ProtMamba Finetuning on Chorismate Mutase Protein Family

Elisa Billard
EPFL Life Sciences Engineering Section
Supervisor Cyril Malbranke, Bitbol Lab

***Abstract*—This project investigates the fine-tuning of the protein language model ProtMamba [1] for functional classification within the chorismate mutase (CM) protein family. We explored sequence embeddings to predict enzymatic activity and assessed their effectiveness using binary classification and regression tasks. Building on the success of embedding-based predictions, we fine-tuned ProtMamba by replacing its next-token prediction layer with a task-specific prediction layer. Our results highlight the impact of evolutionary context on prediction accuracy and demonstrate the potential of fine-tuned protein language models in functional annotation.**

I. Introduction

Understanding protein function and evolution is critical for advancements in biotechnology, medicine, and synthetic biology. Recent developments in protein language models (PLMs) have provided a powerful framework for analyzing protein sequences by leveraging machine learning techniques to model sequence-function relationships. These models, trained on large-scale protein sequence datasets, capture meaningful patterns that are reflective of structural and functional properties.

ProtMamba [1], a protein language model based on the state-space Mamba architecture, has demonstrated significant capabilities in modeling protein sequences. Designed to handle long sequences and utilize a fill-in-the-middle (FIM) training objective, it is capable of predicting variant fitness, and generating novel sequences. Its unique architecture combines autoregressive and masked language modeling approaches, enabling it to learn nuanced representations of protein sequences (see Figure 2). In contrast to other PLMs such as the MSA Transformer [2], it does not need the Multiple Sequence Alignment of protein families, which could introduce a bias.

This study focuses on leveraging ProtMamba embeddings to analyze and predict functional activity in chorismate mutase (CM) proteins. CM plays a vital role in the biosynthesis of aromatic amino acids, making it a biologically and industrially significant enzyme (see Figure 1). By using ProtMamba to extract embeddings and analyzing their predictive power, we aim to explore how evolutionary context contributes to understanding sequence-function relationships in CM proteins.

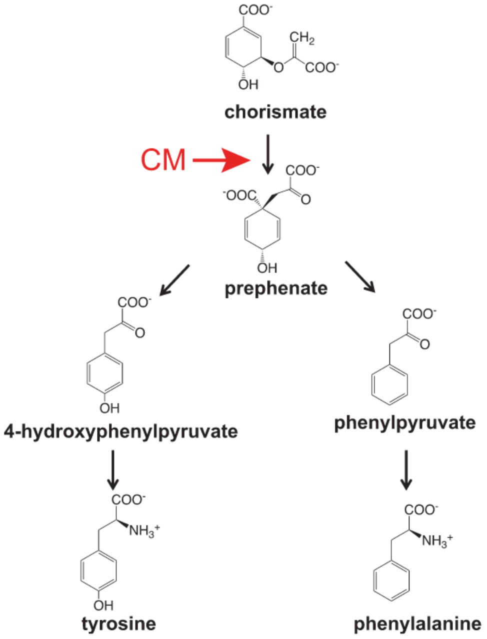

Fig. 1: Chorismate Mutase (CM) in the Shikimate Pathway

We also fine-tune ProtMamba and replace its next-token prediction layer with a task-specific prediction layer. Finally we perform a similar analysis on the RuBisCO protein family.

II. Chorismate Mutase Dataset

The dataset utilized in this study consists of both natural and Direct Coupling Analysis (DCA)-generated sequences of chorismate mutase (CM) by a previous paper [3].

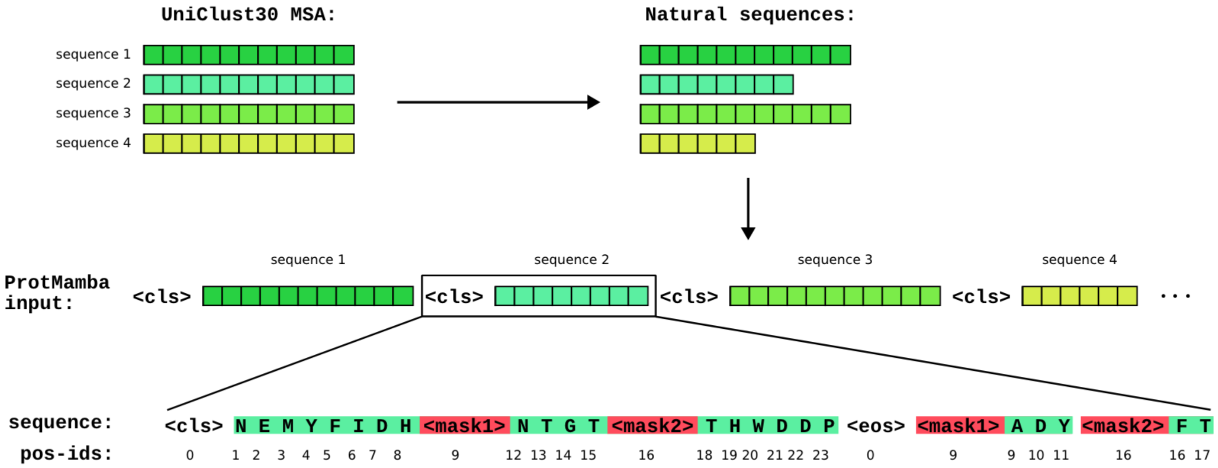

Fig. 2: Input to ProtMamba: each element of the input is a concatenation of unaligned homologous sequences

