Leveraging Multi-modal Representations to Predict Protein Melting Temperatures

Daiheng Zhang[1], **Yan Zeng**[1], **Xinyu Hong**[1], **Jinbo Xu**[1]

[1]MoleculeMind Ltd., Beijing, China
dz367@rutgers.edu, zy1104865009@gmail.com, xinyuhong@moleculemind.com, jinboxu@gmail.com

Abstract

Accurately predicting protein melting temperatures (ΔT_m) is fundamental for assessing protein stability and guiding protein engineering. Leveraging multimodal protein representations has shown great promise in capturing the complex relationships among protein sequences, structures, and functions. In this study, we develop models based on powerful protein language models—including ESM2, ESM3, SaProt, and AlphaFold—using various feature extraction methods to enhance prediction accuracy. By utilizing the ESM3 model, we achieve a new state-of-the-art performance on the s571 test dataset, obtaining a Pearson correlation coefficient (PCC) of 0.50. Furthermore, we conduct a fair evaluation to compare the performance of different protein language models in the ΔT_m prediction task. Our results demonstrate the strength of integrating multimodal protein representations could advance the prediction of protein melting temperatures.

Introduction

Proteins play a pivotal role in various biological applications, such as catalyzing biochemical reactions, immune function, and metabolism regulation. Composed of sequences built from 20 different classes of amino acids, proteins fold into complex structures—both sequence and structure determine their functions (Whisstock and Lesk 2003). Therefore, exploring appropriate protein representations is crucial for related tasks. Large-scale protein language models (PLMs) have demonstrated excellent performance in protein representation capabilities (Bepler and Berger 2021; Lin et al. 2022; Hayes et al. 2024; Su et al. 2023). The pretraining strategies enhance the models' ability to capture nuanced features and patterns in protein sequences and structures, effectively transferring to downstream tasks like understanding protein fitness (Ouyang-Zhang et al. 2024; Chen et al. 2024) and evolutionary dynamics (Hie et al. 2024).

With stabilized structures, downstream engineering of proteins becomes more feasible. Mutations are commonly used in protein engineering to study and improve protein structure and function, making the accurate quantification of mutation effects crucial for studying the evolutionary fitness of proteins (Pandurangan and Blundell 2020). Thermodynamic stability (Pires, Ascher, and Blundell 2014; Umerenkov et al. 2022; Benevenuta et al. 2021; Pandurangan and Blundell 2020) and enzyme kinetic parameters (Li et al. 2022; Yu et al. 2023) are widely explored in mutation-related tasks. Benefiting from deep mutational scanning (DMS) databases containing protein fitness data (Fowler and Fields 2014; Tsuboyama et al. 2023), thermodynamic stability ($\Delta\Delta G$) prediction has been extensively studied, and its performance has greatly improved. However, the prediction of changes in melting temperature (ΔT_m) has been less explored compared to $\Delta\Delta G$ prediction. Deep learning-based methods are largely absent in addressing this problem (Xu, Liu, and Gong 2023; Masso and Vaisman 2014, 2008; Pucci, Bourgeas, and Rooman 2016), partly due to a lack of experimental data and partly because the issue has not received significant attention.

In this paper, we propose a new prediction framework, *ESM3-DTm*, for ΔT_m. We fine-tune three distinct protein language models—ESM2 (Lin et al. 2022), ESM3 (Hayes et al. 2024), and SaProt (Su et al. 2023)—and also explore using OpenFold (Ahdritz et al. 2024) to extract features by incorporating different regression heads into the architecture. Among these approaches, we found that using *ESM3-DTm* accepting both sequence and structure to obtain embeddings yielded the best results, achieving state-of-the-art (SOTA) performance with a Pearson correlation coefficient (PCC) of 0.50, mean absolute error (MAE) of 5.21, and root mean square error (RMSE) of 7.68. We also demonstrate the impact of different finetuning methods on the results.

