A Sequence Embedding Method For Enzyme Optimal Condition Analysis

Xiangjun Li
Shandong University
Zhixin Dou
Shandong University
Yuqing Sun (sun_yuqing@sdu.edu.cn)
Lushan Wang
Shandong University
Bin Gong
Shandong University

Methodology article

Keywords: Protein sequence analysis, Embedding, Bioinformatics

Posted Date: November 4th, 2019

DOI: https://doi.org/10.21203/rs.2.16793/v1

Version of Record: A version of this preprint was published at BMC Bioinformatics on November 10th, 2020. See the published version at https://doi.org/10.1186/s12859-020-03851-5.

METHODOLOGY ARTICLE

A Sequence Embedding Method For Enzyme Optimal Condition Analysis

Xiangjun Li[1†], Zhixin Dou[2†], Yuqing Sun[1*], Lushan Wang[2] and Bin Gong[1]

[*]Correspondence:
sun_yuqing@sdu.edu.cn
[1]School of Software, Shandong University, Shunhua Road, 250101 Jinan, China
Full list of author information is available at the end of the article
[†]Equal contributor

Abstract

Background: An enzyme activity is influenced by the external environment condition. It is important to have an enzyme remain high activity in a specific condition. A usual way is to first determine the optimal condition of an enzyme by either the gradient test or by tertiary structure, and then to use protein engineering to mutate a wild type enzyme for a higher activity in an expected condition.

Results: In this paper, we investigate the optimal condition of an enzyme by directly analyzing the sequence. We propose an embedding method to represent the amino acids and the construct information as vectors in the latent space. These vectors contain information about the correlations between amino acids and sites in the aligned amino acid sequences, as well as the correlations with the optimal conditions. We crawled and processed the amino acid sequence in glycoside hydrolase GH11 family, and got 125 amino acid sequences with optimal pH condition. We used probabilistic approximation method to implement the embedding learning method on these samples. Based on these embedding vectors, we design a computational score to determine the optimal condition for an enzyme and achieves the accuracy 80% on the test proteins in the same family. We also give the mutation suggestion such that it has a higher activity in the expected environment, which is consistent with the professional wet experiments and analysis.

Conclusion: A new computational method is proposed for the sequence based enzyme optimal condition analysis. Compared with the traditional process that involves a lot of wet experiments and requires multiple mutations, this method can get the desired protein for an expected condition in an efficient and effective way.

Keywords: Protein sequence analysis; Embedding; Bioinformatics

Background

Proteins are made up of hundreds of monomers called amino acids that are attached to one another by peptide bonds, forming a long chain defined as primary structure. It further constructs the second structure and the tertiary structure of the protein, which finally determines the protein function and the optimal condition of its activity [1], such as the acid resistance, or the salt tolerance.

Enzymes are a kind of catalytic proteins. The enzymes with the same function are referred as a family, which have some identical or conserved amino acid fragments in sequences that are not easy to be mutated. Although these enzymes have the same biological function, some of them are more active than others at the same alkaline or temperature condition, which are caused by the different parts in the sequences,

defined as the non-conserved fragments. For example, the enzymes in GH11 family can degradate the heteroxylans that constitute the lignocellulosic plant cell wall, but they have different optimal temperatures or pH [2]. In many practical applications, we need to find the most active enzyme in a family under a given condition or to mutate an enzyme for a higher activity under an expected condition than the existing enzymes. For these purposes, biological researchers generally perform a series of gradient tests to measure the optimal condition of each enzyme in a family. Then they adopt the protein engineering to produce an enzyme with an expected optimal condition [3]. But the above biological methods analyze only an enzyme at each wet experiment such that they cost much time, power and resources to get the expected enzyme.

Recently, many works adopt machine learning methods to predict the optimal temperature or pH for enzymes [4]. For example, Dijk et al. use the ratio between the number of residues inside and outside the folding structure of protein as the measure for its hydrophobicity so as to induce the temperature dependence of the protein [5]. Pucci et al. use temperature-dependent statistical potentials to compute the fold free energies at three different temperatures, and use Gibbs-Helmholtz equation to predict the protein thermal stability curve [6]. The SCooP method predicts the full T-dependent stability curve by protein structures and host organisms [7]. However, these methods require the tertiary structure [8], free energy [9], etc., which are not easy to obtained in many cases. Another representative method is based on the correlations between the tertiary structure and free energy [1] of an enzyme. It judges the stability of the mutant by analyzing the change of free energy in this process [10]. For example, Dehouck et al. use a linear combination of statistical potentials whose coefficients depend on the solvent accessibility of the mutated residue to predict the stability changes caused by single site mutations in proteins [11]. Wijma et al. create a library of potentially stable mutations by calculating the change in the free energy of mutations, reduce the size of the library by eliminating false positive predictions and choose the most stable combination of mutations to mutate [12]. These methods are based on molecular dynamics to simulate the state of proteins such that they perform a single protein in one experiment. But it can not analyze a group of proteins at the same time.

In this paper, we propose an embedding method to predicate the optimal condition of an enzyme directly stemming from the amino acid sequence. Amino acids and the construct information in sequence are represented as vectors in the same latent space, which are learned by the compatibility between the sites and expected condition. We propose the compatibility objectives on both a single-site with an amino acid and on multiple sites with different amino acids. Using these vectors, we can predict the optimal condition for a new amino acid sequence. We select the enzymes in the GH11 family as the practical usage, whose optimal pH are already determined. Since from the view of machine learning the quantity of samples is small, we use the statistics to approximate the probability distribution on amino acids on each site of the aligned sequences and generate more samples for the embedding training.

[1] A physics concept to describe the force that causes chemical reactions.

Based on these embeddings, we analyzed the non-conserved segment and the conserved segment of these enzymes. These results are consistent with the professional results on WebLogo[2] [13, 14]. We then design two experiments for biological purposes. One is to compare two enzymes which has a higher optimal pH or which has a higher activity in the same pH environment. Another is to quantify the probability on whether one or more sites mutation on a given enzyme would result it still active in an expected optimal condition. The experimental results are consistent with the biological results. Comparing with other methods, the embedding method is more efficient and effective.

Notions and Dataset

Notions

There are 20 kinds of amino acids known in nature. Let aa denote the set of amino acids in bioinformatics, i.e. $aa = \{Ala, Arg, Asp, Cys, Gln, Glu, His, Ile, Gly, Asn, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val\}$. Since the amino acid sequences in a family are often with different lengths, they are aligned before analysis. After alignment, the identical or similar segments are in the successive columns, which are helpful to find the functional or evolutionary structure. We denote the gap between amino acids as a symbol $'-'$ and the set of elements consisting an aligned amino acid sequence are replaced by $A = \{-\} \cup aa$. Given a family of amino acid sequences, denoted by $\mathbf{F}$, where $\mathbf{F}$ represents the familay of amino acid sequences, the length after alignment is denoted by l. For an amino acid sequence, we use k to indicate the k_{th} site of the sequence, c_k to represent the amino acid at the k_{th} site, and c_k^a for the amino acid of the specific amino acid sequence a at site k.

In this paper, We define the event (k, a) as amino acid $c_k^a \in A$ being on the $k^{th}, k \in [1..l]$ site of an aligned sequence $a \in \mathbf{F}$, and introduce the metric of *capability score* $s(k, a)$ to evaluate how this event influences the optimal condition. Taking alkali resistance as an example, if the *capability score* is higher when amino acid c_k appears at site k, it indicates that the occurrence of this event leads to an increase in the optimal pH of the sequence.

We also consider the correlations between the optimal condition and multiple amino acids appearing on different sites, namely how multiple elements together affect the optimal condition. We define the event (n_i^a, n_j^a) as c_i^a appears on site i and c_j^a appears on site j in the sequence $a \in \mathbf{F}$. We also introduce the *suitability score* $s(n_i^a, n_j^a)$ to denote how much this event induces a better optimal condition. Different with the single site analysis, the site k and the amino acid c_k^a on it are considered together by concatenating their vectors. A higher score means that the occurrence of event (n_i^a, n_j^a) induces a better optimal condition and the probability is formalized by equation 12.

To combine these two objectives together, a hyper parameter $\alpha \in (0..1)$ is introduced to balance them in the embedding learning process. Let $\mathcal{L}_T(\theta)$ and $\cdot\mathcal{L}_C(\theta)$ denote the objective functions for above two scores in the embedding

[2]A sequence logo is created from a collection of aligned sequences and is widely adopted to depict the consensus sequence and diversity of the sequences, such as protein-binding sites in DNA or functional units in proteins. http://weblogo.threeplusone.com

learning process, respectively. The whole objective function is then defined as $\mathcal{L}(\theta) = \alpha \cdot \mathcal{L}_T(\theta) + (1-\alpha) \cdot \mathcal{L}_C(\theta)$. A larger α induces a higher bias on $s(k,a)$. Conversely, a smaller α considers more on the correlations between the optimal condition and multiple amino acids appearing on different sites. We would discuss how to learn the embeddings in the Method section.

Dataset

We crawl the amino acid sequences of the GH11 family on the CAZy website[3]. In order to extract the required optimal pH of each sequence, we investigate the papers related to the proteins in the GH11 family on the *Web of Science* website. There are 272 amino acid sequences in the GH11 family and the length of each sequence is within 128 and 335. After alignment, the length of amino acid sequences is l=380.