A. *Natural Sequences*

The natural CM sequences were curated from experimentally validated protein databases by the previous paper [3]. These 1130 sequences represent diverse evolutionary backgrounds and are annotated with their functional activity, measured through experimental assays.

B. *DCA-Generated Sequences*

The DCA sequences were computationally designed using the approach described in [3] (see Figure 3). This method leverages Direct Coupling Analysis to infer evolutionary constraints from a multiple sequence alignment (MSA) of natural CM sequences. By learning pairwise interactions between residues, the model generates synthetic sequences that reflect the evolutionary landscapes of CM proteins. The MSA used for DCA was constructed from a dataset of natural CM sequences retrieved from public databases such as UniProt and Pfam. Using these alignments, the DCA method identifies statistically significant couplings between residue pairs, which are then used to parameterize a probabilistic model. This model generates synthetic CM sequences that adhere to evolutionary constraints, simulating plausible functional variants. The researcher published 1627 generated sequences that we all use to construct our dataset.

C. *Functional Annotation*

Both the natural and DCA-generated sequences were experimentally or computationally evaluated for their functional activity (see Figure 4). It is expressed as normalized relative enrichment (*norm r.e.*), which serves as an indicator of the sequence's ability to catalyze the conversion of chorismate to prephenate. The distribution of this enrichment is bimodal (see Figure 5), which will be important for the later binary classification.

III. Supervised Learning with Protein Embeddings

A. *Training Dataset Creation*

To observe the effect of giving the model context, we concatenated DCA sequences with varying numbers of natural sequences, referred to as "context". Context lengths of 0, 2, 10, and 50 natural sequences (randomly sampled) were used to create four distinct training datasets. For each context length, tokenized representations of the sequences were generated and stored. The dataset was labeled based on the *norm r.e.* values, or with binary classification labels assigned using a threshold of 0.5. These labeled datasets each of length 1627 enabled classification experiments on both sequence embeddings and context information.

B. *Embedding Extraction*

Protein sequence embeddings were extracted using the state-space model ProtMamba. We extracted the embeddings of the last token of each sequence in final hidden layer of the model. This last token should represent the last sequence of our input (the one we classify), partially modified by the previous

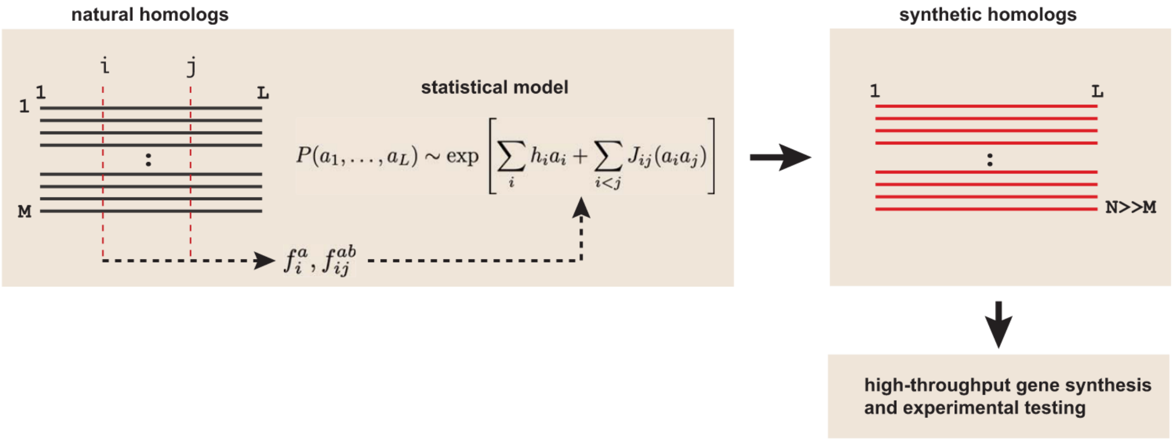
natural homologs
i
j
1
1
L
M
:
statistical model
$P(a_1,\ldots,a_L) \sim \exp\left[\sum_i h_i a_i + \sum_{i<j} J_{ij}(a_i a_j)\right]$
f_i^a, f_{ij}^{ab}
synthetic homologs
1
L
:
N>>M
high-throughput gene synthesis and experimental testing

Fig. 3: Pipeline to generate artificial DCA sequences, starting from a statistical model of the MSA [3]

