Project Paper - Split Protein Prediction

Group 2 - Chuqiao Feng, Yuyan Cai, Yuanhang Li, Sohail Mohammod

December 6, 2024

Abstract

Fluorescent proteins (FPs) are widely used in biomolecular and cellular studies and assay developments, especially in biomolecular complementation assays. However, molecular splitting for complementary purposes in FPs usually require massive efforts of structural validation and experimentation. With advancement of Protein Language Models (PLMs), structural prediction for molecular splitting becomes possible. We propose leveraging Protein Language Models (PLMs), such as ProteinBERT, combined with a Multilayer Perceptron (MLP), to predict optimal protein split sites. By fine-tuning ProteinBERT on labeled datasets for secondary structure and stability, we preserve pre-trained knowledge while adapting to fluorescence-specific tasks. Our pipeline includes encoding protein sequences, extracting attention weights to interpret the model, and visualizing attention distributions. The fine-tuned model predicts split sites by evaluating protein stability and secondary structure features, improving model precision. Results demonstrate robust performance on stability prediction and protein split site identification, validating the model's ability to enhance the creation of split variants for fluorescent proteins.

1 Introduction

Since their discovery, FPs have revolutionized the entire field of molecular and cell biology by tracking various biological processes in real time [**?**]. FPs, which originate from marine organisms such as *Aequorea victoria*, can emit natural fluorescences that can be applied to study protein location tracking, protein-protein interactions (PPI), and dynamic changes in cellular systems [**?**]. After decades of development, advancements in FP engineering has led to various variants with increased brightness, higher stability, or expanded spectral features [**?**]. The capacity to visualize and measure biomolecular processes with high resolution has expedited not only biomedical research, but also drug discovery and clinical diagnostics [**?**]. Among all applications, Bimolecular Fluorescence Complementation (BiFC), which utilizes split FPs to visualize protein-protein interactions, exemplifies this potential by facilitating the study of intracellular signaling pathways, intercellular communications, and complex modulation networks in different cell types, thus enhancing current understanding of biological dynamics at molecular level [**?**]. In real world settings, increasing demands for high-throughput and precise diagnostics have emerged. The traditional way of choosing split site to generate BiFC tool utilize structure information of FPs. Split sites are typically chosen to avoid secondary structures and conserved regions, but the sheer number of suitable candidates makes this task challenging.

To tackle challenges in protein split site searching, the goal of this project is to leverages **Protein Language Models (PLMs)** combined with a **Multilayer Perceptron (MLP)** to predict optimal positions for complementation. The proposed workflow involves encoding protein sequence information using PLMs such as **ProteinBERT** and training the model on BiFC data gathered from the literature. This approach aims to enhance the BiFC toolkit by creating split variants for fluorescent proteins like **mCardinal** and **mNeptune**, expanding the Far-Red and Near-Infrared Fluorescent Protein families.

Combined with the cutting-edge prediction through PLMs, the expanded and optimized BiFC toolkit will empower more accurate and stronger visualization of PPIs, boostering molecular diagnostics and drug discovery. By introducing scalable and versatile assays, Our pipeline not only aims to expand molecular tools, but also seeks to accelerate drug discovery progress and downstream testing.

This innovative application of PLMs represents a significant step forward in computational protein engineering, providing a data-driven approach to enable high-throughput and efficient design of BiFC.