Preliminary

Problem Setup

A protein $P = (a_1, a_2, \ldots, a_L)$ is a sequence of amino acids, where each $a_i \in AA$, and $AA = A, C, \ldots, Y$ represents the 20 standard amino acid types. Let $\mu = (w, m)$ denote a mutation that substitutes the amino acid at position w in P with amino acid type $m \in AA$. Our goal is to predict the change in melting temperature $\Delta T_m \in \mathbb{R}$ for the protein P resulting from the mutation μ.

Protein large language models

Over recent years, large language models have played an ever more significant role in protein research, providing innovative insights and enhanced abilities for comprehend-

ing and modifying proteins(Zhang et al. 2024). Most of them are encoder-only models, which are built upon the encoder of Transformer, enables the encoding of protein sequences or structures into fixed-length vector representations. From this series of mainstream models, we selected ESM2 and SaProt to further explore their representation capabilities. ESM2 is one of the largest architectures among single sequence models and stands out for its role in structure prediction. We adopt the architecture with 640 million parameters and 36 layers. SaProt is a bilingual protein language model featuring structure-aware embeddings, undergoes training on Foldseek's 3Di structures(van Kempen et al. 2022) and amino acid sequences. We used the same settings for SaProt as we did for ESM2. In addition to encoder-only models, we also explored encoder-decoder models. The advantage of having a decoder is that it provides the model with strong generative capabilities. Here, we investigated ESM3, a newly released co-design multimodal model. We used the publicly available version with 1.4 billion parameters. Apart from these language models, AlphaFold(Jumper et al. 2021) has shown its highly effective in predicting protein structures from sequences by leveraging evolutionary information through multiple sequence alignments (MSA). It can also be used for feature extraction. Therefore, we also utilized OpenFold as a backbone for further experiments.

Data

To compare our method with existing models, we use the training and test datasets proposed by **GeoStab**. The training set, s4346, comprises 4,346 single-point mutations across 349 proteins, collected from ProThermDB (Gromiha et al. 1999, 2000, 2002; Kumar et al. 2006) and ThermoMutDB (Xavier et al. 2021), both of which are dedicated ΔT_m databases. The test set, s571, consists of 571 single-point mutations across 37 proteins, also collected from the same sources.

We observed that the baseline method lacks a train/-validation split within the training set, which can easily lead to overfitting. To address this issue, we used MMseqs2(Steinegger and Söding 2017) at 50% sequence identity and then split it into training and validation sets in an 8:2 ratio. Hyperparameter tuning was conducted using this split. After identifying the best hyperparameters, we retrained the model on the combined training and validation sets, aligning with the train-test split setting of the previous baseline **GeoStab**.

The original dataset contains only sequence data. For input to the OpenFold backbone, multiple sequence alignments (MSAs) are required; we computed these using ColabFold (Mirdita et al. 2022). For the ESM3 backbone with PDB input option and for the SaProt backbone input, PDB structures are needed. Our PDB structure dataset consists of two parts: for proteins with available PDB IDs, we retrieved the corresponding structures from the Protein Data Bank (PDB); for proteins with only UniProt IDs and for all mutated structures, we generated PDB structures using ColabFold.

Algorithm 1: *ESM3-DTm* Model

Input: CLS token embedding for wild-type $CLS_{\text{w}} \in \mathbb{R}^d$, mutanted $CLS_{\text{m}} \in \mathbb{R}^d$. Mutated position token embedding for wild-type $a_{\text{w}} \in \mathbb{R}^d$, mutated $a_{\text{m}} \in \mathbb{R}^d$.

Step 1: Regression Heads

Head1 = Flatten($\text{a}_{\text{m}} \otimes \mathbf{a}_{\text{w}} \in \mathbb{R}^{d^2}$) $\xrightarrow{\mathbf{W}} \mathbb{R}^d$

Head2 = LayerNorm($\text{CLS}_{\text{w}} - \mathbf{CLS}_{\text{m}}$) $\oplus$ LayerNorm($\mathbf{a}_{\text{w}} - \mathbf{a}_{\text{m}}$) $\in \mathbb{R}^{2d}$

Step 2: MSE Loss Calculation

$$\mathcal{L}_{\text{Head1}} = \mathbf{MSE}\left(N_1(\mathbf{Head1}) - \Delta T_m\right)$$
$$\mathcal{L}_{\text{Head2}} = \mathbf{MSE}\left(N_2(\mathbf{Head2}) - \Delta T_m\right)$$

where N_1 and N_2 are linear layers connected after **Head1** and **Head2**, respectively. **MSE** stands for Mean Squared Error loss function.