Considering the conserved and non-conserved segments in the aligned sequences, we quantify the importance of each site against the optimal pH, namely which sites the amino acids being on highly influence the alkali resistance. We adopt the information gain as a metric. Give a sequence x, the information entropy $H(y)$ quantify the uncertainty on whether $x's$ optimal pH y is either higher or lower than the average, i.e. $H(y) = -\sum_{i\in\{l,h\}} p(x \in S_i) \cdot \log p(x \in S_i)$, where $S_l \subset F$ is the collection of sequences whose optimal pH are lower than the average, $S_h \subset F$ is the collection of sequences whose optimal pH are higher than the average, and $p(x \in S_i) = \frac{|S_i|}{|\mathbf{F}|}$. Then we quantify how much a concrete site reduces this uncertainty. For site k, the conditional entropy $H(y|c_k)$ quantities the uncertainty on the optimal pH after we know the amino acid c_k, i.e $H(y|c_k) = -\sum_{i\in\{l,h\}} p(x \in S_i|c_k) \cdot \log p(x \in S_i|c_k)$. The conditional entropy on site k is the expectation on different amino acids, i.e. $H(y|k) = \sum_{c_k \in A} \frac{|F_{c_k}|}{|\mathbf{F}|} \cdot H(y|c_k)$, where F_{c_k} denotes the set of sequences with the amino acid c_k on site k. The information gain on site k is $Gain(k) = H(y) - H(y|k)$. The higher this score, the more probability different amino acids on site k influences the optimal condition. The results on GH11 family are scatter plotted in Figure 1, where the X-axis denotes sites of aligned amino acid sequences and Y-axis denotes the information gain on a site. This point would be considered together with a concrete amino acid for mutation, discussed in next section.

Preprocess

We choose the higher alkali-resistance as the expected condition. Then the proteins with the optimal pH higher than 7 are classified into S_h and others into S_l. We generate samples in the following three steps: 1) Sample a protein $a \in S_h$ and a protein $b \in S_l$. 2) Select the site k where the amino acids in a and b are different, i.e. $c_k^a \neq c_k^b$. 3) Put (k, c_k^a) into the positive sample set and (k, c_k^b) into the negative sample set. When learning the embeddings, we select one or more samples from the two sets for each time.

Since the size of a family dataset is small, we adopt the approximation method to generate more samples by the statistics on amino acids in the training dataset. To understand how this method works, we compare the probability distribution on

[3]The CAZy database describes the families of structurally-related catalytic and carbohydrate-binding modules (or functional domains) of enzymes that degrade, modify, or create glycosidic bonds. http://www.cazy.org/

the training samples and all samples by the KL distance, an often adopted metric to evaluate two probability distributions [15],

$$KL\text{-}D(P_i \parallel Q_i) = \sum_{c \in A} P_i(c) \cdot \log \frac{P_i(c)}{Q_i(c)} \quad (1)$$

where P_i and Q_i denote the distribution on the amino acids at site i of an aligned sequence in two datasets, respectively. The smaller, the better.

We set the proportion 50%, 60%, 70%, 80%, and 90%, respectively, to randomly select samples as the training part. The comparison results against the proportion are listed in Figure 2, where X-axis represents the site and the Y-axis represents the KL distance. We can see that the difference decreases with an increasing size of training set. We would further analyze how this difference influences the computationally biological results in next section.

Results

In this section, we discuss how to use the learnt embeddings for biological purposes so as to help biologists for efficient analysis without wet experiments. We consider the important task on an enzyme mutant for an expected optimal condition, which is better than others in a family. A usual way is first to verify the optimal condition of each enzyme in a family by wet experiments and select the one most approaching the expected condition. Then they perform some mutation on the non-conserved segments and wet experiments to verify the activity of the mutant under the expected environment. These biological purposes can be transferred as the following two computational problems: the pH predication and mutation suggestion.

Sequence based optimal pH predication

Given two amino acid sequences in a family, we want to determine by computation which one has a better optimal condition, such as a higher temperature resistance. This result can be used to select an expected enzyme from a family by comparing each pair of them.

For two new amino acid sequences a and b, we calculate the *quantitative scores* of them, which is defined as the combination of *capability score* and *suitability score*, i.e. $Score(a) = \alpha \cdot \sum_{k=1}^{l} S(k, a_k) + (1-\alpha) \cdot \frac{\sum_{i,j=1}^{l} S(n_i^a, n_j^a)}{l}$, and would be also discussed in the Method Section. If $Score(a) > Score(b)$, we judge that a is more alkali-resistance than b or a has a higher activity than b in the same alkaline environment, and vice versa. Since the sites have different effects on the alkali-resistance of proteins, we select the site with the information gain larger than a threshold ω, which would be discussed in details in the experiments.

In addition to comparing the optimal pH between different amino acid sequences from the same family, we also use the embeddings to predict the optimal pH of amino acid sequences. We try different prediction models, and finally choose the Support Vector Regression(SVR) [17] method, where the kernel function is *rbf*, punishment term is 10, and the parameter gamma of *rbf* is 0.0001, as the prediction model. The input of the SVR model is the feature vector of the amino acid sequence and the output is the optimal pH.

Comparison methods and evaluation metric

We use the accuracy as the evaluation criteria. For two amino acid sequences selected from the test set, we compare their optimal pH. The accuracy is computed by the ratio of correct judgments against the number of independent experiments.

Since the current related works of predicting the optimal condition are based on the tertiary structure and/or free energy, there is not any closely related works on the sequence based predication. Therefore, we select several machine leaning algorithms as the comparison methods.

- SVR [17]. This is the traditional predication method. We try several kernel functions, such as rbf, $poly$, $linear$ and $sigmoid$, respectively, and adjust the parameters to verify the effects of the SVR. We select the best results from several variants. The corresponding parameters include: $poly$ as the kernel function, 10 as the punishment term C, 3 as the dimensions of poly functions.
- Neural Network(NN) [18]. There are many factors affect the results of the Neural Network method, such as the activation function, optimization function, the number of hidden layer and the number of nodes on the neural network model.We explore the effects on different settings and select the best one as: the number of nodes in hidden layer to 1000 in one hidden layer, $tanh$ as the activation function and $adam$ as the optimization function.

As a comparison to embeddings, we use the one-hot vector on the set aa of amino acids as the sample features to feed these methods. Namely, each site k of a sequence maps to a vector $x \in \{0,1\}^{|aa|}$ and $x_i = 1$ if and only if the amino acid c_k is at rank i in the lexicographical order of the set of amino acids. The gap maps to zero in a vector. For a given sequence $a \in F$, we predict the value of optimal pH $\hat{y}_a$ and calculate the residual by $\hat{y}_a - y_a$ with its real optimal pH y_a, which are shown in Figure 7.

From the results, we can see that the residuals in embedding model obeys normal distribution, and the variance is the smallest. Our method is better than the comparison methods in predicting the optimal pH.

Then we compare our method with other methods and show the results in Figure 8, together with several settings on the ratio between the training set and the test set, i.e. 5:5, 6:4, 7:3, 8:2 and 9:1. The threshold of the information gain ω is set 0.0478, The dimension of embeddings is set 30 and the preference of the combinative objectives α is set 0.4. We use the 10-fold cross-validation method to verify the accuracy.

Overall, the embedding method gives the best results in most cases comparing with other methods. When the ratio of training set to test set is 8:2, the accuracy is the highest by our method. Comparatively, NN is better than SVR, especially in predicting the optimal pH value for mutants which would be discussed in next subsection.

Considering the ratio of the training set and the test set, a larger ratio often induces a higher accuracy. This is because when the ratio is small, the training set can not reflect the probability distribution of the whole dataset. But a dominate ratio, i.e. 90%, may induce the number of test set too small such that the results are with more randomness and can not reflect representative results.

Experiment on the influence of parameters

In this subsection, we discuss the influence by the super parameter, including the dimensions of embeddings d, the preference of the combinative objectives α for embedding learning and the threshold of the information gain ω for selecting the sites on computation of score.

First, we discuss the influence of different dimensions of embeddings on the accuracy. We try several settings on the dimension of embeddings, i.e. d=5, 10, 30, 50, 90 and 130, respectively and list the result in Table 1

Table 1 Summary on the accuracy against different ratios on training data, the dimensions of embedding, and comparison methods.

train:test	Dimension of embedding vector					
	5	10	30	50	90	130
5:5	0.734	0.753	0.706	0.704	0.694	0.685
6:4	0.755	0.752	0.744	0.751	0.749	0.746
7:3	0.743	0.77	0.787	0.792	0.794	0.794
8:2	0.74	0.779	**0.807**	0.800	0.789	0.787
9:1	0.745	0.773	0.787	0.794	0.799	0.799

Considering the influence by the dimension of embeddings, we verify the accuracy against different settings. We can see that after the dimension is larger than a threshold 30, there is not obvious difference on accuracy. This illustrates that a small dimension cannot hold much implicit information on amino acids and the sites of an aligned sequence. However, a too large dimension may induce the sparse problem and is difficult for convergence due to a small quantity of samples.

The embedding dimension in following experiments is set 30, and the ratio between training set to test set is 8:2. We choose the threshold ω as 0.111, 0.0478, 0.026, 0.0105 and 0.002, and α as 0, 0.1, 0.2, 0.3,..., 1, respectively.

The accuracy results against different parameters, i.e. the balance factor α and the threshold ω on information gain, are shown in Figure 9, where the X-axis represents the values of α, the Y-axis represents the accuracy of our method. It achieves the highest accuracy at $\omega = 0.0478$ and $\alpha = 0.4$.

Considering the threshold of the information gain ω for selecting the sites on computing the score, a suitable ω is required. The best setting is at $\omega = 0.0478$, colored red. A large threshold on the information gain ω induces fewer sites being selected and some important non-conserved sites being ignored. Conversely a small ω may introduce more conserved sites in calculating the score as Eq.2 such that it reduces the accuracy.

Considering the preference on α, both the type of amino acids on single site and interactions between multiple sites contribute to accuracy. When α is at 0.4, i.e. the type of amino acids on single site, our method achieves the best. When α is less than 0.4, our optimization objectives would consider more interaction of different sites. The special case $\alpha = 0$ learnt less about the semantics on amino acids, which only considers the correlations between the optimal condition and multiple amino acids appearing on different sites. However a larger $\alpha > 0.4$ does not influence much on predication.

