nature biotechnology

Brief Communication

https://doi.org/10.1038/s41587-025-02859-7

Democratizing protein language model training, sharing and collaboration

Received: 27 February 2025

Accepted: 12 September 2025

Published online: 24 October 2025

Check for updates

Jin Su[1,2], Zhikai Li[2], Tianli Tao[2], Chenchen Han[2], Yan He[2], Fengyuan Dai[2], Qingyan Yuan[3], Yuan Gao[4], Tong Si[4], Xuting Zhang[2], Yuyang Zhou[2], Junjie Shan[2], Xibin Zhou[2], Xing Chang[2], Shiyu Jiang[2], Dacheng Ma[5], The OPMC*, Martin Steinegger[6], Sergey Ovchinnikov[7] & Fajie Yuan[2]

Training and deploying large-scale protein language models typically requires deep machine learning expertise—a barrier for researchers outside this field. SaprotHub overcomes this challenge by offering an intuitive platform that facilitates training and prediction as well as storage and sharing of models. Here we provide the ColabSaprot framework built on Google Colab, which potentially powers hundreds of protein training and prediction applications, enabling researchers to collaboratively build and share customized models.

Proteins are fundamental to virtually all biological processes and central to medicine and biotechnology[1–3]. Despite this centrality, deciphering protein structure and function has remained a formidable challenge. This landscape was recently transformed by two breakthroughs: The success of AlphaFold2 (ref. 4) ushered in a new era for structural biology by predicting structures with experimental-level accuracy; in parallel, large-scale protein language models (PLMs) are driving unprecedented advances in function prediction.

This progress is driven by a suite of powerful PLMs that have demonstrated remarkable efficacy across diverse tasks[5–14]. However, leveraging these advanced models presents notable technical hurdles for researchers without extensive machine learning (ML) expertise. The challenges span the entire workflow, from model selection and data preprocessing to the training and evaluation of models with billions of parameters. This complexity creates a critical barrier, hindering broader adoption and innovation by the very researchers who stand to benefit most.

ColabFold[15] addressed a similar accessibility barrier for structure prediction by deploying AlphaFold2 on Google Colab, effectively democratizing its use. Despite this success, a critical gap remains: ColabFold does not support the more complex task of training custom models for function prediction.

To bridge this gap, we introduce ColabSaprot and SaprotHub, a platform designed specifically for protein function prediction. Built on Google Colab, ColabSaprot empowers researchers without ML expertise to train their own task-specific PLMs through an intuitive interface. Crucially, the platform supports a broad spectrum of prediction tasks, ensuring that its utility extends far beyond single-task applications.

Complementing this user-friendly platform, we introduce the Open Protein Modeling Consortium (OPMC), an initiative to foster a collaborative ecosystem for community-driven protein language modeling (Supplementary Information). The OPMC framework enables researchers to share their bespoke models, fine-tune existing ones contributed by peers or apply them directly for their own research. This creates a virtuous cycle of sharing, refinement and application, accelerating collective progress in the field. As the inaugural platform integrated with OPMC, SaprotHub represents the first step in realizing this vision for community-centric artificial intelligence (AI) development.

This work comprises three key contributions: a foundation PLM named Saprot[14] (Fig. 1a,b), ColabSaprot (Fig. 2a–c), enabling easy training (or fine-tuning) and inference of Saprot on the Colab platform through the adapter learning technique[16,17], and SaprotHub, a community repository (Fig. 2d) for storing, sharing, searching and collaborative development of fine-tuned Saprot models. By integrating advanced PLMs, cloud-based computing on Colab and adapter-based fine-tuning techniques, it addresses several key challenges—namely, the difficulty of sharing and collectively using large-scale PLMs, the risk of parameter catastrophic forgetting during continual learning[18] and

[1]Zhejiang University, Hangzhou, China. [2]Westlake University, Hangzhou, China. [3]Suzhou Polynovo Biotech Co., Ltd., Suzhou, China. [4]Shenzhen Institute of Synthetic Biology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. [5]Zhejiang Lab, Hangzhou, China. [6]School of Biological Sciences, Seoul National University, Seoul, South Korea. [7]Massachusetts Institute of Technology, Cambridge, MA, USA.
*A list of authors and their affiliations appears at the end of the paper. e-mail: yuanfajie@westlake.edu.cn

https://doi.org/10.1038/s41587-025-02859-7

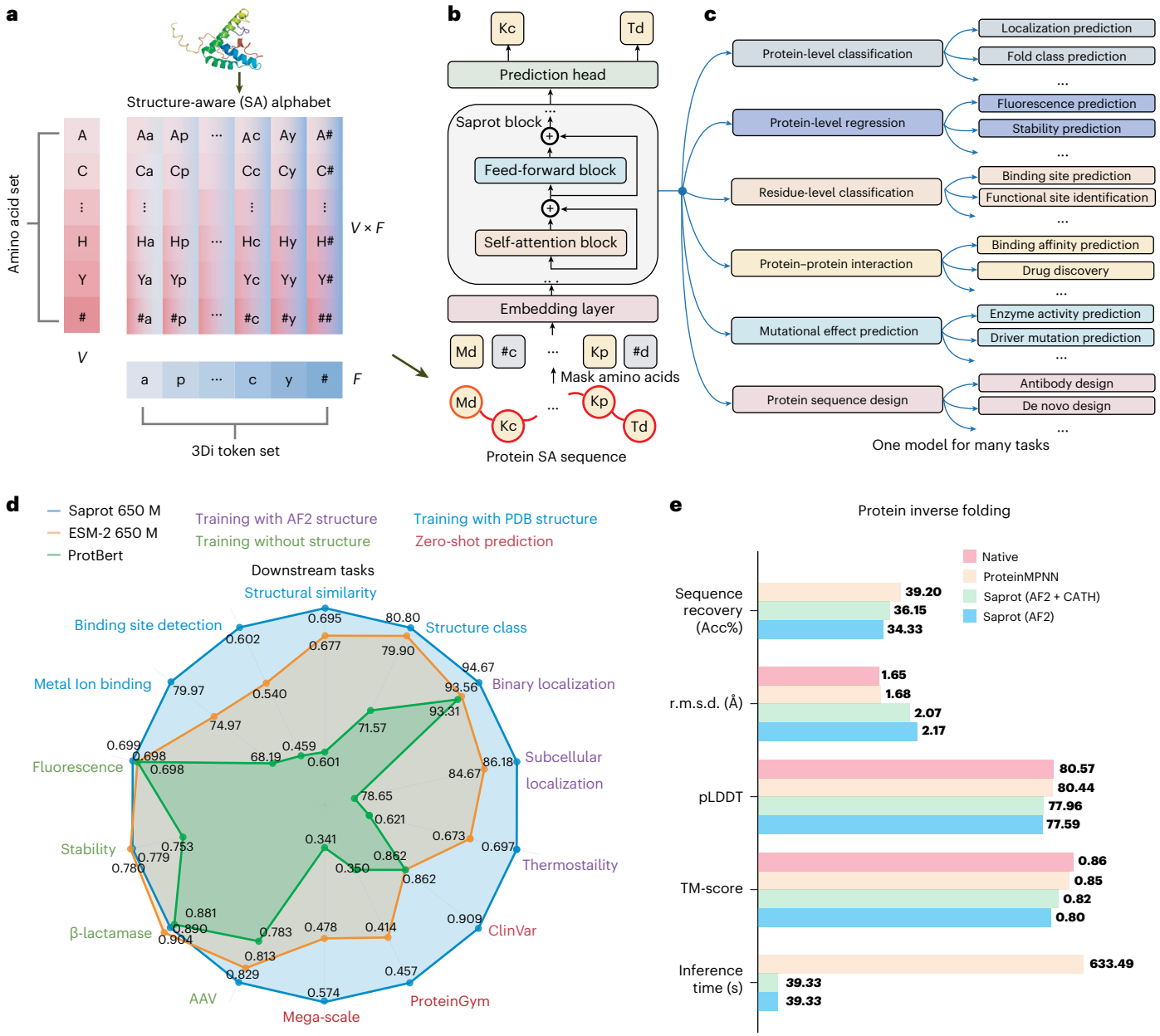

Fig. 1 | Illustration of Saprot. a, The proposed SAA. #a, #p, ..., #y represent only the 3Di token being visible, while A#, C#, ..., Y# represent the AA token being visible. **b**, Model architecture of Saprot. **c**, Saprot supports a wide range of protein prediction tasks (more tasks in Supplementary Table 1). **d**, Performance comparison on 14 diverse prediction tasks. **e**. Performance on protein sequence design. 'AF2+CATH' indicates that Saprot was pretrained using AlphaFold2-predicted structures and fine-tuned with experimental structures from CATH. 'AF2' is a version of Saprot trained only with the AlphaFold2 predicted structure. CATH is also used for training ProteinMPNN.

the need to protect proprietary biological data when sharing results. Below, we detail these three modules.

We first developed Saprot, a cutting-edge, large-scale PLM that forms the foundation for ColabSaprot and SaprotHub. Saprot introduces a novel protein alphabet and representation, distinct from traditional amino acid (AA) sequences or explicit three-dimensional (3D) coordinate structures. This alphabet is structure-aware (SA), with each 'letter' encoding both the AA type and the local geometry of the protein. Formally, the SA alphabet (SAA) is defined as SAA $= \mathcal{V} \times \mathcal{F}$, representing the Cartesian product of $\mathcal{V}$ and $\mathcal{F}$, where $\mathcal{V}$ denotes the 20 AA and $\mathcal{F}$ represents the 20 structural (3Di) letters (Fig. 1a). These 3Di tokens are derived from protein 3D structures through Foldseek discretization[19]. The SAA encompasses all possible combinations of AA (capital letters) and 3Di tokens (lowercase letters) in SAA, such as Aa, Ap, Ac, ..., Ya, Yp, Yy, allowing proteins to be represented as a sequence of SA tokens that captures both primary and tertiary structures (Fig. 1b and Methods). Despite its conciseness, applying the SAA successfully addresses the key challenges of scalability and overfitting in training on large-scale AlphaFold2-generated atomic structures (Supplementary Note 1 and Supplementary Fig. 1). The adoption of 'AA + structural token' sequences (Fig. 1b) for protein representation has garnered increasing attention in much subsequent studies[20–27], emerging as a promising paradigm for protein representation.