Step 3: Ensemble

$$\hat{y}_{\text{ensemble}} = \frac{1}{2}(N_1(\mathbf{Head1}) + N_2(\mathbf{Head2}))$$

$$\mathcal{L}_{\text{ensemble}} = \frac{1}{2}\mathbf{MSE}(\hat{y}_{\text{ensemble}} - \Delta T_m)$$

Experiments

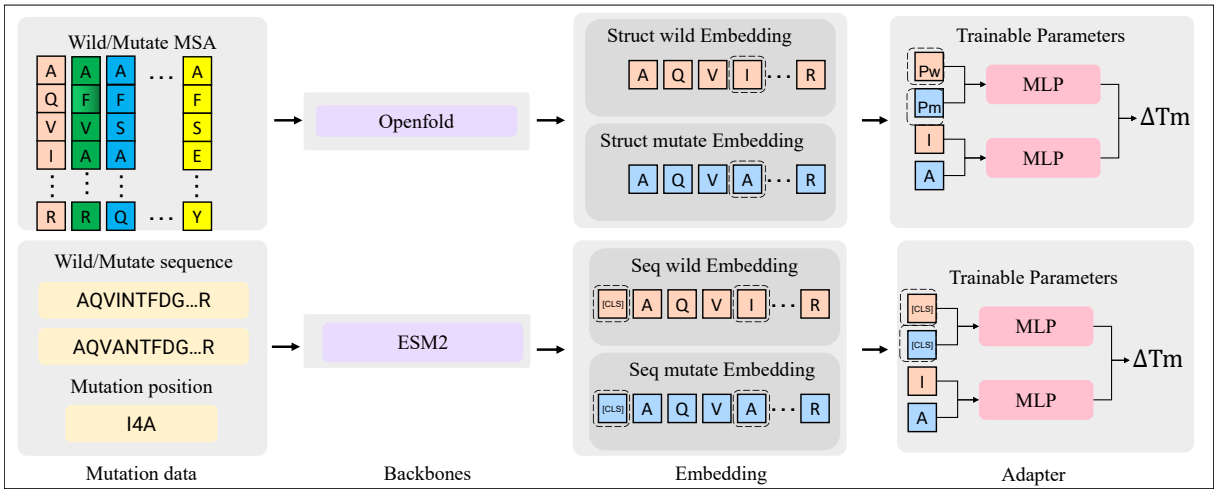
Model Setup and Implementation Details

For each mutation $\mu = (w, m)$, where the amino acid at position w is substituted with amino acid type $m \in AA$, we denote P_w and P_m as the representations of the entire wild-type and mutated protein sequences, respectively. We use a_w and a_m to represent the embeddings at the specific mutated position in the wild-type and mutated proteins. We denote cls_w and cls_m as the CLS token representations of the wild-type and mutated proteins.