Bioinformatics verification

In this subsection, we verify the predication results with the biological wet experiments. We take the enzyme $Xylanase\ A$ from the GH11 family as an example, whose optimal pH is 6.0. Ruller, Alponti and Deliberto et al. generate 5 mutants on $Xylanase\ A$ which can survive in alkaline environment by multiple mutation experiments and determine their activity at 5.5 pH environment by wet experiments [19]. We use Eq.15 to calculate the activity scores of $Xylanase\ A$ and the 5 mutants. The higher the score, the higher the activity of the enzyme at the same pH environment.

Table 2 The activity at 5.5pH and the activity score on $Xylanase\ A$ and Mutant.

Wild and Mutants	Mutation Sites	Activity at 5.5 pH	Score
Wild Xylanase A	N/A	2.27U/ml	4022.35
Mutant1	S22P	**3.58U/ml** ↑	**4024.73** ↑
Mutant2	G13R	1.89U/ml ↓	4018.68 ↓
Mutant3	Q7H/G13R/S22P	**2.45U/ml** ↑	**4022.88**↑
Mutant4	S22P/H156L/S179C	**3.01U/ml** ↑	**4024.73**↑
Mutant5	S31Y	2.20U/ml ↓	4022.35

Both the activity at 5.5 pH [19] and the scores of these enzymes calculated by our method are together listed in Table 2. By comparing the changes in activity and scores between mutants and wild-type $Xylanase\ A$, we can see that the trend of activity are consistent with the scores. These are positive correlations.

A special case is mutant5, which may be caused by the mutation on site 31. Since the information gain on this site is only 0.039 and is lower than the threshold 0.0478, it is not included in calculating the score. But considering the little change in activity after mutation, it is not important to select this mutant for an expected optimal condition.

Embeddings based mutation suggestion

In this subsection, we discuss how to guide mutations on enzymes so as to help biologists wet experiments. Given a family of enzymes and an expected optimal condition like alkali-resistance, we design a new amino acid sequence such that it has a better optimal condition than the wild-type enzymes.

For a given wild-type enzyme, we propose two rules for a single-site mutation: 1) Do not mutate the site that is filled by gap $'-'$ so as to remain the original length of the amino acid sequence. 2) Do not mutate the sites with information gain smaller than 0.0478. Then we calculate the scores of different single-site mutations and select the mutant with the highest score. For multiple-sites mutation, we combine single-site mutations to get multi-sites mutants on a wild-type enzyme, and retain the mutant with the highest score.

We perform single-site and multi-sites mutations on the known wild-type amino acid sequences in GH11 family. From a biological point of view, the more mutation sites, the greater the uncertainty of the mutation and the higher the cost of mutation. So we only do at most 3-sites mutation for amino acid sequences. We use Neural Network model to predict the optimal pH of mutants, and calculate the change of optimal pH after mutation. The results are shown in Figure 10, where the

X-axis represents the change of optimal pH and the Y-axis represents the number of amino acid sequence under the change.

For both mutations, the optimal pH of the 88% amino acid sequence increases. From the prediction results, both mutation suggestions can increase the optimal pH of amino acid sequences and the effect of multi-sites mutation suggestion is better than that of single-site mutation suggestion. Specially, The optimal pH of the 5% amino acid sequence in single-site mutation and 10% amino acid sequence in multi-sites mutation increase more than 1.0.

Discussion

In this proposed method, the learnt embeddings are expected to include the biological semantics for an expected optimal condition. Since the embedding vectors are in the latent space, we adopt the Distributed Stochastic Neighbor Embedding (*t-SNE*) method [16] to reduce the high-dimension vectors to 2d coordinates for illustration.

We first verify whether the capability of amino acids with the sites of a sequence is embedded in the vectors. The learned *capability scores* $s(k, a)$ are illustrated as the thermodynamic chart in Figure 3, where the X-axis denotes the sites and Y-axis denote the amino acids. Red color maps to high scores, while green color maps to low scores. The obviously green areas 1-20, 157-166, 215-238, 250-254 and 366-380 indicate amino acids on these sites influence less the optimal pH, while the red areas indicate that different amino acids on these sites highly influence the optimal pH. Compared with the results by biologist analysis, these areas are mapped to the conserved segments and the potential non-conserved segments, which is the basic task for analyzing the enzymes in a family. This convinces the semantics of embeddings.

Then we verify the biological differences between amino acids. We select Top10 information gain sites and two alkali-resistance amino acids *Lys* and *Arg*, as well as two acid-resistances amino acids *Asp* and *Glu*. We combine the embeddings of each pair of them as Eq.10 and color them as points in Figure 4 by $t - SNE$. The results show that the points that are combined with the same amino acid are clustered together rather than others. Moreover, the points combined with the alkali-resistance amino acids, colored green and pink are clustered more closely than with the acid-resistance amino acid, colored blue and red.

At last we compare the sites with different information gain, the top2 sites 278, 80 and the bottom2 sites 1 and 380. The combinative embeddings by sites and amino acids, as Eq.10, are colored differently in Figure 5, where they are obviously clustered into two groups: sites with high information gain and sites with low information gain. We zoom in the details on site 80 and 278, and the results in Figure 6 show that the pink and purple points co-exist in pairs. The points related to alkali-resistance amino acids are clustered together, while the points related acid-resistance amino acids are clustered together. Comparatively, the points on alkali-resistance amino acids are closer than the acid-resistance amino acids. This illustrates that the embeddings have learned much information on distinguishing the amino acids and their influences on alkali-resistance functions.

As for future work, we plan to analyze the optimal conditions of proteins in different families. Since they have different protein structures and aligned sequences,

it would be more complex and challenging to analyze their conserved and non-conserved sequences and learning the embeddings of sites. Another interesting direction is to analyze multiple optimal conditions together. For different optimal conditions, the effectiveness of sites and amino acids in a protein may be different. The multi-objective analysis would help us find the relationships between different optimal conditions. This requires an elaborate design on the embeddings method so that they can contain more information to distinguish the features of different optimal conditions.

Conclusion

In this paper, we proposed an embedding method to represent the amino acids and the construct information as vectors in the latent space. These vectors contain information about the correlations between amino acids and sites in aligned amino acid sequences. Based on these embeddings, we then predict the optimal pH of a new amino acid sequence in the same family and design a method to suggest the site and direction of a mutation for an expected condition by embeddings. We adopt the amino acid sequences in glycoside hydrolase GH11 family for the verification of our method. And we design two computational experiments and verify the results by wet experiments, which are to predict the optimal pH of amino acid sequences and to give a mutation suggestion. Compared with the traditional method, this method does not require the tertiary structure of a protein or the situation of free energy of the amino acid sequence, which is more efficient and effective.

These advantages attribute to two aspects. One is that we take into account both the correlations between the amino acid at a single site and the interaction between multiple-sites. Another is that the vectors we have learnt reveal the information about the optimal condition implied in amino acids sequence.

Method

In this section, we present the details on the embedding method and the learning process.

The embedding objectives

The optimal condition of an amino acid sequence is determined by which amino acids consist the enzyme and how they construct together in the sequence. Motivated by this point, we introduce two objective functions to learn the correlations between the optimal condition and the sequence: 1) the influence by one amino acid on each site. 2) the mutual influences by two amino acids on different sites. We propose an embedding method to combine these two objectives together by modeling the sites and the amino acids of a sequence into the vectors in the same latent space, denoted by $\mathbf{v}, \mathbf{c} \in R^d$, respectively.

The first objective learns the correlations between the optimal condition and the type of amino acid on each site. We define the event (k, a) as amino acid $c_k^a \in A$ being on the $k^{th}, k \in [1..l]$ site of an aligned sequence $a \in \mathbf{F}$, and introduce the metric of *capability score* $s(k, a)$ to evaluate how this event influences the optimal condition, formalized by:

$$s(k,a) = \mathbf{v_k} \cdot \mathbf{c_k^a} \tag{2}$$

where $\mathbf{v}_k, \mathbf{c}_k^a \in R^d$ are the embeddings of site k and the amino acid c_k^a, respectively. A higher score means c_k^a on site k resulting the sequence a better activity under a given condition.

Let $\mathbf{E}$ be the set of all occurrences of different events, namely the combination of amino acids and sites. We adopt the *softmax* function to model the probability of such an event.

$$p_\theta(k,a) = \frac{\exp(s(k,a))}{\sum_{(k',a')\in\mathbf{E}} \exp(s(k',a'))} \tag{3}$$

where θ denotes the parameters in this model. To maximize the likelihood of the occurrences, the objective loss function is formalized by

$$\mathcal{L}_T(\theta) = -\sum_{(k,a)\in\mathbf{E}} \log p_\theta(k,a) \tag{4}$$

Since the normalization part in the denominator in Eq.3 cost much computation, we use the Noise Contrastive Estimation (NCE) method, proposed by Gutmann et al. [20], to estimate the optimal parameters θ^*. It treats the normalization part as an additional parameter, denoted by C. The Eq.3 is then re-written as:

$$p_\theta(k,a) = \exp(s(k,a) + C) \tag{5}$$

According to NCE, we add artificially generated noise data to the training set. The parameters in probability density function and normalization part can be estimated by discriminating the original data and noise data. Let $p(D=1|(k,a),\theta)$ denote the probability that the optimal condition gets higher when $c_k^a \in A$ appears on site k in sequence $a \in \mathbf{F}$. Let $p(D=0|(k,a),\theta)$ denote the probability that the optimal condition gets lower when c_k^a appears on site k.

$$\begin{aligned} p(D=1|(k,a),\theta) &= \frac{p_\theta(k,a)}{p_\theta(k,a) + p_n(k,a)} \\ &= \sigma(\log p_\theta(k,a) - \log p_n(k,a)) \end{aligned} \tag{6}$$

$$\begin{aligned} p(D=0|(k,a),\theta) &= 1 - p(D=1|(k,a),\theta) \\ &= 1 - \sigma(\log p_\theta(k,a) - \log p_n(k,a)) \end{aligned} \tag{7}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the *sigmoid* function and $p_n(k,a)$ is the artificial noise distribution.