The Saprot model uses a bidirectional Transformer[28] architecture (Fig. 1b). It was pretrained to reconstruct certain partially masked tokens in the SA token sequences (Methods). The model was trained from scratch on 40 million protein SA sequences, filtered from AlphaFold's 214 million proteins[29] at 50% identity (Methods). Saprot is

https://doi.org/10.1038/s41587-025-02859-7

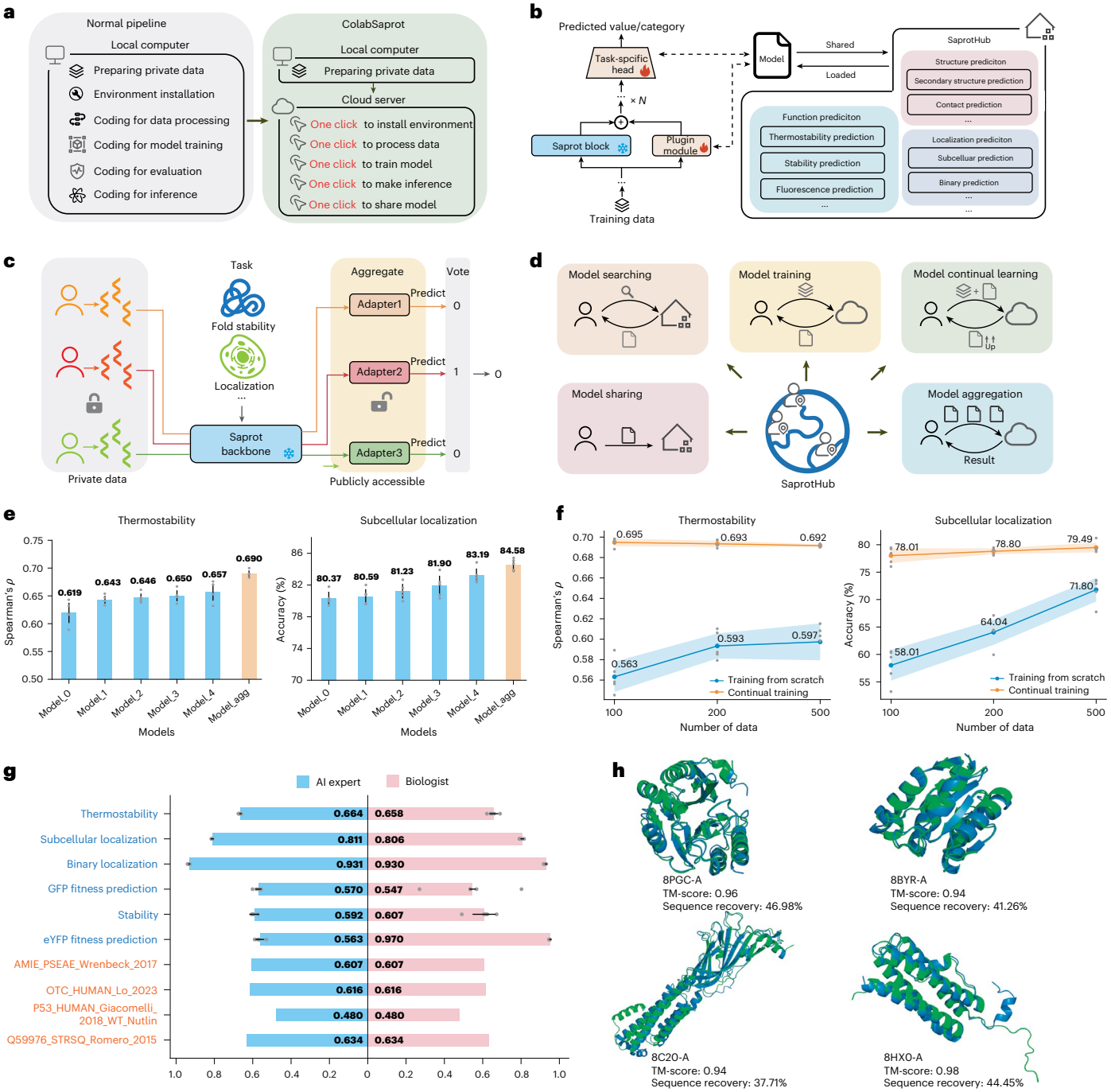

Fig. 2 | Illustration of ColabSaprot and SaprotHub. a, Comparison of model training and inferring between normal pipeline and ColabSaprot. ColabSaprot streamlines the process, offering simplified procedures for model training and inference with just a few clicks. **b**, A lightweight plugin architecture (that is, adapter) is integrated within Saprot to facilitate efficient training, sharing and coconstruction. Throughout this process, the model parameters of the Saprot backbone remain unchanged. **c**, ColabSaprot performs predictions by aggregating multiple shared models using adapters in SaprotHub without the need for private training data. **d**, Illustration of SaprotHub features, enabling biologists to coshare, cobuild, couse and collaborate within the community. **e**, ColabSaprot's community-wide model collaboration mechanism (**c**) allows it to achieve higher performance (orange bars) by aggregating multiple individually trained models (blue bars). Each individual model is trained with its own data, which may or may not overlap (reflecting real-world situations; Supplementary Table 10). Data represent the mean ± s.d. of $n = 5$ independent experiments. **f**, By continually learning on models trained and shared by other biologists, ColabSaprot substantially outperforms those training-from-scratch models, especially when users lack sufficient training data (the x axis represents the number of training examples). Data represent the mean ± s.d. of $n = 5$ independent experiments. **g**, User study on supervised fine-tuning and zero-shot mutation effect prediction tasks (Supplementary Table 11). For fine-tuning tasks, the data represent the mean ± s.d. of $n = 5$ independent experiments. **h**, User study on the inverse folding task. Experimental structures are shown in green, while predicted structures are in blue. TM-score, template modeling score.

available in three sizes, 35M, 650M and 1.3B, indicating the number of parameters (M, million; B, billion) (Supplementary Table 2). The 650M model, used for most evaluations unless otherwise specified (Supplementary Tables 3 and 4), was trained for 3 months using 64 NVIDIA A100 80-GB GPUs, representing computational resources comparable to those used for ESM-2 (650M).

After its pretraining, Saprot has become a general-purpose foundation PLM that excels across diverse protein prediction tasks, such as various supervised training tasks[30–32] (encompassing both regression and classification at the protein level or residue level), zero-shot mutation effect prediction[33,34] and protein sequence design[35] (Fig. 1c,d and Supplementary Tables 1 and 5). Figure 1d shows Saprot's superior performance across 14 different protein prediction tasks compared to two well-known PLMs: ESM-2 (ref. 6) and ProtBert[9] (see detailed task, dataset, baseline and experimental setup descriptions in Methods and Supplementary Table 6; more baseline comparison in Supplementary Tables 7 and 8). For tasks where protein structural information is available (indicated by purple, blue and red colors), Saprot consistently surpasses both models. Even in scenarios without structural data (green tasks), Saprot maintains competitive performance with ESM-2 under the fine-tuning settings. Notably, Saprot substantially outperforms ESM-2 in three zero-shot mutational effect prediction tasks (Methods): Mega-scale[36] (0.574 versus 0.478), ProteinGym[37] (0.457 versus 0.414), and ClinVar[38] (0.909 versus 0.862) (Supplementary Table 7). Furthermore, despite being trained with a masked language modeling objective not specifically optimized for generation, Saprot performs effectively in protein sequence design while achieving a 16-fold acceleration in inference speed compared to ProteinMPNN[35](Fig. 1e) (Methods, Supplementary Fig. 2 and Supplementary Table 9). Additionally, recent studies demonstrated Saprot's effectiveness across a broad spectrum of applications[39,40], including protein engineering and de novo design[41–43], fitness and stability prediction[44,45], molecular understanding[46,47], fast and sensitive structure search[48] and drug–target interaction prediction[49,50]. This versatility across multiple protein-related tasks supports SaprotHub's vision for community collaboration (Supplementary Table 1).

We then developed the ColabSaprot platform by integrating Saprot into Google Colab's infrastructure to support PLM training and prediction (Methods and Supplementary Fig. 3). ColabSaprot enables seamless deployment and execution of various task-specific trained Saprot models, eliminating the need for environment setup and code debugging. It also allows researchers to initiate training sessions with just a few clicks (Fig. 2a). ColabSaprot is designed to accommodate all tasks within the original Saprot framework, enabling direct prediction for tasks such as zero-shot mutation effect prediction[34,51,52] and protein sequence design[35,53]. For mutation effect prediction, it implements single-site, multisite and whole-protein-sequence single-site saturation mutation. For protein design, it can generate de novo sequences on the basis of a given backbone structure.

For these supervised training tasks, a particular focus of this work, users can fine-tune ColabSaprot with their own experimental data. Here, we implement a parameter-efficient fine-tuning technique[16,17] by integrating lightweight adapter networks into ColabSaprot (Fig. 2b and Methods). During training, only adapter parameters are updated, achieving comparable accuracy as fine-tuning all Saprot parameters (Supplementary Fig. 4). This design not only improves learning efficiency[54–57] but also establishes a collaborative and centralized framework that enables biologists to fine-tune task-specific Saprot within the research community, particularly through cloud-based environments (Fig. 2c–f). With adapters and the ColabSaprot interface, researchers can easily store and exchange their retrained Saprot models in SaprotHub by loading or uploading adapter networks instead of the full pretrained model. As adapter networks contain much fewer parameters (around 1% of the whole Saprot model), this method greatly reduces storage, communication and management burdens on SaprotHub, making model accessibility, coconstruction, couse and cosharing possible for a broader and diverse scientific community (Fig. 2b–d and Methods).