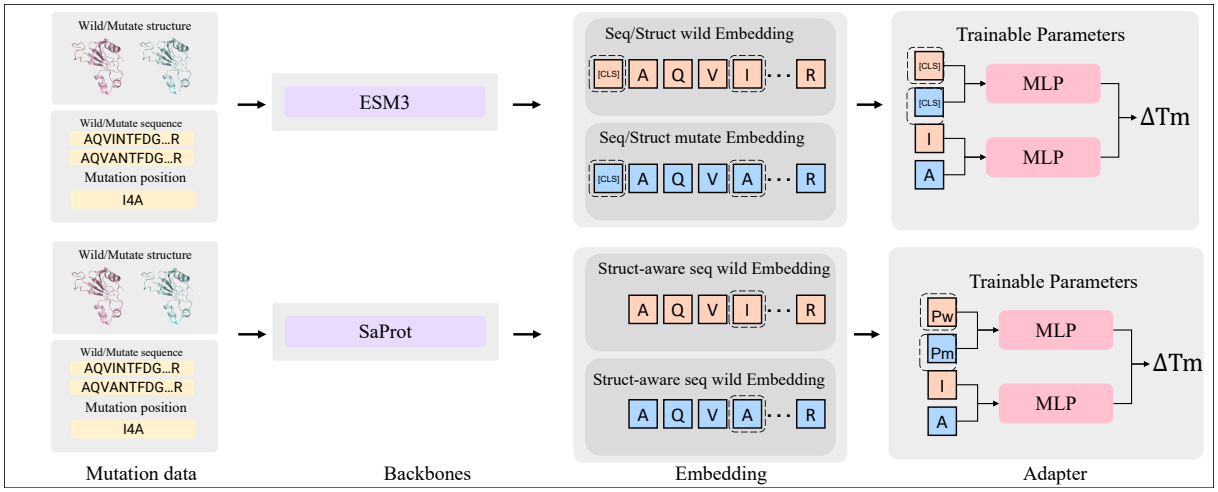
We treat the prediction of the mutation effect on ΔT_m as a regression task involving two sequences: the wild-type and the mutated protein. Our model is built upon a protein language model backbone that accepts inputs of both wild-type and mutated protein sequences, along with structure-related information. In our approach, protein language models serve as feature extractors.

Our most powerful model, *ESM3-DTm*, is built upon ESM3-1.4B (Hayes et al. 2024), a multimodal protein language model that accepts both sequence and PDB structure inputs, as illustrated in Figure 1b. We extract sequence and structure features by applying a linear layer to the final hidden layer outputs from ESM3. We then concatenate the embeddings from $Struct_{\text{cls\_w}}$ and $Seq_{\text{cls\_w}}$ to obtain cls_w, and similarly concatenate $Struct_{\text{cls\_m}}$ and $Seq_{\text{cls\_m}}$ to obtain cls_m. We obtain a_w and a_m in the same manner. Finally, we feed these into a regression head and average the predictions from the individual models in the ensemble. We design various regression heads in Section to predict ΔT_m. The detailed progress is explained in algorithm 1.

We also select other protein language models as feature extractors for comparison. As shown in Figure 1a, ESM2-



(a) Openfold and ESM2 Backbone Architecture



(b) ESM3 and SaProt Backbone Architecture

Figure 1: **Model Architecture**. *ESM3-DTm* efficiently predicts ΔT_m. We also present *ESM2-DTm*, *Saprot-DTm*, and *Openfold-DTm* here. "I4A" means mutation from I to A at position 4.

650M (Lin et al. 2022) and SaProt-650M (Su et al. 2023) have architectures similar to ESM3-1.4B but accept only sequence inputs. Notably, SaProt relies on Foldseek (van Kempen et al. 2022) to obtain structure-aware sequences as input, so we need an additional process when building *Saprot-DTm*. OpenFold (Ahdritz et al. 2024) is another feature extractor we employ. We extract sequence features P_w and P_m from the Evoformer and Structure Module, and also the embeddings at the specific mutated positions, a_w and a_m. These features are then pass through a linear layer.

We train the model using the Adam optimizer(Diederik 2014) with a learning rate of $1 * 10^{-5}$ and the OneCycle scheduler for 10 epochs. Gradient clipping with a norm of 0.1 is applied to ensure stable training. For all protein language backbone, we did not freeze the transformer backbone and trained all model weights in an end-to-end manner. For openfold backbone, we freeze the backbone and only train the linear layer.

Regression Head

For ESM2 and ESM3 backbone, the feature extraction process mainly adopted the corresponding CLS embeddings for global information and mutated position embeddings for local information. We adopt two supervised fine-tuning ways to fusion the wild-type and mutated sequences:

- Outer product of a_w and a_m.
- Linear combination of cls_w and cls_m concatenated with a_w and a_m.