We fit the model by maximizing the expectation of log-posterior probability over the mixture of observed samples and noise samples. The expectation and loss function are formulated by Eq.8 and Eq.9:

$$E_{p_\theta}[\log p(D=1|(k,a),\theta)] + E_{p_n}[\log p(D=0|(k,a),\theta)] \tag{8}$$

$$\begin{aligned}\mathcal{L}_T(\theta) = -[&\log\sigma(\log p_\theta(k,a) - \log p_n(k,a)) \\ &+ \log(1-\sigma(\log p_\theta(k',a') - \log p_n(k',a')))]\end{aligned} \tag{9}$$

The second embedding objective learns the correlations between the optimal condition and multiple amino acids appearing on different sites, namely how multiple elements together affect the optimal condition. Different with the single site analysis, the site k and the amino acid c_k^a on it are considered together by concatenating their vectors,

$$\mathbf{n_k^a} = \mathbf{v_k} \oplus \mathbf{c_k^a} \tag{10}$$

where $\mathbf{v}_k, \mathbf{c}_k^a \in R^d$ are the embeddings of site k and the amino acid c_k^a, respectively, and $\mathbf{n}_k^a \in R^{2d}$ is their joint embedding. We define the event (n_i^a, n_j^a) as c_i^a appears on site i and c_j^a appears on site j in the sequence $a \in \mathbf{F}$. We also introduce the *suitability score* $s(n_i^a, n_j^a)$ to denote how much this event induces a better optimal condition.

$$s(n_i^a, n_j^a) = \mathbf{n_i^a} \cdot \mathbf{n_j^a} = (\mathbf{v_i} \oplus \mathbf{c_i^a}) \cdot (\mathbf{v_j} \oplus \mathbf{c_j^a}) \tag{11}$$

A higher score means that the occurrence of event (n_i^a, n_j^a) induces a better optimal condition and the probability is formalized by equation 12.

$$p_\theta(n_i^a, n_j^a) = \frac{\exp(s(n_i^a, n_j^a))}{\sum_{(i',j')\in\mathbf{E}} \exp(s(n_{i'}^a, n_{j'}^a))} \tag{12}$$

Similar to the discussion of Eq.5 to Eq.9, the loss function is formulated as follow:

$$\begin{aligned}\mathcal{L}_C(\theta) = -[&\log\sigma(\log p_\theta(n_i^a, n_j^a) - \log p_n(n_i^a, n_j^a)) \\ &+ \log(1-\sigma(\log p_\theta(n_i^{a\prime}, n_j^{a\prime}) - \log p_n(n_i^{a\prime}, n_j^{a\prime})))]\end{aligned} \tag{13}$$

We model the loss function as a linear combination of the above two loss functions, where $\alpha \in [0,1]$ is a preference parameter.

$$\mathcal{L}(\theta) = \alpha \cdot \mathcal{L}_T(\theta) + (1-\alpha) \cdot \mathcal{L}_C(\theta) \tag{14}$$

Through the above introduction, we define two score functions $s(k,a)$ and $s(n_i^a, n_j^a)$. Based on two score functions, we introduce a *quantitative score* as a specific optimal condition for a specific sequence $a \in F$. It considers two parts: the compatibility of the amino acid on each site and the influence induced by multiple amino acids on different sites, which are combined by the preference parameter α, defined in Eq.14.

$$Score(a) = \alpha \cdot \sum_{k=1}^{l} S(k, a_k) + (1-\alpha) \cdot \frac{\sum_{i,j=1}^{l} S(n_i^a, n_j^a)}{l} \tag{15}$$

The embedding learning process

We adopt the stochastic gradient descent algorithm to optimize the parameters, proposed by *Adam* [21]. Given the set of amino acid sequences in a family, we partition them into two sets: one for training and another for test. For the expected optimal condition, the training set is further classified into the high optimal condition set S_h and the low optimal condition set S_l. For example, if a higher pH is expected as the optimal condition, the amino acid sequences in S_h have higher pH than those in S_l. We select a sequence a from S_h and the amino acid c_k^a on site i of a as a positive sample, i.e. the pair (i, a). A negative sample (j, b) is similarly chosen by site j on $b \in S_l$ so as to calculate the gradient of parameters. Taking Eq.9 as an example, the gradient of the embedding $\mathbf{v_i}$ of site i in the loss function is calculated by:

$$\frac{\partial \mathcal{L}_T}{\partial \mathbf{v_i}} = -[\sigma(\log p_\theta(i, a) - \log p_n(i, a)) - 1]\mathbf{c_i^a} \tag{16}$$

To calculate Eq.13, we need two positive samples (i, a), (j, a) to compose event (n_i^a, n_j^a), and two negative samples (i', b), (j', b) to compose event $(n_{i'}^b, n_{j'}^b)$.

$$\frac{\partial \mathcal{L}_C}{\partial \mathbf{v_i}} = -[\sigma(\log p_\theta(n_i^a, n_j^a) - \log p_n(n_i^a, n_j^a)) - 1]\mathbf{n_j^a} \tag{17}$$

The learning process would stop until the objective function is converged.

Abbreviations

GH11: Glycoside Hydrolase Family 11; CAZy: Carbohydrate-Active enzymes; NCE: Noise Contrastive Estimation; KL Distance: Kullback Leibler Distance; t-SNE: Distributed Stochastic Neighbor Embedding; SVR: Support Vector Regression; NN: Neural Network

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and material

All data used in the experiments are from CAZy database.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The scientific calculations in this paper have been done on the HPC Cloud Platform of Shandong University. This work was supported by the National Key R&D Program of China (2018YFB0204000,2018YFC0831401), the Major Project of NSF Shandong Province (ZR2018ZB0420), and National Natural Science Foundation of China(91646119,31770054), and the Key Research and Development Program of Shandong Province (2017GGX10114).

Author's contributions

YS is responsible for model building, paper structure editing, and proofreading. LW is responsible for the work guidance and planning of the biological direction, explaining the biological significance of the experiment. BG is responsible for the experimental code structure planning and proofreading work. XL is responsible for experimental work, data analysis, paper writing, and proofreading. ZD prepared raw data, wrote and edited the manuscript, and interpreted biological meaning of experimental results. All authors read and approved the final manuscript.

Funding

This work was supported by the National Key R&D Program of China (2018YFB0204000,2018YFC0831401), the Major Project of NSF Shandong Province (ZR2018ZB0420), and National Natural Science Foundation of China(91646119,31770054), and the Key Research and Development Program of Shandong Province (2017GGX10114).

The funding body played no role in the design of the study, the collection, analysis, and interpretation of the data or in writing of the manuscript.

Author details
[1]School of Software, Shandong University, Shunhua Road, 250101 Jinan, China. [2] State Key Laboratory of Microbial Technology, Shandong University, Binhai Road, 266237 Qingdao, China.

References

1. Marks, D.S., Hopf, T.A., Sander, C.: Protein structure prediction from sequence variation. Nature biotechnology **30**(11), 1072 (2012)
2. Paës, G., Berrin, J.-G., Beaugrand, J.: Gh11 xylanases: structure/function/properties relationships and applications. Biotechnology advances **30**(3), 564–592 (2012)
3. Wu, X., Tian, Z., Jiang, X., Zhang, Q., Wang, L.: Enhancement in catalytic activity of aspergillus niger xynb by selective site-directed mutagenesis of active site amino acids. Applied microbiology and biotechnology **102**(1), 249–260 (2018)
4. Lin, H., Chen, W.: Prediction of thermophilic proteins using feature selection technique. Journal of microbiological methods **84**(1), 67–70 (2011)
5. van Dijk, E., Hoogeveen, A., Abeln, S.: The hydrophobic temperature dependence of amino acids directly calculated from protein structures. PLoS computational biology **11**(5), 1004277 (2015)
6. Pucci, F., Rooman, M.: Stability curve prediction of homologous proteins using temperature-dependent statistical potentials. PLoS computational biology **10**(7), 1003689 (2014)
7. Pucci, F., Kwasigroch, J.M., Rooman, M.: Scoop: an accurate and fast predictor of protein stability curves as a function of temperature. Bioinformatics **33**(21), 3415–3422 (2017)
8. Kc, D.B.: Recent advances in sequence-based protein structure prediction. Briefings in bioinformatics **18**(6), 1021–1032 (2016)
9. Liu, S.: Importance of improving scoring methods in predicting protein free-energy changes, 600–603 (2012)
10. Malinka, F.: Prediction of protein stability changes upon one-point mutations using machine learning. In: Proceedings of the 2015 Conference on Research in Adaptive and Convergent Systems, pp. 102–107 (2015). ACM
11. Dehouck, Y., Kwasigroch, J.M., Gilis, D., Rooman, M.: Popmusic 2.1: a web server for the estimation of protein stability changes upon mutation and sequence optimality. BMC bioinformatics **12**(1), 151 (2011)
12. Wijma, H.J., Floor, R.J., Jekel, P.A., Baker, D., Marrink, S.J., Janssen, D.B.: Computationally designed libraries for rapid enzyme stabilization. Protein Engineering, Design and Selection **27**(2), 49–58 (2014)
13. Crooks, G.E., Hon, G., Chandonia, J.-M., Brenner, S.E.: Weblogo: a sequence logo generator. Genome research **14**(6), 1188–1190 (2004)
14. Schneider, T.D., Stephens, R.M.: Sequence logos: a new way to display consensus sequences. Nucleic acids research **18**(20), 6097–6100 (1990)
15. Zhang, X., Zou, G., Carroll, R.J.: Model averaging based on kullback-leibler distance. Statistica Sinica **25**, 1583 (2015)
16. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
17. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST) **2**(3), 27 (2011)
18. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press, ??? (2016)
19. Ruller, R., Alponti, J., Deliberto, L.A., Zanphorlin, L.M., Machado, C.B., Ward, R.J.: Concommitant adaptation of a gh11 xylanase by directed evolution to create an alkali-tolerant/thermophilic enzyme. Protein Engineering, Design & Selection **27**(8), 255–262 (2014)
20. Gutmann, M., Hyvärinen, A.: Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. Journal of Machine Learning Research **9**, 297–304 (2010)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. Computer Science (2014)

Figure Legends

Figure 1 Information gain for the sites in aligned amino acid sequences in GH11 family.