Additionally, we developed several key features to streamline workflows for researchers. The ColabSaprot interface supports both sequence and structure inputs, offering automated dataset handling capabilities including efficient large-file upload mechanisms, real-time training monitoring with loss visualization, automatic saving and evaluation of the best checkpoints, breakpoint training resumption and numerous safety checks to minimize user errors. To enhance GPU memory efficiency, we implemented adaptive batch sizing through gradient accumulation[4]. The platform also includes customizable settings, allowing researchers to modify code and adjust training parameters for specific research needs.

Lastly, we developed SaprotHub with an integrated search engine, providing a centralized platform for sharing and collaboratively developing peer-retrained Saprot models within the biology community (Fig. 2d and Supplementary Fig. 5). Through the ColabSaprot interface, we implemented features like model storage, model sharing, model search, model continuous learning and multiple model aggregation for enhanced performance (Methods). Specifically, SaprotHub offers three key advantages. First, researchers can share their trained models on SaprotHub for broader scientific applications without worrying about the leakage of private data. This feature effectively promotes knowledge dissemination and has the potential to establish a new paradigm for collaborative research (Fig. 2c). Second, researchers can leverage shared models on SaprotHub contributed by peers to perform continuous training on their own dataset[18]. This approach is particularly advantageous in data-limited scenarios, as fine-tuning from a better pretrained model typically delivers superior predictive performance (Fig. 2f). Furthermore, these fine-tuned models can be contributed back to the hub through one-click upload of the adapter networks, fostering a collaborative ecosystem where the community collectively and iteratively enhances the available PLM resources. Third, as the SaprotHub community expands, it will accumulate a diverse collection of models for various protein prediction tasks, with multiple fine-tuned Saprot models becoming available for specific protein functions. To leverage this growing model collection, we implemented a model aggregation mechanism in ColabSaprot (Methods), enabling users to enhance predictive performance (Fig. 2e) through the integration of multiple existing models (Fig. 2c).

Saprot and SaprotHub have attracted community attention and demonstrated usefulness through multiple wet lab validations. The T.Z. team at a commercial biological company used ColabSaprot for the zero-shot single-point mutation prediction on a xylanase (XP_069217686.1) from *Mycothermus thermophilus* and experimentally validated the top 20 predicted variants. Among these, 13 variants exhibited enhanced enzyme activity, with R59S showing a 2.55-fold improvement and F212N demonstrating a 1.88-fold increase in enzyme activity along with enhanced thermostability (Methods, Supplementary Tables 12–14 and Supplementary Figs. 6 and 7). Similarly, the X.C. lab used ColabSaprot to perform zero-shot single-point mutation predictions on TDG, a uracil-*N*-glycosylase variant. Following experimental validation in HeLa cells, the top 20 variants were incorporated into the nCas9 protein. At the Dicer 1 target site, 17 of the 20 predicted mutations showed enhanced editing efficiency compared to the wild type, as measured by the percentage of T-to-G substitutions at position T5 (the fifth thymine in the target sequence). Notably, three substitutions (L74E, H11K and L74Q) achieved nearly doubled editing efficiency (Methods and Supplementary Table 15). Another OPMC member recently fine-tuned Saprot using a dataset of approximately 140,000 GFP variants with corresponding fluorescence intensities to predict brighter avGFP variants from a pool of 5 million candidates. Experimental validation revealed that seven of the top nine predicted double-site variants exhibited enhanced

fluorescence compared to the wild type, with one variant reaching more than eightfold of the wild-type fluorescence intensity (Methods and Supplementary Table 16). Similarly, the J. Zheng lab shared an eYFP fluorescence prediction model, trained on 100,000 experimentally validated variants, that achieved a Spearman correlation (ρ) of 0.94 with experimental fluorescence intensity on an independent test set of 6,000 variants, demonstrating near-experimental accuracy for double-site and triple-site mutants (Model-EYFP_100K-650M on the SaprotHub webpage). We recently received more feedback from community researchers who obtained positive wet lab results using ColabSaprot.

We also conducted a user study by recruiting 12 biology researchers (without an ML background) and compared their performance to that of an AI expert (Methods). The results demonstrated that, with ColabSaprot and SaprotHub, biology researchers can train and use state-of-the-art PLMs with performance comparable to that of an AI expert (Fig. 2g,h). Notably, in certain scenarios—such as the eYFP fitness prediction task illustrated in Fig. 2g—biologists leveraging preexisting models from SaprotHub achieved higher prediction accuracy than AI experts. This higher performance stems from the fact that these shared models have been trained on larger or higher-quality datasets, highlighting the potential of model sharing within the scientific community—a key argument in this paper.

ColabSaprot and SaprotHub enable biology researchers to train and share sophisticated PLMs for diverse prediction tasks, even without extensive AI expertise. This platform empowers the broader protein research community to contribute and exchange PLMs, facilitating collaborative research and knowledge sharing through peer-trained models. We have made both Saprot and ColabSaprot open-source, providing a framework for other PLMs to develop their own model hubs. Importantly, ColabSaprot and SaprotHub represent just the first step in this evolution; our OPMC members have expanded this ecosystem by integrating more cutting-edge PLMs, including ProTrek (35M and 650M)[58], ESM-2 (35M, 150M and 650M)[6], ESM-1b (650M)[5]), ProtBert (420M)[9] and ProtT5 (1.2B)[9] into the OPMC framework, thereby democratizing access to diverse PLMs for biologists worldwide.

This community-wide participation approach to protein language modeling aligns with the OPMC vision. Our goal here is to inspire and foster the cooperative construction of open PLMs through SaprotHub. We envision SaprotHub as the catalyst that initializes OPMC, driving innovation and collaboration in the field.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-025-02859-7.

References

1. Drews, J. Drug discovery: a historical perspective. *Science* **287**, 1960–1964 (2000).
2. Jacob, François & Monod, J. Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.* **3**, 318–356 (1961).
3. Glickman, M. H. & Ciechanover, A. The ubiquitin–proteasome proteolytic pathway: destruction for the sake of construction. *Phys. Rev.* **82**, 373–428 (2002).
4. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
5. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
6. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
7. Meier, J. et al. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Proc. 35th International Conference on Neural Information Processing Systems* (eds Ranzato, M. et al.) (NIPS, 2021).
8. Rao, R. M. et al. MSA transformer. In *Proc. 38th International Conference on Machine Learning* (eds Meila, M. & Zhang, T.) (PMLR, 2021).
9. Elnaggar, A. et al. ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2021).
10. Heinzinger, M. et al. Bilingual language model for protein sequence and structure. *NAR Genom. Bioinform.* **6**, lqae150 (2024).
11. Notin, P. et al. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) (PMLR, 2022).
12. Nijkamp, E., Ruffolo, J. A., Weinstein, E. N., Naik, N. & Madani, A. ProGen2: exploring the boundaries of protein language models. *Cell Syst.* **14**, 968–978 (2023).
13. Ferruz, N., Schmidt, S. & Höcker, B. ProtGPT2 is a deep unsupervised language model for protein design. *Nat. Commun.* **13**, 4348 (2022).
14. Su, J. et al. SaProt: orotein language modeling with structure-aware vocabulary. In *Proc. 12th International Conference on Learning Representations* (ed Kim, B.) (ICLR, 2023).
15. Mirdita, M. et al. ColabFold: making protein folding accessible to all. *Nat. Methods* **19**, 679–682 (2022).
16. Hu, E. J. et al. LoRA: low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations* https://openreview.net/pdf?id=nZeVKeeFYf9 (ICLR, 2022).
17. Pfeiffer, J. et al. AdapterHub: a framework for adapting transformers. In *Proc. 2020 EMNLP (Systems Demonstrations)* https://aclanthology.org/2020.emnlp-demos.7.pdf (Association for Computational Linguistics, 2020).
18. Kirkpatrick, J. et al. Overcoming catastrophic forgetting in neural networks. *Proc. Natl Acad. Sci. USA* **114**, 3521–3526 (2017).
19. van Kempen, M. et al. Fast and accurate protein structure search with Foldseek. *Nat. Biotechnol.* **42**, 243–246 (2024).
20. Hayes, T. et al. Simulating 500 million years of evolution with a language model. *Science* **387**, 850–858 (2025).
21. Li, M. et al. ProSST: protein language modeling with quantized structure and disentangled attention. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)* https://openreview.net/forum?id=4Z7RZixpJQ&referrer=%5Bthe%20profile%20of%20Bozitao%20Zhong%5D(%2Fprofile%3Fid%3D~Bozitao_Zhong1) (NeurIPS, 2024).
22. Wang, X. et al. DPLM-2: a multimodal diffusion protein language model. In *The Thirteenth International Conference on Learning Representation* https://openreview.net/pdf?id=5z9GjHgerY (ICLR, 2025).
23. Tan, Y., Wang, R., Wu, B., Hong, L. & Zhou, B. Retrieval-enhanced mutation mastery: augmenting zero-shot prediction of protein language model. Preprint at https://arxiv.org/abs/2410.21127 (2024).
24. Pourmirzaei, M., Esmaili, F., Pourmirzaei, M., Wang, D. & Xu, D. Prot2Token: a multi-task framework for protein language processing using autoregressive language modeling. In *ICML 2024 Workshop on Efficient and Accessible Foundation Models for Biological Discovery* https://openreview.net/pdf?id=5z9GjHgerY (2024).
25. Gao, K. et al. Tokenizing 3D molecule structure with quantized spherical coordinates. Preprint at https://arxiv.org/abs/2412.01564 (2024).

