Predicting Enzyme Thermostability Based on Protein Sequence

Shukun Liu, Yizhou Zhang

8th December 2022

1 Introduction

Proteins can be identified by their amino acid sequences and their 3-D structures. Enzymes are a kind of proteins that can catalyze specific chemical reactions, making the reactions happen more easily. They are significantly helpful in accelerating or controlling the production process of biochemical products. Improving the thermostability of enzymes is always a hot topic in biological research, as a tiny change in temperature can result in a large fluctuation in the enzyme performance, bringing plenty of troubles to modern mass production.

Our project originates from an ongoing competition on Kaggle that is hosted by Novozymes, a biotechnology company. In this competition, participants need to predict and rank the thermostability of enzymes coming from various mutations of wild-type enzymes. The prediction is expected to be based on the amino acid sequences and 3-D structures of the enzyme proteins. The competition host only cares about how to rank the thermostability of test samples, instead of the detailed values of the predicted melting temperatures.

Our goal is to extract useful information from the amino acid sequences that are related to the thermostability and apply some optimized simple models to make predictions. In other words, feature engineering is the focus of this project, and 3-D structures are not considered as inputs of the models. XGBoost is selected as our model to do the training and predictions.

The related works, datasets, feature extraction, generation of the design matrix, and experiment results will be introduced in the following part.

2 Related Works

The machine-learning-based exploration of the relationship between the sequence and the property of the protein has been a popular topic for a long time. Some research studies combine the statistical potential analysis with neural networks (Pucci et al., 2016).

According to the summary work by Marabotti et al. (2021), machine learning models such as SVM, random forest, and graph neural network are widely used to predict the catalytic capacity or the thermal stability of enzymes based on their 3-D structure or the acid sequence. A closer example is the work by Miotto et al. (2022), which predicts enzyme thermostability by representing enzyme proteins as energy-weighted graphs and comparing them using ensembles of random interaction networks.

There are also some studies making use of the natural language processing models to predict protein properties. With a deep convolutional network, Khare et al. (2022) compared the performance of a

small transformer model, which was constructed from scratch, with the performance of a well pre-trained large model ProtBERT. In their work, they only used the acid sequence as their model input.

In this project, the performance of the tree-based model XGBoost will be explored, and the prediction will be primarily based on the enzyme protein amino acid sequences.

3 Data

Our data is provided by Novozymes on Kaggle. The training dataset includes the amino acid sequences of enzymes (in the form of "ACCBADTT..", where each capital letter represents one of the 20 amino acids), the pH values of the experiment environment under which the thermostability is measured, and the melting temperatures of enzymes which represent the thermostability of enzymes.

As it is pointed out by the host of the competition, many protein sequences in the training dataset can be identified as the mutation results from some wildtype enzyme proteins. The mutations include substitutions and deletions (no insertion). Some sequences originate from the same wildtype, and therefore we can group the training enzyme samples by their wildtype enzyme protein sequences. The real wildtype sequence of each group may exist or not exist in the training dataset. Also, there may be multiple mutations between the wildtype and its one variant sequence.

The training dataset consists of 28981 samples. After the grouping process, about 4000 samples are left, and the precise number depends on which grouping strategy is applied. Namely, about 25000 samples do not belong to any group. The number of groups also varies from 76 to 157, decided by the grouping strategies which are introduced in detail below. It is found that the number of mutations between a wildtype sequence and one of its variants is either one or two.

The test dataset contains enzymes that are all variants of one specific wildtype protein sequence. Each of the test samples only comes from one mutation of the wildtype sequence. The wildtype sequence exists in the test dataset. The pH values of all the test samples are the same. There are 2413 test samples in total.

4 Features & Methods

4.1 Features & preparations

The first and most important thing is to clarify how the prediction and rankings are made on the test dataset, meanwhile making it applicable to train a model with the training dataset. Our primary method is to find the wildtype enzyme protein sequence of the test dataset, and generate pairs in the form [(test_wildtype, test_sample1), (test_wildtype, test_sample2), ...]. In each pair, we should quantify the differences between the 2 sequences, extract useful information about the proteins and the mutation, and predict the difference between the melting temperatures of the 2 sequences. Finally, all the test samples can be ranked by their predicted melting temperature differences from the wildtype sequence. This process can be illustrated in Fig.1.