2 Materials and Methods

2.1 Graphical overview of methods

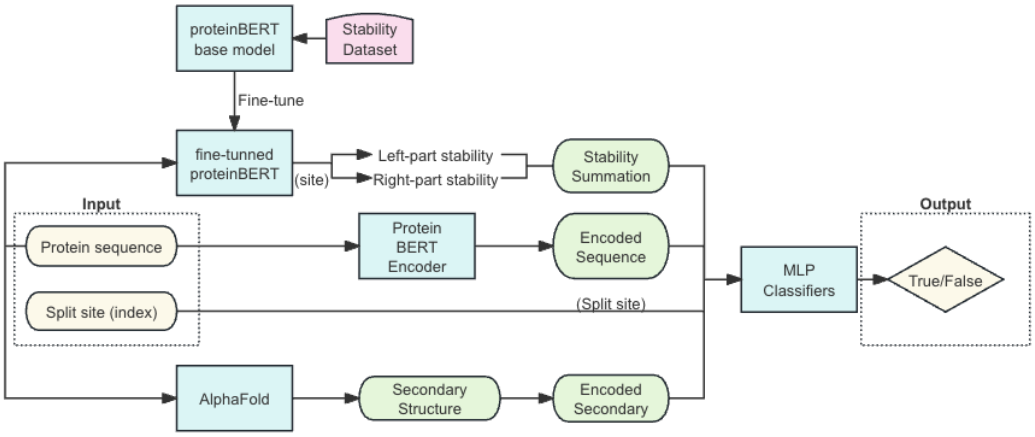


Figure 1: Pipeline of protein split site prediction

2.2 Dataset

2.2.1 Fluorescence split site

The split site data were collected from experimental results, where we looked through over twenty pubslihed papers. Sequence were collected from Protein Data Bank (PDB), UniProt, Fluorescent Protein Data Base (FPbase) and through Addgene plasmid sequence. You can find the data we found at this link: https://github.com/Carrie1013/DS596-ProteinProject/blob/main/raw$_s$*equence.csv*

2.2.2 Protein stability

The stabilities of protein sequence that we used for fine-tuning proteinBERT were taken from the Tasks Assessing Protein Embeddings (TAPE) [4], this is a benchmark for evaluating protein sequence models, and they use data generated by a novel combination of parallel DNA synthesis and protein stability measurements [4]. The datasets include protein sequences labeled with their stabilities.

Protein Stability	**Train**	**Valid**	**Test**
mean	0.18	0.18	1.00
std	0.57	0.57	0.41
min	-1.97	-1.78	-0.27
max	3.40	3.24	2.66
data count	48251	5362	12851