26. Lin, X. et al. Tokenizing foldable protein structures with machine-learned artificial amino-acid vocabulary. Preprint at *bioRxiv* https://doi.org/10.1101/2023.11.27.568722 (2023).
27. Ivanisenko, N. V. et al. SEMA 2.0: web-platform for B-cell conformational epitopes prediction using artificial intelligence. *Nucleic Acids Res.* **52**, W533–W539 (2024).
28. Devlin, J., Chang, Ming-Wei, Lee, K. & Toutanova, K., BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* 4171–4186 (Association for Computational Linguistics, 2019).
29. Varadi, M. et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2022).
30. Rao, R. et al. Evaluating protein transfer learning with tape. In *Proc. 33rd International Conference on Neural Information Processing Systems* (eds Wallach, H. M. et al.) (NIPS, 2019).
31. Kucera, T., Oliver, C., Chen, D. & Borgwardt, K. ProteinShake: building datasets and benchmarks for deep learning on protein structures. In *Proc. 36th International Conference on Neural Information Processing Systems* (eds Oh, A. et al.) (NIPS, 2024).
32. Xu, M. et al. PEER: a comprehensive and multi-task benchmark for protein sequence understanding. In *Proc. 35th International Conference on Neural Information Processing Systems* (eds Ranzato, M. et al.) (2021).
33. Hie, B., Zhong, E. D., Berger, B. & Bryson, B. Learning the language of viral evolution and escape. *Science* **371**, 284–288 (2021).
34. Frazer, J. et al. Disease variant prediction with deep generative models of evolutionary data. *Nature* **599**, 91–95 (2021).
35. Dauparas, J. et al. Robust deep learning-based protein sequence design using proteinmpnn. *Science* **378**, 49–56 (2022).
36. Tsuboyama, K. et al. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature* **620**, 434–444 (2023).
37. Notin, P. et al. ProteinGym: large-scale benchmarks for protein fitness prediction and design. In *Proc. 36th International Conference on Neural Information Processing Systems* (eds Oh, A. et al.) (NIPS, 2024).
38. Landrum, M. J. et al. ClinVar: improving access to variant interpretations and supporting evidence. *Nucleic Acids Res.* **46**, D1062–D1067 (2018).
39. Tan, Y. et al. VenusX: unlocking fine-grained functional understanding of proteins. Preprint at https://arxiv.org/abs/2505.11812 (2025).
40. Yan, S. et al. Protap: a benchmark for protein modeling on realistic downstream applications. Preprint at https://arxiv.org/abs/2506.02052 (2025).
41. Zhou, Z. et al. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nat. Commun.* **15**, 5566 (2024).
42. Dai, F. et al. Toward de novo protein design from natural language. Preprint at *bioRxiv* https://doi.org/10.1101/2024.08.01.606258 (2024).
43. Meshchaninov, V. et al. Diffusion on language model encodings for protein sequence generation Preprint at https://arxiv.org/abs/2403.03726 (2024).
44. Sagawa, T., Kanao, E., Ogata, K., Imami, K. & Ishihama, Y. Prediction of protein half-lives from amino acid sequences by protein language models. Preprint at *bioRxiv* https://doi.org/10.1101/2024.09.10.612367 (2024).
45. Bushuiev, A. et al. Training on test proteins improves fitness, structure, and function prediction. Preprint at https://arxiv.org/abs/2411.02109 (2024).
46. Zhuang, X. et al. Advancing biomolecular understanding and design following human instructions. *Nat. Mach. Intell.* **7**, 1154–1167 (2025).
47. Zhou, X. et al. Decoding the molecular language of proteins with Evola. Preprint at *bioRxiv* https://doi.org/10.1101/2025.01.05.630192 (2025).
48. Wang, L., Zhang, X., Wang, Y. & Xue, Z. SSAlign: ultrafast and sensitive protein structure search at scale. Preprint at *bioRxiv* https://doi.org/10.1101/2025.07.03.662911 (2025).
49. Meng, Z., Meng, Z. & Ounis, I. FusionDTI: fine-grained binding discovery with token-level fusion for drug-target interaction. Preprint at https://arxiv.org/abs/2406.01651 (2024).
50. McNutt, A. T. et al. Scaling structure aware virtual screening to billions of molecules with sprint. Preprint at https://arxiv.org/abs/2411.15418 (2025).
51. He, Y. et al. Protein language models-assisted optimization of a uracil-N-glycosylase variant enables programmable T-to-G and T-to-C base editing. *Mol. Cell* **84**, 1257–1270 (2024).
52. Riesselman, A. J., Ingraham, J. B. & Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **15**, 816–822 (2018).
53. Hsu, C. et al. Learning inverse folding from millions of predicted structures. In *Proc. 39th International Conference on Machine Learning* (eds Chaudhuri, K. et al.) (PMLR, 2022).
54. Sledzieski, S. et al. Democratizing protein language models with parameter-efficient fine-tuning. *Proc. Natl Acad. Sci. USA* **121**, e2405840121 (2024).
55. Zeng, S., Wang, D. & Xu, D. PEFT-SP: parameter-efficient fine-tuning on large protein language models improves signal peptide prediction. *Genome Res.* **34**, 1445–1454 (2024).
56. Sledzieski, S., Kshirsagar, M., Berger, B., Dodhia, R. & Ferres, J. L. Parameter-efficient fine-tuning of protein language models improves prediction of protein-protein interactions. In *Machine Learning for Structural Biology Workshop, NeurIPS 2023* https://www.mlsb.io/papers_2023/Parameter-Efficient_Fine-Tuning_of_Protein_Language_Models_Improves_Prediction_of_Protein-Protein_Interactions.pdf (2023).
57. Wang, D. et al. S-PLM: structure-aware protein language model via contrastive learning between sequence and structure. *Adv. Sci.* **12**, 2404212 (2025).
58. Su, J., Zhou, X., Zhang, X. & Yuan, F. A trimodal protein language model enables advanced protein searches. *Nat. Biotechnol.* https://doi.org/10.1038/s41587-025-02836-0 (2025).

https://doi.org/10.1038/s41587-025-02859-7

The OPMC

Anthony Gitter[8], Milot Mirdita[6], Kevin K. Yang[9], Pascal Notin[10], Debora S. Marks[10], Pranam Chatterjee[11], Rohit Singh[12], Philip A. Romero[12], Michael Heinzinger[13], Jianming Liu[2], Jia Zheng[2], Stan Z. Li[2], Anping Zeng[2], Huaizong Shen[2], Jijie Chai[2], Feng Ju[2], Noelia Ferruz[14,15], Anum Glasgow[16], Philip M. Kim[17], Christopher Snow[18], Vasilis Ntranos[19], Jianyi Yang[20], Liang Hong[21], Caixia Gao[22], Tong Si[23], Michael Bronstein[24], Xing Chang[2], Martin Steinegger[6], Sergey Ovchinnikov[7], Fajie Yuan[2], Jin Su[1,2], Zhikai Li[2], Tianli Tao[2], Chenchen Han[2], Yan He[2], Fengyuan Dai[2], Xuting Zhang[2], Yuyang Zhou[2], Junjie Shan[2], Xibin Zhou[2], Shiyu Jiang[2], Dacheng Ma[5], Yuan Gao[23], Jiawei Zhang[2], Yuliang Fan[2], Yuyang Tao[25], Linqi Cheng[26], Xinzhe Zheng[26], Lei Chen[27], Rui Long[27], Lingjie Kong[28], Zhongji Pu[29], Jiaming Guan[30], Tianyuan Zhang[3], Cheng Li[3] & Qingyan Yuan[3]

[8]University of Wisconsin-Madison, Madison, WI, USA. [9]Microsoft Research, Cambridge, MA, USA. [10]Harvard Medical School, Boston, MA, USA. [11]University of Pennsylvania, Philadelphia, PA, USA. [12]Duke University, Durham, NC, USA. [13]Helmholtz Zentrum München, Neuherberg, Germany. [14]Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology, Barcelona, Spain. [15]Universitat Pompeu Fabra(UPF), Barcelona, Spain. [16]Columbia University, New York, NY, USA. [17]University of Toronto, Toronto, Ontario, Canada. [18]Colorado State University, Fort Collins, CO, USA. [19]University of California, San Francisco, CA, USA. [20]Shandong University, Jinan, China. [21]Shanghai Jiao Tong University, Shanghai, China. [22]Chinese Academy of Sciences, Beijing, China. [23]Chinese Academy of Sciences, Shenzhen, China. [24]University of Oxford, Oxford, United Kingdom. [25]ShanghaiTech University, Shanghai, China. [26]Rice University, Houston, TX, USA. [27]Shenzhen Lions King Hi-Tech Co., Ltd., Shenzhen, China. [28]South China Agricultural University, Guangzhou, China. [29]Xianghu laboratory, Hangzhou, China. [30]Hefei MiQro Era Digital Technology Co., Ltd., Hefei, China.

Methods

Constructing SA protein sequence

The SA vocabulary encompasses both residue and structure information, as illustrated in Fig. 1a. Given a protein P, its primary sequence can be denoted as $(s_1, s_2, \ldots, s_n)$, where $s_i \in \mathcal{V}$ represents the residue at the ith site and $\mathcal{V}$ represents the residue alphabet. Drawing inspiration from the vector quantization learning technique[59], we encode protein 3D structures into discrete residue-like structural tokens. Here, we use Foldseek[19], a fast and accurate protein structure aligner. Through Foldseek, we have a structure alphabet $\mathcal{F}$, wherein P is expressed as the sequence $(f_1, f_2, \ldots, f_n)$, with $f_j \in \mathcal{F}$ representing the 3Di token for the jth residue site. To maintain simplicity, we adhere to the default configuration of Foldseek, which sets the size m of $\mathcal{F}$ to 20. We then combine the residue and structure tokens per residue site, generating a new SA protein sequence $P = (s_1f_1, s_2f_2, \ldots, s_nf_n)$, where $s_if_i \in \mathcal{V} \times \mathcal{F}$ is the so-called SA token naturally fusing both residue and geometric conformation information. The SA token protein sequence can then be fed into a standard Transformer encoder as basic input. It is important to note that we also introduce a mask signal '#' to both the residue and the structure alphabet, which results in 's_i#' and '#f_i', indicating that only residue or structure information is available. The size of the SA vocabulary is $21 \times 21 = 441$.