About extracting features from the protein sequence pairs, two features are used. The simple one is the category of the mutation happening between the wildtype sequence and the sample sequence. There are 400 mutations in theory, as one amino acid may be substituted by one of the other 19 amino acids or be deleted. The categories are represented by arrays where each number represents the number of

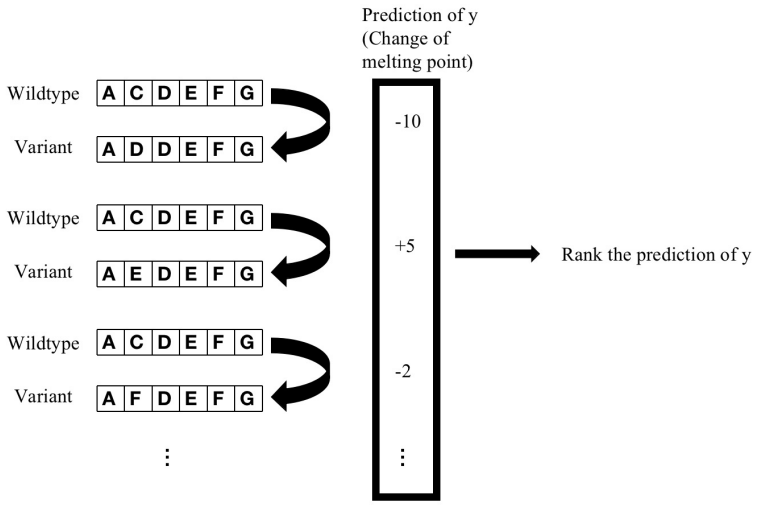

Figure 1: Brief explanation of how we predict

occurrences of a specific mutation. E.g., [0 0 ... 0 1 0... 0 1 0 0] represents 2 different mutations. When there is only one mutation, the array would be similar to a one-hot-encoded categorical variable.

The other feature was inspired by the research by Kumar et al. (2000), which found that proteins in thermophiles would evidently avoid specific amino acids in their alpha-helix structures. That means the proportions of amino acids of a small part of the amino acid sequence may reflect its structural role in the whole sequence. As the structure of a protein is important to its thermostability, it can be inferred that a mutation that changes the structure of the protein may greatly influence the thermostability. Therefore, we propose to make use of the proportions of acids in the neighborhood of the mutation position. This feature should be an array with a length equal to 20, e.g. [0.15 0 0.25 0 ... 0.1], as there are 20 kinds of amino acids. The radius of the neighborhood around the mutation spot is thus a parameter.

The acid proportions of the whole amino acid sequence were actually tried as a feature in **Method 2**. This will be explained later.

Before training, it is noteworthy that 3 grouping strategies were used in this project. The first one is to group the samples by only considering the similarities among the protein sequences. The second grouping strategy additionally requires that the samples in the same group should have the same pH values, and the number of mutations between the wildtype and a variant can only be one. As for measuring the similarity of two sequences, Levenshtein distance, which is also called edit distance, was used. This metric counts the minimal number of operations including deletion and substitution to make two stings equal. The third strategy comes from the work of a participant in the competition, which divides each sequence into 3 parts evenly and makes global comparisons (R. Hatch, 2022).

Based on the principles above, we put forward two types of training for the grouped training samples. **Method 1** is to find a referential sequence for each group and generate the reference-variant pairs directly for the extraction of training features. **Method 2** is to generate variant-variant pairs within the group, which means that one sample will be compared with all the other samples in the same group to extract features. The details are presented in the Method section.

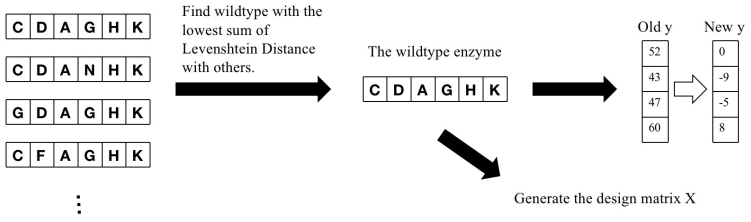

Figure 2: Brief explanation of the process of Method 1

4.2 Method 1

4.2.1 Data preprocessing

Before generating the design matrix, we need to specify the wildtype. In the training dataset, within each group, we find the enzyme with a sequence which has the smallest sum of Levenshtein difference with all other enzymes' sequences and treat it as the wildtype to find the mutation. In the test dataset, since each variant has only one mutation, we are able to find the wildtype by setting the most frequent amino acids in each position as the sequence of the wildtype.

We have two kinds of features, the mutation type and the proportion of amino acids of the neighboring sequence of mutation. Since the variants in the training sample may have more than 1 mutation, we set the number of each mutation type as the first part of our design matrix. The second part of the design matrix is the proportion of the amino acid type in the neighboring sequence of mutation with a radius of 20. The design matrix is bound by these two parts by columns.