Figure 2 Evaluation on the approximate method

Figure 3 The thermodynamic chart for the capability scores on PH for the GH11 family. The X-axis represents the aligned 380 sites of sequences and the Y-axis represents amino acids and the gap in set A.

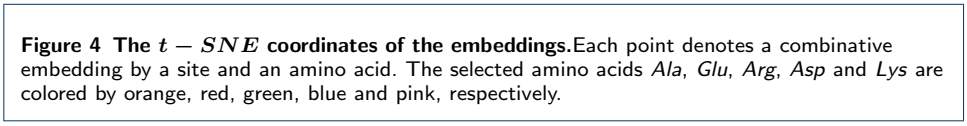

Figure 4 The $t-SNE$ coordinates of the embeddings. Each point denotes a combinative embedding by a site and an amino acid. The selected amino acids *Ala*, *Glu*, *Arg*, *Asp* and *Lys* are colored by orange, red, green, blue and pink, respectively.

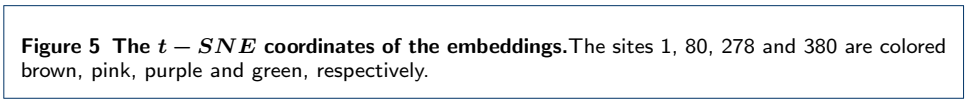

Figure 5 The $t-SNE$ coordinates of the embeddings. The sites 1, 80, 278 and 380 are colored brown, pink, purple and green, respectively.

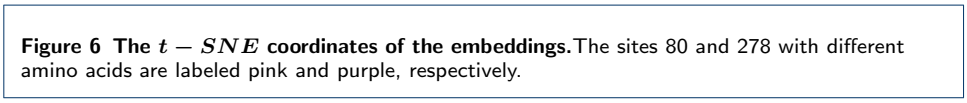

Figure 6 The $t-SNE$ coordinates of the embeddings. The sites 80 and 278 with different amino acids are labeled pink and purple, respectively.

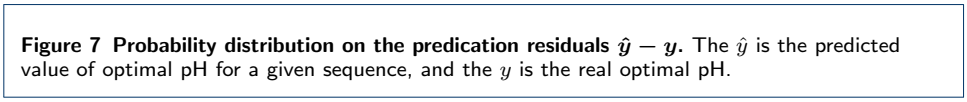

Figure 7 Probability distribution on the predication residuals $\hat{y}-y$. The $\hat{y}$ is the predicted value of optimal pH for a given sequence, and the y is the real optimal pH.

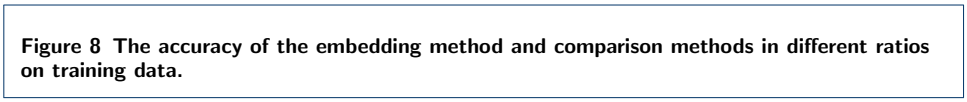

Figure 8 The accuracy of the embedding method and comparison methods in different ratios on training data.

Figure 9 The accuracy against different parameter settings. Each curve represents the accuracy under the given threshold ω.

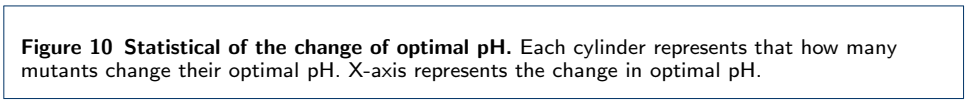

Figure 10 Statistical of the change of optimal pH. Each cylinder represents that how many mutants change their optimal pH. X-axis represents the change in optimal pH.

Figures

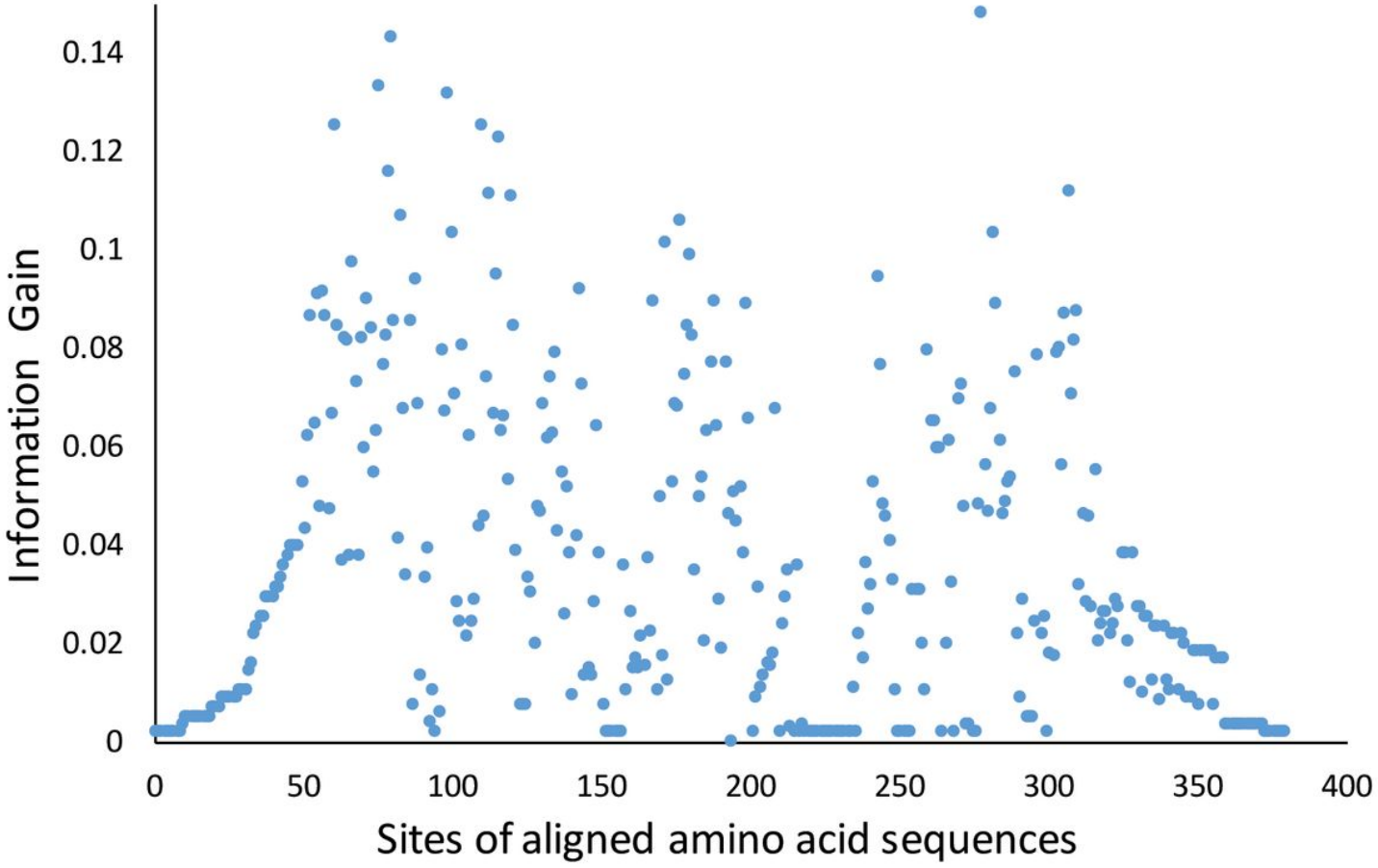

Figure 1

Information gain for the sites in aligned amino acid sequences in GH11 family.