Model architecture and pretraining of Saprot

Saprot follows the same network architecture and parameter size as ESM-2 (ref. 6), which draws inspiration from the BERT[28] model in natural language processing. The primary difference lies in the embedding layer: Saprot incorporates 441 SA tokens in place of the conventional 20 AA tokens. This nearly identical architecture enables straightforward performance comparison to the ESM model.

Saprot is pretrained with the masked language modeling (MLM) objective[28], like ESM-2 and BERT. Formally, for a protein sequence $P = (s_1f_1, s_2f_2, \ldots, s_nf_n)$, the input and output can be represented as input: $(s_1f_1, \ldots, \#f_i, \ldots, s_nf_n) \rightarrow$ output: s_if_i (Fig. 1b). Given that the 3Di token may not always be accurate for certain regions in predicted structures by AlphaFold2, f_i in $\#f_i$ is made visible during training so as to reduce the emphasis the model places on its predictions.

AlphaFold2 (AF2) predictions include predicted local distance difference test (pLDDT) confidence scores that indicate the precision of predicted atom coordinates. Therefore, we implement specialized handling for regions with low confidence scores. During pretraining, regions with pLDDT scores below 70 are processed distinctly. When these regions are selected for MLM prediction, we use the 's_i#' token as the prediction target, while masking the input SA sequence with the '##' token. This approach encourages the model to predict residue types based solely on contextual information. When these low-confidence regions are not selected for MLM prediction, we use the 's_i#' token in the input, ensuring the model relies solely on residue context rather than inaccurate structural information. For downstream tasks, we maintain consistency with the pretraining protocol by applying the same handling to regions with pLDDT scores below 70. These regions are represented using 's_i#' tokens, with only residue information remaining visible.

Saprot 35M and 650M underwent typical pretraining from scratch as described above. In contrast, Saprot 1.3B used an efficient training strategy[60] by architecturally combining two identical 33-layer Saprot 650M models. The initialization process involved duplicating the pretrained 650M model's parameters to populate both the lower (layers 1–33) and upper (layers 34–66) sections of the 1.3B architecture. After initialization, Saprot 1.3B was trained following the same training protocol used for the 35M and 650M models, except that 30% of the protein sequences in a training batch were transformed into the AA sequence only format $(s_1\#, s_2\#, \ldots, s_n\#)$ to enhance the model's ability to handle proteins without available structural information. Training was terminated upon convergence of the loss function.

Processing pretraining dataset

We followed the procedures outlined in ESM-2 (ref. 6) to generate sequence identity filtered protein data. Subsequently, we acquired all AF2 structures through the AlphaFold DB website (https://alphafold.ebi.ac.uk/), using the UniProt IDs of protein sequences. Proteins without structures in AlphaFold DB were removed. This process yielded a collection of approximately 40 million structures. Using Foldseek, we encoded these structures into 3Di tokens. Subsequently, we formulated SA sequences by combining residue and 3Di tokens, treating them as a single SA token at each position. These datasets were used for training all three versions of the Saprot models.

Hyperparameters for pretraining

Following ESM-2 and BERT, during training, 15% of the SA tokens in each batch were masked. We replaced the SA token s_if_i with the $\#f_i$ token 80% of the time, while 10% of the tokens were replaced with randomly selected tokens, and the other 10% tokens remained unchanged. For the optimization of Saprot, we adopted similar hyperparameters to those used in the ESM-2 training phase. Specifically, we used the AdamW optimizer[61], setting $\beta_1 = 0.9$, $\beta_2 = 0.98$ and we used an L_2 weight decay of 0.01. We gradually increased the learning rate from 0 to 4×10^{-4} over the first 2,000 steps and linearly lowered it to 4×10^{-5} from 150,000 steps to 1.5 million steps. The overall training phase lasted approximately 3 million steps. Like the ESM model, we also truncated them to a maximum of 1,024 tokens and our batch size consisted of 512 sequences. Additionally, we used mixed precision training to train Saprot.

Descriptions of baseline models

We compared Saprot to several prominent PLMs (Supplementary TableS 7 and 8). For supervised learning tasks, we compared Saprot to ESM-2 (the 650M version)[6], ProtBert (the BFD version)[9], MIF-ST[62], GearNet[63] and ESM-3 (ref. 20). The first two models use residue sequences as input, while the latter three models incorporate both residue and structures as input. ESM-2 (650M) stands out as the primary baseline for comparison, given its similar model architecture, size and training approach when compared to Saprot. ESM-2 also offers a 15B version, which can be challenging to fine-tune even on GPUs with 80 GB of memory. Therefore, we only conducted comparisons to ESM-2 (15B) for zero-shot mutational effect prediction tasks, which can be achieved without the need for fine-tuning.

For the zero-shot mutational effect prediction task, we compare to the state-of-the-art ESM-2, ProtBert, ESM-1v (ref. 7), Tranception L (without MSA retrieval)[11], MSA Transformer[8], EVE[34] and ESM-3 models. For the protein inverse folding task, we compare to ProteinMPNN[35] as baseline.

The formula for the zero-shot mutation effect prediction task

Previous sequence-based PLMs like the ESM models predict mutational effects using the log odds ratio at the mutated position. The calculation can be formalized as follows:

$$\sum_{t \in T} \left[\log P\left(x_t = s_t^{mt} | x_{\setminus T}\right) - \log P\left(x_t = s_t^{wt} | x_{\setminus T}\right) \right] \tag{1}$$

Here, T represents all mutations and $s_t \in \mathcal{V}$ is the residue type for mutant and wild-type sequence. We slightly modify the formula above to adapt to the SAA in Saprot, where the probability assigned to each residue corresponds to the summation of tokens encompassing that specific residue type, as shown below.

$$\sum_{t \in T} \left[\log \sum_{f \in \mathcal{F}} P\left(x_t = s_t^{mt} f | x_{\setminus T}\right) - \log \sum_{f \in \mathcal{F}} P\left(x_t = s_t^{wt} f | x_{\setminus T}\right) \right] \tag{2}$$

Here, $f \in \mathcal{F}$ is the 3Di token generated by Foldseek and $s_tf \in \mathcal{V} \times \mathcal{F}$ is the SA token in our new alphabet.

Zero-shot mutational effect prediction tasks and datasets

ProteinGym. ProteinGym[37] comprises an extensive collection of deep mutational scanning assays, enabling comprehensive comparison among zero-shot predictors. We evaluate all baseline models (Supplementary Table 7) on ProteinGym's substitution branch, using its provided protein structures and adhering to the standard evaluation protocol outlined in the original paper[37]. We use Spearman's rank correlation as our evaluation metric.

ClinVar. ClinVar[38] is a publicly accessible repository housing information about human genetic variants and their clinical importance in disease. For our analysis, we use data curated from EVE[34], excluding proteins exceeding 1,024 residues in length. To ensure data quality, we restrict our analysis to proteins with reliability ratings of one 'gold star' or higher. Following EVE's methodology, we assess model performance using the area under the curve.

Mega-scale. Mega-scale[36] uses complementary DNA display proteolysis to measure thermodynamic folding stability across protein domains. The dataset encompasses all single mutations and selected double mutants in both natural and de novo designed proteins. For this dataset, we also use Spearman's rank correlation as our evaluation metric.

For each mutation dataset, we provide all variants with the wild-type structure, as AF2 does not reliably distinguish the structural changes induced by single substitutions. Additionally, the ClinVar dataset only provides UniProt IDs; thus, we manually downloaded all AF2 structures and eliminated proteins without structures in AlphaFold DB. Both ProteinGym and Mega-scale datasets provide protein structures, either predicted from AF2 or derived from de novo design.

Supervised fine-tuning tasks and datasets

Fine-tuning Saprot with the AF2 structure. These benchmarks include the Thermostability task from FLIP[64] and the localization prediction task from DeepLoc[65]. The DeepLoc benchmark has two prediction branches: a multiclass classification with ten subcellular locations and a binary classification with two location categories. We use the datasets provided by these original literature. For structural information, we obtain AlphaFold2-predicted structures for all proteins using their corresponding UniProt IDs.

Fine-tuning Saprot with the PDB structure. There are a few tasks providing experimentally determined structures as training data. We evaluate the metal-ion-binding task[66] and a series of tasks from ProteinShake[31], including structure class prediction, structural similarity prediction and binding site detection. The corresponding datasets are provided in the respective literature.

Fine-tuning Saprot without structure. While Saprot is designed to leverage protein structural information, it can still work in scenarios where structural data are not provided during supervised fine-tuning. In these cases, we mask the 3Di token in the SA sequence. We evaluate performance on the fluorescence and stability prediction datasets from TAPE[30], the AAV dataset from FLIP[64] and the β-lactamase landscape prediction dataset from PEER[32].

Regarding the evaluation metrics, we adopt those established in the original literature. Details on dataset splits are described below.

Data split

In the existing literature, datasets are typically partitioned on the basis of protein sequence identity. However, a recent benchmark study called ProteinShake[31] argued that protein structures exhibit higher conservation than sequences, indicating that structure-based splitting provides a more stringent evaluation of model generalization. Inspired by this, we adopted the same structure-based data splitting for most of our evaluation (unless otherwise specified). The splitting criterion is quantified using the LDDT, as proposed in ProteinShake. To be specific, for datasets that include protein structures, we use the default 70% LDDT threshold recommended in ProteinShake[31]. For tasks where only sequence data are available, we retain the original splits provided in the official literature, as these datasets consist of only mutational variants of one single protein. To show performance sensitivity across different splitting criteria, we perform further evaluation by comparing Saprot (35M) and ESM (35M) using a more stringent 30% LDDT threshold and the commonly used 30% sequence identity threshold[32,67]. The corresponding results are presented in Supplementary Table 3.