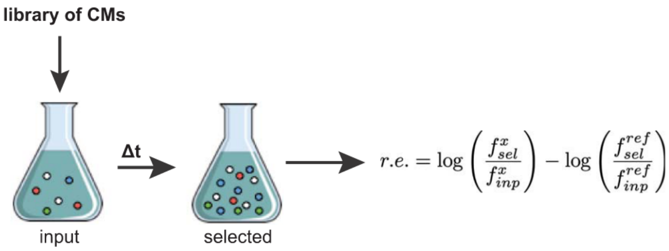
library of CMs
Δt
input
selected
$r.e. = \log\left(\frac{f^x_{sel}}{f^x_{inp}}\right) - \log\left(\frac{f^{ref}_{sel}}{f^{ref}_{inp}}\right)$

Fig. 4: Workflow for functional characterization of CM activity

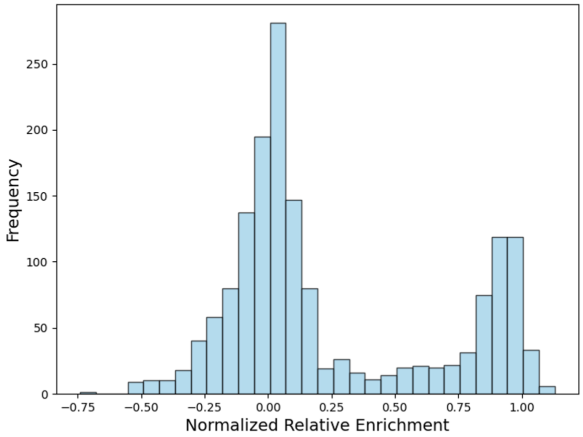

Fig. 5: Distribution of the normalized relative enrichment of the CM proteins, measure of their functional activity

context sequences. These embeddings were saved for each context length and used as input features for downstream classification tasks. The embeddings capture sequence-specific information and their interaction with the provided context, facilitating the evaluation of how context impacts classification performance.

C. *Embedding Visualization*

To explore the structure and separability of the embeddings, we performed dimensionality reduction using UMAP and PCA. The PCA visualizations revealed how the principal components capture variance in the embeddings, while UMAP provided a nonlinear embedding optimized for visualizing both local and global structures. To further interpret the contribution of specific features to the embeddings, we utilized SHAP (Shapley Additive Explanations) values, which helped us identify which features were most influential in the separability of sequences. In particular, PC3 and PC9 were chosen for visualization based on their ability to capture key variance, as highlighted by SHAP analysis. We therefore plot PC3 vs PC9 (see Figure 6), however there is no clear separation between clusters corresponding to functional and non-functional sequences. This could be due to limitations of the embeddings, the reduction to only 2 dimensions or just the biological overlap in functionality.

D. *Functional Activity Prediction*

1) Binary Classification: To predict functional activity (*norm r.e.*) from embeddings, we performed binary classification using models such as Random Forest Classifiers and Logistic Regression. The embeddings served as input features, with labels derived from a thresholded *norm r.e.* value of 0.5.

Classification performance was assessed using the area under the ROC curve (AUC) and the F1 score. A

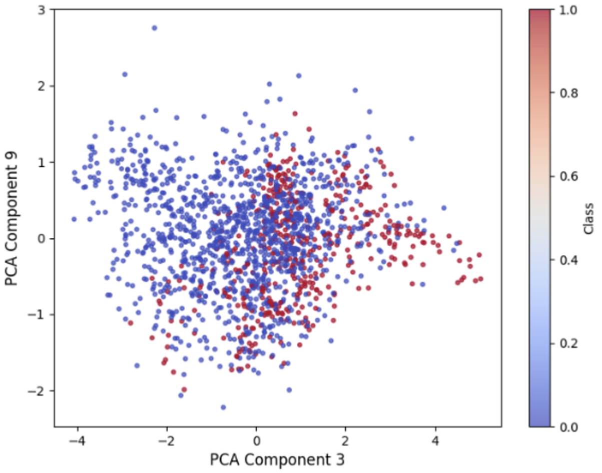

Fig. 6: PCA of the ProtMamba embeddings of the CM sequence to label conditioned by 50 context sequences

systematic evaluation across different context lengths (0, 2, 10 and 50) revealed that context length positively correlates with classification accuracy (see Table I). We chose the logistic regression as it gave the best AUC results across all context length. We then plotted the classification scores according to context length in Figure 7. Both AUC and F1 scores increased with context length, showing gains up to a context length of 10, beyond which the improvement plateaued. These results emphasize the potential of ProtMamba embeddings in distinguishing functional sequences, especially when enriched with evolutionary context.

Context Methods	0	2	10	50
Logistic Regression	**0.88**	**0.89**	**0.89**	**0.90**
Random Forest	0.87	0.87	0.88	0.88
SVM	0.84	0.84	0.84	0.86

TABLE I: AUC Binary Classification results comparison for different methods across context lengths

We were also interested in looking at the influence of the enrichment values of the context on the classification. On Figure 8 we see that the positive context, which includes only sequences with enrichment < 0.5, achieves the best performance across all metrics. This suggests that providing relevant context enhances the model's ability to distinguish functional sequences effectively. Interestingly, the random context slightly outperforms the negative context. This could hint that having some positive enrichment values helps the classification. The reason why this could be the case is not yet very clear.