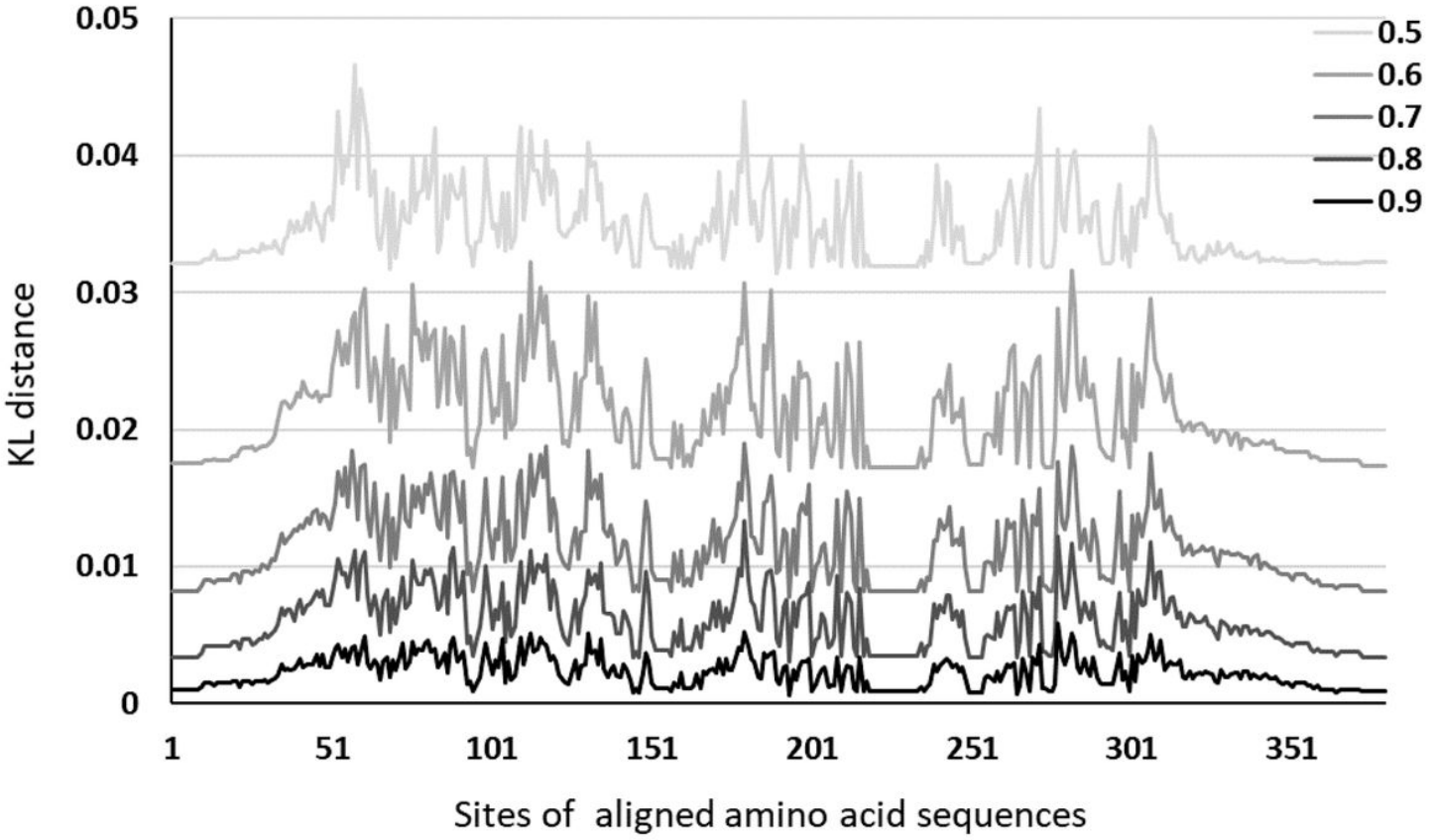

Figure 2

Evaluation on the approximate method

Figure 3

The thermodynamic chart for the capability scores on PH for the GH11 family. The X-axis represents the aligned 380 sites of sequences and the Y-axis represents amino acids and the gap in set A.

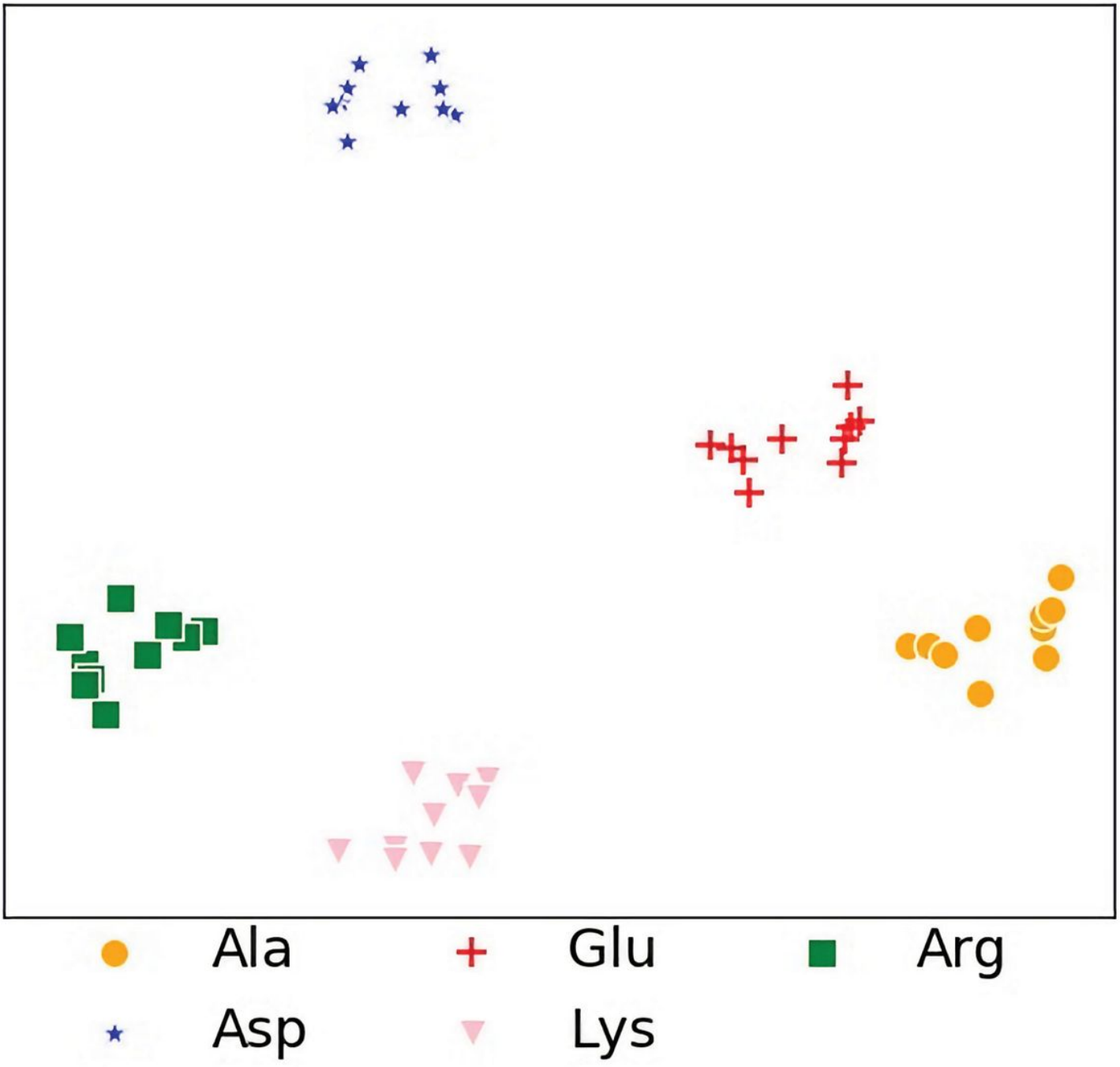

Figure 4

The t-SNE coordinates of the embeddings. Each point denotes a combinative embedding by a site and an amino acid. The selected amino acids Ala, Glu, Arg, Asp and Lys are colored by orange, red, green, blue and pink, respectively.

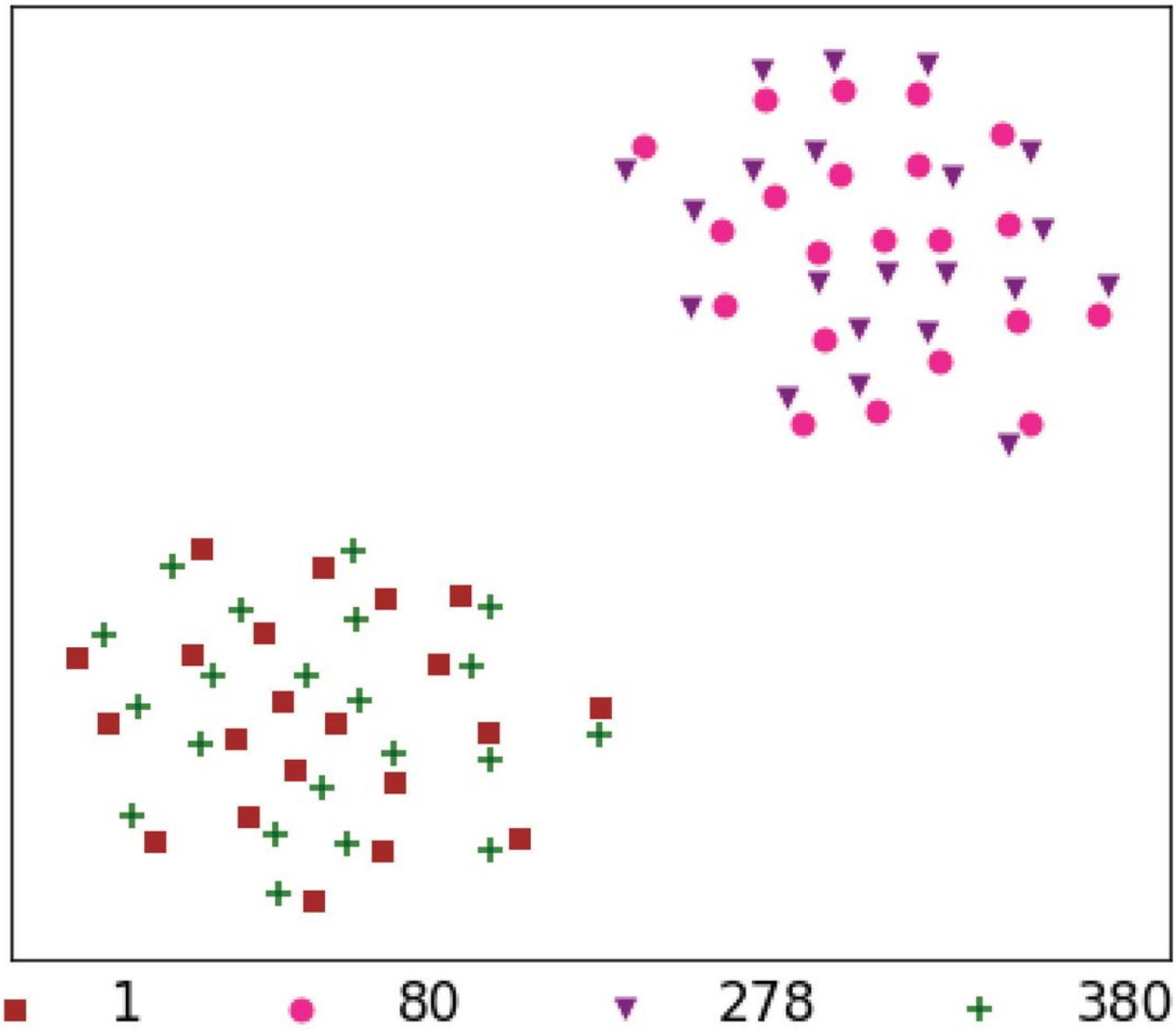

Figure 5

The t-SNE coordinates of the embeddings.The sites 1, 80, 278 and 380 are colored brown, pink, purple and green, respectively.

