

Supplementary Material

a Predict protein-protein binding site

⚠ Since extracting ProtT5 and PSSM is resource intensive, we will guide you through this prediction

1. Extract ProtT5 features via colab or your own device.

1. Extract ProtT5 features via colab or your own device.

Case - 1 Case - 2 Case - 3

If you want to extract features through your own device, then follow these steps:

- Create a virtual environment with the following code:

```
conda create env -n prott5
conda activate prott5
pip install requirement.txt
```
- Download the *requirement.txt* and code we provide via the following link:
[DOWNLOAD](#)
- Open our code via *vscode* or the *jupyter* provided by *Anaconda*

1. Extract ProtT5 features via colab or your own device.

Case - 1 Case - 2 Case - 3

If you want to extract features through *colab*, then follow these steps:

- Jump to *colab* and run it directly through the following link
[COLAB](#)
- Copy one sequence at a time to replace the *example* in the code

b 3. Upload the prepared file.

3. Upload the prepared file.

Upload ProtT5 Upload PSSM Submit

Upload your ProtT5 feature, it must be *.npy* and the dimension is [1, length, 1024]

Drag & drop files here ...

Select file... [Browse ...](#)

3. Upload the prepared file.

Upload ProtT5 Upload PSSM Submit

Upload your PSSM feature, it must be *.pssm* .

Drag & drop files here ...

Select file... [Browse ...](#)

c Download Datasets & Code & Model

[Download Datasets](#) [Download Code](#)

We provide 5 datasets for download, respectively, Dset_72, Dset_164, Dset_186, Dset_448 and Dset_880

We provide the code for training, prediction and feature extraction with instructions for use.

[DOWNLOAD NOW](#) [DOWNLOAD NOW](#)

Supplementary Figure 1 Schematic diagram of web server functional modules. (A) Guidance Control: Guides users for extracting ProtT5 in defined situations. (B) Submit Form: The user can submit ProtT5 and PSSM for the server to complete the prediction. (C) Download Module: Provides high-speed download links for data sets, code, and pre-trained models.

Supplementary Table 1. Details of the training and test sets of our study.

	Dataset	ResiduesNumber	Binding Ratio %
Train	Dset_843	450 598	10.14
	Dset_186	36 219	15.23
Test	Dset_72	18 140	10.60
	Dset_164	33 681	18.10
	Dset_448	116 500	13.57

Supplementary Table 2. Hyperparameter candidates and optimal parameter combinations for the traditional machine learning algorithms.

Algorithms	Parameter Candidate Values	Optimal parameters
XGBoost	{'max_depth': [3, 4, 5, 6, 7, 8], 'eta':[0.001, 0.01, 0.1], 'gamma':[0, 0.001, 0.01], 'n_estimators':[100, 500, 1000]}	[4, 0.01, 0, 500]
CatBoost	{'depth': [5, 6, 7], 'learning_rate' : [0.01, 0.03, 0.1], 'l2_leaf_reg': [1, 4, 9], 'iterations': [100, 300, 1000]}	[7, 0.1, 9, 300]
LGBM	{'n_estimators':range(100, 500, 100), 'learning_rate':[0.01, 0.05, 0.1], 'feature_fraction':[0.7, 1]}	[400, 0.1, 0.7]
SGD	{'loss':['log', 'modified_huber'], 'penalty':['l2', 'l1', 'elasticnet'], 'alpha':[0.0001, 0.001, 0.01]}	['huber', 'elasticnet', '0.001']
MLP	{'solver':['lbfgs', 'sgd', 'adam'], 'alpha':[1e-5, 1e-4, 0.001, 0.01]}	['lbfgs', 1e-5]