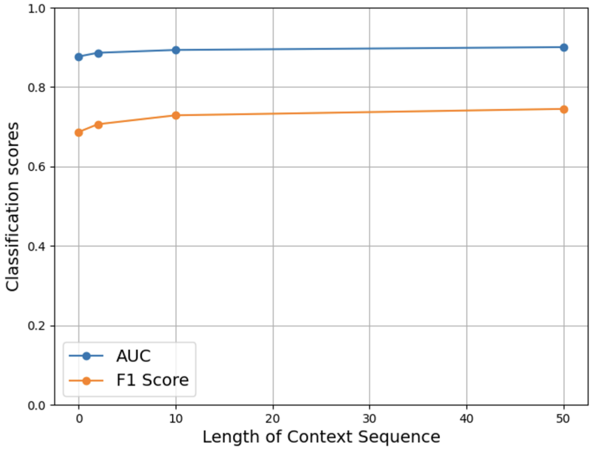

Fig. 7: Results of the classification AUC and F1 score for different lengths of context, using Logistic Regression

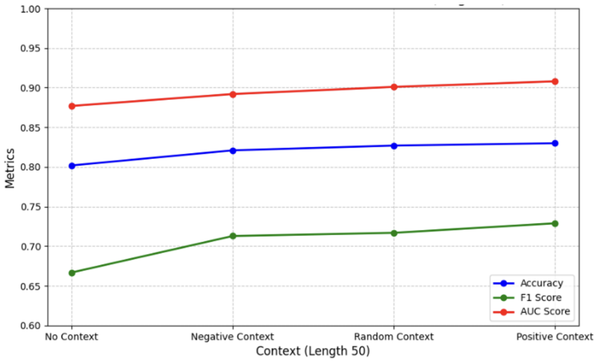

Fig. 8: Classification Performance metrics across contexts of length 50 with different enrichment values

2) Regression: To further investigate the relationship between the embeddings and functional activity, we applied regression models to predict the continuous *norm r.e.* values. Using Random Forest Regressors, Gradient Boosting, and other methods, we evaluated regression performance via Spearman correlation and R-squared. Regression experiments revealed a moderate predictive capability (e.g., Spearman correlation scores around 0.4 to 0.6), which varied with the context length (see Figure 9. Longer context lengths produced embeddings that were a bit more predictive of *norm r.e.* values.

Similarly to the analysis for the binary classification, we observe that the context improve the prediction. However, on Figure 10 the relative enrichment of those sequences seems to improve the prediction when negative (i.e. proteins not working). This analysis provided a more nuanced understanding of how embeddings encode functional activity, complementing

Context Methods	0	2	10	50
Linear Regression	0.46	0.50	0.52	0.50
Lasso ($\alpha = 0.001$)	0.50	0.52	0.55	0.56
Ridge ($\alpha = 0.1$)	0.48	0.51	0.53	0.53
Random Forest Regressor	**0.54**	**0.59**	**0.58**	**0.60**

TABLE II: Spearman Correlation across Context Lengths for different Regression Methods

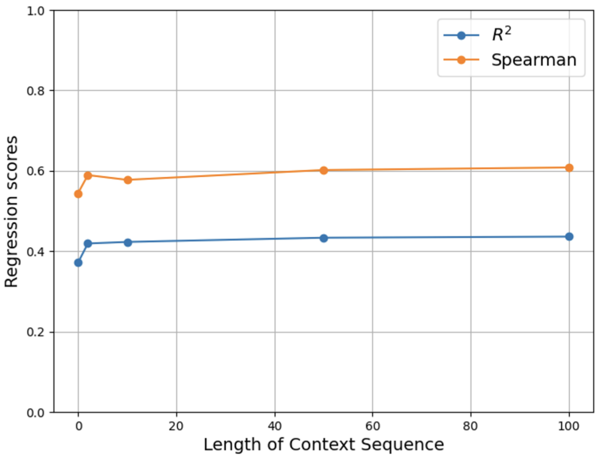

Fig. 9: Results of the regression R-squared and spearman scores for different lengths of context

the binary classification findings.