We set the difference between melting temperatures of the wildtype enzyme and each variant as the new y.

4.2.2 Prediction of test data

We conducted the same method to generate the design matrix for test data. We predict the difference between wildtype and variants and rank the values. This rank is the same as the rank of melting temperatures.

4.3 Method 2

4.3.1 Data preprocessing

To extract features for training, we compare each sample with all the other samples (of different sequences) in the same group to extract features. For example, if a group contains n samples and their protein sequences are unique, then n(n+1)/2 times of comparisons will be made, and a design matrix of n(n+1)/2 rows will be generated by finding the mutation type and the acid proportions around the mutation position in each pair.

In this method, we tried two kinds of regression targets, the melting temperature difference, and the temperature ratio.

Besides, we found that there are at most 2 mutations in each pair. Based on that, when there is a pair of sequences with 2 mutations, our solution is to divide this pair of sequences into 2 pairs of sequences, each of which will represent one mutation respectively. For example, ("$AABCDEE$", "$AABFEE$"), the mutation from "CD" to "F", will be transformed into ("$AABCDEE$", "$AABDEE$") and ("$AABCDEE$",

"$AABCFEE$"). The situation of deletion has a higher priority here. Then, each of these 2 pairs will get one-half of the original temperature difference.

Also, as the one-hot-encoded design matrix normally has more than 800,000 rows and more than 400 columns, which is unaffordable to the RAMs to our machines, we set that bar that the sequence pairs in the training set are considered only when the melting temperature differences are larger than the 20% of one of the two temperatures. This effort reduced the row number of the design matrix to about 300,000.

4.3.2 Prediction of test data

In the predictions on the test dataset, it is possible that the mutation type of a sequence pair in the test dataset has never appeared in the training dataset, making the one-hot-encoder trained on the training dataset invalid for this test sample. In this case, all the numbers in the feature array for the mutation category will be set as 0.

4.4 Learning algorithm: XGBoost

4.4.1 Description of XGBoost

XGBoost is an open-source software library that implements its machine learning algorithms based on Gradient Boosting framework. The step of this algorithm of XGBoost can be explained in three steps:

Suppose we have training set $\{(x_i, y_i)\}_{i=1}^{N}$, and we set the loss function as $L(y, f(x))$.

Step **1**: Start training the model from a constant value that minimizes the loss function.

$$\hat{f}_{(0)}(x) = argmin_{\theta}(L(y, \theta))$$

Step **2**: Solve the optimization problem that finds the learner(i.e. tree) that minimizes the loss function. Set $\phi(x)$ as the result of the optimization problem, make $\hat{f}_i(x) = \alpha * \phi(x)$, where α is the learning rate. Update the model by

$$\hat{f}_{(i)}(x) = \hat{f}_{(i-1)}(x) + \hat{f}_i(x)$$

We iteratively conduct Step 2 until the loss function of the validation set is not decreasing.
Step **3**: Once the stopping criterion is satisfied, the final model is formed. The model is

$$\hat{f}_{(M)}(x) = \hat{f}_{(M-1)}(x) + \hat{f}_M(x) = \sum_{i=1}^{M} \hat{f}_i(x)$$

4.5 Evaluation of models

The models are evaluated based on how accurately it ranked the thermostability of enzymes. The Spearman's rank correlation coefficient is defined as the Pearson correlation coefficient between the rank variables. The formula is shown below:

$$\rho = 1 - \frac{6\Sigma d_i^2}{n(n^2 - 1)}$$

5 Experiments

5.1 Implementation of Method 1 & 2

We use XGBoost to construct our regression model. We randomly separate 10% rows of the training desgin matrix out as the validation set. The XGBoost model will stop training if the result on the validation set is not improving. Additionally, XGBoost is able to accept some parameters including the maximum depth of the tree, the size of the sub-sample for training, and the number of features for training. These measures can prevent the model from overfitting and lessen the variance.

We conduct a grid search for **Method 1** on the following selection of parameters: learning rate, max depth of trees, the size of sub-sample, and the number of features. A training process consists of 10,000 boost rounds usually cost 3 minutes on the online computational platform Colab.

The values we selected for each parameter:

Learning rate	0.001	0.005	0.01
Max depth	3	4	5
Size of sub-sample	0.5	0.6	0.7
Number of features	0.2	0.25	0.3

The result shows that when learning rate equals 0.001, max depth equals 5, size of sub-sample equals 0.6, and number of features equals 0.3, the performance on the validation set is the best.