Protein inverse folding

Saprot for protein inverse folding. To use Saprot for inverse folding, we first encode the protein backbone into 3Di tokens ($f_1, f_2, \ldots, f_n$). We then mask all residue parts of the SA tokens, forming an SA sequence ($\#f_1, \#f_2, \ldots, \#f_n$). This sequence is input into Saprot to predict residue distributions at all positions. In contrast to ProteinMPNN[35] that generates residues in an autoregressive manner (that is, generating next token conditioned on all previous outputs), Saprot is able to simultaneously predict all residues with only one forward propagation.

Fine-tuning Saprot on the CATH dataset. We evaluate two variants of Saprot for protein design: one pretrained using AF2-predicted structures and another further fine-tuned on the CATH dataset[68]. This CATH database, which was also used to train ProteinMPNN[35], is partitioned into training, validation and test sets using an 80:10:10 split, as detailed by Ingraham et al.[69].

Hyperparameters for fine-tuning tasks

We use the AdamW[61] optimizer during fine-tuning, setting $\beta_1 = 0.9$ and $\beta_2 = 0.98$, along with an L_2 weight decay of 0.01. We use a batch size of 64 for all datasets. We empirically found that the optimal learning rate for most baselines are in the range of 1×10^{-5} to 5×10^{-5}. For training with AF2 structure and training with PDB structure, the optimal learning rate was set to 5×10^{-5}, whereas, for training without structure, it was set to 1×10^{-5}. We fine-tuned all model parameters until convergence and selected the best checkpoints based on their performance on the validation set.

Adapter learning

The adapter learning technique, originated in general AI community[70–72], has recently been adopted in protein research[54,56,73,74]. While these studies mainly focused on its advantage for model accuracy and training efficiency, we use adapter here by integrating it with Google Colab to create a platform that enables model fine-tuning, sharing, continuous retraining and collaboration within the biological research community through our online SaprotHub.

In the broader AI community, various types of adapters exist, including Houlsby[70], Pfeiffer[17], Compacter[75] and LoRA[16]. We choose LoRA for its capacity to deliver comparable results with fewer parameters. By integrating learnable low-rank matrices into each Saprot block while freezing the backbone, LoRA enables parameter-efficient fine-tuning and model sharing for these downstream tasks.

Community-wide collaboration

As the SaprotHub community grows, more fine-tuned Saprot models become available for each protein function, facilitating collaborative development among biologists. To evaluate this advantage (Fig. 2e), we adopt the following approach. For each task, we randomly partition the training data into five subsets and train one model on each subset, yielding models model_0 through model_4, simulating models shared by different researchers. For regression tasks (for example, thermostability), the final prediction (model_agg) is computed as the mean

of all model outputs. For classification tasks (for example, subcellular localization), we implement majority voting, where the final prediction (model_agg) is determined by the most frequent class prediction across the ensemble. The lightweight adapter technique is crucial for model aggregation, addressing the challenges of loading and sharing multiple large pretrained models.

To evaluate the effectiveness of model continue learning (Fig. 2f), we randomly sampled 100, 200 and 500 training instances per task to simulate researchers' private data. We then performed fine-tuning using two base models, updating only their adapter parameters: the official Saprot model (blue) and a shared model from SaprotHub (orange). The latter represents models previously trained by researchers with larger or higher-quality datasets. Results demonstrate that continuing training from existing well-trained models substantially outperforms training from scratch with limited data. Note that the official Saprot model (blue) is pretrained on large-scale protein sequence and structure data but has not undergone supervised fine-tuning for specific tasks. This highlights a key advantage of SaprotHub, whereby researchers can build upon others' achievements.

ColabSaprot notebooks

ColabSaprot consists of three key components, with the first focusing on model training. This component enables researchers to rapidly configure the runtime environment, process training data and fine-tune Saprot efficiently. We provide access to four base models: Saprot 35M, Saprot 650M, custom fine-tuned Saprot models and community-shared fine-tuned models from SaprotHub. Additionally, we offer advanced hyperparameter customization options, allowing researchers to tailor their training strategies to specific requirements. These customizable parameters include batch size, learning rate, training steps, etc.

The second component focuses on prediction capabilities. ColabSaprot supports a diverse range of prediction tasks (Supplementary Table 1) using both community-shared models from SaprotHub and locally fine-tuned models. This includes protein-level, residue-level and protein–protein interaction predictions, as well as zero-shot mutational effect predictions and protein design. Each task category offers multiple configuration options to accommodate different research requirements. Moreover, ColabSaprot enhances prediction accuracy by implementing ensemble methods that aggregate multiple models from SaprotHub, delivering more robust and reliable results. This collaborative approach facilitates a community-driven model ecosystem where researchers can leverage and combine multiple models to achieve superior performance.

The third component enables model sharing, searching (implemented in SaprotHub; Supplementary Fig. 5) and community collaboration. Upon completion of training, researchers can contribute their model weights (specifically the adapter weights) to the SaprotHub community repository, making them accessible to the broader biological research community. Through SaprotHub's specialized search engine, researchers can efficiently locate and use relevant models. These shared models can be leveraged for continual learning, direct application or model aggregation to achieve enhanced performance.

User study

We evaluated ColabSaprot-v1 through eight protein prediction tasks with 12 participants who, on the basis of brief conversations and background checks, had biological backgrounds but no prior involvement in coding or ML projects. The evaluation encompassed supervised fine-tuning, zero-shot mutation effect prediction and protein inverse folding tasks. Each participant had 3 days to complete their assignments and received compensation upon submitting either their results or documentation of encountered obstacles through screenshots or recordings. Additional information is provided online (https://drive.google.com/file/d/1LdGRnwt2lttnszNBAJ0F967A8rguPq8b/view?usp=sharing).

For comparison, we engaged an AI expert—a third-year PhD student specializing in ML with over 2 years of research experience in protein-focused AI applications. The expert executed these tasks using the Saprot codebase from our GitHub repository, conducting five independent runs with optimized hyperparameters using different random seeds.

The 12 participants were organized as follows: one assigned to zero-shot mutational effect prediction, another assigned to protein design and the remaining ten randomly divided into two groups of five. Each group handled three supervised fine-tuning tasks, covering six tasks in total. In other words, each supervised fine-tuning task was completed by five participants. We evaluated the average accuracy across all participants, including those who were unable to finish the task completely (when such cases occurred).

For the supervised fine-tuning tasks, we used five public datasets for predicting: thermostability, subcellular and binary localization, GFP fluorescence and stability (Supplementary Table 6). We also included a proprietary eYFP fitness prediction dataset from the X. Zheng lab, containing 3,087 validation and 3,088 test samples, bringing the total to six supervised fine-tuning tasks. To reduce training time and computing power consumption, we randomly selected 1,000 samples from the training set of each dataset while keeping the validation and test sets unchanged.

For model evaluation, all participants generated predictions on designated test sets. In the eYFP task, biology participants leveraged continual learning on a pretrained model from SaprotHub (Model-EYFP-650M (search model name through https://huggingface.co/spaces/SaProtHub/SaprotHub-search), achieving 0.95 Spearman's ρ on test data), while the AI expert used the base Saprot 650M model. This setup is used to show how SaprotHub enables researchers with limited data to build upon existing fine-tuned models. For other tasks, both biology participants and the AI expert trained their models using the same SaProt_650M_AF2 base model with identical training and test sets, ensuring fair comparison.

For the zero-shot mutational effect prediction, we randomly selected four mutation datasets from ProteinGym benchmark[37] for evaluation. Three of these datasets focus on the impact of mutations on enzyme activity, while the fourth addresses drug resistance. We assigned one participant to conduct predictions on these datasets using ColabSaprot in a zero-shot manner.

For the protein inverse folding task, we assigned one participant to use ColabSaprot to generate protein sequences based on given structures. Subsequently, the participant used ESMFold (an interface provided by ColabSaprot) to predict the structures of the generated sequences. To assess Saprot's ability on new proteins, we selected these recently released structures (Fig. 2h).

Participants were provided with related protein datasets (including training and test sets), GPU-enabled Google Colab accounts and detailed task instructions. Participants had 3 days to complete the assignments on their own using ColabSaprot. To encourage honest feedback and thorough documentation, compensation was guaranteed to all participants who documented their challenges through screenshots or recordings, regardless of whether they completed the assigned tasks.

We acknowledge a potential self-selection bias, as the biology participants were volunteers likely interested in novel computational tools. This may imply that their aptitude for learning new software could be higher than the general average for the biology community. However, as the study's primary goal was a comparative analysis of workflows, this bias is not expected to alter the main conclusions regarding the platform's relative efficiency and accessibility. All participants were fully informed about the study.

Experimental validation on the xylanase

There are several steps. Let's name the protein (XP_069217686.1) Mth.