Table 1: Data description of protein stability

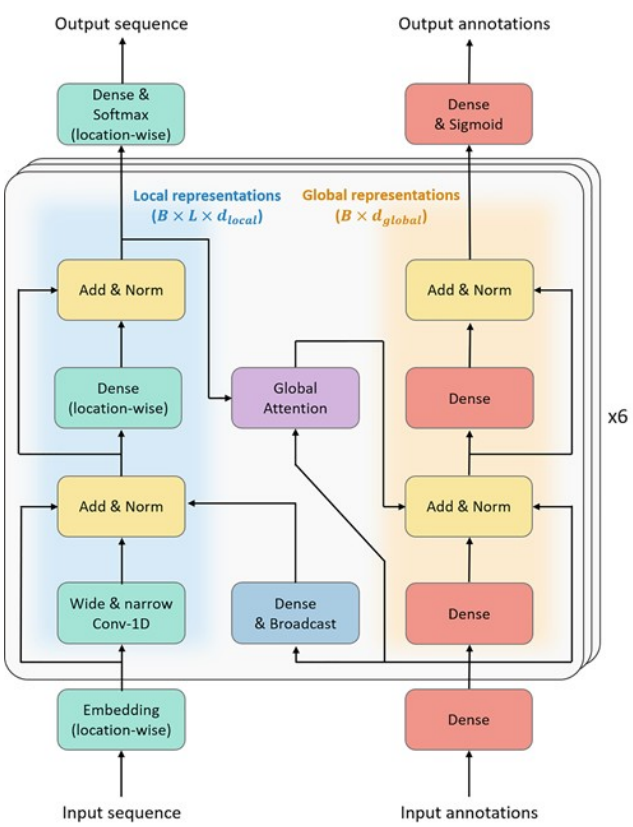
2.3 Principles underlying method

Sequence Encoding with proteinBERT

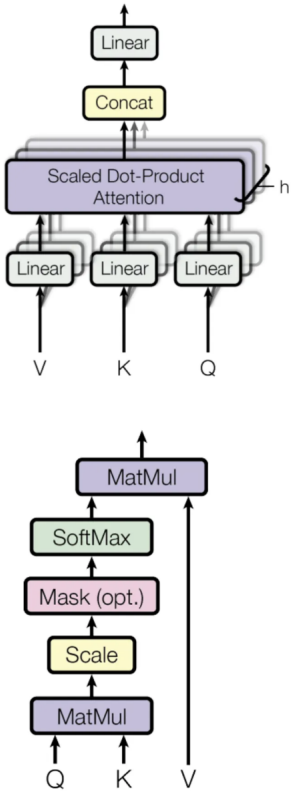
Sequence-based large-scale language models such as BERT are often used in natural language processing (NLP) tasks. There is a significant similarity between NLP and understanding biological sequences, protein sequences are made up of amino acids (e.g. M, V, S, K), which is similar to words in a sentence. Also, just like the meaning of words depends on their context in the sentence, the role of amino acids in

a protein depends on the residues around them. BERT's architecture is designed to capture context in text and is essentially suitable for understanding context in protein sequences. Therefore, many scientists have tried to apply similar language model structures to protein sequence-related tasks.

ProteinBert [2] is one of the sota work in this field. In this paper, ProteinBert is pre-trained on 106M proteins on two simultaneous tasks: bidirectional language modeling of protein sequences and gene ontology (GO) annotation prediction, the pre-training makes the model easily transferable between various protein-related tasks through downstream data sets. In addition, the transformer architecture of proteinBERT employs a self-attention mechanism to capture dependencies between amino acids in a protein sequence. Protein sequences often have long-distance dependencies due to their complex spatial structures, and ProteinBERT is able to capture these implicit relationships through its attention mechanism.



(a) Model structure for ProteinBERT [2].



(b) Structure of multi-head attention layer and scaled dot-product attention.

Figure 2: Model architecture and mechanism

Figure 3: Visualization of attention mechanism in protein sequences

Although inspired by BERT, proteinBERT's architecture is different and includes several innovations. ProteinBERT is a denoising autoencoder [2], the pre-trained model can be directly used for a variety of tasks, but due to the purpose of our project and the complexity of the input, we only use proteinBERT's encoder here in our project. Input a protein sequence, and the model outputs a contextual representation of the amino acid (local embedding) or a feature representation of the entire sequence (global embedding). ProteinBERT's design allows it to learn biologically meaningful patterns from sequences and adapt to a variety of protein-related tasks.

In out project, when we input the protein sequence $(1,)$, the encoder will output an array in shape $(1, seq\_len)$, where seq_len is a hyper-parameter. Below is an example of input and output of model encoder.

Sampled DataFrame:

```
0    MSKGEELFTGVVPILVELDG...
1    MSKGEELFTGVVPILVELDG...
2    MSKGEELFTGVVPILVELDG...
3    MSKGEELFTGVVPILVELDG...
4    MSKGEELFTGVVPILVELDG...
...
2940 MVSELKENMPMKLYMEGTVN...
2941 MVSELKENMPMKLYMEGTVN...
2942 MVSELKENMPMKLYMEGTVN...
2943 MVSELKENMPMKLYMEGTVN...
2944 MVSELKENMPMKLYMEGTVN...
Name: Sequence, Length: 2945, dtype: object
```

Encoded Sequence Array:

```
array([[23, 10, 15, ..., 25, 25, 25],
       [23, 10, 15, ..., 25, 25, 25],
       [23, 10, 15, ..., 25, 25, 25],
       ...,
       [23, 10, 18, ..., 25, 25, 25],
       [23, 10, 18, ..., 25, 25, 25],
       [23, 10, 18, ..., 25, 25, 25]], dtype=int32)
```

Secondary Structure Annotation

The proteinBERT provided a benchmark for secondary structure prediction, however, we found that their task was trained by various of short sequences, while our sequences are too long for this model. Therefore, for better performance and higher accuracy in the next step, we choose another two methods for this task.