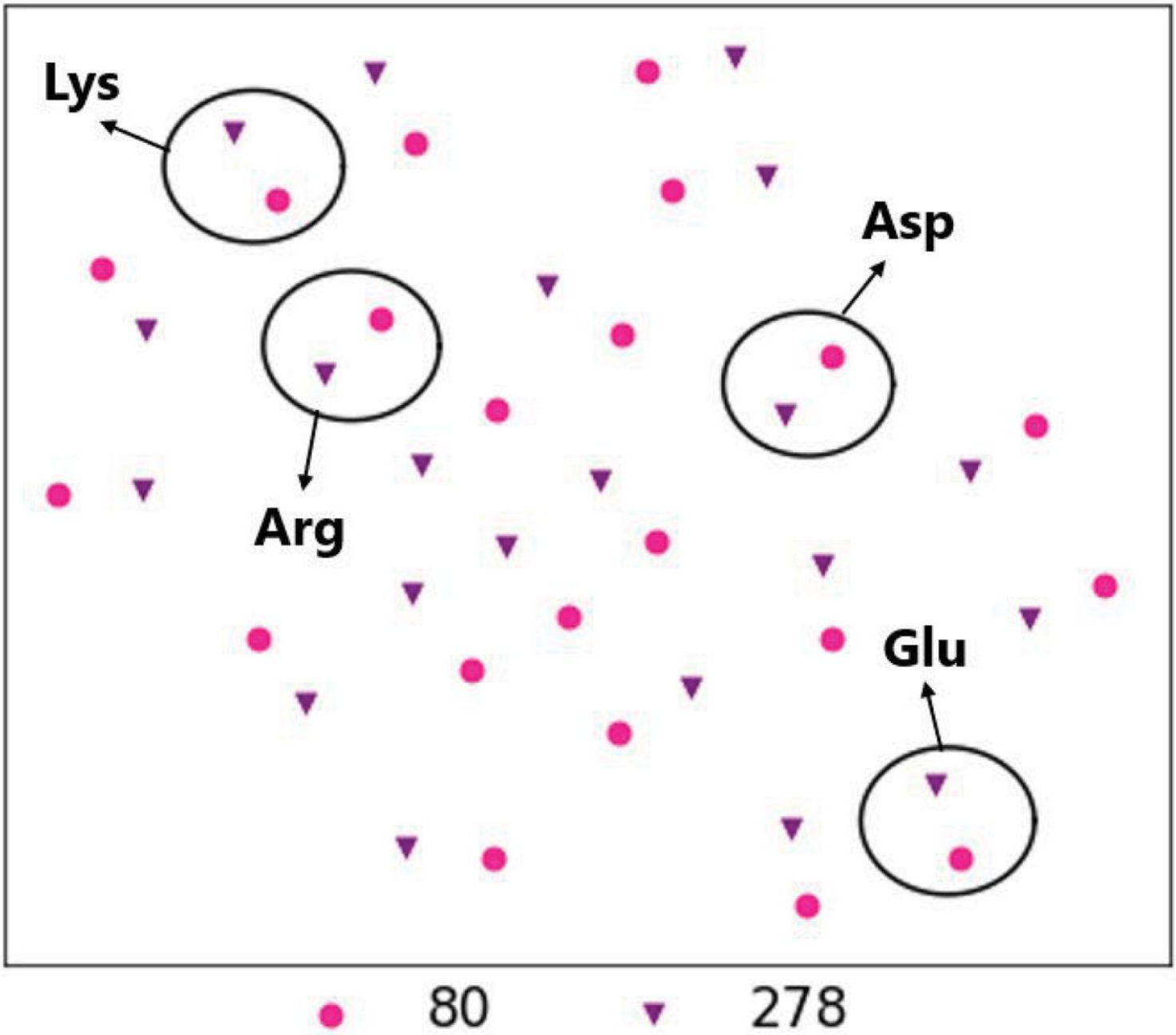

Figure 6

The t - SNE coordinates of the embeddings.The sites 80 and 278 with different amino acids are labeled pink and purple, respectively.

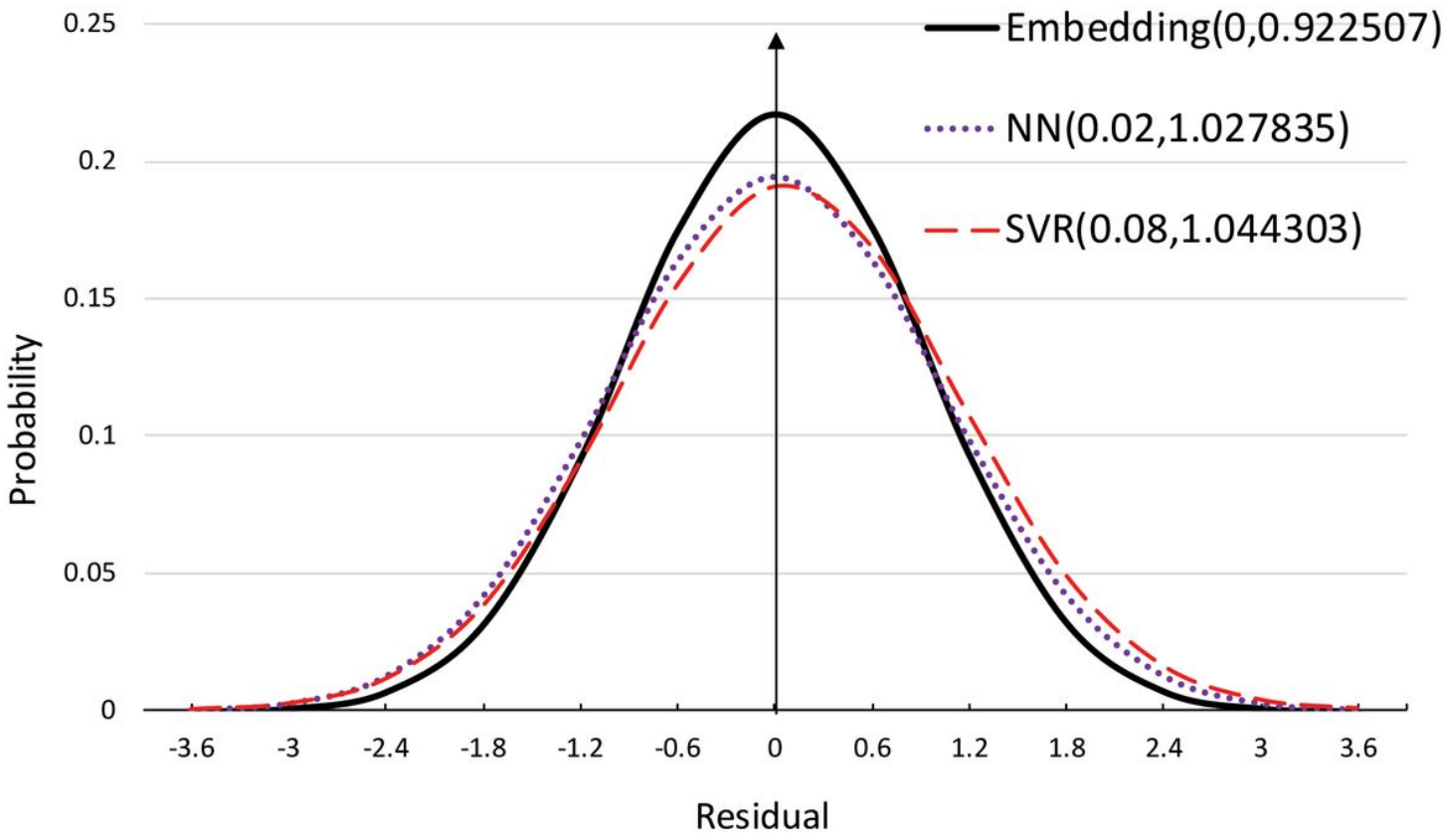

Figure 7

Probability distribution on the predication residuals ^y - y. The ^y is the predicted value of optimal pH for a given sequence, and the y is the real optimal pH.