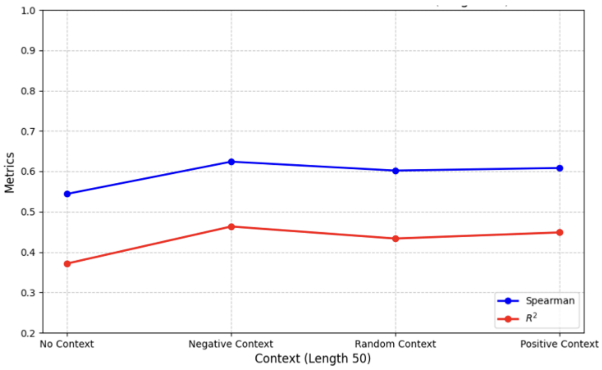

Fig. 10: Regression Performance metrics across contexts of length 50 with different enrichment values

IV. Fine-tuning of ProtMamba

To enhance ProtMamba's ability to predict functional activity from protein embeddings, we explored a fine-tuning approach leveraging selective layer adaptation. Fine-tuning allows the model to adapt pre-trained features to a specific downstream task, such as classification, by updating its parameters with task-specific labeled data. This enables the model to leverage learned representations while tailoring them to new challenges.

A. *Selective Layer Adaptation*

In this method, we fine-tune the model by selectively unfreezing layers of the pre-trained ProtMamba backbone. By freezing earlier layers, we retain their general-purpose features learned during pre-training, while allowing only the last few layers to update during fine-tuning. This approach is particularly useful when downstream tasks differ from the pre-training objectives, as the last layers may be overly specialized for tasks like next-token prediction. This is the case for ProtMamba, which was originally designed for Next Token Prediction but is now adapted for prediction tasks.

To implement this, a parameter was introduced to control the number of trainable layers. By unfreezing only the last few layers of the backbone, we focused learning on task-specific features while avoiding potential overfitting. For example, fine-tuning the last two layers of the backbone allowed adaptation to the classification task without significantly altering the general-purpose features encoded in earlier layers.

B. *Adaptation of ProtMamba*

ProtMamba was initially designed for Next Token Prediction using a masked language modeling objective. The original output layer was therefore replaced with a binary classification or regression head.

C. *Training Process*

1) **Embedding Extraction:** Pretrained ProtMamba embeddings were extracted for each sequence. Fine-tuning was performed by unfreezing the last few layers of ProtMamba and backpropagating the binary classification loss (Binary Cross-Entropy) through these layers.
2) **Layer Selection:** Only the final layers of the ProtMamba backbone were retrained (see the tuning of hyperparameters in Stopping Layer and Unfreezing Layer), while earlier layers remained frozen to preserve their pre-trained features.
3) **Optimization:** Fine-tuning was performed using:
   - *Loss Function:* Cross-Entropy Loss for the Classification and Mean Square Error Loss for the Regression

- *Optimizer:* AdamW with weight decay of 1×10^{-3}
- *Learning Rate Scheduler:* A step-based scheduler that reduces the learning rate by a factor of 0.99 after every epoch, promoting gradual adaptation.

D. Hyperparameter Tuning

1) Learning Rate: Figure 11 illustrates the impact of learning rate selection on model performance, measured by AUC. Among the tested values, the best learning rate was determined to be 1×10^{-4}, as it consistently achieved the highest AUC.

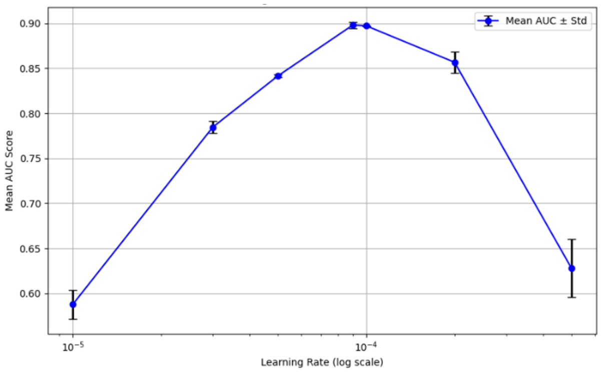

Fig. 11: Learning Rate selection according to best AUC (Binary Classification)

For the regression, the best learning rate was 2×10^{-4} as can be seen on Figure 12.

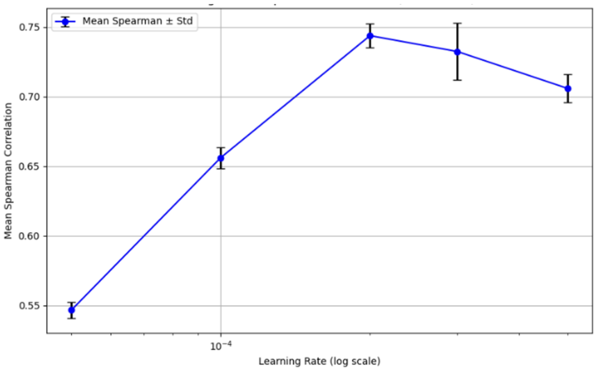

Fig. 12: Learning Rate selection according to best Spearman score (Regression)

2) Stopping Layer: The Stopping Layer refers to where in the ProtMamba model we stop taking the layers for the fine-tuning. This means we keep the layers before that and the ones after this one are discarded. The choice of stopping layer impacts model performance, as illustrated by Figure 13 for the classification and Figure 14 for the regression. We observe that the layers in the middle of the model seem to encode the most useful information of prediction.