We implemented secondary structure prediction with both the PSIPRED [1]and AlphaFold [3] method. Both of the methods have their goods and limitations. PSIPRED is a tool specifically designed to predict the secondary structure of proteins, relying primarily on lightweight models based on neural networks. The input sequence length and compute resources affect its run time, but it is usually done at the minute level and can be written to the code pipeline using APIs. AlphaFold is a 3D structure prediction tool that provides accurate protein models. While generating these models could require significant computational resources and time, AlphaFold could deliver results with high level of accuracy. The output from AlphaFold is provided in **.cif** format. To extract secondary structure information, the DSSP (Dictionary of Secondary Structure of Proteins) algorithm can be applied, which analyzes the structure to determine elements such as alpha helices, beta sheets, and other structural features.

The output secondary structures of these models are in string format, so we need to encode this sequence. Since all elements of secondary structure are letters representing structure categorial information, we choose to use one-hot encoding in this step. Following is a part of GFP protein' secondary structure prediction output.

MSKGEELFTGVVPILVELDGDVNGHKFSVSGEGDATYGKLT...
sample protein sequence

CCCCHHHHCCEEEEEEEEEEEEEEECCCEEEEEEEECCCCC...
sample secondary structure output

Sequence Stability Calculation

We suspect that when a position on the protein sequence is a split protein, the splitted protein is more likely to reassemble because the two subsequent strands generated by the split may have a lower stability state. With this in mind, we found a stability dataset for proteins and used it to fine-tune

proteinBERT so that the model can predict the approximate stability of arbitrary protein sequences. Each split site is divided into two sequences around the site, and the stability of the two sub-sequences is predicted respectively, and the sum is used as the final split site prediction feature. Due to computational complexity, this feature can be chosen not to be used in long sequence prediction scenarios.

2.4 Models and Parameters

An important feature of ProteinBERT is sequence length flexibility. To avoid the risk of overfitting the model to a particular constant length, we found the method by switching the coding length of the protein sequence, using 128, 512, or 1024 labels. The most appropriate $seq = 512$. The loss function minimized by ProteinBERT during pretraining

$$\text{Loss} = -\sum_{i=1}^{l} \log\left(\hat{S}_{i,S_i}\right) - \sum_{j=1}^{8943}\left(A_j \cdot \log\left(\hat{A}_j\right) + (1 - A_j) \cdot \log\left(1 - \hat{A}_j\right)\right)$$

where l is the sequence length, $S_i \in \{1, \ldots, 26\}$ is the sequence's true token at position i, $\hat{S}_{i,k} \in [0, 1]$: the predicted probability that position i has the token k (for any $k \in \{1, \ldots, 26\}$), $A_j \in \{0, 1\}$is the true indicator for annotation j (for any $j \in \{1, \ldots, 8943\}$), $\hat{A}_j \in [0, 1]$ is the predicted probability that the protein has annotation j.

And the the loss function minimized by ProteinBERT that we use during fine-tuning on stability

$$\text{Loss} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where N is the sample size, y_i is the true label of protein stability, and $\hat{y}_i$ is the predicted label of protein stability. As for our final MLP classifier to define whether the site can be a split site or not, we use log loss function and ADAM optimizer, where

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(\hat{y}_{ij})$$

So, through the pipeline, we fine-tuned the proteinBERT and trained the MLP classifier, to get a higher performance, data standardization and dimensional reduction were implemented before the features were added to the classifier. The parameter details of models are here below:

Param Name	**Task**	**Default Value**
`seq_len`	Fine-tune the maximum length of the input sequence.	512
`batch_size`	Fine-tune batch size for each training.	32
`max_epochs_per_stage`	Fine-tune maximum number of rounds for freezing layer training.	40
`lr`	Fine-tune learning rate (for unfrozen model layers).	$1e^{-4}$
`max_iter`	Max MLP classifier training iteration.	500
`seq_len`	Length of encoded sequence to input MLP.	512
`hidden_layer_sizes`	MLP hidden layer size.	(128, 64)
`test_size`	Test size for MLP training.	0.2
`pca_components`	The dimensions added to the model were reduced using PCA.	80

Table 2: Parameter description and default values for models.