Meanwhile, **Method 2** is not suitable for grid search, as the design matrix generated in this method is too large. A training process consisting of 3000 boost rounds would typically cost 2 hours on Colab. We manually tested the models with learning rate 0.001, 0.005, max depth 4, 5, 6, size of sub-samples 0.6, number of features 0.2, mutation neighborhood radius 15, 25, max boost round 1500, 3000.

In **Method 2**, it is found that adding acid proportions of the whole amino acid sequence as a new feature would worsen the performance of the model. Setting the regression targets as the pair temperature ratios would give terrible performance.

All three grouping strategies were used for both methods, among which the third strategy inspired by R. Hatch generally gives the best performance.

With the third grouping strategy, we found the parameters' combinations that gives the highest Spearman's rank correlation coefficients on the test dataset shown below:

method	max depth	sub-sample size	feature num	learning rate	max boost round	score
method 1	5	0.6	0.3	0.001	10000	0.192
method 2	4	0.6	0.2	0.005	3000	0.171

5.2 Huge performance gap between validation set and test set

For **Method1**, we rank the validation set based on their true melting temperature values, and tested our model on it. The Spearman's rank correlation coefficient acquired is higher than 0.7, which is astonishing compared with the result on the test set. This phenomenon is explored as follows.

As we stated in the Data section, our test set contains variants of one wildtype enzyme, but our training set contains variants of extensive kinds of wildtype. We classified the training set into groups based on

the wildtype of each sample. However, when we split up the training set into the sub-training set and validation set, samples from different groups are mixed together. In this situation, a group may have samples in both the training set and the validation set. Consequently, the result on the validation set cannot represent the result of the test set, as the test set is totally isolated from the training set.

Then we improved our **Method1** model by generating the validation set based on the group number. We randomly chose some groups to be completely put into the validation set. We trained our model for multiple times, and the result of validation set never exceed 0.2, which is consistent with the result of test set.

From this, we concluded that our model performs very well if the training samples include variants that have the same wildtype as the test samples. However, our model cannot make a significantly worse prediction on variants generating from a new wildtype. This result is reasonable because there are differences in the structure of enzyme between different groups. We are unable to analyze the 3D structure so it is very likely that we lost some very important information.

6 Conclusion and Future Work

Our team has tried to predict protein thermostability in two different ways. Our first model is to find a referential sequence for each group and generate the reference-variant pairs directly for the extraction of training features. Our second model is to generate variant-variant pairs within the group. By applying the XGBoost model to our data, both of our models achieve Spearman's rank correlation coefficients near 0.2. These results show that our methods caught some useful information to predict thermostability.

Furthermore, we found that our models perform significantly better if the test samples and some of the training samples have the same wildtype. As a result, our models can provide a very good prediction on the thermostability of enzymes if we include some enzymes that have the same wildtype.

However, there is a lot of room for improvement. We need to enhance our model's performance on samples that have a new wildtype. We may extract some valuable information from the 3-D structure of enzymes, with assistance of outer data sources. With more time We believe that our models will be significantly improved with a deeper analysis of the 3-D structure of enzymes.

7 Contributions

Shukun Liu: Literature review. Grouping of enzymes. Implementation of Method 2. Report writing.

Yizhou Zhang: Implementation of Method 1. Report writing.

8 Reference

Hatch, R. (2022). *NOVO: Train Data Contains Wildtype Groups*. Kaggle. https://www.kaggle.com/code/roberthatch/nov train-data-contains-wildtype-groups?scriptVersionId=107371193cellId=1

Khare, E., Gonzalez-Obeso, C., Kaplan, D. L., Buehler, M. J. (2022). CollagenTransformer: End-to-End Transformer Model to Predict Thermal Stability of Collagen Triple Helices Using an NLP Approach. *ACS Biomaterials Science Engineering, 8*(10), 4301-4310.

Kumar, S., Tsai, C.J., Nussinov, R. (2000). Factors enhancing protein thermostability. *Protein Engineering, Design and Selection, 13*(3), 179-191. https://doi.org/10.1093/protein/13.3.179

Marabotti, A., Scafuri, B., Facchiano, A. (2021). Predicting the stability of mutant proteins by computational approaches: an overview. *Briefings in Bioinformatics, 22*(3), bbaa074.

Miotto, M., Armaos, A., Di Rienzo, L., Ruocco, G., Milanetti, E., Tartaglia, G. G. (2022). Thermometer: a webserver to predict protein thermal stability. *Bioinformatics, 38*(7), 2060.

Pucci, F., Bourgeas, R., Rooman, M. (2016). Predicting protein thermal stability changes upon point mutations using statistical potentials: Introducing HoTMuSiC. *Scientific reports, 6*(1), 1-9.