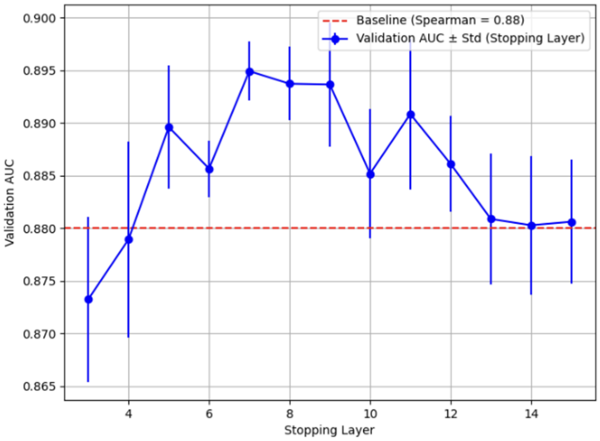

Fig. 13: Stopping layer selection according to best AUC (Binary Classification), with standard deviation from 5 repetitions. The unfreezing layer value was set to 3.

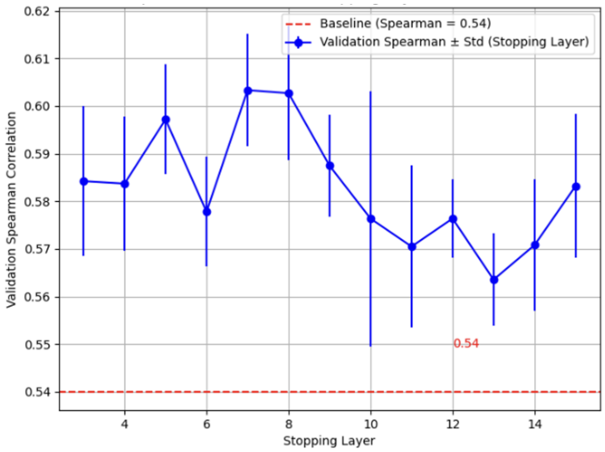

Fig. 14: Stopping layer selection according to best Spearman Correlation (Regression), with standard deviation from 5 repetitions. The unfreezing layer value was set to 3.

3) Unfreezing Layer: The Unfreezing Layer refers to how many layers in the ProtMamba model we retrain. By unfreezing progressively deeper layers, the model can adapt more to the task-specific data, balancing between retaining pre-trained features and optimizing for new objectives. Figure 15 illustrates the effect of unfreezing layer selection on binary classification, and Figure 16 on the regression task. The trend is less clear for this parameter, we do see that unfreezing more layer brings more variability to the model.

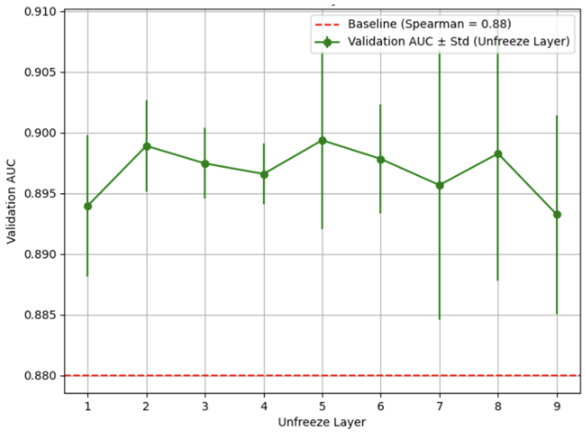

Fig. 15: Unfreezing layer selection according to best AUC (Binary Classification), with standard deviation from 5 repetitions. The stopping layer value was set to 9.

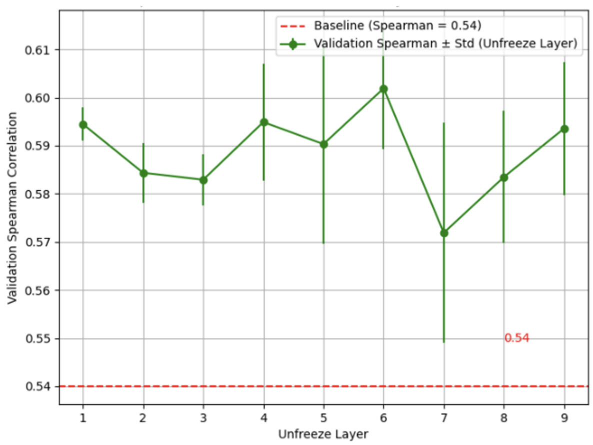

Fig. 16: Unfreezing layer selection according to best Spearman Correlation (Regression), with standard deviation from 5 repetitions. The stopping layer value was set to 9.

E. *Fine-tuning with Context*

Next we are interested in fine-tuning the ProtMamba model using sequences enriched with context (50 sequences in front of the one we want to predict the activity). We hypothesize that by leveraging the information present in the context sequences, the model could predict functional activity more accurately. However, either because of a lack of good training or because the hypothesis is wrong, we could not observe this result.