For the OpenFold backbone, since it does not provide CLS embeddings, we use the outputs of the entire sequence after the Evoformer and Structure Modules as global embeddings. We continue to use the embeddings at the mutated positions for local information. Next, We also adopt two su-

pervised fine-tuning ways to fusion the wild-type and mutated sequences:

- Outer product of a_w and a_m.
- Linear combination of P_w and P_m.

For the SaProt backbone, since it also does not provide CLS embeddings, we use the mean of the entire sequence embedding as global information and mutated position embeddings as local information. We also adopt two supervised fine-tuning ways to fusion the wild-type and mutated sequences:

- Outer product of a_w and a_m.
- Linear combination of P_w and P_m.

Results

We evaluate different methods on in Table 1. We primarily used the Pearson correlation coefficient (PCC), root mean square error (RMSE), and mean absolute error (MAE) to assess model performance. The PCC measures the linear correlation between the predicted and true values, indicating the model's ability to rank mutations by their ΔT_m values. The RMSE quantifies how closely the predicted measurements align with the true measurements, while the MAE provides the average absolute difference between predicted and true values. Here we can see that *ESM3-DTm* surpasses **Geostab** 6.4% in PCC, 1.9% in MAE, and 4.4% in RMSE.

Method	r(↑)	MAE(↓)	RMSE(↓)
Struct-based Methods			
HoTMuSiC	0.33	5.70	8.41
AUTO-MUTE	0.29	5.79	8.50
GeoDTm-3D	0.47	5.31	8.03
Seq-based Methods			
GeoDTm-Seq	0.46	5.55	8.11
ESM3-DTm	**0.50**	**5.21**	**7.68**

Table 1: Comparison with existing models on the S571 dataset. Other results are quoted from (Xu, Liu, and Gong 2023).

Furthermore, we fine-tuned models using the ESM2, ESM3, SaProt, and OpenFold backbones with similar architectures to make a fair comparison of their representation abilities, as presented in Table 2. The results indicate that the multimodal ESM3 backbone achieves the highest performance, suggesting that incorporating structural information benefits the prediction.

Ablation Study

Here we explore the performance of different regression heads. Using the ESM2-650M model as the backbone, we conducted all experiments under consistent settings, including those previously established. The results are presented in Table 3. Based on these findings, we selected the best of three to form our final model. Additionally, we investigated the effects of fine-tuning versus freezing the ESM2-650M backbone during prediction, as shown in Table 4. Our results indicate that fine-tuning the backbone significantly improves prediction performance.

Backbone	r(↑)	MAE(↓)	RMSE(↓)
Seq-based Methods			
ESM2-DTm	0.48	5.31	7.85
ESM3-DTm (seq only)	0.49	5.28	7.77
Multimodal Methods			
ESM3-DTm	**0.50**	**5.21**	**7.68**
SaProt-DTm	0.46	5.51	8.00
OpenFold-DTm	0.35	5.31	8.03

Table 2: Comparison of different backbones on the S571 dataset.

Regression Heads	r(↑)	MAE(↓)
MUT Token concatenation	0.21	6.34
MUT Token outer product	0.41	5.69
MUT Token linear combination	0.40	5.50
CLS Token linear combination	0.33	6.01

Table 3: Comparison of different regression heads on the ESM2 Backbone.

Fine-tuning Option	r	MAE	RMSE
Fully fine-tune	**0.48**	**5.31**	**7.85**
Freezing backbone	0.46	5.50	7.89

Table 4: Comparison of finetuning strategies on ESM2 backbone.