(1) Use AlphaFold3 (https://golgi.sandbox.google.com/) to get the protein structure of Mth.
(2) Use Foldseek or ColabSaprot to get the SA sequence of Mth.
(3) Use ColabSaprot (650M) to perform zero-shot (single-point) mutation effect prediction for the entire SA sequence.
(4) Choose the 20 highest-ranked variants (Supplementary Table 12) excluding A15G and I13P; A15G and I13P were not selected because they are located in the signal peptide region of the protein, which would be removed during the process of protein secretion in *P. pastoris*.
(5) Construct mutants as detailed below.
    (51) The gene sequence of Mth was optimized according to the codon preference of *P. pastoris*, and the plasmid pPIC9K-Mth was synthesized by GenScript.
    (52) Mutations of Mth were generated through site-directed mutagenesis by PCR, using the plasmid pPIC9K-Mth as a template. The primers are listed in Supplementary Table 13.
    (53) After confirmation by DNA sequencing, the wild-type and mutated plasmids were linearized with SalI and used to transform *P. pastoris* strain GS115. The recombinant strains were selected on MD plates (1.34% YNB (yeast nitrogen base without AAs), 2% glucose and 2% agar) and verified by PCR and sequencing.
(6) Conduct enzyme activity assay of wild-type Mth and mutants as detailed below.
    (61) The positive transformants were cultivated in liquid medium BMGY (1% yeast extract, 2% peptone, 1.34% YNB, 1% glycerol and 100 mM potassium phosphate pH 6.0) for 20–24 h. The cells were collected by centrifugation at 3,500*g* and 4 °C for 5 min and then transferred to 250-ml shake flasks containing 25 ml of BMMY (1% yeast extract, 2% peptone, 1.34% YNB, 1% methanol and 100 mM potassium phosphate pH 6.0) with initial optical density at 600 nm (OD_{600}) of 0.5. Fed-batch fermentation was proceeded to express xylanase by feeding 1% methanol per 24 h. All liquid cultures were performed at 30 °C and 250 rpm. After 120 h of cultivation, the supernatants were obtained by centrifugation at 3,500*g* and 4 °C for 5 min and tested for xylanase activity.
    (62) Xylanase activity assay: The reaction mixture contained 0.1 ml of 1% (w/v) beechwood xylan and 0.1 ml of a suitably diluted enzyme solution (100 mM acetate buffer pH 5.0) incubated at 60 °C for 30 min. The amount of reducing sugar released was determined using the 3,5-dinitrosalicylic acid method, with xylose as the standard. Here, 1 U of xylanase activity was defined as the amount of enzyme that catalyzes the release of 1 µmol of xylose equivalent per min under the assay conditions. The enzyme activity was measured at 40–70 °C to determine the optimal temperature of the enzymes, with the pH of the reaction maintained at 5.0. After incubation at 60 °C for various time periods, the residual enzyme activity was measured to assess the thermostability of the enzymes.

Experimental validation on GFP variants

The objective of this task was to engineer brighter avGFP variants as part of the 2024 Critical Assessment of Protein Engineering (CAPE) competition[76]. CAPE, a student-focused challenge modeled after the Critical Assessment of Structure Prediction competition, emphasizes protein function design and variant effect prediction. The parent GFP sequence is based on avGFP (AA sequence information in Supplementary Table 14) derived from Aequorea victoria (UniProt P42212). These top-ranked predicted variants by Saprot were experimentally validated by the CAPE organizers.

The process of fine-tuning Saprot (35M) and subsequent prediction consisted of the following steps:

Step 1: Data preparation. The CAPE organizers provided a dataset of 140,000 GFP variants (including avGFP, amacGFP, cgreGFP and ppluGFP2; Supplementary Table 14), along with their corresponding fitness scores, and the structures of four wild-type GFP proteins (avGFP, amacGFP, cgreGFP and ppluGFP2). Our OPMC member X. Zhang used Foldseek to generate 3Di tokens for these wild-type structures, which were then used to converted to SA tokens by their corresponding variants.
Step 2: Model fine-tuning. The Saprot model underwent full parameter fine-tuning using the SA token sequences of these variants.
Step 3: Variant prediction and validation. A pool of 5 million avGFP double-site mutants was generated through random mutagenesis. The fine-tuned Saprot model from Step 2 was used to predict their fitness scores. The top nine variants were selected for experimental validation.

Experimental validation was conducted using the robotic biofoundries at the Shenzhen Infrastructure for Synthetic Biology according to the following procedure:

Step 1: Expression plasmid pET28a vectors containing designed mutant GFP sequences were ordered from Genescript Biotech and used to transform chemically competent *Escherichia coli* BL21(DE3) cells through heat shock at 42 °C. Two independent clones for each mutant were randomly selected to inoculate 1 ml of noninducing Luria–Bertani medium supplemented with 50 µg ml^{-1} kanamycin (LB+Kan) for plasmid maintenance by antibiotic selection in 96-well microtiter plates to prepare seed cultures, which were grown at 37 °C for 16–20 h.
Step 2: For inducible protein expression, 40 µl of stationary-phase seed cultures were used to reinoculate 4 ml of fresh LB+Kan medium in 24-well microtiter plates, followed by approximately 4 h of incubation at 37 °C to reach the exponential growth phase (OD_{600} = 0.6–0.8). Then, IPTG was added to achieve a final concentration of 1 mM for inducible expression at 18 °C for 20 h.
Step 3: Biomass growth was monitored by using OD_{600} measurement and GFP fluorescence was assessed with excitation at 488 nm and emission at 520 nm in 96-well flat-bottom plates. For each strain, GFP signal intensities were divided by OD_{600} values to calculate biomass-normalized fluorescence.

Experimental validation of the TDG variants

The X.C. lab used ColabSaprot version 1 (650M) for zero-shot prediction of single-site mutation effects of TDG (AA sequence in Supplementary Table 14). They input only the AA sequence with all structural tokens masked into ColabSaprot to identify the top 20 highest-ranked variants. Experimental validation and results of these variants are documented in Supplementary Table 15.

Following protocols described previously[51,77], base editors used in this paper were cloned into a pCMV plasmid with blasticidin resistance. Single guide RNA (sgRNA) was cloned into a pSuper-sgRNA plasmid with puromycin resistance. The TDG sequence was amplified from plasmid TSBE2 (ref. 51) and TDG variants were generated through site-directed mutagenesis by PCR and then fused with SpCas9 (D10A) protein as TSBE2. All primers are listed in Supplementary Table 13. The protein sequence of TSBE2 and protospacer sequences of sgRNA are available from a previous study[51]. The experimental validation process was performed as previously described[51,77].

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The pretraining dataset for training Saprot is available online (https://huggingface.co/datasets/westlake-repl/AF2_UniRef50. Downstream task datasets are all stored online (https://huggingface.co/SaProtHub). Materials for user study (https://drive.google.com/file/d/1LdGRnwt2ltnszNBAJ0F967A8rguPq8b/view?usp=sharing), raw data with detailed wet lab information (https://drive.google.com/file/d/1IYcOqRuF76LusG7DMI4kEr2mJL6aE7g-/view?usp=sharing) and additional wet lab experimental results collected from the research community (https://drive.google.com/file/d/1ZcDl0XYksTcUEaCfPI0tuEVo5CA031dg/view?usp=sharing) are available from a Google Drive. All unique/stable reagents generated in this study are available from the lead contact (yuanfajie@westlake.edu.cn).

Code availability

Saprot is an open-sourced model with MIT license. The code is available from GitHub (https://github.com/westlake-repl/Saprot). The code implementation of ColabSaprot notebook is also available from GitHub (https://github.com/westlake-repl/SaProtHub). ColabSaprot service (latest version: version 2) is available online (https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/SaprotHub_v2.ipynb?hl=en; the previous version is still maintained on the SaprotHub GitHub). All fine-tuned Saprot models can be obtained through SaprotHub (https://huggingface.co/SaProtHub) through the dedicated search engine (https://huggingface.co/spaces/SaProtHub/SaprotHub-search). Our OPMC members have also implemented ColabSeprot (https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/ColabSeprot.ipynb?hl=en), including ColabProTrek (35M and 650M), ColabESM1b (650M), ColabESM2(35M, 150M and 650M), ColabProtBert (420M), SeprotHub (https://huggingface.co/SeprotHub) and an independent ColabProtT5 (https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/ColabProtT5.ipynb).

References

59. van den Oord, A. et al. Neural discrete representation learning. In *Proc. 30th International Conference on Neural Information Processing Systems* (eds Lee, D. D., von Luxburg, U., Garnett, R., Sugiyama, M. & Guyon, I.) (NIPS, 2017).
60. Gong, L. et al. Efficient training of BERT by progressively stacking. In *Proc. 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) (PMLR, 2019).
61. Loshchilov, I. & Hutter, F. Fixing weight decay regularization in Adam. Preprint at *OpenReview* https://openreview.net/forum?id=rk6qdGgCZ (2018).
62. Yang, K. K., Zanichelli, N. & Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *Protein Eng. Des. Sel.* **36**, gzad015 (2023).
63. Zhang, Z. et al. Protein representation learning by geometric structure pretraining. In *First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022* https://openreview.net/pdf?id=V5MEFikiBQy (2023).
64. Dallago, C. et al. FLIP: benchmark tasks in fitness landscape inference for proteins. In *Proc. Neural Information Processing Systems Track on Datasets and Benchmarks* https://openreview.net/pdf?id=p2dMLEwL8tF (2021).
65. Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H. & Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics* **33**, 3387–3395 (2017).
66. Hu, M. et al. Exploring evolution-aware & -free protein language models as protein function predictors. In *Proc. 35th International Conference on Neural Information Processing Systems* (eds Oh, A. et al.) (NIPS, 2023).
67. Gligorijević, V. et al. Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**, 3168 (2021).
68. Orengo, C. A. et al. CATH—a hierarchic classification of protein domain structures. *Structure* **5**, 1093–1109 (1997).
69. Ingraham, J., Garg, V., Barzilay, R. & Jaakkola, T. Generative models for graph-based protein design. In *Proc. 32nd International Conference on Neural Information Processing Systems* (eds Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K. & Cesa-Bianchi, N.) (NIPS, 2018).
70. Houlsby, N. et al. Parameter-efficient transfer learning for NLP. In *Proc. 36th International Conference on Machine Learning* (eds Chaudhuri, K. & Salakhutdinov, R.) (PMLR, 2019).
71. Fu, J. et al. Exploring adapter-based transfer learning for recommender systems: empirical studies and practical insights. In *Proc. 17th ACM International Conference on Web Search and Data Mining* (eds Angélica, L., Lattanzi, S. & Muñoz Medina, A.) (ACM, 2024).
72. Yuan, F., He, X., Karatzoglou, A. & Zhang, L. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proc. 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (eds Huang, J., Chang, Y. & Cheng, X.) (ACM, 2020).
73. Schreiber, A. ESMBind and QBind: LoRA, QLoRA, and ESM-2 for predicting binding sites and post translational modification. Preprint at *bioRxiv* https://doi.org/10.1101/2023.11.13.566930 (2023).
74. Schmirler, R., Heinzinger, M. & Rost, B. Fine-tuning protein language models boosts predictions across diverse tasks. *Nat. Commun.* **15**, 7407 (2024).
75. Karimi Mahabadi, R., Henderson, J. & Ruder, S. COMPACTER: efficient low-rank hypercomplex adapter layers. In *Proc. 35th International Conference on Neural Information Processing Systems* (eds Oh, A. et al.) (NIPS, 2023).
76. Fu, L. et al. Critical Assessment of Protein Engineering (CAPE): a student challenge on the cloud. *ACS Synth. Biol.* **13**, 3782–3787 (2024).
77. He, Y., Zhou, X., Yuan, F. & Chang, X. Protocol to use protein language models predicting and following experimental validation of function-enhancing variants of thymine-*N*-glycosylase. *STAR Protoc.* **5**, 103188 (2024).