V. RuBisCO Protein Family

The ribulose-1,5-bisphosphate carboxylase/oxygenase (RuBisCO) protein family is a cornerstone of the photosynthetic process, catalyzing the fixation of atmospheric carbon dioxide into organic compounds. This enzyme is highly conserved across diverse organisms, including plants, algae, and photosynthetic bacteria. RuBisCO exists in multiple forms (I-IV), with the most studied being Form I, composed of large and small subunits. Despite its critical role in global carbon cycling, RuBisCO operates with relatively low efficiency, characterized by a slow catalytic rate and competitive inhibition by oxygen [4]. These limitations have made it a target for protein engineering and synthetic biology efforts aimed at enhancing photosynthetic efficiency and agricultural productivity. Previous work [5] have conducted a high-throughput kinetic characterization of over 100 bacterial form I rubiscos, giving interesting insight into their activity. The sequences as well as an efficiency metric are available [6], making it possible to make a similar analysis as for the Chorismate Mutase family. We will first try to predict this efficiency metric called `rate_lab`, with the sequence of the large subunit embedded by our model ProtMamba (see Figure 17 for the distribution of the efficiency metric).

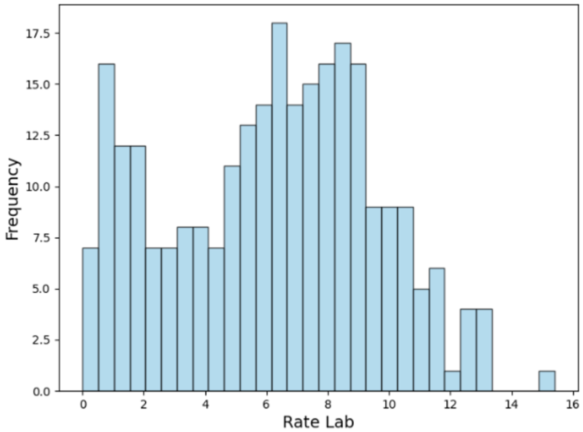

Fig. 17: Distribution of the Rate Lab value of the Rubisco proteins, measure of their functional activity. Proteins with a Rate Lab above 10 are considered active.

A. *Regression on Sequence Embeddings*

In the same way as for the CM protein family, we do supervised learning to predict the activity of the proteins, based on the embeddings extracted from ProtMamba. We obtain a rather good prediction capability with a spearman correlation score of 0.63 with a Random Forest Regressor. We find that the type of context used during the training phase had a significant impact on regression performance metrics. Specifically, when the context was restricted to sequences with positive Rate Lab values (> 10), there was a noticeable improvement in both Spearman correlation and R^2 metrics compared to contexts with mixed or negative Rate Lab values (see Figure 18). This finding underscores the importance of context selection in embedding-based regression tasks, as high-quality, high-performance context sequences provide more relevant information to guide the model's predictions.

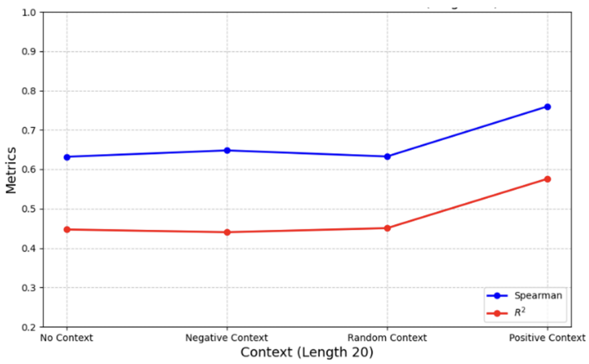

Fig. 18: Regression Performance metrics across contexts of length 20 with different Rate Lab values, Positive is Rate Lab $>$ 10, the regression was performed with Random Forest Regressor

B. *Fine-tuning ProtMamba*

Similarly to what was done for the CM family, we fine-tuned ProtMamba with the new regression objective. We chose the best hyperparameters by looking at the spearman correlation metric (see Figure 19 for the stopping layer and Figure 20 for the unfreezing layer). Globally we observe a similar trend, that is, the layers in the middle contain the most useful information for the prediction. However, we see that the more layers we retrain, the better the prediction, which could indicate that the pretrained model is not that useful.

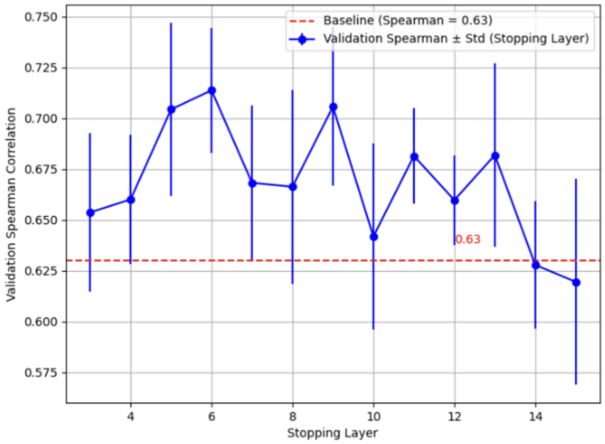

Fig. 19: Stopping layer selection according to best Spearman Correlation, with standard deviation from 5 repetitions. The unfreezing layer value was set to 3. The baseline is the Random Forest Regressor of the supervised task.