We found that in the comparison of protein language model backbones, the results of SaProt are slightly lower than ESM2 and ESM3. Since SaProt does not have a CLS token, we also use the same combination of a_w and a_m for ESM2 when comparing it with SaProt. The result is shown in Table 5. One possible explanation for this result is that SaProt was trained on datasets generated by AlphaFold, whereas we used datasets generated by ColabFold. The differences between these folding models may cause discrepancies in the input PDB structures, leading to suboptimal compatibility with SaProt's model. Another possible reason is that the structure-aware tokens obtained from Foldseek may not accurately capture the changes caused by mutations because Foldseek is primarily designed for sequence alignment rather than for capturing structural variations, which transforms the structure into only 20 tokens. For mutation prediction, this level of granularity may be too coarse, resulting in poor alignment and, consequently, causing the structure to have a negative impact.

Backbone	r	MAE
ESM2	0.30	5.84
SaProt	0.15	6.25

Table 5: Linear combination of avg pooling.

Conclusion

In this work, we propose to utilize multimodal protein language model backbone to make effective prediction on melting temperature. Our findings demonstrate that the multimodal model *ESM3-DTm* outperforms other single-modality models. By effectively incorporating both sequence and structural data in fully fine-tuning strategy, we can achieve more accurate predictions.

Acknowledgments

We would like to acknowledge the support from Beijing Municipal Science & Technology Commission, Administrative Commision of Zhongguancun Science Park (Z221100003522019).

References

Ahdritz, G.; Bouatta, N.; Floristean, C.; Kadyan, S.; Xia, Q.; Gerecke, W.; O'Donnell, T. J.; Berenberg, D.; Fisk, I.; Zanichelli, N.; et al. 2024. OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods*, 1–11.

Benevenuta, S.; Pancotti, C.; Fariselli, P.; Birolo, G.; and Sanavia, T. 2021. An antisymmetric neural network to predict free energy changes in protein variants. *Journal of Physics D: Applied Physics*, 54(24): 245403.

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

Chen, Y.; Xu, Y.; Liu, D.; Xing, Y.; and Gong, H. 2024. An end-to-end framework for the prediction of protein structure and fitness from single sequence. *Nature Communications*, 15(1): 7400.

Diederik, P. K. 2014. Adam: A method for stochastic optimization. *(No Title)*.

Fowler, D. M.; and Fields, S. 2014. Deep mutational scanning: a new style of protein science. *Nature methods*, 11(8): 801–807.

Gromiha, M. M.; An, J.; Kono, H.; Oobatake, M.; Uedaira, H.; Prabakaran, P.; and Sarai, A. 2000. ProTherm, version 2.0: thermodynamic database for proteins and mutants. *Nucleic acids research*, 28(1): 283–285.

Gromiha, M. M.; An, J.; Kono, H.; Oobatake, M.; Uedaira, H.; and Sarai, A. 1999. ProTherm: thermodynamic database for proteins and mutants. *Nucleic acids research*, 27(1): 286–288.

Gromiha, M. M.; Uedaira, H.; An, J.; Selvaraj, S.; Prabakaran, P.; and Sarai, A. 2002. ProTherm, thermodynamic database for proteins and mutants: developments in version 3.0. *Nucleic acids research*, 30(1): 301–302.

Hayes, T.; Rao, R.; Akin, H.; Sofroniew, N. J.; Oktay, D.; Lin, Z.; Verkuil, R.; Tran, V. Q.; Deaton, J.; Wiggert, M.; et al. 2024. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024–07.

Hie, B. L.; Shanker, V. R.; Xu, D.; Bruun, T. U.; Weidenbacher, P. A.; Tang, S.; Wu, W.; Pak, J. E.; and Kim, P. S. 2024. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2): 275–283.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. 2021. Highly accurate protein structure prediction with AlphaFold. *nature*, 596(7873): 583–589.

Kumar, M. S.; Bava, K. A.; Gromiha, M. M.; Prabakaran, P.; Kitajima, K.; Uedaira, H.; and Sarai, A. 2006. ProTherm and ProNIT: thermodynamic databases for proteins and protein–nucleic acid interactions. *Nucleic acids research*, 34(suppl_1): D204–D206.