3 Results

Stability prediction task

Although the loss function is defined above, since the label is numerical but not binary and we can not evaluate it by metrics like accuracy, we use Spearman's rank correlation here to show how our model is. From the figure below, we can see the correlation is very high and close to 1, so the model is definitely performing well.

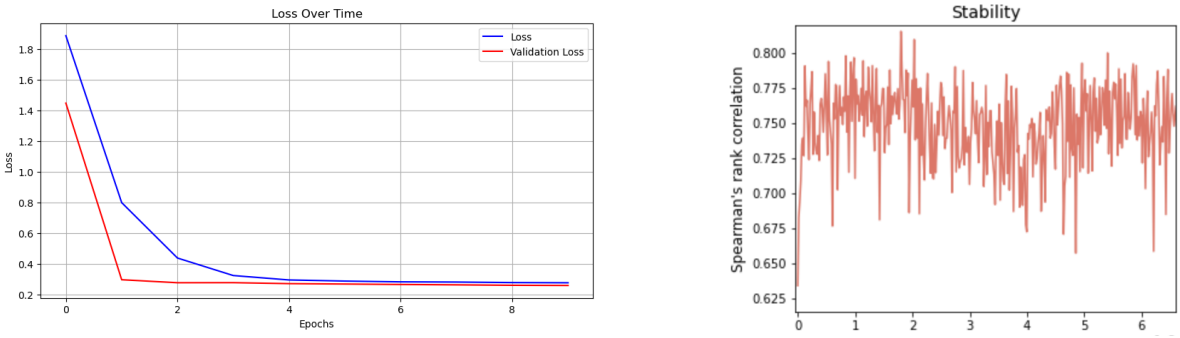


Figure 4: evaluation for stability prediction

Split site prediction task

Due to the sparsity of the data, running the model classification directly may lead to serious overfitting problems, so we used sampling technology before running, respectively fitting the classifier on the unsampled data, resambled data and undersampled data. In addition, we also conducted training on data that did not contain stability features, and the results obtained are shown in the following table.

Dataset	Class	Precision	Recall	F1-score	Support	Accuracy	Macro Avg F1
unsampled	0	0.98	1.00	0.99	795	0.98	0.49
	1	0.00	0.00	0.00	17		
over-sampled	0	1.00	0.84	0.91	794	0.92	0.92
	1	0.87	1.00	0.93	810		
down-sampled	0	0.88	0.64	0.74	11	0.76	0.77
	1	0.69	0.90	0.78	10		

Table 3: Classification Report Comparison for Datasets without stability feature.

Dataset	Class	Precision	Recall	F1-score	Support	Accuracy	Macro Avg F1
unsampled	0	0.97	1.00	0.99	573	0.97	0.49
	1	0.00	0.00	0.00	16		
over-sampled	0	1.00	0.90	0.95	579	0.96	0.95
	1	0.92	1.00	0.96	677		
down-sampled	0	0.40	0.50	0.44	4	0.71	0.62
	1	0.83	0.77	0.80	13		

Table 4: Classification Report Comparison for Datasets with stability

As can be seen from the table, it is obvious that the model shows overfitting phenomenon on the unsampled data set. Finally, we used the model fitted on the undersampled data set with stability characteristics to evaluate the test set, and the prediction accuracy on 20 pieces of data was about 75%, indicating that it was an accurate model.

4 Discussion

- How do your results tie together into a story?

- How do your results answer your research question?

