1e{-}4|5e{-}5$, and $2e{-}4|4e{-}5$. In case there is no improvement in validation accuracy for 3 consecutive epochs, the learning rates are scaled down by a factor of 0.6.

A number of data samples are removed from the dataset of Hermosilla et al. (2020) to ensure no information leakage between splits: 4y84_X, 5l5e_X, 6huu_J, 4qby_J, 4ya9_J, 5mp9_k, 5mpa_k, 3von_E, 3von_b, 3von_p, 3von_i, 6hed_4, 6hec_5, 6he8_4, 6he9_3, 6he7_6, 6he8_k, 6hed_h, 6hea_i, 6hea_h, 6he9_i, 3mg8_I, 4qlq_W, 6huv_I, 5fga_W, 4qby_W, 5mpa_j, 5mp9_j, 5lf1_b, 5lf1_B, 5gjq_j, 1iru_R, 5gjq_k, 5lf0_W, 5m32_I, 5le5_I, 5lf1_I, 5lf3_I, 5gjq_q.

During training, we augment the protein sequences with a 30% probability using the HMM match and insertion distributions obtained by running HMMSCAN Finn et al. (2011) against the Pfam HMM database Mistry et al. (2021).

E.2 Homology prediction (FOLD)

For the homology prediction task we fine-tune the PLM using a batch size of 100 and learning rates (Head LR|Encoder LR) $3.5e{-}4|4e{-}5$, $2.5e{-}4|4e{-}5$, $2e{-}4|4e{-}5$, $1.9e{-}4|5e{-}5$, $1e{-}4|4e{-}5$, $3e{-}4|4e{-}5$. The remaining parameters are similar to the Enzyme task apart from the augmentation probability which we set to 20%.

E.3 FLIP AAV, GB1 and meltome

For all the downstream tasks, we use an MLP with a single hidden layer of size 1024 and a dropout value of 0.25. No augmentation is applied during training, and we select the best model based on the

validation loss. For the AAV tasks, we use a learning rate combination (Head LR|Encoder LR) of $0.001|0.002$, a patience of 25 epochs, a learning rate scheduler patience of 15 epochs and a batch size of 256. For the GB1 tasks, we use the same hyperparameters, but we extend the early stop patience to 40 epochs. Lastly, for the Meltome prediction task, we use a learning rate combination (Head LR|Encoder LR) of $0.0004|0.0002$, scheduler patience of 10 epochs, early stop patience of 20 epochs, and a batch size of 16. We carry out a total of 6 different random runs, with 3 of them incorporating a random validation dataset(30% of the data instead of the predefined splits). We select the best results from either the runs with the random validation split or the predefined one. The learning rate is decayed using a factor of 0.6.

F Train base PLM for longer

We are particularly interested in discerning whether the enhancement in downstream tasks stems from the inclusion of extra tasks or the prolonged training on the language task. Models trained on surface and structure tasks adopt a multi-task objective and undergo an additional $240k$ training steps with a batch size of 256 for the language task, equivalent to extending the base PLM pretraining by $60k$ steps with a batch size of 1024. Therefore, the base PLM trained for $300k$ steps could be directly compared with the PLM trained on additional tasks. Due to a spike in the loss at that training step, we use the model trained for $380k$ steps for comparison (the first improvement on the validation dataset after the spike).

Table 3 indicates that while there are benefits from extended training, the improvements are marginal. In comparison to models trained on additional tasks (see Table 1), the PLM trained solely on the sequence task for a longer duration does not surpass them. This suggests that the observed improvements in downstream tasks are not solely attributed to prolonged training on the language task.

Table 3: Comparison of PLMs trained only on the language task on downstream tasks.

	Num. steps	FOLD%			REACT %
		Fold	Super.	Fam.	
$\text{ProtEMB}_{\text{LM}}$	240k	26.3 $\pm$ 0.96	43.3 $\pm$ 0.41	90.7 $\pm$ 0.44	81.8 $\pm$ 0.39
$\text{ProtEMB}_{\text{LM}}$	380k	26.6 $\pm$ 0.51	44.8 $\pm$ 0.69	90.6 $\pm$ 0.49	82.1 $\pm$ 1.90