Li, F.; Yuan, L.; Lu, H.; Li, G.; Chen, Y.; Engqvist, M. K.; Kerkhoven, E. J.; and Nielsen, J. 2022. Deep learning-based k cat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, 5(8): 662–672.

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*, 2022: 500902.

Masso, M.; and Vaisman, I. 2014. AUTO-MUTE 2.0: a portable framework with enhanced capabilities for predicting protein functional consequences upon mutation. Adv Bioinf. 2014.

Masso, M.; and Vaisman, I. I. 2008. Accurate prediction of stability changes in protein mutants by combining machine learning with structure based computational mutagenesis. *Bioinformatics*, 24(18): 2002–2009.

Mirdita, M.; Schütze, K.; Moriwaki, Y.; Heo, L.; Ovchinnikov, S.; and Steinegger, M. 2022. ColabFold: making protein folding accessible to all. *Nature methods*, 19(6): 679–682.

Ouyang-Zhang, J.; Diaz, D.; Klivans, A.; and Krähenbühl, P. 2024. Predicting a protein's stability under a million mutations. *Advances in Neural Information Processing Systems*, 36.

Pandurangan, A. P.; and Blundell, T. L. 2020. Prediction of impacts of mutations on protein structure and interactions: SDM, a statistical approach, and mCSM, using machine learning. *Protein Science*, 29(1): 247–257.

Pires, D. E.; Ascher, D. B.; and Blundell, T. L. 2014. mCSM: predicting the effects of mutations in proteins using graph-based signatures. *Bioinformatics*, 30(3): 335–342.

Pucci, F.; Bourgeas, R.; and Rooman, M. 2016. Predicting protein thermal stability changes upon point mutations using statistical potentials: Introducing HoTMuSiC. *Scientific reports*, 6(1): 23257.

Steinegger, M.; and Söding, J. 2017. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11): 1026–1028.

Su, J.; Han, C.; Zhou, Y.; Shan, J.; Zhou, X.; and Yuan, F. 2023. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, 2023–10.

Tsuboyama, K.; Dauparas, J.; Chen, J.; Laine, E.; Mohseni Behbahani, Y.; Weinstein, J. J.; Mangan, N. M.; Ovchinnikov, S.; and Rocklin, G. J. 2023. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973): 434–444.

Umerenkov, D.; Shashkova, T. I.; Strashnov, P. V.; Nikolaev, F.; Sindeeva, M.; Ivanisenko, N. V.; and Kardymon, O. L. 2022. PROSTATA: protein stability assessment using transformers. *BioRxiv*, 2022–12.

van Kempen, M.; Kim, S. S.; Tumescheit, C.; Mirdita, M.; Gilchrist, C. L.; Söding, J.; and Steinegger, M. 2022. Foldseek: fast and accurate protein structure search. *Biorxiv*, 2022–02.

Whisstock, J. C.; and Lesk, A. M. 2003. Prediction of protein function from protein sequence and structure. *Quarterly reviews of biophysics*, 36(3): 307–340.

Xavier, J. S.; Nguyen, T.-B.; Karmarkar, M.; Portelli, S.; Rezende, P. M.; Velloso, J. P.; Ascher, D. B.; and Pires, D. E. 2021. ThermoMutDB: a thermodynamic database for missense mutations. *Nucleic acids research*, 49(D1): D475–D479.

Xu, Y.; Liu, D.; and Gong, H. 2023. Improving the prediction of protein stability changes upon mutations by geometric learning and a pre-training strategy. *bioRxiv*, 2023–05.

Yu, H.; Deng, H.; He, J.; Keasling, J. D.; and Luo, X. 2023. UniKP: a unified framework for the prediction of enzyme kinetic parameters. *Nature communications*, 14(1): 8211.

Zhang, Q.; Ding, K.; Lyv, T.; Wang, X.; Yin, Q.; Zhang, Y.; Yu, J.; Wang, Y.; Li, X.; Xiang, Z.; et al. 2024. Scientific large language models: A survey on biological & chemical domains. *arXiv preprint arXiv:2401.14656*.