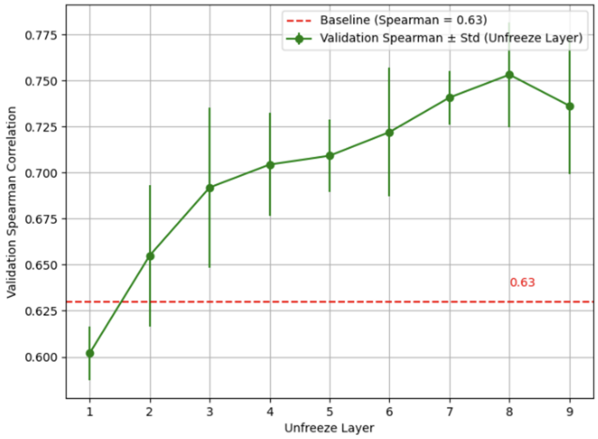

Fig. 20: Unfreezing layer selection according to best Spearman Correlation, with standard deviation from 5 repetitions. The stopping layer value was set to 9. The baseline is the Random Forest Regressor of the supervised task.

VI. Discussion

This project has highlighted the information contained in protein sequences on their functionality. The ProtMamba model successfully represents the sequences enabling the prediction of their efficiency with basic supervised learning methods. Directly fine-tuning the ProtMamba model to perform this task, further improved the predictions.

One aspect that could be interesting to explore is the fine-tuning method, here we use the basic idea of unfreezing some layers, however, some smarter ways like Low Rank Adaptation (LoRA) [7] have been developed. Instead of updating all the parameters of the model, LoRA introduces task-specific trainable parameters while freezing the original pre-trained weights. This significantly reduces computational overhead and memory requirements. LoRA assumes that the weight updates during fine-tuning lie in a low-rank subspace.

Another point to consider is the construction of the training and testing set. In some cases, performing a random train-test split can introduce bias, especially if the data are too similar. When the data lack significant diversity, a random split might result in both training and test sets containing nearly identical or highly correlated samples. This can lead to an overestimation of the performance of the model because the test set may not truly represent new unseen data. The model could perform well due to the repetition of similar patterns in both sets, rather than generalizing effectively to different, previously unseen instances. In our case, using the Protein Gym benchmark, which already provides splits based on similarity, could be a good solution to mitigate this risk. Designed to evaluate the generalization ability of models in protein sequence prediction tasks, it ensures that the training and test sets contain distinct, non-redundant protein sequences.

VII. Conclusion

In this report, we demonstrated the successful application of ProtMamba embeddings for functional classification within the chorismate mutase protein family. Context sequences significantly enhanced classification performance, with longer contexts improving predictive accuracy up to a saturation point. Fine-tuning ProtMamba further improved its ability to distinguish functional from non-functional sequences, showcasing the adaptability of protein language models to task-specific challenges. Despite these advances, fine-tuning with context sequences did not yield the expected benefits, suggesting potential areas for improvement in training strategies or model architectures. These findings emphasize the utility of protein language models for functional annotation and pave the way for future applications in protein design and synthetic biology.

VIII. Acknowledgments

I am deeply grateful to my supervisor Cyril Malbranke for his guidance and to Prof. Bitbol for her advice.

References

[1] Anne-Florence Bitbol Damiano Sgarbossa Cyril Malbranke. "ProtMamba: a homology-aware but alignment-free protein state space model". In: *bioRxiv* (2024).

[2] Roshan Rao et al. "MSA Transformer". In: *bioRxiv* (2021). doi: 10.1101/2021.02.12.430858. url: https://doi.org/10.1101/2021.02.12.430858.

[3] M. J. Rollins et al. "An Evolution-Based Model for Designing Chorismate Mutase Enzymes". In: *Nature Communications* 14 (2023), pp. 1234–1245.

[4] I. Andersson and A. Backlund. "Rubisco Function, Evolution, and Engineering". In: *Annual Review of Biochemistry* 89 (2020), pp. 517–534.

[5] Benoit de Pins et al. "Systematic exploration of bacterial form I Rubisco maximal carboxylation rates". In: *bioRxiv* (2023). doi: 10.1101/2023.07.27.550689. url: https://doi.org/10.1101/2023.07.27.550689.

[6] Benoit de Pins et al. "Rubisco is slow across the tree of life". In: *bioRxiv* (2025). doi: 10.1101/2025.01.19.633714.

[7] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations (ICLR)*. arXiv preprint arXiv:2106.09685. 2022. url: https://arxiv.org/abs/2106.09685.