Acknowledgements

We thank N. Li and the Westlake University HPC Center for computing resources and J. Huang, J. Zeng, D. Li, L. Cao and D. Li for discussions and paper advice. We thank J. M. Jumper for providing insights into AF2 Evoformer and advice on the SA token design. This work was supported by the National Key Research and Development Program of China (2022ZD0115100), the National Natural Science Foundation of China (U21A20427), the National Research Foundation of Korea (RS-2020-NR049543, RS-2021-NR061659, RS-2021-NR056571 and RS-2024-00396026), the Creative-Pioneering Researchers Program and Novo Nordisk Foundation (NNF24SA0092560), the Westlake Center of Synthetic Biology and Integrated Bioengineering, the Zhejiang Province Leading Geese Plan (2025C01094), the Zhejiang Key Laboratory of Low-Carbon Intelligent Synthetic Biology (2024ZY01025) and the Research Center for Industries of the Future (WU2022C030).

Author contributions

F.Y. conceptualized and led this research. J.S. conducted the main research and managed the technical implementation. J. Su, F.Y. and J. Shan designed the SA token and Saprot. J. Su and Z.L. developed the ColabSaprot and SaprotHub. T.T. developed the ColabSeprot and SeprotHub. F.Y., J. Su and S.O. designed the core idea of ColabSaprot and SaprotHub. S.O. and M.S. participated in discussions

of ColabSaprot and SaprotHub, provided constructive ideas and revised the manuscript. C.H. conducted the partial zero-shot mutation experiments. F.D. trained the Saprot 1.3B version. Y.Z. collected the AlphaFold DB dataset. Y.H., X.C. and X.Z. (TDG), Q.Y. (xylanase) and Y.G. and T.S. (GFP) conducted the wet lab experiments. S.J. conducted some experimental result analysis. D.M. participated in early wet lab experiments. J. Su and F.Y. wrote the manuscript. Other OPMC authors contributed to wet lab validation, development of ColabSaprot and ColabSeprot, manuscript writing and proofreading, idea discussions, expert advice and promoting and testing ColabSaprot and ColabSeprot.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-025-02859-7.

Correspondence and requests for materials should be addressed to Fajie Yuan.

Peer review information *Nature Biotechnology* thanks Feiran Li and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Fajie Yuan

Last updated by author(s): Sep 4, 2025

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	☐	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	☐	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	☐	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	The pre-training dataset for training Saprot is available at https://huggingface.co/datasets/westlake-repl/AF2_UniRef50. Downstream taskdatasets are all stored at https://huggingface.co/SaProtHub. User study datasets are available at https://drive.google.com/file/d/1LdGRnwt2lttnszNBAJ0F967A8rguPq8b/view?usp=sharing. The CATH dataset is available at https://www.cathdb.info/. The pre-training datasets were collected from AlphaFold DB: https://alphafold.ebi.ac.uk/ The downstream task datasets were collected from exiting literature, as cited in the manuscript.
Data analysis	All python-generated figures were made using matplotlib==3.9.1 and the embedding visualizations were made using scikit-learn==1.4.0. Protein structures were encoded using Foldseek with the version ef4e960ab84fc502665eb7b84573dfff9c2aa89d and TM-scores were calculated using TMalign with the version 20220412.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

Code and service availability
Saprot is an open-sourced model with MIT license. The code is available at https://github.com/westlake-repl/Saprot. The code implementation of ColabSaprot notebook is available at https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/SaprotHub_v2.ipynb?hl=en (the previous v1 version is still maintained in our SaprotHub github webpage). All fine-tuned Saprot models can be obtained through SaprotHub https://huggingface.co/SaProtHub by the dedicated search engine https://huggingface.co/spaces/SaProtHub/SaprotHub-serach.

Our OPMC members have also implemented ColabSeprot (https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/ColabSeprot.ipynb?hl=en), including ColabProTrek (35M, 650M), ColabESM1b (650M), ColabESM2 (35M, 150M, 650M) and ColabProtBert (420M) and SeprotHub (https://huggingface.co/SeprotHub) and an independent ColabProtT5 (https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/ColabProtT5.ipynb). Our OPMC members will continue to integrate newly developed and state-of-the-art PLMs in the future.

Data Availability
The pre-training dataset for training Saprot is available at https://huggingface.co/datasets/westlake-repl/AF2_UniRef50. Downstream task datasets are all stored at https://huggingface.co/SaProtHub.Materials for user study are available at https://drive.google.com/file/d/1LdGRnwt2lttnszNBAJ0F967A8rguPq8b/view?usp=sharing. The raw data with detailed wet lab information is available at https://drive.google.com/file/d/1IYcOqRuF76LusG7DMI4kEr2mJL6aE7g-/view?usp=sharing. Additional wet lab experimental results collected from the research community are available at https://drive.google.com/file/d/1ZcDl0XYksTcUEaCfPI0tuEVo5CA031dg/view?usp=sharing. All unique/stable reagents generated in this study are available from the lead contact yuanfajie@westlake.edu.cn.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Not applicable
Population characteristics	Not applicable
Recruitment	We recruited 12 college students majoring in biology, all of whom had no prior experience in coding or machine learning, compensating each with 500 Yuan for the user study. Relative statements are described in Methods Section 1.15.
Ethics oversight	Not applicable

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Saprot and ColabSaprot were evaluated on standard benchmark datasets that are publicly available, so we do not consider the sample size in this case.
Data exclusions	For downstream tasks, we only remove proteins without structures in AlphaFoldDB.
Replication	All experiments were conducted with a fixed random seed to ensure the reproducibility.
Randomization	For user study, the tasks assigned to each participant were randomized.
Blinding	Not applicable

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	
Research sample	
Sampling strategy	
Data collection	
Timing	
Data exclusions	
Non-participation	
Randomization	

Ecological, evolutionary & environmental sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	
Research sample	
Sampling strategy	
Data collection	
Timing and spatial scale	
Data exclusions	
Reproducibility	
Randomization	
Blinding	

Did the study involve field work? ☐ Yes ☐ No

Field work, collection and transport

Field conditions	
Location	
Access & import/export	
Disturbance	

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Antibodies

Antibodies used

Validation

Eukaryotic cell lines

Policy information about cell lines and Sex and Gender in Research

Cell line source(s)

Authentication

Mycoplasma contamination

Commonly misidentified lines
(See ICLAC register)

Palaeontology and Archaeology

Specimen provenance

Specimen deposition

Dating methods

☐ Tick this box to confirm that the raw and calibrated dates are available in the paper or in Supplementary Information.

Ethics oversight

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals

Wild animals

Reporting on sex

Field-collected samples

Ethics oversight

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

Clinical trial registration	
Study protocol	
Data collection	
Outcomes	

Dual use research of concern

Policy information about dual use research of concern

Hazards

Could the accidental, deliberate or reckless misuse of agents or technologies generated in the work, or the application of information presented in the manuscript, pose a threat to:

No	Yes	
☐	☐	Public health
☐	☐	National security
☐	☐	Crops and/or livestock
☐	☐	Ecosystems
☐	☐	Any other significant area

Experiments of concern

Does the work involve any of these experiments of concern:

No	Yes	
☐	☐	Demonstrate how to render a vaccine ineffective
☐	☐	Confer resistance to therapeutically useful antibiotics or antiviral agents
☐	☐	Enhance the virulence of a pathogen or render a nonpathogen virulent
☐	☐	Increase transmissibility of a pathogen
☐	☐	Alter the host range of a pathogen
☐	☐	Enable evasion of diagnostic/detection modalities
☐	☐	Enable the weaponization of a biological agent or toxin
☐	☐	Any other potentially harmful combination of experiments and agents

ChIP-seq

Data deposition

☐ Confirm that both raw and final processed data have been deposited in a public database such as GEO.

☐ Confirm that you have deposited or provided access to graph files (e.g. BED files) for the called peaks.

Data access links *May remain private before publication.*	
Files in database submission	
Genome browser session (e.g. UCSC)	

Methodology

Replicates	
Sequencing depth	

Antibodies

Peak calling parameters

Data quality

Software

Flow Cytometry

Plots

Confirm that:

☐ The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).

☐ The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).

☐ All plots are contour plots with outliers or pseudocolor plots.

☐ A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation

Instrument

Software

Cell population abundance

Gating strategy

☐ Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.

Magnetic resonance imaging

Experimental design

Design type

Design specifications

Behavioral performance measures

Acquisition

Imaging type(s)

Field strength

Sequence & imaging parameters

Area of acquisition

Diffusion MRI ☐ Used ☐ Not used

Preprocessing

Preprocessing software

Normalization

Normalization template

Noise and artifact removal

Volume censoring

Statistical modeling & inference

Model type and settings	
Effect(s) tested	

Specify type of analysis: ☐ Whole brain ☐ ROI-based ☐ Both

Statistic type for inference (See Eklund et al. 2016)	
Correction	

Models & analysis

n/a	Involved in the study
☐	☐ Functional and/or effective connectivity
☐	☐ Graph analysis
☐	☐ Multivariate modeling or predictive analysis

Functional and/or effective connectivity	
Graph analysis	
Multivariate modeling and predictive analysis	