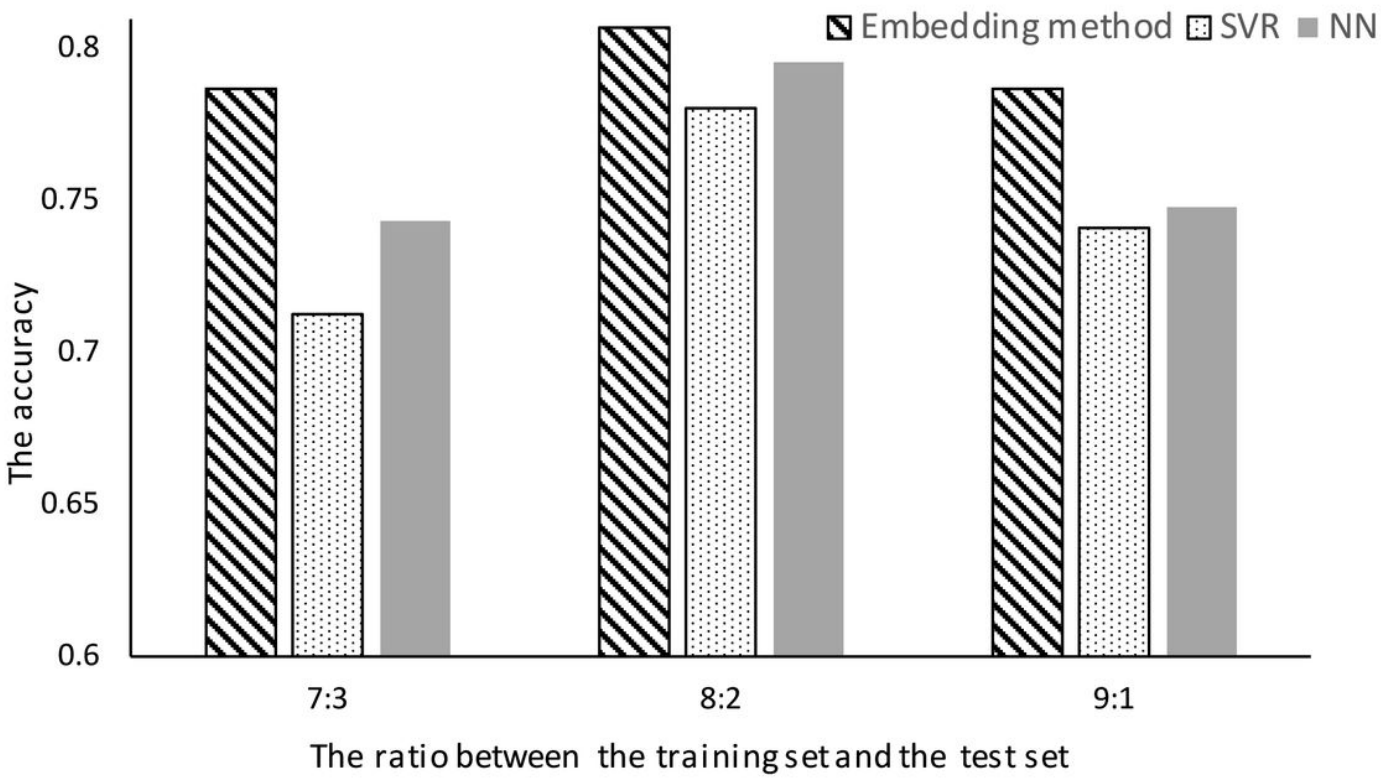

Figure 8

The accuracy of the embedding method and comparison methods in different ratios on training data.

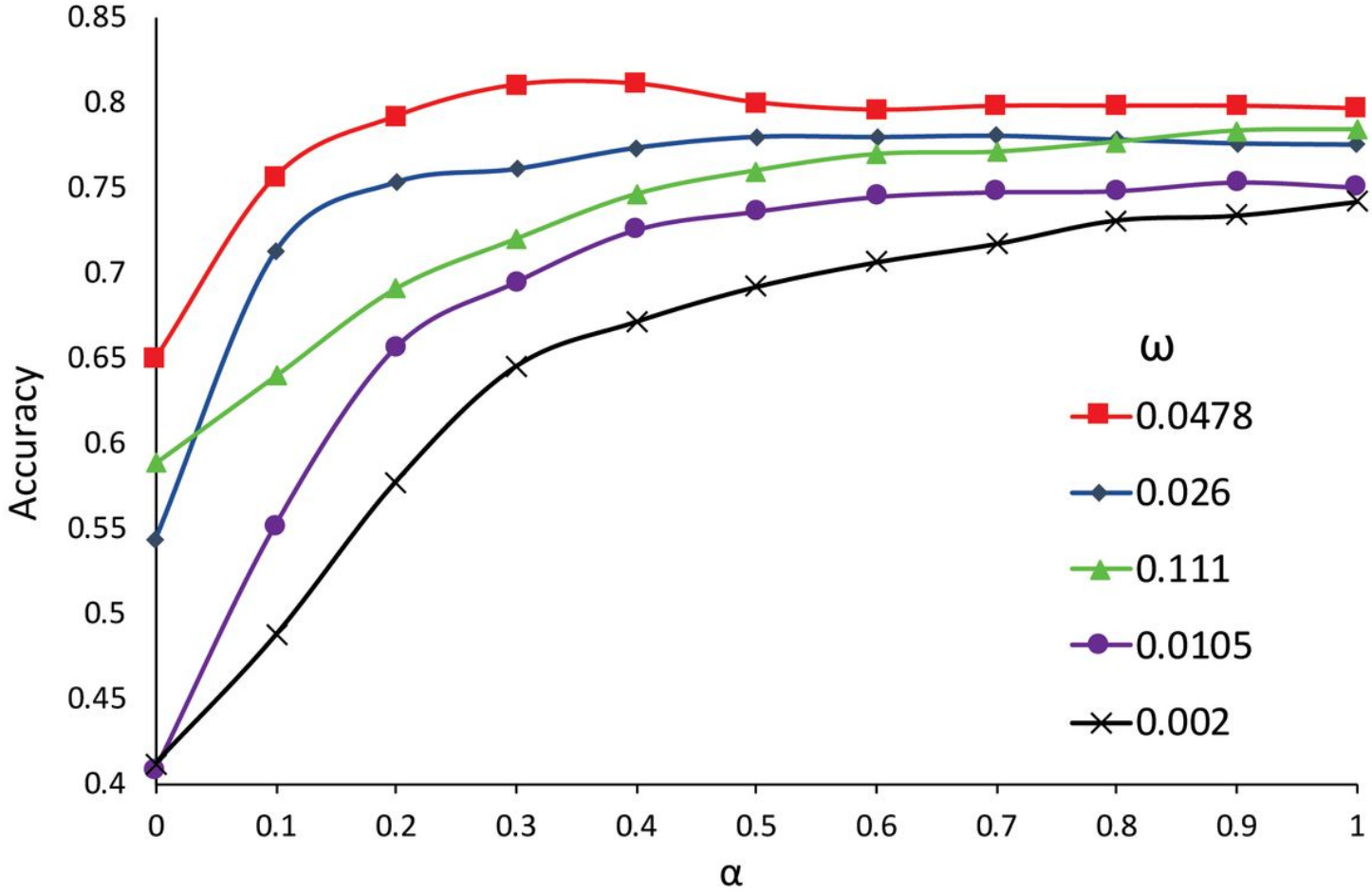

Figure 9

The accuracy against different parameter settings. Each curve represents the accuracy under the given threshold w

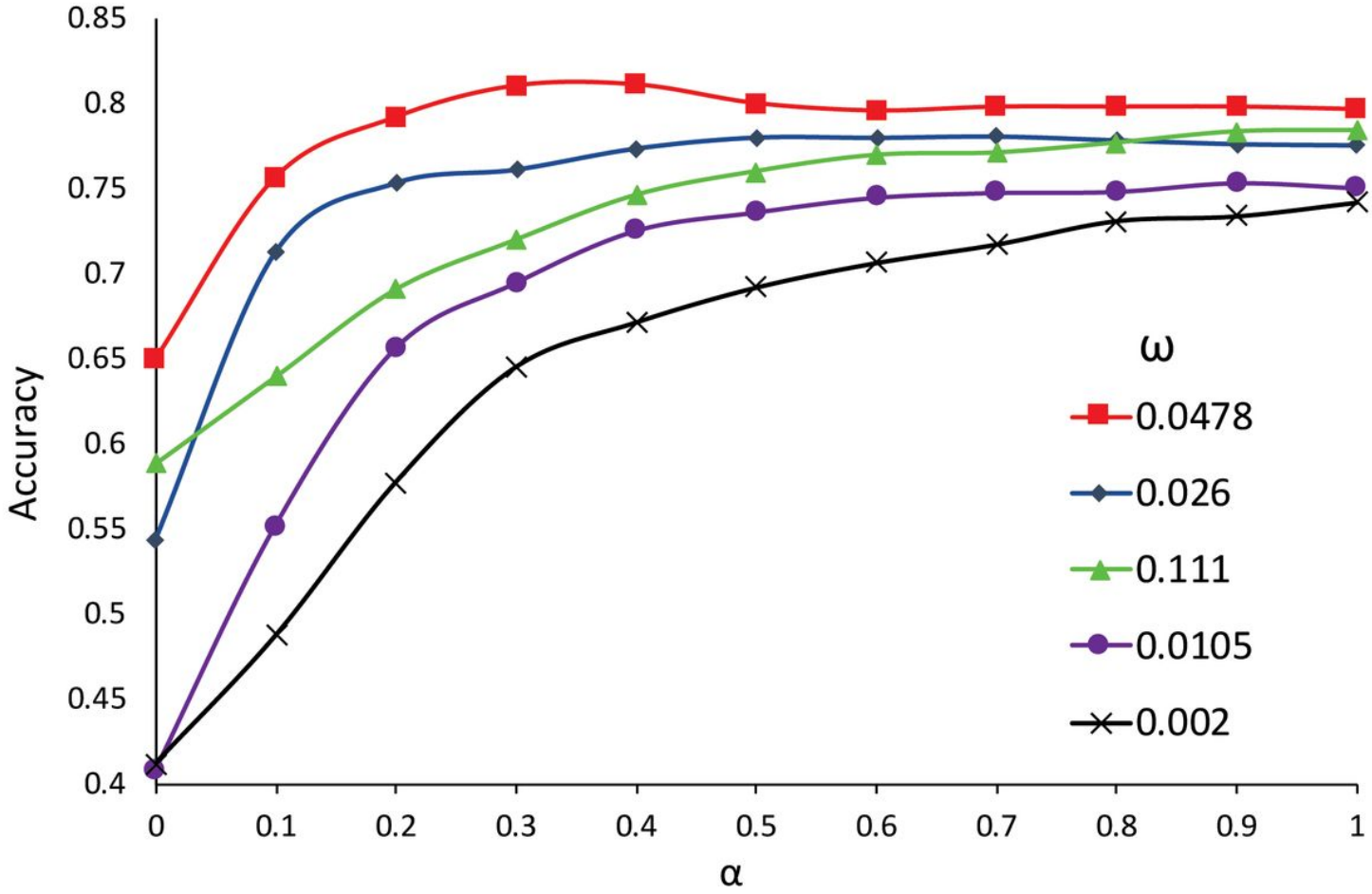

Figure 10

Statistical of the change of optimal pH. Each cylinder represents that how many mutants change their optimal pH. X-axis represents the change in optimal pH.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- bmcarticle.bib
- bmcarticle.log
- bmcarticle.bbl
- vancouver.bst
- bmcartbiblio.sty
- bmcmathphys.bst
- bmcsubmit20191021.tex
- bmcart.cls
- spbasic.bst