Limitation of the project: 1. The dataset is too small. Could worked harder to generate the dataset 2. some data were excluded. in reality sequence could have overlap region or deleted region to recombine 3. need more feature to study. stability might not be that

Possible direction: 1. expand the dataset to include other kinds of split protein, not limited by FP to have a larger set to train 2. learn

5 Conclusion

Code Availability

Our code repository can be found in https://github.com/Carrie1013/DS596-ProteinProject

References

[1] Buchan, D. W. A., & Jones, D. T. (2019). The PSIPRED Protein Analysis Workbench: 20 years on. *Nucleic Acids Research*, *47*(W1), W402–W407. https://doi.org/10.1093/nar/gkz297

[2] Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2022). ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8), 2102–2110. https://doi.org/10.1093/bioinformatics/btac020

[3] Jumper, J. M., et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, *XXX*(XXXX), XXX–XXX. https://doi.org/10.1038/s41586-024-07487-w

[4] Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. (2019). Evaluating Protein Transfer Learning with TAPE. *Advances in Neural Information Processing Systems*, **32**, 9689–9701. Available at: https://pmc.ncbi.nlm.nih.gov/articles/PMC7774645/.

[5] Anson, F., Kanjilal, P., Thayumanavan, S., & Hardy, J. A. (2021). Tracking exogenous intracellular casp-3 using split GFP. *Protein Science*, 30(2), 366–380. https://onlinelibrary.wiley.com/doi/10.1002/pro.3992.

[6] Cabantous, S., Nguyen, H. B., Pedelacq, J.-D., Koraïchi, F., Chaudhary, A., Ganguly, K., et al. (2013). A new protein-protein interaction sensor based on tripartite split-GFP association. *Scientific Reports*, 3(1), 2854. https://www.nature.com/articles/srep02854.

[7] Cabantous, S., Terwilliger, T. C., & Waldo, G. S. (2005). Protein tagging and detection with engineered self-assembling fragments of green fluorescent protein. *Nature Biotechnology*, 23(1), 102–107. https://doi.org/10.1038/nbt1044.

[8] Callahan, B. P., Stanger, M. J., & Belfort, M. (2010). Cut and glow: Protease activation of split green fluorescent protein. *Chembiochem*, 11(16), 2259–2263. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3711563/.

[9] Feng, S., Sekine, S., Pessino, V., Li, H., Leonetti, M. D., & Huang, B. (2017). Improved split fluorescent proteins for endogenous protein labeling. *Nature Communications*, 8(1), 370. https://www.nature.com/articles/s41467-017-00494-8.

[10] Ghosh, I., Hamilton, A. D., & Regan, L. (2000). Antiparallel leucine zipper-directed protein reassembly: Application to the green fluorescent protein. *Journal of the American Chemical Society*, 122(23), 5658–5659. https://pubs.acs.org/doi/10.1021/ja994421w.

[11] Kakimoto, Y., Tashiro, S., Kojima, R., Morozumi, Y., Endo, T., & Tamura, Y. (2018). Visualizing multiple inter-organelle contact sites using the organelle-targeted split-GFP system. *Scientific Reports*, 8(1), 6175. https://www.nature.com/articles/s41598-018-24466-0.

[12] Kamiyama, D., Sekine, S., Barsi-Rhyne, B., Hu, J., Chen, B., Gilbert, L. A., et al. (2016). Versatile protein tagging in cells with split fluorescent protein. *Nature Communications*, 7(1), 11046. https://www.nature.com/articles/ncomms11046.

[13] Kojima, T., Karasawa, S., Miyawaki, A., Tsumuraya, T., & Fujii, I. (2011). Novel screening system for protein–protein interactions by bimolecular fluorescence complementation in *Saccharomyces cerevisiae*. *Journal of Bioscience and Bioengineering*, 111(4), 397–401. https://www.sciencedirect.com/science/article/pii/S1389172310004172.

[14] Kost, L. A., Putintseva, E. V., Pereverzeva, A. R., Chudakov, D. M., Lukyanov, K. A., & Bogdanov, A. M. (2016). Bimolecular fluorescence complementation based on the red fluorescent protein FusionRed. *Russian Journal of Bioorganic Chemistry*, 42(6), 619–623. https://doi.org/10.1134/S1068162016060054.

References